

ALGORITMA PEMOGRAMAN 2

MODUL 9

"PENGURUTAN DATA"



Oleh:

NAMA: M. DAVI ILYAS RENALDO

NIM: 103112400062

KELAS: 12-IF-01

S1 TEKNIK INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

I.DASAR TEORI

12.1 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) 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.

12.4 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.

II. GUIDED

1.

```
package main
import (
    "fmt"
)
type Mahasiswa struct {
    Nama string
    Nilai int
}

func printData(mahasiswas []Mahasiswa) {
    for _, m := range mahasiswas {
        fmt.Printf("%s: %d", m.Nama, m.Nilai)
    }
    fmt.Println()
}

func selectionSort(mahasiswas []Mahasiswa) {
    n := len(mahasiswas)
    fmt.Println("\nProses Selection Sort:")
    for i := 0; i < n-1; i++ {
        minIdx := i
        for j := i + 1; j < n; j++ {
            if mahasiswas[j].Nilai < mahasiswas[minIdx].Nilai {
                minIdx = j
            }
        }
        mahasiswas[i], mahasiswas[minIdx] = mahasiswas[minIdx], mahasiswas[i]
        fmt.Printf("Iterasi ke-%d: ", i+1)
        printData(mahasiswas)
    }
}
```

```

}
func main() {
    data := []Mahasiswa{
        {"Budi", 75},
        {"Ani", 90},
        {"Dedi", 65},
        {"Citra", 85},
        {"Eka", 70},
    }
    fmt.Println("Data Sebelum Selection Sort:")
    printData(data)
    selectionSort(data)
    fmt.Println("\ndata setelah Selection Sort (Berdasarkan Nilai):")
    printData(data)
}

```

OUTPUT:

```

PS C:\Users\ACER\OneDrive\alpro2\modul1> go run "c:\Users\ACER\OneDrive\alpro2\modul1\GUIDED 9\1.go"
Data Sebelum Selection Sort:
Budi: 75 Ani: 90 Dedi: 65 Citra: 85 Eka: 70

Proses Selection Sort:
Iterasi ke-1: Dedi: 65 Ani: 90 Budi: 75 Citra: 85 Eka: 70
Iterasi ke-2: Dedi: 65 Eka: 70 Budi: 75 Citra: 85 Ani: 90
Iterasi ke-3: Dedi: 65 Eka: 70 Budi: 75 Citra: 85 Ani: 90
Iterasi ke-4: Dedi: 65 Eka: 70 Budi: 75 Citra: 85 Ani: 90

data setelah Selection Sort (Berdasarkan Nilai):
Dedi: 65 Eka: 70 Budi: 75 Citra: 85 Ani: 90
PS C:\Users\ACER\OneDrive\alpro2\modul1>

```

DESKRIPSI:

Program ini adalah program untuk mengurutkan data mahasiswa berdasarkan nilai mereka menggunakan algoritma selection sort. Terdapat struktur data Mahasiswa yang menyimpan nama dan nilai mahasiswa. Fungsi printData digunakan untuk mencetak daftar mahasiswa, sedangkan fungsi selectionSort menerapkan sorting dan mencetak setiap iterasi dari proses pengurutan. Di dalam fungsi main, program menginisialisasi data mahasiswa dan menampilkan data sebelum serta sesudah proses sorting, sehingga pengguna dapat melihat efektivitas algoritma dalam mengurutkan data.

2.

```
package main
import "fmt"
type Mahasiswa struct {
    Nama string
    Nilai int
}

func printData(mahasiswas []Mahasiswa) {
    for _, m := range mahasiswas {
        fmt.Printf("%s: %d ", m.Nama, m.Nilai)
    }
    fmt.Println()
}

func insertionSort(mahasiswas []Mahasiswa) {
    fmt.Println("\nProses Insertion Sort:")
    for i := 1; i < len(mahasiswas); i++ {
        key := mahasiswas[i]
        j := i - 1

        for j >= 0 && mahasiswas[j].Nilai > key.Nilai {
            mahasiswas[j+1] = mahasiswas[j]
            j--
        }
        mahasiswas[j+1] = key

        // Tampilkan kondisi array setelah iterasi
        fmt.Printf("Iterasi ke-%d: ", i)
        printData(mahasiswas)
    }
}

func main() {
    data := []Mahasiswa{
        {"Budi", 75},
        {"Ani", 90},
        {"Dedi", 65},
        {"Citra", 85},
        {"Eka", 70},
    }
}
```

```

    fmt.Println("Data sebelum Insertion Sort:")
    printData(data)

    insertionSort(data)

    fmt.Println("\nData setelah Insertion Sort (berdasarkan Nilai):")
    printData(data)
}

```

OUTPUT:

```

> go run "c:\Users\ACER\OneDrive\alpro2\modul1\GUIDED 9\2.go"
Data sebelum Insertion Sort:
Budi: 75 Ani: 90 Dedi: 65 Citra: 85 Eka: 70

Proses Insertion Sort:
Iterasi ke-1: Budi: 75 Ani: 90 Dedi: 65 Citra: 85 Eka: 70
Iterasi ke-2: Dedi: 65 Budi: 75 Ani: 90 Citra: 85 Eka: 70
Iterasi ke-3: Dedi: 65 Budi: 75 Citra: 85 Ani: 90 Eka: 70
Iterasi ke-4: Dedi: 65 Eka: 70 Budi: 75 Citra: 85 Ani: 90

Data setelah Insertion Sort (berdasarkan Nilai):
Dedi: 65 Eka: 70 Budi: 75 Citra: 85 Ani: 90
PS C:\Users\ACER\OneDrive\alpro2\modul1>

```

DESKRIPSI:

Program ini adalah program yang mengurutkan data nilai mahasiswa menggunakan algoritma insertionSort . Setiap mahasiswa nilainya ditampilkan dengan struct yang berisi nama da nilai. Program menampilkan data sebelum diurutkan, kemudian memproses pengurutan dengan metode tersebut dan menampilkan hasil setiap iterasi nya. Dan setelah selesai program tersebut menampilkan nilai mahasiswa yang sudah diurutkan dari yang terkecil hingga terbesar.

III. UNGUIDED

1.

```
// M. DAVI ILYAS RENALDO
package main
import "fmt"

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

func menghitungMedian(data []int) {
    n := len(data)
    var median int
    if n%2 == 0 {
        median = (data[n/2-1] + data[n/2]) / 2
    } else {
        median = data[n/2]
    }
    fmt.Println(median)
}

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

    for {
        fmt.Scan(&input)

        if input == -5313 {
            break
        }

        if input == 0 {
            selectionSort(arrData)
            menghitungMedian(arrData)
        } else {
            arrData = append(arrData, input)
        }
    }
}
```

OUTPUT:

```
PS C:\Users\ACER\OneDrive\alpro2\modul1> go run "c:\Users\ACER\OneDrive\alpro2\modul1\UNGUIDED 9\1.go"
7 23 11 0 5 19 2 29 3 13 17 0 -5313
11
12
PS C:\Users\ACER\OneDrive\alpro2\modul1> 
```

DESKRIPSI:

Program ini adalah program yang mengimplementasikan fungsi untuk mengurutkan data menggunakan algoritma selection sort dan menghitung median dari array. Pengguna diminta

untuk memasukkan bilangan bulat array hingga pengguna memasukkan angka min maka akan mengakhiri inputan. Dan jika memasukkan angka 0 maka program akan mengurutkan array dan mencetak median nilai yang telah diurutkan.

2.

```
// M. DAVI ILYAS RENALDO
package main
import "fmt"

func insertionSort(angka []int) {
    panjang := len(angka)
    for i := 1; i < panjang; i++ {
        kunci := angka[i]
        j := i - 1

        for j >= 0 && angka[j] > kunci {
            angka[j+1] = angka[j]
            j--
        }
        angka[j+1] = kunci
    }
}

func cekJarakTetap(angka []int) (bool, int) {
    if len(angka) < 2 {
        return true, 0
    }

    jarak := angka[1] - angka[0]
    for i := 2; i < len(angka); i++ {
        if angka[i]-angka[i-1] != jarak {
            return false, 0
        }
    }
    return true, jarak
}

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

    for {
        _, err := fmt.Scan(&masukan)
        if err != nil || masukan < 0 {
            break
        }
        daftarAngka = append(daftarAngka, masukan)
    }

    insertionSort(daftarAngka)

    for _, nilai := range daftarAngka {
        fmt.Print(nilai, " ")
    }
    fmt.Println()

    valid, jarak := cekJarakTetap(daftarAngka)
    if valid {
        fmt.Printf("Data berjarak %d\n", jarak)
    } else {
        fmt.Println("Data berjarak tidak tetap")
    }
}
```

OUTPUT:

```
PS C:\Users\ACER\OneDrive\alpro2\modul1> go run "c:\Users\ACER\OneDrive\alpro2\modul1\UNGUIDED 9\2.go"
31 13 25 43 1 7 19 37 -5
1 7 13 19 25 31 37 43
Data berjarak 6
PS C:\Users\ACER\OneDrive\alpro2\modul1> go run "c:\Users\ACER\OneDrive\alpro2\modul1\UNGUIDED 9\2.go"
4 40 14 8 26 1 38 2 32 -31
1 2 4 8 14 26 32 38 40
Data berjarak tidak tetap
PS C:\Users\ACER\OneDrive\alpro2\modul1> █
```

DESKRIPSI:

Program ini adalah program yang mengumpulkan bilangan bulat non integer. Kemudian program ini melakukan pengurutan dengan metode insertion sort. Setelah berhasil mengurutkan program ini akan memeriksa apakah jarak antar angka dalam daftar tersebut tetap atau tidak. Jika jarak antara angka dalam daftar tetap, maka program akan mencetak jarak tersebut dan jika tidak, maka program akan menyatakan bahwa jarak tidak tetap. Hasil akhir ditampilkan dengan urutan angka yang telah disortir.

IV.KESIMPULAN

V.REFERENSI

- modul 9 (Pengurutan Data) praktikum algoritma pemograman 2