

**LAPORAN PRAKTIKUM  
ALGORITMA PEMROGRAMAN 2**

**MODUL 11**

**PENCARIAN NILAI ACAK PADA HIMPUNAN**



Oleh:

**IQBAL BAWANI**

2311102130

S1IF-11-07

**S1 TEKNIK INFORMATIKA**

**INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

**2024**

## I. DASAR TEORI

### Sequential Search

Sequential Search adalah metode pencarian yang dilakukan secara berurutan dari elemen pertama hingga terakhir dalam sebuah array. Proses pencarian berhenti ketika data yang dicari ditemukan, meskipun masih ada elemen yang belum dicek. Algoritma ini bekerja dengan memeriksa setiap elemen array satu per satu mulai dari indeks awal hingga data ditemukan atau mencapai elemen terakhir. Misalkan terdapat array of integer  $T$  dengan indeks dari 0 hingga  $N-1$  dan nilai yang dicari  $X$ . Status pencarian, misalnya menggunakan variabel boolean *found*, akan diperbarui saat elemen array sesuai dengan nilai  $X$ . Perulangan akan dihentikan ketika *found* bernilai *true* (data ditemukan) atau semua elemen telah diperiksa.

### 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.

## II. GUIDED

### Guided1

#### Source code:

```
// guided 1
package main
import (
    "fmt"
)
// Definisi struct mahasiswa
type mahasiswa struct {
    nama    string
    nim     string
    kelas   string
    jurusan string
    ipk     float64
}

// Definisi array bertipe struct mahasiswa
type arrMhs [2023]mahasiswa

// Sequential Search berdasarkan nama
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
}

// Binary Search berdasarkan nim
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 {
            kr = med + 1
        } else {
            found = med
        }
    }
    return found
}

// Fungsi utama untuk demonstrasi
func main() {
    var data arrMhs
    var n int

    // Input jumlah mahasiswa
    fmt.Print("Masukkan jumlah mahasiswa: ")
    fmt.Scanln(&n)

    // Input data mahasiswa
    fmt.Println("Masukkan data mahasiswa:")
    for i := 0; i < n; i++ {
        fmt.Printf("Mahasiswa %d\n", i+1)
        fmt.Print("Nama: ")
        fmt.Scanln(&data[i].nama)
        fmt.Print("NIM: ")
        fmt.Scanln(&data[i].nim)
        fmt.Print("Kelas: ")
        fmt.Scanln(&data[i].kelas)
        fmt.Print("Jurusan: ")
        fmt.Scanln(&data[i].jurusan)
        fmt.Print("IPK: ")
        fmt.Scanln(&data[i].ipk)
    }
}

```

```

// Pencarian berdasarkan nama
var cariNama string
fmt.Print("Masukkan nama mahasiswa yang ingin dicari: ")
fmt.Scanln(&cariNama)
posNama := SeqSearch_3(data, n, cariNama)
if posNama == -1 {
    fmt.Println("Mahasiswa dengan nama tersebut tidak
ditemukan.")
} else {
    fmt.Printf("Mahasiswa ditemukan pada indeks: %d\n",
posNama)
}

// Pencarian berdasarkan nim
var cariNIM string
fmt.Print("Masukkan NIM mahasiswa yang ingin dicari: ")
fmt.Scanln(&cariNIM)
posNIM := BinarySearch_3(data, n, cariNIM)
if posNIM == -1 {
    fmt.Println("Mahasiswa dengan NIM tersebut tidak
ditemukan.")
} else {
    fmt.Printf("Mahasiswa ditemukan pada indeks: %d\n",
posNIM)
}
}

```

**Output =**

```

1353      fmt.Scanln(&cariNama)
1354      posNama := SeqSearch_3(data, n, cariNama)

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

NIM: 2311102130
Kelas: 07
Jurusan: informatika
IPK: 4.0
Mahasiswa 2
Nama: bawani
NIM: 2311102230
Kelas: 07
Jurusan: informatik
IPK: 3.9
Masukkan nama mahasiswa yang ingin dicari: bawani
Mahasiswa ditemukan pada indeks: 1
Masukkan NIM mahasiswa yang ingin dicari: 2311102130
Mahasiswa ditemukan pada indeks: 0
PS C:\Users\PLN\Documents\tugas ittp s2\prakstrukkedua\2311102130_Iqbal-

```

### **Penjelasan =**

Kode Program ini menggunakan dua metode pencarian, yaitu sequential search dan binary search, untuk mencari data mahasiswa berdasarkan nama atau NIM. Pada sequential search, pencarian dilakukan dengan memeriksa elemen array satu per satu dari awal hingga akhir. Jika nama mahasiswa yang dicari ditemukan, fungsi mengembalikan indeksinya, dan jika tidak, mengembalikan `-1`. Metode ini sederhana dan cocok untuk data yang tidak terurut, tetapi menjadi kurang efisien jika jumlah data sangat besar karena memerlukan pengecekan terhadap setiap elemen.

Sebaliknya, binary search digunakan untuk mencari mahasiswa berdasarkan NIM, dengan asumsi data sudah terurut. Metode ini lebih cepat karena langsung memeriksa elemen tengah array, kemudian mempersempit pencarian ke bagian kiri atau kanan berdasarkan hasil perbandingan. Proses ini terus diulang hingga data ditemukan atau tidak ditemukan sama sekali, dengan waktu pencarian yang jauh lebih singkat dibandingkan sequential search. Dengan dua metode ini, program dapat menyesuaikan pencarian tergantung pada kondisi dan struktur data.

## **Guided 2**

### **Source code :**

```
//guided2
package main

import "fmt"
// Define arrInt as an array of integers with a fixed size
```

```

type arrInt [4321]int
// Binary search function for descending sorted array
func BinarySearch_2(T arrInt, n int, X int) int {
    /* Mengembalikan indeks dari X apabila X ditemukan di dalam array T
       yang berisi n buah bilangan bulat terurut secara descending/menciu,
       atau -1 apabila X tidak ditemukan */
    var found int = -1
    var med int
    var kr int = 0
    var kn int = n - 1

    // Iterasi mencari elemen dengan binary search
    for kr <= kn && found == -1 {
        med = (kr + kn) / 2
        if X > T[med] { // Karena descending, jika X lebih besar,
            bergerak ke kiri
            kn = med - 1
        } else if X < T[med] { // Jika X lebih kecil, bergerak ke kanan
            kr = med + 1
        } else { // Jika ditemukan
            found = med
        }
    }
    return found
}

func main() {
    // Contoh penggunaan
    var data arrInt
    var n int
    // Input jumlah elemen
    fmt.Print("Masukkan jumlah elemen dalam array: ")
    fmt.Scanln(&n)
    // Input elemen array (harus terurut secara descending)
    fmt.Println("Masukkan elemen array (harus terurut menurun):")
    for i := 0; i < n; i++ {
        fmt.Printf("Elemen %d: ", i+1)
        fmt.Scanln(&data[i])
    }

    // Input elemen yang dicari
    var search int
    fmt.Print("Masukkan elemen yang ingin dicari: ")
    fmt.Scanln(&search)

    // Panggil fungsi binary search

```

```

        result := BinarySearch_2(data, n, search)

        // Cetak hasil
        if result == -1 {
            fmt.Println("Elemen tidak ditemukan dalam array.")
        } else {
            fmt.Printf("Elemen ditemukan pada indeks: %d\n", result)
        }
    }
}

```

**Output =**

```

PS C:\Users\PLN\Documents\tugas ittp s2\prakstrukkedua\23111
02130_Iqbal-Bawani> go run "c:\Users\PLN\Documents\tugas TUP
sem3\laprak alpro2\praktikum3modul6 rekrusif\Untitled-2.go"

Masukkan jumlah elemen dalam array: 4
Masukkan elemen array (harus terurut menurun):
Elemen 1: 3
Elemen 2: 2
Elemen 3: 1
Elemen 4: 0
Masukkan elemen yang ingin dicari: 3
Elemen ditemukan pada indeks: 0
PS C:\Users\PLN\Documents\tugas ittp s2\prakstrukkedua\23111
02130_Iqbal-Bawani> 

```

**Penjelasan =**

Program ini menggunakan algoritma binary search untuk mencari elemen dalam array yang sudah terurut menurun (descending). Pengguna memasukkan elemen array secara berurutan dari besar ke kecil, lalu memasukkan elemen yang ingin dicari. Algoritma ini bekerja dengan membagi array menjadi dua bagian setiap langkah: jika elemen yang dicari lebih besar dari elemen tengah, pencarian bergerak ke kiri; jika lebih kecil, ke kanan. Hasilnya adalah indeks elemen yang dicari jika ditemukan, atau `-1` jika tidak ada. Binary search sangat cepat untuk array besar asalkan sudah terurut.



### Guided 3

#### Source code :

```
//guided3
package main
import "fmt"
// Define arrStr as an array of strings with a fixed size
type arrStr [1234]string
// Sequential search function
func SeqSearch_1(T arrStr, n int, X string) int {
    /* Mengembalikan indeks dari X apabila X ditemukan di dalam array T
       yang berisi n buah teks, atau -1 apabila X tidak ditemukan */
    var found int = -1
    var j int = 0

    // Iterasi mencari elemen yang cocok
    for j < n && found == -1 {
        if T[j] == X {
            found = j
        }
        j = j + 1
    }
    return found
}
func main() {
    // Contoh penggunaan
    var data arrStr
    var n int

    // Input jumlah elemen
    fmt.Print("Masukkan jumlah elemen dalam array: ")
    fmt.Scanln(&n)

    // Input elemen array
    fmt.Println("Masukkan elemen array:")
    for i := 0; i < n; i++ {
        fmt.Printf("Elemen %d: ", i+1)
        fmt.Scanln(&data[i])
    }
    // Input elemen yang dicari
    var search string
    fmt.Print("Masukkan elemen yang ingin dicari: ")
    fmt.Scanln(&search)
    // Panggil fungsi sequential search
    result := SeqSearch_1(data, n, search)
```

```

    // Cetak hasil
    if result == -1 {
        fmt.Println("Elemen tidak ditemukan dalam array.")
    } else {
        fmt.Printf("Elemen ditemukan pada indeks: %d\n", result)
    }
}

```

**Output =**

```

PROBLEMS 1 TERMINAL ... Code + - Code Editor Icons
PS C:\Users\PLN\Documents\tugas ittp s2\prakstrukkedua\23111
02130_Iqbal-Bawani> go run "c:\Users\PLN\Documents\tugas TUP
sem3\laprak alpro2\praktikum3modul6 rekrusif\Untitled-2.go"

Masukkan jumlah elemen dalam array: 4
Masukkan elemen array:
Elemen 1: 54
Elemen 2: 59
Elemen 3: 31
Elemen 4: 2
Masukkan elemen yang ingin dicari: 31
Elemen ditemukan pada indeks: 2
PS C:\Users\PLN\Documents\tugas ittp s2\prakstrukkedua\23111
02130_Iqbal-Bawani>

```

**Penjelasan =**

Program ini menggunakan sequential search untuk mencari elemen dalam array string. Pengguna diminta memasukkan elemen array dan elemen yang ingin dicari. Algoritma pencarian ini bekerja dengan memeriksa setiap elemen dalam array satu per satu dari awal hingga menemukan elemen yang sesuai. Jika elemen yang dicari ditemukan, program akan mengembalikan indeksinya; jika tidak, akan mengembalikan -1. Metode ini sederhana dan mudah diimplementasikan, cocok untuk array kecil atau data yang tidak terurut, tetapi menjadi kurang efisien jika jumlah data besar.

## UNGUIDED

### Uguided ke1

Soal : 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.

#### Source code

```
//iqbal bawani
package main

import (
    "fmt"
)

type Calon struct {
    nomor int
    suara int
}

func main() {
    var calonkapten [20]Calon
    for i := 0; i < 20; i++ {
        calonkapten[i] = Calon{nomor: i + 1, suara: 0}
    }

    var totalSuara, suaraValid int

    fmt.Println("Masukkan data suara (pisahkan dengan spasi, akhiri dengan 0):")
    var angka int
```

```

for {
    fmt.Scan(&angka)

    if angka == 0 {
        break
    }

    totalSuara++

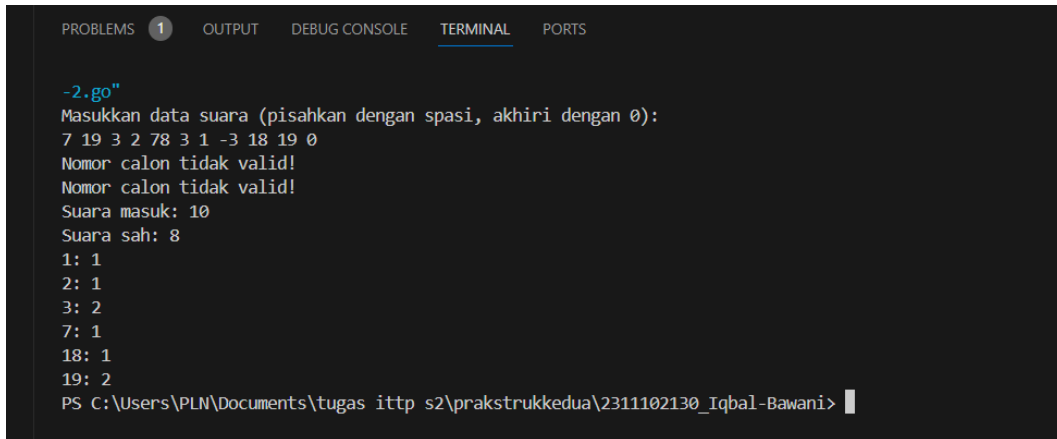
    // Sequential Search untuk mencari calon yang sesuai
    found := false
    for i := 0; i < len(calonkapten); i++ {
        if calonkapten[i].nomor == angka {
            calonkapten[i].suara++
            suaraValid++
            found = true
            break
        }
    }

    // Jika calon tidak valid
    if !found {
        fmt.Println("Nomor calon tidak valid!")
    }
}

fmt.Printf("Suara masuk: %d\n", totalSuara)
fmt.Printf("Suara sah: %d\n", suaraValid)
for _, calon := range calonkapten {
    if calon.suara > 0 {
        fmt.Printf("%d: %d\n", calon.nomor, calon.suara)
    }
}
}

```

## Output =



```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
-2.go"
Masukkan data suara (pisahkan dengan spasi, akhiri dengan 0):
7 19 3 2 78 3 1 -3 18 19 0
Nomor calon tidak valid!
Nomor calon tidak valid!
Suara masuk: 10
Suara sah: 8
1: 1
2: 1
3: 2
7: 1
18: 1
19: 2
PS C:\Users\PLN\Documents\tugas ittp s2\prakstrukkedua\2311102130_Iqbal-Bawani>
```

## Penjelasan =

Program ini menggunakan sequential search untuk mencari calon berdasarkan nomor yang dimasukkan oleh pengguna. Setiap kali suara dimasukkan, program akan mencari calon yang memiliki nomor yang sesuai dengan suara yang diberikan. Pencarian dilakukan dengan memeriksa elemen array calonkapten satu per satu, membandingkan nomor calon di setiap indeks dengan angka yang dimasukkan. Jika ditemukan calon dengan nomor yang cocok, suara calon tersebut akan ditambah dan dianggap sebagai suara sah. Jika nomor calon tidak ditemukan dalam array, program akan memberi peringatan bahwa nomor calon tidak valid.

## Unguided Ke 2

Soal : 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.

### Source code

```
//iqbalbawani
package main

import (
    "fmt"
    "sort"
)

type Calon struct {
    nomor int
    suara int
}

func main() {
    var calonkapten [20]Calon
    for i := 0; i < 20; i++ {
        calonkapten[i] = Calon{nomor: i + 1, suara: 0}
    }

    var totalSuara, suaraValid int

    fmt.Println("Masukkan data suara (pisahkan dengan spasi, akhiri dengan 0):")
    var angka int

    // Membaca input suara
    for {
        fmt.Scan(&angka)

        if angka == 0 {
            break
        }

        totalSuara++

        // Pencarian menggunakan sequential search untuk
        validasi suara
        if angka >= 1 && angka <= 20 {
            calonkapten[angka-1].suara++
            suaraValid++
        }
    }

    // Menampilkan total suara dan suara valid
    fmt.Printf("Suara masuk: %d\n", totalSuara)
```

```

        fmt.Printf("Suara sah: %d\n", suaraValid)

        // Mengurutkan calon berdasarkan suara terbanyak, jika sama
        berdasarkan nomor yang lebih kecil
        sort.SliceStable(calonkapten[:], func(i, j int) bool {
            if calonkapten[i].suara == calonkapten[j].suara {
                return calonkapten[i].nomor <
calonkapten[j].nomor
            }
            return calonkapten[i].suara > calonkapten[j].suara
        })

        // Mencetak ketua RT dan wakil ketua RT
        fmt.Printf("Ketua RT: %d\n", calonkapten[0].nomor)
        fmt.Printf("Wakil ketua: %d\n", calonkapten[1].nomor)
    }

```

## Output =

```

PS C:\Users\PLN\Documents\tugas ittp s2\prakstrukkedua\2311102130_Iqbal-Bawani> go run
\Untitled-2.go
Masukkan data suara (pisahkan dengan spasi, 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\PLN\Documents\tugas ittp s2\prakstrukkedua\2311102130_Iqbal-Bawani>

```

## Penjelasan =

Program ini digunakan untuk menghitung dan menentukan siapa yang terpilih sebagai ketua dan wakil ketua RT berdasarkan data suara yang dimasukkan. Program membaca data suara yang dipisahkan oleh spasi dan berhenti saat angka `0` dimasukkan. Setiap suara yang valid (berada di antara 1 hingga 20) akan menambah jumlah suara calon yang sesuai. Setelah itu, program menampilkan total suara yang masuk dan jumlah suara sah. Untuk menentukan ketua dan wakil ketua RT, program mengurutkan calon berdasarkan suara terbanyak, dan jika ada yang memiliki jumlah suara yang sama, urutan berdasarkan nomor calon yang lebih kecil. Hasilnya adalah nomor ketua dan wakil ketua RT yang terpilih.

### Unguided Ke 3

Soal : Masalah Pencarian Bilangan dalam Daftar Terurut

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  $1 < n \leq 1.000.000$ .  $k$  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 data yang diberikan.

#### Source code

```
// iqbalbawani

unguided 3
package main

import "fmt"

const NMAX = 1000000

var data [NMAX]int

// Fungsi untuk mengisi array dengan data
func isiArray(n int) {
    for i := 0; i < n; i++ {
        fmt.Scan(&data[i])
    }
}

// Fungsi untuk mencari posisi bilangan k dalam array data
func posisi(n, k int) int {
    // Implementasi Binary Search karena data sudah terurut
    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 // Jika tidak ditemukan
}

func main() {
    // Membaca n dan k
    var n, k int
    fmt.Scan(&n, &k)

    // Mengisi array data
    isiArray(n)

    // Mencari posisi k dalam array
    pos := posisi(n, k)

    // Menampilkan hasil
    if pos != -1 {
        fmt.Println(pos)
    } else {
        fmt.Println("TIDAK ADA")
    }
}

```

**Output =**

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\PLN\Documents\tugas ittp s2\prakstrukkedua\2311102130_Iqbal-Bawa
\Untitled-2.go"
12 534
1 3 8 16 32 123 323 323 534 543 823 999
8
PS C:\Users\PLN\Documents\tugas ittp s2\prakstrukkedua\2311102130_Iqbal-Bawa

```

**Penjelasan =**

Program di atas menggunakan binary search untuk mencari posisi bilangan `k` dalam array `data`. Hal ini terlihat dari implementasi fungsi `posisi`, yang menggunakan dua variabel `low` dan `high` untuk membagi array menjadi dua bagian secara berulang, yang merupakan ciri khas dari algoritma binary search. Binary search hanya dapat digunakan jika data dalam array sudah terurut, dan dalam kasus ini, diasumsikan bahwa array `data` sudah dalam urutan terurut. Algoritma ini mencari elemen dengan membandingkan nilai tengah array dan kemudian menentukan apakah pencarian akan dilanjutkan di setengah bagian kiri atau kanan berdasarkan perbandingan tersebut.

## **DAFTAR PUSTAKA**

*Modul Praktikum 11 Pencarian Nilai Acak Pada Himpunan*