LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2

MODUL 12&13

PENGURUTAN DATA

Disusun Oleh :

PRIESTY AMEILIANA MAULIDAH / 2311102175

IF-11-05

Dosen Pengampu :

ARIF AMRULLOH, S.KOM.,M.KOM

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

1.) DASAR TEORI

1. ide algoritma selection

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.

2. 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.

2.) GUIDED

Soal Studi Case

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 masingmasing daerah.

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

Sourcecode

guided 1.go modul 14 sorting 2 | guided 2.go modul 14 sorting 3 | unguided array 3.go | assegment 1.go

modul 14 sorting > guided 1.go > main

```go
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++ {
            // mencari 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 // gunakan m untuk jumlah nomor rumah di daerah ini
        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 56 | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS

```
PS D:\1-Priesty AM\alpro 3 semester 3> go run "d:\1-Priesty AM\alpro 3 semester 3\modul 14 sorting\guided1.go"
masukkan jumlah daerah kerabat (n): 3

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

masukkan jumlah nomor rumah kerabat untuk daerah 2: 6 189 15 27 39 75 133
masukkan 6 nomor rumah kerabat: nomor rumah terurut untuk daerah 2: 15 27 39 75 133 189

masukkan jumlah nomor rumah kerabat untuk daerah 3: 3 4 9 1
masukkan 3 nomor rumah kerabat: nomor rumah terurut untuk daerah 3: 1 4 9
PS D:\1-Priesty AM\alpro 3 semester 3>
```

Deskripsi Program

Program ini mengurutkan nomor rumah kerabat di beberapa daerah menggunakan algoritma selection sort. Pengguna memasukkan jumlah daerah (n),lalu untuk setiap daerah, jumlah nomor rumah (m) dan daftar nomor rumahnya. Nomor rumah diurutkan dari kecil ke besar menggunakan selection sort, dengan elemen terkecil dipindahkan ke posisi awal secara bertahap.

Setelah selesai, program menampilkan nomor rumah yang telah terurut untuk setiap daerah. Contoh: Jika n = 3, untuk input 5 2 1 7 9 13, hasilnya 1 2 7 9 13. Untuk input 189 15 27 39 75 133, hasilnya 15 27 39 75 133 189. Semua hasil ditampilkan terurut sesuai daerah.

GUIDED

Soal Studi Case

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

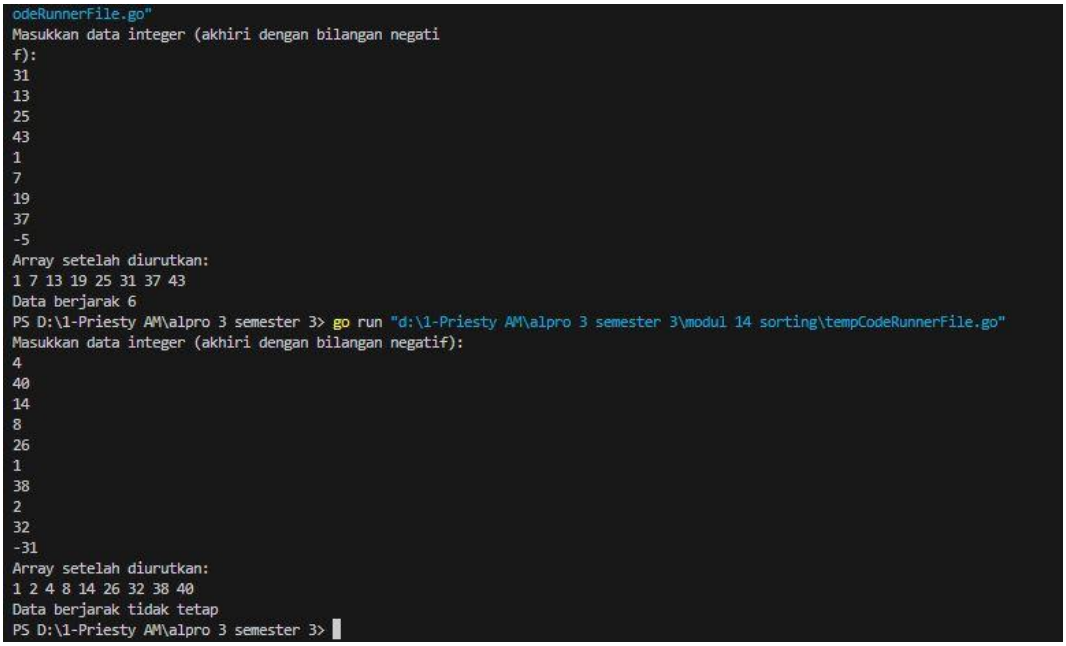
Sourcecode

guided 1.go modul 14 sorting 2 | guided 2.go modul 14 sorting 3 | unguided array 3.go

modul 14 sorting > guided 2.go > main

```go
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



Deskripsi Program

Program ini mengurutkan bilangan bulat yang dimasukkan pengguna menggunakan insertion sort dan memeriksa apakah selisih antar elemen setelah diurutkan konstan. Pengguna memasukkan bilangan hingga bilangan negatif, yang menandai akhir input.

Data diurutkan dengan membandingkan elemen saat ini dengan elemen sebelumnya, lalu menempatkannya di posisi yang sesuai. Setelah terurut, program memeriksa apakah selisih antar elemen tetap. Jika ya, program mencetak selisihnya; jika tidak, program menampilkan bahwa selisih tidak tetap.

Contoh: Input 31 13 25 43 1 7 19 37 -5 menghasilkan data terurut 1 7 13 19 25 31 37 43 dengan selisih tetap 6. Sedangkan input 2 4 40 14 8 26 1 38 2 32 -31 menghasilkan data terurut 1 2 4 8 14 26 32 38 40 tanpa selisih tetap. Program memastikan data diurutkan dan selisih diperiksa dengan akurat.

3.) UNGUIDED
Modul 12&13
Soal Studi Case

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 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.

Sourcecode

unguided 1.go modul 14 sorting 1 | unguided1.go modul 14 sorting 1 | unguided 2.go modul 14 sorti

modul 14 sorting > unguided1.go > main

```go
package main

import (
    "fmt"
    "sort"
)

func main() {
    var n int
    fmt.Scan(&n) // Membaca jumlah daerah

    for i := 0; i < n; i++ {
        var m int
        fmt.Scan(&m) // Membaca jumlah nomor rumah di daerah ini

        oddNumbers := []int{}
        evenNumbers := []int{}

        for j := 0; j < m; j++ {
            var houseNumber int
            fmt.Scan(&houseNumber) // Membaca nomor rumah
            if houseNumber%2 == 0 {
                evenNumbers = append(evenNumbers, houseNumber) // Jika genap
            } else {
                oddNumbers = append(oddNumbers, houseNumber) // Jika ganjil
            }
        }

        // Mengurutkan nomor rumah ganjil secara menaik
        sort.Ints(oddNumbers)

        // Mengurutkan nomor rumah genap secara menurun
        sort.Sort(sort.Reverse(sort.IntSlice(evenNumbers)))

        // Menampilkan hasil untuk nomor ganjil
        for _, num := range oddNumbers {
            fmt.Print(num, " ")
        }
        // Menampilkan hasil untuk nomor genap
        for _, num := range evenNumbers {
            fmt.Print(num, " ")
        }
        fmt.Println() // Baris baru setelah setiap daerah
    }
}
```

Screenshoot Output

```
PS D:\1-Priesty AM\alpro 3 semester 3> go run "d:\1-Priesty AM\alpro 3 semester 3\modul 14 sorting\unguided1.go"
3
5 2 1 7 9 13
1 7 9 13 2
6 189 15 27 39 75 133
15 27 39 75 133 189
3 4 9 1
1 9 4
PS D:\1-Priesty AM\alpro 3 semester 3>
```

Deskripsi Program

Program ini memisahkan bilangan ganjil dan genap dari beberapa kumpulan angka, mengurutkannya, dan menampilkan hasilnya. Pengguna

memasukkan jumlah kumpulan angka (n), lalu untuk setiap kumpulan, jumlah elemen (m) dan angkanya.
Bilangan ganjil diurutkan secara menaik, sedangkan bilangan genap diurutkan secara menurun. Hasilnya, bilangan ganjil ditampilkan lebih dahulu, diikuti bilangan genap, untuk setiap kumpulan.
Contoh:

- Input 5 2 1 7 9 13
  menghasilkan:1 7 9 13 2
- Input 6 189 15 27 39 75 133
  menghasilkan:15 27 39 75 133 189
- Input 3 4 9 1
  menghasilkan:1 9 4

Program ini memastikan setiap kumpulan diproses terpisah dengan hasil yang sesuai.

UNGUIDED

Soal Studi Case

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

odul 14 sorting 1 | unguided 2.go modul 14 sorting 1 | priesty ameiliana maulidah- 2311102175.docx | unguic

modul 14 sorting > unguided 2.go > main

```go
package main

import (
    "fmt"
    "sort"
)

func main() {
    var data []int
    var number int

    for {
        fmt.Scan(&number) // Membaca input

        if number == -5313 {
            break // Menghentikan program jika input -5313
        }

        if number == 0 {
            // Jika menemukan 0, hitung median
            if len(data) == 0 {
                fmt.Println("No data to calculate median")
                continue
            }

            // Mengurutkan data
            sort.Ints(data)

            // Menghitung median
            n := len(data)
            var median int
            if n%2 == 0 {
                median = (data[n/2-1] + data[n/2]) / 2 // Rata-rata dua nilai tengah
            } else {
                median = data[n/2] // Nilai tengah
            }

            // Mencetak median
            fmt.Println(median)
        } else {
            // Menyimpan data yang bukan 0
            data = append(data, number)
        }
    }
}
```

Screenshoot Output

```
PROBLEMS 55  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\1-Priesty AM\alpro 3 semester 3> go run "d:\1-Priesty AM\alpro 3 semester 3\modul 14 sorting\unguided 2.go"
7 23 11 0 5 19 2 29 3 13 17 0 -5313
11
12
PS D:\1-Priesty AM\alpro 3 semester 3>
```

Deskripsi Program

Program ini menghitung median dari data yang dimasukkan pengguna. Setiap kali angka 0 dimasukkan, program akan menghitung dan menampilkan median dari data yang sudah dimasukkan, setelah diurutkan. Program berhenti saat pengguna memasukkan -5313.

Median dihitung dengan:

- Jika jumlah data ganjil, median adalah nilai tengah.
- Jika jumlah data genap, median adalah rata-rata dua nilai tengah.

Contoh:

- Input pertama: 7 23 11 0 → Data diurutkan menjadi 7, 11, 23, median = 11.
- Input kedua: 5 19 2 29 3 13 17 0 → Data diurutkan menjadi 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, median = (11 + 13) / 2 = 12.

Output:

11

12

UNGUIDED

Soal Studi Case

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

unguided max.min 2.go 1 | unguided max.min 3.go 1 | unguided array 3.go

modul 8 > unguided max.min 3.go > main

```go
package main

import (
    "fmt"
)

func main() {
    var n int
    fmt.Print("Masukan banyak data berat balita: ")
    fmt.Scan(&n)

    beratBalita := make([]float64, n)

    for i := 0; i < n; i++ {
        fmt.Printf("Masukan berat balita ke-%d: ", i+1)
        fmt.Scan(&beratBalita[i])
    }

    min := beratBalita[0]
    max := beratBalita[0]
    total := 0.0

    for _, berat := range beratBalita {
        if berat < min {
            min = berat
        }
        if berat > max {
            max = berat
        }
        total += berat
    }

    average := total / float64(n)

    fmt.Printf("Berat balita minimum: %.2f kg\n", min)
    fmt.Printf("Berat balita maksimum: %.2f kg\n", max)
    fmt.Printf("Rerata berat balita: %.2f kg\n", average)
}
```

Screenshoot Output

```
PS D:\1-Priesty AM\alpro 3 semester 3> go run "d:\1-Priesty AM\alpro 3 semester 3\modul 14 sorting\unguided3.go"
Masukkan jumlah buku:
5
Masukkan data buku ke-1 (ID, Judul, Penulis, Penerbit, Eksemplar, Tahun, Rating):
001 pelangi salman pelangi 3 2005 5
Masukkan data buku ke-2 (ID, Judul, Penulis, Penerbit, Eksemplar, Tahun, Rating):
002 kancil astri indria 3 2010 5
Masukkan data buku ke-3 (ID, Judul, Penulis, Penerbit, Eksemplar, Tahun, Rating):
003 kelinci  endyas bintang indonesia 5 2014 5
Masukkan data buku ke-4 (ID, Judul, Penulis, Penerbit, Eksemplar, Tahun, Rating):
Masukkan data buku ke-5 (ID, Judul, Penulis, Penerbit, Eksemplar, Tahun, Rating):
005 noura iput naura 6 2015 5
Masukkan rating yang akan dicari:
5
Buku dengan rating 5 :
ID: 001, Judul: pelangi, Penulis: salman, Penerbit: pelangi, Eksemplar: 3, Tahun: 2005, Rating: 5
ID: 002, Judul: kancil, Penulis: astri, Penerbit: indria, Eksemplar: 3, Tahun: 2010, Rating: 5
ID: 005, Judul: noura, Penulis: iput, Penerbit: naura, Eksemplar: 6, Tahun: 2015, Rating: 5
PS D:\1-Priesty AM\alpro 3 semester 3>
```

Deskripsi Program

Program ini memungkinkan pengguna untuk memasukkan informasi tentang buku-buku yang mereka miliki, termasuk ID, judul, penulis, penerbit, jumlah eksemplar, tahun terbit, dan rating. Pertama, pengguna diminta untuk memasukkan jumlah buku yang ingin didaftarkan, dengan batasan antara 1 hingga 7919. Setelah itu, program akan meminta pengguna untuk mengisi data setiap buku sesuai urutan yang ditentukan. Setelah semua data buku dimasukkan, pengguna dapat mencari buku berdasarkan rating yang diinginkan. Program akan melakukan pencarian dengan membandingkan rating setiap buku dalam daftar, dan jika ditemukan, informasi buku yang sesuai akan ditampilkan, termasuk ID, judul, penulis, penerbit, jumlah eksemplar, tahun terbit, dan rating. Jika tidak ada buku dengan rating yang dicari, program akan memberikan pesan bahwa tidak ada buku yang cocok. Dengan demikian, program ini menggunakan algoritma pencarian linier untuk menemukan buku berdasarkan rating, dan memberikan output yang informatif sesuai dengan data yang dimasukkan oleh pengguna.