

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL 11

Pencarian nilai acak pada himpunan data



Oleh:

AHMAD RUBA'I

103112400074

IF-12-01

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2025

I. DASAR TEORI

A. Sequential Search

Pencarian secara sekuensial ini adalah pencarian yang dilakukan dari data pertama, kedua hingga terakhir secara satu persatu dan berurutan. Ciri khas pencarian ini adalah proses pencarian akan berhenti Ketika data yang dicari ditemukan, walaupun masih terdapat data yang belum di cek 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.

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

B. Binary Search

Ide algoritma adalah: (dengan asumsi data terurut dari kecil membesar(ascending), dan data dengan indeks kecil ada di “kiri” dan indeks besar ada di “kanan”)

	Notasi Algoritma	Notasi dalam bahasa Go
1	kr \leftarrow 0	kr = 0
2	kn \leftarrow n-1	kn = n-1
3	found \leftarrow false	found = false
4	while kr <= kn and not found do	for kr <= kn && !found {
5	med \leftarrow (kr+kn) div 2	med = (kr+kn) / 2
6	if a[med] < X then	if a[med] < X {
7	kr \leftarrow med + 1	kr = med + 1
8	else if a[med] > X then	} else if a[med] > X {
9	kn \leftarrow med - 1	kn = med
10	else	} else {
11	found \leftarrow true	found = true
12	endif	}
13	endwhile	}

II. GUIDED

Guided 1

```
1 //Ahmad Ruba'i
2 //103112400074
3 package main
4
5 import (
6     "fmt"
7     "sort"
8 )
9
10 func sequentialSearch(arr[]float64, target float64)(int, int){
11     iterations := 0
12     for i, val := range arr{
13         iterations++
14         fmt.Printf("sequential step %d: cek arr [%d]=%.f\n", iterations, i, val)
15         if val == target{
16             return i, iterations
17         }
18     }
19     return -1, iterations
20 }
21
22 func binarySearch(arr []float64, target float64)(int, int){
23     iterations := 0
24     low := 0
25     high := len (arr)-1
26     for low <= high {
27         iterations++
28         mid := (low + high)/ 2
29         fmt.Printf("binary step %d: cek arr[%d]= %.f\n", iterations, mid, arr[mid])
30
31         if arr[mid] == target {
32             return mid, iterations
33         }else if target < arr[mid]{
34             high = mid -1
35         }else{
36             low = mid +1
37         }
38     }
39     return -1, iterations
40 }
41
42 func main(){
43     //array awal
44     data:= []float64{2, 7, 9, 1, 5, 6, 18, 13, 25, 20}
45     target := 13.0
46
47     fmt.Println("sequential search (data tidak perluurut):")
```

```

46     idxseq, iterseq := sequentialSearch(data, target)
47     if idxseq != -1{
48         fmt.Printf("hasil : ditemukan di indeks %d dalam %d langkah\n",idxseq, iterseq)
49     }else{
50         fmt.Printf("hasil : tidak ditemukan setelah %d langkah\n",iterseq)
51     }
52
53     //Binary earch array diurutkan
54     sort.Float64s(data)
55     fmt.Println("Binary search (setelah data diurutkan):")
56     fmt.Println("Data terurut:", data)
57
58     idxBin, iterBin := binarySEarch(data, target)
59     if idxBin!= -1 {
60         fmt.Printf("Hasil : Ditemukan indeks %d dalam %d langkah\n", idxBin,iterBin)
61     }else{
62         fmt.Printf("Hasil : Tidak ditemukan Setelah %d langkah\n",iterBin)
63     }
64 }

```

Output:

```

PS C:\VSCODE\103112400074_Modul 8> go run "c:\VS
sequential search (data tidak perlu urut):
sequential step 1: cek arr [0]=2
sequential step 2: cek arr [1]=7
sequential step 3: cek arr [2]=9
sequential step 4: cek arr [3]=1
sequential step 5: cek arr [4]=5
sequential step 6: cek arr [5]=6
sequential step 7: cek arr [6]=18
sequential step 8: cek arr [7]=13
hasil : ditemukan di indeks 7 dalam 8 langkah

Binary search (setelah data diurutkan):
Data terurut: [1 2 5 6 7 9 13 18 20 25]
binary stop 1: cek arr[4]= 7
binary stop 2: cek arr[6]= 13
Hasil : Ditemukan indeks 6 dalam 2 langkah
PS C:\VSCODE\103112400074_Modul 8>

```

Deskripsi Program:

Program di atas digunakan untuk membandingkan dua metode pencarian data dalam array bertipe float64, yaitu **Sequential Search** dan **Binary Search**. Program mencari nilai tertentu (target) dari sebuah array. Untuk **Sequential Search**, program menelusuri elemen satu per satu tanpa memerlukan data yang terurut. Sedangkan untuk **Binary Search**, array diurutkan terlebih dahulu menggunakan fungsi `sort.Float64s`, lalu dilakukan pencarian secara membagi dua bagian data hingga menemukan target. Kedua metode menampilkan setiap langkah pencarian (step) yang dilakukan, dan hasil akhirnya menunjukkan apakah data ditemukan atau tidak, serta berapa banyak langkah (iterasi) yang dibutuhkan oleh masing-masing metode. Program ini membantu memahami perbedaan efisiensi antara kedua metode pencarian tersebut.

Guided 2

```
1 //Ahmad Ruba'i
2 //103112400074
3 package main
4
5 import (
6     "fmt"
7     "sort"
8 )
9
10 type mahasiswa struct {
11     nama, nim, kelas, jurusan string
12     ipk                          float64
13 }
14
15 type arrMhs [2023]mahasiswa
16
17 // Sequential Search berdasarkan nama
18 func SeqSearch_3(T arrMhs, n int, X string) (int, int) {
19     var found int = -1
20     var j int = 0
21     var iterasi int = 0
22
23     for j < n && found == -1 {
24         iterasi++
25         if T[j].nama == X {
26             found = j
27         }
28         j++
29     }
30     return found, iterasi
31 }
32
33 // Binary Search berdasarkan NIM (data harus sudah terurut berdasarkan nim)
34 func BinarySearch_3(T arrMhs, n int, X string) (int, int) {
35     var found int = -1
36     var med int
37     var kr int = 0
38     var kn int = n - 1
39     var iterasi int = 0
40
41     for kr <= kn && found == -1 {
42         iterasi++
43         med = (kr + kn) / 2
44         if X < T[med].nim {
45             kn = med - 1
```

```

46         } else if X > T[med].nim {
47         |     kr = med + 1
48         } else {
49         |     found = med
50         }
51     }
52     return found, iterasi
53 }
54
55 func main() {
56     var data arrMhs
57     n := 10
58
59     // Mengisi data secara manual
60     data = arrMhs{
61         {nama: "Ari", nim: "2201", kelas: "A", jurusan: "Informatika", ipk: 3.4},
62         {nama: "Budi", nim: "2203", kelas: "A", jurusan: "Informatika", ipk: 3.6},
63         {nama: "Cici", nim: "2202", kelas: "B", jurusan: "Sistem Informasi", ipk: 3.5},
64         {nama: "Dina", nim: "2205", kelas: "A", jurusan: "Informatika", ipk: 3.3},
65         {nama: "Eko", nim: "2204", kelas: "B", jurusan: "Sistem Informasi", ipk: 3.7},
66         {nama: "Fajar", nim: "2206", kelas: "C", jurusan: "Informatika", ipk: 3.1},
67         {nama: "Gita", nim: "2209", kelas: "C", jurusan: "Informatika", ipk: 3.8},
68         {nama: "Hana", nim: "2208", kelas: "B", jurusan: "Sistem Informasi", ipk: 3.2},
69         {nama: "Iwan", nim: "2207", kelas: "C", jurusan: "Informatika", ipk: 3.0},
70         {nama: "Joko", nim: "2210", kelas: "A", jurusan: "Informatika", ipk: 3.9},
71     }
72
73     // Pencarian Sequential Search berdasarkan nama
74     namaDicari := "Fajar"
75     idxSeq, iterSeq := SeqSearch_3(data, n, namaDicari)
76
77     fmt.Printf("Sequential Search - Cari nama '%s'\n", namaDicari)
78     if idxSeq != -1 {
79         |     fmt.Printf("Ditemukan di indeks: %d, Iterasi: %d\n", idxSeq, iterSeq)
80     } else {
81         |     fmt.Printf("Tidak ditemukan, Iterasi: %d\n", iterSeq)
82     }
83
84     // Urutkan data berdasarkan NIM untuk binary search
85     sort.Slice(data[:n], func(i, j int) bool {
86         |     return data[i].nim < data[j].nim
87     })
88
89     // Pencarian Binary Search berdasarkan NIM
90     nimDicari := "2206"
91     idxBin, iterBin := BinarySearch_3(data, n, nimDicari)
92
93     fmt.Printf("\nBinary Search - Cari NIM '%s'\n", nimDicari)
94     if idxBin != -1 {
95         |     fmt.Printf("Ditemukan di indeks: %d, Iterasi: %d\n", idxBin, iterBin)
96     } else {
97         |     fmt.Printf("Tidak ditemukan, Iterasi: %d\n", iterBin)
98     }
99 }

```

Output

```
PS C:\VSCODE\103112400074_Modul 8> go
Sequential Search - Cari nama 'Fajar'
Ditemukan di indeks: 5, Iterasi: 6

Binary Search - Cari NIM '2206'
Ditemukan di indeks: 5, Iterasi: 3
PS C:\VSCODE\103112400074_Modul 8>
```

Deskripsi Program:

Program diatas digunakan untuk mencari data mahasiswa menggunakan dua metode pencarian, yaitu **Sequential Search** dan **Binary Search**, dalam bahasa Go. Data mahasiswa disimpan dalam array bertipe struct yang memuat informasi seperti nama, NIM, kelas, jurusan, dan IPK. Untuk pencarian nama, program menggunakan **Sequential Search**, yaitu pencarian satu per satu dari awal hingga akhir data. Sementara itu, pencarian berdasarkan NIM menggunakan **Binary Search**, yaitu metode pencarian yang lebih efisien dengan syarat data sudah terurut. Program ini juga mencetak hasil pencarian beserta jumlah iterasi yang diperlukan oleh masing-masing metode, sehingga dapat dibandingkan efisiensinya.

III. UNGUIDED

Unguided 1

```
1 //Ahmad Ruba'i
2 //103112400074
3 package main
4
5 import "fmt"
6
7 func main() {
8     var suaraMasuk, suaraSah int
9     var suara [21]int
10    var x int
11
12    selesai := false
13    for !selesai {
14        n, _ := fmt.Scan(&x)
15        if n == 0 {
16            selesai = true
17        } else if x == 0 {
18            selesai = true
19        } else {
20            suaraMasuk++
21            if x >= 1 && x <= 20 {
22                suaraSah++
23                suara[x]++
24            }
25        }
26    }
27
28    fmt.Printf("Suara masuk: %d\n", suaraMasuk)
29    fmt.Printf("Suara sah: %d\n", suaraSah)
30    for i := 1; i <= 20; i++ {
31        if suara[i] > 0 {
32            fmt.Printf("%d: %d\n", i, suara[i])
33        }
34    }
35 }
```


Output

```
PS C:\VSCODE\103112400074_Modul 8> go r
7 19 3 2 78 3 1 -3 18 19 0
Suara masuk: 10
Suara sah: 8
1: 1
2: 1
3: 2
7: 1
18: 1
19: 2
PS C:\VSCODE\103112400074_Modul 8> |
```

Deskripsi Program:

Program ini digunakan untuk merekapitulasi hasil pemungutan suara dalam pemilihan Ketua RT. Program akan meminta Pengguna untuk memasukkan nomor calon satu per satu, dan proses input dihentikan dengan memasukkan angka 0. Program menghitung jumlah total suara yang masuk, jumlah suara sah serta mencatat jumlah suara yang diperoleh masing-masing calon. Program akan mencetak jumlah suara masuk, suara sah, dan daftar calon yang mendapatkan suara beserta jumlahnya.

Unguided 2

```
1 // Ahmad Ruba'i
2 // 103112400074
3 package main
4
5 import "fmt"
6
7 func main() {
8     var suaraMasuk, suaraSah int
9     var suara [21]int
10    var x int
11
12    selesai := false
13    for !selesai {
14        n, _ := fmt.Scan(&x)
15        if n == 0 {
16            selesai = true
17        } else if x == 0 {
18            selesai = true
19        } else {
20            suaraMasuk++
21            if x >= 1 && x <= 20 {
22                suaraSah++
23                suara[x]++
24            }
25        }
26    }
27
28    fmt.Println("Suara masuk:", suaraMasuk)
29    fmt.Println("Suara sah:", suaraSah)
30
31    max := 0
32    for i := 1; i <= 20; i++ {
33        if suara[i] > max {
34            max = suara[i]
35        }
36    }
37
38    var pemenang []int
39    for i := 1; i <= 20; i++ {
40        if suara[i] == max && max > 0 {
41            pemenang = append(pemenang, i)
42        }
43    }
44
45    ketua := 0
```

```

46  ✓   if len(pemenang) > 0 {
47      |   ketua = pemenang[0]
48      |   }
49
50      wakil := 0
51  ✓   if len(pemenang) > 1 {
52      |   wakil = pemenang[1]
53  ✓   } else {
54      |   max2 := 0
55  ✓   for i := 1; i <= 20; i++ {
56  ✓   |   if suara[i] > max2 && suara[i] < max {
57      |   |   max2 = suara[i]
58      |   |   }
59      |   }
60  ✓   for i := 1; i <= 20; i++ {
61  ✓   |   if suara[i] == max2 && max2 > 0 {
62      |   |   wakil = i
63      |   |   break
64      |   |   }
65      |   }
66      }
67
68      fmt.Println("Ketua RT:", ketua)
69      fmt.Println("Wakil ketua RT:", wakil)
70  }

```

Output

```

PS C:\VSCODE\103112400074_Modul 8> go
7 19 3 2 78 3 1 -3 18 19 0
Suara masuk: 10
Suara sah: 8
Ketua RT: 3
Wakil ketua RT: 19
PS C:\VSCODE\103112400074_Modul 8>

```

Deskripsi Program:

Program ini digunakan untuk menentukan ketua dan wakil ketua RT berdasarkan suara yang dimasukkan oleh pengguna. Program akan meminta pengguna untuk memasukkan nomor calon, dan proses input akan berhenti jika angka 0 dimasukkan. Program mencatat jumlah total suara yang masuk, menghitung berapa banyak yang sah, lalu menentukan calon dengan suara terbanyak sebagai ketua RT. Jika ada lebih dari satu calon dengan suara terbanyak, maka calon kedua menjadi wakil ketua. Jika hanya satu pemenang, maka wakil ketua diambil dari calon dengan suara terbanyak kedua. Program akan mencetak jumlah suara masuk, suara sah, serta nama ketua dan wakil ketua RT.

Unguided 3

```
1 //Ahmad Ruba'i
2 //103112400074
3 package main
4
5 import "fmt"
6
7 const NMAX = 1000000
8
9 var data [NMAX]int
10
11 func isiArray(n int) {
12     for i := 0; i < n; i++ {
13         fmt.Scan(&data[i])
14     }
15 }
16
17 func posisi(n, k int) int {
18     for i := 0; i < n; i++ {
19         if data[i] == k {
20             return i
21         }
22     }
23     return -1
24 }
25
26 func main() {
27     var n, k int
28     fmt.Scan(&n, &k)
29     isiArray(n)
30     idx := posisi(n, k)
31     if idx == -1 {
32         fmt.Println("TIDAK ADA")
33     } else {
34         fmt.Println(idx)
35     }
36 }
```

Output

```
PS C:\VSCODE\103112400074_Modul 8> go run "c:\V
12 534
1 3 8 16 32 123 323 323 534 543 823 999
8
PS C:\VSCODE\103112400074_Modul 8> go run "c:\V
12 535
1 3 8 16 32 123 323 323 534 543 823 999
TIDAK ADA
PS C:\VSCODE\103112400074_Modul 8> |
```

Deskripsi Program:

Program diatas digunakan untuk mencari posisi suatu bilangan tertentu dalam sebuah deretan angka. Program akan meminta pengguna untuk memasukan dua nilai, yaitu n yang menunjukkan jumlah elemen yang akan dimasukan kedalam array, dan k yaitu nilai yang ingin dicari posisinya. Kemudian, pengguna diminta menginput n buah angka kemudian disimpan ke dalam array data. Fungsi posisi untuk melakukan pencarian secara linear untuk menemukan indeks pertama dari nilai k. jika nilai tersebut ditemukan, program akan menampilkan indeksnya; jika tidak ditemukan, program akan mencetak "TIDAK ADA"

IV. KESIMPULAN

Praktikum ini membahas dua metode pencarian utama, yaitu **Sequential Search** dan **Binary Search**, serta penerapannya dalam bahasa pemrograman Go. **Sequential Search** cocok untuk data yang belum terurut karena mengecek setiap elemen secara berurutan, meskipun kurang efisien untuk data besar. Sementara itu, **Binary Search** lebih efisien dengan kompleksitas waktu $O(\log n)$, tetapi memerlukan data yang sudah terurut. Implementasi dalam Go menunjukkan bahwa **Sequential Search** berguna untuk pencarian sederhana seperti data mahasiswa atau pemilihan Ketua RT, sedangkan **Binary Search** lebih optimal untuk pencarian data terurut seperti NIM. Pemilihan metode harus mempertimbangkan kondisi data, ukuran dataset, dan kebutuhan efisiensi. Dengan memahami kedua metode ini, pengembang dapat memilih solusi terbaik sesuai kebutuhan aplikasi.

V. REFERENSI

Modul 8 praktikum alpro 2