

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

Modul 11

Pencarian Nilai Ekstrem Pada Himpunan Data



Disusun Oleh :

Bintang Putra Angkasa (2311102255)

Kelas: S1-IF-11-06

Dosen Pengampu :

Abednego Dwi Septiadi

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Pencarian nilai ekstrem pada himpunan data bertujuan untuk menemukan titik–titik dengan nilai maksimum atau minimum ,nilai ekstrem merujuk pada nilai maksimum (terbesar) dan minimum (terkecil) yang dapat dicari dengan mengurutkan data atau membandingkan setiap elemen.

II. Guided

GUIDED 1

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)

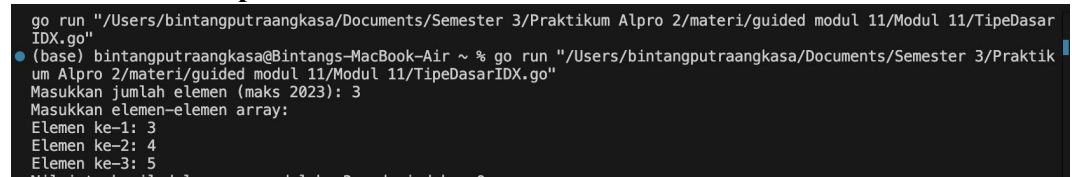
// Menampilkan nilai dan indeks terkecil

fmt.Println("Nilai terkecil dalam array adalah:",
tab[idxMin], "pada indeks:", idxMin)

}

```

Screenshoot Output



```

go run "/Users/bintangputraangkasa/Documents/Semester 3/Praktikum Alpro 2/materi/guided modul 11/Modul 11/TipeDasar
IDX.go"
(base) bintangputraangkasa@Bintang-MacBook-Air ~ % go run "/Users/bintangputraangkasa/Documents/Semester 3/Praktik
um Alpro 2/materi/guided modul 11/Modul 11/TipeDasarIDX.go"
Masukkan jumlah elemen (maks 2023): 3
Masukkan elemen-elemen array:
Elemen ke-1: 3
Elemen ke-2: 4
Elemen ke-3: 5
Nilai terkecil dalam array adalah 3 pada indeks 0

```

Deskripsi program

Program ini ditulis dalam bahasa Go untuk mencari nilai terkecil dalam array yang berisi elemen integer. Program mendeklarasikan array dengan panjang maksimum 2023 elemen dan menggunakan fungsi terkecil untuk mencari indeks nilai terkecil. Fungsi ini membandingkan setiap elemen dan mengembalikan indeks elemen terkecil. Di fungsi main, program meminta input jumlah elemen dan nilai-nilai array dari pengguna, kemudian memanggil fungsi terkecil untuk menemukan dan menampilkan nilai terkecil beserta indeksnya. Program juga memastikan jumlah elemen yang dimasukkan valid antara 1 hingga 2023.

GUIDED 2

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)

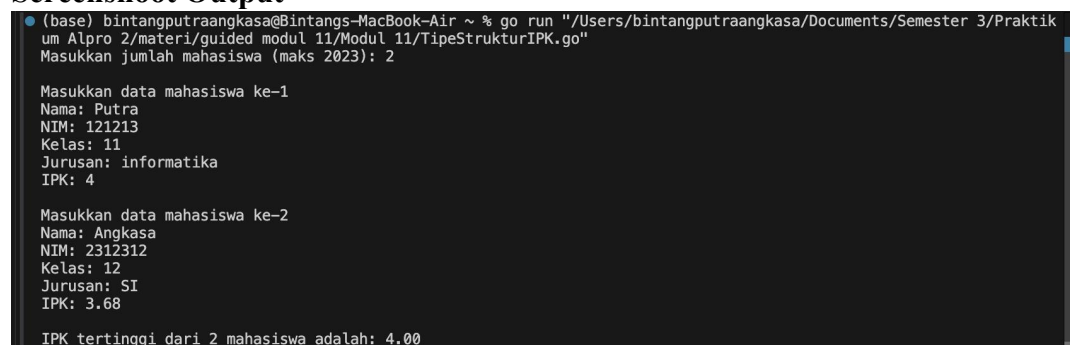
fmt.Printf("\nIPK tertinggi dari %d mahasiswa
```



```
adalah: %.2f\n", n, tertinggi)

}
```

Screenshot Output



```
(base) bintangputraangkasa@Bintang-MacBook-Air ~ % go run "/Users/bintangputraangkasa/Documents/Semester 3/Praktikum Alpro 2/materi/guided modul 11/Modul 11/TipeStrukturIPK.go"
Masukkan jumlah mahasiswa (maks 2023): 2

Masukkan data mahasiswa ke-1
Nama: Putra
NIM: 121213
Kelas: 11
Jurusan: informatika
IPK: 4

Masukkan data mahasiswa ke-2
Nama: Angkasa
NIM: 2312312
Kelas: 12
Jurusan: SI
IPK: 3.68

IPK tertinggi dari 2 mahasiswa adalah: 4.00
```

Deskripsi program

Program ini dibuat dengan bahasa Go untuk mengelola data mahasiswa dan mencari IPK tertinggi. Menggunakan struct mahasiswa yang menyimpan atribut nama, NIM, kelas, jurusan, dan IPK, serta array arrMhs untuk menampung data mahasiswa. Fungsi ipk mencari nilai IPK tertinggi dengan membandingkan setiap IPK mahasiswa. Program meminta input jumlah mahasiswa dan data masing-masing mahasiswa, lalu menampilkan IPK tertinggi setelah memproses data tersebut. Program ini memudahkan pengelolaan data mahasiswa dan pencarian IPK tertinggi.

III. UNGUIDED

Unguided 1

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.

Sourcecode

```
package main

import "fmt"

func main() {
    var N int

    // Input jumlah anak kelinci
    fmt.Print("Masukkan jumlah anak kelinci (N): ")
    fmt.Scan(&N)

    // Validasi jumlah N
    if N > 1000 {
        fmt.Println("Jumlah anak kelinci tidak boleh lebih dari 1000.")
        return
    }

    // Deklarasi array untuk menyimpan berat
    berat := make([]float64, N)

    // Input berat anak kelinci
    fmt.Printf("Masukkan %d berat anak kelinci (pisahkan dengan spasi):\n", N)
    for i := 0; i < N; i++ {
```

```

fmt.Scan(&berat[i])
}

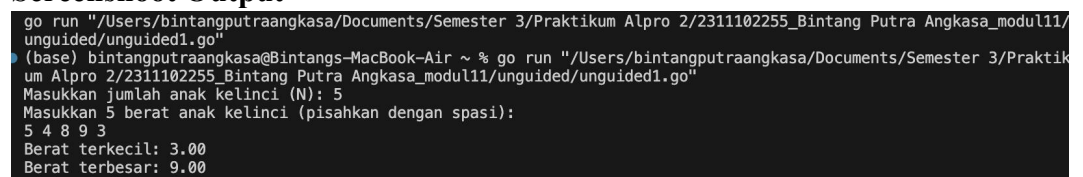
// Inisialisasi berat terkecil dan terbesar
beratTerkecil := berat[0]
beratTerbesar := berat[0]

// Pencarian nilai terkecil dan terbesar
for i := 1; i < N; i++ {
    if berat[i] < beratTerkecil {
        beratTerkecil = berat[i]
    }
    if berat[i] > beratTerbesar {
        beratTerbesar = berat[i]
    }
}

// Cetak hasil
fmt.Printf("Berat terkecil: %.2f\n",
    beratTerkecil)
fmt.Printf("Berat terbesar: %.2f\n",
    beratTerbesar)
}

```

Screenshoot Output



```

go run "/Users/bintangputraangkasa/Documents/Semester 3/Praktikum Alpro 2/2311102255_Bintang Putra Angkasa_modul11/unguided/unguided1.go"
(base) bintangputraangkasa@Bintangs-MacBook-Air ~ % go run "/Users/bintangputraangkasa/Documents/Semester 3/Praktikum Alpro 2/2311102255_Bintang Putra Angkasa_modul11/unguided/unguided1.go"
Masukkan jumlah anak kelinci (N): 5
Masukkan 5 berat anak kelinci (pisahkan dengan spasi):
5 4 8 9 3
Berat terkecil: 3.00
Berat terbesar: 9.00

```

Deskripsi Program

Program ini ditulis dalam bahasa Go untuk mengelola dan mencari berat terkecil serta terbesar dari sejumlah anak kelinci yang dimasukkan oleh pengguna. Program dimulai dengan meminta input jumlah anak kelinci (N) dan memvalidasi agar jumlah tersebut tidak melebihi 1000. Setelah itu, program meminta pengguna untuk memasukkan berat anak kelinci yang disimpan dalam array berat. Program kemudian menginisialisasi

nilai berat terkecil dan terbesar dengan berat anak kelinci pertama, lalu melakukan pencarian untuk nilai terkecil dan terbesar melalui iterasi array. Setelah pencarian selesai, program menampilkan berat terkecil dan terbesar yang ditemukan.

Program ini memberikan cara sederhana untuk memproses data berat anak kelinci dan menampilkan hasil pencarian nilai ekstrem dari data tersebut.

Unguided 2

Sourcecode

```
package main

import (
    "fmt"
)

func main() {
    var jumlahWadah, jumlahIkanPerWadah int
    fmt.Print("Masukkan jumlah wadah (x): ")
    fmt.Scan(&jumlahWadah)
    fmt.Print("Masukkan jumlah ikan per wadah (y): ")
    fmt.Scan(&jumlahIkanPerWadah)

    // Validasi input

    if jumlahWadah < 1 || jumlahWadah > 1000 ||
        jumlahIkanPerWadah < 1 {

        fmt.Println("Nilai jumlah wadah harus antara 1-1000 dan jumlah ikan per wadah harus lebih dari 0.")
        return
    }

    // Membuat array untuk menyimpan berat ikan
    beratIkan := make([]float64,
```

```

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

totalBeratPerWadah := make([]float64, jumlahWadah)
rataRataBeratPerWadah := make([]float64,
jumlahWadah)

// Menghitung total berat dan rata-rata berat ikan
per wadah
for i := 0; i < jumlahWadah; i++ {
totalBerat := 0.0
for j := 0; j < jumlahIkanPerWadah; j++ {
totalBerat += beratIkan[i*jumlahIkanPerWadah+j]
}
totalBeratPerWadah[i] = totalBerat
rataRataBeratPerWadah[i] = totalBerat /
float64(jumlahIkanPerWadah)
}

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

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

```

screenshoot

```
(base) bintangputraangkasa@Bintang-MacBook-Air ~ % go run "/var/folders/yp/9jpxhy1019251crsxmlb5j0000gn/T/tempCo
deRunnerFile.go"
Masukkan jumlah wadah (x): 2
Masukkan jumlah ikan per wadah (y): 2
Masukkan berat ikan satu per satu:
Berat ikan ke-1 (dalam kg): 3
Berat ikan ke-2 (dalam kg): 4
Berat ikan ke-3 (dalam kg): 5
Berat ikan ke-4 (dalam kg): 6

Total berat ikan di setiap wadah:
Wadah 1: 7.00 kg
Wadah 2: 11.00 kg

Rata-rata berat ikan di setiap wadah:
Wadah 1: 3.50 kg
Wadah 2: 5.50 kg
```

Deskripsi

Program ini menghitung total dan rata-rata berat ikan per wadah. Pengguna diminta untuk memasukkan jumlah wadah dan jumlah ikan per wadah, dengan validasi input. Setelah itu, program meminta berat ikan untuk setiap wadah dan menghitung total serta rata-rata berat ikan di setiap wadah. Hasilnya kemudian ditampilkan, yaitu total berat ikan dan rata-rata berat ikan di setiap wadah. Program ini bertujuan untuk memudahkan analisis distribusi berat ikan dalam wadah.

Unguided 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 BeratBalita [100]float64 // Array untuk
```

menyimpan berat balita

```
// Fungsi untuk menghitung berat balita minimum
dan maksimum
func hitungMinMax(berat BeratBalita, jumlahBalita
int, beratMin *float64, beratMax *float64) {
// Inisialisasi beratMin dan beratMax dengan berat
balita pertama
*beratMin = berat[0]
*beratMax = berat[0]

// Menemukan berat balita minimum dan maksimum
for i := 1; i < jumlahBalita; i++ {
if berat[i] < *beratMin {
*beratMin = berat[i]
}
if berat[i] > *beratMax {
*beratMax = berat[i]
}
}
}

// Fungsi untuk menghitung rata-rata berat balita
func hitungRataRata(berat BeratBalita,
jumlahBalita int) float64 {
var totalBerat float64 = 0.0
for i := 0; i < jumlahBalita; i++ {
totalBerat += berat[i]
}
return totalBerat / float64(jumlahBalita)
}

func main() {
var jumlahBalita int
var beratBalita BeratBalita
var beratMin, beratMax float64

// Input jumlah balita
fmt.Print("Masukkan jumlah balita: ")
fmt.Scan(&jumlahBalita)

// Validasi jumlah balita
if jumlahBalita < 1 || jumlahBalita > 100 {
```

```

fmt.Println("Jumlah balita harus antara 1 dan
100.")
return
}

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

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

// Menghitung rata-rata berat balita
rataRata := hitungRataRata(beratBalita,
jumlahBalita)

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

```

Screenshot program

```

go run "/Users/bintangputraangkasa/Documents/Semester 3/Praktikum Alpro 2/2311102255_Bintang Putra Angkasa_modul11/
unguided/unguided3.go"
(base) bintangputraangkasa@Bintang-MacBook-Air ~ % go run "/Users/bintangputraangkasa/Documents/Semester 3/Praktik
um Alpro 2/2311102255_Bintang Putra Angkasa_modul11/unguided/unguided3.go"
Masukkan jumlah balita: 2
Masukkan berat balita ke-1: 4.5
Masukkan berat balita ke-2: 5.5

Berat balita minimum: 4.50 kg
Berat balita maksimum: 5.50 kg
Rata-rata berat balita: 5.00 kg
(base) bintangputraangkasa@Bintang-MacBook-Air ~ %

```


Deskripsi

Program ini menghitung berat balita minimum, maksimum, dan rata-rata. Pengguna memasukkan jumlah balita (antara 1 hingga 100) dan berat setiap balita. Program kemudian menghitung berat balita minimum, maksimum, dan rata-rata menggunakan dua fungsi: `hitungMinMax` dan `hitungRataRata`. Hasilnya ditampilkan setelah perhitungan selesai. Program ini mempermudah analisis data berat balita.