

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL 11

PENCARIAN NILAI EKSTRIM PADA HIMPUNAN DATA



Disusun Oleh :

Natasya Intan Sukma Jiwanti / 2311102279

S1-IF-11-05

Dosen Pengampu :

Arif Amrulloh, S.Kom., M.Kom.

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

DASAR TEORI

Nilai ekstrim dalam himpunan data dapat merujuk pada nilai maksimum dan nilai minimum pada suatu kumpulan data. Nilai maksimum merupakan nilai terbesar atau tertinggi dalam kumpulan data, sedangkan nilai minimum merupakan nilai terendah. Nilai ekstrim ini dicari dengan tujuan agar dapat menganalisis data dan menghasilkan informasi terkait rentang dan distribusi data.

Proses pencarian ini akan dilakukan dengan memproses data secara sekuensial, dimana nilai minimum atau nilai maksimum pada data akan disimpan pada suatu variable untuk kemudian dibandingkan dengan nilai – nilai lain yang tersimpan pada kumpulan data. Algoritma perbandingan yang dilakukan dapat diuraikan sebagai berikut:

1. Nilai Maksimum

```
func findMax(data []int) int {  
    max := math.MinInt64  
    for _, value := range data {  
        if value > max {  
            max = value // Update nilai maksimum  
        }  
    }  
    return max  
}
```

2. Nilai Minimum

```
func findMin(data []int) int {  
    min := math.MaxInt64  
    for _, value := range data {  
        if value < min {  
            min = value // Update nilai minimum  
        }  
    }  
    return min  
}
```

UNGUIDED

1. Unguided 1

Soal Study Case

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 real berikutnya adalah berat dari anak kelinci yang akan dijual.

Keluaran:

Terdiri dari dua buah bilangan real yang menyatakan berat kelinci terkecil dan terbesar.

Sourcecode

```
package main
import "fmt"

type kelinci [1000]float64

func isiarray(a *kelinci, n int, data float64){
    for i := 0; i<n; i++){
        fmt.Print("Masukkan data berat kelinci ke-", i + 1,
" :")
        fmt.Scan(&data)
        a[i] = data
    }
}

func nilai_terkecil(a kelinci, n int) float64{
    var terkecil float64
    terkecil = a[0]
    i:=0
    for i<n{
        if terkecil > a[i]{
            terkecil = a[i]
        }
        i = i+1
    }
}
```

```

    }
    return terkecil
}

func nilai_terbesar(a kelinci, n int) float64{
    var terbesar float64
    terbesar = a[0]
    i :=1
    for i<n{
        if terbesar < a[i]{
            terbesar = a[i]
        }
        i = i+1
    }
    return terbesar
}

func main(){
    var berat kelinci
    var n int
    var data float64
    fmt.Print("Masukkan Banyak Kelinci: ")
    fmt.Scan(&n)
    isiarray(&berat, n, data)
    fmt.Print("Berat kelinci terkecil pada penimbangan kali
    ini adalah ", nilai_terkecil(berat, n), " sedangkan kelinci
    terberat pada penimbangan adalah ", nilai_terbesar(berat,
    n))
}

```

Screenshoot Output

```

PS D:\Praktikum Alpro\modul 8> go run "d:\Praktikum Alpro\modul 8\unguided1.go"
Masukkan Banyak Kelinci: 7
Masukkan data berat kelinci ke-1 :12.3
Masukkan data berat kelinci ke-2 :2.3
Masukkan data berat kelinci ke-3 :10
Masukkan data berat kelinci ke-4 :4.5
Masukkan data berat kelinci ke-5 :7.9
Masukkan data berat kelinci ke-6 :2.79
Masukkan data berat kelinci ke-7 :1.3
Berat kelinci terkecil pada penimbangan kali ini adalah 1.3 sedangkan kelinci terberat pada penimbangan adalah 12.3
PS D:\Praktikum Alpro\modul 8> █

```

Deskripsi Program

Program diatas merupakan program yang digunakan untuk mencari nilai ekstrim pada kumpulan data berat kelinci. Data berat yang menggunakan array, memiliki max range sebanyak 1000, akan diisi menggunakan perulangan dengan batas perulangan adalah n, dimana n akan diinputkan

oleh user. Untuk mencari kelinci terkecil dan terberat kita akan menggunakan variabel tambahan terbesar dan terkecil yang pada awalnya kita setting nilainya sama dengan nilai index ke-0. Saat mencari kelinci terkecil kita akan membandingkan nilai terkecil dengan data yang ada pada indeks ke-i array kelinci, apabila nilai indeks ke-i lebih kecil dibanding nilai yang tersimpan pada variabel terkecil maka nilai variabel terkecil akan berubah menjadi nilai indeks ke-i. Begitu pula saat mencari nilai terbesar. Kita akan membandingkan nilai terberat dengan data yang ada pada indeks ke-i array kelinci, apabila nilai indeks ke-i lebih besar dibanding nilai yang tersimpan pada variabel terberat maka nilai variabel terberat akan berubah menjadi nilai indeks ke-i.

2. Unguided 2

Soal Studi Case

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 real yang menyatakan banyaknya ikan yang akan dijual.

Keluaran:

Terdiri dari dua baris. Baris pertama adalah kumpulan bilangan real 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 real yang menyatakan berat rata-rata ikan di setiap wadah.

Sourcecode

```
package main
import "fmt"

type ikan [1000]float64

func dataikan(b *ikan, n int) {
    for i := 0; i < n; i++ {
        fmt.Print("Masukkan data berat ikan ke-", i+1, " :")
        fmt.Scan(&b[i])
    }
}

func banyakwadah(n int, y int) (int, int) {
    var banyakwadah, wadahterakhir, totalwadah int
    banyakwadah = n / y
    wadahterakhir = n % y
    if wadahterakhir == 0 {
        totalwadah = banyakwadah
    } else {
        totalwadah = banyakwadah + 1
    }
    return totalwadah, wadahterakhir
}

func rataRataPerWadah(b *ikan, n int, y int) (float64, int) {
    {
        totalWadah, _ := banyakwadah(n, y)
        var totalRata, rataseluruhnya, totalBerat float64
        var totalIkan, hitung int

        for wadah := 0; wadah < totalWadah; wadah++ {
            indexmulai := wadah * y
            indexselesai := indexmulai + y
            if indexselesai > n {
                indexselesai = n
            }

            for i := indexmulai; i < indexselesai; i++ {
                totalBerat += b[i]
                hitung++
            }
        }
    }
}
```

```

        if hitung > 0 {
            rata := float64(totalBerat) / float64(hitung)
            fmt.Printf("Rata-rata berat ikan di wadah %d: %.2f\n", wadah + 1, rata)
            totalRata += rata * float64(hitung)
            totalIkan += hitung
        } else {
            fmt.Printf("Wadah %d tidak memiliki ikan.\n", wadah)
        }
    }

    if totalIkan > 0 {
        rataseluruhnya = totalRata / float64(totalIkan)
    }
    return rataseluruhnya, totalIkan
}

func main() {
    var berat ikan
    var n, y int

    fmt.Print("Masukkan jumlah ikan: ")
    fmt.Scan(&n)
    fmt.Print("Masukkan kapasitas maksimum per wadah: ")
    fmt.Scan(&y)
    dataikan(&berat, n)
    totalWadah, wadahTerakhir := banyakwadah(n, y)

    fmt.Println("Total wadah yang dibutuhkan:", totalWadah)
    fmt.Println("Jumlah ikan di wadah terakhir:", wadahTerakhir)
    rataseluruhnya, totalIkan := rataRataPerWadah(&berat, n, y)

    if totalIkan > 0 {
        fmt.Printf("Rata-rata keseluruhan berat ikan: %.2f\n", rataseluruhnya)
    } else {
        fmt.Println("Tidak ada ikan untuk dihitung rata-ratanya.")
    }
}

```

Screenshoot Output

```
PS D:\Praktikum Alpro\modul 8> go run "d:\Praktikum Alpro\modul 8\unguided2.go"
Masukkan jumlah ikan: 9
Masukkan kapasitas maksimum per wadah: 5
Masukkan data berat ikan ke-1 : 1.2
Masukkan data berat ikan ke-2 : 2.3
Masukkan data berat ikan ke-3 : 3.4
Masukkan data berat ikan ke-4 :
4.5
Masukkan data berat ikan ke-5 : 5.6
Masukkan data berat ikan ke-6 : 6.5
Masukkan data berat ikan ke-7 : 5.4
Masukkan data berat ikan ke-8 : 4.3
Masukkan data berat ikan ke-9 : 3.2
Total wadah yang dibutuhkan: 2
Jumlah ikan di wadah terakhir: 4
Rata-rata berat ikan di wadah 1: 3.40
Rata-rata berat ikan di wadah 2: 4.04
Rata-rata keseluruhan berat ikan: 3.81
PS D:\Praktikum Alpro\modul 8> █
```

Deskripsi Program

Program diatas merupakan program yang menggunakan array ikan dengan maksimal 1000. Program diatas akan meminta user untuk menginputkan berat ikan sebanyak x, yang nantinya akan dibagi ke wadah – wadah yang ada dimana, setiap wadah akan menampung ikan dengan maksimal berjumlah y. Untuk sisa ikan yang tidak dapat dibagi y akan masuk ke wadah terakhir. Setelah itu untuk mengetahui rata – rata berat ikan ditiap wadah kita perlu menghitung total berat ikan di tiap wadah dibagi jumlah ikan. Jumlah berat akan dihitung menggunakan for dari index awalnya adalah wadah * y dan index akhir dari perulangan dilihat dari index awal + y. Pada program diatas juga ditambahkan fitur menghitung rata rata seluruhnya dengan menjumlahkan rata – rata di berbagai wadah / jumlah wadah.

3. Unguided 3

Studi Case

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.

Sourcecode

```
package main
import "fmt"

type ikan [1000]float64

func dataikan(b *ikan, n int) {
    for i := 0; i < n; i++ {
        fmt.Print("Masukkan data berat ikan ke-", i+1, " : ")
        fmt.Scan(&b[i])
    }
}

func banyakwadah(n int, y int) (int, int) {
    if y <= 0 {
        return 0, n
    }
    banyakwadah := n / y
    wadahterakhir := n % y
    totalwadah := banyakwadah
    if wadahterakhir > 0 {
        totalwadah++
    }
    return totalwadah, wadahterakhir
}

func totalBeratDanJumlahIkanPerWadah(b *ikan, n int, y int)
([]float64, []int) {
    totalWadah, _ := banyakwadah(n, y)
    beratPerWadah := make([]float64, totalWadah)
    jumlahIkanPerWadah := make([]int, totalWadah)

    for wadah := 0; wadah < totalWadah; wadah++ {
        indexmulai := wadah * y
```

```

        indexselesai := indexmulai + y
        if indexselesai > n {
            indexselesai = n
        }

        var totalBerat float64
        for i := indexmulai; i < indexselesai; i++ {
            totalBerat += b[i]
            jumlahIkanPerWadah[wadah]++
        }
        beratPerWadah[wadah] = totalBerat
    }

    return beratPerWadah, jumlahIkanPerWadah
}

func rataRataPerWadah(beratPerWadah []float64,
    jumlahIkanPerWadah []int) (float64, int) {
    var totalRata, rataseluruhnya float64
    var totalIkan int

    for wadah, totalBerat := range beratPerWadah {
        hitung := jumlahIkanPerWadah[wadah]
        if hitung > 0 {
            fmt.Printf("Total berat ikan di wadah %d:
%.2f\n", wadah+1, totalBerat)
            rata := totalBerat / float64(hitung)
            fmt.Printf("Rata-rata berat ikan di wadah %d:
%.2f\n", wadah+1, rata)
            totalRata += totalBerat
            totalIkan += hitung
        } else {
            fmt.Printf("Wadah %d tidak memiliki ikan.\n",
wadah+1)
        }
    }

    if totalIkan > 0 {
        rataseluruhnya = totalRata / float64(totalIkan)
    }
    return rataseluruhnya, totalIkan
}

func main() {
    var berat ikan

```

```

var n, y int

fmt.Print("Masukkan jumlah ikan: ")
fmt.Scan(&n)
fmt.Print("Masukkan kapasitas maksimum per wadah: ")
fmt.Scan(&y)

if n <= 0 || y <= 0 {
    fmt.Println("Jumlah ikan dan kapasitas maksimum per
wadah harus lebih besar dari 0.")
    return
}

dataikan(&berat, n)
beratPerWadah, jumlahIkanPerWadah :=
totalBeratDanJumlahIkanPerWadah(&berat, n, y)
rataseluruhnya, totalIkan :=
rataRataPerWadah(beratPerWadah, jumlahIkanPerWadah)

if totalIkan > 0 {
    fmt.Printf("Rata-rata keseluruhan berat ikan:
%.2f\n", rataseluruhnya)
} else {
    fmt.Println("Tidak ada ikan untuk dihitung rata-
ratanya.")
}
}

```

Screenshoot Output

```

PS D:\Praktikum Alpro\modul 8> go run "d:\Praktikum Alpro\modul 8\unguided2.go"
Masukkan jumlah ikan: 7
Masukkan kapasitas maksimum per wadah: 4
Masukkan data berat ikan ke-1 : 1.2
Masukkan data berat ikan ke-2 : 2.3
Masukkan data berat ikan ke-3 : 3.4
Masukkan data berat ikan ke-4 : 4.5
Masukkan data berat ikan ke-5 : 4.3
Masukkan data berat ikan ke-6 : 3.2
Masukkan data berat ikan ke-7 : 2.1
Total berat ikan di wadah 1: 11.40
Rata-rata berat ikan di wadah 1: 2.85
Total berat ikan di wadah 2: 9.60
Rata-rata berat ikan di wadah 2: 3.20
Rata-rata keseluruhan berat ikan: 3.00
PS D:\Praktikum Alpro\modul 8> 

```

Deskripsi Program

Program diatas merupakan program pencarian nilai ekstrim yang akan mencari nilai minimum dan maximum, serta mencari rata rata dari data balita yang disimpan menggunakan array berkapasitas 100 anak. Untuk mencari nilai minimum dan maximum kita akan menyimpan nilai variabel min dan max yang nilainya adalah nilai array balita index ke 0. Kemudian, saat mencari kelinci terkecil kita akan membandingkan nilai terkecil dengan data yang ada pada indeks ke-i array kelinci, apabila nilai indeks ke-i lebih kecil dibanding nilai yang tersimpan pada variabel terkecil maka nilai variabel terkecil akan berubah menjadi nilai indeks ke-i. Begitu pula saat mencari nilai terbesar. Kita akan membandingkan nilai terberat dengan data yang ada pada indeks ke-i array kelinci, apabila nilai indeks ke-i lebih besar dibanding nilai yang tersimpan pada variabel terberat maka nilai variabel terberat akan berubah menjadi nilai indeks ke-i. Selain itu untuk mencari rata rata kita akan menjumlahkan semua berat balita yang ada pada array lalu membaginya dengan n.