

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
MODUL 7
PENCARIAN NILAI EKSTRIM PADA HIMPUNAN DATA**



Oleh:

NAMA : SETYO NUGROHO

NIM : 103112400024

KELAS : S1IF-12-01

S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025

I. DASAR TEORI

Pencarian nilai ekstrim (nilai terkecil dan terbesar) dalam kumpulan data adalah salah satu konsep dasar dalam pemrograman dan algoritma. Dalam modul ini, kita akan membahas dua aspek utama: **pencarian nilai ekstrim** dan **penghitungan rerata**. Keduanya merupakan operasi penting dalam pengolahan data numerik.

Pencarian Nilai Ekstrim (Minimum dan Maximum)

Pencarian nilai ekstrim adalah proses menemukan nilai terkecil dan terbesar dalam kumpulan data. Ada dua cara umum untuk melakukan pencarian nilai ekstrim dalam array atau daftar:

- **Pencarian Nilai Minimum (Minimum Search):** Pencarian nilai terkecil dari sebuah kumpulan data.
- **Pencarian Nilai Maximum (Maximum Search):** Pencarian nilai terbesar dari sebuah kumpulan data.

Algoritma Pencarian Nilai Ekstrim

Untuk mencari nilai ekstrim pada array atau daftar data, algoritma yang umum digunakan adalah algoritma **linear search**. Berikut adalah Langkah langkah algoritmanya:

1. Tentukan elemen pertama dalam array sebagai nilai ekstrim (baik itu nilai maksimum atau minimum).
2. Iterasi melalui seluruh elemen array (dimulai dari elemen kedua) dan bandingkan setiap elemen dengan nilai ekstrim saat ini.
3. Jika ditemukan nilai yang lebih kecil (untuk mencari minimum) atau lebih besar (untuk mencari maksimum), perbarui nilai ekstrim tersebut.
4. Setelah iterasi selesai, nilai ekstrim yang tersisa adalah nilai minimum atau maksimum yang valid.

II. GUIDED

1

Source code:

```
package main
```

```
// Setyo Nugroho
```

```
// 103112400024
```

```
import "fmt"
```

```
func main() {
```

```
    var n int
```

```
    fmt.Print("Masukkan jumlah tanaman: ")
```

```
    fmt.Scan(&n)
```

```
    var tinggiTanaman [500]float64
```

```
    for i := 0; i < n; i++ {
```

```
        fmt.Printf("Masukkan tinggi tanaman ke -%d (cm): ", i+1)
```

```
        fmt.Scan(&tinggiTanaman[i])
```

```
    }
```

```
    min, max := tinggiTanaman[0], tinggiTanaman[0]
```

```
    for i := 1; i < n; i++ {
```

```
        if tinggiTanaman[i] < min {
```

```
            min = tinggiTanaman[i]
```

```
        }
```

```

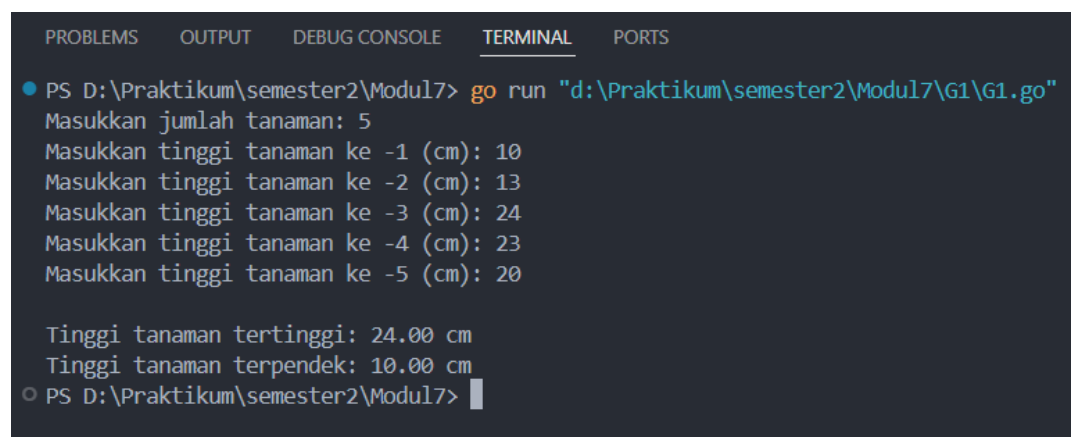
        if tinggiTanaman[i] > max {
            max = tinggiTanaman[i]
        }
    }

    fmt.Printf("\nTinggi tanaman tertinggi: %.2f cm\n", max)

    fmt.Printf("Tinggi tanaman terpendek: %.2f cm\n", min)
}

```

Output:



```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
● PS D:\Praktikum\semester2\Modul7> go run "d:\Praktikum\semester2\Modul7\G1\G1.go"
Masukkan jumlah tanaman: 5
Masukkan tinggi tanaman ke -1 (cm): 10
Masukkan tinggi tanaman ke -2 (cm): 13
Masukkan tinggi tanaman ke -3 (cm): 24
Masukkan tinggi tanaman ke -4 (cm): 23
Masukkan tinggi tanaman ke -5 (cm): 20

Tinggi tanaman tertinggi: 24.00 cm
Tinggi tanaman terpendek: 10.00 cm
○ PS D:\Praktikum\semester2\Modul7>

```

Deskripsi:

Program ini digunakan untuk mencari tinggi tanaman yang tertinggi dan terpendek dari data yang dimasukkan oleh pengguna. Pengguna diminta untuk memasukkan jumlah tanaman, lalu program akan mencatat tinggi setiap tanaman dan akhirnya menampilkan tinggi tanaman yang paling tinggi dan yang paling pendek.

2

Source code:

```
package main
```

```
// Setyo Nugroho
```

```
// 103112400024
```

```
import "fmt"
```

```
func main() {
```

```
    var x, y int
```

```
    fmt.Print("Masukkan jumlah buku dan jumlah buku per rak: ")
```

```
    fmt.Scan(&x, &y)
```

```
    var hargaBuku [500]float64
```

```
    fmt.Println("\nMasukkan harga setiap buku (dalam ribuan Rp):")
```

```
    for i := 0; i < x; i++ {
```

```
        fmt.Scan(&hargaBuku[i])
```

```
    }
```

```
    var hargaRatarata []float64
```

```
    for i := 0; i < x; i++ {
```

```
        total := 0.0
```

```
        for j := i; j < i+y && j < x; j++ {
```

```
            total += hargaBuku[j]
```

```
        }
```

```

        hargaRatarata = append(hargaRatarata, total/float64(y))
    }

    min, max := hargaBuku[0], hargaBuku[0]
    for _, harga := range hargaBuku[:x] {
        if harga < min {
            min = harga
        }
        if harga > max {
            max = harga
        }
    }

    fmt.Printf("\nRata-rata harga per rak: ")
    for _, avg := range hargaRatarata {
        fmt.Printf("%.2f ", avg)
    }

    fmt.Printf("\nHarga termahal: Rp %.2f\n", max)
    fmt.Printf("Harga termurah: Rp %.2f\n", min)

}

```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\Praktikum\semester2\Modul7> go run "d:\Praktikum\semester2\Modul7\G2\G2.go"
● Masukkan jumlah buku dan jumlah buku per rak: 4 2

Masukkan harga setiap buku (dalam ribuan Rp):
100.000
90.000
55.000
95.000

Rata-rata harga per rak: 95.00 72.50 75.00 47.50
Harga termahal: Rp 100.00
Harga termurah: Rp 55.00
○ PS D:\Praktikum\semester2\Modul7> |
```

Deskripsi:

Program ini menghitung rata rata harga buku per rak dan mencari harga buku termahal dan termurah. Pengguna memasukkan jumlah buku dan harga masing masing buku, lalu program menghitung rata rata harga tiap rak dan menampilkan harga termahal serta termurah.

3

Source code:

```
package main

// Setyo Nugroho
// 103112400024

import "fmt"
```

```
func main() {  
    var n int  
  
    fmt.Print("Masukan jumlah siswa: ")  
  
    fmt.Scan(&n)  
  
  
    var nilaiSiswa [200]float64  
  
    var totalNilai float64 = 0  
  
  
    fmt.Println("\nMasukan nilai ujian masing-masing siswa:")  
  
    for i := 0; i < n; i++ {  
        fmt.Scan(&nilaiSiswa[i])  
  
        totalNilai += nilaiSiswa[i]  
    }  
  
  
    rataRata := totalNilai / float64(n)  
  
  
    min, max := nilaiSiswa[0], nilaiSiswa[0]  
  
    var diAtasRataRata int = 0  
  
    for _, nilai := range nilaiSiswa[:n] {  
        if nilai < min {  
            min = nilai  
        }  
  
        if nilai > max {
```



```
        max = nilai
    }
    if nilai > rataRata {
        diAtasRataRata++
    }
}
```

```
fmt.Printf("\nNilai terendah: %.0f\n", min)
fmt.Printf("Nilai terendah: %.0f\n", max)
fmt.Printf("Rata-rata kelas: %.2f\n", rataRata)
fmt.Printf("Jumlah siswa di atas rata-rata: %d\n",
diAtasRataRata)
}
```

Output:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS D:\Praktikum\semester2\Modul7> go run "d:\Prakt
Masukan jumlah siswa: 6

Masukan nilai ujian masing-masing siswa:
90
93
87
88
95
70

Nilai terendah: 70
Nilai tertinggi: 95
Rata-rata kelas: 87.17
Jumlah siswa di atas rata-rata: 4
PS D:\Praktikum\semester2\Modul7> █
```

Deskripsi:

Program ini digunakan untuk menghitung nilai terendah, tertinggi, rata-rata nilai, dan jumlah siswa yang memiliki nilai di atas rata rata.

1. Pengguna memasukkan jumlah siswa dan nilai ujian masing masing.
2. Program menghitung total nilai dan rata rata nilai.
3. Program mencari nilai terendah dan tertinggi.
4. Program menghitung jumlah siswa yang nilainya lebih tinggi dari rata rata.
5. Hasil output berupa nilai terendah, tertinggi, rata rata nilai kelas, dan jumlah siswa di atas rata rata.

III. UNGUIDED

1

Source code:

```
package main
```

```
// Setyo Nugroho
```

```
// 103112400024
```

```
import "fmt"
```

```
func main() {
```

```
    var jumlah int
```

```
    var dataBerat [1000]float64
```

```
    fmt.Print("Input jumlah anak kelinci: ")
```

```
    fmt.Scan(&jumlah)
```

```
    for i := 0; i < jumlah; i++ {
```

```
        fmt.Printf("Berat kelinci ke-%d (kg): ", i+1)
```

```
        fmt.Scan(&dataBerat[i])
```

```
    }
```

```
    terberat := dataBerat[0]
```

```
    teringan := dataBerat[0]
```

```
    for i := 1; i < jumlah; i++ {
```

```

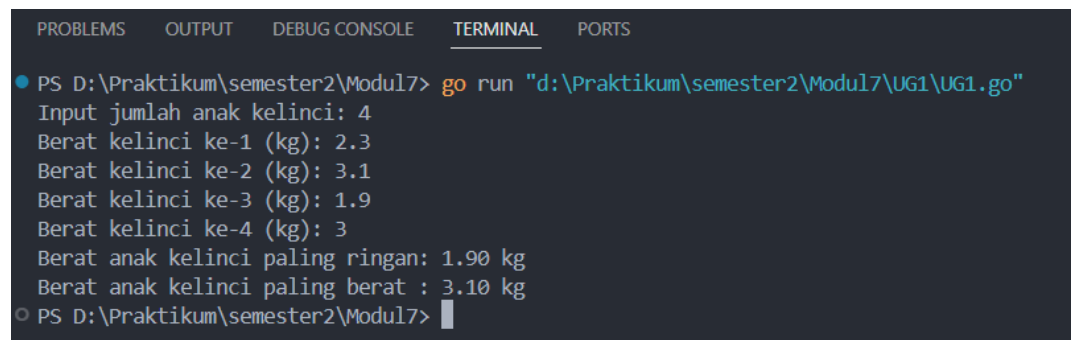
        if dataBerat[i] > terberat {
            terberat = dataBerat[i]
        }

        if dataBerat[i] < teringan {
            teringan = dataBerat[i]
        }
    }

    fmt.Printf("Berat anak kelinci paling ringan: %.2f kg\n", teringan)
    fmt.Printf("Berat anak kelinci paling berat : %.2f kg\n", terberat)
}

```

Output:



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS D:\Praktikum\semester2\Modul7> go run "d:\Praktikum\semester2\Modul7\UG1\UG1.go"
Input jumlah anak kelinci: 4
Berat kelinci ke-1 (kg): 2.3
Berat kelinci ke-2 (kg): 3.1
Berat kelinci ke-3 (kg): 1.9
Berat kelinci ke-4 (kg): 3
Berat anak kelinci paling ringan: 1.90 kg
Berat anak kelinci paling berat : 3.10 kg
○ PS D:\Praktikum\semester2\Modul7>

```

Deskripsi:

Program ini digunakan untuk mencari berat anak kelinci yang paling ringan dan paling berat dari sekumpulan data.

1. Pengguna diminta untuk memasukkan jumlah anak kelinci yang akan ditimbang.
2. Program kemudian meminta input berat setiap anak kelinci (dalam kilogram) dan menyimpannya dalam array dataBerat.

3. Program melakukan iterasi untuk mencari berat terberat dan teringan di antara anak kelinci.
4. Setelah seluruh data diperiksa, program menampilkan berat anak kelinci yang paling ringan dan paling berat.

Secara umum, fungsi dari program ini adalah untuk mencari nilai ekstrim (terberat dan teringan) dalam data yang dimasukkan oleh pengguna

2

Source code:

```
package main
```

```
// Setyo Nugroho
```

```
// 103112400024
```

```
import "fmt"
```

```
func main() {
```

```
    var x, y int
```

```
    var ikan [1000]float64
```

```
    fmt.Print("Masukkan jumlah ikan dan jumlah ikan per wadah: ")
```

```
    fmt.Scan(&x, &y)
```

```
    fmt.Println("Masukkan berat tiap ikan: ")
```

```
    for i := 0; i < x; i++ {
```

```
        fmt.Scan(&ikan[i])
```

```
}
```

```
jumlahWadah := (x + y - 1) / y
```

```
totalBeratWadah := make([]float64, jumlahWadah)
```

```
ikanDiWadah := make([]int, jumlahWadah)
```

```
for i := 0; i < x; i++ {
```

```
    wadah := i / y
```

```
    totalBeratWadah[wadah] += ikan[i]
```

```
    ikanDiWadah[wadah]++
```

```
}
```

```
fmt.Println("\nTotal Berat per Wadah")
```

```
for i := 0; i < jumlahWadah; i++ {
```

```
    fmt.Printf("Wadah Ke %d - Total Berat: %.2f kg\n", i+1,  
totalBeratWadah[i])
```

```
}
```

```
fmt.Println("\nRata rata Berat per Wadah")
```

```
for i := 0; i < jumlahWadah; i++ {
```

```
    rataRata := totalBeratWadah[i] / float64(ikanDiWadah[i])
```

```
    fmt.Printf("Wadah Ke %d Rata rata Berat: %.2f kg\n", i+1, rataRata)
```

```
}
```

```
}
```

Output:

```
PS D:\Praktikum\semester2\Modul7> go run "d:\Praktikum\semester2\Modul7\UG2\
Masukkan jumlah ikan dan jumlah ikan per wadah: 4 2
Masukkan berat tiap ikan:
2.1
2.4
2.8
3

Total Berat per Wadah
Wadah Ke 1 - Total Berat: 4.50 kg
Wadah Ke 2 - Total Berat: 5.80 kg

Rata rata Berat per Wadah
Wadah Ke 1 Rata rata Berat: 2.25 kg
Wadah Ke 2 Rata rata Berat: 2.90 kg
PS D:\Praktikum\semester2\Modul7>
```

Deskripsi:

Program ini dimulai dengan meminta input jumlah ikan dan jumlah ikan per wadah dari pengguna. Kemudian, pengguna diminta untuk memasukkan berat masing masing ikan. Berdasarkan input ini, program menghitung berapa banyak wadah yang dibutuhkan untuk menampung ikan tersebut, dengan cara membagi jumlah ikan dengan kapasitas per wadah dan membulatkan ke atas.

Setelah itu, program mengelompokkan ikan ke dalam wadah, menghitung total berat ikan di setiap wadah, dan menghitung rata rata berat ikan per wadah. Hasilnya kemudian ditampilkan, yaitu total berat ikan per wadah dan rata-rata berat ikan per wadah.

3

Source code:

```
package main
```

```
// Setyo Nugroho
```

```
// 103112400024
```

```
import "fmt"
```

```
type arrBalita [100]float64
```

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

```
    *bMin = arrBerat[0]
```

```
    *bMax = arrBerat[0]
```

```
    for i := 1; i < jumlah; i++ {
```

```
        if arrBerat[i] < *bMin {
```

```
            *bMin = arrBerat[i]
```

```
        }
```

```
        if arrBerat[i] > *bMax {
```

```
            *bMax = arrBerat[i]
```

```
        }
```

```
    }
```

```
}
```

```
func rerata(arrBerat arrBalita, jumlah int) float64 {
```

```
    var total float64 = 0.0
```

```
    for i := 0; i < jumlah; i++ {
```

```
        total += arrBerat[i]
```

```
    }
```



```

    return total / float64(jumlah)
}

func main() {
    var dataBerat arrBalita

    var n int

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

    fmt.Scan(&n)

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

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

        fmt.Scan(&dataBerat[i])
    }

    var beratMin, beratMax float64

    hitungMinMax(dataBerat, n, &beratMin, &beratMax)

    rerataBerat := rerata(dataBerat, n)

    fmt.Printf("\nBerat balita minimum: %.2f kg\n", beratMin)

    fmt.Printf("Berat balita maksimum: %.2f kg\n", beratMax)

```

```
    fmt.Printf("Rerata berat balita: %.2f kg\n", rerataBerat)\n}\n}
```

Output:

```
PS D:\Praktikum\semester2\Modul7> go run "d:\Praktikum\semester2\Modul7\UG3\UG3.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 D:\Praktikum\semester2\Modul7>
```

Deskripsi:

Program ini menerima input jumlah balita dan berat masing masing balita, kemudian menghitung beberapa informasi terkait berat balita, seperti:

1. **Berat Minimum:** Program mencari berat balita teringan.
2. **Berat Maksimum:** Program mencari berat balita terberat.
3. **Rata-rata Berat:** Program menghitung rata-rata berat balita.

Fungsi hitungMinMax digunakan untuk mencari berat minimum dan maksimum dari data berat balita, sedangkan fungsi rerata digunakan untuk menghitung rata-rata berat balita.

Setelah menerima input berat balita, hasil perhitungan berat minimum, maksimum, dan rata rata ditampilkan ke layar. Program juga melakukan pengecekan agar jumlah balita tidak melebihi batas yang ditentukan (antara 1 sampai 100).

Kesimpulan:

Program-program tersebut bertujuan untuk mengolah data berbentuk array yang berisi nilai atau berat, dan kemudian melakukan beberapa

operasi dasar seperti pencarian nilai minimum, maksimum, serta perhitungan rata-rata. Secara garis besar, berikut adalah kesimpulannya:

1. **Pengolahan Data Input:** Setiap program meminta input dari pengguna mengenai jumlah data dan nilai nilai terkait (seperti berat, harga, atau nilai) untuk diproses.
2. **Pencarian Nilai Minimum dan Maksimum:** Program menghitung nilai terkecil dan terbesar dalam data yang dimasukkan. Ini dilakukan dengan membandingkan setiap elemen dalam array.
3. **Perhitungan Rata-Rata:** Program menghitung nilai rata rata dengan menjumlahkan semua elemen dan membaginya dengan jumlah data yang ada.
4. **Penggunaan Fungsi:** Beberapa program memanfaatkan fungsi untuk mengorganisir dan memisahkan tugas tertentu, seperti menghitung rata-rata, nilai minimum, dan maksimum, untuk menjaga kode tetap terstruktur dan mudah dipahami.
5. **Validasi Input:** Pada beberapa program, terdapat validasi untuk memastikan jumlah data yang dimasukkan tidak melebihi kapasitas yang telah ditentukan, menjaga program tetap stabil.

Dengan demikian, tujuan utama dari kode kode tersebut adalah untuk mengolah dan menampilkan informasi statistik dasar (min, max, rata rata) dari data input yang diberikan oleh pengguna. Program ini dapat diterapkan pada berbagai konteks, seperti menghitung nilai ujian siswa, berat kelinci, atau harga buku.

REFERENSI