

**LAPORAN PRAKTIKUM
ALGORITMA PEMROGRAMAN 2**

MODUL 12

SEARCHING



Oleh:

Fadhel Yussie Ramadhan

2311102322

S1 TEKNIK INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Sequential Search Pencarian secara sekuensial ini adalah pencarian yang dilakukan dari data pertama, kedua hingga terakhir secara satu persatu dan berurutan. Ciri khas dari pencarian ini adalah proses pencarian akan berhenti ketika data yang dicari ditemukan, walaupun masih terdapat data yang belum dicek nilainya.

Algoritma ini dikenal dengan nama Sequential Search, karena prosesnya melakukan pengecekan setiap elemen array secara satu persatu dan sekuensial dari data pertama hingga ditemukan atau data terakhir.

- 1) Asumsi terdapat suatu array of integer T dengan indeks dari 0 hingga N-1, dan suatu nilai yang dicari pada array T, yaitu X.
- 2) Status pencarian digunakan untuk menandakan data yang dicari ditemukan atau tidak, misalnya found dengan tipe boolean.
- 3) Pencarian dilakukan dari T[0] sampai ke T[N-1], setiap kali perbandingan dengan X, update nilai found.
- 4) Perulangan harus dihentikan apabila status pencarian found bernilai true (data ditemukan) atau T[N-1] telah dicek. Berikut ini adalah contoh notasi algoritma sekuensial search sesuai dengan penjelasan di atas

	Notasi Algoritma	Notasi dalam bahasa Go
1	<code>found ← false</code>	<code>found = false</code>
2	<code>i ← 0</code>	<code>i = 0</code>
3	<code>while i < n and not found do</code>	<code>for i < n && !found {</code>
4	<code> found ← T[i] == X</code>	<code> found = T[i] == X</code>
5	<code> i ← i + 1</code>	<code> i = i + 1</code>
6	<code>endwhile</code>	<code>}</code>

ii. GUIDED

1. Source Code

```
2. package main
3.
4. import "fmt"
5.
6. // Fungsi untuk mengurutkan array menggunakan selection sort
7. func selectionSort(arr []int) {
8.     n := len(arr)
9.     for i := 0; i < n-1; i++ {
10.         maxIdx := i
11.         for j := i + 1; j < n; j++ {
12.             if arr[j] > arr[maxIdx] { // Cari elemen terbesar
13.                 maxIdx = j
14.             }
15.         }
16.         arr[i], arr[maxIdx] = arr[maxIdx], arr[i] // Tukar elemen
17.     }
18. }
19.
20. func main() {
21.     var n int
22.     fmt.Print("Masukkan jumlah daerah (n): ")
23.     fmt.Scan(&n)
24.
25.     if n <= 0 || n >= 1000 {
26.         fmt.Println("n harus lebih besar dari 0 dan kurang dari
27.         1000.")
28.         return
29.     }
30.     for i := 0; i < n; i++ {
31.         var m int
32.         fmt.Printf("Masukkan jumlah rumah kerabat untuk daerah ke-
33.         %d: ", i+1)
34.         fmt.Scan(&m)
35.
36.         if m <= 0 || m >= 1000000 {
37.             fmt.Println("m harus lebih besar dari 0 dan kurang
38.             dari 1000000.")
39.             return
40.         }
41.         // Masukkan nomor rumah
```

```

41.         houses := make([]int, m)
42.         fmt.Printf("Masukkan nomor rumah kerabat untuk daerah ke-
%d: ", i+1)
43.         for j := 0; j < m; j++ {
44.             fmt.Scan(&houses[j])
45.         }
46.
47.         // Urutkan dengan selection sort
48.         selectionSort(houses)
49.
50.         // Cetak hasil
51.         fmt.Printf("Hasil urutan rumah untuk daerah ke-%d: ", i+1)
52.         for _, house := range houses {
53.             fmt.Printf("%d ", house)
54.         }
55.         fmt.Println()
56.     }
57. }
58. }

```

```

OUTPUT  PROBLEMS 2 DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\USER\Documents\KULIAH\SEMESTER 3\ALPRO 2\PRAKTIKUM\modul 12> go run "c:\Users\USER\Documents\KULIAH\SEMESTER 3\ALPRO 2\PRAKTIKUM\modul 12\g1.go"
Masukkan jumlah daerah (n): 2
Masukkan jumlah rumah kerabat untuk daerah ke-1: 2
Masukkan nomor rumah kerabat untuk daerah ke-1: 01
02
Hasil urutan rumah untuk daerah ke-1: 2 1
Masukkan jumlah rumah kerabat untuk daerah ke-2: 10
Masukkan nomor rumah kerabat untuk daerah ke-2: 20
30
40
50
60
03
04
05
06
07
Hasil urutan rumah untuk daerah ke-2: 60 50 40 30 20 7 6 5 4 3
PS C:\Users\USER\Documents\KULIAH\SEMESTER 3\ALPRO 2\PRAKTIKUM\modul 12>

```

2.

```

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")
    }
}

```

Output

```
OUTPUT PROBLEMS 2 DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\USER\Documents\KULIAH\SEMESTER 3\ALPRO 2\PRAKTIKUM\modul 12> go run "c:\Users\USER\Documents\KULIAH\SEMESTER 3\ALPRO 2\PRAKTIKUM\modul 12\g2.go"
Masukkan data (akhiri dengan bilangan negatif):
3
4
5
6
-1
Hasil pengurutan: [3 4 5 6]
Data berjarak 1
PS C:\Users\USER\Documents\KULIAH\SEMESTER 3\ALPRO 2\PRAKTIKUM\modul 12>
```

II. UNGUIDED

1. Source Code

```
2. package main
3.
4. import (
5.     "fmt"
6.     "sort"
7.     "strings"
8. )
9.
10. func main() {
11.     var n int
12.     fmt.Print("Masukkan jumlah daerah (n): ")
13.     _, err := fmt.Scan(&n)
14.     if err != nil || n <= 0 || n >= 1000 {
15.         fmt.Println("n harus lebih besar dari 0 dan kurang
dari 1000.")
16.         return
17.     }
18.
19.     for i := 0; i < n; i++ {
20.         var m int
21.         fmt.Printf("Masukkan jumlah rumah kerabat untuk
daerah ke-%d: ", i+1)
22.         _, err := fmt.Scan(&m)
23.         if err != nil || m <= 0 || m >= 1000000 {
24.             fmt.Println("m harus lebih besar dari 0 dan
kurang dari 1000000.")
25.             return
26.         }
27.
28.         houses := make([]int, m)
29.         fmt.Printf("Masukkan nomor rumah kerabat untuk daerah
ke-%d: ", i+1)
30.         for j := 0; j < m; j++ {
31.             _, err := fmt.Scan(&houses[j])
32.             if err != nil {
33.                 fmt.Println("Input tidak valid. Silakan
masukkan angka.")
34.                 return
35.             }
36.         }
37.
```



```

38.         var odd, even []int
39.         for _, house := range houses {
40.             if house%2 == 0 {
41.                 even = append(even, house)
42.             } else {
43.                 odd = append(odd, house)
44.             }
45.         }
46.
47.         sort.Ints(odd)
48.         sort.Ints(even)
49.
50.         result := append(odd, even...)
51.         fmt.Printf("Hasil urutan rumah untuk daerah ke-%d:
%s\n", i+1,
strings.Trim(strings.Join(strings.Fields(fmt.Sprint(result)),
" "), "[""))
52.     }
53. }
22.

```

Output:

```

PS C:\Users\USER\Documents\KULIAH\SEMESTER 3\ALPRO 2\PRAKTIKUM\modul 12> go run "c:\Users\USER\Documents\KULIAH\
SEMESTER 3\ALPRO 2\PRAKTIKUM\modul 12\unguided1.go"
Masukkan jumlah daerah (n): 2
Masukkan jumlah rumah kerabat untuk daerah ke-1: 1
Masukkan nomor rumah kerabat untuk daerah ke-1: 20
Hasil urutan rumah untuk daerah ke-1: 20
Masukkan jumlah rumah kerabat untuk daerah ke-2: 2
Masukkan nomor rumah kerabat untuk daerah ke-2: 10
11
Hasil urutan rumah untuk daerah ke-2: 11 10
PS C:\Users\USER\Documents\KULIAH\SEMESTER 3\ALPRO 2\PRAKTIKUM\modul 12>

```

Keterangan:

Program ini meminta pengguna untuk memasukkan jumlah daerah dan nomor rumah kerabat di setiap daerah, kemudian memisahkan, mengurutkan, dan mencetak nomor rumah ganjil dan genap secara terpisah dalam urutan yang diinginkan.

2. Source Code

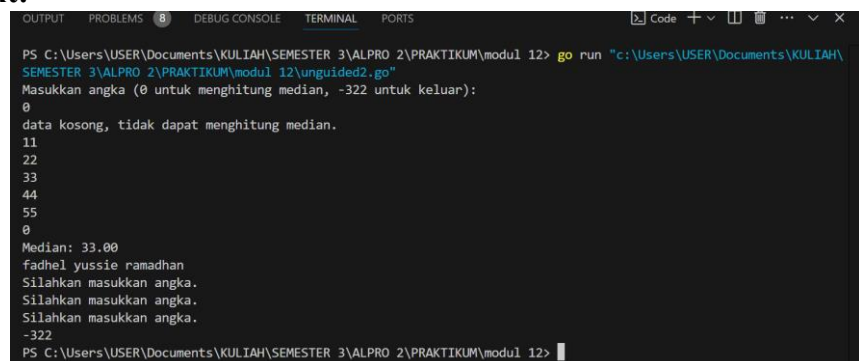
```
54.package main
55.
56.import (
57.    "fmt"
58.)
59.
60.func hitungMedian(arr []int) float64 {
61.    n := len(arr)
62.    if n == 0 {
63.        return 0 // Menghindari pembagian dengan nol
64.    }
65.    if n%2 == 1 {
66.        return float64(arr[n/2])
67.    }
68.    return float64(arr[n/2-1]+arr[n/2]) / 2.0
69.}
70.
71.func insertionSort(arr []int) {
72.    for i := 1; i < len(arr); i++ {
73.        key := arr[i]
74.        j := i - 1
75.        for j >= 0 && arr[j] > key {
76.            arr[j+1] = arr[j]
77.            j = j - 1
78.        }
79.        arr[j+1] = key
80.    }
81.}
82.
83.func main() {
84.    var data []int
85.    var input int
86.
87.    fmt.Println("Masukkan angka (0 untuk menghitung
median, -322 untuk keluar):")
88.
89.    for {
90.        _, err := fmt.Scan(&input)
91.        if err != nil {
```

```

92.         fmt.Println("Silahkan masukkan angka.")
93.         // Clear the input buffer
94.         var discard string
95.         fmt.Scanln(&discard)
96.         continue
97.     }
98.
99.     if input == -322 {
100.         break
101.     }
102.
103.     if input == 0 {
104.         if len(data) == 0 {
105.             fmt.Println("data kosong, tidak
dapat menghitung median.")
106.             continue
107.         }
108.         insertionSort(data)
109.         median := hitungMedian(data)
110.         fmt.Printf("Median: %.2f\n",
median)
111.     } else {
112.         data = append(data, input)
113.     }
114. }
115. }
23.

```

Output:



```

PS C:\Users\USER\Documents\KULIAH\SEMESTER 3\ALPRO 2\PRAKTIKUM\modul 12> go run "c:\Users\USER\Documents\KULIAH\SEMESTER 3\ALPRO 2\PRAKTIKUM\modul 12\unguided2.go"
Masukkan angka (0 untuk menghitung median, -322 untuk keluar):
0
data kosong, tidak dapat menghitung median.
11
22
33
44
55
0
Median: 33.00
fadhel yussie ramadhan
Silahkan masukkan angka.
Silahkan masukkan angka.
Silahkan masukkan angka.
-322
PS C:\Users\USER\Documents\KULIAH\SEMESTER 3\ALPRO 2\PRAKTIKUM\modul 12>

```

Keterangan:

Program ini memungkinkan pengguna untuk memasukkan angka ke dalam sebuah slice, menghitung dan mencetak median dari angka-angka yang dimasukkan setelah pengguna memasukkan angka 0, serta menyediakan opsi untuk keluar dari program dengan memasukkan -322, sambil melakukan validasi input untuk memastikan hanya angka yang diterima.

3. Source Code

```
116.     package main
117.
118.     import "fmt"
119.
120.     const NMAX int = 9999
121.
122.     type Buku struct {
123.         id, judul, penulis, penerbit string
124.         eksemplar, tahun, rating      int
125.     }
126.
127.     type DaftarBuku = [NMAX]Buku
128.
129.     func DaftarkanBuku(pustaka *DaftarBuku, n int)
130.     {
131.         for i := 0; i < n; i++ {
132.             fmt.Printf("[Buku %v]\n", i+1)
133.             fmt.Print("Masukkan id buku: ")
134.             fmt.Scanln(&pustaka[i].id)
135.             fmt.Print("Masukkan judul buku: ")
136.             fmt.Scanln(&pustaka[i].judul)
137.             fmt.Print("Masukkan penulis buku: ")
138.             fmt.Scanln(&pustaka[i].penulis)
139.             fmt.Print("Masukkan penerbit buku: ")
140.             fmt.Scanln(&pustaka[i].penerbit)
141.             fmt.Print("Masukkan eksemplar buku: ")
142.             fmt.Scanln(&pustaka[i].eksemplar)
143.             fmt.Print("Masukkan tahun buku: ")
144.             fmt.Scanln(&pustaka[i].tahun)
145.             fmt.Print("Masukkan rating buku: ")
146.             fmt.Scanln(&pustaka[i].rating)
147.             fmt.Println()
```

```

148.     }
149.
150.     func CetakTerFavorit(pustaka DaftarBuku, n int)
151.     {
152.         max := pustaka[0]
153.         for i := 1; i < n; i++ {
154.             if pustaka[i].rating > max.rating {
155.                 max = pustaka[i]
156.             }
157.         }
158.         fmt.Println("[Buku Favorit]")
159.         fmt.Printf("- Judul: %v\n", max.judul)
160.         fmt.Printf("- Penulis: %v\n", max.penulis)
161.         fmt.Printf("- Penerbit: %v\n",
162. max.penerbit)
163.         fmt.Printf("- Tahun: %v\n", max.tahun)
164.         fmt.Printf("- Rating: %v\n", max.rating)
165.     }
166.
167.     func UrutBuku(pustaka *DaftarBuku, n int) {
168.         var temp Buku
169.         for i := 1; i < n; i++ {
170.             temp = pustaka[i]
171.             j := i - 1
172.             for j >= 0 && pustaka[j].rating >
173. temp.rating {
174.                 pustaka[j+1] = pustaka[j]
175.                 j--
176.             }
177.             pustaka[j+1] = temp
178.         }
179.     }
180.
181.     func Cetak5Terbaru(pustaka DaftarBuku, n int) {
182.         fmt.Printf("[%v Buku Favorit]\n", n)
183.         for i := 0; i < n && i < 5; i++ {
184.             fmt.Printf("%v. %v\n", i+1,
185. pustaka[i].judul)
186.         }
187.     }

```

```

185.
186.     func CariBuku(pustaka DaftarBuku, n int, r int)
187.     {
188.         low, high := 0, n-1
189.         for low <= high {
190.             mid := (low + high) / 2
191.             if r == pustaka[mid].rating {
192.                 fmt.Printf("[Buku dengan Rating
193.                 %v]\n", r)
194.                 fmt.Printf("- Judul: %v\n",
195.                 pustaka[mid].judul)
196.                 fmt.Printf("- Penulis: %v\n",
197.                 pustaka[mid].penulis)
198.                 fmt.Printf("- Penerbit: %v\n",
199.                 pustaka[mid].penerbit)
200.                 fmt.Printf("- Tahun: %v\n",
201.                 pustaka[mid].tahun)
202.                 fmt.Printf("- Eksemplar: %v\n",
203.                 pustaka[mid].eksemplar)
204.                 fmt.Printf("- Rating: %v\n",
205.                 pustaka[mid].rating)
206.                 return
207.             } else if r > pustaka[mid].rating {
208.                 low = mid + 1
209.             } else {
210.                 high = mid - 1
211.             }
212.         }
213.         fmt.Println("Tidak ada buku dengan rating
214.         seperti itu.")
215.     }
216.
217.     func main() {
218.         var pustaka DaftarBuku
219.         var nPustaka, rBuku int
220.
221.         fmt.Print("Masukkan jumlah buku: ")
222.         fmt.Scanln(&nPustaka)
223.
224.         DaftarkanBuku(&pustaka, nPustaka)
225.         CetakTerFavorit(pustaka, nPustaka)
226.         UrutBuku(&pustaka, nPustaka)

```

```

218.          Cetak5Terbaru(pustaka, nPustaka)
219.
220.          fmt.Print("Masukkan rating buku yang ingin
dicari: ")
221.          fmt.Scanln(&rBuku)
222.
223.          CariBuku(pustaka, nPustaka, rBuku)}
24.

```

Output:

```

OUTPUT  PROBLEMS 10 DEBUG CONSOLE TERMINAL PORTS

SEMESTER 3\ALPRO 2\PRAKTIKUM\modul 12\unguided3.go"
Masukkan jumlah buku: 1
[Buku 1]
Masukkan id buku: 2311102322
Masukkan judul buku: Laprak
Masukkan penulis buku: FadhelyussieRamadhan
Masukkan penerbit buku: PtAuran
Masukkan eksemplar buku: 12
Masukkan tahun buku: 2024
Masukkan rating buku: 9.9

[Buku Favorit]
- Judul: Laprak
- Penulis: FadhelyussieRamadhan
- Penerbit: PtAuran
- Tahun: 2024
- Rating: 9
[1 Buku Favorit]
1. Laprak
Masukkan rating buku yang ingin dicari: [Buku dengan Rating 9]
- Judul: Laprak
- Penulis: FadhelyussieRamadhan
- Penerbit: PtAuran
- Tahun: 2024
- Eksemplar: 12
- Rating: 9
PS C:\Users\USER\Documents\KULIAH\SEMESTER 3\ALPRO 2\PRAKTIKUM\modul 12>

```

Keterangan:

Program ini mengelola daftar buku dengan memungkinkan pengguna untuk mendaftarkan buku, mencetak buku dengan rating tertinggi, mengurutkan buku berdasarkan rating, mencetak lima buku terbaru, dan mencari buku berdasarkan rating tertentu, semua dengan menggunakan struktur data dan fungsi yang terpisah untuk setiap fungsionalitas.