

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2  
MODUL 11  
PENCARIAN NILAI ACAK DALAM HIMPUNAN DATA**



Disusun Oleh:

NAMA: MULIA AKBAR NANDA PRATAMA

NIM: 103112400034

KELAS: 12 IF 01

**PROGRAM STUDI S1 TEKNIK INFORMATIKA  
FAKULTAS INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2025/2026**

## I. DASAR TEORI

### a. Sequential Search

Pencarian nilai sekuensial ini adalah pencarian yang dilakukan dari data pertama hingga data terakhir secara satu persatu dan berurutan. Ciri khas dari pencarian ini adalah proses pencarian akan berhenti Ketika data yang dicari telah 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. Berikut ini adalah contoh notasi algoritma sekuensial search:

NOTASI ALGORITMA	NOTASI DALAM GO
<pre>found <math>\beta</math> false i <math>\beta</math> 0 while i &lt; n and not found do found <math>\beta</math> T[i] == X i <math>\beta</math> i + 1 endwhile</pre>	<pre>found = false i = 0 for i &lt; n &amp;&amp; !found { found = T[i] == X i = i + 1 }</pre>

### b. Binary Search

Ide algoritma ini adalah (dengan asumsi data terurut dari terkecil membesar (*ascending*), dan data dengan indeks kecil ada di kiri dan indeks besar ada di kanan). Algoritma ini dikenal dengan nama binary search. Berikut contoh notasi:

NOTASI ALGORITMA	NOTASI DALAM GO
kr $\beta$ 0 kn $\beta$ n-1 found $\beta$ false while kr $\leq$ kn and not found do med $\beta$ (kr+kn) div 2 if a[med] < X then kr $\beta$ med + 1 else if a[med] > X then kn $\beta$ med - 1 else found $\beta$ true endif endwhile	kr = 0 kn = n-1 found = false for kr $\leq$ kn && !found { med = (kr+kn) / 2 if a[med] < X { kr = med + 1 } else if a[med] > X { kn = med } else { found = true } }

## II. GUIDED

### 1. //program

```
package main

import (
    "fmt"
    "sort"
)

func sequentialSearch(arr[]float64, target float64)(int,
int){
    iterations := 0
    for i, val := range arr{
        iterations++
        fmt.Printf("sequential step %d: cek arr
[%d]=%.f\n", iterations, i, val)
        if val == target{
            return i, iterations
        }
    }
    return -1, iterations
}

func binarySEarch(arr []float64,target float64)(int,
int){
    iterations := 0
    low := 0
    high := len (arr)-1
    for low <= high {
        iterations++
        mid := (low + high)/ 2
        fmt.Printf("binary stop %d: cek arr[%d]= %.f\n",
iterations, mid, arr[mid])

        if arr[mid ]== target {
            return mid, iterations
        }else if target < arr[mid]{
            high = mid -1
        }else{

```

```

        low = mid - 1
    }
}
return -1, iterations
}
func main(){
    //array awal
    data:= []float64{2, 7, 9, 1, 5, 6, 18, 13, 25, 20}
    target := 13.0

    fmt.Println("sequential search (data tidak perlu
urut):")
    idxseq, iterseq := sequentialSearch(data, target)
    if idxseq != -1{
        fmt.Printf("hasil : ditemukan di indeks %d dalam
%d langkah\n\n",idxseq, iterseq)
    }else{
        fmt.Printf("hasil : tidak ditemukan setelah %d
langkah\n",iterseq)
    }

    //Binary earch array diurutkan
    sort.Float64s(data)
    fmt.Println("Binary search (setelah data diurutkan):")
    fmt.Println("Data terurut:", data)

    idxBin, iterBin := binarySEarch(data, target)
    if idxBin!= -1 {
        fmt.Printf("Hasil : Ditemukan indeks %d dalam %d
langkah\n", idxBin,iterBin)
    }else{
        fmt.Printf("Hasil : Tidak ditemukan Setelah %d
langkah\n",iterBin)
    }
}
}

```

//output

```
PS C:\Users\pratama> go run "c:\VSCODE\103112400034_MODULE8\GUIDED 1\tempCodeRunnerFile.go"
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:\Users\pratama> 
```

deskripsi: program di atas merupakan program yang dibuat dengan Bahasa go yang membandingkan dua metode pencarian yaitu sekuensial dan biner.

## 2. //program

```
package main

import (
    "fmt"
    "sort"
)

type mahasiswa struct {
    nama, nim, kelas, jurusan string
    ipk                          float64
}

type arrMhs [2023]mahasiswa

// Sequential Search berdasarkan nama
func SeqSearch_3(T arrMhs, n int, X string) (int, int) {
    var found int = -1
    var j int = 0
    var iterasi int = 0

    for j < n && found == -1 {
        iterasi++
        if T[j].nama == X {
            found = j
        }
        j++
    }
    return found, iterasi
}

// Binary Search berdasarkan NIM (data harus sudah
// terurut berdasarkan nim)
func BinarySearch_3(T arrMhs, n int, X string) (int, int)
{
    var found int = -1
    var med int
    var kr int = 0
    var kn int = n - 1
```

```

var iterasi int = 0

for kr <= kn && found == -1 {
    iterasi++
    med = (kr + kn) / 2
    if X < T[med].nim {
        kn = med - 1
    } else if X > T[med].nim {
        kr = med + 1
    } else {
        found = med
    }
}
return found, iterasi
}

func main() {
    var data arrMhs
    n := 10

    // Mengisi data secara manual
    data = arrMhs{
        {nama: "Ari", nim: "2201", kelas: "A", jurusan:
        "Informatika", ipk: 3.4},
        {nama: "Budi", nim: "2203", kelas: "A", jurusan:
        "Informatika", ipk: 3.6},
        {nama: "Cici", nim: "2202", kelas: "B", jurusan:
        "Sistem Informasi", ipk: 3.5},
        {nama: "Dina", nim: "2205", kelas: "A", jurusan:
        "Informatika", ipk: 3.3},
        {nama: "Eko", nim: "2204", kelas: "B", jurusan:
        "Sistem Informasi", ipk: 3.7},
        {nama: "Fajar", nim: "2206", kelas: "C", jurusan:
        "Informatika", ipk: 3.1},
        {nama: "Gita", nim: "2209", kelas: "C", jurusan:
        "Informatika", ipk: 3.8},
        {nama: "Hana", nim: "2208", kelas: "B", jurusan:
        "Sistem Informasi", ipk: 3.2},
    }
}

```



```

        {nama: "Iwan", nim: "2207", kelas: "C", jurusan:
        "Informatika", ipk: 3.0},
        {nama: "Joko", nim: "2210", kelas: "A", jurusan:
        "Informatika", ipk: 3.9},
    }

    // Pencarian Sequential Search berdasarkan nama
    namaDicari := "Fajar"
    idxSeq, iterSeq := SeqSearch_3(data, n, namaDicari)

    fmt.Printf("Sequential Search - Cari nama '%s'\n",
    namaDicari)
    if idxSeq != -1 {
        fmt.Printf("Ditemukan di indeks: %d, Iterasi:
        %d\n", idxSeq, iterSeq)
    } else {
        fmt.Printf("Tidak ditemukan, Iterasi: %d\n",
        iterSeq)
    }

    // Urutkan data berdasarkan NIM untuk binary search
    sort.Slice(data[:n], func(i, j int) bool {
        return data[i].nim < data[j].nim
    })

    // Pencarian Binary Search berdasarkan NIM
    nimDicari := "2206"
    idxBin, iterBin := BinarySearch_3(data, n,
    nimDicari)

    fmt.Printf("\nBinary Search - Cari NIM '%s'\n",
    nimDicari)
    if idxBin != -1 {
        fmt.Printf("Ditemukan di indeks: %d, Iterasi:
        %d\n", idxBin, iterBin)
    } else {
        fmt.Printf("Tidak ditemukan, Iterasi: %d\n",
        iterBin)
    }
}

```

//output

```
PS C:\Users\pratama> go run "c:\VSCODE\103112400034_MODUL8\GUIDED 2\tempCodeRunnerFile.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:\Users\pratama>
```

Deskripsi: program di atas dibuat menggunakan Bahasa go. Program Go ini mengelola data mahasiswa (disimpan dalam struct mahasiswaberisi nama, NIM, dll.) dan menunjukkan cara mencari data tersebut menggunakan pencarian sekuensial berdasarkan nama dan pencarian biner berdasarkan NIM. Fungsi mainpertama-tama melakukan pencarian sekuensial. Kemudian data mahasiswa diurutkan berdasarkan NIM, lalu dilakukan pencarian biner. Program ini secara ringkas membandingkan kedua metode pencarian pada data pelajar, menyoroti bahwa pencarian sekuensial tidak memerlukan pengurutan, sedangkan pencarian biner memerlukannya untuk efisiensi.

### III. UNGUIDED

#### 1. //program

```
package main

import "fmt"

func main() {
    var suara int
    hitungSuara := make(map[int]int)
    totalSuaraMasuk := 0
    totalSuaraSah := 0

    fmt.Println("NAMA: MULIA AKBAR NANDA PRATAMA\nNIM: 103112400034")
    fmt.Println("silahkan masukkan nomor calon:")

    for {
        fmt.Scan(&suara)
        totalSuaraMasuk++

        if suara == 0 {
            totalSuaraMasuk--
            break
        }

        if suara >= 1 && suara <= 20 {
            hitungSuara[suara] = hitungSuara[suara] + 1
            totalSuaraSah++
        }
    }

    fmt.Println("Suara masuk:", totalSuaraMasuk)
    fmt.Println("Suara sah:", totalSuaraSah)

    for calon, jumlah := range hitungSuara {
        if jumlah > 0 {
            fmt.Printf("%d: %d\n", calon, jumlah)
        }
    }
}
```

```
}  
}
```

//output

```
PS C:\VSCODE\103112400034_MODUL8\UNGUIDED 1> go run "c:\VSCODE\103112400034_MODUL8\UNGUIDED 1\un1.go"  
NAMA: MULIA AKBAR NANDA PRATAMA  
NIM: 103112400034  
silahkan masukkan nomor calon:  
7 19 3 2 78 3 1 -3 18 19 0  
Suara masuk: 10  
Suara sah: 8  
3: 2  
2: 1  
1: 1  
18: 1  
7: 1  
19: 2  
PS C:\VSCODE\103112400034_MODUL8\UNGUIDED 1>
```

Deskripsi: program di atas dibuat dengan Bahasa go. Program Go ini adalah alat hitung suara sederhana. User memasukkan nomor-nomor calon (angka 1 sampai 20) dan diakhiri dengan angka 0. Program kemudian akan menghitung berapa banyak suara sah yang masuk dan menampilkan jumlah suara yang diperoleh oleh masing-masing calon.

## 2. //program

```
package main

import "fmt"

func main() {
    var num, total, sah int
    suara := [21]int{}

    for {
        _, err := fmt.Scan(&num)
        if err != nil || num == 0 {
            break
        }
        total++
        if num >= 1 && num <= 20 {
            sah++
            suara[num]++
        }
    }

    ketua, wakil := 0, 0
    max1, max2 := 0, 0

    for i := 1; i <= 20; i++ {
        if suara[i] > max1 {
            max2 = max1
            wakil = ketua
            max1 = suara[i]
            ketua = i
        } else if suara[i] > max2 {
            max2 = suara[i]
            wakil = i
        }
    }

    fmt.Printf("suara yg masuk: %d\nsuara yg sah: %d\n", total, sah)
```

```
        fmt.Printf("Ketua RT: %d\nWakil ketua: %d\n",  
ketua, wakil)  
    }
```

//output

```
PS C:\VSCODE\103112400034_MODUL8\UNGUIDED 2> go run "c:\VSCODE\103112400034_MODUL8\UNGUIDED 2\un2.go"  
7 19 3 2 78 3 1 -3 18 19 0  
suara yg masuk: 10  
suara yg sah: 8  
Ketua RT: 3  
Wakil ketua: 19  
PS C:\VSCODE\103112400034_MODUL8\UNGUIDED 2> |
```

Deskripsi: Program yang dibuat menggunakan Bahasa Go ini menghitung hasil pemilihan ketua RT secara sederhana. program mencatat jumlah suara masuk dan suara sah, lalu menentukan pemenang berdasarkan suara terbanyak (nomor kecil menang jika seri). Ketua dipilih dari suara terbanyak pertama, wakil dari suara terbanyak kedua. Output menampilkan total suara, suara valid, serta ketua dan wakil terpilih

### 3. //program

```
package main

import "fmt"

const NMAX = 1000000

var data [NMAX]int

func main() {
    var n, k int
    fmt.Scan(&n, &k)
    isiArray(n)

    pos := posisi(n, k)
    if pos != -1 {
        fmt.Println(pos)
    } else {
        fmt.Println("TIDAK ADA")
    }
}

func isiArray(n int) {
    for i := 0; i < n; i++ {
        fmt.Scan(&data[i])
    }
}

func posisi(n, k int) int {
    for i := 0; i < n; i++ {
        if data[i] == k {
            return i
        }
    }
    return -1
}
```

//output

```
PS C:\VSCODE\103112400034_MODULE8\UNGUIDED 3> go run "c:\VSCODE\103112400034_MODULE8\UNGUIDED 3\un3.go"
12 534
1 3 8 16 32 123 323 323 534 543 823 999
8
PS C:\VSCODE\103112400034_MODULE8\UNGUIDED 3> go run "c:\VSCODE\103112400034_MODULE8\UNGUIDED 3\un3.go"
12 535
1 3 8 16 32 123 323 323 534 543 823 999
TIDAK ADA
PS C:\VSCODE\103112400034_MODULE8\UNGUIDED 3> |
```

Deskripsi: program yang dibuat dengan Bahasa go itu bertujuan untuk mencari posisi sebuah bilangan tertentu dalam sebuah daftar bilangan yang telah di urut secara menurun.



#### **IV. KESIMPULAN**

## **V. REFERENSI**

*MODUL 11. PENCARIAN NILAI ACAK PADA HIMPUNAN  
DATA TELKOM UNIVERSITY PURWOKERTO.*