

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL 7



**DISUSUN OLEH:
RIZKINA AZIZAH
103112400082
S1 IF-12-01**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025**

GUIDED

1. Guided 1

Source Code:

```
package main

import (
    "fmt"
)

type arrBalita [100]float64

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

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

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

```

func main() {
    var beratBalita arrBalita
    var jumlahData int

    // Input jumlah data berat balita
    fmt.Print("Masukan banyak data berat balita: ")
    fmt.Scan(&jumlahData)

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

    // Variabel untuk menyimpan berat minimum dan maksimum
    var beratMin, beratMax float64

    // Menghitung berat minimum dan maksimum
    hitungMinMax(beratBalita, &beratMin, &beratMax, jumlahData)

    // Menghitung rerata berat balita
    rata := rerata(beratBalita, jumlahData)

    // Menampilkan hasil
    fmt.Printf("Berat balita minimum: %.2f kg\n", beratMin)
    fmt.Printf("Berat balita maksimum: %.2f kg\n", beratMax)
    fmt.Printf("Rerata berat balita: %.2f kg\n", rata)
}

```

Deskripsi Program:

- Program ini digunakan untuk mencatat serta menganalisis data berat balita
- *type arrBalita [100]float64* sebagai array dengan kapasitas 100 untuk menyimpan berat balita dalam satuan kilogram.

- Fungsi *hitungMinMax* menghitung dan mengubah nilai berat minimum dan maksimum dari array *arrBerat*
- Fungsi *rerata* Menghitung dan mengembalikan rerata berat balita dari array *arrBerat*.
- Fungsi *main* berfungsi untuk
 - Mengambil input jumlah dan berat balita.
 - Memanggil *hitungMinMax* untuk mendapatkan berat minimum dan maksimum..
 - Memanggil *rerata* untuk menghitung rerata berat..
 - Menampilkan hasil

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

    var hargaRatarata []float64
    for i := 0; i < x; i++ {
```

```

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)
}

```

Deskripsi Program:

- Program ini digunakan untuk menghitung dan menampilkan rata-rata harga buku per rak, serta harga termahal dan termurah dari sekumpulan buku.
- Mengambil input jumlah buku (x) dan jumlah buku per rak (y) dari pengguna
- Menginisialisasi array hargaBuku untuk menyimpan harga buku.
- Meminta pengguna untuk memasukkan harga setiap buku ke dalam array hargaBuku

- Menghitung rata-rata harga per rak dengan menjumlahkan harga buku dalam kelompok rak dan membaginya dengan y, lalu menyimpan hasilnya dalam slice hargaRatarata.
- Menghitung rata-rata harga per rak dengan menjumlahkan harga buku dalam kelompok rak dan membaginya dengan y, lalu menyimpan hasilnya dalam slice hargaRatarata.
- Menampilkan rata-rata harga per rak, harga termahal, dan harga termurah ke layar.

3. Guided

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 diatas rata-rata: %d\n", diAtasRataRata)
}

```

Deskripsi Program:

- Program ini digunakan untuk membaca nilai ujian sejumlah siswa, kemudian menghitung dan menampilkan nilai terkecil, terbesar, rata-rata kelas, serta jumlah siswa yang nilainya di atas rata-rata. Mengambil input jumlah buku (x) dan jumlah buku per rak (y) dari pengguna
- Struktur kode :
 - Input jumlah siswa dan nilai nilai
 - Hitung total, minimum, maksimum, dan jumlah yang di atas rata-rata
 - Menampilkan hasil

UNGUIDED

1. Unguided 1

Source Code:

```
//Rizkina Azizah 103112400082

package main

import "fmt"

func main() {
    var N int
    var berat [1000]float64
    fmt.Print("jumlah anak kelinci: ")
    fmt.Scan(&N)
    for i := 0; i < N; i++ {
        fmt.Printf("berat anak kelinci ke-%d (kg): ", i+1)
        fmt.Scan(&berat[i])
    }

    min, max := berat[0], berat[0]
    for i := 1; i < N; i++ {
        if berat[i] < min {
            min = berat[i]
        }
        if berat[i] > max {
            max = berat[i]
        }
    }
    fmt.Println("Berat kelinci terkecil:", min)
    fmt.Println("Berat kelinci terbesar:", max)
}
```

Deskripsi Program:

- Program ini digunakan untuk menghitung dan merekap jumlah suara masuk dan suara sah dari input angka.
- var N int: Variabel untuk menyimpan jumlah anak kelinci.
- var berat [1000]float64: Array untuk menyimpan berat anak kelinci, dengan kapasitas maksimum 1000.
- Program meminta pengguna untuk memasukkan jumlah anak kelinci (N).
- Menggunakan loop for, program meminta pengguna untuk memasukkan berat setiap anak kelinci satu per satu. Berat disimpan dalam array berat.
- Menginisialisasi min dan max dengan berat anak kelinci pertama (berat[0]).
- Menggunakan loop for untuk membandingkan setiap berat kelinci dengan min dan max. Jika berat kelinci lebih kecil dari min, maka min diperbarui. Jika lebih besar dari max, maka max diperbarui.
- Program menampilkan berat kelinci terkecil dan terbesar ke layar

2. Latihan 2

Source Code:

```
//RizkinaAzizah 103112400082
package main

import "fmt"

func main() {
    var jumlahIkan, jumlahWadah int
    const maxCapacity = 1000
    var beratIkan [maxCapacity]float64

    fmt.Scan(&jumlahIkan, &jumlahWadah)

    for i := 0; i < jumlahIkan; i++ {
        fmt.Scan(&beratIkan[i])
    }

    var totalBerat [maxCapacity]float64
```

```

var jumlahDalamWadah [maxCapacity]int

for i := 0; i < jumlahIkan; i++ {
    indeksWadah := i / jumlahWadah
    totalBerat[indeksWadah] += beratIkan[i]
    jumlahDalamWadah[indeksWadah]++
}

for i := 0; i < (jumlahIkan+jumlahWadah-1)/jumlahWadah; i++ {
    fmt.Printf("%.2f ", totalBerat[i])
}
fmt.Println()

for i := 0; i < (jumlahIkan+jumlahWadah-1)/jumlahWadah; i++ {
    if jumlahDalamWadah[i] > 0 {
        beratRata := totalBerat[i] / float64(jumlahDalamWadah[i])
        fmt.Printf("%.2f ", beratRata)
    } else {
        fmt.Printf("0.00 ")
    }
}
fmt.Println()
}

```

Deskripsi program:

- Program ini digunakan untuk menentukan tarif ikan yang akan dijual ke pasar.
- jumlahIkan: Menyimpan jumlah ikan yang akan dijual.
- jumlahWadah: Menyimpan jumlah wadah untuk ikan.
- beratIkan: Array untuk menyimpan berat ikan.
- totalBerat: Array untuk menyimpan total berat ikan di setiap wadah.
- jumlahDalamWadah: Array untuk menghitung jumlah ikan dalam setiap wadah.
- indeksWadah: Menentukan indeks wadah berdasarkan urutan ikan

3. Latihan 3

Source Code:

```
//Rizkina Azizah
package main

import "fmt"

type arrBalita [100]float64

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

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

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

func main() {
    var beratBalita arrBalita
```

```

var jumlahData int

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

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

var beratMin, beratMax float64

hitungMinMax(beratBalita, &beratMin, &beratMax, jumlahData)

rata := rerata(beratBalita, jumlahData)

fmt.Printf("Berat balita minimum: %.2f kg\n", beratMin)
fmt.Printf("Berat balita maksimum: %.2f kg\n", beratMax)
fmt.Printf("Rerata berat balita: %.2f kg\n", rata)
}

```

Deskripsi Program:

- Program ini digunakan untuk mencatat data berat balita (dalam kg).
- *arrBalita* didefinisikan sebagai array dengan kapasitas 100 untuk menyimpan berat balita.
- Fungsi *hitungMinMax* digunakan Menghitung berat minimum dan maksimum dari array *arrBerat*. Fungsi ini menerima pointer untuk *bMin* dan *bMax* agar dapat mengubah nilai di luar fungsi.
- Fungsi *rerata* untuk menghitung dan mengembalikan rerata berat balita dalam array.
- Fungsi *main* berfungsi untuk:
 - Meminta input jumlah data berat balita.
 - Meminta input jumlah data berat balita.

- Memanggil fungsi *hitungMinMax* untuk mendapatkan berat minimum dan maksimum.
- Memanggil fungsi *rerata* untuk menghitung rerata berat balita
- Menampilkan hasil