



ALGORITMA SEARCHING

Wahyu Cahyo Utomo, S.Kom., M.Cs.

Searching Algorithm

Linear
Search

Binary
Search

LINEAR SEARCH ALGORITHM

- Linear Search disebut juga dengan Sequential Search.
- Teknik pencarian data dengan menelusuri semua data satu per satu.
- Dalam pencarian Linear, kita cukup menelusuri daftar sepenuhnya dan mencocokkan setiap elemen daftar dengan item yang lokasinya dapat ditemukan.
- Apabila ditemukan kecocokan data, maka program akan mengembalikan output, jika tidak pencarian akan terus berlanjut hingga akhir dari array tersebut.

Langkah-Langkah Linear Search

1. Pertama, kita harus lakukan perulangan elemen array menggunakan for loop.
2. Dalam setiap perulangan for loop, bandingkan elemen pencarian dengan setiap elemen array.
 - Jika elemen cocok, maka kembalikan indeks elemen array yang sesuai.
 - Jika elemen tidak cocok, maka pindah ke elemen selanjutnya.
3. Jika tidak ada kecocokan atau elemen pencarian tidak ada dalam array yang diberikan, kembalikan -1.

Implementasi Linear Search

0	1	2	3	4	5	6	7	8
70	40	30	11	57	41	25	14	52



Element Array yang akan dicari
Adalah **K = 41**

Implementasi Linear Search

0	1	2	3	4	5	6	7	8
70	40	30	11	57	41	25	14	52

↑
 $K \neq 70$

0	1	2	3	4	5	6	7	8
70	40	30	11	57	41	25	14	52

↑
 $K \neq 40$

0	1	2	3	4	5	6	7	8
70	40	30	11	57	41	25	14	52

↑
 $K \neq 30$

0	1	2	3	4	5	6	7	8
70	40	30	11	57	41	25	14	52

↑
 $K \neq 11$

0	1	2	3	4	5	6	7	8
70	40	30	11	57	41	25	14	52

↑
 $K \neq 57$

0	1	2	3	4	5	6	7	8
70	40	30	11	57	41	25	14	52

↑
 $K = 41$

Implementasi Linear Search

0	1	2	3	4	5	6	7	8
70	40	30	11	57	41	25	14	52

↑
K=41



Nilai K = 41 berada di indeks ke - 5

BINARY SEARCH ALGORITHM

- Binary Search adalah teknik pencarian yang bekerja secara efisien pada daftar yang diurutkan.
- Untuk mencari elemen ke dalam beberapa daftar data menggunakan teknik pencarian biner, kita harus memastikan bahwa daftar tersebut diurutkan.
- Binary Search membagi data menjadi 2 bagian, dan item dibandingkan dengan data elemen tengah.
- Jika terjadi kecocokan, maka lokasi elemen tengah akan dikembalikan.

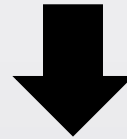
Langkah-Langkah Binary Search

1. Baca elemen pencarian dari pengguna.
2. Temukan elemen tengah dalam daftar yang diurutkan.
3. Bandingkan elemen pencarian dengan elemen tengah dalam daftar yang diurutkan.
4. Jika keduanya cocok, maka tampilkan bahwa pesan bahwa elemen sudah ditemukan.
5. Jika tidak cocok, maka periksa apakah elemen pencarian lebih kecil atau lebih besar dari elemen tengah.
6. Jika elemen pencarian lebih kecil dari elemen tengah, ulangi langkah 2, 3, 4 dan 5 untuk sublist kiri elemen tengah.
7. Jika elemen pencarian lebih besar dari elemen tengah, ulangi langkah 2, 3, 4 dan 5 untuk sublist kanan elemen tengah.
8. Ulangi proses yang sama sampai kita menemukan elemen pencarian dalam daftar atau sampai sublist hanya berisi satu elemen.



Implementasi Binary Search

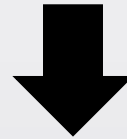
0	1	2	3	4	5	6	7	8
10	12	24	29	39	40	51	56	69



Element Array yang akan dicari
Adalah **K = 56**

Implementasi Binary Search

0	1	2	3	4	5	6	7	8
10	12	24	29	39	40	51	56	69



Diketahui:

$l = 0$

$h = 8$

$mid = (l + h) / 2 \rightarrow (0 + 8) / 2 = 4$

Implementasi Binary Search

0	1	2	3	4	5	6	7	8
10	12	24	29	39	40	51	56	69

↑

Nilai tengah = 39


tengah < K \rightarrow 39 < 56

l = mid + 1 = 5, h = 8

mid = (l + h) / 2 \rightarrow (5 + 8) / 2 = 6

Implementasi Binary Search

0	1	2	3	4	5	6	7	8
10	12	24	29	39	40	51	56	69



Nilaitengah = 51


Nilaitengah < K \rightarrow 51 < 56

l = mid + 1 = 7, h = 8

mid = (l + h) / 2 \rightarrow (5 + 8) / 2 = 7

Implementasi Binary Search

0	1	2	3	4	5	6	7	8
10	12	24	29	39	40	51	56	69



Nilaitengah = 56

Nilaitengah = K \rightarrow 56 = 56

elemen ditemukan di indeks ke - 7

LINEAR SEARCH VS BINARY SEARCH

Dasar Perbandingan	Linear Search	Binary Search
Definisi	Linear Search mulai mencari dari elemen pertama dan membandingkan setiap elemen dengan elemen yang dicari sampai elemen tersebut tidak ditemukan	Binary Search menemukan posisi elemen yang dicari dengan menemukan elemen tengah array
Sorted Data	Elemen tidak perlu diurutkan	Elemen harus urut
Implementasi	Diimplementasikan pada struktur data linear seperti array, linked list, dll	Diimplementasikan pada struktur data yang memiliki traversal dua arah
Approach	Sekuensial	Membagi dua bagian
Ukuran	Lebih prefer untuk data yang kecil	Lebih prefer untuk data yang besar
Efisiensi	Kurang efisien untuk data besar	Lebih efisien untuk data besar
Worst-case Scenario	Elemen berada di akhir	Mengurangi ruang pencarian hingga hanya tersisa satu elemen
Best-case Scenario	Elemen berada di awal	Elemen berada di tengah
Dimensi Array	Array single dan multidimensi	Array multidimensi





Terima Kasih

