

**LAPORAN RENCANA PRAKTIKUM  
STRUKTUR DATA**



**NAMA : INDRA FIQI RIPANI**  
**NIM : 213010503002**  
**KELAS : F**  
**MODUL : VI (PENCARIAN (*SEARCHING*))**

**Program Studi S1 Teknik Informatika**  
**Fakultas Teknik**  
**Universitas Palangka Raya**  
**Palangka Raya, Kalimantan Tengah**  
**2022**

## MODUL VI

### PENCARIAN (*SEARCHING*)

#### I. TUJUAN PRAKTIKUM

1. Mahasiswa memahami bagaimana logika yang digunakan dalam setiap jenis-jenis *searching*.
2. Mahasiswa mampu menerapkan algoritma *searching* ke dalam program.

#### II. LANDASAN TEORI

Algoritma pencarian atau *searching* adalah metode untuk menemukan nilai (data) tertentu dalam sekumpulan data yang bertipe sama. Contohnya untuk mengubah (*update*) data tertentu, langkah pertama yang harus dilakukan adalah mencari keberadaan data tersebut di dalam kumpulannya. Jika data yang dicari ditemukan, maka data tersebut dapat diubah nilainya dengan data yang baru. Pada modul ini akan dibahas algoritma pencarian Beruntun (*Sequential Search*), Bagi Dua (*Binary Search*), dan Knuth Moris Prath String Search.

##### 1. *Sequential Search*

Gagasan pada algoritma pencarian Beruntun ini adalah membandingkan setiap elemen larik satu per satu secara beruntun, mulai dari elemen pertama, sampai elemen yang dicari ditemukan, atau seluruh elemen larik sudah diperiksa. Berikut ini adalah pseudocode untuk fungsi pencarian beruntun.

```
Function SeqSearch(input L: LarikInt, input n:integer, input x:integer) →  
Boolean {mengembalikan nilai true jika x ditemukan di dalam larik L[1..n],  
atau nilai false jika x tidak  
ditemukan.}  
DEKLARASI  
i: integer {pencatat indeks larik}  
ALGORITMA:  
i ← 1  
while (i < n) and (L[i] ≠ x) do  
i ← i + 1 endwhile  
{i = n or L[i] = x} If L[i] = x then {x ditemukan}  
Return true  
Else Return false {x tidak ditemukan di dalam larik L }  
Endif
```

## 2. Selection Search

Gagasan pada algoritma pencarian Bagi Dua adalah membagi data atas dua bagian sebelum melakukan pencarian. Untuk memudahkan proses pencarian elemen elemen data sudah dalam kondisi terurut. Dalam proses pencarian Bagi Dua, diperlukan dua buah indeks larik, yaitu indeks terkecil dan indeks terbesar. Berikut ini adalah pseudocode untuk fungsi pencarian Bagi Dua.

```
Function BinarySearch (input L : LarikInt, input n:integer, input x: integer)
→integer {mengembalikan indeks elemen larik L[1..n] yang mengandung
nilai x; bila x tidak ditemukan, maka indek yang dikembalikan adalah -1}

DEKLARASI
i, j : integer {indeks kiri dan indek kanan larik}
k: integer {indeks elemen tengah}
ketemu: boolean {flag untuk menentukan apakah X ditemukan}
ALGORITMA:
i←1 {ujung kiri larik}
j←n {ujung kanan larik}
ketemu ← false {asumsikan x belum ditemukan}
while(not ketemu) and (i≤j) do
k ← (i+j) div 2 {bagidua larik L pada posisi k}
if(L[k] = x) then ketemu ← true
else { L[k] ≠ x } if(L[k] > x) then {Lakukan pencarian pada larik bagian
kanan, set indeks ujung kiri larik yang baru}
i←k+1 else {Lakukan pencarian pada larik bagian kiri, set indeks ujung
kanan larik yang baru}
j←k-1
endif
endif
endwhile {ketemu = true or i>j}
If ketemu then {x ditemukan} return k else {x tidak ditemukan di dalam
larik}
return -1
endif
```

## 3. Knuth Moris Prath String Search

Knuth Moriss Prath (KMP) merupakan salah satu algoritma yang digunakan untuk melakukan pencocokan string atau mencari string tertentu

dalam suatu kumpulan teks. Algoritma KMP secara umum adalah seperti berikut.

- 1) Pencocokan pola dimulai pada awal teks, dari kiri ke kanan. KMP akan mencocokkan karakter per karakter di dalam pola dengan karakter-karakter pada teks yang bersesuaian, sampai salah satu kondisi berikut dipenuhi. Sisipkan  $y$  pada tempat yang sesuai di antara  $L[1]...L[i]$ .
  - a. Karakter pada pola dan teks yang dibandingkan tidak cocok.
  - b. Semua karakter pada pola cocok, kemudian posisi penemuan pola pada teks diberitahukan.
- 2) Proses pencocokan pola akan dilakukan sampai pola berada di ujung teks. Algoritma KMP melakukan proses awal (*preprocessing*) terhadap pola  $P$  dengan menghitung fungsi pinggiran. Pada beberapa literatur disebut fungsi overlap, fungsi failure, fungsi awalan, dan sebagainya. Fungsi ini mengindikasikan pergeseran Pola yang mungkin dengan menggunakan perbandingan yang dibentuk sebelum pencarian string. Berikut ini pseudocode untuk menghitung Pinggiran dalam bentuk prosedur, pada bahasa pemrograman C++ prosedur adalah fungsi yang tidak menghasilkan nilai balik.

```
Procedure HitungPinggiran (input m: integer, P:string)
{Menghitung nilai pinggiran b[1..m] untuk pola P[1..m]}

DEKLARASI:
    k, g : integer

ALGORITMA:
    b[] = array[1..m] of integer
    b[0] = 0
    q=1
    k=0

    for q ← 1 to m-1 do
        While ((k>0) and (P[q] ≠P[k]))
            Do
                k← b[k]
            endwhile
            if P[q] = P[k]
                k←k+1
            endif

        b[q] ←k
    endfor
```

```

Function KMPSearch(input m: integer, n: integer, P: string, T: string) → boolean
{mengembalikan true jika pola P ditemukan dalam teks T}

DEKLARASI:
    i, j: integer
    ketemu: boolean

ALGORITMA:
    HitungPinggiran(m,P)
    i ← 0
    j ← 0
    ketemu = false

    while ((j > 0) and (P[j] ≠ T[i]))
    do
        j ← b[j-1]
    endwhile
    if P[j] = T[i] then
        j ← j+1
    endif
    if j = m then
        ketemu ← true
    else
        i ← i+1
    endif
    return ketemu

```

### III. TUGAS

1. Toko A memiliki sistem untuk menyimpan pembeli berdasarkan ID dari si pembeli. ID pembeli dibuat berdasarkan jam menit dan detik. Sebagai contoh, pembeli A membeli barang pada 19:54, maka ID pelanggan adalah 1954. Buatlah program untuk mencari ID pelanggan dari minimal 10 data dengan metode Sequential Search dan Binary Search. Bila tidak ditemukan akan ditampilkan pesan "Data Tidak Ditemukan!". Bila data ditemukan, akan ditampilkan pesan "Data Ditemukan pada indeks ke-n".

2. Buatlah program untuk mencari kata “mimpimu” dan “coba” dari kalimat “mimpimu memiliki akhir yang tidak terbatas, tarik napas yang dalam dan coba lagi”.