

**ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 7**

**NILAI MAX/MIN**



Oleh:

M.HANIF AL FAIZ

103112400042

12IF-01

**S1 TEKNIK INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2025**

## **I. DASAR TEORI**

Pemrograman dengan bahasa Go (Golang) menekankan prinsip kesederhanaan, efisiensi, dan keunggulan dalam menangani konkruensi. Bahasa ini memiliki fitur-fitur utama seperti penentuan tipe data saat kompilasi, pengelolaan memori otomatis melalui *garbage collection*, serta kemampuan untuk mengembalikan lebih dari satu nilai dari suatu fungsi. Dalam praktiknya, Go memanfaatkan konsep dasar pemrograman seperti perulangan, rekursi, percabangan, serta struktur data seperti array dan slice. Selain itu, validasi input merupakan aspek penting guna mencegah terjadinya kesalahan dalam program.

## II. UNGUIDED

1.

```
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[1] > max {  
        max = tinggiTanaman[i]  
    }  
}  
  
fmt.Printf("\nTinggi tanaman tertinggi: %.2f cm\n", max)  
fmt.Printf("Tinggi tanaman terpendek: %.2f cm\n", min)  
}
```

OUTPUT:

```
Masukkan jumlah tanaman: 3  
Masukkan tinggi tanaman ke-1 (cm): 3  
Masukkan tinggi tanaman ke-2 (cm): 4  
Masukkan tinggi tanaman ke-3 (cm): 5  
  
Tinggi tanaman tertinggi: 4.00 cm  
Tinggi tanaman terpendek: 3.00 cm
```

DESKRIPSI: Program ini menghitung tinggi tanaman tertinggi dan terpendek dari data yang dimasukkan pengguna.

2.

```
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.Println("\nMasukkan harga setiap buku (dalam ribuan Rp): ")
    for i := 0; i < x; i++ {
        fmt.Scan(&hargaBuku[i])spesi
    }

    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: %.2f Rp\n", max)
fmt.Printf("Harga termurah: %.2f Rp\n", min)
}

```

OUTPUT:

```

PS D:\LAPRAKKKKK> go run "d:\LAPRAKKKKK\103112400042_MODUL7\cosol2.go"
Masukkan jumlah buku dan jumlah buku per rak: 3 4

Masukkan harga setiap buku (dalam ribuan Rp):
3000
3000
30000

Rata-rata harga per rak: 9000.00
Harga termahal: 30000.00 Rp
Harga termurah: 3000.00 Rp

```

### DESKRIPSI:

Program ini menghitung rata-rata harga buku per rak, serta menentukan harga buku termahal dan termurah dari data yang dimasukkan pengguna.

3.

```

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:

```
PS D:\LAPRAKKKKK> go run "d:\LAPRAKKKKK\103112400042_MODUL7\cosol3.go"
Masukkan jumlah siswa: 4

Masukkan nilai ujian masing-masing siswa:
40
70
90
100

Nilai terendah: 40
Nilai tertinggi: 100
Rata-rata kelas: 75.00
Jumlah siswa di atas rata-rata: 2
```

DESKRIPSI:

Program ini digunakan untuk menganalisis nilai ujian siswa, termasuk menghitung nilai rata-rata, nilai tertinggi, nilai terendah, dan jumlah siswa yang nilainya di atas rata-rata

### III. GUIDED

1.

```
// M.HANIF AL FAIZ

// 103112400042

package main
```

```
import (  
    "fmt"  
)  
  
func main() {  
    var n int  
  
    fmt.Print("Masukkan jumlah anak kelinci yang akan ditimbang: ")  
    fmt.Scan(&n)  
  
    var beratKelinci [1000]float64  
  
    fmt.Println("\nMasukkan berat setiap anak kelinci (dalam kg):")  
    for i := 0; i < n; i++ {  
        fmt.Printf("Berat kelinci ke-%d: ", i+1)  
        fmt.Scan(&beratKelinci[i])  
    }  
  
    min, max := beratKelinci[0], beratKelinci[0]  
    total := 0.0  
    for i := 0; i < n; i++ {  
        if beratKelinci[i] < min {  
            min = beratKelinci[i]  
        }  
        if beratKelinci[i] > max {  
            max = beratKelinci[i]  
        }  
    }  
}
```

```

    }

    total += beratKelinci[i]
}

rataRata := total / float64(n)

fmt.Printf("\nBerat kelinci terkecil: %.2f kg\n", min)
fmt.Printf("Berat kelinci terbesar: %.2f kg\n", max)
fmt.Printf("Rata-rata berat kelinci: %.2f kg\n", rataRata)
}

```

OUTPUT:

```

PS D:\LAPRAKKKKK> go run "d:\LAPRAKKKKK\103112400042_MODUL7\latsol1.go"
Masukkan jumlah anak kelinci yang akan ditimbang: 3

Masukkan berat setiap anak kelinci (dalam kg):
Berat kelinci ke-1: 4
Berat kelinci ke-2: 5
Berat kelinci ke-3: 6

Berat kelinci terkecil: 4.00 kg
Berat kelinci terbesar: 6.00 kg
Rata-rata berat kelinci: 5.00 kg

```

DESKRIPSI:

Program ini digunakan untuk menentukan berat terkecil dan terbesar dari sejumlah anak kelinci yang akan ditimbang

2.

```
// M.HANIF AL FAIZ
// 103112400042

package main

import (
    "fmt"
)

func main() {
    var x, y int

    fmt.Print("Masukkan jumlah ikan (x) dan jumlah ikan per wadah (y): ")
    fmt.Scan(&x, &y)

    var beratIkan [1000]float64

    fmt.Println("\nMasukkan berat setiap ikan (dalam kg):")

    for i := 0; i < x; i++ {
        fmt.Printf("Berat ikan ke-%d: ", i+1)
        fmt.Scan(&beratIkan[i])
    }

    var totalBeratWadah []float64

    wadahCount := (x + y - 1) / y
```

```

for i := 0; i < wadahCount; i++ {
    total := 0.0
    start := i * y
    end := start + y
    if end > x {
        end = x
    }
    for j := start; j < end; j++ {
        total += beratIkan[j]
    }
    totalBeratWadah = append(totalBeratWadah, total)
}

fmt.Println("\nTotal berat ikan di setiap wadah:")
for i, total := range totalBeratWadah {
    fmt.Printf("Wadah %d: %.2f kg\n", i+1, total)
}

totalBerat := 0.0
for _, berat := range totalBeratWadah {
    totalBerat += berat
}

rataRata := totalBerat / float64(len(totalBeratWadah))

```

```
    fmt.Printf("\nBerat rata-rata ikan di setiap wadah: %.2f kg\n", rataRata)
}
```

#### OUTPUT:

```
PS D:\LAPRAK\103112400042_MODUL7> go run "d:\LAPRAK\103112400042_MODUL7\tempCodeRunnerFile.go"
Masukkan jumlah ikan (x) dan jumlah ikan per wadah (y): 3 4

Masukkan berat setiap ikan (dalam kg):
Berat ikan ke-1: 3
Berat ikan ke-2: 3
Berat ikan ke-3: 3

Total berat ikan di setiap wadah:
Wadah 1: 9.00 kg

Berat rata-rata ikan di setiap wadah: 9.00 kg
PS D:\LAPRAK\103112400042_MODUL7>
```

#### DESKRIPSI:

Program ini digunakan untuk menghitung total berat ikan di setiap wadah dan rata-rata berat ikan per wadah berdasarkan data yang dimasukkan pengguna

3

```
// M.HANIF AL FAIZ

// 103112400042

package main
```

```
import (
    "fmt"
)

type arrBalita [100]float64

func hitungMinMax(arrBerat arrBalita, n int, bMin, bMax *float64) {
    *bMin = arrBerat[0]
    *bMax = 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 {
    total := 0.0
    for i := 0; i < n; i++ {
        total += arrBerat[i]
    }
}
```

```

    return total / float64(n)
}

func countAboveAverage(arrBerat arrBalita, n int, rataRata float64) int {
    count := 0
    for i := 0; i < n; i++ {
        if arrBerat[i] > rataRata {
            count++
        }
    }
    return count
}

func main() {
    var n int
    var berat arrBalita
    var bMin, bMax float64

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

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

```



```

}

hitungMinMax(berat, n, &bMin, &bMax)

rataRata := rerata(berat, n)

count := countAboveAverage(berat, n, rataRata)

fmt.Printf("\nBerat balita minimum: %.2f kg\n", bMin)
fmt.Printf("Berat balita maksimum: %.2f kg\n", bMax)
fmt.Printf("Rerata berat balita: %.2f kg\n", rataRata)
fmt.Printf("Jumlah balita dengan berat di atas rata-rata: %d\n", count)
}

```

OUTPUT:

```

PS D:\LAPRAK\> go run "d:\LAPRAK\10311240028_MODUL7\latsol3.go"
Masukan banyak data berat balita: 3
Masukan berat balita ke-1: 3
Masukan berat balita ke-2: 3
Masukan berat balita ke-3: 3

Berat balita minimum: 3.00 kg
Berat balita maksimum: 3.00 kg
Rerata berat balita: 3.00 kg

```

DESKRIPSI:

Program ini digunakan untuk mencatat data berat balita, kemudian menghitung berat balita terkecil, terbesar, dan rata-rata berat balita..

#### **IV. KESIMPULAN**

Dokumen ini menyajikan berbagai contoh penerapan bahasa Go dalam pemrograman. Penggunaan algoritma yang efisien—seperti rekursi untuk operasi perkalian dan pencarian bilangan ganjil—serta optimasi dalam perhitungan menunjukkan bagaimana kompleksitas dapat ditekan. Beragam program yang dikembangkan memperlihatkan kemampuan Go dalam menyelesaikan berbagai persoalan, mulai dari perhitungan aritmetika sederhana hingga sistem berbasis pola seperti pemberian hadiah. Validasi input dan penanganan kasus tepi turut meningkatkan keandalan program. Secara keseluruhan, Go menawarkan solusi pemrograman yang efisien dan sangat cocok bagi mahasiswa yang ingin memahami konsep-konsep dasar hingga menengah dalam dunia pemrograman.

## **REFERENSI**

Programmer Zaman Now. "ALGORITMA & STRUKTUR DATA - SEMESTER 2." YouTube, 2023, [https://youtu.be/IO\\_vkyJnMas?si=4jSJTZ5Zd2Fi5Vd2](https://youtu.be/IO_vkyJnMas?si=4jSJTZ5Zd2Fi5Vd2).