

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN**

MODUL XI

PENCARTIAN NILAI EKSTRIM PADA HIMPUNAN DATA



Disusun Oleh :

SHAFa ADILA SANTOSO / 2311102158

IF-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

Pencarian adalah aktivitas umum yang sering dilakukan dalam kehidupan sehari-hari. Contohnya termasuk mencari file di dalam folder komputer, menemukan teks dalam dokumen, dan banyak lagi. Pencarian yang dipelajari pada modul ini merupakan pencarian nilai ekstrim yang merupakan salah satu algoritma pencarian yang digunakan untuk mencari nilai terkecil dan terbesar dari suatu kumpulan data. Untuk algoritmanya sendiri cukup sederhana karena datanya diproses secara sequential, jadi nilai atau indeks dari nilai maksimum yang sudah diproses untuk dibandingkan dengan data berikutnya. Dimana nilai yang sudah tersimpan hingga akhir algoritma merupakan nilai maksimum diacri. Algoritma secara umum:

1. Jadikan data pertama sebagai nilai ekstrim
2. Lakukan validasi nilai ekstrim dari data kedua hingga data terakhir
3. Apabila semua data telah dicek maka nilai ekstrim yang dimiliki adalah valid

Pencarian Nilai Ekstrim

1. Pencarian pada Array bertipe data dasar

Untuk menentukan nilai ekstrim pada array bertipe data dasar dapat dimulai dengan menginisialisasi nilai ekstrem dengan elemen pertama dari array. Iterasikan setiap elemen dari elemen kedua hingga terakhir, bandingkan dengan nilai ekstrem yang ada. Jika elemen saat ini lebih ekstrim, perbarui nilai ekstrim tersebut. Setelah semua elemen diperiksa, nilai ekstrem yang tersisa adalah hasilnya. Misalnya terdefinisi sebuah array of Integer dengan kapasitas 2023, dan array berisi sejumlah N bilangan bulat kemudian pencarian nilai terkecil dilakukan pada array tersebut.

2. Pencarian pada array bertipe data terstruktur

Untuk menentukan nilai ekstrim dalam array bertipe data terstruktur, inisialisasi nilai ekstrim dengan elemen pertama dari array berdasarkan atribut yang akan dibandingkan. Iterasikan semua elemen, dan bandingkan atribut tersebut. Perbarui nilai ekstrem jika elemen saat ini lebih ekstrim. Nilai ekstrim yang tersisa setelah seluruh elemen diperiksa adalah hasilnya. Sebagai contoh misalnya terdapat array yang digunakan untuk menyimpan data mahasiswa, kemudian terdapat fungsi IPK yang digunakan untuk mencari data mahasiswa dengan IPK tertinggi.

II. UNGUIDED

1. 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_158 int
    fmt.Print("Jumlah anak kelinci: ")
    fmt.Scanln(&n_158)

    if n_158 > 0 && n_158 <= 1000 {
        berat := make([]float64, n_158)

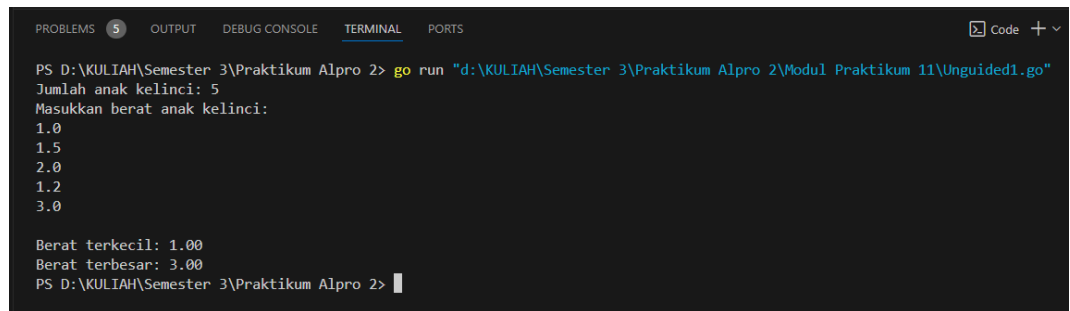
        fmt.Println("Masukkan berat anak kelinci:")
        for i := 0; i < n_158; i++ {
            fmt.Scan(&berat[i])
        }

        terkecil := berat[0]
        terbesar := berat[0]

        for i := 1; i < n_158; i++ {
            if berat[i] < terkecil {
                terkecil = berat[i]
            }
            if berat[i] > terbesar {
                terbesar = berat[i]
            }
        }

        fmt.Printf("\nBerat terkecil: %.2f\n", terkecil)
        fmt.Printf("Berat terbesar: %.2f\n", terbesar)
    } else {
        fmt.Println("Jumlah anak kelinci harus antara 1 hingga 1000.")
    }
}
```

Screenshot Output

A screenshot of a terminal window showing the execution of a Go program. The terminal has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The command 'go run "d:\KULIAH\Semester 3\Praktikum Alpro 2\Modul Praktikum 11\Unguided1.go"' is entered. The program prompts for the number of cats (5) and then for the weight of each cat (1.0, 1.5, 2.0, 1.2, 3.0). It then outputs the minimum weight (1.00) and the maximum weight (3.00).

```
PS D:\KULIAH\Semester 3\Praktikum Alpro 2> go run "d:\KULIAH\Semester 3\Praktikum Alpro 2\Modul Praktikum 11\Unguided1.go"
Jumlah anak kelinci: 5
Masukkan berat anak kelinci:
1.0
1.5
2.0
1.2
3.0

Berat terkecil: 1.00
Berat terbesar: 3.00
PS D:\KULIAH\Semester 3\Praktikum Alpro 2>
```

Deskripsi Program

Program diatas merupakan program Go yang digunakan untuk mencari berat terkecil dan terbesar kelinci yang akan dijual. Program dimulai dengan meminta input jumlah anak kelinci (n) yang harus berada dalam rentang 1 hingga 1000. Jika jumlah tersebut valid, pengguna diminta memasukkan berat setiap anak kelinci, yang disimpan dalam sebuah array bernama berat. Program menggunakan algoritma iterasi untuk membandingkan setiap berat dalam array, dengan inisialisasi nilai awal berat terkecil dan terbesar dari elemen pertama array. Saat iterasi berlangsung, nilai berat terkecil akan diperbarui jika ditemukan nilai yang lebih kecil, dan nilai berat terbesar akan diperbarui jika ditemukan nilai yang lebih besar. Setelah semua data diproses, program menampilkan berat terkecil dan terbesar dalam format dua angka desimal. Jika input jumlah anak kelinci tidak sesuai, program memberikan pesan kesalahan.

2. 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 dimasukkan ke dalam wadah. Baris kedua terdiri dari 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 dimasukkan 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_158, y_158 int

    fmt.Print("Masukkan total ikan dan kapasitas maksimum
dalam wadah: ")
    fmt.Scanln(&x_158, &y_158)

    if x_158 > 0 && x_158 <= 1000 {
        berat := make([]float64, x_158)
        jumlahIkan := make([]int, x_158)

        fmt.Println("Masukkan berat ikan:")
        for i := 0; i < x_158; i++ {
            fmt.Printf("Berat ikan ke-%d: ", i+1)
            fmt.Scanln(&berat[i])
            jumlahIkan[i] = y_158
        }

        totalBerat := make([]float64, x_158)
        for i := 0; i < x_158; i++ {
            totalBerat[i] = berat[i] *
float64(jumlahIkan[i])
        }

        fmt.Println("\nTotal berat ikan di setiap wadah:")
        for i := 0; i < x_158; i++ {
            fmt.Printf("Wadah %d: %.2f kg\n", i+1,
totalBerat[i])
        }

        fmt.Println("\nBerat rata-rata ikan di setiap
wadah:")
        for i := 0; i < x_158; i++ {
            rataRataPerWadah := berat[i]
            fmt.Printf("Wadah %d: %.2f kg\n", i+1,
rataRataPerWadah)
        }

    } else {
        fmt.Println("Jumlah wadah harus antara 1 hingga
1000.")
    }
}

```

Screenshot Output

```
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\KULIAH\Semester 3\Praktikum Alpro 2> go run "d:\KULIAH\Semester 3\Praktikum Alpro 2\Modul Praktikum 11\Unguided2.go"
Masukkan total ikan dan kapasitas maksimum dalam wadah: 3 10
Masukkan berat ikan:
Berat ikan ke-1: 2.0
Berat ikan ke-2: 1.5
Berat ikan ke-3: 2.5

Total berat ikan di setiap wadah:
Wadah 1: 20.00 kg
Wadah 2: 15.00 kg
Wadah 3: 25.00 kg

Berat rata-rata ikan di setiap wadah:
Wadah 1: 2.00 kg
Wadah 2: 1.50 kg
Wadah 3: 2.50 kg
PS D:\KULIAH\Semester 3\Praktikum Alpro 2> |
```

Deskripsi Program

Program diatas merupakan program Go yang dibuat untuk menghitung tiotal berat ikan dan berat rata rata ikan disetiap wadah berdasarkan berat ikan dan kapasitas maksimum perwadah yang dimasukkan oleh pengguna. Program dimulai dengan meminta input jumlah total ikan (x) dan kapasitas maksimum ikan dalam satu wadah (y). Jika jumlah ikan valid (antara 1 hingga 1000), program meminta pengguna untuk memasukkan berat setiap ikan, yang kemudian disimpan dalam array berat. Kapasitas maksimum untuk setiap ikan di setiap wadah juga diatur dalam array jumlahIkan. Program menghitung total berat ikan per wadah dengan mengalikan berat setiap ikan dengan kapasitasnya, menyimpannya dalam array totalBerat, dan menampilkannya. Selain itu, program menghitung berat rata-rata ikan per wadah yang sebenarnya sama dengan berat individu ikan karena kapasitas yang sama digunakan, dan hasilnya juga ditampilkan. Jika input jumlah ikan tidak valid, program memberikan pesan kesalahan.

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

Buatlah program dengan spesifikasi subprogram sebagai berikut:

```
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 */
...
}
```

Sourcecode

```

package main

import "fmt"

func hitungMinMax_158(arrBalita []float64) (float64,
float64, float64) {
    bMin := arrBalita[0]
    bMax := arrBalita[0]
    total := 0.0

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

    rataRata := total / float64(len(arrBalita))
    return bMin, bMax, rataRata
}

func main() {
    var n int
    fmt.Print("Masukkan jumlah data berat balita: ")
    fmt.Scanln(&n)

    if n > 0 {
        arrBalita := make([]float64, n)

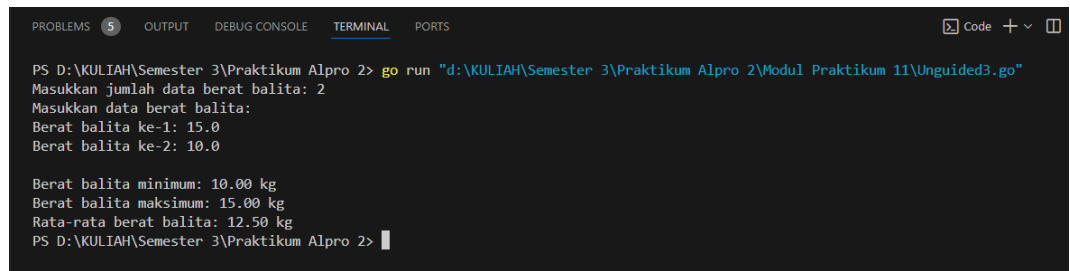
        fmt.Println("Masukkan data berat balita:")
        for i := 0; i < n; i++ {
            fmt.Printf("Berat balita ke-%d: ", i+1)
            fmt.Scanln(&arrBalita[i])
        }

        bMin, bMax, rataRata :=
hitungMinMax_158(arrBalita)

        fmt.Printf("\nBerat balita minimum: %.2f kg\n",
bMin)
        fmt.Printf("Berat balita maksimum: %.2f kg\n",
bMax)
        fmt.Printf("Rata-rata berat balita: %.2f kg\n",
rataRata)
    } else {
        fmt.Println("Jumlah balita harus lebih dari 0.")
    }
}

```

Screenshoot Output



```
PS D:\KULIAH\Semester 3\Praktikum Alpro 2> go run "d:\KULIAH\Semester 3\Praktikum Alpro 2\Modul Praktikum 11\Unguided3.go"
Masukkan jumlah data berat balita: 2
Masukkan data berat balita:
Berat balita ke-1: 15.0
Berat balita ke-2: 10.0

Berat balita minimum: 10.00 kg
Berat balita maksimum: 15.00 kg
Rata-rata berat balita: 12.50 kg
PS D:\KULIAH\Semester 3\Praktikum Alpro 2> |
```

Deskripsi Program

Program diatas merupakan program Go yang digunakan untuk menghitung berat minimum, maksimum, dan rata-rata dari sejumlah data berat balita. Program dimulai dengan meminta input jumlah data berat balita (n). Jika jumlah valid (lebih dari 0), pengguna diminta memasukkan berat masing-masing balita, yang disimpan dalam array `arrBalita`. Fungsi `hitungMinMax_158` digunakan untuk mencari nilai minimum, maksimum, dan rata-rata dengan algoritma iterasi, di mana setiap berat dibandingkan untuk memperbarui nilai minimum dan maksimum, serta dijumlahkan untuk menghitung rata-rata. Hasil akhir berupa berat minimum, maksimum, dan rata-rata ditampilkan. Jika jumlah balita tidak valid, program menampilkan pesan kesalahan.

DAFTAR PUSTAKA

- [1] Asisten Praktikum, “Modul 11 PENCARTIAN NILAI EKSTRIM PADA HIMPUNAN DATA”, Modul Praktikum, 2024