

LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2

MODUL XII & XIII  
PENGURUTAN DATA



Disusun Oleh :

AFRIZAL DWI NUGRAHA / 2311102136

S1 IF 11 05

Dosen Pengampu :

Arif Amrulloh, S.Kom., M.Kom.

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

## I. DASAR TEORI

### 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

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

### 2. Algoritma Selection Sort

Adapun algoritma selection sort pada untuk mengurutkan array bertipe data bilangan bulat secara membesar atau ascending adalah sebagai berikut ini!

```
..    ...
5    type arrInt [4321]int
..    ...
15   func selectionSort1(T *arrInt, n int){
16   /* I.S. terdefinisi array T yang berisi n bilangan bulat
17      F.S. array T terurut secara ascending atau membesar dengan SELECTION SORT */
18      var t, i, j, idx_min int
19
```

```

20     i = 1
21     for i <= n-1 {
22         idx_min = i - 1
23         j = i
24         for j < n {
25             if T[idx_min] > T[j] {
26                 idx_min = j
27             }
28             j = j + 1
29         }
30         t = T[idx_min]
31         T[idx_min] = T[i-1]
32         T[i-1] = t
33         i = i + 1
34     }
35 }

```

Sama halnya apabila array yang akan diurutkan adalah bertipe data struct, maka tambahkan field pada saat proses perbandingan nilai ekstrim, kemudian tipe data dari variabel t sama dengan struct dari arraynya.

```

..     ...
5     type mahasiswa struct {
..         nama, nim, kelas, jurusan string
..         ipk float64
..     }
..     type arrMhs [2023]mahasiswa
..     ...
15    func selectionSort2(T * arrMhs, n int){
16        /* I.S. terdefinisi array T yang berisi n data mahasiswa
17           F.S. array T terurut secara ascending atau membesar berdasarkan ipk dengan
18           menggunakan algoritma SELECTION SORT */
19         var i, j, idx_min int
20         var t mahasiswa
21         i = 1
22         for i <= n-1 {
23             idx_min = i - 1
24             j = i
25             for j < n {
26                 if T[idx_min].ipk > T[j].ipk {
27                     idx_min = j
28                 }
29                 j = j + 1
30             }
31             t = T[idx_min]
32             T[idx_min] = T[i-1]
33             T[i-1] = t
34             i = i + 1
35         }
36     }

```

## II. GUIDED

1. 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.

Masukan dimulai dengan sebuah integer  $n$  ( $0 < n < 1000$ ), banyaknya daerah kerabat Hercules tinggal. Isi  $n$  baris berikutnya selalu dimulai dengan sebuah integer  $n$  ( $0 < m < 1000000$ ) yang menyatakan banyaknya rumah kerabat di daerah tersebut, diikuti dengan rangkaian bilangan bulat positif, nomor rumah para kerabat. Keluaran terdiri dari  $n$  baris, yaitu rangkaian rumah kerabatnya terurut membesar di masing-masing daerah.

Sourcecode

```
package main

import (
    "fmt"
)

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

func main() {
    var n int
    fmt.Print("Masukan jumlah daerah kerabat(n): ")
    fmt.Scanln(&n)

    for daerah := 1; daerah <= n; daerah++ {
        var m int
        fmt.Printf("Masukan jumlah nomor rumah kerabat %d: ", daerah)
        fmt.Scan(&m)

        arr := make([]int, m)
        fmt.Printf("Masukkan %d nomor rumah kerabat: ", m)
        for i := 0; i < m; i++ {
            fmt.Scan(&arr[i])
        }
        selectionSort(arr, m)
    }
}
```

```

        fmt.Printf("Nomor rumah terurut untuk daerah %d: ", daerah)
        for _, num := range arr {
            fmt.Printf("%d ", num)
        }
        fmt.Println()
    }
}

```

### Screenshoot Output

```

PS C:\Users\PC BRANDED\OneDrive\ドキュメント\Afriza\VS\new\Semester03> go run
temuan09\tempCodeRunnerFile.go
Masukan jumlah daerah kerabat(n): 3
Masukan jumlah nomor rumah kerabat 1: 6
Masukkan 6 nomor rumah kerabat: 5 2 1 7 9 13
Nomor rumah terurut untuk daerah 1: 1 2 5 7 9 13
Masukan jumlah nomor rumah kerabat 2: 7
Masukkan 7 nomor rumah kerabat: 6 189 15 27 39 75 133
Nomor rumah terurut untuk daerah 2: 6 15 27 39 75 133 189
Masukan jumlah nomor rumah kerabat 3: 4
Masukkan 4 nomor rumah kerabat: 3 4 9 1
Nomor rumah terurut untuk daerah 3: 1 3 4 9
PS C:\Users\PC BRANDED\OneDrive\ドキュメント\Afriza\VS\new\Semester03>

```

### Deskripsi Program

Untuk mengurutkan daftar nomor rumah kerabat di beberapa lokasi, program di atas menggunakan algoritma Sort Selection.

2. Buatlah sebuah program yang digunakan untuk membaca data integer seperti contoh yang diberikan di bawah ini, kemudian diurutkan (menggunakan metoda insertion sort), dan memeriksa apakah data yang terurut berjarak sama terhadap data sebelumnya.

Masukan terdiri dari sekumpulan bilangan bulat yang diakhiri oleh bilangan negatif. Hanya bilangan non negatif saja yang disimpan ke dalam array.

Keluaran terdiri dari dua baris. Baris pertama adalah isi dari array setelah dilakukan pengurutan, sedangkan baris kedua adalah status jarak setiap bilangan yang ada di dalam array. "Data berjarak x" atau "data berjarak tidak tetap".

Sourcecode

```
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 di temukan
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 di urutkan:")
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\PC BRANDED\OneDrive\ドキュメント\AfrizalVS\new\Semester03> go run
temuan09\tempCodeRunnerFile.go
Masukkan data integer (akhiri dengan bilangan negatif):
31 13 25 43 1 7 19 37 -5
Array setelah diurutkan:
1 7 13 19 25 31 37 43
Data berjarak 6
PS C:\Users\PC BRANDED\OneDrive\ドキュメント\AfrizalVS\new\Semester03> go run
temuan09\tempCodeRunnerFile.go
Masukkan data integer (akhiri dengan bilangan negatif):
4 40 14 8 26 1 38 2 32 -31
Array setelah diurutkan:
1 2 4 8 14 26 32 38 40
Data berjarak tidak tetap
PS C:\Users\PC BRANDED\OneDrive\ドキュメント\AfrizalVS\new\Semester03>

```

### Deskripsi Program

Program di atas membaca bilangan bulat hingga ditemukan bilangan negatif, mengurutkan bilangan tersebut dengan algoritma insertion sort, lalu memeriksa apakah jarak antar elemen di array yang sudah diurutkan konsisten. Hasil akhirnya menampilkan array yang sudah diurutkan dan status jaraknya.



### III. UNGUIDED

1. Belakangan diketahui ternyata Hercules itu tidak berani menyeberang jalan, maka selalu diusahakan agar hanya menyeberang jalan sesedikit mungkin, hanya diujung jalan. Karena nomor rumah sisi kiri jalan selalu ganjil dan sisi kanan jalan selalu genap, maka buatlah program kerabat dekat yang akan menampilkan nomor rumah mulai dari nomor yang ganjil lebih dulu terurut membesar dan kemudian menampilkan nomor rumah dengan nomor genap terurut mengecil. Format Masukan masih persis sama seperti sebelumnya.

Keluaran terdiri dari n baris, yaitu rangkaian rumah kerabatnya terurut membesar untuk nomor ganjil, diikuti dengan terurut mengecil untuk nomor genap, di masing-masing daerah.

No	Masukan	Keluaran
1	3 5 2 1 7 9 13 6 189 15 27 39 75 133 3 4 9 1	1 13 12 8 2 15 27 39 75 133 189 8 4 2

Keterangan: Terdapat 3 daerah dalam contoh masukan. Baris kedua berisi campuran bilangan ganjil dan genap. Baris berikutnya hanya berisi bilangan ganjil, dan baris terakhir hanya berisibilangan genap.

Petunjuk:

- Waktu pembacaan data, bilangan ganjil dan genap dipisahkan ke dalam dua array yang berbeda, untuk kemudian masing-masing diurutkan tersendiri.
- Atau, tetap disimpan dalam satu array, diurutkan secara keseluruhan. Tetapi pada waktu pencetakan, mulai dengan mencetak semua nilai ganjil lebih dulu, kemudian setelah selesai cetaklah semua nilai genapnya.

Sourcecode

```
package main

import (
    "fmt"
    "sort"
)

func main() {
    var n int
    fmt.Print("Masukan jumlah daerah kerabat (n): ")
    fmt.Scanln(&n)

    for daerah := 1; daerah <= n; daerah++ {
        var m int
        fmt.Printf("Masukan jumlah nomor rumah kerabat %d: ", daerah)
        fmt.Scan(&m)

        arr := make([]int, m)
        odd := []int{}
        even := []int{}
```

```

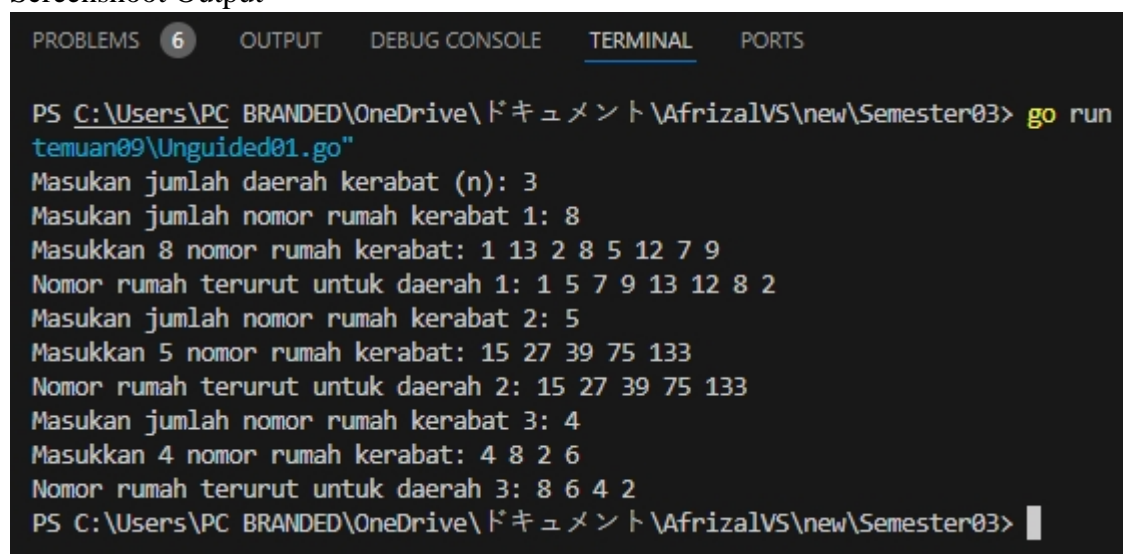
    fmt.Printf("Masukkan %d nomor rumah kerabat: ", m)
    for i := 0; i < m; i++ {
        fmt.Scan(&arr[i])
        // Pisahkan bilangan ganjil dan genap
        if arr[i]%2 == 0 {
            even = append(even, arr[i])
        } else {
            odd = append(odd, arr[i])
        }
    }

    // Urutkan bilangan ganjil membesar
    sort.Ints(odd)
    // Urutkan bilangan genap mengecil
    sort.Sort(sort.Reverse(sort.IntSlice(even)))

    // Cetak hasil
    fmt.Printf("Nomor rumah terurut untuk daerah %d: ", daerah)
    for _, num := range odd {
        fmt.Printf("%d ", num)
    }
    for _, num := range even {
        fmt.Printf("%d ", num)
    }
    fmt.Println()
}
}

```

### Screenshoot Output



```

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\PC BRANDED\OneDrive\ドキュメント\AfrizalVS\new\Semester03> go run
temuan09\Unguided01.go
Masukan jumlah daerah kerabat (n): 3
Masukan jumlah nomor rumah kerabat 1: 8
Masukkan 8 nomor rumah kerabat: 1 13 2 8 5 12 7 9
Nomor rumah terurut untuk daerah 1: 1 5 7 9 13 12 8 2
Masukan jumlah nomor rumah kerabat 2: 5
Masukkan 5 nomor rumah kerabat: 15 27 39 75 133
Nomor rumah terurut untuk daerah 2: 15 27 39 75 133
Masukan jumlah nomor rumah kerabat 3: 4
Masukkan 4 nomor rumah kerabat: 4 8 2 6
Nomor rumah terurut untuk daerah 3: 8 6 4 2
PS C:\Users\PC BRANDED\OneDrive\ドキュメント\AfrizalVS\new\Semester03>

```

### Deskripsi Program

Program ini bertujuan untuk mengurutkan nomor rumah anggota keluarga di daerah tertentu sesuai dengan aturan khusus. Nomor rumah terbagi menjadi dua kategori, ganjil dan genap, dan masing-masing diurutkan secara terpisah sesuai dengan aturan berikut:

Nomor ganjil diurutkan dari atas ke bawah, sedangkan nomor genap diurutkan dari atas ke bawah. Output: Nomor ganjil ditampilkan terlebih dahulu, dan nomor genap kemudian.

2. Kompetisi pemrograman yang baru saja berlalu diikuti oleh 17 tim dari berbagai perguruan tinggi ternama. Dalam kompetisi tersebut, setiap tim berlomba untuk menyelesaikan sebanyak mungkin problem yang diberikan. Dari 13 problem yang diberikan, ada satu problem yang menarik. Problem tersebut mudah dipahami, hampir semua tim mencoba untuk menyelesaikannya, tetapi hanya 3 tim yang berhasil. Apa sih problemnya? "Median adalah nilai tengah dari suatu koleksi data yang sudah terurut. Jika jumlah data genap, maka nilai median adalah rerata dari kedua nilai tengahnya. Pada problem ini, semua data merupakan bilangan bulat positif, dan karenanya rerata nilai tengah dibulatkan ke bawah."

Buatlah program median yang mencetak nilai median terhadap seluruh data yang sudah terbaca, jika data yang dibaca saat itu adalah 0.

**Masukan** berbentuk rangkaian bilangan bulat. Masukan tidak akan berisi lebih dari 1000000 data, tidak termasuk bilangan 0. Data 0 merupakan tanda bahwa median harus dicetak, tidak termasuk data yang dicari mediannya. Data masukan diakhiri dengan bilangan bulat -5313.

**Keluaran** adalah median yang diminta, satu data per baris.

No	Masukan	Keluaran
1	7 23 11 0 5 19 2 29 3 13 17 0 -5313	11 12

**Keterangan:**

Sampai bilangan 0 yang pertama, data terbaca adalah 7 23 11, setelah tersusun: 7 11 23, maka median saat itu adalah 11. Sampai bilangan 0 yang kedua, data adalah 7 23 11 5 19 2 29 3 13 17, setelah tersusun diperoleh: 2 3 5 7 11 13 17 19 23 29. Karena ada 10 data, genap, maka median adalah  $(11+13)/2=12$ .

**Petunjuk:**

Untuk setiap data bukan 0 (dan bukan marker -5313541) simpan ke dalam array, Dan setiap kali menemukan bilangan 0, urutkanlah data yang sudah tersimpan dengan menggunakan metode insertion sort dan ambil mediannya.

**Sourcecode**

```
package main

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

func main() {
    var data []int

    // Membaca input dari stdin
    scanner := bufio.NewScanner(os.Stdin)
```

```

for scanner.Scan() {
    input := scanner.Text()
    num, err := strconv.Atoi(input)
    if err != nil {
        fmt.Println("Input tidak valid")
        continue
    }

    // Jika angka -5313 ditemukan, hentikan program
    if num == -5313 {
        break
    }

    // Jika angka 0 ditemukan, hitung dan cetak median
    if num == 0 {
        if len(data) == 0 {
            fmt.Println("Tidak ada data untuk menghitung median")
            continue
        }

        // Mengurutkan data dengan QuickSort
        sort.Ints(data)
        median := CalculateMedian(data)
        fmt.Printf("Mediannya adalah %d\n", median)
    } else {
        // Menambahkan data baru ke array
        data = append(data, num)
    }
}

// Fungsi untuk menghitung median
func CalculateMedian(arr []int) int {
    length := len(arr)
    if length%2 == 1 {
        // Jika jumlah data ganjil, ambil nilai tengah
        return arr[length/2]
    } else {
        // Jika jumlah data genap, ambil rata-rata dua nilai tengah
        // dan bulatkan ke bawah
        sum := arr[length/2-1] + arr[length/2]
        return sum / 2
    }
}

```

```
PROBLEMS 7 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\PC BRANDED\OneDrive\ドキュメント\Afriza\VS\new\Semester03> go run
temuan09\Unguided02.go"
7
23
11
0
Mediannya adalah 11
5
19
2
29
3
13
17
0
Mediannya adalah 12
█
```

### Screenshoot Output

#### Deskripsi Program

Program ini adalah aplikasi sederhana yang memungkinkan pengguna memasukkan serangkaian angka, menghitung median dari angka-angka tersebut, dan menampilkan hasilnya. Program beroperasi dengan membaca input pengguna, menyimpan angka-angka tersebut dalam sebuah list, dan menghitung median ketika ada angka 0.

3. Sebuah program perpustakaan digunakan untuk mengelola data buku di dalam suatu perpustakaan. Misalnya terdefinisi struct dan array seperti berikut ini:

```
const nMax : integer = 7919
type Buku = <
    id, judul, penulis, penerbit : string
    eksemplar, tahun, rating : integer >

type DaftarBuku = array [ 1..nMax] of Buku
Pustaka : DaftarBuku
nPustaka: integer
```

**Masukan** terdiri dari beberapa baris. Baris pertama adalah bilangan bulat N yang menyatakan banyaknya data buku yang ada di dalam perpustakaan. N baris berikutnya, masing-masingnya adalah data buku sesuai dengan atribut atau field pada struct. Baris terakhir adalah bilangan bulat yang menyatakan rating buku yang akan dicari.

**Keluaran** terdiri dari beberapa baris. Baris pertama adalah data buku terfavorit, baris kedua adalah lima judul buku dengan rating tertinggi, selanjutnya baris terakhir adalah data buku yang dicari sesuai rating yang diberikan pada masukan baris terakhir.

Lengkapi subprogram-subprogram dibawah ini, sesuai dengan I.S. dan F.S yang diberikan.

```

procedure DaftarkanBuku(in/out pustaka : DaftarBuku, n : integer)
{I.S. sejumlah n data buku telah siap para piranti masukan
 F.S. n berisi sebuah nilai, dan pustaka berisi sejumlah n data buku}

procedure CetakTerfavorit(in pustaka : DaftarBuku, in n : integer)
{I.S. array pustaka berisi n buah data buku dan belum terurut
 F.S. Tampilan data buku (judul, penulis, penerbit, tahun)
terfavorit, yaitu memiliki rating tertinggi}

procedure UrutBuku( in/out pustaka : DaftarBuku, n : integer )
{I.S. Array pustaka berisi n data buku
 F.S. Array pustaka terurut menurun/mengecil terhadap rating.
Catatan: Gunakan metoda Insertion sort}

procedure Cetak5Terbaru( in pustaka : DaftarBuku, n integer )
{I.S. pustaka berisi n data buku yang sudah terurut menurut rating
 F.S. Laporan 5 judul buku dengan rating tertinggi
 Catatan: Isi pustaka mungkin saja kurang dari 5}

procedure CariBuku(in pustaka : DaftarBuku, n : integer, r : integer )
{I.S. pustaka berisi n data buku yang sudah terurut menurut rating
 F.S. Laporan salah satu buku (judul, penulis, penerbit, tahun,
eksemplar, rating) dengan rating yang diberikan. Jika tidak ada buku
dengan rating yang ditanyakan, cukup tuliskan "Tidak ada buku dengan
rating seperti itu". Catatan: Gunakan pencarian biner/belah dua.}

```

## Sourcecode

```

package main

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

func main() {
    var data []int

    // Membaca input dari stdin
    scanner := bufio.NewScanner(os.Stdin)
    for scanner.Scan() {
        input := scanner.Text()
        num, err := strconv.Atoi(input)
        if err != nil {
            fmt.Println("Input tidak valid")
            continue
        }
    }
}

```



```
}

// Jika angka -5313 ditemukan, hentikan program
if num == -5313 {
    break
}

// Jika angka 0 ditemukan, hitung dan cetak median
if num == 0 {
    if len(data) == 0 {
        fmt.Println("Tidak ada data untuk menghitung median")
        continue
    }
}
```

PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\PC BRANDED\OneDrive\ドキュメント\AfrizalVS\new\Semester03> go run
temuan09\Unguided03.go"
Masukkan jumlah buku: 2
Masukkan data buku (ID, Judul, Penulis, Penerbit, Eksemplar, Tahun, Rating):
101 Dune Frank_Herbert Ace_Books 10 1965 9
Masukkan data buku (ID, Judul, Penulis, Penerbit, Eksemplar, Tahun, Rating):
102 The_Hobbit J.R.R._Tolkien HarperCollins 15 1937 8
Buku dengan rating tertinggi:
- Dune (Frank_Herbert), Ace_Books, 1965, Rating: 9
5 Buku dengan rating tertinggi:
1. Dune - Rating: 9
2. The_Hobbit - Rating: 8
Masukkan rating buku yang ingin dicari: 8
Buku ditemukan:
- The_Hobbit (J.R.R._Tolkien), HarperCollins, 1937, Rating: 8
PS C:\Users\PC BRANDED\OneDrive\ドキュメント\AfrizalVS\new\Semester03> 
```

```
} else {
    // Jika jumlah data genap, ambil rata-rata dua nilai tengah
    dan bulatkan ke bawah
    sum := arr[length/2-1] + arr[length/2]
    return sum / 2
}
}
```

### Screenshoot Output

### Deskripsi Program

Aplikasi sederhana untuk manajemen perpustakaan ini memungkinkan pengguna mengelola daftar buku dengan fitur seperti penambahan data buku, pencarian buku berdasarkan rating, dan pencetakan daftar buku sesuai dengan kriteria tertentu.

### **Daftar Pustaka :**

1. MODUL 12&13. PENGURUTAN DATA