

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
MODUL 11 (PENCARIAN NILAI ACAK
PADA HIMPUNAN DATA)**



Oleh:

ANGGUN WAHYU WIDIYANA

103112480280

12-IF-01

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
TAHUN AJARAN 2025/2026**

TEORI DASAR

Algoritma Sekuensial Search (Sequential Search)

Sekuensial search adalah metode pencarian data dengan cara memeriksa setiap elemen secara berurutan dari awal hingga akhir. Metode ini tidak memerlukan data terurut dan cocok untuk data yang ukurannya kecil atau tidak terstruktur.

```
func sequentialSearch(data []int, target int) int {
    for i, v := range data {
        if v == target {
            return i // indeks ditemukan
        }
    }
    return -1 // tidak ditemukan
}
```

Tabel 1.1 Contoh Sequential Search

Algoritma Binary Search

Binary search adalah metode pencarian yang efisien untuk data yang sudah terurut. Algoritma ini bekerja dengan membagi dua ruang pencarian secara berulang, membandingkan nilai tengah dengan nilai yang dicari, dan mempersempit pencarian ke bagian kiri atau kanan.

```
func binarySearch(data []int, target int) int {
    low, high := 0, len(data)-1
    for low <= high {
        mid := (low + high) / 2
        if data[mid] == target {
            return mid
        } else if data[mid] < target {
            low = mid + 1
        } else {
            high = mid - 1
        }
    }
    return -1
}
```

Tabel 1.2 Contoh Binary Search

Pencarian pada Array Bertipe Data Struct

Dalam Go, data kompleks dapat disimpan dalam array bertipe struct. Pencarian dilakukan dengan membandingkan atribut tertentu dalam struct, biasanya menggunakan sekuensial search. Jika data terurut berdasarkan atribut tersebut, binary search juga dapat diterapkan.

GUIDED

Contoh Soal 1

Source Code:

```
//Nama : Anggun Wahyu Widiyana (103112480280)
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] = %.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

```

PS C:\Users\User\Documents\VALPRO 2\103112480280_MODULE8> go run "c:\Users\User\Documents\VALPRO 2\103112480280_MODULE8\103112480280_Guided8\1.go"
Sequential Search (data tidak perluurut)
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:\Users\User\Documents\VALPRO 2\103112480280_MODULE8> go run "c:\Users\User\Documents\VALPRO 2\103112480280_MODULE8\103112480280_Guided8\1.go"

```

Deskripsi Program

Program ini mengimplementasikan dua metode pencarian nilai dalam sebuah array bertipe data `float64` menggunakan bahasa Go: **sekuensial search** dan **binary search**. Pada bagian pertama, program melakukan pencarian secara sekuensial pada data yang belum diurutkan. Setiap langkah pencarian dicetak untuk menunjukkan elemen yang sedang diperiksa, dan program mengembalikan indeks elemen yang ditemukan beserta jumlah iterasi yang dilakukan.

Setelah itu, data diurutkan secara ascending menggunakan fungsi `sort.Float64s()` dari package `sort`. Kemudian, program melakukan pencarian binary search pada data yang sudah terurut. Binary search bekerja dengan membagi ruang pencarian menjadi dua bagian pada setiap langkah, dan juga menampilkan langkah-langkah pencarian secara detail. Hasil pencarian berupa indeks dan jumlah iterasi juga ditampilkan.

Program ini memberikan gambaran jelas tentang perbedaan efisiensi antara sekuensial search dan binary search, serta pentingnya pengurutan data sebelum melakukan binary search..

Contoh Soal 2

Source Code:

```
//Nama : Anggun Wahyu Widiyana (103112480280)
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
        }
    }
    return found, iterasi
}
```

```

        } 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\User\Documents\ALPRO 2\103112480280_MODULE8> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE8\103112480280_Guided8\2.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\User\Documents\ALPRO 2\103112480280_MODULE8> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE8\103112480280_Guided8\2.go"

```

Deskripsi Program

Program ini mengelola data mahasiswa menggunakan tipe data struct dalam bahasa Go, yang berisi atribut seperti nama, NIM, kelas, jurusan, dan IPK. Program menyediakan dua metode pencarian data: **Sequential Search** berdasarkan nama mahasiswa dan **Binary Search** berdasarkan NIM.

Pada bagian pertama, program melakukan pencarian secara sekuensial untuk menemukan indeks mahasiswa berdasarkan nama yang dicari. Metode ini memeriksa setiap elemen secara berurutan hingga menemukan nama yang sesuai atau mencapai akhir data.

Sebelum melakukan binary search, data diurutkan berdasarkan NIM menggunakan fungsi `sort.Slice`. Binary search kemudian digunakan untuk mencari mahasiswa berdasarkan NIM dengan cara membagi ruang pencarian menjadi dua bagian secara berulang hingga menemukan data yang dicari atau ruang pencarian habis.

Program menampilkan hasil pencarian beserta jumlah iterasi yang dilakukan, sehingga memberikan gambaran efisiensi kedua metode pencarian pada data bertipe struct.

UNGUIDED

Soal 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.

Masukan 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.

No .	Masukan	Keluaran
1	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

Source Code:

```
//Nama : Anggun Wahyu Widiyana (103112480280)
package main

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

func main() {
    const maxCalon = 20
    var suara [maxCalon]int
    totalData := 0
    totalValid := 0

    fmt.Println("Masukkan data suara (akhiri dengan 0):")
```

```

scanner := bufio.NewScanner(os.Stdin)
if scanner.Scan() {
    input := scanner.Text()
    data := strings.Fields(input)

    for _, v := range data {
        num, err := strconv.Atoi(v)
        if err != nil {
            continue
        }
        if num == 0 {
            break
        }
        totalData++
        if num >= 1 && num <= maxCalon {
            suara[num-1]++
            totalValid++
        }
    }
}

fmt.Printf("Suara masuk: %d\n\n", totalData)
fmt.Printf("Suara sah: %d\n\n", totalValid)
for i, count := range suara {
    if count > 0 {
        fmt.Printf("%d:%d\n", i+1, count)
    }
}
}

```

Output

```

PS C:\Users\User\Documents\VALPRO 2\103112480280_MODULE8> go run "c:\Users\User\Documents\VALPRO 2\103112480280_MODULE8\103112480280_Unguided8\1.go"
Masukkan data suara (akhiri dengan 0):
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:\Users\User\Documents\VALPRO 2\103112480280_MODULE8> go run "c:\Users\User\Documents\VALPRO 2\103112480280_MODULE8\103112480280_Unguided8\1.go"

```

Deskripsi Program

Program ini menerima input data suara pemilihan ketua RT dalam satu baris, yang berisi nomor calon dengan kemungkinan adanya data tidak valid. Program menghitung total suara yang masuk dan memvalidasi suara yang sah, yaitu nomor calon antara 1 hingga 20. Validasi dilakukan dengan memeriksa setiap angka secara berurutan (sekuensial search) tanpa perlu mengurutkan data. Setelah validasi, program menyimpan jumlah suara sah untuk setiap calon dalam array dan menampilkan hasil berupa total suara masuk, total suara sah, dan jumlah suara untuk setiap calon yang menerima suara..

Soal 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 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.

No.	Masukan	Keluaran
1	7 19 3 2 78 3 1 -3 18 19 0	Suara masuk: 10 Suara sah: 8 Ketua RT: 3 Wakil ketua: 19

Source Code

```
//Nama : Anggun Wahyu Widiyana (103112480280)
package main

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

func main() {
    const maxCalon = 20
    var suara [maxCalon]int
    totalData, totalValid := 0, 0

    fmt.Println("Masukkan suara (akhiri dengan 0):")
    scanner := bufio.NewScanner(os.Stdin)
    if scanner.Scan() {
        input := scanner.Text()
        data := strings.Fields(input)

        for _, v := range data {
            num, err := strconv.Atoi(v)
            if err != nil {
                continue
            }
            if num == 0 {
```

```

        break
    }
    totalData++
    if num >= 1 && num <= maxCalon {
        suara[num-1]++
        totalValid++
    }
}

// Cari suara terbanyak (ketua)
max1 := -1
ketua := 0
for i := 0; i < maxCalon; i++ {
    if suara[i] > max1 {
        max1 = suara[i]
        ketua = i + 1
    } else if suara[i] == max1 && (i+1) < ketua {
        ketua = i + 1
    }
}

// Cari suara terbanyak kedua (wakil), nomor terkecil dan
bukan ketua
max2 := -1
wakil := 0
for i := 0; i < maxCalon; i++ {
    if i+1 == ketua {
        continue
    }
    if suara[i] > max2 {
        max2 = suara[i]
        wakil = i + 1
    } else if suara[i] == max2 && (i+1) < wakil {
        wakil = i + 1
    }
}

fmt.Printf("Suara masuk: %d\n\n", totalData)
fmt.Printf("Suara sah: %d\n\n", totalValid)
fmt.Printf("Ketua RT: %d\n\n", ketua)
fmt.Printf("Wakil ketua: %d\n", wakil)
}

```

Output

```

PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE8> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE8\103112480280_Unguided8\2.go"
Masukkan suara (akhiri dengan 0):
7 19 3 2 78 3 1 -3 18 19 0
Suara masuk: 10

Suara sah: 8

Ketua RT: 3

Wakil ketua: 19
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE8> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE8\103112480280_Unguided8\2.go"

```

Deskripsi Program

Program ini menggunakan metode pencarian sekuensial untuk menentukan pemenang pemilihan ketua RT dan wakil ketua RT berdasarkan jumlah suara yang masuk. Program membaca data suara yang dimasukkan oleh pengguna, memvalidasi suara yang sah (nomor calon antara 1 sampai 20), dan menghitung total suara yang masuk serta suara yang sah. Selanjutnya, program mencari calon dengan suara terbanyak sebagai ketua RT dan calon dengan suara terbanyak kedua sebagai wakil ketua RT. Jika terdapat calon dengan jumlah suara yang sama, program memilih nomor calon terkecil sesuai aturan. Pendekatan sekuensial search ini sesuai dengan modul pencarian nilai acak pada himpunan data, karena pencarian dilakukan dengan memeriksa setiap elemen secara berurutan untuk menentukan pemenang..

Soal 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.

```
package main

import "fmt"

const NMAX = 1000000 var data [NMAX]int

func main(){
/* buatlah kode utama yang membaca baris pertama (n dan k).
kemudian data diisi
oleh prosedur isiArray(n), dan pencarian oleh fungsi posisi(n,k),
dan
setelah
itu output dicetak. */
}
func isiArray(n int){
/* I.S. terdefinisi integer n, dan sejumlah n data sudah siap pada
piranti masukan.
F.S. Array data berisi n (<=NMAX) bilangan */
}

func posisi(n, k int) int {
/* mengembalikan posisi k dalam array data dengan n elemen. Posisi
dimulai dari
posisi 0. Jika tidak ada kembalikan -1 */
}
```

Contoh masukan dan keluaran:

No.	Masukan	Keluaran	Penjelasan
-----	---------	----------	------------

1	12 534 1 3 8 16 32 123 323 323 534 543 823 999	8	Data 534 berada pada posisi ke-8 dihitung dari awal data.
2	12 535 1 3 8 16 32 123 323 323 534 543 823 999	TIDAK ADA	

Source Code

```
//Nama : Anggun Wahyu Widiyana (103112480280)
package main

import "fmt"

const NMAX = 1000000

var data [NMAX]int

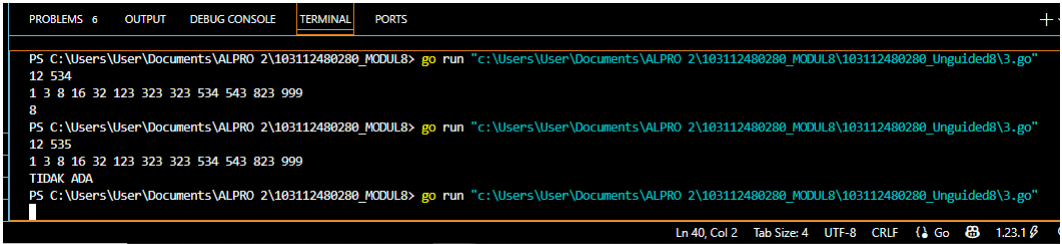
func main() {
    var n, k int
    fmt.Scan(&n, &k)
    isiArray(n)
    pos := posisi(n, k)
    if pos == -1 {
        fmt.Println("TIDAK ADA")
    } else {
        fmt.Println(pos)
    }
}

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

func posisi(n, k int) int {
    low, high := 0, 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



```
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE8> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE8\103112480280_Unguided8\3.go"  
12 534  
1 3 8 16 32 123 323 323 534 543 823 999  
8  
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE8> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE8\103112480280_Unguided8\3.go"  
12 535  
1 3 8 16 32 123 323 323 534 543 823 999  
TIDAK ADA  
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE8> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE8\103112480280_Unguided8\3.go"
```

Deskripsi Program

Program ini menggunakan metode **binary search**, yang lebih efisien untuk data yang sudah terurut. Setelah membaca input n , k , dan data array, fungsi `isiArray` mengisi array dengan data. Fungsi `posisi` melakukan pencarian biner dengan cara membagi ruang pencarian menjadi dua bagian pada setiap iterasi. Nilai tengah array dibandingkan dengan k , jika sama maka posisi tengah dikembalikan. Jika k lebih besar, pencarian dilanjutkan pada bagian kanan; jika lebih kecil, pada bagian kiri. Jika tidak ditemukan, fungsi mengembalikan `-1`.

KESIMPULAN

Metode pencarian sekuensial dan binary search merupakan teknik fundamental dalam pencarian nilai acak pada himpunan data. Pencarian sekuensial melakukan pemeriksaan elemen satu per satu tanpa memerlukan data terurut, sehingga sangat fleksibel dan mudah diimplementasikan, terutama untuk dataset kecil atau tidak terurut. Namun, metode ini memiliki efisiensi yang rendah dengan kompleksitas waktu $O(n)$. Sebaliknya, binary search memanfaatkan data yang sudah terurut untuk melakukan pencarian secara efisien dengan kompleksitas $O(\log n)$, melalui pembagian ruang pencarian secara berulang. Oleh karena itu, binary search sangat direkomendasikan untuk dataset besar yang sudah terurut, sedangkan sekuensial search lebih cocok untuk data kecil atau yang belum diurutkan. Pemilihan metode yang tepat sangat penting untuk mengoptimalkan performa pencarian sesuai karakteristik data.

REFERENSI

Informatika. (2025). *Algoritma dan Pemrograman 2 (Pencarian Nilai Acak pada Himpunan Data. Modul-11*. Informatika.

Widi Astuti, *Penggunaan Algoritma Sequential Searching pada Aplikasi*, Jurnal STKIP PGRI Tulungagung,
<https://jurnal.stkippgritulungagung.ac.id/index.php/jipi/article/download/2646/1164>

Atika Riesta, *Modul Pertemuan 5 - Sequential Search*, Fakultas Teknologi Informasi Universitas Budi Luhur,
<http://atikariesta.budiluhur.blog/wp-content/uploads/2020/03/Modul-Pertemuan-5-Sequential-Search.pdf>