

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
MODUL: 10
MATERI: PENCARIAN NILAI EKSTRIM PADA HIMPUNAN DATA



DISUSUN OLEH:

NAMA: JESIKA METANIA RAHMA ARIFIN

NIM: 103112400080

KELAS: 12 IF 01

DOSEN:

Dimas Fanny Hebrisianto Permadi S.ST, M.Kom

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025/2026

DASAR TEORI

1. Ide Pencarian Nilai Max/Min

Pencarian adalah suatu proses yang lazim dilakukan di dalam kehidupan sehari-hari. Contoh penggunaannya dalam kehidupan nyata sangat beragam, misalnya pencarian file di dalam directory komputer, pencarian suatu teks di dalam sebuah dokumen, pencarian buku pada rak buku, dan contoh lainnya. Pertama pada modul ini akan dipelajari salah satu algoritma pencarian nilai terkecil atau terbesar pada sekumpulan data, atau biasa disebut pencarian nilai ekstrim. Ide algoritma sederhana sekali. Karena data harus diproses secara sekuensial, maka nilai atau indeks ke nilai maksimum dari data yang telah diproses disimpan untuk dibandingkan dengan data berikutnya. Nilai yang berhasil disimpan sampai algoritma tersebut berakhir adalah nilai maksimum yang dicari.

Adapun algoritmanya secara umum adalah sebagai berikut:

- 1) Jadikan data pertama sebagai nilai ekstrim
- 2) Lakukan validasi nilai ekstrim dari data kedua hingga data terakhir.
 - Apabila nilai ekstrim tidak valid, maka update nilai ekstrims tersebut dengan data yang dicek.
- 3) Apabila semua data telah dicek, maka nilai ekstrim yang dimiliki adalah valid.

Berikut ini adalah notasi dalam pseudocode dan bahasa Go, misalnya untuk pencarian nilai terbesar atau maksimum:

	Notasi Algoritma	Notasi dalam bahasa Go
1	max \leftarrow 1	max = 0
2	i \leftarrow 2	i = 1
3	while i \leq n do	for i < n {
4	if a[i] > a[max] then	if a[i] > a[max] {
5	max \leftarrow i	max = i
6	endif	}
7	i \leftarrow i + 1	i = i + 1
8	endwhile	}

2. Pencarian Nilai Ekstrim pada Array Bertipe Data Dasar

```
type arrInt [2023]int
...

func terkecil_1(tabInt arrInt, n int) int {
    /* mengembalikan nilai terkecil yang terdapat di dalam tabInt yang berisi n
    bilangan bulat */
    var min int = tabInt[0] // min berisi data pertama
    var j int = 1           // pencarian dimulai dari data berikutnya
    for j < n {
        if min > tabInt[j] { // pengecekan apakah nilai minimum valid
            min = tabInt[j] // update nilai minimum dengan yang valid
        }
        j = j + 1
    }
    return min // returnkan nilai minimumnya
}
```

Potongan program di atas sedikit berbeda dengan sebelumnya karena penggunaan indeks array pada bahasa Go di mulai dari nol atau "0" seperti penjelasan pada modul 9. Selanjutnya, pada penjelasan di awal bab 3 telah disampaikan bahwa pada pencarian yang terpenting adalah posisi atau indeks dari nilai yang dicari dalam kumpulan data atau array. Oleh karena itu modifikasi pada program di atas dapat dilihat pada potongan program berikut ini!

```

...
type arrInt [2023]int
...

func terkecil_2(tabInt arrInt, n int) int {
/* mengembalikan indeks nilai terkecil yang terdapat di dalam tabInt yang berisi
n bilangan bulat */
    var idx int = 0           // idx berisi indeks data pertama
    var j int = 1             // pencarian dimulai dari data berikutnya
    for j < n {
        if tabInt[idx] > tabInt[j] { // pengecekan apakah nilai minimum valid
            idx = j                 // update nilai minimum dengan yang valid
        }
        j = j + 1
    }
    return idx                 // returnkan indeks nilai minimumnya
}

```

3. Pencarian Nilai Ekstrim pada Array Bertipe Data Terstruktur

Pada kasus yang lebih kompleks pencarian ekstrim dapat juga dilakukan, misalnya mencari data mahasiswa dengan nilai terbesar, mencari lagu dengan durasi terlama, mencari pembalap yang memiliki catatan waktu balap tercepat, dan sebagainya. Sebagai contoh misalnya terdapat array yang digunakan untuk menyimpan data mahasiswa, kemudian terdapat fungsi IPK yang digunakan untuk mencari data mahasiswa dengan IPK tertinggi.

```

...
type mahasiswa struct {
    nama, nim, kelas, jurusan string
    ipk float64
}
type arrMhs [2023]mahasiswa
...

func IPK_2(T arrMhs, n int) int {
/* mengembalikan indeks mahasiswa yang memiliki ipk tertinggi pada array T yang
berisi n mahasiswa */
    var idx int = 0
    var j int = 1
    for j < n {
        if T[idx].ipk < T[j].ipk {
            idx = j
        }
        j = j + 1
    }
    return idx
}

```

GUIDED 1:

```

package main

import (
    "fmt"
)

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

```

```

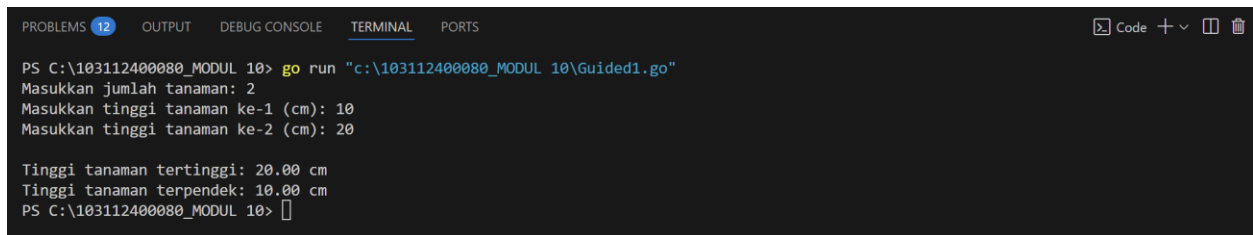
var tinggiTanaman [500]float64
for i := 0; i < n; i++ {
    fmt.Printf("Masukkan tinggi tanaman ke-%d (cm): ", i+1)
    fmt.Scan(&tinggiTanaman[i])
}

min, max := tinggiTanaman[0], tinggiTanaman[0]
for i := 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 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400080_MODUL 10> go run "c:\103112400080_MODUL 10\Guided1.go"
Masukkan jumlah tanaman: 2
Masukkan tinggi tanaman ke-1 (cm): 10
Masukkan tinggi tanaman ke-2 (cm): 20

Tinggi tanaman tertinggi: 20.00 cm
Tinggi tanaman terpendek: 10.00 cm
PS C:\103112400080_MODUL 10>

```

DESKRIPSI PROGRAM:

Program ini adalah program yang di buat untuk menghitung tinggi maximum dan tinggi minimum dari suatu tanaman.

GUIDED 2:

```

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 ribuan Rp):")
    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 {
        if harga < min {
            min = harga
        }
        if harga > max {
            max = harga
        }
    }
    fmt.Printf("\nRata-rata harga buku per rak:")
    for _, avg := range hargaRataRata {
        fmt.Printf("%.2f", avg)
    }
    fmt.Printf("\nHarga termahal: %.2f Rp\n", max)
    fmt.Printf("\nHarga termurah: %.2f Rp\n", min)
}

```

OUTPUT:

```

PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400080_MODUL 10> go run "c:\103112400080_MODUL 10\Guided2.go"
Masukkan jumlah buku dan jumlah buku per rak: 2 3

Masukkan harga setiap buku (dalam ribuan Rp):
10000
5000

Rata-rata harga buku per rak:5000.00
Harga termahal: 10000.00 Rp

Harga termurah: 0.00 Rp
PS C:\103112400080_MODUL 10>

```

DESKRIPSI PROGRAM:

Program ini adalah program yang di buat untuk menghitung nilai maximum dan minimum dari suatu harga masing- masing buku.

GUIDED 3:

```

package main

import "fmt"

func main() {

```

```

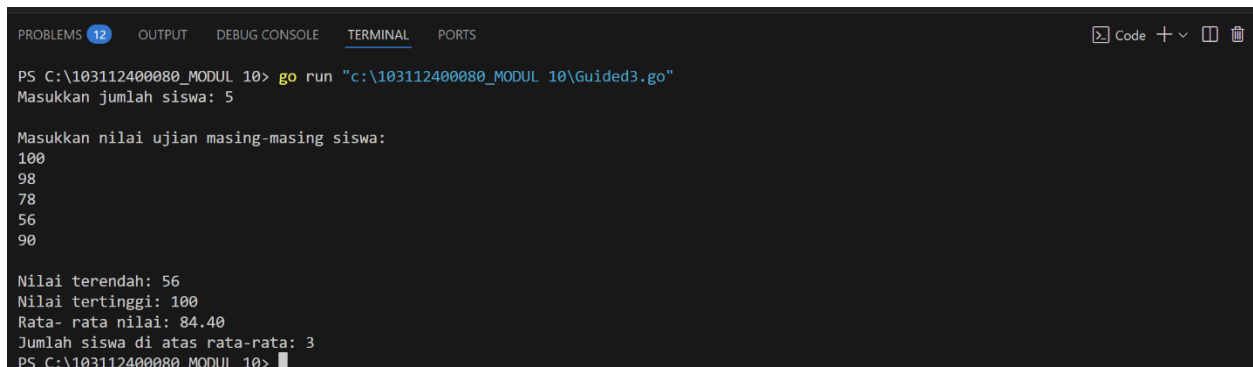
var n int
fmt.Print("Masukkan jumlah siswa: ")
fmt.Scan(&n)

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 nilai: %.2f\n", rataRata)
fmt.Printf("Jumlah siswa di atas rata-rata: %d\n", diAtasRataRata)
}

```

OUTPUT:



```

PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400080_MODUL 10> go run "c:\103112400080_MODUL 10\Guided3.go"
Masukkan jumlah siswa: 5

Masukkan nilai ujian masing-masing siswa:
100
98
78
56
90

Nilai terendah: 56
Nilai tertinggi: 100
Rata- rata nilai: 84.40
Jumlah siswa di atas rata-rata: 3
PS C:\103112400080_MODUL 10>

```

DESKRIPSI PROGRAM:

Program ini adalah program yang dibuat untuk menghitung nilai maximum dan minimum serta nilai rata-rata dari suatu mahasiswa.

UNGUIDED 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.

```
// JESIKA METANIA RAHMA ARIFIN
// 10311240080

package main

import "fmt"

func main() {
    var anak_Kelinci int
    fmt.Print("Masukkan jumlah anak kelinci: ")
    fmt.Scan(&anak_Kelinci)

    if anak_Kelinci > 1000 {
        fmt.Println("Jumlah anak kelinci melebihi kapasitas (maksimal 1000).")
        return
    }
    var berat [1000]float64
    for i := 0; i < anak_Kelinci; i++ {
        fmt.Printf("Berat kelinci ke-%d: ", i+1)
        fmt.Scan(&berat[i])
    }

    min := berat[0]
    max := berat[0]

    for i := 1; i < anak_Kelinci; i++ {
        if berat[i] < min {
            min = berat[i]
        }
        if berat[i] > max {
            max = berat[i]
        }
    }

    fmt.Printf("Berat terkecil: %.2f\n", min)
    fmt.Printf("Berat terbesar: %.2f\n", max)
}
```

OUTPUT:

```
PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400080_MODUL 10> go run "c:\103112400080_MODUL 10\Unguided1.go"
Masukkan jumlah anak kelinci: 3
Berat kelinci ke-1: 2.5
Berat kelinci ke-2: 3.4
Berat kelinci ke-3: 3.7
Berat terkecil: 2.50
Berat terbesar: 3.70
PS C:\103112400080_MODUL 10> |
```

DESKRIPSI PROGRAM:

Program ini adalah program yang di buat untuk menentukn berat maximum dan berat minimum dari suatu anak kelinci.

UNGUIDED 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.

```
// JESIKA METANIA RAHMA ARIFIN
// 103112400080

package main

import "fmt"

func main() {
    var x, y int
    fmt.Scan(&x, &y)

    berat := make([]float64, x)
    for i := 0; i < x; i++ {
        fmt.Scan(&berat[i])
    }

    jumlahWadah := (x + y - 1) / y
    total := make([]float64, jumlahWadah)

    for i := 0; i < x; i++ {
        total[i/y] += berat[i]
    }
    for _, t := range total {
```

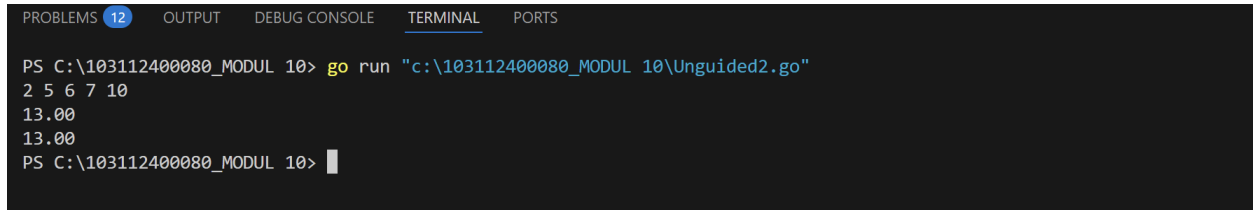


```

        fmt.Printf("%.2f ", t)
    }
    fmt.Println()
    var sum float64
    for _, t := range total {
        sum += t
    }
    fmt.Printf("%.2f\n", sum/float64(jumlahWadah))
}

```

OUTPUT:



```

PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400080_MODULE 10> go run "c:\103112400080_MODULE 10\Unguided2.go"
2 5 6 7 10
13.00
13.00
PS C:\103112400080_MODULE 10>

```

DESKRIPSI PROGRAM:

Program ini adalah program yang di buat untuk menampung berat ikan yang akan dijual.

UNGUIDED 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:

```

// JESIKA METANIA RAHMA ARIFIN
// 103112400080

package main

import (
    "fmt"
)

// Tipe data array khusus
type arrBalita [100]float64

// Fungsi untuk menghitung berat minimum dan maksimum
func hitungMinMax(arr arrBalita, n int, bMin, bMax *float64) {
    *bMin = arr[0]
    *bMax = arr[0]
    for i := 1; i < n; i++ {
        if arr[i] < *bMin {
            *bMin = arr[i]
        }
        if arr[i] > *bMax {

```

```

        *bMax = arr[i]
    }
}

// Fungsi untuk menghitung rata-rata
func rerata(arr arrBalita, n int) float64 {
    var total float64
    for i := 0; i < n; i++ {
        total += arr[i]
    }
    return total / float64(n)
}

func main() {
    var n int
    var data arrBalita

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

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

    var min, max float64
    hitungMinMax(data, n, &min, &max)
    rata := rerata(data, n)

    // Output hasil
    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", rata)
}

```

OUTPUT:

```

PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400080_MODUL 10> go run "c:\103112400080_MODUL 10\Unguided3.go"
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: 9.9
Berat balita minimum: 4.10 kg
Berat balita maksimum: 9.90 kg
Rerata berat balita: 6.38 kg
PS C:\103112400080_MODUL 10>

```

DESKRIPSI PROGRAM:

Program ini adalah program yang dibuat untuk menghitung berat maximum dan minimum bayi serta menghitung berat rata-rata bayi pada suatu data sample.

KESIMPULAN

Pencarian nilai ekstrem pada himpunan data merupakan proses penting dalam analisis data yang bertujuan untuk menemukan nilai maksimum dan minimum dari sekumpulan data. Nilai ekstrem ini memberikan informasi yang berguna dalam memahami rentang data, mendeteksi outlier, serta mendukung pengambilan keputusan yang lebih akurat dalam berbagai bidang seperti statistik, sains, ekonomi, dan teknologi. Dengan menggunakan metode yang tepat, baik secara manual maupun melalui bantuan perangkat lunak, proses pencarian nilai ekstrem dapat dilakukan secara efisien dan efektif.