

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
MODUL XII & XIII
PENGURUTAN DATA**



Oleh:

OKTAVANIA AYU RAHMADANTY

2311102240

S1IF-11-02

**S1 TEKNIK INFORMATIKA
UNIVERSITAS TELKOM PURWOKERTO**

2024

I. DASAR TEORI

I.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	}

I.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 variable 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 }

```

I.3 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 Telkom University keterurutan. informatics lab
- 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.

	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	$j \leftarrow i$	$j = i$
4	$temp \leftarrow a[j]$	$temp = a[j]$
5	while $j > 0$ and $temp > a[j-1]$ do	for $j > 0$ && $temp > a[j-1]$ {
6	$a[j] \leftarrow a[j-1]$	$a[j] = a[j-1]$
7	$j \leftarrow j - 1$	$j = j - 1$
8	endwhile	}
9	$a[j] \leftarrow temp$	$a[j] = temp$
10	$i \leftarrow i + 1$	$i = i + 1$
11	endwhile	}

I.4 Algoritma Insertion Sort

Adapun algoritma insertion sort pada untuk mengurutkan array bertipe data bilangan bulat secara mengecil atau descending sebagai berikut ini!

```

..  ...
5   type arrInt [4321]int
..  ...
15  func insertionSort1(T *arrInt, n int){
16  /* I.S. terdefinisi array T yang berisi n bilangan bulat
17     F.S. array T terurut secara mengecil atau descending dengan INSERTION SORT*/
18     var temp, i, j int
19     i = 1
20     for i <= n-1 {
21         j = i
22         temp = T[j]
23         for j > 0 && temp > T[j-1] {
24             T[j] = T[j-1]
25             j = j - 1
26         }
27         T[j] = temp
28         i = i + 1
29     }
30 }

```

Sama halnya apabila array yang akan diurutkan adalah bertipe data struct, maka tambahkan field pada saat proses perbandingan dalam pencarian posisi, kemudian tipe data dari variable temp sama dengan struct dari arraynya.

```

..  ...
5   type mahasiswa struct {
..     nama, nim, kelas, jurusan string
..     ipk float64
.. }
.. type arrMhs [2023]mahasiswa
.. ...
15  func insertionSort2(T * arrMhs, n int){
16  /* I.S. terdefinisi array T yang berisi n data mahasiswa
17     F.S. array T terurut secara mengecil atau descending berdasarkan nama dengan
18     menggunakan algoritma INSERTION SORT */
19     var temp i, j int
20     var temp mahasiswa
21     i = 1
22     for i <= n-1 {
23         j = i
24         temp = T[j]
25         for j > 0 && temp.nama > T[j-1].nama {
26             T[j] = T[j-1]
27             j = j - 1
28         }
29         T[j] = temp
30         i = i + 1
31     }
32 }

```

II. GUIDED

Guided 1

Source Code

```
package main

import "fmt"

// Fungsi untuk mengurutkan array menggunakan
selection sort

func selectionSort(arr []int) {

    n := len(arr)

    for i := 0; i < n-1; i++ {

        maxIdx := i

        for j := i + 1; j < n; j++ {

            if arr[j] > arr[maxIdx] { // Cari elemen
terbesar

                maxIdx = j

            }

        }

        arr[i], arr[maxIdx] = arr[maxIdx], arr[i] //
Tukar elemen

    }

}

func main() {

    var n int

    fmt.Print("Masukkan jumlah daerah (n): ")

    fmt.Scan(&n)
```

```
    if n <= 0 || n >= 1000 {  
        fmt.Println("n harus lebih besar dari 0 dan  
kurang dari 1000.")  
        return  
    }  
  
    for i := 0; i < n; i++ {  
        var m int  
        fmt.Printf("Masukkan jumlah rumah kerabat  
untuk daerah ke-%d: ", i+1)  
        fmt.Scan(&m)  
  
        if m <= 0 || m >= 1000000 {  
            fmt.Println("m harus lebih besar dari 0  
dan kurang dari 1000000.")  
            return  
        }  
  
        // Masukkan nomor rumah  
        houses := make([]int, m)  
        fmt.Printf("Masukkan nomor rumah kerabat untuk  
daerah ke-%d: ", i+1)  
        for j := 0; j < m; j++ {  
            fmt.Scan(&houses[j])  
        }  
  
        // Urutkan dengan selection sort
```

```

        selectionSort(houses)

        // Cetak hasil

        fmt.Printf("Hasil urutan rumah untuk daerah
ke-%d: ", i+1)

        for _, house := range houses {

            fmt.Printf("%d ", house)

        }

        fmt.Println()

    }
}

```

Screenshot hasil program

```

PS D:\SEMESTER 3\Laprak Alpro 2\2311102240_Oktavania Ayu Rahmadanty_Modul 11>
SEMESTER 3\Laprak Alpro 2\2311102240_Oktavania Ayu Rahmadanty_Modul 11\guided\guided 1\g
Masukkan jumlah daerah (n): 3
Masukkan jumlah rumah kerabat untuk daerah ke-1: 5
Masukkan nomor rumah kerabat untuk daerah ke-1: 9 1 4 7 23
Hasil urutan rumah untuk daerah ke-1: 23 9 7 4 1
Masukkan jumlah rumah kerabat untuk daerah ke-2: 4
Masukkan nomor rumah kerabat untuk daerah ke-2: 8 6 2 14
Hasil urutan rumah untuk daerah ke-2: 14 8 6 2
Masukkan jumlah rumah kerabat untuk daerah ke-3: 6
Masukkan nomor rumah kerabat untuk daerah ke-3: 9 6 4 13 15 35
Hasil urutan rumah untuk daerah ke-3: 35 15 13 9 6 4
PS D:\SEMESTER 3\Laprak Alpro 2\2311102240_Oktavania Ayu Rahmadanty_Modul 11>

```

Penjelasan:

Program di atas adalah implementasi dalam bahasa Go untuk mengurutkan nomor rumah kerabat di beberapa daerah menggunakan algoritma selection sort secara menurun (descending). Pengguna diminta memasukkan jumlah daerah n , dan untuk setiap daerah, memasukkan jumlah rumah kerabat m beserta nomor rumahnya. Setelah itu, program mengurutkan nomor rumah di setiap daerah secara menurun menggunakan selection sort dan mencetak hasil urutan tersebut. Program juga memvalidasi input sehingga n harus

lebih dari 0 dan kurang dari 1000, serta m harus lebih dari 0 dan kurang dari 1.000.000, untuk memastikan input berada dalam rentang yang wajar.

Guided 2

Source Code

```
package main

import (
    "fmt"
    "math"
)

// Fungsi insertion sort untuk mengurutkan array
func insertionSort(arr []int) {
    n := len(arr)
    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 data berjarak tetap
func isDataConsistentlySpaced(arr []int) (bool, int) {
    if len(arr) < 2 {
        return true, 0 // Array dengan kurang dari 2
        elemen dianggap berjarak tetap
    }

    // Hitung selisih awal
    diff := int(math.Abs(float64(arr[1] - arr[0])))

    for i := 1; i < len(arr)-1; i++ {
        currentDiff := int(math.Abs(float64(arr[i+1] -
arr[i])))
        if currentDiff != diff {
            return false, 0 // Jika ada selisih yang
            berbeda, tidak berjarak tetap
        }
    }

    return true, diff
}

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

```

```
        fmt.Println("Masukkan data (akhiri dengan bilangan  
negatif):")  
  
        for {  
            fmt.Scan(&input)  
            if input < 0 {  
                break  
            }  
            data = append(data, input)  
        }  
  
        // Urutkan data menggunakan insertion sort  
        insertionSort(data)  
  
        // Periksa apakah data berjarak tetap  
        isConsistent, diff :=  
isDataConsistentlySpaced(data)  
  
        // Cetak hasil  
        fmt.Println("Hasil pengurutan:", data)  
        if isConsistent {  
            fmt.Printf("Data berjarak %d\n", diff)  
        } else {  
            fmt.Println("Data berjarak tidak tetap")  
        }  
    }  
}
```

Screenshot hasil program

```
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\SEMESTER 3\Laprak Alpro 2\Oktavania Ayu Rahmadanty_2311102240_Modul 12&13> go
ty_2311102240_Modul 12&13\guided\guided 2\tempCodeRunnerFile.go"
Masukkan data (akhiri dengan bilangan negatif):
31 13 25 43 1 7 19 37 -5
Hasil pengurutan: [1 7 13 19 25 31 37 43]
Data berjarak 6
PS D:\SEMESTER 3\Laprak Alpro 2\Oktavania Ayu Rahmadanty_2311102240_Modul 12&13> |
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\SEMESTER 3\Laprak Alpro 2\2311102240_Oktavania Ayu Rahmadanty_Modul 11>
go run "d:\SEMESTER 3\Laprak Alpro 2\2311102240_Oktavania Ayu Rahmadanty_Modul 11\
Masukkan data (akhiri dengan bilangan negatif):
9
2
5
-1
Hasil pengurutan: [2 5 9]
Data berjarak tidak tetap
PS D:\SEMESTER 3\Laprak Alpro 2\2311102240_Oktavania Ayu Rahmadanty_Modul 11> |
```

Penjelasan:

Program di atas adalah implementasi dalam bahasa Go untuk mengurutkan data numerik menggunakan algoritma insertion sort dan memeriksa apakah data tersebut memiliki jarak yang tetap antara setiap elemen setelah diurutkan. Pengguna diminta untuk memasukkan sejumlah data, yang dihentikan dengan bilangan negatif. Data tersebut kemudian diurutkan secara menaik menggunakan insertion sort. Selanjutnya, program memeriksa konsistensi jarak antar elemen (selisih absolut yang sama di antara elemen-elemen berurutan). Hasil akhirnya mencakup data yang sudah diurutkan dan informasi apakah jarak antar elemen tetap atau tidak, serta besar jaraknya jika tetap.

III. UNGUIDED

Unguided 1

Source Code

```
package main

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

func main() {
    scanner := bufio.NewScanner(os.Stdin)

    fmt.Println("Masukkan jumlah daerah:")
    scanner.Scan()
    jumlahDaerah, _ := strconv.Atoi(scanner.Text())

    hasil := make([][]int, jumlahDaerah)

    for i := 0; i < jumlahDaerah; i++ {
        fmt.Printf("Masukkan bilangan untuk daerah %d:\n", i+1)
        scanner.Scan()
        masukan := scanner.Text()

        bilangan := prosesInput(masukan)

        bilanganGanjil, bilanganGenap :=
        pisahkanGanjilGenap(bilangan)

        sort.Sort(sort.Reverse(sort.IntSlice(bilanganG
        anjil)))
        sort.Ints(bilanganGenap)

        hasil[i] = append(bilanganGanjil,
        bilanganGenap...)
    }
}
```

```

        fmt.Println("\nKeluaran:")
        for i, hasilDaerah := range hasil {
            fmt.Printf("Daerah %d: %v\n", i+1,
hasilDaerah)
        }
    }

func prosesInput(masukan string) []int {
    bagian := strings.Fields(masukan)
    bilangan := make([]int, len(bagian))
    for i, b := range bagian {
        bilangan[i], _ = strconv.Atoi(b)
    }
    return bilangan
}

func pisahkanGanjilGenap(bilangan []int) ([]int,
[]int) {
    var bilanganGanjil, bilanganGenap []int
    for _, bil := range bilangan {
        if bil%2 == 0 {
            bilanganGenap = append(bilanganGenap, bil)
        } else {
            bilanganGanjil = append(bilanganGanjil,
bil)
        }
    }
    return bilanganGanjil, bilanganGenap
}

```

Screenshot hasil program

```
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\SEMESTER 3\Laprak Alpro 2\Oktavania Ayu Rahmadanty_2311102240_Modul 12&13> go
13\unguided\unguided 1\unguided1.go"
Masukkan jumlah daerah:
3
Masukkan bilangan untuk daerah 1:
5 2 1 7 9 13
Masukkan bilangan untuk daerah 2:
6 189 15 27 39 75 133
Masukkan bilangan untuk daerah 3:
3 4 9 1

Keluaran:
Daerah 1: [13 9 7 5 1 2]
Daerah 2: [189 133 75 39 27 15 6]
Daerah 3: [9 3 1 4]
PS D:\SEMESTER 3\Laprak Alpro 2\Oktavania Ayu Rahmadanty_2311102240_Modul 12&13> |
```

Penjelasan:

Program ini memproses sejumlah data bilangan bulat yang dikelompokkan berdasarkan daerah, lalu memisahkan bilangan ganjil dan genap untuk setiap daerah. Setelah memisahkan, bilangan ganjil diurutkan secara menurun (descending) dan bilangan genap diurutkan secara menaik (ascending). Hasil pengurutan untuk setiap daerah digabungkan dengan urutan ganjil terlebih dahulu, diikuti oleh genap, dan disimpan. Pada akhirnya, program mencetak hasil pengurutan untuk setiap daerah secara berurutan. Input data diberikan dalam bentuk string untuk setiap daerah, diproses menjadi array bilangan, lalu dipisahkan dan diurutkan sesuai ketentuan.

Unguided 2

Source Code

```
package main

import (
    "fmt"
)

func main() {
    var data []int
    for {
        var bilangan int
        fmt.Scan(&bilangan)

        if bilangan == -5313 {
            break
        }
        \
        if bilangan == 0 {
            insertionSort(data)

            median := hitungMedian(data)
            fmt.Println(median)
        } else {
            data = append(data, bilangan)
        }
    }
}

func insertionSort(data []int) {
    for i := 1; i < len(data); i++ {
        key := data[i]
        j := i - 1

        for j >= 0 && data[j] > key {
            data[j+1] = data[j]
            j--
        }

        data[j+1] = key
    }
}
```



```
func hitungMedian(data []int) int {
    n := len(data)
    if n == 0 {
        return 0
    }

    if n%2 == 1 {
        return data[n/2]
    } else {
        return (data[(n/2)-1] + data[n/2]) / 2
    }
}
```

Screenshot hasil program

```
PS D:\SEMESTER 3\Laparak Alpro 2\Oktavania Ayu Rahmadanty_2311102240_Modul 12&13> go run ty_2311102240_Modul 12&13\unguided\unguided2\unguided2.go"
7 23 11 0 5 19 2 29 3 13 17 0 -5313
11
12
PS D:\SEMESTER 3\Laparak Alpro 2\Oktavania Ayu Rahmadanty_2311102240_Modul 12&13> 
```

Penjelasan:

Program ini menghitung nilai median dari kumpulan data bilangan bulat positif yang diinput secara berurutan, di mana setiap kali bilangan 0 muncul, median dari data yang sudah terbaca dihitung dan dicetak. Program membaca bilangan satu per satu, menyimpan bilangan ke dalam array, dan mengurutkan array menggunakan insertion sort saat menemukan bilangan 0. Median dihitung berdasarkan jumlah data: jika ganjil, median adalah elemen tengah; jika genap, median adalah rata-rata dua elemen tengah yang dibulatkan ke bawah. Proses berakhir ketika bilangan -5313 ditemukan, dan hasil median dicetak setiap kali 0 muncul dalam input.

Unguided 3

Source Code

```
package main

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

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

func main() {
    var n int

    scanner := bufio.NewScanner(os.Stdin)

    fmt.Print("Masukkan jumlah data buku: ")
    scanner.Scan()
    n, _ = strconv.Atoi(scanner.Text())

    buku := make([]Buku, n)

    for i := 0; i < n; i++ {
        fmt.Printf("Masukkan data buku ke-%d (ID, Judul, Penulis, Penerbit, Eksemplar, Tahun, Rating):\n", i+1)

        scanner.Scan()
        id, _ := strconv.Atoi(scanner.Text())

        scanner.Scan()
        judul := scanner.Text()
```

```

scanner.Scan()
penulis := scanner.Text()

scanner.Scan()
penerbit := scanner.Text()

scanner.Scan()
eksemplar, _ := strconv.Atoi(scanner.Text())

scanner.Scan()
tahun, _ := strconv.Atoi(scanner.Text())

scanner.Scan()
rating, _ := strconv.Atoi(scanner.Text())

buku[i] = Buku{id, judul, penulis, penerbit,
eksemplar, tahun, rating}
}

var favorit Buku
for _, b := range buku {
    if b.Rating > favorit.Rating {
        favorit = b
    }
}

fmt.Println("Buku Terfavorit:")
fmt.Printf("Judul: %s, Penulis: %s, Penerbit: %s,
Tahun: %d\n", favorit.Judul, favorit.Penulis,
favorit.Penerbit, favorit.Tahun)

for i := 0; i < len(buku); i++ {
    for j := i + 1; j < len(buku); j++ {
        if buku[j].Rating > buku[i].Rating {
            buku[i], buku[j] = buku[j], buku[i]
        }
    }
}

fmt.Println("5 Buku dengan Rating Tertinggi:")
for i := 0; i < 5 && i < len(buku); i++ {
    fmt.Printf("%d. Judul: %s, Penulis: %s,
Penerbit: %s, Tahun: %d\n", i+1, buku[i].Judul,
buku[i].Penulis, buku[i].Penerbit, buku[i].Tahun)
}

```

```

    fmt.Print("Masukkan rating untuk mencari buku: ")
    scanner.Scan()
    cariRating, _ := strconv.Atoi(scanner.Text())

    found := false
    for _, b := range buku {
        if b.Rating == cariRating {
            fmt.Printf("Judul: %s, Penulis: %s,
Penerbit: %s, Tahun: %d\n", b.Judul, b.Penulis,
b.Penerbit, b.Tahun)
            found = true
            break
        }
    }
    if !found {
        fmt.Println("Tidak ada buku dengan rating
seperti itu.")
    }
}

```

Screenshot hasil program

The screenshot shows a terminal window with the following output:

```

11223344
BUMI
Tere Liye
Gramedia Pustaka Utama
2
2016
4
Masukkan data buku ke-2 (ID, Judul, Penulis, Penerbit, Eksemplar, Tahun, Rating):
55667788
Sebuah Usaha Melupakan
Boy Candra
Mediakita
306
2019
3
Buku Terfavorit:
Judul: BUMI, Penulis: Tere Liye, Penerbit: Gramedia Pustaka Utama, Tahun: 2016
5 Buku dengan Rating Tertinggi:
1. Judul: BUMI, Penulis: Tere Liye, Penerbit: Gramedia Pustaka Utama, Tahun: 2016
2. Judul: Sebuah Usaha Melupakan, Penulis: Boy Candra, Penerbit: Mediakita, Tahun: 2019
Masukkan rating untuk mencari buku: 4
Judul: BUMI, Penulis: Tere Liye, Penerbit: Gramedia Pustaka Utama, Tahun: 2016
PS D:\SEMESTER 3\Laprak Alpro 2\Oktavania Ayu Rahmadanty_2311102240_Modul 12&13>

```

Penjelasan:

Error "declared and not used" pada Golang terjadi karena variabel rating dideklarasikan di awal fungsi main tetapi tidak pernah digunakan, sebab nama yang sama digunakan kembali dalam loop untuk membaca input. Solusinya adalah menghapus deklarasi variabel rating yang tidak diperlukan di luar loop. Variabel rating cukup dibuat dan digunakan langsung di dalam loop untuk membaca data buku, atau saat membaca input untuk pencarian rating. Dengan begitu, setiap variabel yang dideklarasikan benar-benar digunakan, sehingga error tersebut tidak muncul lagi.