

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
MODUL 12 & 13
PENGURUTAN DATA**



Oleh:

NAMA : AHMAD TITANA NANDA PRAMUDYA

NIM : 2311102042

KELAS : IF 11 02

**S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2024**

I. DASAR TEORI

Algoritma Selection Sort

Pengurutan secara seleksi Ini idenya adalah mencari nilai ekstrim pada sekumpulan data, kemudian meletakkan pada posisi yang seharusnya. Pada penjelasan berikut ini data akan diurut membesar (ascending), dan data dengan Indeks kecil ada di "kiri" dan indeks besar ada di "kanan".

- 1) Cari nilai terkecil di dalam rentang data tersisa
- 2) Pindahkan/tukar tempat dengan data yang berada pada posisi paling kiri pada rentang data tersisa tersebut.
- 3) Ulangi proses ini sampai tersisa hanya satu data saja.

Algoritma ini dikenal juga dengan nama Selection Sort, yang mana pada algoritma ini melibatkan dua proses yaitu pencarian indeks nilai ekstrim dan proses pertukaran dua nilai atau swap.

Algoritma 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. Pada penjelasan berikut ini data akan diurut mengecil (descending), dan data dengan Indeks kecil ada di "kiri" dan indeks besar ada di "kanan".

- 1) Untuk satu data yang belum terurut dan sejumlah data yang sudah diurutkan: Geser data yang sudah terurut tersebut (ke kanan), sehingga ada satu ruang kosong untuk memasukkan data yang belum terurut ke dalam kam University keterurutan. dalam data yang sudah terurut dan tetap menjaga.
- 2) Ulangi proses tersebut untuk setiap data yang belum terurut terhadap rangkaian data yang sudah terurut.

Algoritma ini dikenal juga dengan nama Insertion Sort, yang mana pada algoritma ini melibatkan dua proses yaitu pencarian sekuensial dan penyisipan.

II. GUIDED

No 1.

Source code:

```
package main

import "fmt"

// Fungsi untuk mengurutkan array menggunakan selection sort
func selectionSort(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] { // Cari elemen terbesar
                maxIdx = j
            }
        }
        arr[i], arr[maxIdx] = arr[maxIdx], arr[i] // Tukar elemen
    }
}

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

    if n <= 0 || n >= 1000 {
        fmt.Println("n harus lebih besar dari 0 dan kurang dari 1000.")
        return
    }

    for i := 0; i < n; i++ {
        var m int
        fmt.Printf("Masukkan jumlah rumah kerabat untuk daerah ke-%d: ", i+1)
        fmt.Scan(&m)

        if m <= 0 || m >= 1000000 {
            fmt.Println("m harus lebih besar dari 0 dan kurang dari 1000000.")
            return
        }

        // Masukkan nomor rumah
        houses := make([]int, m)
        fmt.Printf("Masukkan nomor rumah kerabat untuk daerah ke-%d: ", i+1)
```

```

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

    // Urutkan dengan selection sort
    selectionSort(houses)

    // Cetak hasil
    fmt.Printf("Hasil urutan rumah untuk daerah ke-%d: ", i+1)
    for _, house := range houses {
        fmt.Printf("%d ", house)
    }
    fmt.Println()
}
}

```

Output :

```

PS D:\titan\titan 2\PRAKTIKUM ALPRO\2311102042_Ahmad Titana Nanda Pramudya_Modul11\guided> go run Guided1.go
malformed import path "Guided1.go": invalid char ','
PS D:\titan\titan 2\PRAKTIKUM ALPRO\2311102042_Ahmad Titana Nanda Pramudya_Modul11\guided> go run Guided1.go
Masukkan jumlah daerah (n): 3
Masukkan jumlah rumah kerabat untuk daerah ke-1: 5 2 1 7 9 13
Masukkan nomor rumah kerabat untuk daerah ke-1: Hasil urutan rumah untuk daerah ke-1: 13 9 7 2 1
Masukkan jumlah rumah kerabat untuk daerah ke-2: 7
Masukkan nomor rumah kerabat untuk daerah ke-2: 6 189 15 27 39 75 133
Hasil urutan rumah untuk daerah ke-2: 189 133 75 39 27 15 6
Masukkan jumlah rumah kerabat untuk daerah ke-3: 4
Masukkan nomor rumah kerabat untuk daerah ke-3: 2 4 9 1
Hasil urutan rumah untuk daerah ke-3: 9 4 2 1
PS D:\titan\titan 2\PRAKTIKUM ALPRO\2311102042_Ahmad Titana Nanda Pramudya_Modul11\guided>

```

Penjelasan :

Program ini membaca data input dari beberapa daerah dan mengurutkan nomor rumah kerabat dalam urutan menurun (descending order) menggunakan algoritma selection sort.

NO.2

Sourcode :

```
package main

import (
    "fmt"
    "math"
)

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

        for j >= 0 && arr[j] > key {
            arr[j+1] = arr[j]
            j--
        }
        arr[j+1] = key
    }
}

func isDataConsistentlySpaced(arr []int) (bool, int) {
    if len(arr) < 2 {
        return true, 0 // Array dengan kurang dari 2 elemen dianggap berjarak
        tetap
    }
    diff := int(math.Abs(float64(arr[1] - arr[0])))

    for i := 1; i < len(arr)-1; i++ {
        currentDiff := int(math.Abs(float64(arr[i+1] - arr[i])))
        if currentDiff != diff {
            return false, 0 // Jika ada selisih yang berbeda, tidak berjarak
            tetap
        }
    }

    return true, diff
}

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

    fmt.Println("Masukkan data (akhiri dengan bilangan negatif):")
    for {
        fmt.Scan(&input)
        if input < 0 {
            break
        }
    }
}
```

```
        }
        data = append(data, input)
    }
    insertionSort(data)

    isConsistent, diff := isDataConsistentlySpaced(data)

    fmt.Println("Hasil pengurutan:", data)
    if isConsistent {
        fmt.Printf("Data berjarak %d\n", diff)
    } else {
        fmt.Println("Data berjarak tidak tetap")
    }
}
```

Output :

```
PS D:\titan\titan 2\PRAKTIKUM ALPRO\2311102042_Ahmad Titana Nanda Pramudya_Modul11\guided> go run Guided2.go
Masukkan data (akhiri dengan bilangan negatif):
31 13 25 43 1 7 19 37 -5
Hasil pengurutan: [1 7 13 19 25 31 37 43]
Data berjarak 6
PS D:\titan\titan 2\PRAKTIKUM ALPRO\2311102042_Ahmad Titana Nanda Pramudya_Modul11\guided>
```

Penjelasan :

Program di atas melakukan dua tugas Yaitu Mengurutkan array menggunakan Insertion Sort Dan Memeriksa apakah elemen array berjarak tetap Program menghitung selisih antara elemen berturut-turut dalam array terurut. Jika selisih semua elemen konsisten, maka array dianggap berjarak tetap dan nilai selisihnya ditampilkan Jika tidak konsisten, program menginformasikan bahwa array tidak berjarak tetap.

III. UNGUIDE NO.1

Source code :

```
package main

import (
    "fmt"
    "sort"
)

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

    if n <= 0 || n >= 1000 {
        fmt.Println("n harus lebih besar dari 0 dan kurang dari 1000.")
        return
    }

    for i := 0; i < n; i++ {
        var m int
        fmt.Printf("Masukkan jumlah rumah kerabat untuk daerah ke-%d: ",
i+1)
        fmt.Scan(&m)

        if m <= 0 || m >= 1000000 {
            fmt.Println("m harus lebih besar dari 0 dan kurang dari 1000000.")
            return
        }

        houses := make([]int, m)
        fmt.Printf("Masukkan nomor rumah kerabat untuk daerah ke-%d: ",
i+1)
        for j := 0; j < m; j++ {
            fmt.Scan(&houses[j])
        }

        var odd, even []int
        for _, house := range houses {
            if house%2 == 0 {
                even = append(even, house)
            } else {
                odd = append(odd, house)
            }
        }
    }
}
```



```

sort.Ints(odd)
sort.Ints(even)

fmt.Printf("Hasil urutan rumah untuk daerah ke-%d: ", i+1)
for _, house := range odd {
    fmt.Printf("%d ", house)
}
for _, house := range even {
    fmt.Printf("%d ", house)
}
fmt.Println()
}
}

```

Output :

```

PS D:\titan\titan 2\PRAKTIKUM ALPRO\2311102042_Ahmad Titana Nanda Pramudya_Modul11\unguided> go run 1.go
Masukkan jumlah daerah (n): 3
Masukkan jumlah rumah kerabat untuk daerah ke-1: 6
Masukkan nomor rumah kerabat untuk daerah ke-1: 5 2 1 7 9 13
Hasil urutan rumah untuk daerah ke-1: 1 5 7 9 13 2
Masukkan jumlah rumah kerabat untuk daerah ke-2: 7
Masukkan nomor rumah kerabat untuk daerah ke-2: 6 189 15 27 39 75 133
Hasil urutan rumah untuk daerah ke-2: 15 27 39 75 133 189 6
Masukkan jumlah rumah kerabat untuk daerah ke-3: 4
Masukkan nomor rumah kerabat untuk daerah ke-3: 3 4 9 1
Hasil urutan rumah untuk daerah ke-3: 1 3 9 4
PS D:\titan\titan 2\PRAKTIKUM ALPRO\2311102042_Ahmad Titana Nanda Pramudya_Modul11\unguided>

```

Penjelasan :

Program ini untuk memisahkan nomor rumah kerabat dalam suatu daerah menjadi dua kelompok, yaitu bilangan ganjil dan bilangan genap, lalu mengurutkannya secara menaik sebelum ditampilkan.

NO.2

Source code :

```
package main

import (
    "fmt"
    "sort"
)

func hitungMedian(data []int) int {
    n := len(data)
    if n == 0 {
        return 0
    }

    if n%2 != 0 {
        return data[n/2]
    }

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

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

    fmt.Printf("Masukkan angka :")

    for {

        var num int
        fmt.Scan(&num)

        if num < 0 {
            break
        }

        if num == 0 {

            tempData := make([]int, len(data))
            copy(tempData, data)
            sort.Ints(tempData)
```

```

        median := hitungMedian(tempData)
        masukan = append(masukan, median)
    } else {

        data = append(data, num)
    }
}

fmt.Println("Hasil keluaran:")
for _, median := range masukan {
    fmt.Println(median)
}

```

Ouput :

```

PS D:\2-go\32721-syntax-error-non-declaration-statement-outside-function-body> go run 2.go
PS D:\titan\titan 2\PRAKTIKUM ALPRO\2311102042_Ahmad Titana Nanda Pramudya_Modul11\unguided> go run 2.go
Masukkan angka :7 23 11 0 5 19 2 29 3 13 17 0 -5313
Hasil keluaran:
11
12
PS D:\titan\titan 2\PRAKTIKUM ALPRO\2311102042_Ahmad Titana Nanda Pramudya_Modul11\unguided>

```

Penjelasan :

Program ini adalah solusi praktis untuk menghitung median dari data dinamis dengan penghentian berbasis angka negatif..

NO.3

Source code :

```
package main

import (
    "fmt"
    "sort"
)

const nMax = 7919

type Buku struct {
    ID      string
    Judul   string
    Penulis string
    Penerbit string
    Eksemplar int
    Tahun   int
    Rating  int
}

type DaftarBuku []Buku

func tampilkanBuku(buku Buku) {
    fmt.Printf("ID: %s, Judul: %s, Penulis: %s, Penerbit: %s, Tahun: %d, Rating: %d\n",
        buku.ID, buku.Judul, buku.Penulis, buku.Penerbit, buku.Tahun, buku.Rating)
}

func cetakTerfavorit(pustaka DaftarBuku) {
    if len(pustaka) == 0 {
        fmt.Println("Tidak ada buku di pustaka.")
        return
    }
    terfavorit := pustaka[0]
```

```

for _, buku := range pustaka {
    if buku.Rating > terfavorit.Rating {
        terfavorit = buku
    }
}

fmt.Println("Buku Terfavorit:")

tampilkanBuku(terfavorit)
}

func urutkanBuku(pustaka DaftarBuku) {
    sort.Slice(pustaka, func(i, j int) bool {
        return pustaka[i].Rating > pustaka[j].Rating
    })
}

func cetak5Terbaru(pustaka DaftarBuku) {
    if len(pustaka) == 0 {
        fmt.Println("Tidak ada buku di pustaka.")
        return
    }

    fmt.Println("5 Buku dengan Rating Tertinggi:")

    for i := 0; i < 5 && i < len(pustaka); i++ {
        tampilkanBuku(pustaka[i])
    }
}

func cariBuku(pustaka DaftarBuku, rating int) {
    urutkanBuku(pustaka)

    left, right := 0, len(pustaka)-1

    for left <= right {

```

```

        mid := (left + right) / 2

        if pustaka[mid].Rating == rating {

            fmt.Println("Buku ditemukan dengan rating tersebut:")

            tampilkanBuku(pustaka[mid])

            return

        } else if pustaka[mid].Rating < rating {

            right = mid - 1

        } else {

            left = mid + 1

        }

    }

    fmt.Println("Tidak ada buku dengan rating seperti itu.")
}

func main() {

    var n int

    fmt.Print("Masukkan jumlah buku: ")

    fmt.Scan(&n)

    pustaka := make(DaftarBuku, n)

    for i := 0; i < n; i++ {

        fmt.Printf("Masukkan data untuk buku ke-%d (ID, Judul, Penulis, Penerbit, Eksemplar, Tahun, Rating):\n", i+1)

        fmt.Scan(&pustaka[i].ID, &pustaka[i].Judul, &pustaka[i].Penulis, &pustaka[i].Penerbit,
            &pustaka[i].Eksemplar, &pustaka[i].Tahun, &pustaka[i].Rating)

    }

    var ratingCari int

```

```

    fmt.Print("Masukkan rating buku yang ingin dicari: ")

    fmt.Scan(&ratingCari)

    cetakTerfavorit(pustaka)

    urutkanBuku(pustaka)

    cetak5Terbaru(pustaka)

    cariBuku(pustaka, ratingCari)

}

```

Output :

```

PS D:\titan\titan 2\PRAKTIKUM ALPRO\2311102042_Ahmad Titana Nanda Pramudya_Modul11\unguided> go run 3
Masukkan jumlah buku: 3
Masukkan data untuk buku ke-1 (ID, Judul, Penulis, Penerbit, Eksemplar, Tahun, Rating):
001 HerryPotter Rowling Bloomsbury 10 1997 95
Masukkan data untuk buku ke-2 (ID, Judul, Penulis, Penerbit, Eksemplar, Tahun, Rating):
002 Hobbit Tolkein Harpercollins 8 1937 89
Masukkan data untuk buku ke-3 (ID, Judul, Penulis, Penerbit, Eksemplar, Tahun, Rating):
003 Twilight Meyer LittleBrown 5 2005 79
Masukkan rating buku yang ingin dicari: 89
Buku Terfavorit:
ID: 001, Judul: HerryPotter, Penulis: Rowling, Penerbit: Bloomsbury, Tahun: 1997, Rating: 95
5 Buku dengan Rating Tertinggi:
ID: 001, Judul: HerryPotter, Penulis: Rowling, Penerbit: Bloomsbury, Tahun: 1997, Rating: 95
ID: 002, Judul: Hobbit, Penulis: Tolkein, Penerbit: Harpercollins, Tahun: 1937, Rating: 89
ID: 003, Judul: Twilight, Penulis: Meyer, Penerbit: LittleBrown, Tahun: 2005, Rating: 79
Buku ditemukan dengan rating tersebut:
ID: 002, Judul: Hobbit, Penulis: Tolkein, Penerbit: Harpercollins, Tahun: 1937, Rating: 89
PS D:\titan\titan 2\PRAKTIKUM ALPRO\2311102042_Ahmad Titana Nanda Pramudya_Modul11\unguided>

```

Penjelasan :

Program ini untuk pengelolaan data buku dalam sebuah perpustakaan. Program ini mampu melakukan berbagai operasi penting seperti menemukan buku dengan rating tertinggi, mengurutkan buku berdasarkan rating secara menurun, mencetak daftar 5 buku terbaik, dan mencari buku berdasarkan rating tertentu menggunakan pencarian biner.