

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL XII & XIII  
PENGURUTAN DATA**



**Disusun Oleh :**

**ALTHAF TEGAR SOFYAN / 2311102217**

**Kelas 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

## 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 dan data dengan indeks kecil di kiri dan indeks besar ada di kanan.

1. Cari nilai terkecil didalam 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 ← 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	}
U	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	}

## 12.2 Algoritma Selection Sort

Adapun algoritma selection sort pada untuk mengurutkan array bertipe data bilangan bulat secara membesar atau ascending. Sama halnya apabila array akan diurutkan adalah bertipe data struct, maka tambahkan field pada saat proses perbandingan nilai ekstrim, kemudian tipa data dari variable t sama dengan struct dari arraynya.

---

```

type mahasiswa struct (
    nama, nim, kelas, jurusan string
    ipk float64

type arrMhs [2023)mahasiswa

func selectionsort2(T * arrPlhs, n int)(
    /* Z.S. terdefinisi array T yang berisi n data mahasiswa
       F.S. array T terurut secara ascending atau nilai besar berdasarkan ipk dengan
       menggunakan algoritma Selection Sort */

    var i, j, idx_min int
    var t mahasiswa

    for i = 1 to n-1 {
        idx_min = i
        for j = i+1 to n {
            if T[idx_min].ipk > T[j].ipk {
                idx_min = j
            }
        }
        t = T[idx_min]
        T[idx_min] = T[i]
        T[i] = t
        i = i + 1
    }
}

```

## 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  $m$  ( $0 < m < 1000000$ ) 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.

No	Hasukan	Keluaran
1	3	1 7 9 13
	5 21 7 9 13	2
	6 1 89 1 27 75 133	15 2 3 75 133 189
	5 39	7 9
	3 4 1	1 9
	9	4

Keterangan: Terdapat 3 daerah dalam contoh input, dan di masing-masing daerah mempunyai 5, 6, dan 3 kerabat.

### 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()
    }
}

```

## Screenshoot Output

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\vscode 1> go run "c:\vscode 1\Golang\laprak 12\guided 1.go"
Masukkan jumlah daerah kerabat (n): 3

Masukkan jumlah nomor rumah kerabat untuk daerah 1: 5
Masukkan 5 nomor rumah kerabat: 2 1 7 9 13
Nomor rumah terurut untuk daerah 1: 1 2 7 9 13

Masukkan jumlah nomor rumah kerabat untuk daerah 2: 3
Masukkan 3 nomor rumah kerabat: 23 11 12
Nomor rumah terurut untuk daerah 2: 11 12 23

Masukkan jumlah nomor rumah kerabat untuk daerah 3: 4
Masukkan 4 nomor rumah kerabat: 12 23 22 31
Nomor rumah terurut untuk daerah 3: 12 22 23 31
PS C:\vscode 1> █
```

**Deskripsi Program:** Program diatas untuk mengurutkan nomor rumah kerabat pada beberapa daerah menggunakan algoritma Selection Sort. Program akan meminta pengguna untuk memasukkan jumlah daerah yang akan diurutkan. Kemudian untuk tiap daerah tersebut, pengguna akan diminta untuk memasukkan nomor rumah yang akan diurutkan. Setelah itu, program akan menggunakan Selection Sort untuk mengurutkan nomor rumah dari terkecil ke terbesar.

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". Contoh masukan dan keluaran

No	Masukan	Keluaran
1	31 13 25 43 1 7 19 37 -5	1 7 13 19 25 31 37 43 Data berjarak 6
2	4 40 14 8 26 1 38 2 32 -31	1 2 4 8 14 26 32 38 40 Data berjarak tidak tetap

## 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")
}
}
}

```

### Screenshoot Output

```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\vscode 1> go run "c:\vscode 1\Golang\laprak 12\guided 2.go"
Masukkan data integer (akhiri dengan bilangan negatif):
15 25 10 35 5 20 30 -5
Array setelah diurutkan:
5 10 15 20 25 30 35
Data berjarak 5
PS C:\vscode 1>

```

**Deskripsi Program:** Program diatas untuk mengurutkan menggunakan algoritma Insertion Sort dan untuk pemeriksaan apakah selisih antar elemen array konstan. Pengguna akan diminta memasukkan deret bilangan integer, diakhiri dengan bilangan negative sebagai tanda akhir input. Data yang dimasukkan diurutkan secara menaik, kemudian program memeriksa apakah selisih antar elemen dalam array yang telah diurutkan tetap. Hasil pengurutan dan status konsistensi selisih elemen akan ditampilkan, termasuk selisih konstanta jika ada atau akan mendapat pesan bahwa selisih tidak tetap jika tidak konsisten.



### III. UNGUIDED

1. Buatlah program menggunakan bahasa pemrograman golang dari pernyataan soal berikut ini. 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	Hasukan	Keluaran
1	3 5 2 7 9 13 1 6 189 1 27 75 133 5 39 3 4 1 9	1 13 12 8 2 1 2 3 7 133 189 5 7 9 5 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 berisi bilangan 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.

## Source Code

```
package main

//Althaf Tegar 2311102217 IF06

import (
    "fmt"
    "sort"
)

func main() {
    var n217 int
    fmt.Print("Masukkan jumlah daerah: ")
    fmt.Scan(&n217)

    for daerah_n217 := 1; daerah_n217 <= n217; daerah_n217++ {
        var m17 int
        fmt.Printf("\nMasukkan jumlah nomor rumah untuk daerah
%d: ", daerah_n217)
        fmt.Scan(&m17)

        arr := make([]int, m17)
        fmt.Printf("Masukkan %d nomor rumah: ", m17)
        for i := 0; i < m17; i++ {
            fmt.Scan(&arr[i])
        }

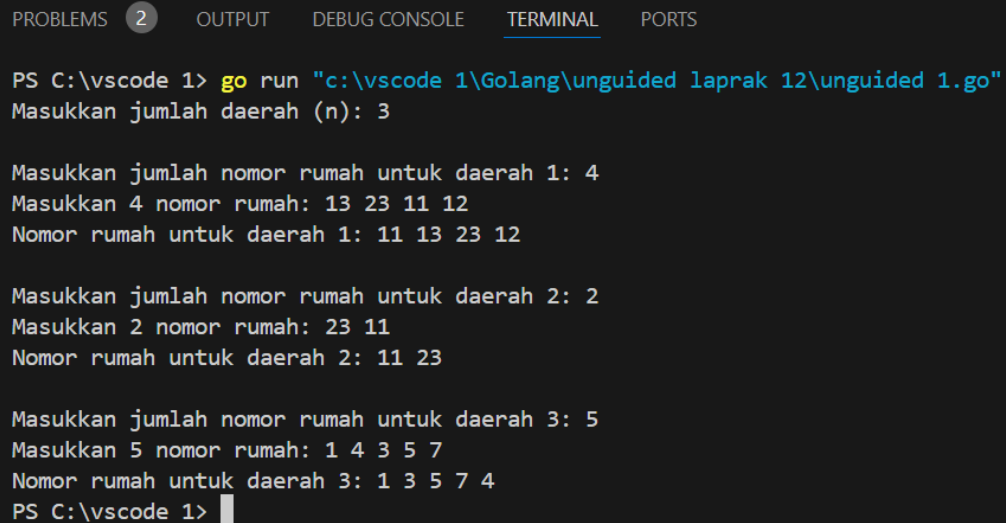
        var ganjil, genap []int

        sort.Ints(ganjil)
        sort.Sort(sort.Reverse(sort.IntSlice(genap)))

        // Pisahkan bilangan ganjil dan genap
        for _, num := range arr {
            if num%2 == 0 {
                genap = append(genap, num)
            } else {
                ganjil = append(ganjil, num)
            }
        }

        fmt.Printf("Nomor rumah untuk daerah %d: ",
daerah_n217)
        for _, num := range ganjil {
            fmt.Printf("%d ", num)
        }
        for _, num := range genap {
            fmt.Printf("%d ", num)
        }
        fmt.Println()
    }
}
```

## Screenshoot Output



The screenshot shows a terminal window with the following text:

```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\vscode 1> go run "c:\vscode 1\Golang\unguided laprak 12\unguided 1.go"
Masukkan jumlah daerah (n): 3

Masukkan jumlah nomor rumah untuk daerah 1: 4
Masukkan 4 nomor rumah: 13 23 11 12
Nomor rumah untuk daerah 1: 11 13 23 12

Masukkan jumlah nomor rumah untuk daerah 2: 2
Masukkan 2 nomor rumah: 23 11
Nomor rumah untuk daerah 2: 11 23

Masukkan jumlah nomor rumah untuk daerah 3: 5
Masukkan 5 nomor rumah: 1 4 3 5 7
Nomor rumah untuk daerah 3: 1 3 5 7 4
PS C:\vscode 1> 
```

**Deskripsi Program:** Program diatas bertujuan untuk mengelola data nomor rumah dibeberapa daerah. Program akan meminta pengguna untuk memasukkan jumlah daerah dan daftar nomor rumah untuk masing masing daerah. Setiap nomor rumah akan dipisahkan menjadi dua kelompok yaitu bilangan ganjil dan bilangan genap. Nomor ganjil akan diurutkan dalam urutaan ke atas dan nomor genap akan di urutkan ke bawah. Kemudian hasilnya akan ditampilkan dengan format, nomor ganjil terlebih dahulu lalu di ikuti nomor genap untuk tiap daerahnya. Pada program ini menggunakan fungsi `sort.ascending` untuk pengurutan naik dan `sort.descending` untuk pengurutan kebawah.

2. Buatlah Sebuah program 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* *bolesi* 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	Ilasukan	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.

## Source Code

```
package main
// Althaf Tegar 2311102217 IF 06

import "fmt"

const maxSize = 1000000

func sortArray(data2217 []int, ukuran int) {
    for start2 := 0; start2 < ukuran-1; start2++ {
        smallest := start2
        for check := start2 + 1; check < ukuran; check++ {
            if data2217[check] < data2217[smallest] {
                smallest = check
            }
        }
        data2217[start2], data2217[smallest] = data2217[smallest],
data2217[start2]
    }
}

func hitungMedian217(data []int, size int) float64 {
    mid := size / 2
    if size%2 == 0 {
        return float64(data[mid-1]+data[mid]) / 2.0
    }
    return float64(data[mid])
}

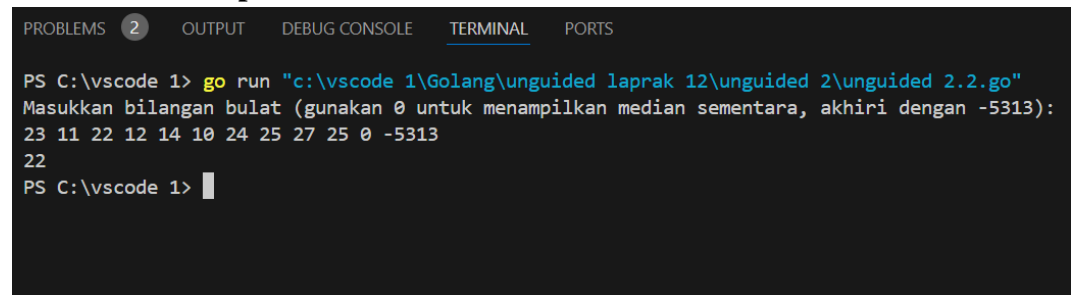
func main() {
    var values [maxSize]int
    var sorted [maxSize]int
    var count int
    var input int

    fmt.Println("Masukkan bilangan bulat (gunakan 0 untuk menampilkan
median sementara, akhiri dengan -5313):")
    for {
        fmt.Scan(&input)
        if input == -5313 {
            break
        }

        if input != 0 {
            values[count] = input
            count++
        } else if count > 0 {
            for i := 0; i < count; i++ {
                sorted[i] = values[i]
            }

            sortArray(sorted[:], count)
            median := hitungMedian217(sorted[:], count)
            fmt.Printf("%.0f\n", median)
        }
    }
}
```

## Screenshoot Output



```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\vscode 1> go run "c:\vscode 1\Golang\unguided laprak 12\unguided 2\unguided 2.2.go"
Masukkan bilangan bulat (gunakan 0 untuk menampilkan median sementara, akhiri dengan -5313):
23 11 22 12 14 10 24 25 27 25 0 -5313
22
PS C:\vscode 1> 
```

**Deskripsi Program:** Program tersebut menerima bilangan bulat untuk menampilkan bilai median Ketika angka 0 dimasukkan dan input akan dihentikan Ketika terdapat angka -5313 dimasukkan. Program menggunakan array values untuk menyimpan semua bilangan yang dimasukkan oleh pengguna. Ketika angka 0 dimasukkan, array values akan disalin ke array sorted, yang kemudian diurutkan menggunakan fungsi `sortArray`. Terdapat fungsi `hitungMedian217` yang digunakan untuk menghitung median dari array yang telah di urutkan. Median dihihtung berdasarkan jumlah elemen dalam array: jika jumlah elemen ganjil, median adalah elemen tengah, sedangkan jika genap, median dihitung sebagai rata-rata dua elemen tengah. Program menampilkan median dalam format bilangan bulat setiap kali angka 0 dimasukkan.

3. 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.}
```

## Source Code

```
package main
package main

// Althaf Tegar 2311102217 IF 06

import "fmt"

const kapasitasMaks = 7919

type Buku struct {
    ID, Judul, Penulis, Penerbit    string
    Eksemplar, TahunTerbit, Rating int
}

type KoleksiBuku2217 [kapasitasMaks]Buku
```

```

func TambahBuku(perpustakaan *KoleksiBuku2217, jumlahBuku
*int) {
    fmt.Print("Masukkan ID Buku: ")
    fmt.Scanln(&perpustakaan[*jumlahBuku].ID)
    fmt.Print("Masukkan Judul Buku: ")
    fmt.Scanln(&perpustakaan[*jumlahBuku].Judul)
    fmt.Print("Masukkan Nama Penulis: ")
    fmt.Scanln(&perpustakaan[*jumlahBuku].Penulis)
    fmt.Print("Masukkan Nama Penerbit: ")
    fmt.Scanln(&perpustakaan[*jumlahBuku].Penerbit)
    fmt.Print("Masukkan Jumlah Eksemplar: ")
    fmt.Scanln(&perpustakaan[*jumlahBuku].Eksemplar)
    fmt.Print("Masukkan Tahun Terbit: ")
    fmt.Scanln(&perpustakaan[*jumlahBuku].TahunTerbit)
    fmt.Print("Masukkan Rating Buku (1-10): ")
    fmt.Scanln(&perpustakaan[*jumlahBuku].Rating)
    *jumlahBuku++
}

func TampilkanBukuTerfavorit(perpustakaan
KoleksiBuku2217, totalBuku int) {
    fmt.Println("\nBuku Terfavorit:")
    ratingTertinggi := 0
    for i := 0; i < totalBuku; i++ {
        if perpustakaan[i].Rating > ratingTertinggi {
            ratingTertinggi = perpustakaan[i].Rating
        }
    }

    for i := 0; i < totalBuku; i++ {
        if perpustakaan[i].Rating == ratingTertinggi {
            fmt.Printf("Judul: %s, Penulis: %s, Penerbit:
%s, Tahun: %d, Rating: %d\n",
                perpustakaan[i].Judul,
                perpustakaan[i].Penulis, perpustakaan[i].Penerbit,
                perpustakaan[i].TahunTerbit, perpustakaan[i].Rating)
        }
    }
}

func UrutkanBuku(perpustakaan *KoleksiBuku2217, totalBuku
int) {
    for i := 1; i < totalBuku; i++ {
        bukuSementara := (*perpustakaan)[i]
        j := i - 1
        for j >= 0 && (*perpustakaan)[j].Rating <
bukuSementara.Rating {
            (*perpustakaan)[j+1] = (*perpustakaan)[j]
            j--
        }
        (*perpustakaan)[j+1] = bukuSementara
    }
}

func Tampilkan5BukuTerbaik(perpustakaan KoleksiBuku2217,
totalBuku int) {
    fmt.Println("5 Buku dengan Rating Tertinggi:")
    for i := 0; i < totalBuku && i < 5; i++ {
        fmt.Println(perpustakaan[i].Judul)
    }
}

```



```

    }
}

func CariBerdasarkanRating(perpustakaan KoleksiBuku2217,
totalBuku, ratingDicari int) {
    fmt.Printf("\nBuku dengan Rating %d:\n",
ratingDicari)
    ditemukan := false
    for i := 0; i < totalBuku; i++ {
        if perpustakaan[i].Rating == ratingDicari {
            fmt.Printf("Judul: %s, Penulis: %s, Penerbit:
%s, Tahun: %d, Rating: %d\n",
                perpustakaan[i].Judul,
perpustakaan[i].Penulis, perpustakaan[i].Penerbit,
perpustakaan[i].TahunTerbit, perpustakaan[i].Rating)
            ditemukan = true
        }
    }
    if !ditemukan {
        fmt.Println("Tidak ada buku dengan rating
tersebut.")
    }
}

func main() {
    var perpustakaan KoleksiBuku2217
    var jumlahBuku, totalBuku, ratingDicari int

    fmt.Print("Masukkan jumlah buku: ")
    fmt.Scanln(&totalBuku)

    fmt.Println("\nInput data buku:")
    for i := 0; i < totalBuku; i++ {
        fmt.Printf("\nBuku ke-%d:\n", i+1)
        TambahBuku(&perpustakaan, &jumlahBuku)
    }

    TampilkanBukuTerfavorit(perpustakaan, totalBuku)

    fmt.Println()
    UrutkanBuku(&perpustakaan, totalBuku)
    Tampilkan5BukuTerbaik(perpustakaan, totalBuku)

    fmt.Print("\nMasukkan rating untuk pencarian: ")
    fmt.Scanln(&ratingDicari)
    CariBerdasarkanRating(perpustakaan, totalBuku,
ratingDicari)
}

```

## Screenshoot Output

```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\vscode 1> go run "c:\vscode 1\Golang\unguided laprak 12\unguided 3\unguided 3.go"
Masukkan jumlah buku: 4

Input data buku:

Buku ke-1:
Masukkan ID Buku: 2217
Masukkan Judul Buku: CINTA
Masukkan Nama Penulis: ALTHAF
Masukkan Nama Penerbit: ANDROMEDA
Masukkan Jumlah Eksemplar: 45
Masukkan Tahun Terbit: 2020
Masukkan Rating Buku (1-10): 8

Buku ke-2:
Masukkan ID Buku: 2218
Masukkan Judul Buku: PUPUS
Masukkan Nama Penulis: NOVARIA
Masukkan Nama Penerbit: ANDIANI
Masukkan Jumlah Eksemplar: 50
Masukkan Tahun Terbit: 2021
Masukkan Rating Buku (1-10): 9

Buku ke-3:
Masukkan ID Buku: 2219
Masukkan Judul Buku: BREAK
Masukkan Nama Penulis: RENNA
Masukkan Nama Penerbit: LAPOSENA
Masukkan Jumlah Eksemplar: 65
Masukkan Tahun Terbit: 2022
Masukkan Rating Buku (1-10): 7

Buku ke-4:
Masukkan ID Buku: 2220
Masukkan Judul Buku: HILANG
Masukkan Nama Penulis: LENARIA
Masukkan Nama Penerbit: SUBITANIA
Masukkan Jumlah Eksemplar: 60
Masukkan Tahun Terbit: 2023
Masukkan Rating Buku (1-10): 10

Buku Terfavorit:
Judul: HILANG, Penulis: LENARIA, Penerbit: SUBITANIA, Tahun: 2023, Rating: 10

5 Buku dengan Rating Tertinggi:
HILANG
PUPUS
CINTA
BREAK

Masukkan rating untuk pencarian: 9

Buku dengan Rating 9:
Judul: PUPUS, Penulis: NOVARIA, Penerbit: ANDIANI, Tahun: 2021, Rating: 9
PS C:\vscode 1>
```

**Deskripsi Program:** Program ini berfungsi untuk manajemen perpustakaan sederhana yang memungkinkan pengguna untuk mendata berdasarkan rating. Program dimulai dengan pengguna menentukan jumlah buku yang akan didaftarkan. Setiap buku memiliki data seperti ID, judul, penulis, penerbit, jumlah eksemplar, tahun terbit, dan rating. Setelah

data dimasukkan, program menampilkan buku dengan rating tertinggi (buku favorit). Buku-buku kemudian diurutkan berdasarkan rating dalam urutan dari terfavorit hingga kurang favorit, dan program menampilkan hingga lima buku dengan rating tertinggi. Selain itu, pengguna dapat mencari buku berdasarkan rating tertentu, dan program akan menampilkan semua buku dengan rating yang sesuai atau memberikan notifikasi jika tidak ditemukan. Program menggunakan array tetap untuk menyimpan data buku dan memanfaatkan berbagai fungsi untuk memproses dan menampilkan informasi secara terorganisasi.