

**LAPORAN PRAKTIKUM
ALGORITMA PEMROGRAMAN 2
MODUL 10
PENCARIAN NILAI EKSTRIM
PADA HIMPUNAN DATA**



Oleh:

MUHAMMAD AGHA ZULFADHLI

2311102015

S1-IF11-02

**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.

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

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

Misalnya terdefinisi sebuah array of integer dengan kapasitas 2023, dan array berisi sejumlah N bilangan bulat, kemudian pencarian nilai terkecil dilakukan pada array tersebut. Perhatikan potongan program

```
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  }
```

Sebagai contoh misalnya terdapat array yang digunakan untuk menyimpan data mahasiswa, kemudian terdapat fungsi IPK yang digunakan untuk mencari data mahasiswa dengan IPK tertinggi.

```

..  ...
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 }

```

II. GUIDED

1. Guided 1

Source code

```
package main

import "fmt"

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

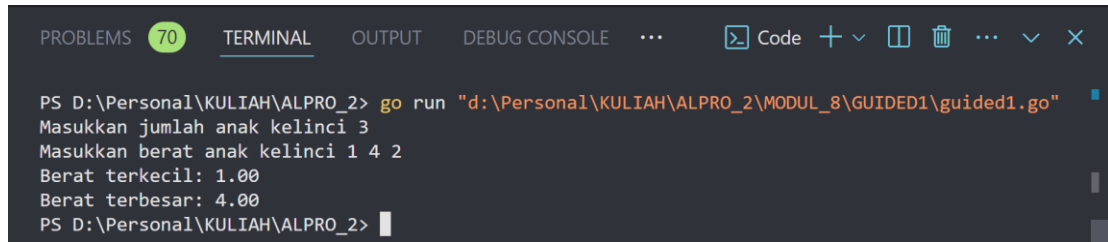
    fmt.Print("Masukkan jumlah anak kelinci ")
    fmt.Scan(&N)
    fmt.Print("Masukkan berat anak kelinci ")
    for i := 0; i < N; i++ {
        fmt.Scan(&berat[i])
    }

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

    for i := 0; i < N; 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)
}
```

Screenshoot program



```
PS D:\Personal\KULIAH\ALPRO_2> go run "d:\Personal\KULIAH\ALPRO_2\MODUL_8\GUIDED1\guided1.go"
Masukkan jumlah anak kelinci 3
Masukkan berat anak kelinci 1 4 2
Berat terkecil: 1.00
Berat terbesar: 4.00
PS D:\Personal\KULIAH\ALPRO_2>
```

Deskripsi program

Program ini menerima input jumlah dan berat masing-masing anak kelinci, kemudian menentukan berat minimum dan maksimum di antara mereka. Pertama, pengguna diminta memasukkan jumlah anak kelinci (N), lalu berat badan masing-masing anak kelinci disimpan dalam array berat. Nilai awal min dan max diatur sebagai berat pertama dalam array. Selanjutnya, program memeriksa setiap berat dalam array; jika berat lebih kecil dari min, min diperbarui, dan jika lebih besar dari max, max diperbarui. Akhirnya, program mencetak berat terkecil dan terbesar anak kelinci tersebut.

2. Guided 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])
    }

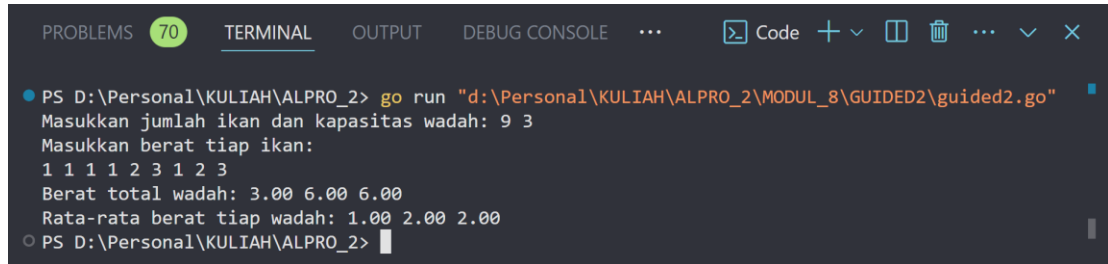
    jumlahWadah := (x + y - 1) / y
    totalBeratWadah := make([]float64, jumlahWadah)

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

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

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

Screenshoot program

A screenshot of a code editor's terminal window. The terminal shows the execution of a Go program. The user enters the command 'go run "d:\Personal\KULIAH\ALPRO_2\MODUL_8\GUIDED2\guided2.go"'. The program prompts for the number of fish and tank capacity, then for the weight of each fish. It then calculates and displays the total weight for each tank and the average weight per tank.

```
PS D:\Personal\KULIAH\ALPRO_2> go run "d:\Personal\KULIAH\ALPRO_2\MODUL_8\GUIDED2\guided2.go"
Masukkan jumlah ikan dan kapasitas wadah: 9 3
Masukkan berat tiap ikan:
1 1 1 1 2 3 1 2 3
Berat total wadah: 3.00 6.00 6.00
Rata-rata berat tiap wadah: 1.00 2.00 2.00
PS D:\Personal\KULIAH\ALPRO_2>
```

Deskripsi program

Program ini menghitung distribusi berat ikan ke dalam beberapa wadah dengan kapasitas tertentu, serta menampilkan berat total dan rata-rata berat ikan per wadah. Pertama, pengguna memasukkan jumlah ikan (x) dan kapasitas wadah (y). Setelah itu, berat masing-masing ikan dimasukkan ke dalam array berat. Program kemudian menghitung jumlah wadah yang diperlukan (jumlahWadah) dan membuat array totalBeratWadah untuk menyimpan total berat ikan di setiap wadah. Setiap ikan dialokasikan ke wadah berdasarkan indeksinya, dan beratnya ditambahkan ke total wadah tersebut. Program kemudian mencetak total berat setiap wadah, diikuti dengan rata-rata berat per wadah, dihitung dengan membagi total berat wadah dengan kapasitasnya.

III. UNGUIDED

1. Unguided 1

Source code

```
package main

import "fmt"

type arrBalita [100]float64

var jml_balita int

func hitungMinMax(arrBerat arrBalita, bMin, bMax *float64) {
    *bMin = arrBerat[0]
    *bMax = arrBerat[0]
    for i := 0; i < jml_balita; i++ {
        if arrBerat[i] < *bMin {
            *bMin = arrBerat[i]
        }
        if arrBerat[i] > *bMax {
            *bMax = arrBerat[i]
        }
    }
}

func Rerata(arrBerat arrBalita) float64 {
    var jumlah float64
    for _, value := range arrBerat {
        jumlah += value
    }
    return jumlah / float64(jml_balita)
}

func main() {
    var balitas arrBalita
    var min, max float64

    fmt.Print("Masukan Banyak data berat balita: ")
    fmt.Scan(&jml_balita)
    for i := 0; i < jml_balita; i++ {
```



```

        fmt.Print("Masukan berat balita ke-", i+1, ": ")
        fmt.Scan(&balitas[i])
    }
    hitungMinMax(balitas, &min, &max)
    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", Rerata(balitas))
}

```

Screenshoot program



```

PROBLEMS 70 TERMINAL OUTPUT DEBUG CONSOLE ... Code + - 
PS D:\Personal\KULIAH\ALPRO_2> go run "d:\Personal\KULIAH\ALPRO_2\MODUL_8\1.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 D:\Personal\KULIAH\ALPRO_2>

```

Deskripsi program

Program ini menerima data berat badan balita, lalu menghitung dan menampilkan berat minimum, maksimum, dan rata-rata dari data tersebut. Pengguna diminta memasukkan jumlah balita dan berat masing-masing balita. Fungsi `hitungMinMax` menentukan nilai berat minimum dan maksimum, sementara fungsi `Rerata` menghitung rata-rata. Program kemudian mencetak hasil minimum, maksimum, dan rata-rata berat badan balita tersebut.