



**ANALISIS ALGORITMA *SEQUENTIAL SEARCH* DAN *BINARY SEARCH*
PADA *BIG DATA***

Yoga Religia¹

Program Studi Teknik Informatika, STT Pelita Bangsa
Korespondensi email: yoga.religia@pelitabangsa.ac.id

Abstrak:

Pemakaian data berkembang pesat pada jaman yang dipenuhi dengan berbagai teknologi yang ada. Dimulai dari banyaknya kemudahan yang bisa dinikmati sehari-hari hingga memanfaatkannya untuk berbagai bidang contohnya untuk analisis suatu bisnis. Adanya fasilitas teknologi yang mumpuni memudahkan orang dalam pencarian (*Searching*) informasi dalam berbagai bentuk data yang ada. Akan tetapi pada masalah pencarian data akan menimbulkan masalah dikemudian hari apabila data tersebut dalam jumlah yang besar atau lebih dikenal dengan *Big Data*. *Big Data* tidak hanya mengarah pada jumlah memori yang tersimpan akan tetapi atribut-atribut yang melekat pada suatu data. Perlu penanganan yang lebih dalam mengelola *Big Data* terutama pada saat pencarian suatu kata atau kalimat tertentu sehingga waktu untuk memproses tidak memakan waktu yang terlalu lama. Algoritma *Searching* merupakan sebuah proses untuk memeriksa sekumpulan elemen (daftar elemen) untuk menemukan sebuah elemen tertentu. Dari berbagai macam algoritma *Searching* yang ada, dalam penelitian ini kami bermaksud untuk memberikan gambaran perbandingan antara dua jenis algoritma *Searching* yaitu *Sequential Search* dan *Binary Search* dalam menangani *Big Data*. Penelitian ini akan menunjukkan analisis kinerja dari kedua algoritma tersebut dalam sebuah data yang besar (*Big Data*). Dari hasil analisis dan percobaan dapat disimpulkan bahwa algoritma *Binary Search* memiliki kompleksitas waktu yang lebih cepat dibanding dengan *Sequential Search*.

Informasi Artikel:

Diterima : 22 Juni 2019
Direvisi : 30 Juni 2019
Dipublikasikan : 17 Juli 2019

Kata kunci:

Big Data, Binary Search, Kompleksitas Algoritma Pencarian, Sequential Search

I. Pendahuluan

Algoritma dan struktur data berperan sangat besar dalam pembangunan sebuah perangkat lunak, baik dalam pembuatan desain serta implementasinya. Sebuah algoritma adalah satu set intruksi untuk melakukan tugas atau fungsi tertentu sehingga tercapai tujuan yang diinginkan dengan proses jalan menghabiskan waktu tertentu [1].

Kompleksitas dari sebuah algoritma menggunakan fungsi $g(n)$ yang memberikan batas atas dari jumlah operasi yang berada dalam algoritma tersebut saat sebuah masukan sebanyak n . Kompleksitas sendiri dibagi menjadi dua, yang pertama kompleksitas waktu adalah jumlah waktu yang diperlukan komputer saat menjalankan algoritma tersebut. Yang kedua yaitu kompleksitas ruang, yaitu jumlah ruang memori komputer yang dibutuhkan saat algoritma itu melakukan proses dari awal sampai selesai [6].

Data *Searching* adalah proses yang ada dan dibutuhkan dalam pembangunan sebuah aplikasi, ada banyak algoritma yang bisa diterapkan namun tidak semua algoritma memiliki efisiensi yang baik selama algoritma itu berjalan. Untuk data yang sedikit mungkin perbedaannya belum terlalu besar, namun saat ini perkembangan data juga semakin berkembang yang biasa disebut *Big Data*. *Big Data* adalah sekumpulan data yang sangat besar bisa terdiri dari data *text*, numerik ataupun multimedia yang disimpan dalam sebuah database tertentu [2].

Ada beberapa algoritma yang bisa digunakan untuk melakukan pencarian, salah satunya adalah *Sequential Search*, yaitu algoritma yang dasar dan simpel dari pencarian, dimana menggunakan metode pencarian data dari paling awal

hingga paling akhir dari sebuah list sampai data ditemukan [7]. Selain itu ada algoritma *Binary Search*, yaitu sebuah *list* yang sudah terurut kemudian dibagi menjadi dua bagian [10]. Awalnya adalah membandingkan inputan dengan nilai tengah, selanjutnya dibandingkan ke kanan atau ke kiri sesuai dengan urutan listnya.

Oleh karena itu, dengan masalah adanya data yang besar kami ingin menguji kecepatan waktu proses dan kompleksitas algoritma pencarian dengan menggunakan algoritma *Sequential* dan algoritma *Binary*.

II. Landasan Teori

2.1. *Sequential Search*

Sequential Search adalah cara untuk pencarian data dalam array 1 dimensi. Data yang akan dicari nanti akan ditelusuri dalam semua elemen-elemen array dari awal sampai akhir, dan data yang dicari tersebut tidak perlu diurutkan terlebih dahulu. Terdapat 2 kemungkinan yang akan terjadi dalam waktu pencarian data *Sequential Search*, diantaranya yaitu [7] :

1. *Best Case*

Best case / kemungkinan terbaik akan terjadi apabila data yang dicari terletak pada index array yang paling depan, sehingga waktu yang dibutuhkan untuk mencari data sedikit.

2. *Worse Case*

Worse case / kemungkinan terburuk akan terjadi apabila data yang dicari terletak pada index array yang paling akhir, sehingga waktu yang dibutuhkan untuk mencari data akan sangat lama.

Untuk meningkatkan efisiensi pencarian data pada *Sequential Search* dapat dilakukan dengan cara

menghentikan *looping* dengan menggunakan **BREAK** apabila data yang dicari sudah ketemu [7].

2.2. Binary Search

Binary Search adalah cara untuk pencarian data pada array yang sudah terurut. karena salah satu syarat dalam *Binary Search* adalah data sudah dalam keadaan terurut. dengan kata lain apabila data belum dalam keadaan terurut, pencarian *Binary* tidak dapat dilakukan. *Binary Search* ini dilakukan untuk [10] :

- a. Memperkecil jumlah operasi perbandingan yang harus dilakukan antara data yang dicari dengan data yang ada didalam tabel, khususnya untuk jumlah data yang sangat besar ukurannya.
- b. Beban komputasi lebih kecil karena pencarian dilakukan dari depan, belakang dan tengah.
- c. Prinsip dasarnya adalah melakukan proses pembagian ruang pencarian secara berulang-ulang sampai data ditemukan atau sampai pencarian tidak dapat dibagi lagi (berarti terdapat kemungkinan data tidak ditemukan).

2.3. Big Data

Menurut Dumbill (2012) , *Big Data* adalah data yang melebihi proses kapasitas dari kovensi sistem *database* yang ada [4]. Data terlalu besar dan terlalu cepat atau tidak sesuai dengan struktur arsitektur *database* yang ada. Untuk mendapatkan nilai dari data, maka harus memilih jalan alternatif untuk memprosesnya.

Big Data menurut Richard adalah sekumpulan data (data set) yang besarnya jauh melebihi kemampuan *database software tools* pada umumnya untuk melakukan capture, menyimpan,

mengelola dan menganalisis [3]. Tujuan *Big Data* adalah untuk menciptakan nilai tambah dan memberikan implikasi mengenai bagaimana organisasi harus didesain, diorganisir dan dikelola.

Big Data mengacu pada *dataset* yang ukurannya diluar kemampuan dari *database software tools* untuk meng-capture, menyimpan, me-manage dan menganalisis. Ukuran *Big Data* sekitar beberapa lusin TeraByte sampai ke beberapa PetaByte tergantung jenis Industri. Isi dari *Big Data* adalah Transaksi+interaksi dan observasi atau bisa dibilang segalanya yang berhubungan dengan jaringan internet, jaringan komunikasi, dan jaringan satelit [5].

Big Data dapat di artikan kedalam 9 karakter (IBM) menurut responden sehingga disimpulkan oleh IBM, *Big Data* adalah data yang memiliki *scope* informasi yang sangat besar, model informasi yang real-time, memiliki volume yang besar, dan beraskan *social media data* [8]. Jadi dapat disimpulkan bahwa *Big Data* adalah *dataset* yang memiliki volume besar dan salah satu isinya berdasarkan *social media data*, dan informasi. *Big Data* selalu yang terbaru (*latest data*) sehingga model informasinya real-time, dan *scope* informasinya tidak terfocus pada industri-indrusrtri kecil saja atau industri-indrusrtri besar saja melainkan semuanya baik industry kecil maupun besar [9].

III. Metode Penelitian

Pada penelitian ini akan dilakukan beberapa tahapan yang meliputi latar belakang penelitian, analisis kebutuhan, metode yang digunakan, implementasi, pengujian, analisis pengujian dan hasil. adapun tahapan dari penelitian ini dapat dilihat pada gambar 3.1

Metode yang akan digunakan dalam penelitian ini akan melalui beberapa tahapan yaitu :

1. Mengetahui kompleksitas secara teoritis antara dua algoritma yang akan dilakukan komparasi.
2. Menggunakan data sampel data transaksi perpindahan barang di perusahaan Darmanusa Intermedia sebanyak 150000 *record*.
3. Menghitung lama waktu proses yang berjalan saat proses pencarian dengan menggunakan spesifikasi komputer sebagai berikut:
 - a. *Processor* Intel core i5, v5
 - b. Ram 4 GB
 - c. *Harddisk* 500 GB
 - d. *Operating system* Elementary OS freya
 - e. *Source code* menggunakan bahasa python.



Gambar 3.1 Tahapan Penelitian

IV. Hasil Pengujian

4.1. Kompleksitas *Sequential*

Search Kompleksitas algoritma *Search* adalah tergantung dari jumlah perbandingan yang terjadi dalam perulangan saat melakukan pencarian data dalam sebuah list. Misalkan ada n elemen di dalam list. berikut adalah notasi dalam bentuk larik.

Berikut adalah ekspresi jumlah rata-rata perbandingan yang ada pada algoritma *Sequential Search* adalah sebuah kasus yang berhasil:

- **Kasus terbaik** ialah elemen yang dicari berada diposisi pertama pada list, jadinya hanya melakukan sekali perbandingan dengan notasi $f(n) = O(1)$. Kasus rata rata ialah elemen yang dicari berada di tengah-tengah list, dengan notasi:
 $f(n) = O[(n+1)/2]$
- **Kasus terburuk** ialah elemen yang dicari berada diakhir list, sehingga ia melakukan pencarian atau komparasi data sampai selesai atau sebanyak list itu.
- **Lama Waktu Process *Sequential Search*** : Untuk mengukur lama waktu *Sequential Search* kami melakukan pengujian sebanyak lima kali. Yaitu menguji dalam keadaan worst case, dan best case. Rata - rata lama proses *worst case* ialah 0,014357 ms, sedangkan rata - rata lama proses untuk keadaan *best case* ialah 0,000942 ms.

4.2. Kompleksitas *Binary Search*

Kompleksitas algoritma *Binary Search* ialah tergantung dari banyaknya data, dan data yang dimiliki adalah data yang sudah terurut, pada penelitian ini data terurut secara ascending, untuk kompleksitasnya jika *keyword* yang dimasukkan adalah sebagai mid value dari *list* maka itulah kasus terburuknya.

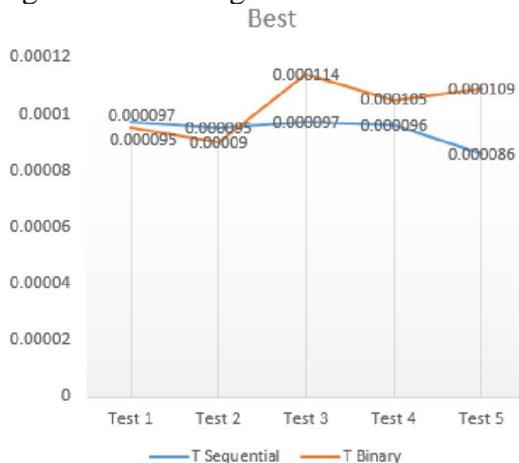
Namun kasus terburuknya ialah *keyword* adalah data terendah atau tertinggi dari data yang dimiliki serta kemungkinan data tidak ditemukan juga memiliki waktu yang lama untuk proses pencarian.

Untuk notasi matematisnya ialah dengan konsep membagi dua *block* data dari n menjadi setengah menjadi $\log_2 n$. Jadi proses pencarian bisa direduksi menjadi setengah dari semua $\log n$.

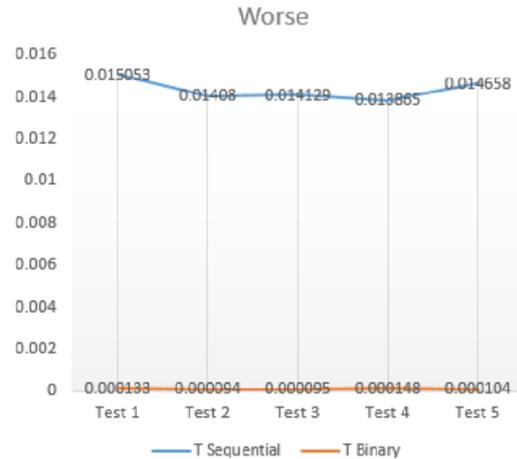
- **Lama Waktu Process Binary Search** : Untuk *Binary Search* kami melakukan pengujian sebanyak lima kali masing masing pada kondisi *worst case* dan *best case*, dengan waktu lama proses *worst case* didapat rata rata 0,000115 ms, dan lama waktu proses dalam keadaan *best case* ialah 0,000103 ms.

4.3. Perbandingan Kedua Algoritma

Dari pengujian yang telah dilakukan dapat disimulasikan dalam sebuah grafik agar dapat dibaca dan dipahami lebih mudah. Adapun grafik yang dimaksud dibagi kedalam 2 jenis, yaitu grafik perbandingan *best case* dan grafik perbandingan *worst case*, yang masing-masing dapat dilihat pada gambar 4.1 dan gambar 4.2.



Gambar 4.1. Best Case



Gambar 4.2. Worse Case

V. Kesimpulan

Algoritma *Searching* merupakan algoritma yang sangat penting dalam pengelolaan sistem manajemen database. Data yang besar (*Big Data*) perlu diolah untuk memudahkan dalam pencarian data. Dari pembahasan diatas, kami membandingkan beberapa algoritma *Searching* seperti *Sequential Search* dan *Binary Search* dengan mengimplementasi-kan dalam berbagai segi antara lain konsep, algoritma, koding dan kompleksitas waktu. Dari percobaan dengan menggunakan bahasa phyton, dapat disimpulkan algoritma *Sequential Search* memiliki kompleksitas waktu lebih besar dibanding dengan *Binary Search*.

VI. Saran

Pada penelitian peneliti ini hanya terbatas membandingkan 2 algoritma *Searching*, diharapkan dari penelitian ini akan semakin banyak muncul penelitian terkait kemangkusan algoritma *Searching* yang lain.

Daftar Pustaka

- [1] Roy Debadrita Roy , Kudu Arnab, “A Comparative Analysis of Three Different Types of *Searching* Algorithms in Data Structure”, Vol.3, issue 5, 2016
- [2] BM. (n.d.). Analytics: The realworld use of *Big Data*. Retrieved from How innovative enterprises extract value from uncertain data: <http://www935.ibm.com/services/us/gbs/thoughtleadership/ibv-big-data-at-work.html>
- [3] Putri Sujana Aprianti, “Memanfaatkan *Big Data* untuk Mendeteksi Emosi”, Vol.2, 2015
- [4] Dumbill, E. *Big Data* Now Current Perspective. O'Reilly Media. 2014.
- [5] Zoumpatianos Kostas, Ideros Stratos, Palpanas Themis. “Indexing for Interactive Exploration of Bid Data Series”. 2013
- [6] Laxmi R., Saranya, S. Pujtha. “Data *Searching* and Implementing Faultolerance using Sector Sphere File System”. ICAESM. 2014
- [7] English Thomas.” No More Lunch: Analysis of *Sequential Search*”. 2008
- [8] Cheng Weiqing, Yang Geng, Zhang Shanshan, Zhang Shaobai. “Stratified
- [9] Multi-Ring Distributed *Search* Model for *Big Data*”. 2016.
- [10] Wan Dingsheng, Xiao Yan, Zhang Pengcheng, Leung Hareton. “Hydrological *Big Data* Prediction Based On Similarity *Search* and Improved BP Neural Network”. 2017.
- [11] Lim Hyesook, Lee Nara. “Survey and Proposal on *Binary Search* Algorithms for Longest Prefix Match”. Vol.14 no.3. 2014