



ALGORITMA PEMROGRAMAN

INTERPOLATION SEARCH

TIM PENYUSUN: - DOSEN
- MOCH. DAFFA SHAFWAN CHAIRULLAH

PRESENTED BY: LAB. TEKNIK INFORMATIKA
UNIVERSITAS MUHAMMADIYAH MALANG

ALGORITMA PEMROGRAMAN

CAPAIAN PEMBELAJARAN MATA KULIAH

1. Mahasiswa dapat menjelaskan dan menerapkan konsep Algoritma Pencarian Interpolasi.
2. Mahasiswa mampu menganalisa kompleksitas dari Algoritma Pencarian Interpolasi.

SUB CAPAIAN PEMBELAJARAN MATA KULIAH

1. Mahasiswa mampu membuat program yang menerapkan algoritma Pencarian Interpolasi menggunakan bahasa pemrograman Java.

KEBUTUHAN HARDWARE & SOFTWARE

- Laptop/ PC
- Eclipse/ Netbeans/ dsb.

MATERI PRAKTIKUM

Di modul sebelumnya, kalian telah mempelajari algoritma searching, yaitu linear search dan binary search. Pada modul kali ini kita akan mempelajari dan mempraktekkan mengenai algoritma pencarian yang lain, yaitu interpolation search. Algoritma ini merupakan pengembangan dari algoritma pencarian biner.

Dalam algoritma pencarian biner (binary search), data harus dalam keadaan terurut kemudian dibagi menjadi dua berdasarkan nilai tengahnya. Hal ini berbeda dengan algoritma pencarian interpolasi.

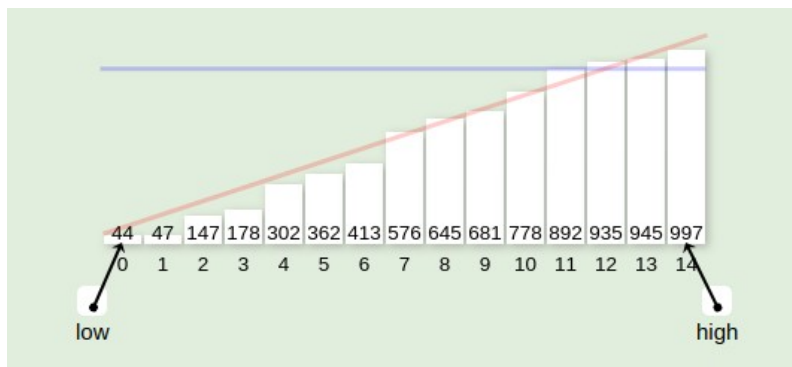
Pada algoritma pencarian interpolasi (interpolation search), data dicari berdasarkan nilai interpolasi / antara dua kutubnya. Dua kutub disini adalah dua titik pada data, yaitu data pada titik/indeks terendah (low) dan titik/indeks tertinggi (high). Sama seperti pencarian biner, data harus diurutkan terlebih dahulu. Pada interpolation search, ada sebuah pointer/penunjuk untuk memprediksi dimana lokasi/posisi indeks data yang kita cari, sebut saja Pos. Kemudian ada variabel untuk mewakili data yang kita cari, sebut saja Key. Berikut adalah rumus untuk mencari nilai dari pos :

$$pos = low + \frac{key - data[low]}{data[high] - data[low]} \times (high - low)$$

Kemudian setelah ditemukan nilai dari pos, maka kita akan membandingkan data yang ada di indeks pos dengan data yang ada di indeks high dan low. Dalam menemukan data yang ingin dicari, ada tiga ketentuan.

1. Jika **data[posisi] == key**, maka **data ditemukan** dan program **berhenti**.
2. Jika **data[posisi] > key**, maka nilai **high = pos - 1**, dan program diulang dari menghitung nilai pos hingga ketentuan 1 tercapai.
3. Jika **data[posisi] < key**, maka nilai **low = pos + 1**, dan program diulang dari menghitung nilai pos hingga ketentuan 1 tercapai.

Berikut adalah visualisasi dari interpolation search,



Misal kita mempunyai array yang sudah terurutkan dengan jumlah data sebanyak 15. Indeks terakhir kita inisialisasi sebagai high dan indeks pertama kita inisialisasi sebagai low. Kita ingin mencari keberadaan elemen dengan nilai 892. berikut adalah langkahnya :

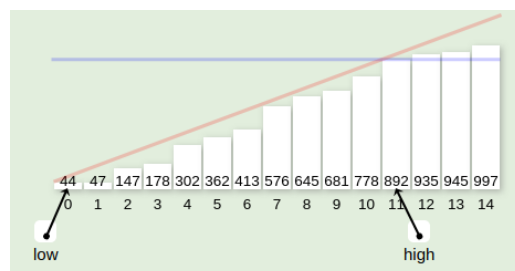
$$pos = 0 + \frac{892 - data[0]}{data[14] - data[0]} \times (14 - 0)$$

$$pos = 0 + \frac{892 - 44}{997 - 44} \times (14)$$

$$pos = 12.45 \text{ (indeks ke 12)}$$

$$pos = 12$$

1. Hitung nilai pos dengan rumus diatas
2. pada data[pos], tidak ditemukan elemen 892.
3. Karena data[pos] > 892, maka nilai high berubah dari 14 menjadi 11 (12 - 1), high = 11.



4. Kemudian menghitung nilai pos yang baru

$$pos = 0 + \frac{892 - data[0]}{data[11] - data[0]} \times (11 - 0)$$

$$pos = 0 + \frac{892 - 44}{892 - 44} \times (11)$$

$$pos = 11$$

5. karena `data[pos] == 892`, maka data ditemukan, program berhenti.

Jadi, bagaimana menurut kalian tentang pencarian interpolasi ? Menarik bukan ? Untuk lebih lengkapnya ada beberapa referensi menarik untuk kalian

<https://iti.cs.vt.edu/AV/Development/interpolationSearchAV.html>
<https://www.geeksforgeeks.org/interpolation-search/>

LEMBAR KERJA

TUGAS PRAKTIKUM

1. Buatlah sebuah program dinamis (berdasarkan input user) untuk memasukkan data angka. Kemudian lakukan sorting dengan menggunakan algoritma sorting yang telah dipelajari. Setelah itu lakukan pencarian data dengan menggunakan algoritma interpolation search.
2. Hitunglah Big-O dari keseluruhan program.
3. Jelaskan kepada asisten mengenai interpolation search.
4. Jelaskan kepada asisten alur penghitungan dari interpolation search berdasarkan nilai yang diinput.

RUBRIK PENILAIAN

Penilaian Kategori 1 (Maksimal)	
-Mengerjakan & menyelesaikan program tanpa eror. -Menjelaskan dengan lancar baik saat Demo dan saat di Tanya	B+ atau A
Penilaian Kategori 2 (Cukup)	
Mengerjakan & menyelesaikan program tetapi tidak sesuai dengan yang di tugaskan. Atau Tidak lancar, bingung saat menjelaskan dan saat di Tanya	C+ sampai B+
Penilaian Kategori 3 (Kurang)	
Program setengah / semua salah. Atau Tidak bisa menjelaskan/ bertele tele	D sampai C
Penilaian Kategori 4 (Sangat Kurang)	
- Ketahuan program sama persis dengan orang lain. -Tidak bisa menjelaskan / bertele tele	E sampai D