

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL XI

Pencarian Nilai Ekstrim pada Himpunan Data



Disusun Oleh :

Alfin Adriansyah/ 2311102264

S1IF_11_06

Dosen Pengampu :

ABEDNEGO DWI SEPTIADI

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Pencarian merupakan proses yang umum dilakukan dalam kehidupan sehari-hari. Contohnya, pencarian file dalam direktori komputer, pencarian teks dalam dokumen, pencarian buku di rak, dan berbagai situasi lainnya. Dalam pembahasan awal modul ini, akan diperkenalkan algoritma untuk mencari nilai terkecil atau terbesar dari sekumpulan data, yang sering disebut pencarian nilai ekstrim. Konsep algoritmanya sangat sederhana. Karena data diproses secara berurutan, nilai atau indeks dari nilai maksimum yang ditemukan disimpan dan dibandingkan dengan elemen berikutnya. Nilai yang tetap tersimpan hingga proses selesai adalah nilai maksimum yang dicari.

II. GUIDED

1. Tipedasar

Sourcecode

```
package main

import "fmt"

// Mendeklarasikan tipe data array arrInt dengan panjang
2023
type arrInt [2023]int

// Fungsi untuk mencari indeks elemen terkecil dalam
array
func terkecil(tabInt arrInt, n int) int {
    var idx int = 0 // idx menyimpan indeks elemen
    terkecil
    var j int = 1
    for j < n {
        if tabInt[idx] > tabInt[j] {
            idx = j // Simpan indeks j jika elemen
di indeks j lebih kecil
        }
        j = j + 1
    }
    return idx
}

// Fungsi main untuk menguji fungsi terkecil
func main() {
    var n int
    var tab arrInt

    // Meminta input jumlah elemen array
    fmt.Print("Masukkan jumlah elemen (maks 2023): ")
    fmt.Scan(&n)

    // Validasi input jumlah elemen
    if n < 1 || n > 2023 {
        fmt.Println("Jumlah elemen harus antara 1 dan
2023.")
        return
    }

    // Memasukkan elemen-elemen array
    fmt.Println("Masukkan elemen-elemen array:")
    for i := 0; i < n; i++ {
        fmt.Print("Elemen ke-", i+1, ": ")
        fmt.Scan(&tab[i])
    }
}
```

```

        // Memanggil fungsi terkecil untuk menemukan indeks
        elemen terkecil
        idxMin := terkecil(tab, n)

        // Menampilkan nilai dan indeks terkecil
        fmt.Println("Nilai terkecil dalam array adalah:",
        tab[idxMin], "pada indeks:", idxMin)
    }

```

Screenshoot Output

```

Masukkan jumlah elemen (maks 2023): 5
Masukkan elemen-elemen array:
Elemen ke-1: 1
Elemen ke-2: 2
Elemen ke-3: 3
Elemen ke-4: 4
Elemen ke-5: 5
Nilai terkecil dalam array adalah: 1 pada indeks: 0
PS D:\ALPRO2> 

```

Deskripsi Program

Program Go ini dibuat untuk mencari dan menampilkan nilai terkecil beserta indeksnya dari sekumpulan angka yang dimasukkan pengguna ke dalam array dengan kapasitas maksimal 2023 elemen.

2. Tipe Struktur

Sourcecode

```

package main

import "fmt"

// Definisi struct mahasiswa dengan atribut nama, nim,
kelas, jurusan, dan ipk
type mahasiswa struct {
    nama, nim, kelas, jurusan string
    ipk                        float64
}

// Definisi tipe data array mahasiswa dengan kapasitas
maksimal 2023
type arrMhs [2023]mahasiswa

// Fungsi untuk mencari IPK tertinggi dalam array
mahasiswa
func ipk(T arrMhs, n int) float64 {
    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
    }

    // Fungsi main untuk mengisi data mahasiswa dan mencari
    // IPK tertinggi
    func main() {
        var n int
        var dataMhs arrMhs

        // Meminta input jumlah mahasiswa
        fmt.Print("Masukkan jumlah mahasiswa (maks 2023): ")
        fmt.Scan(&n)

        // Validasi jumlah mahasiswa yang dimasukkan
        if n < 1 || n > 2023 {
            fmt.Println("Jumlah mahasiswa harus antara 1
dan 2023.")
            return
        }

        // Mengisi data mahasiswa
        for i := 0; i < n; i++ {
            fmt.Printf("\nMasukkan data mahasiswa ke-
%d\n", i+1)
            fmt.Print("Nama: ")
            fmt.Scan(&dataMhs[i].nama)
            fmt.Print("NIM: ")
            fmt.Scan(&dataMhs[i].nim)
            fmt.Print("Kelas: ")
            fmt.Scan(&dataMhs[i].kelas)
            fmt.Print("Jurusan: ")
            fmt.Scan(&dataMhs[i].jurusan)
            fmt.Print("IPK: ")
            fmt.Scan(&dataMhs[i].ipk)
        }

        // Mencari dan menampilkan IPK tertinggi
        tertinggi := ipk(dataMhs, n)
        fmt.Printf("\nIPK tertinggi dari %d mahasiswa
adalah: %.2f\n", n, tertinggi)
    }

```

Screenshoot Output

```
PS D:\ALPRO2> go run "d:\ALPRO2\Modul 11\Guided\Tip
Masukkan jumlah mahasiswa (maks 2023): 2

Masukkan data mahasiswa ke-1
Nama: alfin
NIM: 2311102264
Kelas: if06
Jurusan: informatika
IPK: 4.0

Masukkan data mahasiswa ke-2
Nama: fahrial
NIM: 2311102241
Kelas: if06
Jurusan: informatika
IPK: 4.0

IPK tertinggi dari 2 mahasiswa adalah: 4.00
PS D:\ALPRO2> █
```

Deskripsi Program

Program Go ini dirancang untuk mengelola data mahasiswa (nama, NIM, kelas, jurusan, dan IPK) dengan menggunakan struktur data array berkapasitas 2023 mahasiswa. Program meminta pengguna memasukkan data sejumlah mahasiswa yang diinginkan, kemudian menghitung dan menampilkan nilai IPK tertinggi dari seluruh mahasiswa yang telah diinput.

III. 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"

type arrayBerat [1000]float64

func terkecil(tabel arrayBerat, n int) float64 {
    var kecil float64 = tabel[0]
    var i int = 1
    for i < n {
        if tabel[i] < kecil {
            kecil = tabel[i]
        }
        i = i + 1
    }
    return kecil
}

func terbesar(tabel arrayBerat, n int) float64 {
    var besar float64 = tabel[0]
    var i int = 1
    for i < n {
        if tabel[i] > besar {
            besar = tabel[i]
        }
        i = i + 1
    }
    return besar
}

func main() {
    var jumlah int
    var beratKelinci arrayBerat

    fmt.Print("Masukkan jumlah kelinci (maksimal 1000):")
    ")
```

```

        fmt.Scan(&jumlah)

        // Validasi input
        if jumlah < 1 || jumlah > 1000 {
            fmt.Println("Jumlah kelinci harus antara 1
sampai 1000")
            return
        }

        fmt.Println("Masukkan berat kelinci (dalam kg):")
        for i := 0; i < jumlah; i++ {
            fmt.Printf("Berat kelinci ke-%d: ", i+1)
            fmt.Scan(&beratKelinci[i])
        }

        beratTerkecil := terkecil(beratKelinci, jumlah)
        beratTerbesar := terbesar(beratKelinci, jumlah)

        fmt.Printf("\nBerat kelinci terkecil: %.2f kg\n",
beratTerkecil)
        fmt.Printf("Berat kelinci terbesar: %.2f kg\n",
beratTerbesar)
    }
}

```

Screenshoot Output

```

PS D:\ALPRO2> go run "d:\ALPRO2\Modul 11\ungu
Masukkan jumlah kelinci (maksimal 1000): 6
Masukkan berat kelinci (dalam kg):
Berat kelinci ke-1: 12
Berat kelinci ke-2: 11
Berat kelinci ke-3: 10
Berat kelinci ke-4: 9
Berat kelinci ke-5: 12
Berat kelinci ke-6: 13

Berat kelinci terkecil: 9.00 kg
Berat kelinci terbesar: 13.00 kg
PS D:\ALPRO2> 

```


Deskripsi Program

Program Go ini dirancang untuk mendata berat anak kelinci yang akan dijual ke pasar dengan kapasitas maksimal 1000 data. Program meminta input jumlah dan berat masing-masing kelinci, kemudian menggunakan fungsi terkecil dan terbesar untuk mencari dan menampilkan berat kelinci teringan dan terberat dari semua data yang dimasukkan.

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"

type arrayBerat [1000]float64

func hitungRataRata(wadah []float64) float64 {
    var total float64
    for _, berat := range wadah {
        total += berat
    }
    return total / float64(len(wadah))
}

func main() {
    var jumlahIkan, ikanPerWadah int
    var beratIkan arrayBerat

    fmt.Print("Masukkan jumlah ikan dan ikan per wadah: ")
}
```

```

        fmt.Scan(&jumlahIkan, &ikanPerWadah)

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

        jumlahWadah := (jumlahIkan + ikanPerWadah - 1) /
        ikanPerWadah
        wadah := make([]float64, jumlahWadah)

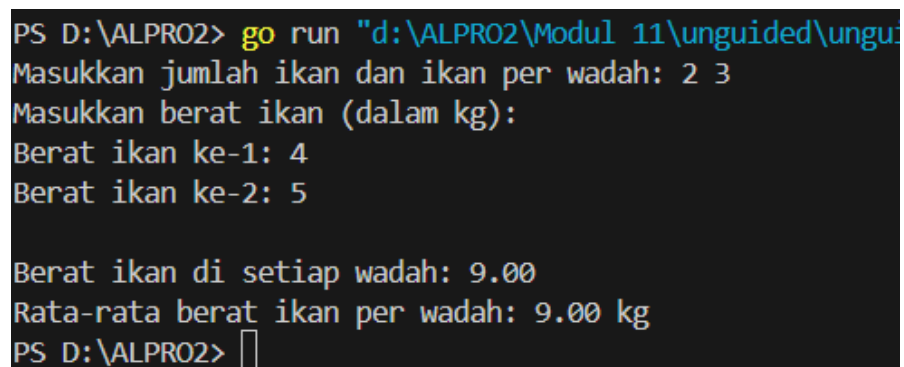
        for i := 0; i < jumlahIkan; i++ {
            wadahKe := i / ikanPerWadah
            wadah[wadahKe] += beratIkan[i]
        }

        fmt.Print("\nBerat ikan di setiap wadah: ")
        for _, berat := range wadah {
            fmt.Printf("%.2f ", berat)
        }

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

```

Screenshoot Output



```

PS D:\ALPRO2> go run "d:\ALPRO2\Modul 11\unguided\ungui
Masukkan jumlah ikan dan ikan per wadah: 2 3
Masukkan berat ikan (dalam kg):
Berat ikan ke-1: 4
Berat ikan ke-2: 5

Berat ikan di setiap wadah: 9.00
Rata-rata berat ikan per wadah: 9.00 kg
PS D:\ALPRO2> 

```

Deskripsi Program

Program Go ini dirancang untuk menentukan tarif ikan yang akan dijual ke pasar dengan kapasitas maksimal 1000 ikan. Program menerima input berupa jumlah ikan total (x) dan jumlah ikan per wadah (y), serta berat masing-masing ikan. Program kemudian menghitung dan menampilkan total berat ikan di setiap wadah serta rata-rata berat ikan per wadah

menggunakan fungsi `hitungRataRata` untuk membantu penentuan tarif penjualan.

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 reratanya.

Sourcecode

```
package main

import "fmt"

type arrBalita [100]float64

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

    for i := 1; i < 4; i++ {
        if arrBerat[i] < *bMin {
            *bMin = arrBerat[i]
        }
        if arrBerat[i] > *bMax {
            *bMax = arrBerat[i]
        }
    }
}

func rerata(arrBerat arrBalita) float64 {
    var total float64
    for i := 0; i < 4; i++ {
        total += arrBerat[i]
    }
    return total / 4
}

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

    fmt.Printf("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(&beratBalita[i])
    }
    hitungMinMax(beratBalita, &bMin, &bMax)
    avg := rerata(beratBalita)

    fmt.Printf("Berat balita minimum: %.2f kg\n",
bMin)
    fmt.Printf("Berat balita maksimum: %.2f kg\n",
bMax)
    fmt.Printf("Rerata berat balita: %.2f kg\n",
avg)
}

```

Screenshoot Output

```

PS D:\ALPRO2> go run "d:\ALPRO2\Modul 11\unguided\unguided3.go"
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
PS D:\ALPRO2> 

```

Deskripsi Program

Program Go ini dirancang untuk mencatat dan menganalisis data berat balita di sebuah Posyandu. Program memiliki kapasitas menyimpan 100 data balita dalam array dan dilengkapi dengan dua fungsi utama: **hitungMinMax** untuk mencari berat terendah dan tertinggi balita, serta fungsi **rerata** untuk menghitung rata-rata berat semua balita. Program meminta input jumlah balita dan berat masing-masing balita, kemudian menampilkan hasil analisis berupa berat minimum, maksimum, dan rata-rata dari seluruh data yang dimasukkan..