LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2

MODUL XII-XIII

PENGURUTAN DATA

Disusun Oleh :
Wisnu Rananta Raditya Putra / 2311102013
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

Ide 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) Pindahkanltukar 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.

	Notasi Algoritma	Notasi dalam bahasa Go
1	i ← 1	i = 1
2	while i <= n-1 do	for i <= n-1 {
3	idx_min ← i - 1	idx_min = i - 1
4	j ← i	j = i
5	while j < n do	for j < n {
6	if a[idx_min] > a[j] then	if a[idx_min] > a[j] {
7	idx_min ← j	idx_min = j
8	endif	}
9	j ← j + 1	j = j + 1
10	endwhile	}
11	t ← a[idx_min]	t = a[idx_min]
12	a[idx_min] ← a[i-1]	a[idx_min] = a[i-1]
13	a[i-1] ← t	a[i-1] = t
14	i ← i + 1	i = i + 1
15	endwhile	}

Ide 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 data yang sudah terurut dan tetap menjaga keterurutan.

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.

	Notasi Algoritma	Notasi dalam bahasa Go
1	`i ← 1`	`i = 1`
2	`while i <= n-1 do`	`for i <= n-1 {`
3	`    j ← i`	`    j = i`
4	`    temp ← a[j]`	`    temp = a[j]`
5	`    while j > 0 and temp > a[j-1] do`	`    for j > 0 && temp > a[j-1] {`
6	`        a[j] ← a[j-1]`	`        a[j] = a[j-1]`
7	`        j ← j - 1`	`        j = j - 1`
8	`    endwhile`	`    }`
9	`    a[j] ← temp`	`    a[j] = temp`
10	`    i ← i + 1`	`    i = i + 1`
11	`endwhile`	`}`

II. GUIDED

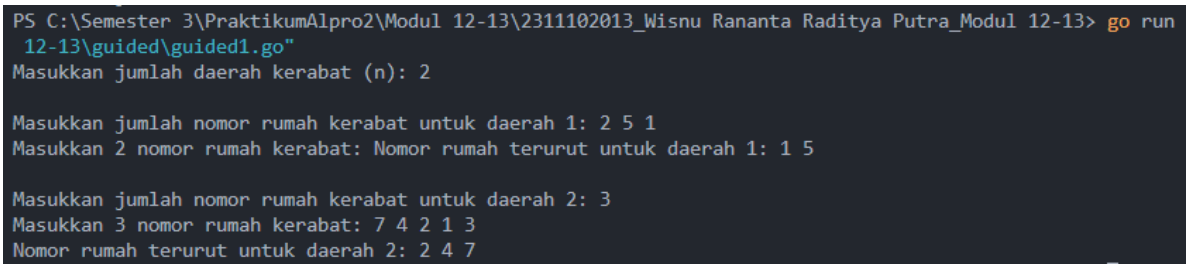
Guided 1

Source Code

```
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()
    }
}
```

Screenshots Output:



Deskripsi:

Program ini ditulis dalam bahasa Go untuk mengurutkan nomor rumah kerabat di beberapa daerah menggunakan Selection Sort. User diminta memasukkan jumlah daerah yang ingin diproses, lalu untuk setiap daerah, user memasukkan jumlah nomor rumah serta daftar nomor rumahnya. Nomor rumah diurutkan dari terkecil ke terbesar. Setelah proses pengurutan selesai, program menampilkan daftar nomor rumah yang sudah terurut untuk setiap daerah.

Guided 2

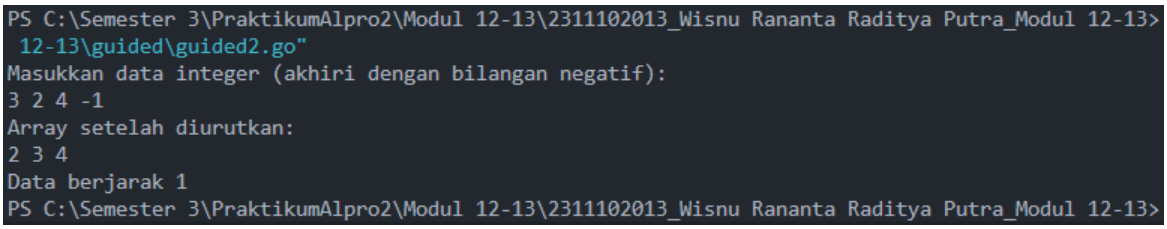
Source Code

```
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")
    }
}
```

Screenshots Output:



Deskripsi:

Program ini ditulis dalam bahasa Go untuk mengurutkan bilangan bulat menggunakan metode Insertion Sort. Program di atas meminta user memasukkan deretan bilangan bulat (integer) hingga user memasukkan bilangan negatif sebagai penanda akhir input. Bilangan-bilangan yang dimasukkan kemudian diurutkan. Setelah itu, program memeriksa apakah selisih antara setiap elemen yang berurutan dalam array hasil pengurutan adalah tetap. Jika selisihnya tetap, program akan menampilkan selisih tersebut; jika tidak, program akan menyatakan bahwa data memiliki jarak yang tidak tetap. Hasil pengurutan dan status jarak elemen ditampilkan kepada user.

III. UNGUIDED

Unguided 1

Source Code:

```
//WISNU RANANTA RADITYA PUTRA (2311102013) IF-11-06
package main
import "fmt"

func selectionSort(arr []int, asc_2311102013 bool) {
    n := len(arr)
    for i := 0; i < n-1; i++ {
        idx := i
        for j := i + 1; j < n; j++ {
            if (asc_2311102013 && arr[j] < arr[idx]) || (!asc_2311102013
&& arr[j] > arr[idx]) {
                idx = j
            }
        }
        arr[i], arr[idx] = arr[idx], arr[i]
```

```
    }
}

func printSeparator() {
    fmt.Println("\n==================================================")
}

func printDashedSeparator() {
    fmt.Println("--------------------------------------------------")
}

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

    var odd, even []int
    for _, num := range arr {
        if num%2 == 0 {
            even = append(even, num)
        } else {
            odd = append(odd, num)
        }
    }
    return odd, even
}

func printSortedResults(i int, odd, even []int) {

    selectionSort(odd, true)
    selectionSort(even, false)

    fmt.Printf("\nNomor rumah terurut untuk daerah ke-%d:\n", i+1)

    for _, num := range odd {
        fmt.Printf("%d ", num)
    }

    for _, num := range even {
        fmt.Printf("%d ", num)
    }
    fmt.Println()
}

func main() {
```

```
    var n int
    printSeparator()
    fmt.Print("Masukkan jumlah daerah kerabat (n): ")
    fmt.Scan(&n)

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

        odd, even := processDaerah(i, m)

        printSortedResults(i, odd, even)
        printDashedSeparator()
    }
}
```

Screenshots Output:

```
PS C:\Semester 3\PraktikumAlpro2\Modul 12-13\2311102013_Wisnu Rananta Raditya Putra_Modul 12-13> go run "c:\
 12-13\unguided\unguided1.go"

==================================================
Masukkan jumlah daerah kerabat (n): 3
--------------------------------------------------
Masukkan jumlah rumah kerabat di daerah ke-1 (m): 5 2 1 7 9 13
Masukkan 5 nomor rumah kerabat untuk daerah ke-1:
Nomor rumah terurut untuk daerah ke-1:
1 7 9 13 2
--------------------------------------------------
--------------------------------------------------
Masukkan jumlah rumah kerabat di daerah ke-2 (m): 6 189 15 27 39 75 133
Masukkan 6 nomor rumah kerabat untuk daerah ke-2:
Nomor rumah terurut untuk daerah ke-2:
15 27 39 75 133 189
--------------------------------------------------
--------------------------------------------------
Masukkan jumlah rumah kerabat di daerah ke-3 (m): 3 4 9 1
Masukkan 3 nomor rumah kerabat untuk daerah ke-3:
Nomor rumah terurut untuk daerah ke-3:
1 9 4
--------------------------------------------------
PS C:\Semester 3\PraktikumAlpro2\Modul 12-13\2311102013_Wisnu Rananta Raditya Putra_Modul 12-13>
```

Deskripsi:

Program ini merupakan program sederhana yang menggunakan bahasa Go untuk engelompokkan dan mengurutkan nomor rumah kerabat di beberapa daerah. User diminta memasukkan jumlah daerah, lalu untuk setiap daerah, memasukkan jumlah rumah dan daftar nomor rumahnya. Nomor rumah dipisah menjadi dua kelompok: ganjil dan genap. Nomor rumah ganjil diurutkan secara naik (ascending), sedangkan nomor rumah genap diurutkan secara turun (descending). Hasil pengurutan dari tiap daerah ditampilkan secara terpisah dengan format yang rapi, dilengkapi garis pemisah untuk memperjelas tampilan.

Unguided 2

Source Code

```
//WISNU RANANTA RADITYA PUTRA (2311102013) IF-11-06
package main

import (
    "bufio"
    "fmt"
    "os"
    "strconv"
    "strings"
)

func selectionSort(arr []int) {
    n := len(arr)
    for i := 0; i < n-1; i++ {
        minIdx_2311102013 := i
        for j := i + 1; j < n; j++ {
            if arr[j] < arr[minIdx_2311102013] {
                minIdx_2311102013 = j
            }
        }
        arr[i], arr[minIdx_2311102013] = arr[minIdx_2311102013], arr[i]
    }
}

func calculateMedian(arr []int) float64 {
    n := len(arr)
    if n%2 == 0 {
        return float64(arr[n/2-1]+arr[n/2]) / 2.0
    }
    return float64(arr[n/2])
}

func main() {
    scanner := bufio.NewScanner(os.Stdin)
    fmt.Println("Masukkan angka yang dipisahkan dengan spasi (akhiri
dengan -5313):")
    scanner.Scan()
    line := scanner.Text()
    parts := strings.Split(line, " ")
    var data []int

    for _, part := range parts {
        num, err := strconv.Atoi(part)
        if err != nil {
            fmt.Println("Input tidak valid, harap masukkan angka bulat
saja.")
```

```
            return
        }

        if num == -5313 {
            break
        } else if num == 0 {
            selectionSort(data)
            median := calculateMedian(data)
            fmt.Printf("%.0f\n", median)
        } else {
            data = append(data, num)
        }
    }
}
```

Screenshots Output:

```
PS C:\Semester 3\PraktikumAlpro2\Modul 12-13\2311102013_Wisnu Rananta Raditya Putra_Modul 12-13> go
 12-13\unguided\unguided2.go"
Masukkan angka yang dipisahkan dengan spasi (akhiri dengan -5313):
7 24 11 0 5 19 2 29 3 13 17 0 -5313
11
12
PS C:\Semester 3\PraktikumAlpro2\Modul 12-13\2311102013_Wisnu Rananta Raditya Putra_Modul 12-13>
```

Deskripsi:

Program di atas meminta user memasukkan deretan angka yang dipisahkan oleh spasi. User dapat terus menambahkan angka ke dalam data sampai memasukkan angka -5313, yang menandakan akhir input. Jika user memasukkan angka 0, program akan mengurutkan data menggunakan metode Selection Sort, lalu menghitung dan menampilkan nilai median dari data tersebut. Program dirancang untuk membaca input dalam satu baris dan menangani angka bulat saja, dengan pesan kesalahan jika input tidak valid.

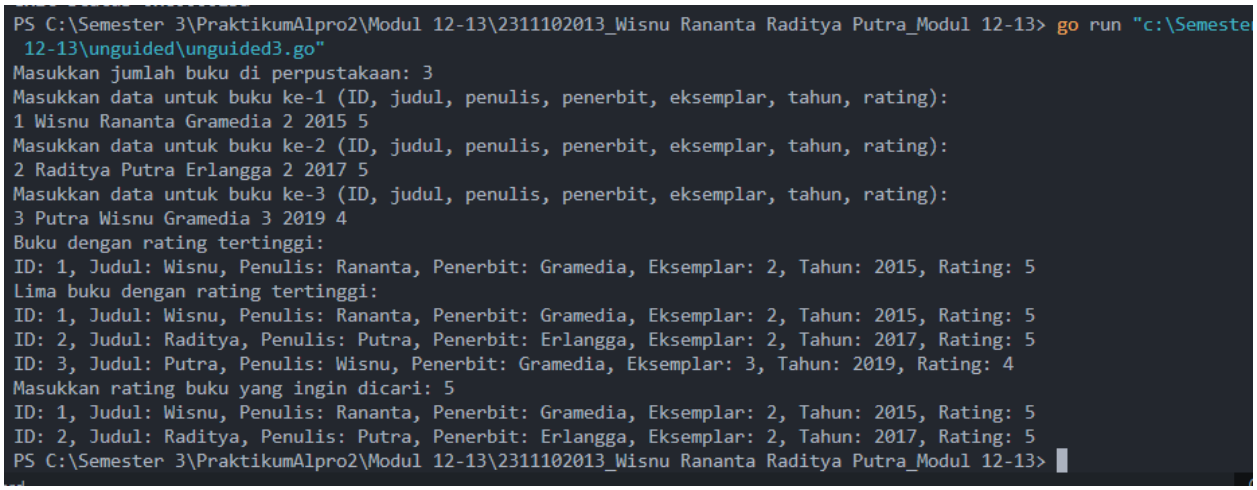
Unguided 3

```
//WISNU RANANTA RADITYA PUTRA (2311102013) IF-11-06
package main
import (
    "fmt"
)
type Buku struct {
    id_2311102013 int
    eksemplar int
    tahun int
    rating int
    judul, penerbit, penulis string
}
```

```
func DaftarkanBuku(pustaka *[]Buku, n int) {
    for i := 0; i < n; i++ {
        var buku Buku
        fmt.Printf("Masukkan data untuk buku ke-%d (ID, judul, penulis,
penerbit, eksemplar, tahun, rating):\n", i+1)
        fmt.Scan(&buku.id_2311102013, &buku.judul, &buku.penulis,
&buku.penerbit, &buku.eksemplar, &buku.tahun, &buku.rating)
        *pustaka = append(*pustaka, buku)
    }
}

func CetakFavorit(pustaka []Buku, n int) {
    if len(pustaka) == 0 {
        fmt.Println("Pustaka kosong.")
        return
    }
    terfavorit := pustaka[0]
    for _, buku := range pustaka {
        if buku.rating > terfavorit.rating {
            terfavorit = buku
        }
    }
    fmt.Println("Buku dengan rating tertinggi:")
    fmt.Printf("ID: %d, Judul: %s, Penulis: %s, Penerbit: %s,
Eksemplar: %d, Tahun: %d, Rating: %d\n",
        terfavorit.id_2311102013, terfavorit.judul, terfavorit.penulis,
terfavorit.penerbit, terfavorit.eksemplar, terfavorit.tahun,
terfavorit.rating)
}

func UrutkanBuku(pustaka *[]Buku, n int) {
    for i := 1; i < len(*pustaka); i++ {
        key := (*pustaka)[i]
        j := i - 1
        for j >= 0 && (*pustaka)[j].rating < key.rating {
            (*pustaka)[j+1] = (*pustaka)[j]
            j--
        }
        (*pustaka)[j+1] = key
    }
}

func Cetak5Terbaik(pustaka []Buku, n int) {
    fmt.Println("Lima buku dengan rating tertinggi:")
    for i := 0; i < 5 && i < len(pustaka); i++ {
        buku := pustaka[i]
        fmt.Printf("ID: %d, Judul: %s, Penulis: %s, Penerbit: %s,
Eksemplar: %d, Tahun: %d, Rating: %d\n",
```

```
            buku.id_2311102013, buku.judul, buku.penulis,
buku.penerbit, buku.eksemplar, buku.tahun, buku.rating)
    }
}

func CariBuku(pustaka []Buku, n int, r int) {
    ditemukan := false
    for _, buku := range pustaka {
        if buku.rating == r {
            ditemukan = true
            fmt.Printf("ID: %d, Judul: %s, Penulis: %s, Penerbit: %s,
Eksemplar: %d, Tahun: %d, Rating: %d\n",
                buku.id_2311102013, buku.judul, buku.penulis,
buku.penerbit, buku.eksemplar, buku.tahun, buku.rating)
        }
    }
    if !ditemukan {
        fmt.Println("Tidak ada buku dengan rating tersebut.")
    }
}
func main() {
    var n int
    fmt.Print("Masukkan jumlah buku di perpustakaan: ")
    fmt.Scan(&n)

    if n <= 0 || n > 7919 {
        fmt.Println("Jumlah buku harus antara 1 hingga 7919.")
        return
    }

    var pustaka []Buku

    DaftarkanBuku(&pustaka, n)
    CetakFavorit(pustaka, n)
    UrutkanBuku(&pustaka, n)

    Cetak5Terbaik(pustaka, n)
    var rating int
    fmt.Print("Masukkan rating buku yang ingin dicari: ")
    fmt.Scan(&rating)
    CariBuku(pustaka, n, rating)
}
```

Screenshots Output:



Deskripsi:

Program di atas digunakan untuk mengelola data buku di perpustakaan. User diminta memasukkan jumlah buku (dengan batas maksimal 7919) dan data tiap buku, seperti ID, judul, penulis, penerbit, eksemplar, tahun, dan rating. Program memiliki beberapa fitur, yaitu mencetak buku dengan rating tertinggi, mengurutkan buku berdasarkan rating secara menurun, menampilkan lima buku dengan rating tertinggi, dan mencari buku berdasarkan rating tertentu. Program dirancang untuk membantu mengelola dan menganalisis koleksi buku dengan fokus pada rating sebagai indikator utama.