

**LAPORAN PRAKTIKUM
ALGORITMA PEMROGRAMAN 2**

MODUL X

PENCARIAN NILAI EKSTRIM PADA HIMPUNAN DATA



Oleh:

Fadhel Yussie Ramadhan

2311102322

S1 TEKNIK INFORMATIKA

INSTITUT TEKNOLOGI TELKOM PURWOKERTO

2024

I. DASAR TEORI

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	<code>max ← 1</code>	<code>max = 0</code>
2	<code>i ← 2</code>	<code>i = 1</code>
3	<code>while i <= n do</code>	<code>for i < n {</code>
4	<code> if a[i] > a[max] then</code>	<code> if a[i] > a[max] {</code>
5	<code> max ← i</code>	<code> max = i</code>
6	<code> endif</code>	<code> }</code>
7	<code> i ← i + 1</code>	<code> i = i + 1</code>
8	<code>endwhile</code>	<code>}</code>

II. GUIDED

1. Source Code

```
package main

import (
    "fmt"
)

func main() {
    var N int
    var berat [1000]float64

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

    fmt.Println("Masukkan berat anak kelinci: ")
    for i := 0; i < N; i++ {
        fmt.Scan(&berat[i]) // Perbaikan: tambahkan '&'
    }

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

    for i := 1; i < N; i++ {
        if berat[i] < min {
            min = berat[i]
        }
        if berat[i] > max { // Perbaikan: hapus '()' setelah
'max'
            max = berat[i]
        }
    }

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

Output

```
OUTPUT PROBLEMS 1 DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\USER\Documents\SEMESTER 3\ALPRO 2\PRAKTIKUM\modul10> go run "c:\Users\USER\Documents\SEMESTER 3\ALPRO 2\PRAKTIKUM\modul10\G1\g1.go"
Masukkan jumlah anak kelinci: 2
Masukkan berat anak kelinci:
5
6
Berat terkecil: 5.00
Berat terbesar: 6.00
PS C:\Users\USER\Documents\SEMESTER 3\ALPRO 2\PRAKTIKUM\modul10> |
```

Keterangan

Program ini meminta input jumlah anak kelinci dan berat masing-masing, lalu menentukan berat terkecil dan terbesar. Berat dimasukkan dalam array, dan perbandingan dilakukan melalui loop. Hasil akhir ditampilkan dengan format dua desimal.

2. Source Code

```
package main
import (
    "fmt"
)

func main() {
    var x, y int
    fmt.Print("masukkan jumlah ikan dan kapasitas wadah: ")
    fmt.Scan(&x, &y)

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

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

```

for i := 0; i < x; i++ {
    indekswadah := i / y
    totalBeratWadah[indekswadah] += berat[i]
}

fmt.Println("total berat tiap wadah: ")
for _, total := range totalBeratWadah {
    fmt.Printf("%.2f", total)
}
fmt.Println()

fmt.Println("rata rata berat tiap wadah: ")
for _, total := range totalBeratWadah {
    ratarata := total / float64(y)
    fmt.Printf("%.2f", ratarata)
}
fmt.Println()
}

```

Output



```

PS C:\Users\USER\Documents\SEMESTER 3\ALPRO 2\PRAKTIKUM\modul10> go run "c:\Users\USER\Documents\SEMESTER 3\ALPR
0 2\PRAKTIKUM\modul10\G2\g2.go"
masukkan jumlah ikan dan kapasitas wadah: 2
2
masukkan berat tiap ikan:
3
2
total berat tiap wadah:
5.00
rata rata berat tiap wadah:
2.50
PS C:\Users\USER\Documents\SEMESTER 3\ALPRO 2\PRAKTIKUM\modul10>

```

Keterangan

Program ini menghitung jumlah wadah berdasarkan kapasitas dan jumlah ikan, lalu mendistribusikan ikan ke wadah secara berurutan. Berat total dan rata-rata tiap wadah dihitung serta ditampilkan dalam format dua desimal. Algoritma ini cocok untuk simulasi pengelompokan.

III. UNGUIDED

1. Source Code

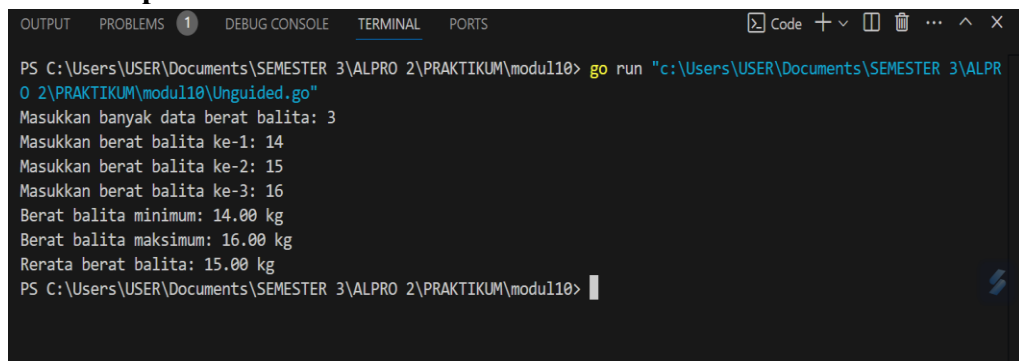
```
2. package main
3. import "fmt"
4.
5. type arrBalita [100]float64
6.
7. func hitungMinMax(arrBerat arrBalita, n int, bMin, bMax
   *float64){
8.     *bMin = arrBerat[0]
9.     *bMax = arrBerat[0]
10.    for i := 1; i < n; i++){
11.        if arrBerat[i] < *bMin {
12.            *bMin = arrBerat[i]
13.        }
14.        if arrBerat[i] > *bMax {
15.            *bMax = arrBerat[i]
16.        }
17.    }
18.}
19.
20. func hitungRerata(arrBerat arrBalita, n int) float64{
21.     total := 0.0
22.     for i := 0; i < n; i++){
23.         total += arrBerat[i]
24.     }
25.     return total / float64(n)
26.}
```

```

27.
28.func main(){
29.    var n int
30.    var berat arrBalita
31.    var min, max float64
32.
33.    fmt.Print("Masukkan banyak data berat balita: ")
34.    fmt.Scan(&n)
35.
36.    for i := 0; i < n; i++ {
37.        fmt.Printf("Masukkan berat balita ke-%d: ", i+1)
38.        fmt.Scan(&berat[i])
39.    }
40.
41.    hitungMinMax(berat, n, &min, &max)
42.    rerata := hitungRerata(berat, n)
43.
44.    fmt.Printf("Berat balita minimum: %.2f kg\n", min)
45.    fmt.Printf("Berat balita maksimum: %.2f kg\n", max)
46.    fmt.Printf("Rerata berat balita: %.2f kg\n", rerata)
47.}

```

Output:



```

PS C:\Users\USER\Documents\SEMESTER 3\ALPRO 2\PRAKTIKUM\modul10> go run "c:\Users\USER\Documents\SEMESTER 3\ALPRO 2\PRAKTIKUM\modul10\Unguided.go"
Masukkan banyak data berat balita: 3
Masukkan berat balita ke-1: 14
Masukkan berat balita ke-2: 15
Masukkan berat balita ke-3: 16
Berat balita minimum: 14.00 kg
Berat balita maksimum: 16.00 kg
Rerata berat balita: 15.00 kg
PS C:\Users\USER\Documents\SEMESTER 3\ALPRO 2\PRAKTIKUM\modul10>

```

Keterangan:

Program ini mengolah data berat balita untuk menghitung nilai minimum, maksimum, dan rata-rata. Prosesnya menggunakan fungsi terpisah: `hitungMinMax` untuk menentukan berat minimum dan maksimum, serta `hitungRerata` untuk menghitung rata-rata. Input berat balita dimasukkan melalui perulangan, kemudian hasilnya ditampilkan dalam format dua desimal.