

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL XII
PENGURUTAN DATA**



Disusun Oleh :

Tri Marselinus Sitanggang / 2311102209

IF-11-06

Dosen Pengampu :

ABEDNEGO DWI SEPTIADI

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Selection Sort

Pengurutan secara seleksi adalah salah satu algoritma pengurutan yang bekerja dengan cara mencari nilai ekstrim (terkecil atau terbesar) dalam suatu daftar, kemudian menempatkannya pada posisi yang benar, lalu mengulangi proses ini untuk elemen-elemen yang tersisa hingga seluruh daftar terurut.

Insertion Sort

Pengurutan secara insertion ini idenya adalah menyisipkan suatu nilai pada posisi yang seharusnya. Berbeda dengan pengurutan seleksi, yang mana pada pengurutan ini tidak dilakukan pencarian nilai ekstrim terlebih dahulu, cukup memilih suatu nilai tertentu kemudian mencari posisinya secara sequential search.

II. GUIDED

1. Soal Latihan Selection Sort No. 1

Sourcecode

```
package main

import "fmt"

func selectionSort(arr []int) {
    n := len(arr)
    for i := 0; i < n-1; i++ {
        minIndex := i
        for j := i + 1; j < n; j++ {
            if arr[j] < arr[minIndex] {
                minIndex = j
            }
        }
        // Tukar elemen
        arr[i], arr[minIndex] = arr[minIndex], arr[i]
    }
}

func main() {
    var n int
    fmt.Scan(&n)

    for i := 0; i < n; i++ {
        var m int
        fmt.Scan(&m)

        rumah := make([]int, m)
        for j := 0; j < m; j++ {
            fmt.Scan(&rumah[j])
        }

        selectionSort(rumah)
        fmt.Println(rumah)
    }
}
```

Screenshoot Output

```
PS C:\ALPRO 2\PRAKTIKUM\MODUL 12> go run "c:\ALPRO 2\PRAKTIKUM\MODUL 12\Guided 1\Guided 1.go"
3
5 2 1 7 9 13
[1 2 7 9 13]
6 189 15 27 39 75 133
[15 27 39 75 133 189]
3 4 9 1
[1 4 9]
PS C:\ALPRO 2\PRAKTIKUM\MODUL 12> |
```

Deskripsi Program

Program diatas adalah program untuk mengurutkan array. Program ini meminta input dari pengguna: jumlah kasus uji **n**, dan untuk setiap kasus uji, jumlah elemen dalam array **m** diikuti oleh elemen-elemen array tersebut. Fungsi **selectionSort** mengurutkan array dengan cara menemukan elemen terkecil di setiap iterasi dan menukarnya dengan elemen pertama dari bagian array yang belum diurutkan. Setelah diurutkan, array tersebut dicetak ke layar.

2. Soal Latihan Insertion Sort No. 1

Sourcecode

```
package main

import "fmt"

func insertionSort(arr []int) {
    for i := 1; i < len(arr); i++ {
        key := arr[i]
        j := i - 1
        for j >= 0 && arr[j] > key {
            arr[j+1] = arr[j]
            j = j - 1
        }
        arr[j+1] = key
    }
}

func cekJarak(arr []int) string {
    jarak := arr[1] - arr[0]
    for i := 2; i < len(arr); i++ {
        if arr[i]-arr[i-1] != jarak {
            return "Data berjarak tidak
tetap"
        }
    }
    return fmt.Sprintf("Data berjarak %d", jarak)
}
```

```

func main() {
    var angka []int
    var input int

    fmt.Println("Masukkan angka (akhiri dengan angka
negatif):")
    for {
        fmt.Scan(&input)
        if input < 0 {
            break
        }
        angka = append(angka, input)
    }

    insertionSort(angka)
    fmt.Println("Array setelah diurutkan:", angka)
    fmt.Println(cekJarak(angka))
}

```

Screenshoot Output

```

PS C:\ALPRO 2\PRAKTIKUM\MODUL 12> go run "c:\ALPRO 2\PRAKTIKUM\MODUL 12\Guided 2\Guided 2.go"
Masukkan angka (akhiri dengan angka negatif):
31 13 25 43 1 7 19 37 -5
Array setelah diurutkan: [1 7 13 19 25 31 37 43]
Data berjarak 6
PS C:\ALPRO 2\PRAKTIKUM\MODUL 12>

```

Deskripsi Program

Program di atas adalah program untuk memasukkan serangkaian angka (diakhiri dengan angka negatif) dan kemudian mengurutkan angka-angka tersebut menggunakan algoritma **insertion sort**. Setelah array diurutkan, program akan memeriksa apakah selisih antara elemen-elemen dalam array tetap konstan. Jika jarak antara elemen-elemen tersebut tidak tetap, program akan mencetak "Data berjarak tidak tetap". Jika jaraknya tetap, program akan mencetak "Data berjarak X", dengan X adalah selisih tetap antara elemen-elemen array. Program ini mengandalkan dua fungsi utama: **insertionSort** untuk mengurutkan array dan **cekJarak** untuk memeriksa keseragaman jarak antar elemen.

III. UNGUIDED

1. Soal Latihan Selection Sort No. 2

Sourcecode

```
package main

import "fmt"

func main() {
    var n int
    fmt.Scan(&n)

    for i := 0; i < n; i++ {
        var m int
        fmt.Scan(&m)

        rumah := make([]int, m)
        for j := 0; j < m; j++ {
            fmt.Scan(&rumah[j])
        }

        ganjil := make([]int, 0)
        genap := make([]int, 0)
        for _, num := range rumah {
            if num%2 == 1 {
                ganjil = append(ganjil,
num)
            } else {
                genap = append(genap,
num)
            }
        }

        for i := 1; i < len(ganjil); i++ {
            key := ganjil[i]
            j := i - 1
            for j >= 0 && ganjil[j] > key {
                ganjil[j+1] = ganjil[j]
                j = j - 1
            }
            ganjil[j+1] = key
        }

        for i := 1; i < len(genap); i++ {
            key := genap[i]
            j := i - 1
            for j >= 0 && genap[j] < key {
                genap[j+1] = genap[j]
                j = j - 1
            }
            genap[j+1] = key
        }
    }
}
```

```

    }

    for _, num := range ganjil {
        fmt.Print(num, " ")
    }
    for i := len(genap) - 1; i >= 0; i-- {
        fmt.Print(genap[i], " ")
    }
    fmt.Println()
}
}

```

Screenshoot Output

```

PS C:\ALPRO 2\PRAKTIKUM\MODUL 12> go run "c:\ALPRO 2\PRAKTIKUM\MODUL 12\Unguided 1\Unguided 1.go"
3
5 2 1 7 9 13
1 7 9 13 2
6 189 15 27 39 75 133
15 27 39 75 133 189
3 4 9 1
1 9 4
PS C:\ALPRO 2\PRAKTIKUM\MODUL 12>

```

Deskripsi Program

Program di atas adalah program untuk mengurutkan inputan dengan beberapa kondisi. Setiap angka dipisahkan menjadi dua kelompok: angka ganjil dan angka genap. Angka ganjil diurutkan secara menaik menggunakan algoritma **insertion sort**, sementara angka genap diurutkan secara menurun. Setelah itu, program mencetak angka-angka ganjil yang telah diurutkan, diikuti dengan angka-angka genap yang juga telah diurutkan, dengan format angka-angka genap yang dicetak dari belakang (menurun). Program ini mengulang proses ini untuk setiap kasus uji yang diberikan oleh pengguna.

2. Soal Latihan Selection Sort No. 3

Sourcecode

```
package main

import "fmt"

func insertionSort(nums []int) {
    for i := 1; i < len(nums); i++ {
        key := nums[i]
        j := i - 1
        for j >= 0 && nums[j] > key {
            nums[j+1] = nums[j]
            j = j - 1
        }
        nums[j+1] = key
    }
}

func median(nums []int) float64 {
    insertionSort(nums)
    n := len(nums)
    if n%2 == 0 {
        return float64(nums[n/2-1]+nums[n/2]) / 2
    }
    return float64(nums[n/2])
}

func main() {
    var nums []int
    var num int
    for {
        fmt.Scan(&num)
        if num == 0 {
            if len(nums) > 0 {
                fmt.Println("Median:", median(nums))
            }
        } else if num == -5313 {
            break
        } else {
            nums = append(nums, num)
        }
    }
}
```


Screenshoot Output

```
PS C:\ALPRO 2\PRAKTIKUM\MODUL 12> go run "c:\ALPRO 2\PRAKTIKUM\MODUL 12\Unguided 2\Unguided 2.go"
7 23 11 0 5 19 2 29 3 13 17 0 -5313
Median: 11
Median: 12
PS C:\ALPRO 2\PRAKTIKUM\MODUL 12>
```

Deskripsi Program

Program di atas menerima input angka secara berulang dari pengguna, dan setiap kali angka 0 dimasukkan, program akan menghitung dan mencetak **median** dari angka-angka yang sudah dimasukkan sebelumnya. Angka-angka tersebut disortir terlebih dahulu menggunakan algoritma **insertion sort** sebelum median dihitung. Jika jumlah angka genap, median dihitung sebagai rata-rata dua angka tengah, sementara jika jumlah angka ganjil, median adalah angka di tengah. Program berhenti ketika angka -5313 dimasukkan. Input selain angka 0 atau -5313 akan ditambahkan ke dalam daftar angka yang sedang diproses.

3. Soal Latihan Insertion Sort No. 2

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 DaftarkanBuku(daftar_buku *DaftarBuku, n int) {
    fmt.Println("Masukkan data buku dengan format (id\njudul 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]
    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_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
            }
        }
        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 < n; 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
    for i := 0; i < n; i++ {
        if daftar_buku[i].rating == rating {
            ditemukan = true
            fmt.Print("Buku dengan rating ", rating, "
ditemukan: ")
            fmt.Println(daftar_buku[i].judul)
        }
    }

    if !ditemukan {
        fmt.Println("Buku dengan rating", rating, "tidak
ditemukan")
    }
}

func main() {
    var daftar_buku DaftarBuku

```

```

var n int
fmt.Print("Masukkan jumlah buku: ")
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)
}

```

Screenshoot Output

```

PS C:\ALPRO 2\PRAKTIKUM\MODUL 12> go run "c:\ALPRO 2\PRAKTIKUM\MODUL 12\Unguided 3\Unguided 3.go"
Masukkan jumlah buku: 10
Masukkan data buku dengan format (id judul penulis penerbit eksemplar tahun rating)
1 BookA AuthorA PublisherA 10 2015 5
2 BookB AuthorB PublisherB 8 2018 4
3 BookC AuthorC PublisherC 5 2019 5
4 BookD AuthorD PublisherD 10 2016 3
5 BookE AuthorE PublisherE 12 2018 4
6 BookF AuthorF PublisherF 7 2019 3
7 BookG AuthorG PublisherG 5 2017 5
8 BookH AuthorH PublisherH 6 2014 4
9 BookI AuthorI PublisherI 9 2016 5
10 BookJ AuthorJ PublisherJ 8 2020 4

Buku terfavorit adalah:" BookA "

5 Buku dengan rating tertinggi:
1 . BookA , rating: 5
2 . BookC , rating: 5
3 . BookG , rating: 5
4 . BookI , rating: 5
5 . BookE , rating: 4

Masukkan rating buku yang dicari: 3
Buku dengan rating 3 ditemukan: BookD
Buku dengan rating 3 ditemukan: BookF
PS C:\ALPRO 2\PRAKTIKUM\MODUL 12>

```

Deskripsi Program

Program di atas adalah program untuk mengelola data buku menggunakan struktur data Buku, yang mencakup informasi seperti ID, judul, penulis, penerbit, jumlah eksemplar, tahun terbit, dan rating. Program ini memungkinkan pengguna untuk mendaftarkan buku-buku, mengurutkan buku berdasarkan rating, dan kemudian menampilkan buku dengan rating tertinggi. Fungsi **CetakTerfavorit** mencetak buku dengan rating tertinggi, sementara **Cetak5Terbaru** menampilkan lima buku dengan rating tertinggi. Program juga memungkinkan pencarian buku berdasarkan rating melalui fungsi **CariBuku**. Semua buku diurutkan menggunakan algoritma **selection sort** berdasarkan rating.