

**‘LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2  
MODUL 7  
PENCARIAN NILAI EKSTRIM PADA HIMPUNAN DATA**



Oleh:

SAVILA NUR FADILLA

103112400031

IF-12-01

**S1 TEKNIK INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2025**



### 7.3 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_1(T arrMhs, n int) float64 {
    /* mengembalikan ipk terkecil yang dimiliki mahasiswa pada array T yang berisi
    n mahasiswa */
    var tertinggi float64 = T[0].ipk
    var j int = 1
    for j < n {
        if tertinggi < T[j].ipk {
            tertinggi = T[j].ipk
        }
        j = j + 1
    }
    return tertinggi
}
```

### 7.4 Array

Array adalah kumpulan data bertipe sama, yang disimpan dalam sebuah variabel. Array memiliki kapasitas yang nilainya ditentukan saat pembuatan, menjadikan elemen/data yang disimpan di array tersebut jumlahnya tidak boleh melebihi yang sudah dialokasikan. Default nilai tiap elemen array pada awalnya tergantung dari tipe datanya. Jika int maka tiap element zero value-nya adalah 0, jika bool maka false, dan seterusnya. Setiap elemen array memiliki indeks berupa angka yang merepresentasikan posisi urutan elemen tersebut. Indeks array dimulai dari 0.

### 7.5 Struct

Struct adalah kumpulan definisi variabel (atau property) dan atau fungsi (atau method), yang dibungkus sebagai tipe data baru dengan nama tertentu. Property dalam struct, tipe datanya bisa bervariasi. Mirip seperti map, hanya saja key-nya sudah didefinisikan di awal, dan tipe data tiap itemnya bisa berbeda.

## II. GUIDED

### 1.) Source Code

```
package main

import "fmt"

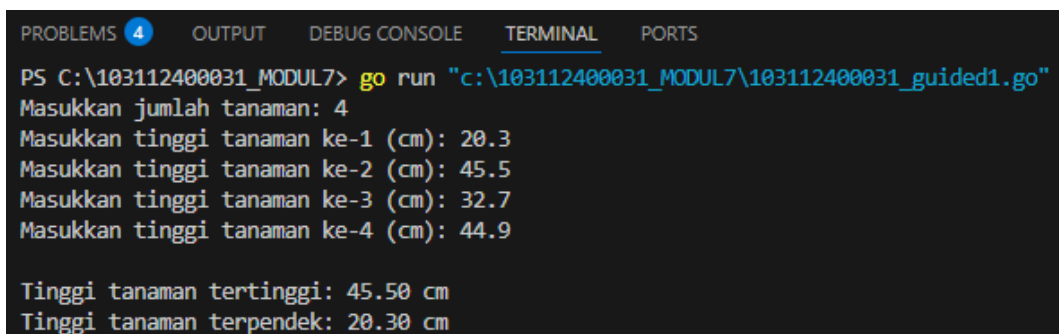
func main() {
    var n int
    fmt.Print("Masukkan jumlah tanaman: ")
    fmt.Scan(&n)

    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 :



The screenshot shows a terminal window with the following content:

```
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400031_MODUL7> go run "c:\103112400031_MODUL7\103112400031_guided1.go"
Masukkan jumlah tanaman: 4
Masukkan tinggi tanaman ke-1 (cm): 20.3
Masukkan tinggi tanaman ke-2 (cm): 45.5
Masukkan tinggi tanaman ke-3 (cm): 32.7
Masukkan tinggi tanaman ke-4 (cm): 44.9

Tinggi tanaman tertinggi: 45.50 cm
Tinggi tanaman terpendek: 20.30 cm
```

Penjelasan : Program ini bertujuan untuk menampilkan tinggi tanaman tertinggi dan terpendek. Program meminta kita menginput jumlah tanaman, lalu program meminta kita menginput tinggi masing masing tanaman (dalam cm). Data tinggi tanaman itu disimpan dalam array dengan kapasitas 500, max digunakan untuk mencari tinggi tanaman tertinggi, min digunakan untuk mencari tinggi tanaman terpendek. Nilai awal min dan max diambil dari data pertama, lalu dibandingkan dengan data lain menggunakan perulangan for. Jika ada yang lebih kecil dari min, maka nilai min diperbarui. Jika ada yang lebih besar dari max, maka nilai max diperbarui. Setelah itu program akan menampilkan hasilnya. Output berupa dua baris di mana baris pertama merupakan tinggi tanaman tertinggi dalam cm dan baris kedua merupakan tinggi tanaman terpendek dalam cm.

## 2.) Source Code

```
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.Print("\nmasukkan harga setiap buku (dalam Rupiah) : ")
    for i := 0; i < x; i++ {
        fmt.Scan(&hargaBuku[i])
    }

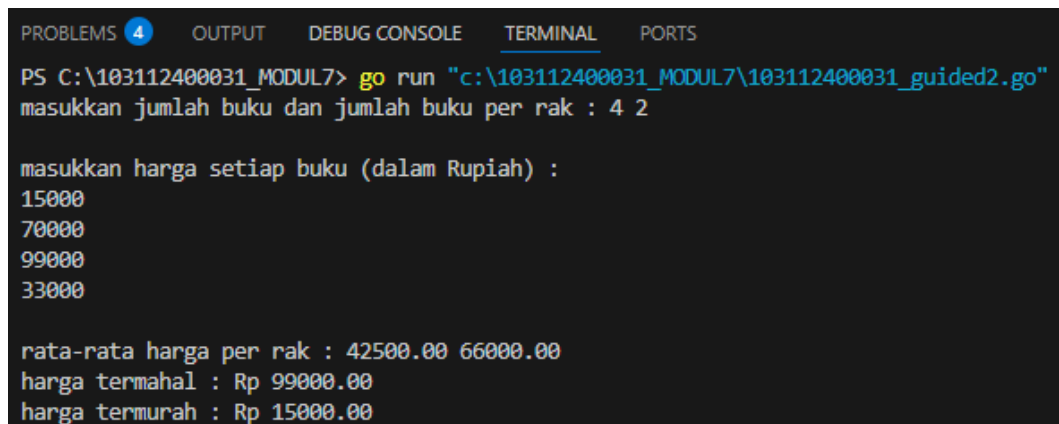
    var hargaRataRata []float64
    for i := 0; i < x; i += y {
        total := 0.0
        for j := i; j < i+y && j < x; j++ {
            total += hargaBuku[j]
        }
        hargaRataRata = append(hargaRataRata, total/float64(y))
    }

    min, max := hargaBuku[0], hargaBuku[0]
    for _, harga := range hargaBuku[:x] {
        if harga < min {
            min = harga
        }
        if harga > max {
            max = harga
        }
    }

    fmt.Printf("\nrata-rata harga per rak : ")
    for _, avg := range hargaRataRata {
        fmt.Printf("%.2f ", avg)
    }

    fmt.Printf("\nharga termahal : Rp %.2f\n", max)
    fmt.Printf("harga termurah : Rp %.2f\n", min)
}
```

Output :



```
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400031_MODUL7> go run "c:\103112400031_MODUL7\103112400031_guided2.go"
masukkan jumlah buku dan jumlah buku per rak : 4 2

masukkan harga setiap buku (dalam Rupiah) :
15000
70000
99000
33000

rata-rata harga per rak : 42500.00 66000.00
harga termahal : Rp 99000.00
harga termurah : Rp 15000.00
```

Penjelasan : Program ini bertujuan untuk menghitung rata-rata harga buku per rak dan menentukan harga buku yang termahal dan termurah. Program meminta kita menginput jumlah buku dan jumlah buku per rak, lalu program meminta kita menginput harga setiap buku (dalam rupiah). Data harga buku itu disimpan dalam array dengan kapasitas 500, max digunakan untuk mencari harga buku termahal, min digunakan untuk mencari harga buku termurah. Nilai awal min dan max diambil dari data pertama, lalu dibandingkan dengan data lain menggunakan perulangan for. Jika ada yang lebih kecil dari min, maka nilai min diperbarui. Jika ada yang lebih besar dari max, maka nilai max diperbarui. Program juga akan menghitung rata rata harga buku, program menjumlahkan harga buku buku yang ada di rak tersebut, lalu membaginya dengan jumlah buku per rak (y). Hasil rata-rata disimpan dalam slice hargaRataRata. Setelah itu program akan menampilkan hasilnya. Output berupa tiga baris di mana baris pertama merupakan rata rata harga per rak, baris kedua merupakan harga termahal dalam rupiah dan baris ketiga merupakan harga termurah dalam rupiah.

### 3.) Source Code

```
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

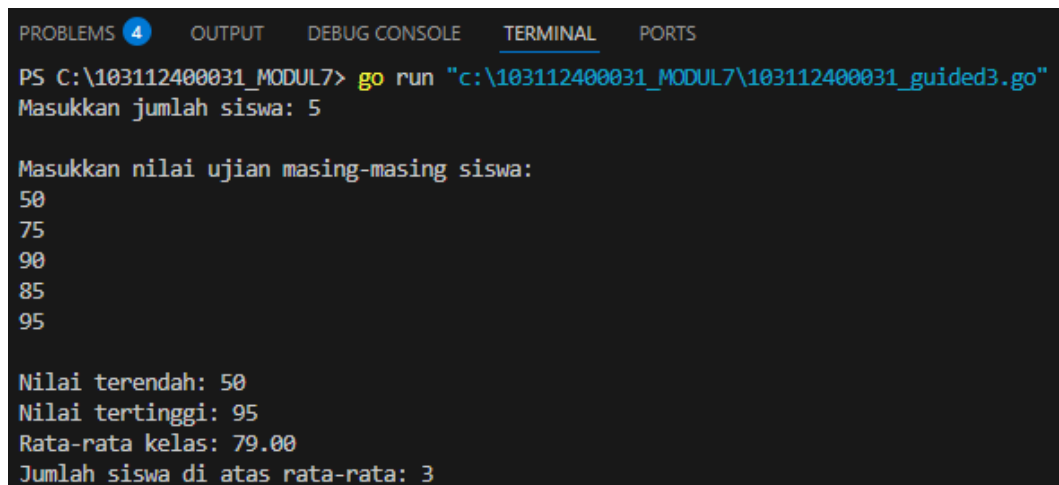
    fmt.Println("\nMasukkan nilai ujian masing-masing siswa: ")
    for i := 0; i < n; i++ {
        fmt.Scan(&nilaiSiswa[i])
        totalNilai += nilaiSiswa[i]
    }

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



Output :



```
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400031_MODUL7> go run "c:\103112400031_MODUL7\103112400031_guided3.go"
Masukkan jumlah siswa: 5

Masukkan nilai ujian masing-masing siswa:
50
75
90
85
95

Nilai terendah: 50
Nilai tertinggi: 95
Rata-rata kelas: 79.00
Jumlah siswa di atas rata-rata: 3
```

Penjelasan : Program ini bertujuan untuk menampilkan nilai tertinggi, nilai terendah, menghitung rata rata kelas, dan menentukan jumlah siswa yang nilainya di atas rata rata. Program meminta kita menginput jumlah siswa, lalu program meminta kita menginput nilai ujian masing masing siswa. Data nilai siswa itu disimpan dalam array dengan kapasitas 200, max digunakan untuk mencari nilai tertinggi, min digunakan untuk mencari nilai terendah. Nilai awal min dan max diambil dari data pertama, lalu dibandingkan dengan data lain menggunakan perulangan for. Jika ada yang lebih kecil dari min, maka nilai min diperbarui. Jika ada yang lebih besar dari max, maka nilai max diperbarui. Program juga akan menghitung rata rata kelas dengan menjumlahkan seluruh nilai siswa yang akan dijadikan total nilai, lalu membagi total nilai dengan jumlah siswa berdasarkan inputan. Lalu program juga akan menentukan jumlah siswa yang nilainya di atas rata rata dengan perulangan for di bagian yang sama dengan min max, jika nilai lebih dari rata rata maka akan bertambah satu. Setelah itu program akan menampilkan hasilnya. Output berupa empat baris di mana baris pertama merupakan nilai terendah, baris kedua merupakan nilai tertinggi, baris ketiga merupakan rata rata kelas, dan baris keempat merupakan jumlah siswa di atas rata rata.

### III. UNGUIDED

#### 1.) Source code

```
// Savila Nur Fadilla
// 103112400031

package main

import "fmt"

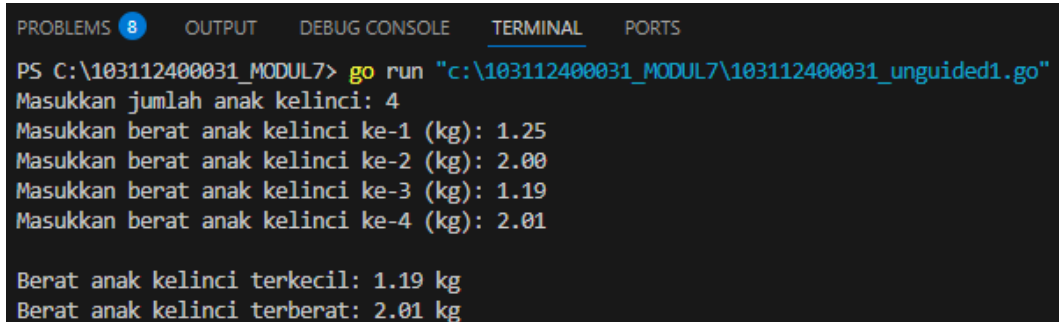
func main() {
    var N int
    fmt.Print("Masukkan jumlah anak kelinci: ")
    fmt.Scan(&N)

    var beratKelinci [1000]float64
    for i := 0; i < N; i++ {
        fmt.Printf("Masukkan berat anak kelinci ke-%v (kg): ", i+1)
        fmt.Scan(&beratKelinci[i])
    }

    min, max := beratKelinci[0], beratKelinci[0]
    for i := 1; i < N; i++ {
        if beratKelinci[i] < min {
            min = beratKelinci[i]
        }
        if beratKelinci[i] > max {
            max = beratKelinci[i]
        }
    }

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

Output :



The screenshot shows a terminal window with a dark background. At the top, there are tabs for 'PROBLEMS' (with a blue circle containing the number 8), 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is active), and 'PORTS'. The terminal text shows the command to run a Go program: `PS C:\103112400031_MODUL7> go run "c:\103112400031_MODUL7\103112400031_unguided1.go"`. The program prompts for the number of children and their weights. The user enters 4, and then four weights: 1.25, 2.00, 1.19, and 2.01. The program then outputs the minimum weight (1.19 kg) and the maximum weight (2.01 kg).

```
PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400031_MODUL7> go run "c:\103112400031_MODUL7\103112400031_unguided1.go"
Masukkan jumlah anak kelinci: 4
Masukkan berat anak kelinci ke-1 (kg): 1.25
Masukkan berat anak kelinci ke-2 (kg): 2.00
Masukkan berat anak kelinci ke-3 (kg): 1.19
Masukkan berat anak kelinci ke-4 (kg): 2.01

Berat anak kelinci terkecil: 1.19 kg
Berat anak kelinci terbesar: 2.01 kg
```

Penjelasan : Program ini bertujuan untuk menampilkan berat anak kelinci terkecil dan terberat. Program meminta kita menginput jumlah anak kelinci, lalu program meminta kita menginput berat masing masing anak kelinci (dalam kg). Data berat kelinci itu disimpan dalam array dengan kapasitas 1000, max digunakan untuk mencari berat anak kelinci terberat, min digunakan untuk mencari berat anak kelinci terkecil. Nilai awal min dan max diambil dari data pertama, lalu dibandingkan dengan data lain menggunakan perulangan for. Jika ada yang lebih kecil dari min, maka nilai min diperbarui. Jika ada yang lebih besar dari max, maka nilai max diperbarui. Setelah itu program akan menampilkan hasilnya. Output berupa dua baris di mana baris pertama merupakan berat anak kelinci terkecil dalam kg dan baris kedua merupakan berat anak kelinci terberat dalam kg.

## 2.) Source code

```
// Savila Nur Fadilla
// 103112400031

package main

import "fmt"

func main() {
    var x, y int
    fmt.Print("Masukkan banyak ikan yang akan dijual: ")
    fmt.Scan(&x)

    fmt.Print("Masukkan kapasitas ikan per wadah: ")
    fmt.Scan(&y)

    var beratIkan [1000]float64
    for i := 0; i < x; i++ {
        fmt.Printf("Masukkan berat ikan ke-%v (kg): ", i+1)
        fmt.Scan(&beratIkan[i])
    }

    jumlahWadah := (x + y - 1) / y
    var totalWadah [1000]float64
    var totalBerat float64

    for i := 0; i < x; i++ {
        wadahKe := i / y
        totalWadah[wadahKe] += beratIkan[i]
    }

    fmt.Println("\nTotal berat ikan per wadah:")
    for i := 0; i < jumlahWadah; i++ {
        fmt.Printf("Wadah %v: %.2f kg\n", i+1, totalWadah[i])
        totalBerat += totalWadah[i]
    }

    rataRata := totalBerat / float64(jumlahWadah)
    fmt.Printf("\nRata-rata berat ikan per wadah: %.2f kg\n", rataRata)
}
```

Output :

```
PROBLEMS 9 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400031_MODUL7> go run "c:\103112400031_MODUL7\103112400031_unguided2.go"
Masukkan banyak ikan yang akan dijual: 6
Masukkan kapasitas ikan per wadah: 2
Masukkan berat ikan ke-1 (kg): 2
Masukkan berat ikan ke-2 (kg): 1.5
Masukkan berat ikan ke-3 (kg): 3
Masukkan berat ikan ke-4 (kg): 2.5
Masukkan berat ikan ke-5 (kg): 2
Masukkan berat ikan ke-6 (kg): 1.5

Total berat ikan per wadah:
Wadah 1: 3.50 kg
Wadah 2: 5.50 kg
Wadah 3: 3.50 kg

Rata-rata berat ikan per wadah: 4.17 kg
```

Penjelasan : Program ini bertujuan untuk menghitung rata-rata berat ikan per wadah dan menghitung berat ikan per wadah. Program meminta kita menginput banyak ikan yang akan dijual dan kapasitas ikan per wadah, lalu program meminta kita berat setiap ikan. Data berat ikan itu disimpan dalam array dengan kapasitas 1000. Program akan menghitung jumlah wadah yang digunakan dengan rumus  $\text{jumlahWadah} := (x + y - 1) / y$ , rumus ini untuk membulatkan ke atas jika ikan tidak habis dibagi maka sisanya akan dimasukkan ke wadah baru. Lalu program akan mengelompokkan ikan ke dalam wadah dengan rumus  $\text{wadahKe} := i / y$  untuk menentukan ikan ke  $i$  masuk ke wadah seberapa berdasarkan urutan, setelah ditentukan berdasarkan urutan program akan menentukan total berat ikan per wadah. Program juga akan menghitung rata rata berat ikan per wadah, (membagi total berat dengan jumlah wadah). Setelah itu program akan menampilkan hasilnya. Output berupa total berat ikan per wadah dalam kg dan rata rata berat ikan per wadah dalam kg.

### 3.) Source code

```
// Savila Nur Fadilla
// 103112400031

package main

import "fmt"

type arrBalita [100]float64

func hitungMinMax(arrBerat arrBalita, n int, bMin, bMax *float64) {
    *bMin, *bMax = arrBerat[0], arrBerat[0]

    for i := 1; i < n; i++ {
        if arrBerat[i] < *bMin {
            *bMin = arrBerat[i]
        }
        if arrBerat[i] > *bMax {
            *bMax = arrBerat[i]
        }
    }
}

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

func main() {
    var data arrBalita
    var n int

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

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

    var min, max float64
```

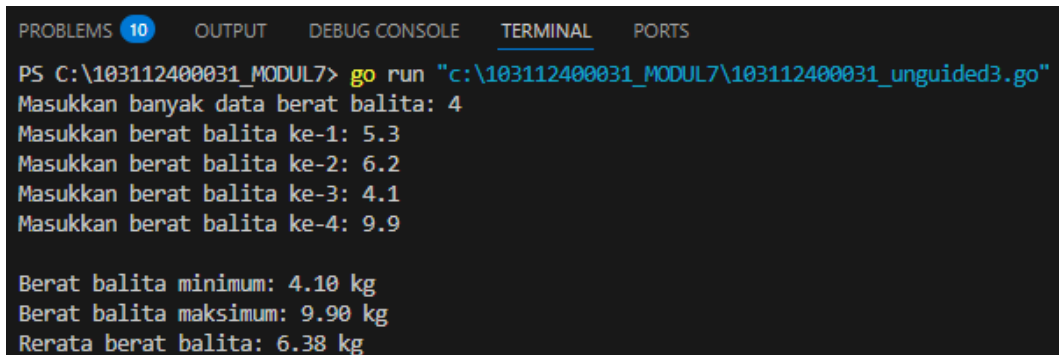
```

hitungMinMax(data, n, &min, &max)
rataRata := rerata(data, n)

fmt.Printf("\nBerat balita minimum: %.2f kg\n", min)
fmt.Printf("Berat balita maksimum: %.2f kg\n", max)
fmt.Printf("Rerata berat balita: %.2f kg\n", rataRata)
}

```

Output :



```

PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400031_MODUL7> go run "c:\103112400031_MODUL7\103112400031_unguided3.go"
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 minimum: 4.10 kg
Berat balita maksimum: 9.90 kg
Rerata berat balita: 6.38 kg

```

Penjelasan : Program ini bertujuan untuk menampilkan berat balita minimum, berat balita maksimum, dan rata rata berat balita. Program meminta kita menginput jumlah atau banyak data berat balita, lalu program meminta kita menginput berat masing masing balita. Data balita itu disimpan dalam array dengan kapasitas 100, max digunakan untuk mencari nilai tertinggi, min digunakan untuk mencari nilai terendah. Nilai awal min dan max diambil dari data pertama, lalu dibandingkan dengan data lain menggunakan perulangan for. Jika ada yang lebih kecil dari min, maka nilai min diperbarui. Jika ada yang lebih besar dari max, maka nilai max diperbarui. Program juga akan menghitung rata rata berat balita dengan menjumlahkan seluruh berat balita yang akan dijadikan total, lalu membagi total berat dengan jumlah atau banyak data berdasarkan inputan. Setelah itu program akan menampilkan hasilnya. Output berupa tiga baris di mana baris pertama merupakan berat balita minimum dalam kg, baris kedua merupakan berat balita maksimum dalam kg, dan baris ketiga merupakan rata rata berat balita dalam kg.

#### **IV. KESIMPULAN**

Berdasarkan laporan ini, dapat disimpulkan jika soal soal tersebut menggunakan pencarian nilai min atau max, array, fungsi atau subprogram, perulangan for, if else, struct. Struct dalam pemrograman Bahasa go digunakan untuk membuat tipe data yang lebih kompleks dengan mengelompokkan data serta menggabungkan berbagai tipe data yang berbeda. Array digunakan untuk menyimpan beberapa nilai dengan tipe data yang sama dalam satu variable. Penggunaan fungsi juga supaya program lebih mudah dipahami, lebih terstruktur, dan jadi lebih rapih. Min dan Max digunakan untuk mencari nilai minimum dan maksimum.



## **REFERENSI**

Telkom University. (2025). *Modul Praktikum Algoritma dan Pemrograman 2*.

<https://dasarpemrogramangolang.novalagung.com/>