

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



**DISUSUN OLEH:**  
**SHEILA STEPHANIE ANINDYA**  
**103112400086**  
**S1 IF-12-01**  
**S1 TEKNIK INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO**  
**2025**

## **I. Dasar Teori**

Pencarian adalah aktivitas yang sering dilakukan dalam kehidupan sehari-hari, seperti mencari file dalam komputer, buku di perpustakaan, atau data dalam database. Salah satu bentuk pencarian yang umum adalah pencarian nilai ekstrim, yaitu mencari nilai terbesar atau terkecil dalam suatu kumpulan data. Algoritma pencarian nilai ekstrim bekerja dengan cara mengevaluasi setiap elemen dalam data dan memperbarui nilai ekstrim berdasarkan perbandingan yang dilakukan. Proses ini berlanjut hingga seluruh data diproses, dan nilai ekstrim yang tersisa adalah nilai maksimum atau minimum.

Pencarian nilai ekstrim banyak digunakan dalam ilmu komputer, khususnya dalam pemrograman dan pengolahan data. Algoritma ini dapat diterapkan untuk berbagai tujuan, seperti memilih elemen terbaik dalam data atau mengidentifikasi outlier. Pencarian nilai ekstrim juga berperan penting dalam analisis data dan pengambilan keputusan dalam bidang kecerdasan buatan.

Secara keseluruhan, meskipun sederhana, pencarian nilai ekstrim sangat efisien dan memiliki aplikasi yang luas dalam pengolahan data dan pengambilan keputusan.

## II. GUIDED

### 1. Source Code:

```
package main

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 := 0; i < n; i++ {
        if tinggiTanaman[i] < min {
            min = tinggiTanaman[i]
        }
        if tinggiTanaman[i] > max {
            max = tinggiTanaman[i]
        }
    }

    fmt.Printf("\n tinggi tanaman tertinggi : %.2f cm\n", max)
    fmt.Printf("\n tinggi tanaman terpendek : %.2f cm\n", min)
}
```

Output:

```
masukkan jumlah tanaman : 3
masukkan tinggi tanaman ke-1 (cm) : 13
masukkan tinggi tanaman ke-2 (cm) : 24
masukkan tinggi tanaman ke-3 (cm) : 15

tinggi tanaman tertinggi : 24.00 cm

tinggi tanaman terpendek : 13.00 cm
PS C:\Sheila Stephanie Anindya\SMT 2\Alpro 2\code>
```

Penjelasan :

Program ini meminta input tinggi tanaman, kemudian mencari dan menampilkan tinggi tanaman tertinggi dan terpendek.

2. Source Code:

```
package main

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 Rupiah) : ")
    for i := 0; i < x; i++ {
        fmt.Scan(&hargaBuku[i])
    }

    var hargaRataRata []float64
    for i := 0; i < x; i += y {
        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("\n harga termahal : %.2f Rp\n", max)
fmt.Printf("\n harga termurah : %.2f Rp\n", min)
}

```

Output:

```

masukkan jumlah buku dan jumlah buku per rak : 6 2

masukkan harga setiap buku (dalam Rupiah) :
40
30
50
50
10
15

rata-rata harga per rak : 35.00 50.00 12.50
harga termahal : 50.00 Rp

harga termurah : 10.00 Rp

```

Penjelasan :

Program ini meminta input jumlah buku dan jumlah buku per rak, kemudian menghitung rata-rata harga buku per rak dan mencari harga termahal serta termurah dari sekumpulan buku.

### 3. Source Code:

```
package main

import "fmt"

func main() {
    var n int
    fmt.Print("masukkan jumlah siswa : ")
    fmt.Scan(&n)

    var nilaiSiswa [200]float64
    var totalNilai float64 = 0

    fmt.Println("\nmasukkan 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 tertinggi : %.0f\n", max)
fmt.Printf("rata-rata kelas : %.2f\n", rataRata)
fmt.Printf("jumlah siswa diatas rata-rata : %d\n", diatasRataRata)
}

```

Output:

```

masukkan jumlah siswa : 5

masukkan nilai ujian masing-masing siswa :
90
80
86
75
40

nilai terendah : 40
nilai tertinggi : 90
rata-rata kelas : 74.20
jumlah sisea diatas rata-rata : 4

```

Penjelasan :

Program ini meminta input nilai ujian dari sejumlah siswa, kemudian menghitung nilai terendah, tertinggi, rata-rata kelas, dan jumlah siswa yang memiliki nilai di atas rata-rata.

### III. UNGUIDED

#### 1. Source Code:

```
package main

import "fmt"

func main() {
    var n int

    fmt.Print("jumlah anak kelinci : ")
    fmt.Scan(&n)

    if n <= 0 {
        fmt.Println("jumlah anak kelinci tidak valid.")
        return
    }

    var beratKelinci [1000]float64
    fmt.Println("berat anak kelinci :")
    for i := 0; i < n; i++ {
        fmt.Printf("berat ke-%d: ", i+1)
        fmt.Scan(&beratKelinci[i])
    }

    beratTerkecil := beratKelinci[0]
    beratTerbesar := beratKelinci[0]

    for i := 1; i < n; i++ {
        if beratKelinci[i] < beratTerkecil {
            beratTerkecil = beratKelinci[i]
        }
        if beratKelinci[i] > beratTerbesar {
            beratTerbesar = beratKelinci[i]
        }
    }
}
```



```

    }
}

fmt.Printf("berat terkecil: %.2f kg\n", beratTerkecil)
fmt.Printf("berat terbesar: %.2f kg\n", beratTerbesar)
}

```

Output:

```

jumlah anak kelinci : 3
berat anak kelinci :
berat ke-1: 20
berat ke-2: 32
berat ke-3: 15
berat terkecil: 15.00 kg
berat terbesar: 32.00 kg
PS C:\Sheila Stephanie Anindya\SMT 2\

```

Penjelasan :

Program ini meminta input jumlah anak kelinci dan berat masing-masing kelinci, kemudian menghitung dan menampilkan berat terkecil dan terbesar dari anak-anak kelinci tersebut.

2. Source Code:

```

package main

import "fmt"

func main() {
    var x, y int

    fmt.Print("jumlah ikan dan jumlah ikan per wadah : ")
    fmt.Scan(&x, &y)

    if x <= 0 || y <= 0 {
        fmt.Println("tidak valid.")
    }
}

```

```

    return
}

var beratIkan [1000]float64

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

var beratPerWadah []float64
var totalBerat float64
for i := 0; i < x; i++ {
    totalBerat += beratIkan[i]

    if (i+1)%y == 0 || i == x-1 {
        beratPerWadah = append(beratPerWadah, totalBerat)
        totalBerat = 0
    }
}

fmt.Println("berat ikan di setiap wadah :")
for i, berat := range beratPerWadah {
    fmt.Printf("Wadah %d: %.2f kg\n", i+1, berat)
}

rataRata := beratPerWadah[0] / float64(y)
fmt.Printf("\nrata-rata berat ikan per wadah: %.2f kg\n", rataRata)
}

```

Output:

```
jumlah ikan dan jumlah ikan per wadah : 6 3
berat ikan :
berat ikan ke-1: 1.4
berat ikan ke-2: 2.4
berat ikan ke-3: 2.4
berat ikan ke-4: 1.5
berat ikan ke-5: 1.6
berat ikan ke-6: 1.0
berat ikan di setiap wadah :
Wadah 1: 6.20 kg
Wadah 2: 4.10 kg

rata-rata berat ikan per wadah: 2.07 kg
PS C:\Sheila Stephanie Anindya\SMT 2\Alpro 2\code>
```

Penjelasan :

Program ini meminta input jumlah ikan, jumlah ikan per wadah, dan berat masing-masing ikan, kemudian menghitung total berat ikan di setiap wadah.

3. Source Code:

```
package main
import "fmt"

func main() {
    var n int
    fmt.Print("Masukkan jumlah balita: ")
    fmt.Scan(&n)

    var berat []float64
    var total, min, max float64

    for i := 0; i < n; i++ {
        var b float64
        fmt.Printf("Masukkan berat balita ke-%d (kg): ", i+1)
        fmt.Scan(&b)
        berat = append(berat, b)
```

```

        if i == 0 {
            min = b
            max = b
        } else {
            if b < min {
                min = b
            }
            if b > max {
                max = b
            }
        }
        total += b
    }
    rata := total / float64(n)

    fmt.Printf("\nBerat balita terkecil: %.2f kg\n", min)
    fmt.Printf("Berat balita terbesar: %.2f kg\n", max)
    fmt.Printf("Rata-rata berat balita: %.2f kg\n", rata)
}

```

Output:

```

Masukkan jumlah balita: 4
Masukkan berat balita ke-1 (kg): 5.3
Masukkan berat balita ke-2 (kg): 6.2
Masukkan berat balita ke-3 (kg): 4.1
Masukkan berat balita ke-4 (kg): 9.9

Berat balita terkecil: 4.10 kg
Berat balita terbesar: 9.90 kg
Rata-rata berat balita: 6.38 kg
PS C:\Sheila Stephanie Anindya\SMT 2\Alpro 2\code>

```

Penjelasan :

Program ini meminta input jumlah balita dan berat masing-masing balita, kemudian menghitung berat terkecil, terbesar, dan rata-rata berat balita.

#### **IV. RINGKASAN**

Pencarian nilai ekstrim adalah proses mencari nilai terbesar atau terkecil dalam suatu kumpulan data, seperti file di komputer atau data dalam database. Algoritma ini bekerja dengan mengevaluasi setiap elemen dan memperbarui nilai ekstrim berdasarkan perbandingan. Proses berlanjut hingga seluruh data diproses, menghasilkan nilai maksimum atau minimum. Algoritma ini penting dalam pemrograman, analisis data, dan pengambilan keputusan, serta sering digunakan untuk memilih elemen terbaik atau mengidentifikasi outlier dalam berbagai aplikasi.

## **V. DAFTAR PUSTAKA**

1. Telkom University. (n.d.). *Modul 10 Pencarian Nilai Ekstrim Pada Himpunan Data Praktikum Algoritma dan Pemrograman*
2. Fakultas Informatika, Telkom University.