Modul 6 Searching

1.1 Kompetensi

1. Mahasiswa mampu menjelaskan mengenai algoritma *Searching*.
2. Mahasiswa mampu membuat dan mendeklarasikan struktur algoritma *Searching*.
3. Mahasiswa mampu menerapkan dan mengimplementasikan algoritma *Searching*.

1.2 Alat Dan Bahan:

- PC/Leptop
- IDE Bahasa Pemrogaman C++

1.3 Ulasan Teori:

Searching merupakan proses untuk menemukan suatu data atau informasi dari sekumpulan data/informasi yang ada. Algoritma pencarian/*searching* algorithm merupakan algoritma yang menerima suatu kata kunci sebagai kriteria pencarian, dan dengan langkah-langkah tertentu akan mencari rekaman yang sesuai dengan kata kunci tersebut. Di dalam modul ini akan diperkenalkan 2 teknik pencarian data yaitu pencarian beruntun/sequential searching dan pencarian biner/binary search

1.3.1. Pencarian Beruntun/*Sequential Search*

Sequential Search adalah teknik pencarian data dimana data dicari secara urut dari depan ke belakang atau dari awal sampai akhir. Kelebihan dari proses pencarian secara sequential ini jika data yang dicari terletak didepan, maka data akan ditemukan dengan cepat. Tetapi dibalik kelebihannya ini, teknik ini juga memiliki kekurangan. Pertama, jika data yang dicari terletak dibelakang atau paling akhir, maka akan membutuhkan waktu yang lama dalam proses pencariannya. Kedua, beban komputer akan semakin bertambah jika jumlah data dalam array sangat banyak.

Teknik pencarian data dari array yang paling mudah adalah dengan cara *sequential search*, dimana data dalam array dibaca 1 demi satu, diurutkan dari index terkecil ke index terbesar, maupun sebaliknya.

Contoh :

Array :

int a[5] = {0,3,6,10,1} (index array pada bahasa C dimulai dari index ke 0) jika kita ingin mencari bilangan 6 dalam array tersebut, maka proses yang terjadi kita mencari

a. dari array index ke-0, yaitu 0, dicocokan dengan bilangan yang akan dicari, jika tidak sama, maka mencari ke index berikutnya
b. pada array index ke-1, juga bukan bilangan yang dicari, maka kita mencari lagi pada index berikutnya
c. pada array index ke-2, ternyata bilangan yang kita cari ada ditemukan, maka kita keluar dari looping pencarian.

1.3.2. Binary Search

Metode pencarian yang kedua adalah *binary search*, pada metode pencarian ini, bila data belum terutur maka data harus diurutkan terlebih dahulu. Pencarian Biner (*Binary Search*) dilakukan untuk :

- Memperkecil jumlah operasi pembandingan yang harus dilakukan antara data yang dicari dengan data yang ada khususnya untuk jumlah data yang sangat besar ukurannya.
- Prinsip dasarnya adalah melakukan proses pembagian ruang pencarian secara berulang-ulang sampai data ditemukan atau sampai ruang pencarian tidak dapat dibagi lagi (berarti ada kemungkinan data tidak ditemukan).
- Syarat utama untuk pencarian biner adalah data harus sudah terurut, misalkan terurut menaik.

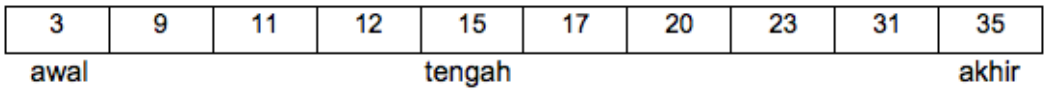
Berikut ini prinsip-prinsip pada metode *binary search*:

1. Data sudah urut
2. Ambil posisi awal = 0 dan posisi akhir = n-1
3. Cari posisi tengah = (awal+akhir)/2
4. Bandingkan data yang dicari dengan data yang di posisi tengah
5. Jika lebih kecil, maka proses pencarian dilakukan kembali tetapi posisi akhir diubah menjadi tengah-1
6. Jika lebih besar, maka proses pencarian dilakukan kembali tetapi posisi awal diubah menjadi tengah+1

7. Pencarian biner ini akan berakhir ketika data ditemukan atau posisi awal>akhir

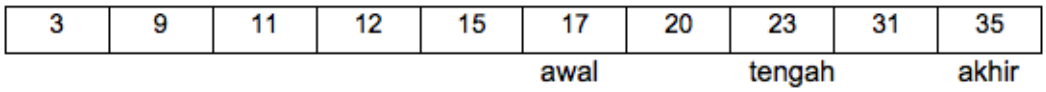
Contoh:

Langkah 1



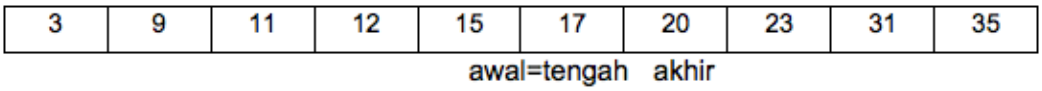
Awal = 0, akhir = 9, tengah = (0+9)/2 = 4

Langkah 2



- Kunci yang dicari adalah nilai 17
- Bandingkan 17 dengan data yang di tengah.
- Karena lebih besar maka nilai awal = tengah+1 = 5, akhir = 9, tengah = (5+9)/2 = 7

Langkah 3



- Bandingkan 17 dengan data yang di tengah (23).
- Karena lebih kecil maka nilai awal = 5, akhir = tengah-1 = 6, tengah = (5+6)/2 = 5

Langkah 4

- Bandingkan 17 dengan data yang di tengah (17).
- Ketemu

Keunggulan utama dari algoritma *binary search* adalah kompleksitas algoritmanya yang lebih kecil daripada kompleksitas algoritma sequential search. Hal ini menyebabkan waktu yang dibutuhkan algoritma binary search dalam mencari sebuah record dalam sebuah tabel lebih kecil daripada waktu yang dibutuhkan algoritma *sequential search*.