

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL X
SEARCHING**



Oleh:

Muhammad Rifki Fadhillah

2311102032

IF 11 02

**S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2024/2025**

I. DASAR TEORI

Pencarian Nilai Ekstrim

Pencarian nilai ekstrim bertujuan untuk menemukan nilai maksimum dan minimum dalam sebuah dataset (array). Berikut adalah cara untuk mencari nilai tersebut:

Inisialisasi: Set nilai maksimum dan minimum pertama kali dengan elemen pertama dalam array.

Iterasi: Bandingkan setiap elemen dalam array dengan nilai maksimum dan minimum yang telah ditemukan.

Update: Jika elemen lebih besar dari nilai maksimum, perbarui nilai maksimum. Jika elemen lebih kecil dari nilai minimum, perbarui nilai minimum.

Contoh pseudecodenya

```
Mulai
  Masukkan jumlah data (n)
  Buat array arr[] dengan panjang n
  Untuk i = 1 hingga n
    Masukkan elemen arr[i]
  Akhir untuk
  Set max = arr[1] dan min = arr[1]
  Untuk setiap elemen value dalam arr
    Jika value lebih besar dari max
      Set max = value
    Jika value lebih kecil dari min
      Set min = value
  Akhir untuk

  Tampilkan max dan min
Selesai
```

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])
    }

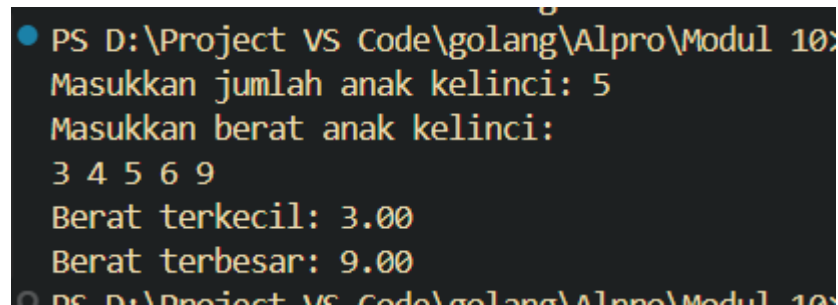
    //Inisialisasi nilai min dan max dengan elemen pertama

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

OUTPUT

A screenshot of a terminal window showing the execution of a Go program. The prompt is 'PS D:\Project VS Code\golang\Alpro\Modul 10>'. The user enters 'Masukkan jumlah anak kelinci: 5'. The prompt changes to 'Masukkan berat anak kelinci:'. The user enters '3 4 5 6 9'. The program outputs 'Berat terkecil: 3.00' and 'Berat terbesar: 9.00'.

```
PS D:\Project VS Code\golang\Alpro\Modul 10>
Masukkan jumlah anak kelinci: 5
Masukkan berat anak kelinci:
3 4 5 6 9
Berat terkecil: 3.00
Berat terbesar: 9.00
PS D:\Project VS Code\golang\Alpro\Modul 10>
```

DESKRIPSI PROGRAM

Program ini bertujuan untuk menentukan berat terkecil dan terbesar dari sejumlah anak kelinci berdasarkan data yang diinput oleh pengguna. Pada awalnya, program meminta pengguna untuk memasukkan jumlah anak kelinci yang akan dihitung beratnya. Setelah itu, program meminta pengguna untuk mengisi berat masing-masing anak kelinci.

Setelah semua data berat dimasukkan, program menginisialisasi nilai awal untuk berat terkecil dan terbesar menggunakan berat anak kelinci pertama. Selanjutnya, program memeriksa setiap berat yang dimasukkan dengan menggunakan perulangan. Jika berat tertentu lebih kecil dari nilai terkecil saat ini, nilai tersebut menggantikan nilai terkecil. Demikian pula, jika berat tertentu lebih besar dari nilai terbesar saat ini, nilai tersebut menggantikan nilai terbesar.

Setelah seluruh perbandingan selesai, program menampilkan hasil berupa berat terkecil dan berat terbesar, yang diformat hingga dua angka di belakang koma. Program ini dirancang untuk memberikan hasil yang akurat berdasarkan data berat yang diinputkan.

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 //pembulatan ke atas jika x
tidak habis dibagi y
    totalBeratWadah := make([]float64, jumlahWadah)

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

    //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 _, total := range totalBeratWadah {
        rataRata := total / float64(y)
        fmt.Printf("%.2f ", rataRata)
    }
    fmt.Println()
}
```

OUTPUT

```
● PS D:\Project VS Code\golang\Alpro\Modul 10> go run "d:\Pr
Masukkan jumlah ikan dan kapasitas wadah: 4 4
Masukkan berat tiap ikan:
5 3 2 4
Total berat tiap wadah:
14.00
Rata-rata berat tiap wadah:
3.50
```

DESKRIPSI PROGRAM

Program ini bertujuan untuk mengelompokkan ikan ke dalam wadah dengan kapasitas tertentu, lalu menghitung total berat dan rata-rata berat untuk setiap wadah. Pengguna diminta untuk memasukkan jumlah ikan, kapasitas wadah, serta berat masing-masing ikan.

Setelah data dimasukkan, program menghitung jumlah wadah yang diperlukan. Perhitungan ini dilakukan dengan rumus pembulatan ke atas menggunakan $(x + y - 1) / y$, di mana x adalah jumlah ikan dan y adalah kapasitas wadah. Selanjutnya, program menggunakan array untuk menyimpan total berat ikan di setiap wadah. Berat ikan ditambahkan ke wadah sesuai dengan indeks yang dihitung dari posisi ikan dalam daftar.

Setelah semua ikan dikelompokkan, program menampilkan total berat setiap wadah. Kemudian, program menghitung rata-rata berat tiap wadah dengan membagi total berat dengan kapasitas wadah, lalu menampilkan hasilnya. Jika jumlah ikan tidak habis dibagi kapasitas wadah, rata-rata untuk wadah terakhir akan didasarkan pada jumlah ikan aktual di wadah tersebut.

III. UNGUIDED

1. SOURCE CODE

```
package main

import(
    "fmt"
)

type arrBalita [100]float64

func hitungMinMax(arrBerat arrBalita, x int, bMin, bMax *float64){
    *bMin = arrBerat[0]
    *bMax = arrBerat[0]
    for i := 1; i < x; i++ {
        if arrBerat[i] > *bMax {
            *bMax = arrBerat[i]
        }
        if arrBerat[i] < *bMin {
            *bMin = arrBerat[i]
        }
    }
}

func rerata(arrBerat arrBalita, x int) float64{
    sum:= 0.0
    for i := 0; i < x; i++ {
        sum += arrBerat[i]
    }
    return sum/ (float64(x))
}
```

```
func main(){
    var x int
    var arrBalita[100] float64
    var bMin,bMax float64

    fmt.Print("Masukkan banyak data berat balita: ")
    fmt.Scan(&x)

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

    hitungMinMax(arrBalita,x,&bMin,&bMax)
    avg := rerata(arrBalita,x)
```

```
fmt.Printf("Berat balita maksimal: %.2f kg\n", bMax)
fmt.Printf("Berat balita minimum: %.2f kg\n", bMin)
fmt.Printf("Rerata berat balita: %.2f kg", avg )
}
```

OUTPUT

```
PS D:\Project VS Code\golang\Alpro\Modul 10> go run "d
Masukkan banyak data berat 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

Berat balita maksimal: 9.90 kg
Berat balita minimum: 4.10 kg
Rerata berat balita: 6.38 kg
```

DESKRIPSI PROGRAM

Program ini ditulis dalam bahasa Go dan bertujuan untuk menghitung berat badan balita. Program ini meminta input data dari pengguna tentang jumlah balita serta berat badan masing-masing balita, kemudian melakukan beberapa perhitungan statistik pada data yang dimasukkan.

Setelah program meminta pengguna untuk memasukkan jumlah balita, program kemudian menerima input berat badan untuk masing-masing balita. Data ini disimpan dalam array `arrBalita` yang dapat menampung hingga 100 elemen. Setelah data dimasukkan, program kemudian menghitung beberapa statistik: nilai maksimum, minimum, dan rata-rata dari berat badan balita.

Untuk menghitung nilai maksimum dan minimum, fungsi `hitungMinMax` digunakan. Fungsi ini memeriksa setiap elemen dalam array dan memperbarui nilai maksimum dan minimum sesuai dengan nilai yang ditemukan. Nilai minimum dan maksimum tersebut kemudian disimpan dalam variabel `bMin` dan

bMax, yang dipassing menggunakan pointer agar dapat dimodifikasi di dalam fungsi.

Program juga menghitung rata-rata berat badan balita menggunakan fungsi rerata. Fungsi ini menjumlahkan semua elemen dalam array dan kemudian membaginya dengan jumlah elemen untuk mendapatkan rata-rata.

Setelah semua perhitungan selesai, program menampilkan hasilnya, yaitu berat badan balita yang terbesar (maksimal), terkecil (minimal), dan rata-rata berat badan dari semua balita yang dimasukkan.