



SEARCHING

Pencarian (Searching)

- Pada suatu data seringkali dibutuhkan pembacaan kembali informasi (*information retrieval*) dengan cara searching.
- Searching adalah proses pencarian data yang ada pada suatu deret data dengan cara menelusuri data-data tersebut.
- Tahapan paling penting pada searching: memeriksa jika data yang dicari sama dengan data yang ada pada deret data.

Algoritma Pencarian

1. Input x (x =data yang dicari)
2. Bandingkan x dengan deret data
3. Jika ada data yang sama cetak pesan “Ada”
4. Jika tidak ada data yang sama cetak pesan “tidak ada”.

Algoritma Pencarian

Macam algoritma pencarian :

- Sequential Search
- Binary Search

Sequential Search

Merupakan teknik yang sederhana dan langsung dapat digunakan pada struktur data baik array maupun linked-list.

Pencarian data **secara urut** mulai dari data pertama sampai kunci yang dicari ditemukan atau sampai seluruh data telah dicari dan tidak ditemukan

Dilakukan pada data yang tidak terurut

(I) Sequential Search

- Disebut juga linear search atau Metode pencarian beruntun.
- Tidak efisien untuk data yang list yang besar
- Adalah suatu teknik pencarian data yang akan menelusuri tiap elemen satu per-satu dari awal sampai akhir.
- Data awal = tidak harus dalam kondisi terurut.

Case

- Best case : jika data yang dicari terletak di indeks array terdepan (elemen array pertama) sehingga waktu yang dibutuhkan untuk pencarian data sangat sebentar (minimal).
- Worst case : jika data yang dicari terletak di indeks array terakhir (elemen array terakhir) sehingga waktu yang dibutuhkan untuk pencarian data sangat lama (maksimal).

Ilustrasi Sequential Search

- Misalnya terdapat array satu dimensi sebagai berikut:

0	1	2	3	4	5	6	7	indeks
8	10	6	-2	11	7	1	100	value

- Kemudian program akan meminta data yang akan dicari, misalnya **6 ($x = 6$)**.
- Iterasi :
 - $6 = 8$ (tidak!)
 - $6 = 10$ (tidak!)
 - $6 = 6$ (Ya!) => output : “Ada” pada index ke-2
- Jika sampai data terakhir tidak ditemukan data yang sama maka output : “ data yang dicari tidak ada”.

Best & Worst Case

- Contoh :

DATA = 5 6 9 2 8 1 7 4

bestcase ketika $x = 5$

worstcase ketika $x = 4$

* x = key/data yang dicari

Contoh Sequential Search

	Nim	Nama	IPK
[0]	135410001	Mulyadi	2.94
[1]	135410005	Willy Johan	3.15
[2]	135410003	Anthony Liberty	2.78
[3]	135410004	Ferry Santoso	3.37
[4]	135410002	Jaya Mulya	2.93
[5]	135410007	Budi Santoso	3.01
[6]	135410006	Indra Gunawan	3.56
[7]	135410008	M. Rudito W	3.44

Contoh Sequential Search

Kunci pencarian? 135410004

NIM[0] == kunci? → tidak

NIM[1] == kunci? → tidak

NIM[2] == kunci? → tidak

NIM[3] == kunci? → ya → Ferry

Santoso, 3.37

Contoh Sequential Search

Kunci pencarian? 135410010

NIM[0] == kunci? → tidak

NIM[1] == kunci? → tidak

NIM[2] == kunci? → tidak

NIM[3] == kunci? → tidak

NIM[4] == kunci? → tidak

NIM[5] == kunci? → tidak

NIM[6] == kunci? → tidak

NIM[7] == kunci? → tidak

Semua data telah di cari, kunci tidak ditemukan

Algoritma 8.1. Linear Search

function pencarianLinier(input array : larik; kunci, ukuran : integer) : integer

Deklarasi

ketemu : boolean

i, n : integer

Deskripsi

ketemu \leftarrow false { belum ketemu }

n \leftarrow 1 { mulai dari elemen pertama }

while ((n < ukuran) and (not ketemu)) do

 if (array[n] = kunci) then { dibandingkan }

 ketemu \leftarrow true { data ketemu }

 i \leftarrow n { pada posisi ke-i, posisi disimpan }

 endif

 else n \leftarrow n+1 { cek data berikutnya }

endwhile

if ketemu then pencarianLinier \leftarrow i { data ketemu pada posisi ke-i }

else pencarianLinier \leftarrow -1 { data tidak ketemu }

endif

end

Bahasa C++

```
#include <iostream.h>
#define UKURAN 100
int pencarianLinier(int array[], int kunci, int ukuran)
{   int i;
    for (i=0; i<=ukuran-1; ++i)
        if (array[i] == kunci)
            return i;
    return -1;}

```



Thank YOU!!!

