

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL XII**

**PENGURUTAN DATA**



**Disusun Oleh :**

**Arjun Ahmad Santoso / 2311102211**

**S1IF-11-06**

**Dosen Pengampu :**

**Abednego Dwi Septiadi**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## **I. DASAR TEORI**

### **4.1 Ide Pengurutan**

Pengurutan adalah proses yang sangat lazim dalam dunia pemrograman. Pengurutan dapat dilakukan secara ascending maupun descending terhadap nilai tertentu. Terdapat beberapa algoritma pengurutan, diantaranya adalah Selection Sort, Insertion Sort, Bubble Sort dan Quick Sort.

## II. GUIDED

1.

### Soal Guided

Soal Latihan Selection Sort No. 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

```

C:\Lecture\Semester_3\Praktikum_Algoritma_Pemrograman_2\modul_12>go run guided\1.go
Masukkan jumlah daerah kerabat (n): 4
-5111022114Masukkan jumlah nomor rumah kerabat untuk daerah 1: 3
23111022114Masukkan 3 nomor rumah kerabat: 9 6 3
Nomor rumah terurut untuk daerah 1: 3 6 9
F Guidedmodul
Modul12pdfMasukkan jumlah nomor rumah kerabat untuk daerah 2: 4
Masukkan 4 nomor rumah kerabat: 5 8 9 2
Nomor rumah terurut untuk daerah 2: 2 5 8 9
Masukkan jumlah nomor rumah kerabat untuk daerah 3: 2
Masukkan 2 nomor rumah kerabat: 3 2
Nomor rumah terurut untuk daerah 3: 2 3
Masukkan jumlah nomor rumah kerabat untuk daerah 4: 4
Masukkan 4 nomor rumah kerabat: 4 7 3 1
Nomor rumah terurut untuk daerah 4: 1 3 4 7
C:\Lecture\Semester_3\Praktikum_Algoritma_Pemrograman_2\modul_12>

```

## Deskripsi Program

Program di atas merupakan contoh program yang menerapkan algoritma Selection Sort. Algoritma Selection Sort digunakan untuk mengurutkan data nomor rumah dari yang terkecil hingga ke yang terbesar.

2.

### Soal Guided

Soal Latihan Insertion Sort No. 1

### 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**



### III. UNGUIDED

1.

#### Soal Unguided

Soal Latihan Selection Sort No.2

#### Sourcecode

```
package main

import (
    "fmt"
)

// Fungsi untuk mengurutkan array menggunakan Selection Sort
func selectionSortKunjungan(arr []int, n int) {
    for i := 0; i < n-1; i++ {
        idxEx := i
        minGanjil := -1
        for j := i + 1; j < n; j++ {
            // Cari elemen terkecil
            if (arr[idxEx] % 2) == 1 {
                if (arr[j] % 2) == 1 {
                    if arr[j] < arr[idxEx] {
                        idxEx = j
                    }
                }
            } else {
                if (arr[j] % 2) == 1 {
                    if (minGanjil == -1 || arr[j] < minGanjil) {
                        idxEx = j
                        minGanjil = arr[j]
                    }
                } else {
                    if arr[j] > arr[idxEx] {
                        idxEx = j
                    }
                }
            }
        }
        // Tukar elemen terkecil dengan elemen di posisi i
        arr[i], arr[idxEx] = arr[idxEx], 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)

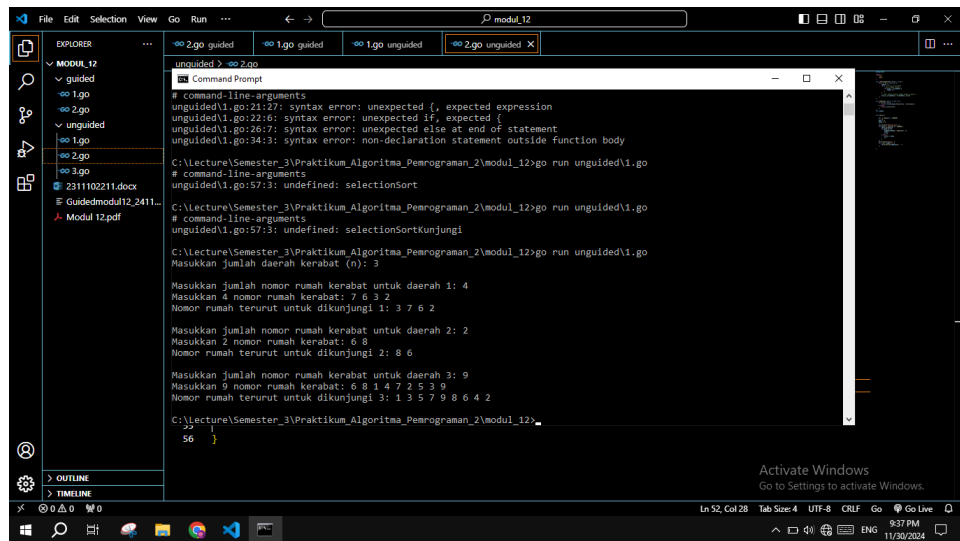
        // Membaca nomor rumah untuk daerah ini
        arr := make([]int, m)
        fmt.Printf("Masukkan %d nomor rumah kerabat: ", m)
        for i := 0; i < m; i++ {
            fmt.Scan(&arr[i])
        }

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

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

```

**Screenshot Output**



## Deskripsi Program

Program di atas memuat suatu algoritma yang digunakan untuk mengatur sebuah array bilangan, dengan konfigurasi sebagai berikut:

- Bilangan ganjil selalu muncul sebelum bilangan genap
- Sekumpulan bilangan ganjil terurut secara menaik
- Sekumpulan bilangan genap terurut secara menurun

Contoh masukan dan keluaran dapat dilihat pada gambar di atas.

2.

## Soal Unguided

Soal Latihan Selection Sort No. 3

## Sourcecode

```

package main

import (
    "fmt"
)

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 median(arr []int, n int) int {
    if (n % 2) == 0 {
        return int((arr[n/2 - 1] + arr[n/2])/2)
    } else {
        return arr[(n-1)/2]
    }
}

func main() {

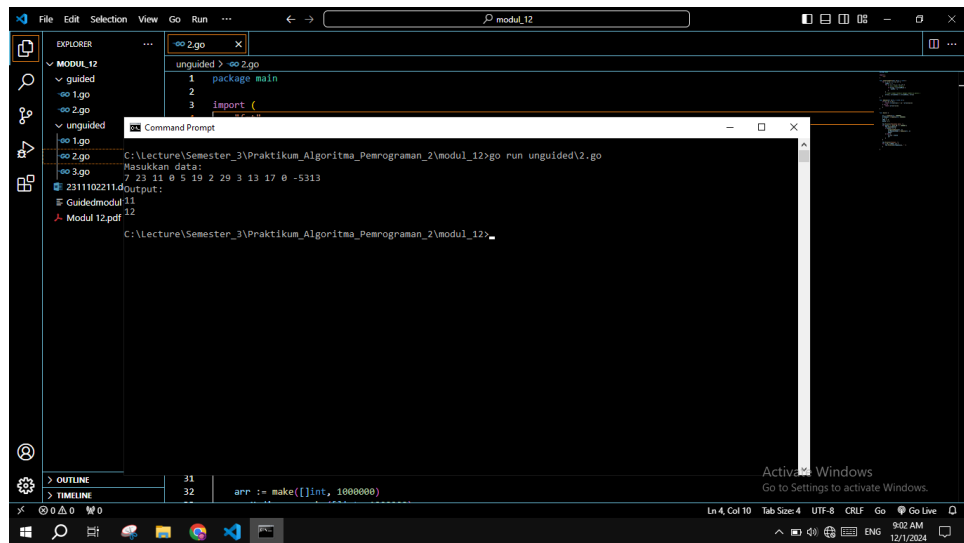
    arr := make([]int, 1000000)
    arrMedian := make([]int, 1000000)
    n := 0
    nMed := 0
    datum := 0

    fmt.Println("Masukkan data: ")
    for datum != -5313 && n < 1000000 {
        fmt.Scan(&datum)
        if datum == 0 {
            selectionSort(arr, n)
            arrMedian[nMed] = median(arr, n)
            nMed++
        } else {
            arr[n] = datum
            n++
        }
    }

    fmt.Println("Output: ")
    for i:=0; i<nMed; i++ {
        fmt.Println(arrMedian[i], " ")
    }
}

```

**Screenshoot Output**



### Deskripsi Program

Program di atas memuat algoritma yang berfungsi untuk mencetak median dari sekumpulan data yang muncul sebelum setiap angka 0 seperti pada contoh keluaran di atas. Program akan berakhir jika masukan berupa angka -5313.

3.

### Soal Unguided

Soal Latihan Insertion Sort No. 2

### Sourcecode

```
package main

import (
    "fmt"
    "math"
)

const nMax int = 7919

type Buku struct {
    id, judul, penulis, penerbit string
    eksemplar, tahun, rating int
}
type DaftarBuku [nMax]Buku

func DaftarkanBuku(daftar_buku *DaftarBuku, n int) {
    fmt.Println("Masukkan data buku di setiap baris (id judul
```

```

    penulis penerbit eksemplar tahun rating)")
    for i:=0; i<n; i++ {
        fmt.Scan(&daftar_buku[i].id, &daftar_buku[i].judul,
&daftar_buku[i].penulis, &daftar_buku[i].penerbit,
&daftar_buku[i].eksemplar, &daftar_buku[i].tahun,
&daftar_buku[i].rating)
    }
}
func CetakTerfavorit(daftar_buku DaftarBuku, n int) {
    var buku_terfavorit Buku = daftar_buku[0]
    // // Jika buku belum diurutkan maka uncomment bagian ini
    // for i:=1; i<n; i++ {
    //     if buku_terfavorit.rating < daftar_buku[i].rating {
    //         buku_terfavorit = daftar_buku[i]
    //     }
    // }
    fmt.Println("Buku terfavorit adalah buku dengan judul:
\\", buku_terfavorit.judul, "\\")
}
func UrutBukuBerdasarkanRating(daftar_buku *DaftarBuku, n int)
{
    for i := 0; i < n-1; i++ {
        idxMax := i
        for j := i + 1; j < n; j++ {
            if daftar_buku[j].rating >
daftar_buku[idxMax].rating {
                idxMax = j
            }
        }
        // Tukar elemen terkecil dengan elemen di posisi i
        daftar_buku[i], daftar_buku[idxMax] =
daftar_buku[idxMax], daftar_buku[i]
    }
}
func Cetak5Terbaru(daftar_buku DaftarBuku, n int) {
    fmt.Println("5 Buku dengan rating tertinggi: ")
    for i:=0; i<5; i++ {
        fmt.Println(i+1, ". ", daftar_buku[i].judul, ",
rating: ", daftar_buku[i].rating)
    }
    fmt.Print("\\n")
}
func CariBuku(daftar_buku DaftarBuku, n, rating int) {
    ditemukan := false
    top := n-1

```

```

        bottom := 0
        middle := int((top + bottom)/2)
        // Mengecek index pertama dan terakhir daftar_buku
        terlebih dahulu sebelum lanjut ke proses Binary Search
        if daftar_buku[bottom].rating == rating && rating != 0 {
            ditemukan = true
            fmt.Println(daftar_buku[bottom].judul)
            // Mulai menelusuri buku-buku yang berada di dekat
            buku daftar_buku[0] dan mencetak judulnya jika ratingnya sama
            for i:=bottom+1; i<n; i++ {
                if(daftar_buku[i].rating != rating) { break }
                fmt.Println(daftar_buku[i].judul)
            }
            return
        }
        if daftar_buku[top].rating == rating && rating != 0 {
            ditemukan = true
            fmt.Println(daftar_buku[top].judul)
            // Mulai menelusuri buku-buku yang berada di dekat
            buku daftar_buku[0] dan mencetak judulnya jika ratingnya sama
            for i:=top+1; i>=0; i-- {
                if(daftar_buku[i].rating != rating) { break }
                fmt.Println(daftar_buku[i].judul)
            }
            return
        }
        for middle > 0 && middle < n {
            if daftar_buku[middle].rating == rating && rating != 0
        {
            ditemukan = true
            fmt.Println(daftar_buku[middle].judul)
            // Mulai menelusuri buku-buku yang berada di dekat
            buku daftar_buku[middle] dan mencetak judulnya jika ratingnya
            sama
            for i:=middle-1; i>=0; i-- {
                if(daftar_buku[i].rating != rating) { break }
                fmt.Println(daftar_buku[i].judul)
            }
            for i:=middle+1; i<n; i++ {
                if(daftar_buku[i].rating != rating) { break }
                fmt.Println(daftar_buku[i].judul)
            }
            return
        } else if daftar_buku[middle].rating < rating {
            bottom = middle

```

```

        middle = int(math.Ceil(float64(top + bottom)/2))
    } else {
        top = middle
        middle = int(math.Floor(float64(top + bottom)/2))
    }
    fmt.Println(middle)
}
if ! ditemukan {
    fmt.Println("Buku dengan rating ", rating, " tidak
ditemukan")
}
}

func main() {

    var daftar_buku DaftarBuku
    var n int
    fmt.Print("Masukkan jumlah buku (n): ")
    fmt.Scan(&n)

    DaftarkanBuku(&daftar_buku, n)
    UrutBukuBerdasarkanRating(&daftar_buku, n)

    fmt.Print("\n")
    CetakTerfavorit(daftar_buku, n)
    fmt.Print("\n")
    Cetak5Terbaru(daftar_buku, n)

    var rating int
    fmt.Print("Masukkan rating buku yang dicari: ")
    fmt.Scan(&rating)
    CariBuku(daftar_buku, n, rating)

}

// Contoh input:
// 1 Buku_A Jhon Penerbit_A 200 2024 4
// 2 Buku_B Alex Penerbit_A 150 2022 3
// 3 Buku_C Jhon Penerbit_B 200 2020 4
// 4 Buku_D Steve Penerbit_A 150 2024 4
// 5 Buku_E Jhon Penerbit_A 300 2020 5
// 6 Buku_F Jhon Penerbit_B 400 2024 4
// 7 Buku_G Alex Penerbit_A 200 2024 3
// 8 Buku_H Jhon Penerbit_A 150 2022 3

```

## Screenshoot Output

```
File Edit Selection View Go Run ... mod_12
EXPLORER
  MODUL_12
    guided
    1.go
    2.go
    unguided
      3.go
      111
      112
      113
      114
      Command Prompt
        C:\Lecture\Semester_3\Praktikum_Algoritma_Pemrograman_2\modul_12>go run unguided\3.go
        Masukkan jumlah buku (n): 8
        Masukkan data buku di setiap baris (id judul penulis penerbit eksemplar tahun rating)
        2311102211d1 Buku_A Jhon Penerbit_A 200 2024 4
        Guidedmodul2 Buku_B Alex Penerbit_A 150 2022 3
        Modul 12.pdf Buku_C Jhon Penerbit_B 200 2020 4
        4 Buku_D Steve Penerbit_A 150 2024 4
        5 Buku_E Jhon Penerbit_A 300 2020 5
        6 Buku_F Jhon Penerbit_B 400 2024 4
        7 Buku_G Alex Penerbit_A 200 2024 3
        8 Buku_H Jhon Penerbit_A 150 2022 3

        Buku terfavorit adalah buku dengan judul: " Buku_E "

        5 Buku dengan rating tertinggi:
        1 - Buku_E , rating: 5
        2 - Buku_C , rating: 4
        3 - Buku_D , rating: 4
        4 - Buku_A , rating: 4
        5 - Buku_F , rating: 4

        Masukkan rating buku yang dicari: 4
        Buku_A
        Buku_D
        Buku_C
        Buku_F

        C:\Lecture\Semester_3\Praktikum_Algoritma_Pemrograman_2\modul_12>
```

## Deskripsi Program

Program di atas adalah program untuk mengelola daftar buku pada suatu perpustakaan. Program di atas menerapkan algoritma sorting untuk mengurutkan daftar buku berdasarkan ratingnya.