

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL XII

PENGURUTAN DATA



Disusun Oleh : R o r y

Refaldo Sinulingga

Dosen Pengampu :

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

DASAR TEORI

Dasar Teori

Dalam dunia pemrograman, pengaturan data adalah aspek penting yang mendasari pengembangan algoritma. Pengaturan data dalam algoritma bertujuan untuk mengorganisasi, menyimpan, dan memproses data secara efisien sehingga menghasilkan solusi optimal untuk suatu masalah. Proses ini melibatkan pemilihan struktur data yang tepat, penggunaan teknik pengurutan (*sorting*), pencarian (*searching*), dan manipulasi data lainnya sesuai dengan kebutuhan aplikasi.

Struktur data adalah salah satu fondasi utama dalam pengaturan data. Struktur data seperti array, linked list, stack, queue, hash table, dan tree digunakan untuk menyimpan dan mengorganisasi data dalam format yang memudahkan akses dan manipulasi. Misalnya, array digunakan untuk menyimpan data secara berurutan, sedangkan tree memungkinkan penyimpanan data secara hierarkis. Pemilihan struktur data yang tepat sangat bergantung pada kebutuhan algoritma, seperti kecepatan akses, efisiensi penyimpanan, atau kemudahan manipulasi.

Algoritma pengurutan (*sorting*) adalah teknik penting dalam pengaturan data. Pengurutan memungkinkan data diatur berdasarkan suatu urutan tertentu, seperti dari nilai terkecil ke terbesar atau sebaliknya. Contoh algoritma *sorting* meliputi *bubble sort*, *merge sort*, *quick sort*, dan *heap sort*. Pengurutan mempermudah operasi lain, seperti pencarian data dengan algoritma seperti *binary search*, yang memanfaatkan sifat data yang sudah terurut untuk mempercepat proses pencarian.

I. GUIDED

1.

Sourcecode

```
package main

import (
    "fmt"
)

// Fungsi untuk mengurutkan array menggunakan Selection Sort
func selectionSort_aldo_323(arr_aldo_323 []int, n_aldo_323 int) {
    for i_aldo_323 := 0; i_aldo_323 < n_aldo_323-1; i_aldo_323++ {
        idxMin_aldo_323 := i_aldo_323
        for j_aldo_323 := i_aldo_323 + 1; j_aldo_323 < n_aldo_323; j_aldo_323++ {
            // Cari elemen terkecil
            if arr_aldo_323[j_aldo_323] < arr_aldo_323[idxMin_aldo_323] {
                idxMin_aldo_323 = j_aldo_323
            }
        }
        // Tukar elemen terkecil dengan elemen di posisi i
        arr_aldo_323[i_aldo_323], arr_aldo_323[idxMin_aldo_323] = arr_aldo_323[idxMin_aldo_323], arr_aldo_323[i_aldo_323]
    }
}

func main() {
    var n_aldo_323 int
    fmt.Println("=== Program Pengurutan Nomor Rumah Kerabat ===")
    fmt.Print("Masukkan jumlah daerah kerabat (n): ")
    fmt.Scan(&n_aldo_323)

    fmt.Println("\nMulai memasukkan data setiap daerah...\n")

    // Proses tiap daerah
    for daerah_aldo_323 := 1; daerah_aldo_323 <= n_aldo_323; daerah_aldo_323++ {
        var m_aldo_323 int
        fmt.Printf("Daerah %d:\n", daerah_aldo_323)
        fmt.Print("Masukkan jumlah nomor rumah kerabat: ")
        fmt.Scan(&m_aldo_323)

        // Membaca nomor rumah untuk daerah ini
        arr_aldo_323 := make([]int, m_aldo_323)
        fmt.Printf("Masukkan %d nomor rumah kerabat (pisahkan dengan spasi): ", m_aldo_323)
        for i_aldo_323 := 0; i_aldo_323 < m_aldo_323; i_aldo_323++ {
            fmt.Scan(&arr_aldo_323[i_aldo_323])
        }

        // Urutkan array dari terkecil ke terbesar
        selectionSort_aldo_323(arr_aldo_323, m_aldo_323)

        // Tampilkan hasil
        fmt.Printf("Nomor rumah terurut: ")
        for _, num_aldo_323 := range arr_aldo_323 {
            fmt.Printf("%d ", num_aldo_323)
        }
        fmt.Println("\n")
    }

    fmt.Println("=== Program Selesai ===")
}
```

Screenshoot Output

```
=== Program Pengurutan Nomor Rumah Kerabat ===
Masukkan jumlah daerah kerabat (n): 2

Mulai memasukkan data setiap daerah...

Daerah 1:
Masukkan jumlah nomor rumah kerabat: 2
Masukkan 2 nomor rumah kerabat (pisahkan dengan spasi): 12 15
Nomor rumah terurut: 12 15

Daerah 2:
Masukkan jumlah nomor rumah kerabat: 2
Masukkan 2 nomor rumah kerabat (pisahkan dengan spasi): 6 12
Nomor rumah terurut: 6 12

=== Program Selesai ===
```

Deskripsi Program

Program ini merupakan aplikasi berbasis terminal untuk mengurutkan nomor rumah relatif di beberapa wilayah menggunakan algoritma pengurutan selektif. Pengguna memasukkan jumlah wilayah, kemudian jumlah nomor rumah dan daftar nomor rumah setiap wilayah. Program ini mengurutkan nomor rumah dari terkecil hingga terbesar dan menampilkan hasilnya untuk setiap area. Proses ini diulangi hingga seluruh area diproses. Dirancang sederhana dan mudah digunakan, program ini memberikan demonstrasi praktis penerapan algoritma pengurutan dasar. Cocok untuk kebutuhan penyortiran data sederhana atau mempelajari algoritma penyortiran.

2.

Sourcecode

```
package main

import (
    "fmt"
)

// Fungsi untuk mengurutkan array menggunakan Insertion Sort
func insertionSort_aldo_323(arr_aldo_323 []int, n_aldo_323 int) {
    for i_aldo_323 := 1; i_aldo_323 < n_aldo_323; i_aldo_323++ {
        key_aldo_323 := arr_aldo_323[i_aldo_323]
        j_aldo_323 := i_aldo_323 - 1
        // Geser elemen yang lebih besar dari key ke kanan
        for j_aldo_323 >= 0 && arr_aldo_323[j_aldo_323] > key_aldo_323 {
            arr_aldo_323[j_aldo_323+1] = arr_aldo_323[j_aldo_323]
            j_aldo_323--
        }
        arr_aldo_323[j_aldo_323+1] = key_aldo_323
    }
}

// Fungsi untuk memeriksa apakah selisih elemen array tetap
func isConstantDifference_aldo_323(arr_aldo_323 []int, n_aldo_323 int) (bool, int) {
    if n_aldo_323 < 2 {
        return true, 0
    }

    difference_aldo_323 := arr_aldo_323[1] - arr_aldo_323[0]
    for i_aldo_323 := 1; i_aldo_323 < n_aldo_323-1; i_aldo_323++ {
        if arr_aldo_323[i_aldo_323+1]-arr_aldo_323[i_aldo_323] != difference_aldo_323 {
            return false, 0
        }
    }
    return true, difference_aldo_323
}

func main() {
    var arr_aldo_323 []int
    var num_aldo_323 int

    fmt.Println("=== Program Memeriksa Selisih Elemen Array ===")
    fmt.Println("Masukkan data integer satu per satu (akhiri dengan bilangan negatif):")
    // Input data hingga bilangan negatif ditemukan
    for {
        fmt.Print("Masukkan bilangan: ")
        fmt.Scan(&num_aldo_323)
        if num_aldo_323 < 0 {
            break
        }
        arr_aldo_323 = append(arr_aldo_323, num_aldo_323)
    }

    // Periksa jika array kosong
    if len(arr_aldo_323) == 0 {
        fmt.Println("\nTidak ada data yang dimasukkan.")
        return
    }

    n_aldo_323 := len(arr_aldo_323)
    // Urutkan array menggunakan Insertion Sort
    insertionSort_aldo_323(arr_aldo_323, n_aldo_323)

    // Periksa apakah selisih elemen tetap
    isConstantDifference_aldo_323, difference_aldo_323 := isConstantDifference_aldo_323(arr_aldo_323, n_aldo_323)
    fmt.Println("\n=== Hasil Program ===")
    // Tampilkan hasil pengurutan
    fmt.Print("Array setelah diurutkan: ")
    for _, val_aldo_323 := range arr_aldo_323 {
        fmt.Printf("%d ", val_aldo_323)
    }
}
```

```
}
fmt.Println()

// Tampilkan status jarak
if isConstant_aldo_323 {
    fmt.Printf("Selisih antar elemen tetap: %d\n", difference_aldo_323)
} else {
    fmt.Println("Selisih antar elemen tidak tetap.")
}
fmt.Println("=== Program Selesai ===")
}
```

Screenshot output

```
=== Hasil Program ===
Array setelah diurutkan: 8 12 14
Selisih antar elemen tidak tetap.
=== Program Selesai ===
PS C:\Users\VIVOB00K> go run "C:\Users\VIVOB00K\AppData\Local\Temp\tempCodeRunnerFile.go"
=== Program Memeriksa Selisih Elemen Array ===
Masukkan data integer satu per satu (akhiri dengan bilangan negatif):
Masukkan bilangan: 12
Masukkan bilangan: 8
Masukkan bilangan: -1

=== Hasil Program ===
Array setelah diurutkan: 8 12
Selisih antar elemen tetap: 4
=== Program Selesai ===
```

Deskripsi Program

Tujuan dari program ini adalah untuk memeriksa apakah ada perbedaan antar elemen dalam array setelah mengurutkan elemen dalam urutan menaik menggunakan algoritma insertion sort. Pengguna diminta memasukkan bilangan bulat satu per satu hingga angka negatif dimasukkan untuk menunjukkan akhir input. Program kemudian mengurutkan array menggunakan penyisipan dan menampilkan array yang diurutkan untuk melihat apakah perbedaan antara elemen-elemen dalam array konsisten. Jika perbedaan masih ada, program akan menampilkannya. Jika tidak, program akan memberi tahu Anda bahwa perbedaan antar item belum terselesaikan. Program ini dirancang untuk membantu pengguna memahami pola perbedaan numerik dalam data yang mereka masukkan.

II. UNGUIDED

Soal Studi Case

1.

Sourcecode

```
package main

import (
    "fmt"
    "sort"
)

type Buku_aldo_323 struct {
    ID_aldo_323      int
    Judul_aldo_323   string
    Penulis_aldo_323 string
    Penerbit_aldo_323 string
    Eksemplar_aldo_323 int
    Tahun_aldo_323   int
    Rating_aldo_323  int
}

func main() {
    var n_aldo_323 int
    fmt.Println("=== Program Pengelolaan Data Buku Perpustakaan ===")
    fmt.Print("Masukkan jumlah buku dalam perpustakaan: ")
    fmt.Scan(&n_aldo_323)

    var daftarBuku_aldo_323 []Buku_aldo_323
    fmt.Println("\nMasukkan data setiap buku:")
    for i_aldo_323 := 0; i_aldo_323 < n_aldo_323; i_aldo_323++ {
        var buku_aldo_323 Buku_aldo_323
        fmt.Printf("\nData buku ke-%d:\n", i_aldo_323+1)
        fmt.Print(" ID: ")
        fmt.Scan(&buku_aldo_323.ID_aldo_323)
        fmt.Print(" Judul: ")
        fmt.Scan(&buku_aldo_323.Judul_aldo_323)
        fmt.Print(" Penulis: ")
        fmt.Scan(&buku_aldo_323.Penulis_aldo_323)
        fmt.Print(" Penerbit: ")
        fmt.Scan(&buku_aldo_323.Penerbit_aldo_323)
        fmt.Print(" Eksemplar: ")
        fmt.Scan(&buku_aldo_323.Eksemplar_aldo_323)
        fmt.Print(" Tahun: ")
        fmt.Scan(&buku_aldo_323.Tahun_aldo_323)
        fmt.Print(" Rating: ")
        fmt.Scan(&buku_aldo_323.Rating_aldo_323)

        daftarBuku_aldo_323 = append(daftarBuku_aldo_323, buku_aldo_323)
    }

    var ratingCari_aldo_323 int
    fmt.Print("\nMasukkan rating buku yang ingin dicari: ")
    fmt.Scan(&ratingCari_aldo_323)

    sort.Slice(daftarBuku_aldo_323, func(i_aldo_323, j_aldo_323 int) bool {
        return daftarBuku_aldo_323[i_aldo_323].Rating_aldo_323 > daftarBuku_aldo_323[j_aldo_323].Rating_aldo_323
    })

    fmt.Println("\n=== Data 5 Buku Terfavorit ===")
    for i_aldo_323 := 0; i_aldo_323 < len(daftarBuku_aldo_323) && i_aldo_323 < 5; i_aldo_323++ {
        buku_aldo_323 := daftarBuku_aldo_323[i_aldo_323]
        fmt.Printf("%d. %s (Rating: %d)\n", i_aldo_323+1, buku_aldo_323.Judul_aldo_323, buku_aldo_323.Rating_aldo_323)
    }

    fmt.Println("\n=== Data Buku dengan Rating Sesuai ===")
    found_aldo_323 := false
    for _, buku_aldo_323 := range daftarBuku_aldo_323 {
```

```

        if buku_aldo_323.Rating_aldo_323 == ratingCari_aldo_323 {
            fmt.Printf("ID: %d, Judul: %s, Penulis: %s, Penerbit: %s, Eksemplar: %d, Tahun: %d, Rating: %d\n",
                buku_aldo_323.ID_aldo_323, buku_aldo_323.Judul_aldo_323, buku_aldo_323.Penulis_aldo_323,
                buku_aldo_323.Penerbit_aldo_323, buku_aldo_323.Eksemplar_aldo_323, buku_aldo_323.Tahun_aldo_323,
                buku_aldo_323.Rating_aldo_323)
            found_aldo_323 = true
        }
    }

    if !found_aldo_323 {
        fmt.Printf("Tidak ada buku dengan rating %d\n", ratingCari_aldo_323)
    }

    fmt.Println("=== Program Selesai ===")
}

```

Screeنشott output

```

=== Program Pengelolaan Data Buku Perpustakaan ===
Masukkan jumlah buku dalam perpustakaan: 2

Masukkan data setiap buku:

Data buku ke-1:
ID: 001
Judul: matematika
Penulis: aldo
Penerbit: nusantara
Eksemplar: 30
Tahun: 2017
Rating: 5

Data buku ke-2:
ID: 002
Judul: b.indo
Penulis: aldo
Penerbit: nusantara
Eksemplar: 20
Tahun: 2018
Rating: 4

Masukkan rating buku yang ingin dicari: 4

=== Data 5 Buku Terfavorit ===
1. matematika (Rating: 5)
2. b.indo (Rating: 4)

=== Data Buku dengan Rating Sesuai ===
ID: 2, Judul: b.indo, Penulis: aldo, Penerbit: nusantara, Eksemplar: 20, Tahun: 2018, Rating: 4
=== Program Selesai ===

```

Deskripsi program

Program ini digunakan untuk mengelola data buku dalam perpustakaan dengan merepresentasikan setiap buku menggunakan *struct*. Pengguna dapat memasukkan data buku, seperti ID, judul, penulis, penerbit, jumlah eksemplar, tahun terbit, dan rating. Program akan mengurutkan buku berdasarkan rating secara menurun dan menampilkan lima buku dengan rating tertinggi. Selain itu, program juga memungkinkan pengguna untuk mencari dan menampilkan data buku dengan rating tertentu. Hasilnya ditampilkan secara rapi agar mudah dibaca.

2. 12.3/4

Sourcecode

```
package main

import (
    "fmt"
    "sort"
    "strconv"
    "strings"
)

func main() {
    // Input bilangan dalam bentuk array string
    input_aldo_323 := []string{"7", "23", "11", "19", "2", "29", "13", "17", "-3131314"}

    // Array untuk menyimpan bilangan valid
    var validNumbers_aldo_323 []int

    // Filter bilangan valid (positif dan maksimal 1 juta)
    for _, numStr_aldo_323 := range input_aldo_323 {
        num_aldo_323, err_aldo_323 := strconv.Atoi(numStr_aldo_323)
        if err_aldo_323 == nil && num_aldo_323 > 0 && num_aldo_323 <= 1000000 {
            validNumbers_aldo_323 = append(validNumbers_aldo_323, num_aldo_323)
        }
    }

    // Mengurutkan bilangan valid
    sort.Ints(validNumbers_aldo_323)

    // Menampilkan hasil
    fmt.Println("Hasil Pemrosesan Bilangan:")
    fmt.Println(strings.Repeat("=", 40))
    fmt.Printf("Bilangan Valid: %s\n", strings.Trim(fmt.Sprint(validNumbers_aldo_323), "[]"))
    fmt.Println(strings.Repeat("-", 40))
}
```

Scenshoot output

```
PS C:\Users\VIVOB00K> go run "C:\Users\VIVOB00K\AppData\Local\Temp\tempCodeRunnerFile.go"
Hasil Pemrosesan Bilangan:
=====
Bilangan Valid: 2 7 11 13 17 19 23 29
-----
PS C:\Users\VIVOB00K> go run "C:\Users\VIVOB00K\AppData\Local\Temp\tempCodeRunnerFile.go"
Hasil Pemrosesan Bilangan:
=====
Bilangan Valid: 2 7 11 13 17 19 23 29
-----
PS C:\Users\VIVOB00K>
```

Deskripsi Program

Program di atas dimaksudkan untuk memproses array string yang berisi angka dalam format teks. Program ini pertama-tama memfilter angka valid, yaitu angka positif yang tidak melebihi 1 juta. Nomor yang valid kemudian disimpan dalam array validNumbers_aldo_323. Program kemudian menggunakan fungsi sort.Ints untuk mengurutkan angka dalam urutan menaik. Terakhir, hasilnya akan ditampilkan di layar dengan daftar nomor yang valid setelah disaring dan diurutkan. Pembatas garis digunakan untuk menampilkan hasil program dengan lebih jelas.

3.

Sourcecode

```
package main

import (
    "fmt"
    "sort"
    "strconv"
    "strings"
)

func main() {
    // Membaca input dari pengguna
    var input_aldo_323 string
    fmt.Println("Masukkan deretan bilangan bulat (pisahkan dengan spasi):")
    fmt.Scanln(&input_aldo_323)

    // Memisahkan input menjadi array string
    numStrs_aldo_323 := strings.Fields(input_aldo_323)

    // Konversi string ke integer
    var nums_aldo_323 []int
    for _, str_aldo_323 := range numStrs_aldo_323 {
        num_aldo_323, err_aldo_323 := strconv.Atoi(str_aldo_323)
        if err_aldo_323 != nil {
            fmt.Println("Input tidak valid. Harap masukkan bilangan bulat.")
            return
        }
        nums_aldo_323 = append(nums_aldo_323, num_aldo_323)
    }

    // Filter bilangan ganjil
    var oddNums_aldo_323 []int
    for _, num_aldo_323 := range nums_aldo_323 {
        if num_aldo_323%2 != 0 {
            oddNums_aldo_323 = append(oddNums_aldo_323, num_aldo_323)
        }
    }
}
```

```

// Periksa apakah ada bilangan ganjil
if len(oddNums_aldo_323) == 0 {
    fmt.Println("Tidak ada bilangan ganjil pada data yang dimasukkan.")
    return
}

// Urutkan bilangan ganjil
sort.Ints(oddNums_aldo_323)

// Hitung median dari bilangan ganjil
var median_aldo_323 float64
n_aldo_323 := len(oddNums_aldo_323)
if n_aldo_323%2 == 0 {
    median_aldo_323 = float64(oddNums_aldo_323[n_aldo_323/2-1]+oddNums_aldo_323[n_aldo_323/2]) / 2.0
} else {
    median_aldo_323 = float64(oddNums_aldo_323[n_aldo_323/2])
}

// Tampilkan hasil
fmt.Println("Bilangan ganjil yang ditemukan:", oddNums_aldo_323)
fmt.Printf("Median dari bilangan ganjil: %.2f\n", median_aldo_323)
}

```

Scrennshoot output

Deskripsi

```

PS C:\Users\VIVOB00K> go run "C:\Users\VIVOB00K\AppData\Local\Temp\tempCodeRunnerFile.go"
Masukkan deretan bilangan bulat (pisahkan dengan spasi):
3 6 9 0
Bilangan ganjil yang ditemukan: [3]
Median dari bilangan ganjil: 3.00
PS C:\Users\VIVOB00K> go run "C:\Users\VIVOB00K\AppData\Local\Temp\tempCodeRunnerFile.go"
Masukkan deretan bilangan bulat (pisahkan dengan spasi):
2 4 8 0
Tidak ada bilangan ganjil pada data yang dimasukkan.
PS C:\Users\VIVOB00K>

```

Deskripsi program

Program ini mengambil masukan serangkaian bilangan bulat yang dipisahkan spasi dari pengguna dan memprosesnya untuk menemukan bilangan ganjil dan menghitung median dari bilangan ganjil tersebut. Pada langkah pertama, program membagi masukan menjadi array string dan mengubah setiap elemen menjadi integer. Jika terjadi kesalahan selama konversi, program akan mengeluarkan pesan kesalahan dan keluar. Program tersebut kemudian menyaring hanya angka ganjil dan menyimpannya dalam susunan terpisah. Program kemudian memeriksa apakah angka ganjil ditemukan. Jika tidak, program akan mengeluarkan pesan terkait. Dalam hal ini, angka ganjil akan diurutkan dalam urutan menaik. Program kemudian menghitung median dari bilangan ganjil yang diurutkan dengan mengambil rata-rata jika bilangan ganjil, dan rata-rata dua bilangan tengah jika bilangan genap. Terakhir, program dengan jelas menampilkan angka ganjil yang ditemukan dan median yang dihitung.