

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
MODUL: 11
MATERI: PENCARIAN NILAI ACAK PADA HIMPUNAN DATA



DISUSUN OLEH:

NAMA: JESIKA METANIA RAHMA ARIFIN

NIM: 103112400080

KELAS: 12 IF 01

DOSEN:

Dimas Fanny Hebrisianto Permadi S.ST, M.Kom

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025/2026

DASAR TEORI

11.1 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	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	}

11.2 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")

1) Ambil salah satu data dalam rentang data yang ada, algoritma di bawah menggunakan rentang dari kiri  s.d. kanan . Untuk kemudahan dan alasan lainnya, biasanya yang diambil adalah paling tengah dalam rentang tersebut.

2) Jika data terambil tersebut terlalu kecil, maka ubah/geser rentang data ke sebelah kanan posisi data tersebut. Ini karena jika data terambil terlalu kecil, maka semua data sebelah kirinya juga akan terlalu kecil dari yang ingin dicari.

3) Begitu juga sebaliknya jika data terambil terlalu besar. Algoritma ini dikenal dengan nama Binary Search. Berikut ini adalah contoh notasi algoritma binary search sesuai dengan penjelasan di atas.

	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 $kr \leq kn$ and not found do	for $kr \leq kn \ \&\& \ !found \{$
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	}

Sebagai catatan algoritma binary search untuk array terurut membesar (ascending) akan berbeda dengan terurut mengecil (descending), sehingga algoritma ini tidak akan berjalan apabila terbalik. Misalnya array terurut secara membesar, tetapi algoritma binary search yang digunakan untuk array terurut mengecil. Hal ini mengakibatkan pencarian binary search tidak akan berhasil. Selanjutnya bagaimana contoh algoritma pencarian data apabila dituliskan ke dalam suatu fungsi pencarian dan diketahui array terurut secara mengecil atau descending.

11.3 Pencarian pada Array Bertipe Data Struct

Algoritma pencarian pada array bertipe data struct tidak berbeda jauh dengan tipe data dasar, kita hanya perlu menambahkan field kategori dari pencarian yang akan dilakukan. Khusus untuk algoritma binary search, maka keterurutan array harus sama dengan field kategori dari pencariannya. Misalnya apabila pencarian berdasarkan nim mahasiswa, maka algoritma binary search hanya akan bisa digunakan apabila array terurut berdasarkan nim. Algoritma binary search tidak akan berjalan apabila array terurut berdasarkan nim mahasiswa, tetapi pencarian yang dilakukan berdasarkan filed nama atau kategori yang lain selain nim. Solusinya adalah array harus diurutkan berdasarkan nim terlebih dahulu, atau gunakan algoritma sequential search. Berikut adalah contoh apabila array mahasiswa terurut berdasarkan nim secara membesar (ascending) dan dilakukan pencarian menggunakan algoritma sekuensial dan biner.

```
...
type mahasiswa struct {
nama, nim, kelas, jurusan string
ipk float64
}
type arrMhs [2023]mahasiswa
...
```

```

func SeqSearch_3(T arrMhs, n int, X string) int {
/* mengembalikan indeks mahasiswa dengan nama X, atau -1 apabila tidak ditemukan
pada array T yang berisi n data mahasiswa */
var found int = -1
var j int = 0
for j < n && found == -1 {
if T[j].nama == X {
found = j
}
j = j + 1
}
return found
}

func BinarySearch_3(T arrMhs, n int, X string) int {
/* mengembalikan indeks mahasiswa dengan nim X, atau -1 apabila tidak ditemukan
pada array T yang berisi n data mahasiswa dan terurut membesar berdasarkan nim
*/
var found int = -1
var med int
var kr int = 0
var kn int = n - 1
for kr <= kn && found == -1 {
med = (kr + kn) / 2
if X < T[med].nim {
kn = med - 1
}else if X > T[med].nim {

```

GUIDED 1:

```
package main

import (
    "fmt"
    "sort"
)

func sequentialSearch(arr []float64, target float64) (int, int) {
    iterations := 0
    for i, val := range arr {
        iterations++
        fmt.Printf("Squential Step %d: cek arr[%d] = %.1f\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 Step %d cek arr[%d] = %.1f\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() {
    data := []float64{2, 7, 9, 1, 5, 6, 18, 13, 25, 20}
    target := 13.0

    fmt.Println("Sequential Search (data tidak perluurut)")
    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\n", iterSeq)
    }
}

```

```

    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 di indeks %d dalam %d langkah\n", idxBin, iterBin)

    } else {

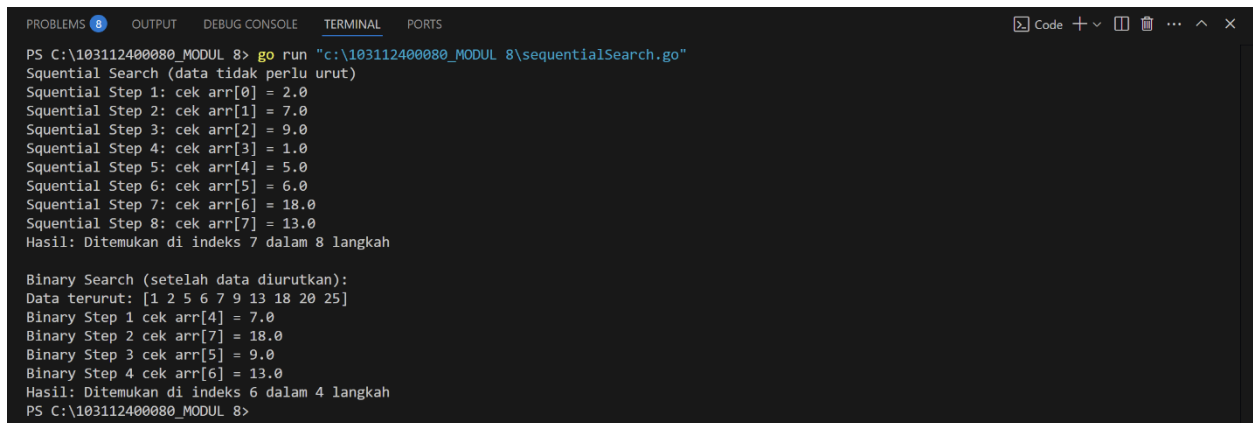
        fmt.Printf("Hasil: Tidak ditemukan setelah %d langkah\n", iterBin)

    }

}

```

OUTPUT:



```

PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400080_MODUL 8> go run "c:\103112400080_MODUL 8\sequentialSearch.go"
Sequential Search (data tidak perlu urut)
Sequential Step 1: cek arr[0] = 2.0
Sequential Step 2: cek arr[1] = 7.0
Sequential Step 3: cek arr[2] = 9.0
Sequential Step 4: cek arr[3] = 1.0
Sequential Step 5: cek arr[4] = 5.0
Sequential Step 6: cek arr[5] = 6.0
Sequential Step 7: cek arr[6] = 18.0
Sequential Step 8: cek arr[7] = 13.0
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 Step 1 cek arr[4] = 7.0
Binary Step 2 cek arr[7] = 18.0
Binary Step 3 cek arr[5] = 9.0
Binary Step 4 cek arr[6] = 13.0
Hasil: Ditemukan di indeks 6 dalam 4 langkah
PS C:\103112400080_MODUL 8>

```

DESKRIPSI PROGRAM:

Program ini adalah program yang dibuat untuk mengecek sequential searc dan binary search.

GUIDED 2:

```
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:

```

PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400080_MODUL 8> go run "c:\103112400080_MODUL 8\SequentialSearch2.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:\103112400080_MODUL 8>

```

DESKRIPSI PROGRAM:

Program ini adalah program yang dibuat untuk menampilkan sequential search untuk melihat “cari nama Fajar” dan ditemukan dalam iterasi ke berapa dalam program.

UNGUIDED 1:

Pada pemilihan ketua RT yang baru saja berlangsung, terdapat 20 calon ketua yang bertanding memperebutkan suara warga. Perhitungan suara dapat segera dilakukan karena warga cukup mengisi formulir dengan nomor dari calon ketua RT yang dipilihnya. Seperti biasa, selalu ada pengisian yang tidak tepat atau dengan nomor pilihan di luar yang tersedia, sehingga data juga harus divalidasi. Tugas Anda untuk membuat program mencari siapa yang memenangkan pemilihan ketua RT. Buatlah program pilkart yang akan membaca, memvalidasi, dan menghitung suara yang diberikan dalam pemilihan ketua RT tersebut. Masukkan hanya satu baris data saja, berisi bilangan bulat valid yang kadang tersisipi dengan data tidak valid. Data valid adalah integer dengan nilai di antara 1 s.d. 20 (inklusif). Data berakhir jika ditemukan sebuah bilangan dengan nilai 0. Keluaran dimulai dengan baris berisi jumlah data suara yang terbaca, diikuti baris yang berisi berapa banyak suara yang valid. Kemudian sejumlah baris yang mencetak data para calon apa saja yang mendapatkan suara.

```
// JESIKA METANIA RAHMA ARIFIN
```

```
// 103112400080
```

```
package main
```

```
import (  
    "fmt"  
)
```

```
func main() {  
    var suara [21]int  
    var angka int  
    suaraMasuk := 0  
    suaraSah := 0
```

```
for {  
    _ , err := fmt.Scan(&angka)  
    if err != nil || angka == 0 {  
        break  
    }  
  
    suaraMasuk++  
  
    if angka >= 1 && angka <= 20 {  
        suara[angka]++  
        suaraSah++  
    }  
}  
  
fmt.Printf("Suara masuk: %d\n", suaraMasuk)  
fmt.Printf("Suara sah: %d\n", suaraSah)  
  
for i := 1; i <= 20; i++ {  
    if suara[i] > 0 {  
        fmt.Printf("%d: %d\n", i, suara[i])  
    }  
}  
}
```

OUTPUT:

A screenshot of a Go IDE terminal window. The terminal shows the execution of a Go program. The prompt is 'PS C:\103112400080_MODUL 8>'. The user enters 'go run "c:\103112400080_MODUL 8\Unguided1.go"'. The program outputs: '7 19 3 2 78 3 1 -3 18 19 0', 'Suara masuk: 10', 'Suara sah: 8', and a list of candidates: '1: 1', '2: 1', '3: 2', '7: 1', '18: 1', '19: 2'. The prompt returns to 'PS C:\103112400080_MODUL 8>'.

```
PS C:\103112400080_MODUL 8> go run "c:\103112400080_MODUL 8\Unguided1.go"
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:\103112400080_MODUL 8>
```

DESKRIPSI PROGRAM:

Program ini adalah program yang di buat untuk menghitung suara yang masuk dan suara sah dalam pemilihan ketua RT pada suatu desa.

UNGUIDED 2:

Berdasarkan program sebelumnya, buat program pilkart yang mencari siapa pemenang pemilihan ketua RT. Sekaligus juga ditentukan bahwa wakil ketua RT adalah calon yang mendapatkan suara terbanyak kedua. Jika beberapa calon mendapatkan suara terbanyak yang sama, ketua terpilih adalah dengan nomor peserta yang paling kecil dan wakilnya dengan nomor peserta terkecil berikutnya. Masukan hanya satu baris data saja, berisi bilangan bulat valid yang kadang tersisipi dengan data tidak valid. Data valid adalah bilangan bulat dengan nilai di antara 1 s.d. 20 (inklusif). Data berakhir jika ditemukan sebuah bilangan dengan nilai 0. Keluaran dimulai dengan baris berisi jumlah data suara yang terbaca, diikuti baris yang berisi berapa banyak suara yang valid. Kemudian tercetak calon nomor berapa saja yang menjadi pasangan ketua RT dan wakil ketua RT yang baru.

```
// JESIKA METANIA RAHMA ARIFIN
```

```
// 103112400080
```

```
package main
```

```
import "fmt"
```

```
func main() {
```

```
    var suara [21]int
```

```
    var angka int
```

```
    jumlahMasuk := 0
```

```
jumlahSah := 0

for {
    fmt.Scan(&angka)
    if angka == 0 {
        break
    }
    jumlahMasuk++

    if angka >= 1 && angka <= 20 {
        suara[angka]++
        jumlahSah++
    }
}

fmt.Println("Suara masuk:", jumlahMasuk)
fmt.Println("Suara sah:", jumlahSah)

// Cari ketua (suara terbanyak) dan wakil (terbanyak kedua)
ketua := 0
wakil := 0
for i := 1; i <= 20; i++ {
    if suara[i] > suara[ketua] || (suara[i] == suara[ketua] && i < ketua) {
        wakil = ketua
        ketua = i
    }
}
```

```

        } else if (suara[i] > suara[wakil] || (suara[i] == suara[wakil] && i < wakil)) && i !=
ketua {

        wakil = i

    }

}

fmt.Println("Ketua RT:", ketua)

fmt.Println("Wakil Ketua RT:", wakil)

}

```

OUTPUT:

```

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400080_MODUL 8> go run "c:\103112400080_MODUL 8\Unguided2.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:\103112400080_MODUL 8>

```

DESKRIPSI PROGRAM:

Program ini adalah program yang dibuat untuk melihat suara yang masuk, suara sah, dan penentuan ketua dan wakil ketua RT.

UNGUIDED 3:

Diberikan n data integer positif dalam keadaan terurut membesar dan sebuah integer lain k, apakah bilangan k tersebut ada dalam daftar bilangan yang diberikan? Jika ya, berikan indeksnya, jika tidak sebutkan "TIDAK ADA". Masukan terdiri dari dua baris. Baris pertama berisi dua buah integer positif, yaitu n dan k. n menyatakan banyaknya data, dimana $1 < n \leq 1000000$. k adalah bilangan yang ingin dicari. Baris kedua berisi n buah data integer positif yang sudah terurut membesar. Keluaran terdiri dari satu baris saja, yaitu sebuah bilangan yang menyatakan posisi data yang dicari (k) dalam kumpulan data yang diberikan. Posisi data dihitung dimulai dari angka 0. Atau memberikan keluaran "TIDAK ADA" jika data k tersebut tidak ditemukan dalam kumpulan. Program yang dibangun harus menggunakan subprogram dengan mengikuti kerangka yang sudah diberikan berikut ini.

```
// JESIKA METANIA RAHMA ARIFIN
```

```
// 103112400080
```



```
package main
```

```
import "fmt"
```

```
const NMAX = 1000000
```

```
var data [NMAX]int
```

```
func main() {
```

```
    var n, k int
```

```
    fmt.Scan(&n, &k)    // baca jumlah data dan angka yang dicari
```

```
    isiArray(n)        // isi array dengan n bilangan
```

```
    pos := posisi(n, k) // cari posisi 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 {

    low := 0

    high := n - 1

    for low <= high {

        mid := (low + high) / 2

        if data[mid] == k {

            return mid

        } else if data[mid] < k {

            low = mid + 1

        } else {

            high = mid - 1

        }

    }

    return -1

}

```

OUTPUT:

```

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400080_MODUL 8> go run "c:\103112400080_MODUL 8\Unguided3.go"
12 534
1 3 8 16 32 123 323 323 534 543 823 999
8
PS C:\103112400080_MODUL 8> go run "c:\103112400080_MODUL 8\Unguided3.go"
12 535
1 3 8 16 32 123 323 323 534 543 823 999
TIDAK ADA
PS C:\103112400080_MODUL 8>

```

DESKRIPSI PROGRAM:

Program ini adalah program yang di buat untuk melihat ada dan tidaknya bilangan bulat terurut membesar.

KESIMPULAN

Pada modul ini, telah dipelajari cara melakukan pencarian nilai acak dalam suatu himpunan data. Proses ini melibatkan pemahaman terhadap struktur data seperti array atau list, serta penerapan algoritma pencarian yang efisien. Dengan menggunakan fungsi acak (random), kita dapat mengambil nilai secara acak dari himpunan data yang tersedia, baik untuk tujuan sampling, simulasi, maupun analisis statistik. Penggunaan metode pencarian seperti *linear search* atau *binary search* juga dibahas untuk menemukan nilai tertentu dalam data. Melalui praktik ini, pemahaman tentang manipulasi data dan penggunaan fungsi acak dalam pemrograman menjadi lebih mendalam dan aplikatif.