

ANALISIS PERBANDINGAN PENGGUNAAN METODE *BINARY SEARCH* DENGAN *REGULAR SEARCH EXPRESSION*

¹Fenina Adline Twince Tobing✉, ²Rena Nainggolan

¹Program Studi Informatika, Universitas Multimedia Nusantara, Tangerang, Indonesia

²Program Studi Komputerisasi Akuntansi, Universitas Methodist Indonesia, Medan, Indonesia

Email: fenina.tobing@umn.ac.id

ABSTRACT

The search system is a feature that is indispensable for an application or website. By comparing two algorithms that are often used, namely Binary Search and Regular Search Expression (REGEX) algorithms in a simple search system is a problem that will be discussed in this journal. Analysis of the two algorithms is carried out to solve problems in the search system so that the search algorithm can be applied more precisely and effectively. The results prove that the Binary Search has the advantage of searching large amounts of data in an ordered state and has a more effective iteration. While the Regular Expression Search has the advantage of performing searches that are not completely known about the results and keys, besides that this algorithm also allows you to search based on certain patterns in the data.

Keyword: *Searching Binary Search, Regular Search Expression.*

ABSTRAK

Sistem Pencarian menjadi salah satu fitur yang sangat diperlukan pada sebuah aplikasi atau website. Dengan membandingkan dua algoritma yang sering digunakan yaitu *Binary Search* dan algoritma *Regular Search Expression* (REGEX) dalam suatu sistem pencarian sederhana adalah permasalahan yang akan dibahas dalam jurnal ini. Analisis kedua algoritma dilakukan untuk menyelesaikan permasalahan dalam sistem pencarian, sehingga algoritma pencarian dapat diterapkan lebih tepat dan efektif lagi. Hasil penelitian membuktikan bahwa *Binary search* memiliki kelebihan dalam melakukan pencarian pada data berjumlah besar dengan keadaan terurut serta memiliki iterasi yang lebih efektif. Sedangkan *Regular Expression Search* memiliki kelebihan dalam melakukan pencarian yang tidak diketahui secara lengkap mengenai hasil dan kunci, selain itu algoritma ini juga memungkinkan untuk melakukan pencarian berdasarkan pola tertentu pada data.

Kata Kunci: *Pencarian Binary Search, Regular Search Expression.*

PENDAHULUAN

Seiring dengan perkembangan teknologi, pembuatan aplikasi *mobile* dan *website* menjadi hal yang sangat umum saat ini. Hampir semua orang memanfaatkan adanya platform baru yang terus bermunculan. Dengan adanya platform yang dapat digunakan untuk memberikan informasi ini dapat mengubah gaya hidup dan produktivitas manusia dijamin sekarang. Sistem pencarian menjadi salah satu fitur yang sangat diperlukan pada sebuah aplikasi maupun website agar lebih efektif.

Dalam suatu sistem pencarian, terdapat banyak algoritma yang dapat digunakan untuk mencapai tujuan pencarian. Algoritma sendiri merupakan suatu prosedur yang jelas untuk menyelesaikan suatu persoalan dengan menggunakan langkah-langkah tertentu dan terbatas jumlahnya (Suarga, 2012). Algoritma-algoritma tersebut memiliki kelebihan dan kekurangan masing-masing. Sangatlah penting untuk mengetahui kelebihan dan

kekurangan algoritma yang akan digunakan untuk mencapai tujuan yang diinginkan.

Permasalahan yang ditemukan adalah ketidaksesuaian algoritma pencarian yang digunakan terhadap suatu platform. Akibatnya, pengguna memiliki fitur yang terbatas dalam menggunakan platform itu sendiri yang kemudian berakhir dengan ketidakpuasan pengguna terhadap platform yang digunakan.

Penulis mengangkat permasalahan utama ini untuk mencari akar masalah dan solusi yang lebih tepat dengan cara membandingkan 2 algoritma pencarian yang sering digunakan, yaitu algoritma pencarian *Binary Search* dan *Regular Search Expression* (REGEX). Analisa meliputi tentang perbandingan kedua algoritma, kegunaan masing-masing algoritma, dan kesesuaian algoritma terhadap sistem pencarian yang dibutuhkan.

METODE PENELITIAN

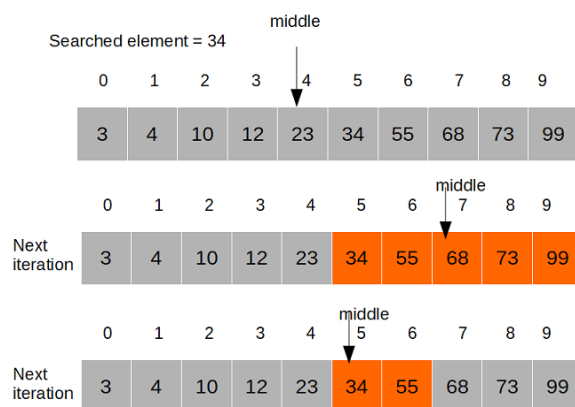
Definisi dan Konsep Algoritma

Regular search expression (REGEX) adalah metode atau algoritma pencarian string yang memiliki konsep dasar pencarian bukan hanya kepada string secara spesifik, namun melakukan pencarian berdasarkan pola dari string tersebut (Nagpure, 2019). Konsep pencarian ini memungkinkan kita untuk melakukan pencarian tanpa kita perlu mengetahui secara lengkap dan tepat data yang akan dicari.

Binary Search merupakan sebuah metode atau algoritma fungsi pencarian yang dirancang untuk berjalan dengan konsep memanfaatkan array dari biner yang sudah tersusun secara urut (Syahputra & Sinurat, 2016). Fungsi ini sederhana, hanya dengan melakukan perbandingan dalam menjalankan fungsinya. Fungsi tersebut membentuk batas-batas untuk dijadikan ruang lingkup pencarian kunci tersebut, dan dilakukan secara berulang hingga mendapatkan kuncinya.

Metode dan Cara Kerja

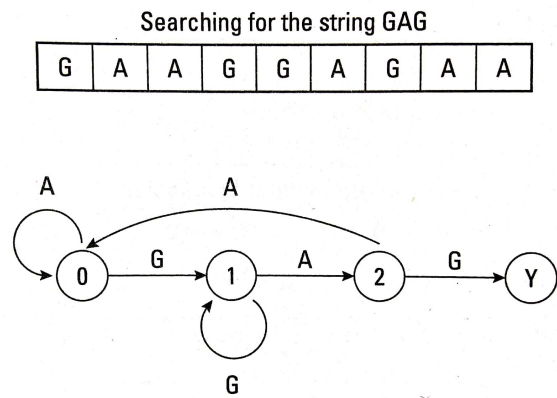
Dalam penerapan algoritma pencarian Binary Search menggunakan metode *Divid and Conquer*. Prinsip dasar dalam menerapkan algoritma pencarian ini adalah melakukan pembagian ruang lingkup data secara berulang hingga mendapatkan kunci atau hingga lingkup data tidak bisa dibagi lagi. Untuk dapat menjalankan proses algoritma pencarian ini, seluruh data diperlukan sudah terurut agar dapat dilakukan pembagian kelompok terhadap ruang lingkup data tersebut (Umar, Riadi, & Muthohirin, 2017).



Gambar 1. Langkah kerja Binary Search

Pada penerapan algoritma Regular Expression Search, berbasis pada pencarian string Finite Automaton (Baeza-Yates & Gonnet, 1996). Algoritma tersebut memungkinkan untuk melakukan pencarian string secara berurut, dengan membentuk beberapa keadaan/state dan grafik berarah, di mana setiap simpul/arah menggambarkan satu karakter

string yang ingin dicari. Algoritma Finite Automaton ini sangat efektif, algoritma ini mengeksekusi setiap karakter tepat sekali dengan waktu konstan tiap karakternya.



Gambar 2. Contoh algoritma Finite Automaton

Contoh algoritma Finite Automaton dalam pencarian string ada pada gambar di atas. String yang akan dicari adalah **GAG** dari keseluruhan string **GAAGGAGAA**. Algoritma akan membentuk keadaan atau state berupa 0, 1, 2 dst, hingga Y sebagai tujuan pencarian. State 0 menunjukkan titik mulai pencarian dan setiap node atau panah menunjukkan string yang dicari.

Karakter pertama yang dicari adalah karakter G, maka ia akan berpindah ke state 1. Kemudian karakter kedua yang dicari adalah karakter A, maka ia akan berpindah ke state 2. Namun, Karakter ketiga yang ia temukan adalah karakter A lagi, maka kembali ke state 0. Karakter keempat adalah mencari karakter G, maka kembali bergerak ke state 1. Karakter ke 5 yang dicari adalah karakter G, maka tetap berada di state 1. Karakter ke 6 yang dicari adalah karakter A, maka akan bergerak ke state 2. Hingga pada akhirnya pada karakter 7 ditemukan karakter G, sehingga bergerak ke state Y yang menunjukkan kalau string sudah ditemukan.

Selain menggunakan algoritma pencarian Finite Automaton, algoritma Regular Search Expression menggabungkan pencarian dengan melihat sebuah pola atau *pattern* yang digunakan dalam proses pencarian. Algoritma Regular Search Expression menggabungkan kedua cara tersebut untuk melakukan pencarian lebih efektif lagi dengan memberikan beberapa pattern yang memudahkan pengguna untuk melakukan pencarian. Regular expression tersebut sudah tersedia di dalam beberapa bahasa pemrograman seperti Perl, PHP, Java, C#, dan Visual Basic (Hock-Chuan, 2018).

Dengan adanya pencarian berdasarkan pattern tersebut, pengguna memiliki keleluasaan

dalam melakukan pencarian. Pengguna dapat melakukan pencarian dengan lebih spesifik mengenai hasil yang ingin ditampilkan, hingga hasil yang tidak ingin ditampilkan. Sebagai contoh, apabila kita melakukan pencarian dengan algoritma pencarian biasa, kita tidak akan menemukan sebuah string apabila tidak mengetahui keseluruhan string tersebut. Misalnya, pengguna ingin melakukan pencarian string **Dimas**, tetapi hanya mengingat 3 huruf awal yaitu **Dim**, maka pencarian tidak akan menemukan hasil tersebut. Sebaliknya, dengan menggunakan algoritma Regular Expression, pengguna tetap dapat menemukan string secara keseluruhan walaupun hanya mengingat beberapa string.

HASIL DAN PEMBAHASAN

Implementasi

Binary search merupakan salah satu algoritma yang relatif efisien, sehingga membuat algoritma tersebut menjadi algoritma pencarian yang populer dan sering digunakan. Andaikan ingin mencari posisi sebuah nama dari seluruh nama yang ada di dunia yang telah dicatat dan diurutkan, Binary Search dapat menyelesaikannya dengan maksimal sebanyak 35 iterasi. Binary Search memiliki waktu searching yang lebih cepat untuk data yang besar, terbukti dengan metode dan jumlah iterasi yang dilakukan oleh algoritma Binary Search tersebut. Hal ini membuat algoritma Binary Search memiliki kelebihan seperti kecilnya beban komputasi yang digunakan saat menjalankan algoritma pencarian (Alexander, 2011). Namun, algoritma Binary Search juga memiliki kekurangan, yaitu data harus diurutkan atau disorting terlebih dahulu. Hal ini menjadikan Algoritma Binary Search tidak baik untuk melakukan pencarian terhadap data berantai.

Pseudocode metode Binary Search adalah sebagai berikut (Umar et al., 2017):

```
Procedure pencarian biner
  X ← sorted array
  Y ← size of array
  Z ← value to be searched
  Set data_low = 1
  Set data_up = Y
  while Z not found
    if data_up < low
      EXIT: Z does not exist.
    Set mid_point = low + (data_up - data_low) / 2
    if X[mid_point] < Z
      set data_low = mid_point + 1
    if X[mid_point] > Z
      set data_up = mid_point - 1
    if X[mid_point] = Z
      EXIT: Z found at location mid_point
  end while
end procedure
```

Gambar 3. Pseudocode Binary Search

Dasar dari algoritma Binary Search seperti yang sudah dijelaskan pada dasar teori adalah membandingkan nilai dari kunci dengan nilai dari data yang dicari. Untuk mencari data berupa string atau text, bahasa pemrograman perlu mengubah data string menjadi pecahan angka, atau lebih mudah dipahami dengan ASCII. Sehingga pada pencarian string menggunakan algoritma Binary Search, nilai dari data yang dibandingkan adalah nilai ASCII dari string tersebut.

Algoritma Regular Expression Search memiliki fungsionalitas yang jauh lebih fleksibel dibandingkan algoritma Binary Search. Seperti teori yang sudah disampaikan, algoritma Regular Expression Search mampu mencari sebuah data tanpa mengetikkan data tersebut secara lengkap. Hal ini yang menyebabkan algoritma Regular Expression Search banyak digunakan dalam search engine favorit yang sering kita gunakan seperti Google, Yahoo, dsb.

Penggunaan algoritma ini juga sangat luas dan tidak hanya terletak pada search engine yang rumit. Algoritma ini bisa ditemukan hampir dalam setiap fitur atau aplikasi yang memiliki fitur "Find" atau "Replace", biasanya diakses dengan shortcut "Ctrl + F" pada keyboard. Saat ini banyak aplikasi yang sudah mendukung regular expression di dalamnya, mengingat fungsionalitas dari algoritma ini namun dengan beberapa cara yang berbeda. Banyak aplikasi yang mengembangkan regular expression sesuai dengan kebutuhannya dan memilih menggunakan regex dibanding fitur search biasa. Dengan adanya regular expression dapat menghemat waktu dalam pengerjaan coding.

Apabila pada Binary Search kunci yang dicari merupakan sebuah string dari sebuah array, lain halnya dengan Regular Expression. Kunci pada regular expression berupa character yang dicari dari sebuah string. Berikut merupakan bagaimana Regular Expression dijalankan.

```
<SCRIPT LANGUAGE="JavaScript"><!--  
function demoMatchClick() {  
  var re = new RegExp(document.demoMatch.regex.value);  
  if (document.demoMatch.subject.value.match(re)) {  
    alert("Successful match");  
  } else {  
    alert("No match");  
  }  
}  
  
function demoShowMatchClick() {  
  var re = new RegExp(document.demoMatch.regex.value);  
  var m = re.exec(document.demoMatch.subject.value);  
  if (m == null) {  
    alert("No match");  
  } else {  
    var s = "Match at position " + m.index + ":\n";  
    for (i = 0; i < m.length; i++) {  
      s = s + m[i] + "\n";  
    }  
    alert(s);  
  }  
}  
</SCRIPT>
```

Gambar 4. Source Code Regular Expression

Dalam JavaScript sudah terdapat built-in function methods untuk penggunaan regex, seperti `exec()` pada source code diatas yang berfungsi untuk menguji apakah terdapat kecocokan pada string. Dalam penggunaan Regular Expression juga terdapat beberapa kode yang dibuat agar dapat digunakan user untuk memudahkan dalam pencarian, yaitu *brackets*, *metacharacters*, dan *quantifiers*.

Pengujian Algoritma

Pengujian bertujuan untuk mencari perbedaan dari kedua algoritma pencarian. Perbedaan inilah yang akan menjadi dasar dari kesimpulan jurnal ini, yaitu agar kita dapat mengetahui hasil dari algoritma pencarian yang dijalankan pada Binary Search maupun Regular Expression Search serta dapat mengetahui dengan tepat kegunaan dari setiap algoritma pencarian. Pengujian algoritma akan dilakukan dengan melakukan pencarian pada kedua algoritma. Pencarian dilakukan dengan format yang sama agar perbedaan dari kedua algoritma tersebut dapat ditemukan dan dianalisa kembali.

Pengujian algoritma akan menggunakan algoritma yang sudah dibuat oleh penulis dan sudah dibahas pada bagian sebelumnya. Contoh yang akan digunakan adalah mencari "adimas" dari keseluruhan string "adimas putra". Menurut teori dan pembahasan sebelumnya, penggunaan algoritma Binary Search tidak akan berhasil menemukan hasil pencarian, sedangkan algoritma Regular Expression Search dapat menemukan hasil pencarian. Untuk membuktikan, kita akan langsung mencoba memasukkan input string ke dalam algoritma masing-masing melalui bahasa pemrograman Javascript.

Pengujian pada algoritma Binary Search:

```
let arr = ["adimas putra"];  
let x = "adimas";
```

Gambar 5. Variabel pengujian Binary Search

Pencarian string 'adimas' terhadap string 'adimas putra'
Element not found!

Gambar 6. Hasil pengujian Binary Search

This page says
Successful match
OK

Gambar 8. Hasil pengujian Regular Expression 1

This page says
Match at position 0:
adimas
OK

Gambar 9. Hasil pengujian Regular Expression 2

Pengujian pada kedua algoritma membuktikan bahwa sesuai teori dan pembahasan yang sudah dipaparkan sebelumnya adalah benar. Hasil dari pengujian tersebut membuktikan bahwa algoritma Binary Search tidak dapat menemukan string apabila kunci tidak tepat sesuai dengan hasil yang diharapkan. Sebaliknya, pada algoritma Regular Expression Search memungkinkan kita untuk tetap menemukan string walaupun kunci yang dimasukkan tidak tepat sesuai dengan hasil yang diharapkan

KESIMPULAN

Berdasarkan dari pembahasan dapat ditarik kesimpulan bahwa kedua algoritma pencarian tersebut merupakan algoritma yang populer dengan kelebihan masing-masing. Algoritma Binary Search yang unggul dengan kecepatan dan kesederhanaan menjadikannya digemari oleh banyak orang, dan juga Regular Expression dengan fleksibilitas dan fungsionalitas yang sangat baik. Meskipun begitu keduanya juga memiliki beberapa batasan yang menjadikan tidak dapat mengandalkan sepenuhnya hanya pada salah satu algoritma, namun juga bergantung pada fungsinya masing-masing.

Pada algoritma Binary Search, membutuhkan data yang sudah diurutkan/sorted terlebih dahulu, sehingga cocok digunakan pada platform dengan data yang sudah berurut dan tidak merupakan data berantai. Algoritma Binary Search juga sangat efektif untuk

memukan hasil pencarian dengan sangat cepat serta menggunakan beban komputasi yang kecil. Selain itu, algoritma Binary Search juga sangat cocok untuk digunakan pada pencarian data yang banyak, karena algoritma ini menggunakan metode Divide and Conquer, dimana algoritma mengabaikan komputasi pada array yang tidak perlu.

Sedangkan pada algoritma Regular Expression Search, tidak dibutuhkan data yang sudah diurutkan/sorted. Algoritma Regular Expression Search memungkinkan kita untuk mencari data secara acak tanpa perlu menuliskan kata kunci pencarian secara lengkap. Algoritma Regular Expression Search juga memiliki fleksibilitas tinggi dengan memberikan fitur-fitur atau syntax yang sudah tersedia didalam beberapa bahasa pemrograman seperti C, C++, Javascript, dsb. Fitur ini memungkinkan kita untuk mencari sebuah pattern dari kumpulan data, hingga memberikan batasan mengenai apa yang ingin kita cari maupun tidak ingin kita cari. Algoritma Regular Expression Search sangat cocok digunakan dalam sistem pencarian pada search engine, web services, dan platform sejenis.

DAFTAR PUSTAKA

- Alexander, R. (2011). Penerapan Teknik Binary Search Tree Sebagai Alternatif Penyimpanan Data. *Makalah IF2091 Struktur Diskrit*, 1–5.
- Baeza-Yates, R. A., & Gonnet, G. H. (1996). Fast Text Searching for Regular Expressions or Automaton Searching on Tries. *Journal of the ACM*, 43(6), 809–810.
- Hock-Chuan, C. (2018). Regular Expressions (Regex). Retrieved September 7, 2020, from <https://www3.ntu.edu.sg/>
- Nagpure, A. (2019). Regular Expressions | A Complete Beginners Tutorial. Retrieved September 7, 2020, from <https://blog.usejournal.com/>
- Suarga. (2012). *Algoritma dan Pemrograman*. Yogyakarta: Andi Offset.
- Syahputra, G., & Sinurat, B. (2016). Implementasi Teknik Binary Search Pada Kamus Indonesia - Batak Toba. *Journal Of Informatics Pelita Nusantara (JIPN)*, 1(1), 28–37.
- Umar, R., Riadi, I., & Muthohirin, B. F. (2017). Implementasi Algoritma Binary Search Pada Aplikasi Konkordansi Al-Qur'an Berbasis Android. *Seminar Nasional Teknologi Informasi Dan Komunikasi - SEMANTIKOM*, 49–54.