

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 12
PENGURUTAN DATA**



Disusun Oleh :

Yoga Hogantara / 2311102153

IF-11-05

Dosen Pengampu :

Arif Amrulloh, S.Kom., M.Kom

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Algoritma sorting merupakan suatu algoritma untuk meletakkan Kumpulan Kumpulan data kedalam urutan tertentu, yang dibagi menjadi algoritma sorting ascending dan descending.

1. Insertion sort

Insertion sort mengurutkan suatu elemen data dengan cara dicek satu satu mulai dari yang kedua hingga terakhir

2. Selection sort

Selection sort mengurutkan suatu elemen data dengan cara membandingkan elemen data sekarang dengan elemen data berikutnya hingga akhir, jika ditemukannya elemen data yang paling kecil maka akan ditukar dengan elemen data sekarang.

II. GUIDED

1. 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 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 terdekat 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)

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

Screenshot Output

```
masukkan jumlah daerah kerabat (n): 2

Masukkan jumlah nomor rumah kerabat terdekat untuk daerah 1: 3
Masukkan 3 nomor rumah kerabat: 1 3 5
Nomor rumah terurut untuk daerah 1: 1 3 5

Masukkan jumlah nomor rumah kerabat terdekat untuk daerah 2: 4
Masukkan 4 nomor rumah kerabat: 2 1 4 3
Nomor rumah terurut untuk daerah 2: 1 2 3 4
PS D:\ZIP\vscode\savecode\New folder> |
```

Deskripsi Program

Program tersebut berfungsi untuk mengurutkan nomor rumah kerabat berdasarkan daerah yang diberikan. Inputan pada program tersebut di inputkan oleh user yang berfungsi untuk jumlah daerah, dan jumlah no rumah kerabat pada daerah yang nantinya akan diurutkan . Program tersebut menggunakan algoritma selection sort untuk mengurutkan no rumah kerabat. Program tersebut mengurutkan no dari terkecil hingga terbesar.

2. Sourcecode

```
package main
import "fmt"

func insertionsort(arr []int, n int){
    for i := 1; i < n; i++ {
        key := arr[i]
        j:= i-1

        //GESER ELEMENT YANG LEBIH BESAR DARI KEY KE
        KANAN
        for j >= 0 && arr[j] > key {
            arr[j+1]=arr[j]
            j--
        }
        arr[j+1]=key
    }
}

func isconstantdifference(arr []int,n int)(bool,
int){
    if n <2{
        return true , 0
    }

    difference := arr[1] - arr [0]
    for i := 1; i < n; 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 ILANGAN 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 SETIAP ELEMENT TETAP
        isConstan,difference :=
        isconstantdifference(arr, n)

        //TEMPILKAN HASIL PENGURUTAN
        fmt.Println("ARRAY SETELAH DIURUTKAN")
        for _,val := range arr{
            fmt.Printf("%d",val)
        }
        fmt.Println()
        //TAMPILKAN STATUS JARAK

        if isConstan{
            fmt.Printf("DATA BERJARAK %d\n",difference)
        }else {
            fmt.Println("DATA BERJARAK TIDAK TETAP")
        }
    }
}

```

Screenshoot Output

```

MASUKKAN DATA INTEGER (AKHIRI DENGAN BILANGAN NEGATIF):
2 1 3 4 1 2 -1
ARRAY SETELAH DIURUTKAN
112234
DATA BERJARAK TIDAK TETAP
PS D:\ZIP\vscode\savecode\New folder> 

```

Deskripsi Program

Program tersebut berfungsi untuk mengurutkan nilai integer dan jarak dari setiap nilai dari inputan yang di inputkan oleh user apakah berjarak tetap atau tidak. Program tersebut menggunakan algoritma insertion sort untuk mengurutkan nilai integernya yang di akhiri dengan memasukan bilangan

negatif. Pada program tersebut dalam mengurutkan nilai nya diurutkan dari nilai terkecil ke terbesar.

III. UNGUIDED

1. Sourcecode

```
package main
import "fmt"

func selectionSortmembesar(arr []int) {
    n := len(arr)
    for i := 0; i < n-1; i++ {
        minIdx := i
        for j := i + 1; j < n; j++ {
            if arr[j] < arr[minIdx] {
                minIdx = j
            }
        }
        arr[i], arr[minIdx] = arr[minIdx], arr[i]
    }
}

func selectionSortmengecil(arr []int) {
    n := len(arr)
    for i := 0; i < n-1; i++ {
        maxIdx := i
        for j := i + 1; j < n; j++ {
            if arr[j] > arr[maxIdx] {
                maxIdx = j
            }
        }
        arr[i], arr[maxIdx] = arr[maxIdx], arr[i]
    }
}

func main() {
    var n int
    fmt.Print("INPUT JUMLAH BARIS: ")
    fmt.Scanln(&n)

    inputs := make([][]int, n)
    for i := 0; i < n; i++ {
        var m int
        fmt.Printf("INPUT JUMLAH ANGKA PADA BARIS KE-%d: ", i+1)
        fmt.Scanln(&m)

        row := make([]int, m)
        fmt.Printf("INPUT ANGKA BARIS KE-%d : ", i+1)
        for j := 0; j < m; j++ {
            fmt.Scan(&row[j])
        }
    }
}
```

```

        inputs[i] = row
    }
    for _, input := range inputs {
        var ganjil []int
        var genap []int
        for _, num := range input {
            if num%2 == 0 {
                genap = append(genap, num)
            } else {
                ganjil = append(ganjil, num)
            }
        }
        selectionSortmembesar(ganjil)
        selectionSortmengecil(genap)
        output := append(ganjil, genap...)
        for _, num := range output {
            fmt.Printf("%d ", num)
        }
        fmt.Println()
    }
}

```

Screenshoot Output

```

INPUT JUMLAH BARIS: 3
INPUT JUMLAH ANGKA PADA BARIS KE-1: 5
INPUT ANGKA BARIS KE-1 : 2 1 7 9 13
INPUT JUMLAH ANGKA PADA BARIS KE-2: 6
INPUT ANGKA BARIS KE-2 : 189 15 27 39 75 133
INPUT JUMLAH ANGKA PADA BARIS KE-3: 3
INPUT ANGKA BARIS KE-3 : 4 9 1
1 7 9 13 2
15 27 39 75 133 189
1 9 4

```

Deskripsi Program

Program di atas berfungsi untuk mengurutkan angka dalam beberapa baris yang di tentukan user. Pada Program diatas menggunakan algoritma selection sort untuk mengurukan angka yang di inputkan dengan hasil output untuk angka ganjil diurutkan dalam urutan menaik (ascending) dan untuk angka genap diurutkan dalam urutan angka menurun (descending). Pada program tersebut terdapat percabangan if else untuk mengetahui angka ganjil dan genap nya, sehingga pada outputan akan sesuai dengan persyaratan.

2. Source code

```
package main
import "fmt"

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

func median(arr []int) int {
    if len(arr)%2 == 1 {
        return arr[len(arr)/2]
    }
    return (arr[len(arr)/2-1] + arr[len(arr)/2]) /
2
}

func main() {
    var n int
    var data []int

    fmt.Println("INPUT DATA END -5313:")

    for {
        fmt.Scan(&n)
        if n == -5313 {
            break
        }
        if n == 0 {
            if len(data) > 0 {
                selectionsort(data)
                fmt.Println(median(data))
                data = []int{}
            }
        } else {
            data = append(data, n)
        }
    }
}
```


Screenshoot Output

```
INPUT DATA END -5313:
7 23 11 0 5 19 2 29 3 13 17 0 -5313
11
13
PS D:\ZIP\vscode\savecode\New folder>
```

Deskripsi Program

Pada program tersebut berfungsi untuk mencari median dari suatu data yang di inputkan user. Pada program inputan data akan diurutkan terlebih dahulu menggunakan algoritma selection sort. Dan menggunakan kondisi if else untuk kondisi dimana jika pada inputan memasukan 0 maka data akan dirutkan menggunakan selection sort dan dihitung median nya. Program akan berjalan hingga menerima inputan -5313 dimasukkan untuk menghentikan program.

3. Source code

```
package main
import "fmt"

const Max = 7919
type Buku struct {
    id        int
    judul     string
    penulis   string
    penerbit  string
    eksemplar int
    tahun     int
    rating    int
}
type DaftarBuku struct {
    Pustaka []Buku
    nPustaka int
}

func DaftarkanBuku(pustaka *DaftarBuku, buku Buku) {
    if pustaka.nPustaka < Max {
        pustaka.Pustaka = append(pustaka.Pustaka, buku)
        pustaka.nPustaka++
    }
}

func CetakFavorit(pustaka DaftarBuku) {
    fmt.Println("Daftar Buku:")
    for _, buku := range pustaka.Pustaka {
        fmt.Printf("ID: %d, Judul: %s, Penulis: %s, Penerbit: %s, Rating: %d\n",
            buku.id, buku.judul, buku.penulis,
            buku.penerbit, buku.rating)
    }
}
```

```

func UrutBuku(pustaka *DaftarBuku) {
    for i := 1; i < pustaka.nPustaka; i++ {
        key := pustaka.Pustaka[i]
        j := i - 1
        for j >= 0 && pustaka.Pustaka[j].rating <
key.rating {
            pustaka.Pustaka[j+1] = pustaka.Pustaka[j]
            j--
        }
        pustaka.Pustaka[j+1] = key
    }
}

func Cetak5Tertinggi(pustaka DaftarBuku) {
    fmt.Println("5 Buku dengan Rating Tertinggi:")
    for i := 0; i < 5 && i < pustaka.nPustaka; i++ {
        buku := pustaka.Pustaka[i]
        fmt.Printf("Judul: %s, Penulis: %s, Rating:
%d\n", buku.judul, buku.penulis, buku.rating)
    }
}

func CariBuku(pustaka DaftarBuku, rating int) {
    fmt.Printf("Buku dengan Rating %d:\n", rating)
    found := false
    for _, buku := range pustaka.Pustaka {
        if buku.rating == rating {
            fmt.Printf("Judul: %s, Penulis: %s, Penerbit:
%s, Tahun: %d\n",
                buku.judul, buku.penulis, buku.penerbit,
buku.tahun)
            found = true
        }
    }
    if !found {
        fmt.Println("Tidak ada buku dengan rating
tersebut.")
    }
}

func main() {
    var pustaka DaftarBuku
    var n int
    fmt.Print("Masukkan jumlah buku: ")
    fmt.Scanln(&n)

    for i := 0; i < n; i++ {
        var buku Buku
        fmt.Printf("Masukkan data buku ke-%d (ID Judul
Penulis Penerbit Eksemplar Tahun Rating):\n", i+1)
        fmt.Scan(&buku.id, &buku.judul, &buku.penulis,
&buku.penerbit, &buku.eksemplar, &buku.tahun,
&buku.rating)
        DaftarkanBuku(&pustaka, buku)
    }
}

```

```
CetakFavorit(pustaka)
UrutBuku(&pustaka)
Cetak5Tertinggi(pustaka)
var rating int
fmt.Print("Masukkan rating buku yang ingin dicari: ")
fmt.Scanln(&rating)
CariBuku(pustaka, rating)
}
```

Screenshoot Output

```
Masukkan jumlah buku: 3
Masukkan data buku ke-1 (ID Judul Penulis Penerbit Eksemplar Tahun Rating):
1 pejantantangguh yhota gramde 5 2025 9
Masukkan data buku ke-2 (ID Judul Penulis Penerbit Eksemplar Tahun Rating):
2 minimalmandi yht gramde 7 2021 7
Masukkan data buku ke-3 (ID Judul Penulis Penerbit Eksemplar Tahun Rating):
3 gigikuning tyo gramde 5 2022 8
Daftar Buku:
ID: 1, Judul: pejantantangguh, Penulis: yhota, Penerbit: gramde, Rating: 9
ID: 2, Judul: minimalmandi, Penulis: yht, Penerbit: gramde, Rating: 7
ID: 3, Judul: gigikuning, Penulis: tyo, Penerbit: gramde, Rating: 8
5 Buku dengan Rating Tertinggi:
Judul: pejantantangguh, Penulis: yhota, Rating: 9
Judul: gigikuning, Penulis: tyo, Rating: 8
Judul: minimalmandi, Penulis: yht, Rating: 7
Masukkan rating buku yang ingin dicari: 9
Buku dengan Rating 9:
Judul: pejantantangguh, Penulis: yhota, Penerbit: gramde, Tahun: 2025
```

Deskripsi Program

Program tersebut berfungsi untuk memasukan data buku, mengurutkan data buku berdasar rating, dengan maksimal jumlah buku 7919. Pada program terdapat struct buku yang berisikan id , judul, penulis, penerbit, eksemplar, tahu, rating buku, dan daftar buku yang berisikan pustaka dan npustaka. Program tersebut menggunakan beberapa fungsi yang berfungsi untuk mencari buku, mencetak 5 buku tertinggi, mengurutkan buku, mencetak buku favorit, dan mendaftarkan buku.