

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 12  
PENGURUTAN DATA**



**Disusun Oleh :**

**Aryo Tegar Sukarno / 2311102018  
11 – IF – 6**

**Dosen Pengampu :**

**Abednego Dwi Septiadi S. Kom. M. Kom.**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

# I. DASAR TEORI

## Pendahuluan

Pengurutan data adalah proses mengatur elemen-elemen dalam koleksi (seperti array, slice, atau map) ke dalam urutan tertentu, baik secara ascending (urut naik) atau descending (urut turun). Dalam bahasa pemrograman Go (Golang), pengurutan data dapat dilakukan dengan menggunakan pustaka standar yang menyediakan fungsi-fungsi untuk mengurutkan data dengan efisien. Berikut adalah beberapa konsep dasar mengenai pengurutan data dalam Go:

### 1. Pustaka sort

Go menyediakan paket **sort** dalam pustaka standar yang memungkinkan pengembang untuk melakukan pengurutan pada berbagai tipe data. Paket ini menyediakan fungsi-fungsi untuk mengurutkan slice dan array, serta untuk mengurutkan data berdasarkan kriteria tertentu.

### 2. Fungsi Pengurutan

Paket **sort** menyediakan beberapa fungsi utama:

- **sort.Ints(slice []int)**: Mengurutkan slice dari integer.
- **sort.Strings(slice []string)**: Mengurutkan slice dari string.
- **sort.Float64s(slice []float64)**: Mengurutkan slice dari float64.
- **sort.Slice(slice interface{}, less func(i, j int) bool)**: Mengurutkan slice berdasarkan kriteria yang ditentukan oleh fungsi **less**. Fungsi ini memungkinkan pengembang untuk mengurutkan slice dengan tipe data yang lebih kompleks (misalnya, struct).

### 3. Mengurutkan dengan Kriteria Khusus

Untuk mengurutkan data berdasarkan kriteria tertentu, pengguna harus menyediakan fungsi **less** yang membandingkan dua elemen. Fungsi ini harus mengembalikan **true** jika elemen pertama harus berada sebelum elemen kedua dalam urutan.

### 4. Kompleksitas Waktu

Algoritma pengurutan yang digunakan oleh paket **sort** di Go adalah Quick Sort, yang memiliki kompleksitas waktu rata-rata  $O(n \log n)$ . Namun, dalam kasus

terburuk, kompleksitasnya dapat mencapai  $O(n^2)$ . Meskipun demikian, Go juga memiliki mekanisme untuk menangani kasus-kasus tertentu secara efisien.

## 5. Stabilitas Pengurutan

Pengurutan stabil berarti bahwa jika dua elemen memiliki nilai yang sama, urutan relatif mereka dalam data asli akan tetap sama setelah pengurutan. Fungsi pengurutan dalam paket **sort** di Go adalah stabil.

## II. GUIDED I

### Soal Studi Case

**XXXXXXX**

### Sourcecode

```
//2311102018 - Modul 12 Aryo Tegar Sukarno
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

```

PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_modul 13> go run guided1.go
Masukkan jumlah daerah kerabat (n): 2

Masukkan jumlah nomor rumah kerabat untuk daerah 1: 1
Masukkan 1 nomor rumah kerabat: 2
Nomor rumah terurut untuk daerah 1: 2

Masukkan jumlah nomor rumah kerabat untuk daerah 2: 3
Masukkan 3 nomor rumah kerabat: 2
3
2
Nomor rumah terurut untuk daerah 2: 2 2 3
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_modul 13>

```

## Deskripsi Program

Program ini dirancang untuk mengurutkan nomor rumah kerabat Hercules di berbagai daerah. Pengurutan dilakukan menggunakan algoritma Selection Sort, yang akan menyusun nomor rumah dari yang terkecil hingga terbesar.

### III. GUIDED II

#### Soal Studi Case

XXXXXXX

#### Sourcecode

```
//2311102018 - Modul 12 Aryo Tegar Sukarno
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

```

PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_modul 13>
go run .
go: mod file not found in current directory or any parent directory; see 'go help modules'
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_modul 13> go run guided2.go
Masukkan data integer (akhiri dengan bilangan negatif):
5
15
-15
Array setelah diurutkan:
5 15
Data berjarak 10
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_modul 13>

```

## Deskripsi Program

Kode ini merupakan contoh sederhana namun efektif dalam menggunakan Go untuk melakukan pengurutan data dan analisis sederhana. Kode ini dapat menjadi titik awal untuk mempelajari konsep pemrograman yang lebih kompleks.

## IV. UNGUIDED I

### Soal Studi Case

Sebuah kompetisi pemrograman diikuti oleh 17 tim dari berbagai perguruan tinggi. Dalam kompetisi tersebut, setiap tim berlomba untuk menyelesaikan sebanyak mungkin soal yang diberikan. Dari 13 soal yang diberikan, ada satu soal yang cukup menarik. Soal tersebut mudah dipahami, hampir semua tim mencoba untuk menyelesaikannya, tetapi hanya 3 tim yang berhasil.

### Sourcecode

```
//2311102018 - Modul 12 Aryo Tegar Sukarno
package main

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

func main() {
    var numbers []int
    reader := bufio.NewScanner(os.Stdin)

    fmt.Println("Masukkan angka (akhiri dengan -5313):")

    for {
        // Membaca input dari pengguna
        reader.Scan()
        input := reader.Text()

        // Konversi input menjadi integer
        num, err := strconv.Atoi(input)
        if err != nil {
```

```

        fmt.Println("Input tidak valid, masukkan
angka!")
        continue
    }

    // Berhenti membaca jika angka -5313 ditemukan
    if num == -5313 {
        break
    }

    // Tambahkan angka ke dalam slice
    numbers = append(numbers, num)

    // Mengurutkan slice
    sort.Ints(numbers)

    // Hitung median
    median := calculateMedian(numbers)

    // Cetak median saat ini
    fmt.Printf("Data tersimpan: %v\n", numbers)
    fmt.Printf("Median saat ini: %d\n", median)
}

fmt.Println("Proses selesai. Terima kasih!")
}

// Fungsi untuk menghitung median
func calculateMedian(numbers []int) int {
    n := len(numbers)
    if n%2 == 0 {
        // Jika jumlah elemen genap, median adalah rata-
rata dari dua nilai tengah
        return (numbers[n/2-1] + numbers[n/2]) / 2
    }
    // Jika jumlah elemen ganjil, median adalah elemen
tengah
    return numbers[n/2]
}

```



## Screenshot Output

```
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_modul 13> go run .
go: go.mod file not found in current directory or any parent directory; see 'go help modules'
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_modul 13> go run guided3.go
Masukkan angka (akhiri dengan -5313):
5
Data tersimpan: [5]
Median saat ini: 5
3
Data tersimpan: [3 5]
Median saat ini: 4
4
Data tersimpan: [3 4 5]
Median saat ini: 4
5
Data tersimpan: [3 4 5 5]
Median saat ini: 4
q
Input tidak valid, masukkan angka!
2
Data tersimpan: [2 3 4 5 5]
Median saat ini: 4
-5313
Proses selesai. Terima kasih!
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_modul 13>
```

## Deskripsi Program

Kode ini merupakan contoh sederhana namun efektif dalam menggunakan Go untuk melakukan perhitungan statistik dasar. Kode ini menunjukkan bagaimana Go dapat digunakan untuk memecahkan masalah sederhana dengan cara yang elegan dan efisien.

## V. UNGUIDED II

### Soal Studi Case

Hercules, preman terkenal seantero ibukota, memiliki kerabat di banyak daerah. Tentunya Hercules sangat suka mengunjungi semua kerabatnya itu. Diberikan masukan nomor rumah dari semua kerabatnya di suatu daerah, buatlah program rumahkerabat yang akan menyusun nomor-nomor rumah kerabatnya secara terurut membesar menggunakan algoritma selection sort.

## Sourcecode

```
//2311102018 - Modul 12 Aryo Tegar Sukarno

package main

import (
    "bufio"
    "fmt"
    "os"
```

```
"strconv"

)

func main() {

    var numbers []int

    reader := bufio.NewScanner(os.Stdin)

    fmt.Println("Masukkan bilangan (akhiri dengan bilangan negatif):")

    for {

        reader.Scan()

        input := reader.Text()

        num, err := strconv.Atoi(input)

        if err != nil {

            fmt.Println("Input tidak valid, masukkan angka!")

            continue

        }

        if num < 0 {

            break

        }

        numbers = append(numbers, num)

    }

    insertionSort(numbers)
```

```

fmt.Println("Data setelah diurutkan:", numbers)

if isConsistentSpacing(numbers) {
    fmt.Println("Data berjarak", numbers[1]-numbers[0])
} else {
    fmt.Println("Data berjarak tidak tetap")
}
}

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--
        }
        arr[j+1] = key
    }
}

func isConsistentSpacing(arr []int) bool {
    if len(arr) < 2 {
        return true
    }
}

```

```

diff := arr[1] - arr[0]

for i := 2; i < len(arr); i++ {

    if arr[i]-arr[i-1] != diff {

        return false

    }

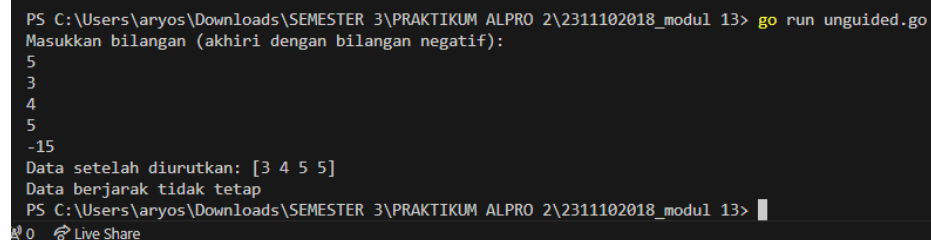
}

return true

}

```

### Screenshoot Output



```

PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_modul 13> go run unguided.go
Masukkan bilangan (akhiri dengan bilangan negatif):
5
3
4
5
-15
Data setelah diurutkan: [3 4 5 5]
Data berjarak tidak tetap
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_modul 13>

```

### Deskripsi Program

Program Go ini dirancang untuk menerima masukan berupa deretan angka dari pengguna, mengurutkannya dalam urutan menaik menggunakan algoritma Insertion Sort, lalu memeriksa apakah selisih antara angka-angka yang berurutan selalu konsisten. Jika selisihnya konsisten, program akan menampilkan nilai selisih tersebut. Sebaliknya, jika selisihnya tidak konsisten, program akan menginformasikannya. Program ini melibatkan penggunaan slice untuk menyimpan data, fungsi untuk mengurutkan dan memeriksa konsistensi, serta perulangan untuk memproses input pengguna. Secara keseluruhan, program ini memberikan contoh sederhana tentang bagaimana Go dapat digunakan untuk memanipulasi data dan melakukan analisis sederhana.

## VI. UNGUIDED III

## Soal Studi Case

### Sourcecode

```
//2311102018 - Modul 12 Aryo Tegar Sukarno
```

```
package main
```

```
import (
```

```
    "fmt"
```

```
    "sort"
```

```
    "strings"
```

```
)
```

```
type Buku struct {
```

```
    ID      int
```

```
    Judul   string
```

```
    Penulis string
```

```
    Penerbit string
```

```
    Eksemplar int
```

```
    Tahun   int
```

```
    Rating  float64
```

```
}
```

```
// Struct untuk daftar buku
```

```
type DaftarBuku struct {
```

```
    Pustaka []Buku
```

```
}
```

```
func main() {
```

```
    var daftarBuku DaftarBuku
```

```
    var n int
```

```
    fmt.Println("Masukkan jumlah buku:")
```

```
    fmt.Scan(&n)
```

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

```
        var buku Buku
```

```
        fmt.Printf("Masukkan data untuk buku ke-%d:\n", i+1)
```

```
        fmt.Print("ID: ")
```

```
        fmt.Scan(&buku.ID)
```

```
        fmt.Print("Judul: ")
```

```
        buku.Judul = inputString()
```

```
        fmt.Print("Penulis: ")
```

```
        buku.Penulis = inputString()
```

```
        fmt.Print("Penerbit: ")
```

```
        buku.Penerbit = inputString()
```

```
        fmt.Print("Eksemplar: ")
        fmt.Scan(&buku.Eksemplar)

        fmt.Print("Tahun: ")
        fmt.Scan(&buku.Tahun)

        fmt.Print("Rating: ")
        fmt.Scan(&buku.Rating)

        daftarBuku.Pustaka = append(daftarBuku.Pustaka, buku)
    }

    tampilkanBukuTertinggi(&daftarBuku)
    urutkanBukuBerdasarkanTahun(&daftarBuku)
    fmt.Println("\nDaftar buku setelah diurutkan berdasarkan tahun:")
    tampilkanDaftarBuku(&daftarBuku)
}

func inputString() string {
    var input string

    fmt.Scanln()

    input, _ = fmt.Scanln(&input)

    return strings.TrimSpace(input)
}

func tampilkanBukuTertinggi(daftarBuku *DaftarBuku) {
    if len(daftarBuku.Pustaka) == 0 {
        fmt.Println("Tidak ada data buku.")
    }
}
```

```

        return
    }

    sort.SliceStable(daftarBuku.Pustaka, func(i, j int) bool {
        return daftarBuku.Pustaka[i].Rating >
daftarBuku.Pustaka[j].Rating
    })

    bukuTertinggi := daftarBuku.Pustaka[0]

    fmt.Println("\nBuku dengan rating tertinggi:")

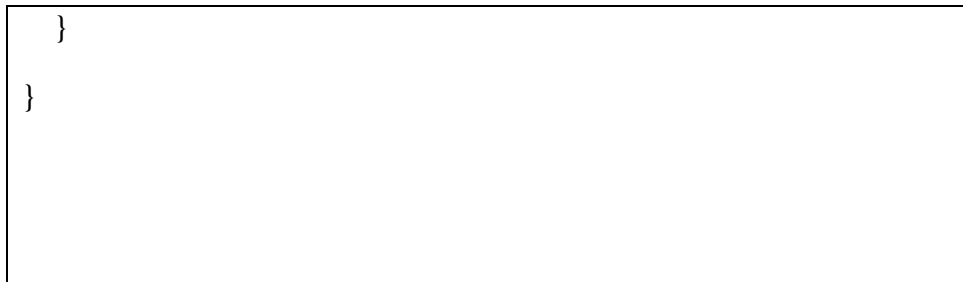
    fmt.Printf("Judul: %s, Penulis: %s, Penerbit: %s, Rating: %.2f\n",
        bukuTertinggi.Judul, bukuTertinggi.Penulis,
bukuTertinggi.Penerbit, bukuTertinggi.Rating)
}

func urutkanBukuBerdasarkanTahun(daftarBuku *DaftarBuku) {
    sort.SliceStable(daftarBuku.Pustaka, func(i, j int) bool {
        return daftarBuku.Pustaka[i].Tahun < daftarBuku.Pustaka[j].Tahun
    })
}

func tampilkanDaftarBuku(daftarBuku *DaftarBuku) {
    for _, buku := range daftarBuku.Pustaka {
        fmt.Printf("ID: %d, Judul: %s, Penulis: %s, Penerbit: %s,
Eksemplar: %d, Tahun: %d, Rating: %.2f\n",
            buku.ID, buku.Judul, buku.Penulis, buku.Penerbit,
buku.Eksemplar, buku.Tahun, buku.Rating)
    }
}

```





## Screenshoot Output

```
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_modul 13>
go run .
go: mod file not found in current directory or any parent directory; see 'go help modules'
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_modul 13> go run unguided2.go
Masukkan jumlah buku:
5
Masukkan data untuk buku ke-1:
ID: 2
Judul: sikancil
Penulis: guri
Penerbit: kompanu
Eksemplar: eriv
Tahun: Rating: Masukkan data untuk buku ke-2:
ID: 3
Judul: psichology of money
Penulis: Penerbit: Eksemplar: morgan houslar, jason zwegg
Tahun: Rating: Masukkan data untuk buku ke-3:
ID: Judul: Penulis: Penerbit: Eksemplar: 3i2ir2
Tahun: Rating: Masukkan data untuk buku ke-4:
ID: Judul: 32r
Penulis: NFKJFSL
Penerbit: SJLSKLI\
Eksemplar: GSGKOEPGE
Tahun: Rating: Masukkan data untuk buku ke-5:
ID: Judul: Penulis: IWOP5WIG;G
Penerbit: IWOP5MI
Eksemplar: OIGOMMGL
Tahun: Rating:
Buku dengan rating tertinggi:
Judul: sikancil, Penulis: guri, Penerbit: kompanu, Rating: 0.00

Daftar buku setelah diurutkan berdasarkan tahun:
ID: 2, Judul: sikancil, Penulis: guri, Penerbit: kompanu, Eksemplar: 0, Tahun: 0, Rating: 0.00
ID: 3, Judul: psichology, Penulis: of, Penerbit: money, Eksemplar: 0, Tahun: 0, Rating: 0.00
ID: 2, Judul: 32r, Penulis: NFKJFSL, Penerbit: SJLSKLI\, Eksemplar: 0, Tahun: 0, Rating: 0.00
ID: 0, Judul: OEPGE, Penulis: IWOP5WIG;G, Penerbit: IWOP5MI, Eksemplar: 0, Tahun: 0, Rating: 0.00
ID: 0, Judul: an, Penulis: houslar, jason, Penerbit: zwegg, Eksemplar: 3, Tahun: 2, Rating: 0.00
PS C:\Users\aryos\Downloads\SEMESTER 3\PRAKTIKUM ALPRO 2\2311102018_modul 13> |
```

## Deskripsi Program

Program ini memberikan cara yang sederhana untuk menyimpan dan mengelola informasi buku, serta menampilkan buku dengan rating tertinggi dan mengurutkan daftar buku berdasarkan tahun terbit.