

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL 10

PENCARIAN NILAI EXSTRIM PADA HIMPUNAN DATA



Disusun Oleh:

NAMA:IKRAM IRIANSYAH

NIM : 2311102184

KELAS : IF-11-02

S1 TEKNIK INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

A. Ide Pencarian Nilai MAX/MIN

Pencarian adalah suatu proses yang umum 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.

Ide algoritma sederhana, karena data harus diproses secara sekuensial, maka nilai atau indeks ke nilai maksimum dari data yang telah diproses disimpan sampai algoritma tersebut berakhir adalah nilai maksimum yang dicari. Algoritma secara umum :

- 1) Jadikan nilai pertama sebagai nilai ekstrim
- 2) Lakukan validasi nilai ekstrim dari data kedua hingga data terakhir.
 - Apabila nilai ekstrim tidak valid, maka update nilai ekstrim 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>

B. Pencarian Nilai Ekstrim pada Array Bertipe Data Dasar

Misalnya terdefinisi sebuah array of integer dengan kapasitas 2023, dan array terisi sejumlah N bilangan bulat, kemudian pencarian nilai terkecil dilakukan pada array tersebut. Perhatikan potongan program dalam bahasa Go berikut ini :

```
5  type arrInt [2023]int
..  ...
15
16 func terkecil_1(tabInt arrInt, n int) int {
17  /* mengembalikan nilai terkecil yang terdapat di dalam tabInt yang berisi n
18  bilangan bulat */
19      var min int = tabInt[0]          // min berisi data pertama
20      var j int = 1                    // pencarian dimulai dari data berikutnya
21      for j < n {
22          if min > tabInt[j] {          // pengecekan apakah nilai minimum valid
23              min = tabInt[j]          // update nilai minimum dengan yang valid
24          }
25          j = j + 1
26      }
27      return min                       // returnkan nilai minimumnya
28 }
```

Potongan program di atas sedikit berbeda dengan sebelumnya karena penggunaan indeks array pada bahasa Go di mulai dari nol atau “0” . Maka program di atas dapat dimodifikasi menjadi seperti program berikut ini!

```
..  ...
5  type arrInt [2023]int
..  ...
15
16 func terkecil_2(tabInt arrInt, n int) int {
17  /* mengembalikan indeks nilai terkecil yang terdapat di dalam tabInt yang berisi
18  n bilangan bulat */
19      var idx int = 0                  // idx berisi indeks data pertama
20      var j int = 1                    // pencarian dimulai dari data berikutnya
21      for j < n {
22          if tabInt[idx] > tabInt[j] { // pengecekan apakah nilai minimum valid
23              idx = j                  // update nilai minimum dengan yang valid
24          }
25          j = j + 1
26      }
27      return idx                       // returnkan indeks nilai minimumnya
28 }
```

C. Pencarian Nilai Ekstrim pada Array Bertipe Data Terstruktur

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

```
.. ...
5  type mahasiswa struct {
..     nama, nim, kelas, jurusan string
..     ipk float64
.. }
.. type arrMhs [2023]mahasiswa
.. ...
15
16 func IPK_1(T arrMhs, n int) float64 {
17     /* mengembalikan ipk terkecil yang dimiliki mahasiswa pada array T yang berisi
18     n mahasiswa */
19     var tertinggi float64 = T[0].ipk
20     var j int = 1
21     for j < n {
22         if tertinggi < T[j].ipk {
23             tertinggi = T[j].ipk
24         }
25         j = j + 1
26     }
27     return tertinggi
28 }
```

Apabila diperhatikan potongan program diatas, maka akan diperoleh nilai ipk tertinggi, tetapi kita tidak akan memperoleh identitas mahasiswa dengan ipk tertinggi tersebut. Maka program dapat dimodifikasi menjadi :

```
.. ...
5  type mahasiswa struct {
..     nama, nim, kelas, jurusan string
..     ipk float64
.. }
.. type arrMhs [2023]mahasiswa
.. ...
15
16 func IPK_2(T arrMhs, n int) int {
17     /* mengembalikan indeks mahasiswa yang memiliki ipk tertinggi pada array T yang
18     berisi n mahasiswa */
19     var idx int = 0
20     var j int = 1
21     for j < n {
22         if T[idx].ipk < T[j].ipk {
23             idx = j
24         }
25         j = j + 1
26     }
27     return idx
28 }
```

Sehingga melalui algoritma di atas, identitas mahasiswa dapat diperoleh, misalnya T[idx].nama, T[idx].nim, T[idx].kelas, hingga T[idx].jurusan.

I. 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])
    }

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

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

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

fmt.Printf("Berat kelinci terbesar: %.2f\n", max)

}
```

OUTPUT

```
PS C:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 10> go run "c:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 10\GUIDED\guided1.go"
Masukkan jumlah anak kelinci: 5
Masukkan berat anak kelinci:
2.4
2.2
3.1
2.8
26
Berat kelinci terkecil: 2.20
Berat kelinci terbesar: 26.00
PS C:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 10>
```

DESKRIPSI PROGRAM

Program diatas merupakan implementasi dari pencarian nilai ekstrim yang digunakan untuk mendata berat anak kelinci yang akan dijual. Program menggunakan array dengan kapasitas 1000 untuk menampung data. Program meminta user untuk menginputkan jumlah data yang akan disimpan beserta masing-masing berat data. Lalu program akan mencari nilai ekstrim dari data yaitu mencari serta menampilkan nilai minimal dan maksimal dari data yang diinputkan oleh user.

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)
    jumlahIkanDiWadah := make([]int, jumlahWadah)

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

```

    jumlahIkanDiWadah[indeksWadah]++
}

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

// Output rata-rata berat tiap wadah
fmt.Println("Rata-rata berat tiap wadah:")
for i := 0; i < jumlahWadah; i++ {
    if jumlahIkanDiWadah[i] > 0 {
        rataRata := totalBeratWadah[i] / float64(jumlahIkanDiWadah[i])
        fmt.Printf("%.2f ", rataRata)
    } else {
        fmt.Printf("0.00 ") // jika wadah kosong
    }
}
fmt.Println()
}

```

OUTPUT

```

PS C:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 10> go run "c:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 10\GUIDED\guided2.go"
Masukkan jumlah ikan dan kapasitas wadah: 5 8
Masukkan berat tiap ikan:
2.4
3.4
4.4
3.8
.
Total berat tiap wadah:
14.00
Rata-rata berat tiap wadah:
2.80
PS C:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 10>

```

DESKRIPSI PROGRAM

Menghitung total berat dan rata-rata berat ikan di setiap wadah berdasarkan jumlah ikan dan kapasitas wadah yang ditentukan oleh pengguna. Pengguna diminta memasukkan jumlah ikan, kapasitas maksimum wadah, dan berat masing-masing ikan. Program membagi ikan ke dalam wadah secara berurutan berdasarkan indeks, menghitung total berat ikan di setiap wadah, dan menampilkan hasilnya. Selain itu, program juga menghitung rata-rata berat tiap wadah berdasarkan kapasitas maksimum yang dimasukkan. Hasil total dan rata-rata berat ditampilkan dengan format dua angka di belakang koma.

II. UNGUIDED

1. SOURCE CODE

```
package main

import "fmt"

type arrBalita [100]float64

func minmax(arrBerat arrBalita, bMin, bMax *float64) {
    *bMin = arrBerat[0]
    *bMax = arrBerat[0]

    for i := 1; i < len(arrBerat); i++ {
        if arrBerat[i] != 0 {
            if arrBerat[i] < *bMin {
                *bMin = arrBerat[i]
            }
            if arrBerat[i] > *bMax {
                *bMax = arrBerat[i]
            }
        }
    }
}

func ratarata(arrBerat arrBalita, n int) float64 {
    var total float64
    for i := 0; i < n; i++ {
        total += arrBerat[i]
    }
    return total / float64(n)
}

func main() {
    var balita arrBalita
    var n int
    var min, max float64

    fmt.Print("Masukkan jumlah balita : ")
    fmt.Scan(&n)

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

    minmax(balita, &min, &max)

    rerata := ratarata(balita, n)
```



```
fmt.Printf("\n= Hasil Analisis Data Berat Balita =\n")
fmt.Printf("Berat Terkecil : %.2f kg\n", min)
fmt.Printf("Berat Terbesar : %.2f kg\n", max)
fmt.Printf("Berat Rata-rata : %.2f kg\n", rerata)
}
```

OUTPUT

```
PS C:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 10> go run "c:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 10\UNGUIDED\unguided1.go"
Masukkan jumlah 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

= Hasil Analisis Data Berat Balita =
Berat Terkecil : 4.10 kg
Berat Terbesar : 9.90 kg
Berat Rata-rata : 6.38 kg
PS C:\Users\Ikram\Documents\Folder Ikram\Document Kuliah\Semester 3\PRAKTIKUM ALPRO 2\MODUL 10>
```

DESKRIPSI PROGRAM

Pengguna diminta untuk memasukkan jumlah balita dan berat masing-masing balita. Program menggunakan array untuk menyimpan data berat, kemudian fungsi `hitungMinMax` digunakan untuk menemukan berat minimum dan maksimum, sementara fungsi `rata Rata` menghitung rata-rata berat balita. Hasil analisis ditampilkan dalam format angka desimal dengan satuan kilogram (kg), sehingga memudahkan pengguna untuk memahami distribusi berat balita yang dimasukkan.