

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL 11

PENCARIAN NILAI EXTRIM PADA HUMPUNAN DATA



Disusun Oleh :

**Aryo Tegar Sukarno / 2311102018
11 – IF – 6**

Dosen Pengampu :

Abednego Dwi Septiadi S. Kom. M. Kom.

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Pendahuluan

Pencarian nilai ekstrim (maksimum dan minimum) dalam suatu himpunan data merupakan operasi yang umum dilakukan dalam pemrograman. Dalam bahasa Golang, kita dapat memanfaatkan berbagai cara untuk mencapai tujuan ini.

Konsep Dasar

- **Himpunan Data:** Kumpulan data yang biasanya direpresentasikan dalam bentuk array atau slice.
- **Nilai Ekstrim:** Nilai terbesar (maksimum) dan terkecil (minimum) dalam suatu himpunan data.
- **Algoritma:** Langkah-langkah sistematis untuk menyelesaikan suatu masalah.

Metode Pencarian Nilai Ekstrim

Menggunakan Perulangan:

- **Ide:** Melakukan iterasi pada setiap elemen dalam himpunan data, membandingkan nilai setiap elemen dengan nilai maksimum dan minimum yang telah disimpan.
- **Kelebihan:** Fleksibel untuk berbagai jenis data.
- **Kekurangan:** Kinerja kurang optimal untuk data yang sangat besar.

Kesimpulannya adalah pencarian nilai ekstrim adalah operasi dasar dalam pemrograman. Golang menyediakan berbagai cara untuk mencapai tujuan ini. Pemilihan metode yang tepat akan bergantung pada konteks permasalahan yang sedang dihadapi.

II. GUIDED I

Soal Studi Case

XXXXXXX

Sourcecode

```
package main

import "fmt"

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

// Fungsi untuk mencari elemen terkecil dalam array
func terkecil(tabInt arrInt, n int) int {
    var min int = tabInt[0]
    var j int = 1
    for j < n {
        if min > tabInt[j] {
            min = tabInt[j]
        }
        j = j + 1
    }
    return min
}

// Fungsi main
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 nilai
    terkecil
```

```

        minVal := terkecil(tab, n)

        // Menampilkan nilai terkecil
        fmt.Println("Nilai terkecil dalam array adalah:",
minVal)
    }

```

Screenshoot Output

```

PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_Modul 11> go run .
go: go.mod file not found in current directory or any parent directory; see 'go help modules'
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_Modul 11> go run TipeDasar.go
Masukkan jumlah elemen (maks 2023): 5
Masukkan elemen-elemen array:
Elemen ke-1: 3
Elemen ke-2: 2
Elemen ke-3: 5
Elemen ke-4: 6
Elemen ke-5: 7
Nilai terkecil dalam array adalah: 2
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_Modul 11>

```

Deskripsi Program

Program ini dirancang untuk mencari dan menampilkan elemen terkecil dari sebuah array integer yang dimasukkan oleh pengguna. Panjang maksimum array adalah 2023 elemen, dan program memvalidasi jumlah elemen yang diinput oleh pengguna.

III. GUIDED II

Soal Studi Case

Xxxxxxx

Sourcecode

```

2311102018
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

```

go: go.mod file not found in current directory or any parent directory; see 'go help modules'
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_Modul 11\Modul 11> go run TipeDasarIDX.go
Masukkan jumlah elemen (maks 2023): 6
Masukkan elemen-elemen array:
Elemen ke-1: 1
Elemen ke-2: 4
Elemen ke-3: 5
Elemen ke-4: 3
Elemen ke-5: 5
Elemen ke-6: 6
Nilai terkecil dalam array adalah: 1 pada indeks: 0
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_Modul 11\Modul 11>

```

Deskripsi Program

Program ini bertujuan untuk menemukan **nilai terkecil** dalam sebuah array integer beserta **indeksnya**. Pengguna dapat memasukkan elemen-elemen array, dan program akan menghitung elemen terkecil menggunakan fungsi yang dirancang khusus.

IV. GUIDED III

Soal Studi Case

XXXXXXX

Sourcecode

```
2311102018
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 indeks IPK tertinggi dalam array
mahasiswa
func indeksIPKTertinggi(T arrMhs, n int) int {
    var idx int = 0 // Inisialisasi indeks IPK
    tertinggi pada indeks pertama
    for j := 1; j < n; j++ {
        if T[idx].ipk < T[j].ipk {
            idx = j // Update indeks jika ditemukan IPK
yang lebih tinggi
        }
    }
    return idx
}

// Fungsi main untuk mengisi data mahasiswa dan mencari
IPK tertinggi beserta indeksnya
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)
}

// Mendapatkan indeks IPK tertinggi
idxTertinggi := indeksIPKTertinggi(dataMhs, n)

// Menampilkan data mahasiswa dengan IPK tertinggi
fmt.Printf("\nMahasiswa dengan IPK tertinggi:\n")
fmt.Printf("Nama: %s\n", dataMhs[idxTertinggi].nama)
fmt.Printf("NIM: %s\n", dataMhs[idxTertinggi].nim)
fmt.Printf("Kelas: %s\n",
dataMhs[idxTertinggi].kelas)
fmt.Printf("Jurusan: %s\n",
dataMhs[idxTertinggi].jurusan)
fmt.Printf("IPK: %.2f\n", dataMhs[idxTertinggi].ipk)
}

```

Screenshoot Output

```
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_Modul 11\Modul 11> go run .
go: go.mod file not found in current directory or any parent directory; see 'go help modules'
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_Modul 11\Modul 11> go run TipeStrukturIPK_IDX.go
Masukkan jumlah mahasiswa (maks 2023): 3

Masukkan data mahasiswa ke-1
Nama: Lexxy
NIM: 2311102015
Kelas: 04
Jurusan: Informatika
IPK: 3.8

Masukkan data mahasiswa ke-2
Nama: Aryo
NIM: 2311102018
Kelas: 05
Jurusan: Informatika
IPK: 4

Masukkan data mahasiswa ke-3
Nama: Rynzee
NIM: 2311102018
Kelas: 2
Jurusan: Informatika
IPK: 4

Mahasiswa dengan IPK tertinggi:
Nama: Aryo
NIM: 2311102018
Kelas: 05
Jurusan: Informatika
IPK: 4.00
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_Modul 11\Modul 11> |
```

Deskripsi Program

Program ini bertujuan untuk menyimpan data mahasiswa (termasuk nama, NIM, kelas, jurusan, dan IPK), kemudian mencari dan menampilkan **IPK tertinggi** dari seluruh data mahasiswa yang telah dimasukkan oleh pengguna.

V. UNGUIDED I

Soal Studi Case

Diberikan sebuah program untuk mencatat berat badan sejumlah anak kelinci yang akan dijual. Program ini menggunakan sebuah daftar (array) dengan kapasitas maksimal 1000 data untuk menyimpan berat badan semua anak kelinci.

Sourcecode

```
//2311102018-Aryo Tegar Sukarno

package main

import (
    "fmt"
)
```



```
func main() {  
  
    const maxCapacity = 1000  
  
    var weights [maxCapacity]float64  
  
    var count int  
  
    for {  
  
        fmt.Println("\nProgram Pencatatan Berat Badan Anak Kelinci")  
  
        fmt.Println("1. Tambah berat badan")  
  
        fmt.Println("2. Lihat semua berat badan")  
  
        fmt.Println("3. Hapus data terakhir")  
  
        fmt.Println("4. Keluar")  
  
        fmt.Print("Pilih opsi: ")  
  
        var choice int  
  
        fmt.Scan(&choice)  
  
  
        switch choice {  
  
        case 1:  
  
            if count >= maxCapacity {  
  
                fmt.Println("Kapasitas maksimum telah tercapai. Tidak bisa  
menambah data lagi.")  
  
                continue  
  
            }  
  
            fmt.Print("Masukkan berat badan anak kelinci (kg): ")  
  
            var weight float64  
  
            fmt.Scan(&weight)  
  
            if weight <= 0 {
```

```
        fmt.Println("Berat badan harus lebih dari 0.")
        continue
    }
    weights[count] = weight
    count++

    fmt.Println("Data berhasil ditambahkan.")

case 2:
    if count == 0 {
        fmt.Println("Tidak ada data yang tercatat.")
    } else {
        fmt.Println("Berat badan anak kelinci yang tercatat:")
        for i := 0; i < count; i++ {
            fmt.Printf("%d. %.2f kg\n", i+1, weights[i])
        }
    }

case 3:
    if count == 0 {
        fmt.Println("Tidak ada data untuk dihapus.")
    } else {
        count--

        fmt.Printf("Data terakhir dengan berat %.2f kg telah
dihapus.\n", weights[count])
    }

case 4:

    fmt.Println("Keluar dari program. Terima kasih!")
```

```

        return

    default:

        fmt.Println("Pilihan tidak valid. Silakan coba lagi.")

    }

}

}

```

Screenshoot Output

```

PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_Modul 11\Modul 11> go run .
go: go.mod file not found in current directory or any parent directory; see 'go help modules'
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_Modul 11\Modul 11> go run unguided1.go

Program Pencatatan Berat Badan Anak Kelinci
1. Tambah berat badan
2. Lihat semua berat badan
3. Hapus data terakhir
4. Keluar
Pilih opsi: 1
Masukkan berat badan anak kelinci (kg): 78
Data berhasil ditambahkan.

Program Pencatatan Berat Badan Anak Kelinci
1. Tambah berat badan
2. Lihat semua berat badan
3. Hapus data terakhir
4. Keluar
Pilih opsi: 1
Masukkan berat badan anak kelinci (kg): 90
Data berhasil ditambahkan.

Program Pencatatan Berat Badan Anak Kelinci
1. Tambah berat badan
2. Lihat semua berat badan
3. Hapus data terakhir
4. Keluar
Pilih opsi: 80
Pilihan tidak valid. Silakan coba lagi.

Program Pencatatan Berat Badan Anak Kelinci
1. Tambah berat badan
2. Lihat semua berat badan
3. Hapus data terakhir
4. Keluar
Pilih opsi: 45
Pilihan tidak valid. Silakan coba lagi.

Program Pencatatan Berat Badan Anak Kelinci
1. Tambah berat badan
2. Lihat semua berat badan
3. Hapus data terakhir
4. Keluar
Pilih opsi: 2
Berat badan anak kelinci yang tercatat:
1. 78.00 kg
2. 90.00 kg

```

Deskripsi Program

Program ini digunakan untuk mencatat, melihat, dan mengelola data berat badan anak kelinci yang dimasukkan oleh pengguna. Program memungkinkan pengguna untuk menambah data, menampilkan semua data yang telah dicatat, menghapus data terakhir, atau keluar dari program.

VI. UNGUIDED 2

Soal Studi Case

Diberikan sebuah program yang digunakan untuk menentukan harga jual ikan di pasar. Program ini menggunakan sebuah array (daftar) dengan kapasitas maksimal 1000 untuk menyimpan data berat ikan yang akan dijual.

Sourcecode

```
//2311102018-Aryo Tegar Sukarno

package main

import "fmt"

const maxCapacity = 1000
const pricePerKg = 50000

func calculateTotalPrice(weights [maxCapacity]float64, n int) float64 {
    var totalPrice float64 = 0
    for i := 0; i < n; i++ {
        totalPrice += weights[i] * pricePerKg
    }
    return totalPrice
}

func main() {
```

```

var weights [maxCapacity]float64

var n int

fmt.Print("Masukkan jumlah ikan yang akan dijual (maks 1000): ")

fmt.Scan(&n)

if n < 1 || n > maxCapacity {

    fmt.Println("Jumlah ikan harus antara 1 dan 1000.")

    return

}

fmt.Println("Masukkan berat ikan (dalam kilogram):")

for i := 0; i < n; i++ {

    fmt.Printf("Berat ikan ke-%d: ", i+1)

    fmt.Scan(&weights[i])

    if weights[i] <= 0 {

        fmt.Println("Berat ikan harus lebih dari 0.")

        i--

    }

}

totalPrice := calculateTotalPrice(weights, n)

fmt.Println("\nDaftar berat ikan dan harga jual:")

for i := 0; i < n; i++ {

    fmt.Printf("Ikan ke-%d: Berat = %.2f kg, Harga = %.2f IDR\n",
i+1, weights[i], weights[i]*pricePerKg)

}

fmt.Printf("\nTotal harga jual semua ikan: %.2f IDR\n", totalPrice)

}

```

Screenshoot Output

```
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_Modul 11\Modul 11> go run unguided2.go
Masukkan jumlah ikan yang akan dijual (maks 1000): 2
Masukkan berat ikan (dalam kilogram):
Berat ikan ke-1: 3
Berat ikan ke-2: 2

Daftar berat ikan dan harga jual:
Ikan ke-1: Berat = 3.00 kg, Harga = 150000.00 IDR
Ikan ke-2: Berat = 2.00 kg, Harga = 100000.00 IDR

Total harga jual semua ikan: 250000.00 IDR
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_Modul 11\Modul 11> |
```

Deskripsi Program

Program ini digunakan untuk menghitung harga jual ikan di pasar berdasarkan berat ikan yang dimasukkan oleh pengguna. Program juga menyediakan total harga jual dari semua ikan yang telah dicatat.

VII. UNGUIDED 3 Soal Studi Case

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

```
//2311102018

package main

import "fmt"

func findMin(weights []float64, n int) float64 {
    min := weights[0]
    for i := 1; i < n; i++ {
        if weights[i] < min {
            min = weights[i]
        }
    }
}
```

```

    }

    return min
}

func findMax(weights []float64, n int) float64 {
    max := weights[0]
    for i := 1; i < n; i++ {
        if weights[i] > max {
            max = weights[i]
        }
    }
    return max
}

func calculateAverage(weights []float64, n int) float64 {
    var total float64
    for i := 0; i < n; i++ {
        total += weights[i]
    }
    return total / float64(n)
}

func main() {
    var n int

    fmt.Print("Masukkan jumlah balita: ")

    fmt.Scan(&n)

    if n < 1 {

```

```
        fmt.Println("Jumlah balita harus lebih dari 0.")

        return
    }

    weights := make([]float64, n)

    fmt.Println("Masukkan berat balita (dalam kg):")

    for i := 0; i < n; i++ {

        fmt.Printf("Berat balita ke-%d: ", i+1)

        fmt.Scan(&weights[i])

        if weights[i] <= 0 {

            fmt.Println("Berat balita harus lebih dari 0.")

            i--

        }

    }

    minWeight := findMin(weights, n)

    maxWeight := findMax(weights, n)

    averageWeight := calculateAverage(weights, n)

    fmt.Println("\nHasil Analisis Berat Balita:")

    fmt.Printf("Berat terkecil: %.2f kg\n", minWeight)

    fmt.Printf("Berat terbesar: %.2f kg\n", maxWeight)

    fmt.Printf("Rata-rata berat: %.2f kg\n", averageWeight)

}
```


Screenshoot Output

```
Rata-rata Berat: 3.75 kg
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_Modul 11\Modul 11> go run unguided3.go
Masukkan jumlah balita: 4
Masukkan berat balita (dalam kg):
Berat balita ke-1: 2
Berat balita ke-2: 3
Berat balita ke-3: 4
Berat balita ke-4: 6

Hasil Analisis Berat Balita:
Berat terkecil: 2.00 kg
Berat terbesar: 6.00 kg
Rata-rata berat: 3.75 kg
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_Modul 11\Modul 11> |
```

Deskripsi Program

Program ini membantu mencatat berat balita (dalam kilogram) yang hadir di Posyandu. Data ini dianalisis untuk menemukan: Berat balita terkecil, Berat balita terbesar, Rata-rata berat balita.