

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL XII & XIII
PENGURUTAN DATA**



Disusun Oleh :

Andika Indra Prastawa / 2311102033

Kelas

IF- 11- 06

Dosen Pengampu :

Abednego Dwi Septiadi, S.Kom., M.Kom

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Salah satu fungsi pemrograman yang paling penting adalah pengurutan data, yang digunakan dalam pembuatan aplikasi berbasis Golang. Tugas pengurutan adalah untuk mengatur elemen dalam suatu koleksi data, seperti array, slice, atau list, disusun dalam urutan tertentu, baik secara menaik maupun turun.

Konsep Dasar Pengurutan Data: Pengurutan data dimaksudkan untuk membuat pencarian, pengolahan, dan analisis data lebih mudah. Metode pengurutan terbagi menjadi dua kategori utama. Pengurutan internal dilakukan di dalam memori komputer dan berguna untuk data kecil hingga sedang. Pengurutan eksternal dilakukan dengan memori eksternal dan berguna untuk dataset besar yang tidak dapat dimuat sepenuhnya ke dalam memori.

Salah satu fungsi pemrograman yang paling penting adalah pengurutan data, yang digunakan dalam pembuatan aplikasi berbasis Golang. Tugas pengurutan adalah untuk mengatur elemen dalam suatu koleksi data, seperti array, slice, atau list, disusun dalam urutan tertentu, baik secara menaik maupun turun.

Paket ini menyediakan beberapa fungsi utama:

- `sort.Ints(slice []int)`: Mengurutkan slice tipe int secara ascending.
- `sort.Strings(slice []string)`: Mengurutkan slice tipe string secara ascending.
- `sort.Float64s(slice []float64)`: Mengurutkan slice tipe float64 secara ascending.

II. GUIDED

1. Guided 1

Sourcecode

```
package main

import (
    "fmt"
)

// Fungsi untuk mengurutkan array menggunakan Selection Sort
func selectionSort(arr []int, n int) {
    for i := 0; i < n-1; i++ {
        idxMin := i
        for j := i + 1; j < n; j++ {
            // Cari elemen terkecil
            if arr[j] < arr[idxMin] {
                idxMin = j
            }
        }
        // Tukar elemen terkecil dengan elemen di posisi i
        arr[i], arr[idxMin] = arr[idxMin], arr[i]
    }
}

func main() {
    var n int
    fmt.Print("Masukkan jumlah daerah kerabat (n): ")
    fmt.Scan(&n)

    // Proses tiap daerah
    for daerah := 1; daerah <= n; daerah++ {
        var m int
        fmt.Printf("\nMasukkan jumlah nomor rumah kerabat untuk daerah %d: ", daerah)
        fmt.Scan(&m)

        // Membaca nomor rumah untuk daerah ini
        arr := make([]int, m)
        fmt.Printf("Masukkan %d nomor rumah kerabat: ", m)

        for i := 0; i < m; i++ {
            fmt.Scan(&arr[i])
        }

        // Urutkan array dari terkecil ke terbesar
        selectionSort(arr, m)

        // Tampilkan hasil
    }
}
```

```

        fmt.Printf("Nomor rumah terurut untuk daerah %d:
", daerah)
        for _, num := range arr {
            fmt.Printf("%d ", num)
        }
        fmt.Println()
    }
}

```

Screenshoot Output

```

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\LENOVO\Downloads\modul 12 praktikum alpro> go run "c:\Use
Masukkan jumlah daerah kerabat (n): 2

Masukkan jumlah nomor rumah kerabat untuk daerah 1: 2
Masukkan 2 nomor rumah kerabat: 1
2
Nomor rumah terurut untuk daerah 1: 1 2

Masukkan jumlah nomor rumah kerabat untuk daerah 2: 3
Masukkan 3 nomor rumah kerabat: 10
11
12
Nomor rumah terurut untuk daerah 2: 10 11 12
PS C:\Users\LENOVO\Downloads\modul 12 praktikum alpro>

```

Deskripsi Program

Program mengurutkan nomor rumah kerabat di beberapa daerah. Pengguna diminta memasukkan jumlah daerah kerabat terlebih dahulu, kemudian untuk setiap daerah, dimasukkan jumlah nomor rumah kerabat yang akan diurutkan beserta daftar nomor rumah tersebut. Program mengurutkan nomor rumah dalam urutan menaik menggunakan algoritma Selection Sort, di mana elemen terkecil dipindahkan ke posisi yang sesuai secara iteratif. Setelah pengurutan selesai, program menampilkan daftar nomor rumah yang telah diurutkan untuk setiap daerah.

2. Soal Studi Case

Sourcecode

```
package main

import (
    "fmt"
)

// Fungsi untuk mengurutkan array menggunakan Insertion Sort
func insertionSort(arr []int, n int) {
    for i := 1; i < n; i++ {
        key := arr[i]
        j := i - 1

        // Geser elemen yang lebih besar dari key ke
        kanan
        for j >= 0 && arr[j] > key {
            arr[j+1] = arr[j]
            j--
        }
        arr[j+1] = key
    }
}

// Fungsi untuk memeriksa apakah selisih elemen array
tetap
func isConstantDifference(arr []int, n int) (bool, int)
{
    if n < 2 {
        return true, 0
    }

    difference := arr[1] - arr[0]
    for i := 1; i < n-1; i++ {
        if arr[i+1]-arr[i] != difference {
            return false, 0
        }
    }
    return true, difference
}

func main() {
    var arr []int
    var num int

    // Input data hingga bilangan negatif ditemukan
    fmt.Println("Masukkan data integer (akhiri dengan
    bilangan negatif):")
    for {
        fmt.Scan(&num)
        if num < 0 {
            break
        }
    }
}
```

```

    }
    arr = append(arr, num)
}

n := len(arr)

// Urutkan array menggunakan Insertion Sort
insertionSort(arr, n)

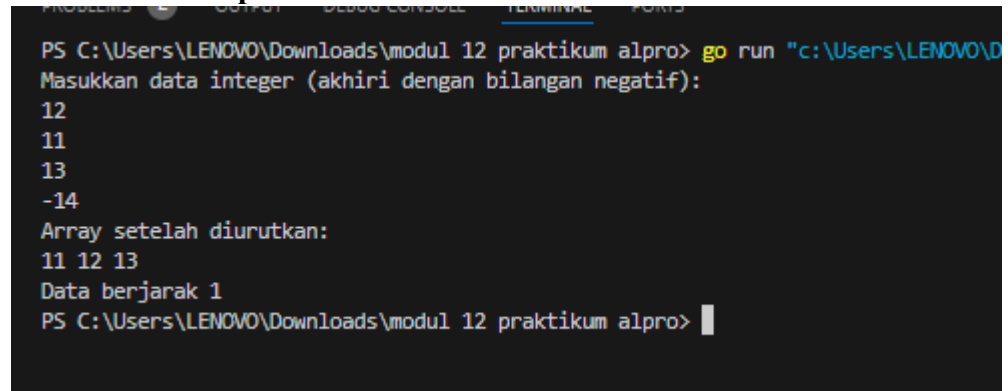
// Periksa apakah selisih elemen tetap
isConstant, difference := isConstantDifference(arr,
n)

// Tampilkan hasil pengurutan
fmt.Println("Array setelah diurutkan:")
for _, val := range arr {
    fmt.Printf("%d ", val)
}
fmt.Println()

// Tampilkan status jarak
if isConstant {
    fmt.Printf("Data berjarak %d\n", difference)
} else {
    fmt.Println("Data berjarak tidak tetap")
}
}

```

Screenshoot Output



```

PS C:\Users\LENOVO\Downloads\modul 12 praktikum alpro> go run "c:\Users\LENOVO\Downloads\modul 12 praktikum alpro.go"
Masukkan data integer (akhiri dengan bilangan negatif):
12
11
13
-14
Array setelah diurutkan:
11 12 13
Data berjarak 1
PS C:\Users\LENOVO\Downloads\modul 12 praktikum alpro>

```

Deskripsi Program

Program memeriksa konsistensi selisih antar elemen dalam array. Pengguna diminta untuk memasukkan serangkaian bilangan integer positif, diakhiri dengan bilangan negatif sebagai penanda akhir input. Setelah input diterima,

array tersebut diurutkan secara menaik menggunakan algoritma Insertion Sort. Program kemudian memeriksa apakah selisih antara elemen-elemen yang berdekatan dalam array yang telah diurutkan tetap konstan. Hasilnya ditampilkan berupa array yang sudah terurut dan informasi mengenai apakah data memiliki jarak tetap, termasuk nilai selisihnya jika konstan.

III. UNGUIDED

1. Unguided 1

Sourcecode

```
package main

import (
    "fmt"
    "sort"
)

func pisahganjilgenap_2311102033(arr []int) ([]int, []int) {
    var ganjil, genap []int
    for _, num := range arr {
        if num%2 == 1 {
            ganjil = append(ganjil, num)
        } else {
            genap = append(genap, num)
        }
    }
    return ganjil, genap
}

func urutnomerrumah_2311102033(ganjil []int, genap []int) []int {
    sort.Ints(ganjil)
    sort.Sort(sort.Reverse(sort.IntSlice(genap)))
    return append(ganjil, genap...)
}

func main() {
    var n int
    fmt.Print("Masukkan jumlah daerah kerabat (n): ")
    fmt.Scan(&n)

    // Garis pemisah
    fmt.Println("\n" + string('-'*40))

    for daerah := 1; daerah <= n; daerah++ {
        var m int
        fmt.Printf("Masukkan jumlah nomor rumah kerabat untuk daerah %d: ", daerah)
        fmt.Scan(&m)

        arr := make([]int, m)
        fmt.Printf("Masukkan %d nomor rumah kerabat: ", m)

        for i := 0; i < m; i++ {
            fmt.Scan(&arr[i])
        }
    }
}
```



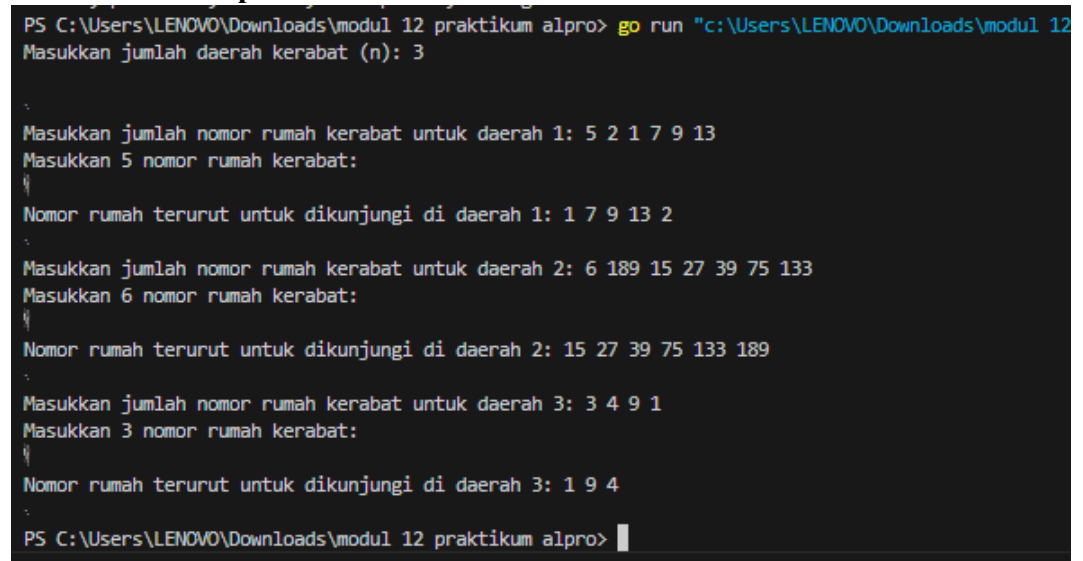
```

        ganjil, genap :=
pisahganjilgenap_2311102033(arr)
        hasil := urutnomerrumah_2311102033(ganjil,
genap)

        // Garis pemisah sebelum hasil
        fmt.Println("\n" + string('='*40))
        fmt.Printf("Nomor rumah terurut untuk dikunjungi
di daerah %d: ", daerah)
        for _, num := range hasil {
            fmt.Printf("%d ", num)
        }
        fmt.Println()
        // Garis pemisah setelah hasil
        fmt.Println(string('-' * 40))
    }
}

```

Screenshoot Output



```

PS C:\Users\LENOVO\Downloads\modul 12 praktikum alpro> go run "c:\Users\LENOVO\Downloads\modul 12
Masukkan jumlah daerah kerabat (n): 3
:
Masukkan jumlah nomor rumah kerabat untuk daerah 1: 5 2 1 7 9 13
Masukkan 5 nomor rumah kerabat:
?
Nomor rumah terurut untuk dikunjungi di daerah 1: 1 7 9 13 2
:
Masukkan jumlah nomor rumah kerabat untuk daerah 2: 6 189 15 27 39 75 133
Masukkan 6 nomor rumah kerabat:
?
Nomor rumah terurut untuk dikunjungi di daerah 2: 15 27 39 75 133 189
:
Masukkan jumlah nomor rumah kerabat untuk daerah 3: 3 4 9 1
Masukkan 3 nomor rumah kerabat:
?
Nomor rumah terurut untuk dikunjungi di daerah 3: 1 9 4
:
PS C:\Users\LENOVO\Downloads\modul 12 praktikum alpro>

```

Deskripsi Program

Program mengurutkan nomor rumah kerabat berdasarkan kategori ganjil dan genap di beberapa daerah. Pengguna diminta untuk memasukkan jumlah daerah, kemudian untuk setiap daerah, memasukkan jumlah dan daftar nomor rumah. Program memisahkan nomor rumah ke dalam dua kategori: ganjil dan genap. Nomor ganjil diurutkan secara naik, sedangkan nomor genap diurutkan secara turun. Hasil pengurutan dari kedua kategori

digabungkan dan ditampilkan untuk setiap daerah dengan format yang terstruktur dan dipisahkan oleh garis pemisah untuk memperjelas tampilan. Program ini dirancang untuk memberikan hasil yang sistematis dan mudah dipahami.

2. Unguided 2

Sourcecode

```
package main

import (
    "fmt"
    "sort"
)

func main() {
    var input_2311102033 int
    var data_2311102033 []int

    fmt.Println("Masukkan bilangan (Dan akhiri dengan -5313):")
    for {
        fmt.Scan(&input_2311102033)

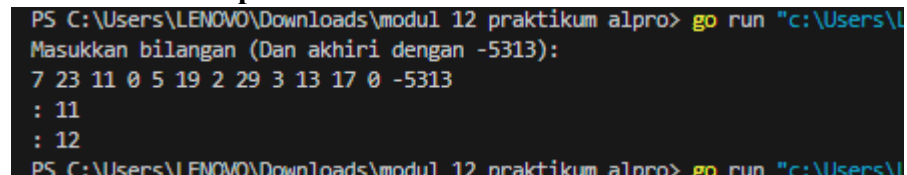
        if input_2311102033 == -5313 {
            break
        }

        if input_2311102033 == 0 {
            if len(data_2311102033) > 0 {
                sort.Ints(data_2311102033)
                median :=
menghitungMedian_2311102033(data_2311102033)
                fmt.Printf(" : %d\n", median)
            }
        } else {
            data_2311102033 = append(data_2311102033,
input_2311102033)
        }
    }

    func menghitungMedian_2311102033(data_2311102033 []int)
int {
    n := len(data_2311102033)
    if n%2 == 1 {
        return data_2311102033[n/2]
    }
}
```

```
        return (data_2311102033[(n/2)-1] +
data_2311102033[n/2]) / 2
    }
```

Screenshoot Output



```
PS C:\Users\LENOVO\Downloads\modul 12 praktikum alpro> go run "c:\Users\LENOVO\Downloads\modul 12 praktikum alpro.go"
Masukkan bilangan (Dan akhiri dengan -5313):
7 23 11 0 5 19 2 29 3 13 17 0 -5313
: 11
: 12
PS C:\Users\LENOVO\Downloads\modul 12 praktikum alpro> go run "c:\Users\LENOVO\Downloads\modul 12 praktikum alpro.go"
```

Deskripsi Program

Program membaca serangkaian bilangan dari pengguna hingga nilai sentinel -5313 dimasukkan, dan memiliki fitur untuk menghitung median data. Saat bilangan 0 dimasukkan, program mengurutkan daftar bilangan yang telah dimasukkan (kecuali angka nol dan sentinel), lalu menghitung dan mencetak median dari bilangan-bilangan tersebut. Median dihitung dengan mencari elemen tengah jika jumlah data ganjil, atau rata-rata dua elemen tengah jika jumlah data genap. Proses ini terus berulang hingga pengguna menghentikan masukan dengan memasukkan sentinel.

3. Unguided 3

Sourcecode

```
package main

import (
    "fmt"
)

const NMAX int = 7919

type Buku struct {
    id, judul, penulis, penerbit string
    eksemplar, tahun, rating      int
}

type DaftarBuku [NMAX]Buku
```

```

func DaftarBuku_2311102033(pustaka *DaftarBuku, N *int)
{
    fmt.Print("Masukkan jumlah buku: ")
    fmt.Scanln(N)

    if *N > NMAX {
        fmt.Printf("Jumlah buku tidak boleh lebih dari
%d.\n", NMAX)
        *N = 0
        return
    }

    for i := 0; i < *N; i++ {
        fmt.Printf("\nMasukkan data buku ke-%d:\n", i+1)
        fmt.Print("ID Buku: ")
        fmt.Scanln(&pustaka[i].id)
        fmt.Print("Judul Buku: ")
        fmt.Scanln(&pustaka[i].judul)
        fmt.Print("Penulis: ")
        fmt.Scanln(&pustaka[i].penulis)
        fmt.Print("Penerbit: ")
        fmt.Scanln(&pustaka[i].penerbit)
        fmt.Print("Jumlah Eksemplar: ")
        fmt.Scanln(&pustaka[i].eksemplar)
        fmt.Print("Tahun Terbit: ")
        fmt.Scanln(&pustaka[i].tahun)
        fmt.Print("Rating Buku: ")
        fmt.Scanln(&pustaka[i].rating)
    }
}

func CetakTerfavorit_2311102033(pustaka DaftarBuku, n
int) {
    if n == 0 {
        fmt.Println("Tidak ada buku yang terdaftar.")
        return
    }

    bukuFav := pustaka[0]
    for i := 1; i < n; i++ {
        if bukuFav.rating < pustaka[i].rating {
            bukuFav = pustaka[i]
        }
    }

    fmt.Printf("\nBuku terfavorit: %s oleh %s, penerbit
%s, tahun %d dengan rating %d\n",
        bukuFav.judul, bukuFav.penulis,
        bukuFav.penerbit, bukuFav.tahun, bukuFav.rating)
}

func UrutanBuku_2311102033(pustaka *DaftarBuku, n int) {
    for i := 1; i < n; i++ {

```

```

        temp := pustaka[i]
        j := i
        for j > 0 && pustaka[j-1].rating < temp.rating {
            pustaka[j] = pustaka[j-1]
            j--
        }
        pustaka[j] = temp
    }
}

func CetakTerbaru_2311102033(pustaka DaftarBuku, n int)
{
    fmt.Println("\nTop 5 Buku dengan Rating Tertinggi:")
    limit := n
    if n > 5 {
        limit = 5
    }
    for i := 0; i < limit; i++ {
        fmt.Printf("%d. %s (Rating: %d)\n", i+1,
pustaka[i].judul, pustaka[i].rating)
    }
}

func CariBuku_2311102033(pustaka DaftarBuku, n int, r
int) {
    if n == 0 {
        fmt.Println("Tidak ada buku yang terdaftar.")
        return
    }

    found := false
    for i := 0; i < n; i++ {
        if pustaka[i].rating == r {
            found = true
            fmt.Printf("\nBuku ditemukan: %s oleh %s,
penerbit %s, tahun %d, eksemplar %d, rating %d\n",
                pustaka[i].judul, pustaka[i].penulis,
pustaka[i].penerbit, pustaka[i].tahun,
pustaka[i].eksemplar, pustaka[i].rating)
            break
        }
    }
    if !found {
        fmt.Println("\nTidak ada buku dengan rating
tersebut.")
    }
}

func main() {
    var pustaka DaftarBuku
    var nPustaka int
    var rating int

```

```

    DaftarBuku_2311102033(&pustaka, &nPustaka)

    CetakTerfavorit_2311102033(pustaka, nPustaka)

    UrutanBuku_2311102033(&pustaka, nPustaka)
    CetakTerbaru_2311102033(pustaka, nPustaka)

    fmt.Print("\nMasukkan rating untuk mencari buku: ")
    fmt.Scanln(&rating)
    CariBuku_2311102033(pustaka, nPustaka, rating)
}

```

Screenshoot Output

```

PS C:\Users\LENOVO\Downloads\modul 12 praktikum alpro> go run "c:\Users\LENOVO\Downloads\modul 1
Masukkan jumlah buku: 3

Masukkan data buku ke-1:
ID Buku: 11
Judul Buku: dilan
Penulis: kaka
Penerbit: ptsukacita
Jumlah Eksemplar: 2
Tahun Terbit: 2019
Rating Buku: 8

Masukkan data buku ke-2:
ID Buku: 22
Judul Buku: milea
Penulis: yanto
Penerbit: gramed
Jumlah Eksemplar: 4
Tahun Terbit: 2018
Rating Buku: 7

Masukkan data buku ke-3:
ID Buku: 33
Judul Buku: kisahsekolah
Penulis: elsa
Penerbit: gramed
Jumlah Eksemplar: 3
Tahun Terbit: 2020
Rating Buku: 9

Buku terfavorit: kisahsekolah oleh elsa, penerbit gramed, tahun 2020 dengan rating 9

Top 5 Buku dengan Rating Tertinggi:
1. kisahsekolah (Rating: 9)
2. dilan (Rating: 8)
3. milea (Rating: 7)

Masukkan rating untuk mencari buku: 8

Buku ditemukan: dilan oleh kaka, penerbit ptsukacita, tahun 2019, eksemplar 2, rating 8
PS C:\Users\LENOVO\Downloads\modul 12 praktikum alpro>

```

Deskripsi Program

Program mengelola daftar buku yang memungkinkan pengguna untuk memasukkan data buku, seperti ID, judul, penulis, penerbit, jumlah eksemplar, tahun terbit, dan rating. Program ini menyediakan beberapa fitur utama: mencatat dan menyimpan daftar buku, menampilkan buku dengan rating tertinggi (terfavorit), mengurutkan buku berdasarkan rating secara menurun, mencetak lima buku dengan rating tertinggi, serta mencari buku berdasarkan rating tertentu. Dengan batas maksimum jumlah buku sebanyak 7919, program ini dirancang untuk memberikan informasi yang terorganisir dan mempermudah pengguna dalam menemukan buku sesuai kebutuhan mereka.