LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2

MODUL VIII

PENCARIAN NILAI EKSTRIM PADA HIMPUNAN DATA

Disusun Oleh :

Siti Madina Halim Siregar / 2311102243

S1IF-11-05

Dosen Pengampu :

Arif Amrulloh, S.Kom., M.Kom.

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Pencarian nilai ekstrim dalam pemrograman dan matematika adalah proses untuk menemukan nilai maksimum dan minimum dari suatu fungsi atau himpunan data.

Pencarian nilai ekstrem (minimum dan maksimum) pada array melibatkan identifikasi elemen terkecil atau terbesar dalam array. Pendekatan ini berbeda sedikit tergantung pada jenis array: array bertipe dasar (seperti integer, float, dll.) dan array bertipe terstruktur (seperti array objek atau array dengan tipe data kompleks).

- Array Bertipe Dasar

  Array bertipe dasar adalah array yang berisi elemen-elemen dengan tipe data primitif, seperti int, float, double, dan lain-lain.
- Array Bertipe Terstruktur

  Array bertipe terstruktur adalah array yang berisi elemen dengan struktur lebih kompleks, seperti array objek atau tuple dengan beberapa atribut.

Perbedaan utama

Aspek	Array Bertipe Dasar	Array Bertipe Terstruktur
Elemen	Nilai primitif (angka, karakter, dll.)	Objek atau struktur kompleks
Kriteria pencarian	Berdasarkan nilai elemen secara langsung	Berdasarkan atribut tertentu
Pendekatan	Iterasi sederhana	Iterasi dengan seleksi berdasarkan atribut
Fungsi Bawaan	`min()` dan `max()` langsung	`min()` dan `max()` dengan parameter `key`

Pengertian Golang

Golang adalah bahasa pemrograman open-source yang memiliki sintaksis sederhana namun kuat, memungkinkan pengembang untuk menulis kode dengan cepat dan efisien. Bahasa ini menggunakan tipe data statis dan menghasilkan kode biner yang dikompilasi, sehingga dapat berjalan dengan cepat dan efisien.

II. UNGUIDED

1. Soal Studi Case

Sebuah program digunakan untuk mendata berat anak kelinci yang akan dijual ke pasar. Program ini menggunakan array dengan kapasitas 1000 untuk menampung data berat anak kelinci yang akan dijual.Masukan terdiri dari sekumpulan bilangan, yang mana bilangan pertama adalah bilangan bulat N yang menyatakan banyaknya anak kelinci yang akan ditimbang beratnya. Selanjutnya N bilangan riil berikutnya adalah berat dari anak kelinci yang akan dijual.

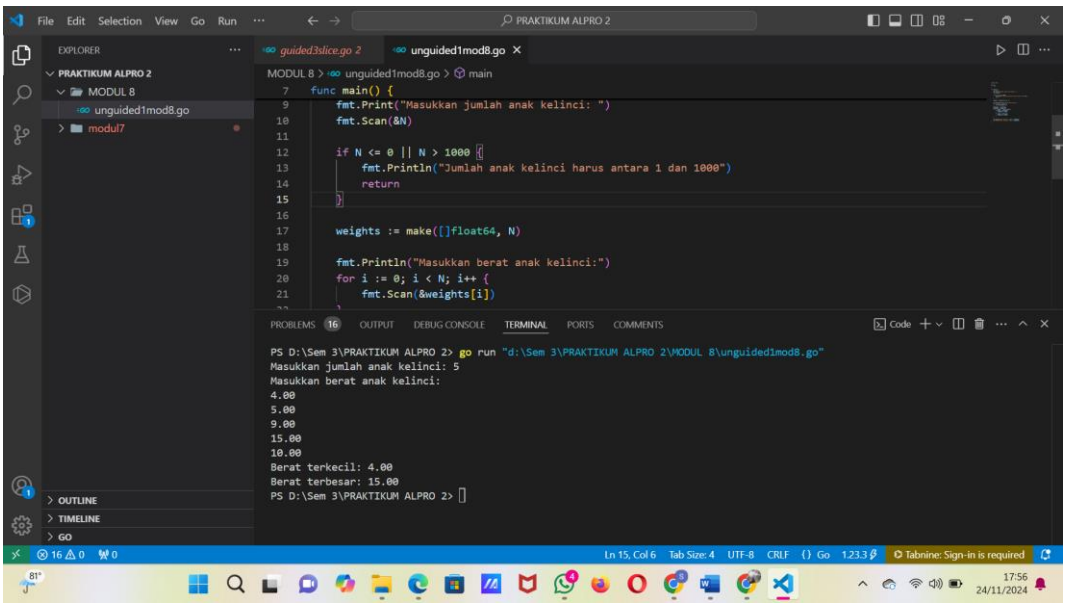
Keluaran terdiri dari dua buah bilangan riil yang menyatakan berat kelinci terkecil dan terbesar.

Sourcecode

```
package main

import (
    "fmt"
)

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

    if N <= 0 || N > 1000 {
        fmt.Println("Jumlah anak kelinci harus antara 1 dan
1000")
        return
    }

    weights := make([]float64, N)

    fmt.Println("Masukkan berat anak kelinci:")
    for i := 0; i < N; i++ {
        fmt.Scan(&weights[i])
    }
    minWeight := weights[0]
    maxWeight := weights[0]
    for _, weight := range weights {
        if weight < minWeight {
            minWeight = weight
        }
        if weight > maxWeight {
```

```
            maxWeight = weight
        }
    }

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

Screenshoot Output



Deskripsi Program

- Program meminta jumlah anak kelinci yang akan ditimbang.
- Jika jumlah valid, program meminta berat masing-masing kelinci dan menyimpannya dalam array.
- Program kemudian menentukan berat terkecil dan terbesar dengan cara: Membandingkan setiap elemen dalam array dengan nilai ekstrem saat ini (minWeight dan maxWeight).
- Loop for _, weight := range weights: Loop ini iterasi setiap elemen dalam array weights.
- Program mencetak hasil berupa berat terkecil dan terbesar.

2. Soal Studi Case

Sebuah program digunakan untuk menentukan tarif ikan yang akan dijual ke pasar. Program ini menggunakan array dengan kapasitas 1000 untuk menampung data berat ikan yang akan dijual.

Masukan terdiri dari dua baris, yang mana baris pertama terdiri dari dua bilangan bulat x dan y. Bilangan x menyatakan banyaknya ikan yang akan dijual, sedangkan y adalah banyaknya ikan yang akan dimasukan ke dalam wadah. Baris kedua terdiri dari sejumlah x bilangan riil sejumlah x bilangan riil yang menyatakan banyaknya ikan yang akan dijual.

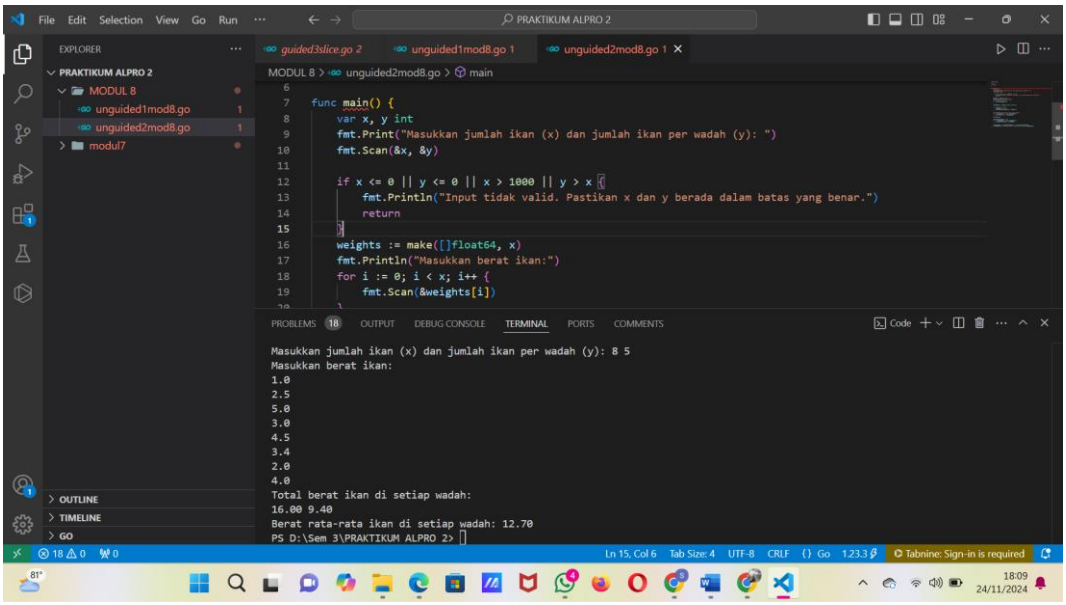
Keluaran terdiri dari dua baris. Baris pertama adalah kumpulan bilangan riil yang menyatakan total berat ikan di setiap wadah (jumlah wadah tergantung pada nilai x dan y, urutan ikan yang dimasukan ke dalam wadah sesuai urutan pada masukan baris ke-2). Baris kedua adalah sebuah bilangan riil yang menyatakan berat rata-rata ikan di setiap wadah.

Sourcecode

```
 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)

    if x <= 0 || y <= 0 || x > 1000 || y > x {
        fmt.Println("Input tidak valid. Pastikan x dan y
berada dalam batas yang benar.")
        return
    }
    weights := make([]float64, x)
    fmt.Println("Masukkan berat ikan:")
    for i := 0; i < x; i++ {
        fmt.Scan(&weights[i])
    }

    totalWeights := make([]float64, (x+y-1)/y)
```

```
    for i := 0; i < x; i++ {
        wadahIndex := i / y
        totalWeights[wadahIndex] += weights[i]
    }

    fmt.Println("Total berat ikan di setiap wadah:")
    for _, totalWeight := range totalWeights {
        fmt.Printf("%.2f ", totalWeight)
    }
    fmt.Println()

    totalWeightAllWadah := 0.0
    for _, totalWeight := range totalWeights {
        totalWeightAllWadah += totalWeight
    }

    averageWeight := totalWeightAllWadah /
float64(len(totalWeights))
    fmt.Printf("Berat rata-rata ikan di setiap wadah:
%.2f\n", averageWeight)
}
```

Screenshoot Output



Deskripsi Program

- Input Data:
  - Pertama, program membaca dua bilangan bulat x dan y. x adalah jumlah ikan yang akan dijual, dan y adalah jumlah ikan yang akan dimasukkan ke dalam satu wadah.
  - Kedua, program membaca daftar berat ikan yang akan dijual dari pengguna.
- Validasi Input:
  - Jika x atau y tidak valid (misalnya y lebih besar dari x atau nilai x dan y kurang dari atau sama dengan 0), program akan menampilkan pesan kesalahan dan berhenti.
- Menghitung Total Berat di Setiap Wadah:
  - Program mengalokasikan array totalWeights untuk menampung total berat ikan di setiap wadah. Jumlah wadah dihitung dengan rumus (x + y - 1) / y, yang memastikan kita mendapatkan jumlah wadah yang cukup untuk menampung semua ikan.
  - Berat ikan dimasukkan ke dalam wadah dengan cara mengiterasi berat ikan, dan setiap ikan dimasukkan ke wadah yang sesuai berdasarkan indeksnya (i / y).
- Output:
  - Baris pertama menampilkan total berat ikan di setiap wadah.
  - Baris kedua menghitung dan menampilkan berat rata-rata ikan per wadah, yaitu total berat di semua wadah dibagi jumlah wadah.

.

3. Soal Studi Case

Pos Pelayanan Terpadu (posyandu) sebagai tempat pelayanan kesehatan perlu mencatat data berat bailta (dalam kg). Petugas akan memasukkan data tersebut ke dalam array. Dari data yang diperoleh akan dicari berat balita terkecil, terbesar, dan reratanya.

Buatlah program dengan spesifikasi subprogram sebagai berikut:

Perhatikan sesi interaksi pada contoh berikut ini (teks bergaris bawah adalah Input/read

```
type arrBalita [100]float64

func hitungMinMax(arrBerat arrBalita; bMin, bMax *float64) {
/* I.S. Terdefinisi array dinamis arrBerat
  Proses: Menghitung berat minimum dan maksimum dalam array
  F.S. Menampilkan berat minimum dan maksimum balita */
    ...
}

function rerata (arrBerat arrBalita) real {
/* menghitung dan mengembalikan rerata berat balita dalam array */
    ...
}
```

Perhatikan sesi interaksi pada contoh dibawah ini (bergaris bawah adalah input)

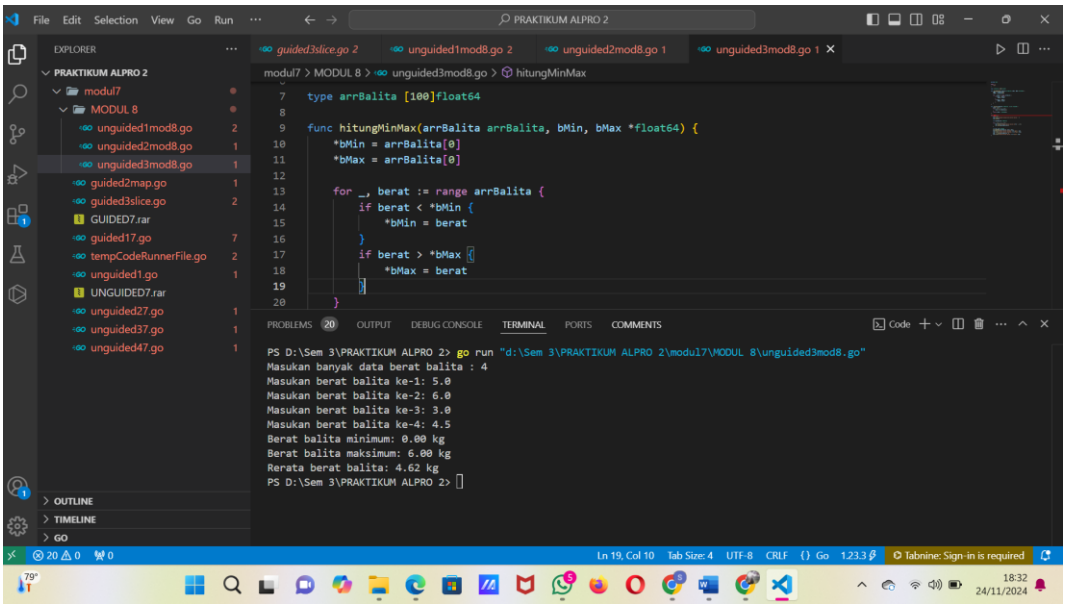
```
Masukan banyak data berat balita : 4
Masukan berat balita ke-1: 5.3
Masukan berat balita ke-2: 6.2
Masukan berat balita ke-3: 4.1
Masukan berat balita ke-4: 9.9
Berat balita minimum: 4.10 kg
Berat balita maksimum: 9.90 kg
Rerata berat balita: 6.38 kg
```

Sourcecode

```
package main

import (
    "fmt"
)

type arrBalita [100]float64

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

    for _, berat := range arrBalita {
        if berat < *bMin {
            *bMin = berat
        }
        if berat > *bMax {
            *bMax = berat
        }
    }
}

func rerata(arrBalita arrBalita, n int) float64 {
    total := 0.0
    for i := 0; i < n; i++ {
        total += arrBalita[i]
    }
    return total / float64(n)
}
func main() {
    var x int
    fmt.Print("Masukan banyak data berat balita : ")
    fmt.Scan(&x)
    var beratBalita arrBalita
    for i := 0; i < x; i++ {
        fmt.Printf("Masukan berat balita ke-%d: ", i+1)
        fmt.Scan(&beratBalita[i])
    }

    var min, max float64
    hitungMinMax(beratBalita, &min, &max)
    fmt.Printf("Berat balita minimum: %.2f kg\n", min)
```

```
    fmt.Printf("Berat balita maksimum: %.2f kg\n", max)
    avg := rerata(beratBalita, x)
    fmt.Printf("Rerata berat balita: %.2f kg\n", avg)
}
```

Screenshoot Output



Deskripsi Program

- Mendeklarasikan tipe data arrBalita yang merupakan array dengan kapasitas 100 elemen untuk menyimpan berat balita dalam tipe float64.
- hitungMinMax: Fungsi ini menerima array arrBalita dan dua pointer bMin serta bMax untuk mencari berat balita terkecil dan terbesar. Fungsi ini mengiterasi seluruh array dan membandingkan tiap elemen untuk menemukan nilai terkecil dan terbesar.
- rerata: Fungsi ini menerima array arrBalita dan jumlah data n, lalu menghitung rata-rata berat balita dengan menjumlahkan semua elemen dalam array dan membaginya dengan jumlah elemen. Mengambil input jumlah data berat balita (x), kemudian meminta input berat balita satu per satu.
- Fungsi hitungMinMax dipanggil untuk mendapatkan berat balita terkecil dan terbesar, sementara fungsi rerata digunakan untuk menghitung rata-rata berat balita.
- Output berupa berat balita terkecil, terbesar, dan rata-rata berat balita ditampilkan di layar.

Kesimpulan

Pencarian nilai ekstrem (minimum dan maksimum) pada array melibatkan identifikasi elemen terkecil atau terbesar dalam array. Pendekatan ini berbeda sedikit tergantung pada jenis array: array bertipe dasar (seperti integer, float, dll.) dan array bertipe terstruktur (seperti array objek atau array dengan tipe data kompleks).