

PENERAPAN ALGORITMA BRUTE FORCE UNTUK PENCARIAN KAMUS ISTILAH SISTEM MANAJEMEN BASIS DATA.pdf *by*

Submission date: 17-Apr-2023 07:30AM (UTC+0700)

Submission ID: 2066396778

File name: PENERAPAN ALGORITMA BRUTE FORCE UNTUK PENCARIAN KAMUS ISTILAH SISTEM MANAJEMEN BASIS DATA.pdf (440.06K)

Word count: 4005

Character count: 22621

PENERAPAN ALGORITMA BRUTE FORCE UNTUK PENCARIAN KAMUS ISTILAH SISTEM MANAJEMEN BASIS DATA

Heri Purwanto¹⁾, Muhamad Fazri Annafi²⁾, Rikky Wisnu Nugraha³⁾, Rudy Sofian⁴⁾, dan Fahmi Reza Ferdiansyah⁵⁾

¹⁾Program Studi Sistem Informasi, Fakultas Teknik, Universitas Sangga Buana YPKP
Jl. Phh. Mustofa No.68, Cikutra, Kec. Cibeunying Kidul, Kota Bandung, Jawa Barat 40124

²⁾Program Studi Teknik Informatika, Fakultas Teknologi Informasi dan Digital, Institut Digital Ekonomi LPKIA

³⁾Program Studi Sistem Informasi, Fakultas Teknologi Informasi dan Digital, Institut Digital Ekonomi LPKIA

⁴⁾Program Studi Manajemen Informatika, Fakultas Teknologi Informasi dan Digital, Institut Digital Ekonomi LPKIA

Jl. Soekarno Hatta No.456, Batununggal, Kec. Bandung Kidul, Kota Bandung, Jawa Barat 40266

e-mail: heri.purwanto@usbykp.ac.id¹⁾, 170914042@fellow.lpkia.ac.id²⁾, r.wisnunugraha@lpkia.ac.id³⁾, rudysofian@lpkia.ac.id⁴⁾, fahmirezaf@lpkia.ac.id⁵⁾

ABSTRAK

Tujuan penelitian ini untuk menerapkan sebuah algoritma pada sebuah aplikasi kamus pencarian informasi mengenai sistem manajemen basis data bagi para programmer yang akan mempelajari mengenai konsep dari Data Base Management System (DBMS). Aplikasi tersebut menggunakan sebuah algoritma brute force, dimana algoritma tersebut digunakan untuk melakukan pencarian antara pola dan teks pada sebuah aplikasi pencarian berbasis web. Metode penelitian yang digunakan yaitu dengan mengkombinasikan antara metode penelitian kuantitatif dan juga kualitatif, sedangkan pengembangan perangkat lunak menggunakan pendekatan prototype. Pengujian dari algoritma ini menggunakan metode String Matching untuk mencocokkan pola dengan teks pencarian, selain itu juga dilakukan pengamatan terhadap waktu pencarian. Setelah dilakukan pengujian didapatkan waktu rata-rata sebesar 0.01675 detik dengan demikian penerapan algoritma brute force dapat digunakan untuk menyelesaikan permasalahan dalam melakukan pencarian kamus istilah DBMS yang sederhana, mudah dipahami dalam penerapannya, lebih efektif untuk data yang terbatas, source code yang dihasilkan sedikit dan menghasilkan algoritma yang dapat dieksekusi untuk beberapa masalah seperti pencocokan dan pencarian.

Kata Kunci: Algoritma Brute Force DBMS, metode gabungan, penerapan algoritma, String Matching

ABSTRACT

The purpose of this research is to apply an algorithm to an application dictionary of information about database management systems for programmers who will learn about the concepts of the DBMS. The application uses a brute force algorithm, which is used to perform searches between patterns and text in a web-based search application. The research method used by the authors is a combined research method. Testing of this algorithm uses the String Matching method to match patterns with the search text, besides that, observations are also made on the search time. After testing, an average time of 0.01675 seconds is obtained, thus the application of a brute force algorithm can be used to solve the problem of searching a DBMS term dictionary simply, easy to understand in its application, more effective for limited data, the resulting source code is small and generate executable algorithms for several problems such as matching and searching.

Keywords: Brute Force algorithm, DBMS, combined methods, application of algorithms, String Matching

I. PENDAHULUAN

Menurut Febbri dan Schwab *database* (basis data) adalah sistem berkas yang disatukan terutama untuk meminimalkan redundansi data. Kemudian istilah database menurut Chou merupakan seperangkat informasi berguna yang diatur dengan cara tertentu [1]. Penyusunan *database* (basis data) melibatkan proses memasukkan data ke dalam

media penyimpanan data yang dikelola oleh *Database Management System* (DBMS) [2]. Penggunaan DBMS saat ini sangat penting dalam skala besar atau dalam skala kecil [3].

Menurut KBBI kamus istilah merupakan kamus yang berisikan tentang istilah, pengertian, makna atau konsep dari ilmu bidang tertentu seperti kamus [4]. Kamus istilah biasanya berbentuk buku dalam menemukan istilah yang diinginkan, pencarian masih dilakukan

dengan membaca buku setiap lembar dan pencarian dilakukan berdasarkan alfabet [5], ataupun pada indeks buku dan halaman daftar isi.

Kemajuan teknologi saat ini mengalami banyak perubahan yang sangat drastis, seperti halnya dibidang komunikasi dan informasi. Manusia mulai memanfaatkan penggunaan teknologi untuk memudahkan setiap pekerjaan [6]. Teknologi yang digunakan saat ini adalah mesin pencarian. Teknologi ini memungkinkan pencari informasi untuk memasukkan kata kunci ke dalam mesin pencari [7]. Dengan teknologi ini, kita dapat menggunakan teknologi mesin pencarian untuk menemukan apa yang kita butuhkan. Kita bisa mencari apapun dengan memasukkan kata kunci [8].

Berdasarkan penelitian terdahulu tentang pencarian kata pada kamus istilah komputer berbasis android dalam penelitiannya yaitu penggunaan salah satu metode pencarian menggunakan pengetahuan algoritma untuk melakukan sebuah pencarian menggunakan kata istilah yang ada pada komputer menggunakan android [5]. Sedangkan penelitian lain, melakukan sebuah analisa pencarian istilah Statistika dengan penerapan algoritma *Brute Force*, penerapan dari analisis algoritma pencarian ini, dengan membangun sebuah perangkat lunak istilah statistika menggunakan sebuah bahasa pemrograman Visual Basic [9].

Algoritma *brute force* merupakan pendekatan dalam memecahkan sebuah permasalahan, umumnya

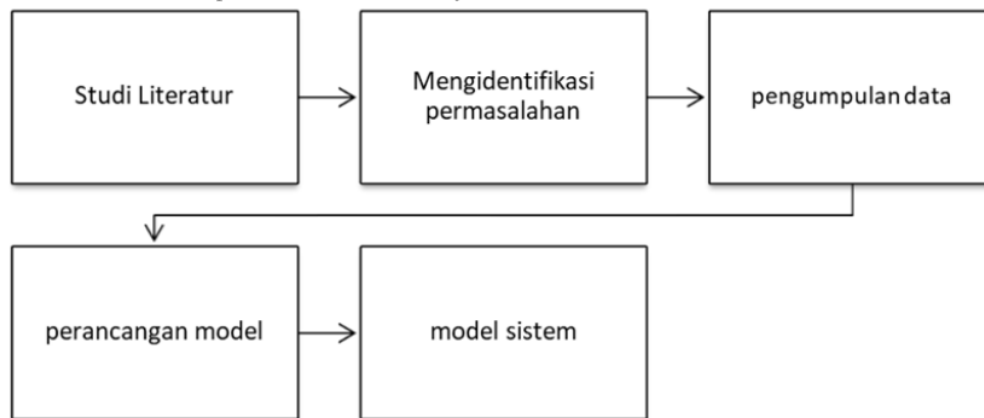
berdasarkan deklarasi dan definisi konsep yang terlibat. Dalam menyelesaikan permasalahan dari Algoritma *brute force* dilakukan dengan menggunakan cara yang sangat sederhana, langsung dan cara yang jelas [10]. Penggunaan teknik yang pada umumnya hanya dapat digunakan jika penyusun algoritma lebih mendahulukan dalam memperoleh solusi dari problem secara langsung apa adanya [11].

Tujuan penelitian ini untuk menerapkan algoritma pada sebuah aplikasi pencarian informasi mengenai database management system bagi para programmer yang akan mempelajari mengenai konsep dari DBMS. Aplikasi tersebut menggunakan sebuah algoritma *brute force* dalam penerapan yang langsung dan sederhana, penggunaan sebuah algoritma tersebut dilakukan dengan cara mencari antara pola dan teks yang sama dan pada diaplikasikan pada perangkat lunak pencarian berbasis web.

II. METODE PENELITIAN

A. Analisis permasalahan

Metode penelitian gabungan adalah salah satu penelitian yang menggabungkan dua atau lebih metode penelitian, yaitu penelitian kuantitatif dan kualitatif. Penelitian ini juga dapat dikatakan analisis data yang diintegrasikan ke dalam hasil dan akhirnya disimpulkan [12]. Dapat dilihat pada Gambar 1 merupakan sebuah tahapan analisis yang digunakan pada penelitian ini.



Gambar 1 Tahapan Analisis

Proses pada Gambar 1 dijabarkan seperti dibawah ini:

1. penelitian ini dimulai dengan studi literatur tentang penerapan algoritma *brute force* pada kamus istilah Sistem Manajemen Basis Data berbasis web.
2. Langkah selanjutnya ialah mengidentifikasi permasalahan yang muncul dalam sebuah pencarian berbasis buku atau kamus cetak.
3. Pengumpulan data dengan mencari bentuk dari algoritma *brute force* yang tepat saat digunakan

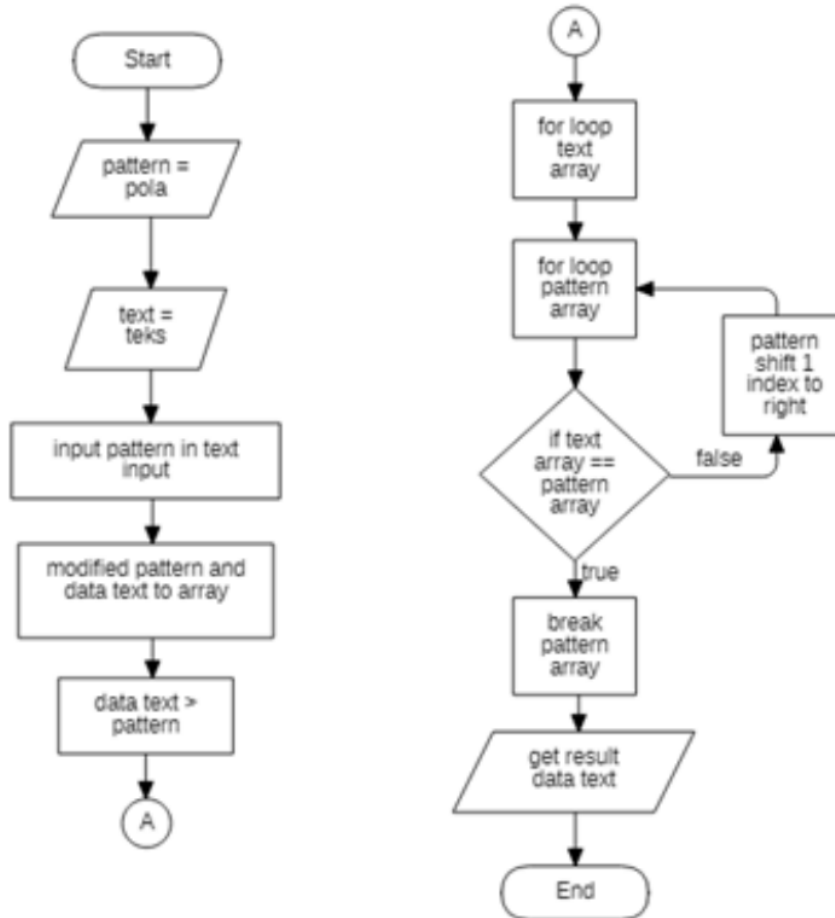
serta teori dari database management system.

4. Pada tahap perancangan model ini, pencarian informasi mengenai DBMS akan dikaitkan dengan algoritma *brute force*.
5. Langkah terakhir adalah membuat bentuk pemodelan pencarian informasi kamus istilah Sistem Manajemen Basis Data dengan berbasis web.

B. Rancangan metode permasalahan

Flowchart Diagram atau diagram alir merupakan diagram yang dapat menunjukkan alur dari sebuah pemecahan masalah yang jelas, yakni dengan menerapkan tahapan kegiatan sistematis dan juga logis. Diagram ini menggambarkan tahapan dua dimensi

berupa simbol-simbol grafis. Setiap simbol menunjukkan setiap jalur pengendalian dan kegiatan operasi. *Flowchart* juga dapat dikatakan sebagai simbol yang menggambarkan proses yang mengalir dalam suatu sistem [13].



Gambar 2 *Flowchart* Diagram

Pada Gambar 2 menjelaskan mengenai *flowchart* dalam menentukan algoritma *Brute Force*. Berikut penjelasan Gambar 2:

1. Memberikan inisiasi untuk pola dan teks.
2. Pola adalah data yang akan diinputkan ke dalam text input. Teks adalah data yang ada pada aplikasi.
3. Input pencarian pada text input.
4. Program akan mengubah pattern atau pola ke dalam data index array. Ubah data text ke dalam index array juga.
5. Perlu diperhatikan bahwa text harus lebih banyak index array daripada pattern index array.
6. Kemudian gunakan perulangan pada text untuk pergerakan dari text yang dimulai dari kiri ke

kanan. Di dalam text array masukan kembali perulangan untuk pattern

7. Bandingkan index array tersebut dengan menggunakan pengkondisian jika *text array* sama dengan *pattern array*. maka hentikan perulangan, dan mulai lagi pattern dengan index berikutnya, jika tidak maka lanjutkan ke perulangan teks.
8. Munculkan hasil dari data text yang sama dengan pattern.

C. *Brute Force* Algorithm

Algoritma *Brute Force* merupakan pendekatan yang dilakukan secara langsung dalam memecahkan sebuah

masalah, berdasarkan pola pencarian dan pengertian yang melibatkan [14]. Penggunaan algoritma tersebut dilakukan untuk mencocokkan pattern dengan data text antara indeks 0 dan panjang dari teks(n)-panjang dari pattern(m) untuk mendapatkan text yang dicari [15]. Menurut penelitian [16] langkah-langkah brute force adalah sebagai berikut:

1. Pencocokan *pattern* dari teks awal.
2. Mencocokkan pola per karakter terhadap teks yang dimulai dari kiri ke arah kanan, sampai salah satu dari karakter pada pola dan text jika perbandingan tidak sama akan terus berulang hingga semua karakter pattern sampai cocok dengan text.
3. Algoritma tersebut akan terus menggeser index pada pola sebesar satu index ke sebelah kanan dan akan kembali lagi pada langkah kedua sampai pola tersebut ada diteks terakhir.

Algoritma ini mempunyai beberapa kelebihan dan kelemahan. Kelebihan dari Algoritma ini yakni [17]:

- a. Cocok digunakan untuk menyelesaikan hampir sebagian besar masalah.
- b. Sederhana dan mudah dimengerti
- c. Menghasilkan algoritma yang layak untuk beberapa permasalahan yang dianggap penting seperti pencarian, pencocokkan, pengurutan string atau perkaian matriks.
- d. Menghasilkan algoritma standar untuk tugas-tugas komputasi, menentukan elemen minimum atau maksimum di dalam sebuah urutan

Sedangkan kelemahan dari algoritma ini yaitu sebagai berikut :

- a. Jarang menghasilkan algoritma yang efektif
- b. Beberapa permasalahan dengan algoritma ini lambat.
- c. Kurang unggul daripada algoritma pencarian lainnya

Algoritma ini juga sangat cocok untuk melakukan pengecekan pada setiap posisi *string* dalam teks dimulai dari karakter pertama hingga terakhir. Setelah melakukan pengecekan pada karakter yang pertama, dilanjutkan dilakukan proses pemindahan setiap *string* dari satu posisi kearah sebelumnya atau karakter akan berganti posisi menuju karakter kedua dan begitu seterusnya. Proses membandingkan karakter pada setiap teks dapat dikatakan selesai pada tahap manapun selama proses pencarian berlangsung [18]. Seperti dicontohkan pada ka¹ ini, terdapat beberapa karakter yang akan dicocokkan sebagai text yaitu “backbone” dan yang akan dicocokkan adalah pola = “bone”. Pada tahap awal karakter pertama b akan mendapatkan hasil terjadi kesamaan (*match*), hal itu disebabkan karena karakter yang dibandingkan pertama kali huruf “b”, sementara itu pada karakter dua “o” terjadi ketidakcocokan

(*mismatch*) pada teks karena karakter kedua pada pola “o” tidak sama dengan karakter huruf pada teks, maka proses pergeseran dilakukan yaitu dengan berpindah dari satu posisi ke arah kanan. Pencocokan karakter yang dikatakan tidak cocok *mismatch* akan terjadi proses pergeseran tersebut sampai karakter padapattern keseluruhan match dengan karakter pada teks.

D. String Matching

Metode pengujian pencocokan *text* pada penelitian ini menggunakan metode *string matching*. *String Matching* merupakan sebuah metode untuk mencocokkan *text* berdasarkan sebuah *pattern* [8], [14]. Teks adalah sebuah *string* yang akan dicari sedangkan n merupakan panjang dari sebuah teks tersebut. Pola merupakan *string* yang mempunyai panjang sebanyak m karakter yang merupakan pencarian sebuah teks [14].

Metode ini juga mirip dengan metode pencarian pada sebuah perambanan yang memiliki mesin pencarian berdasarkan kata kunci tertentu yang dikerjakan oleh mesin seperti *Googel*, *Bing*, *Yahoo* dan lain sebagainya [14]. *String Matching* juga bisa dikatakan sebagai sebuah alur pencarian dimana setiap kueri akan dihadapkan kepada sebuah pola yang digunakan untuk *string* yang lebih panjang [19].

Perumusan *String matching* atau pencocokan teks dilakukan dengan cara [8]:

$$x = x[0 \dots m - 1] \quad (1)$$

$$y = y[0 \dots n - 1] \quad (2)$$

Pengertian:

x = Pola atau pattern
 m = panjang pola
 y = text atau teks
 n = panjang text

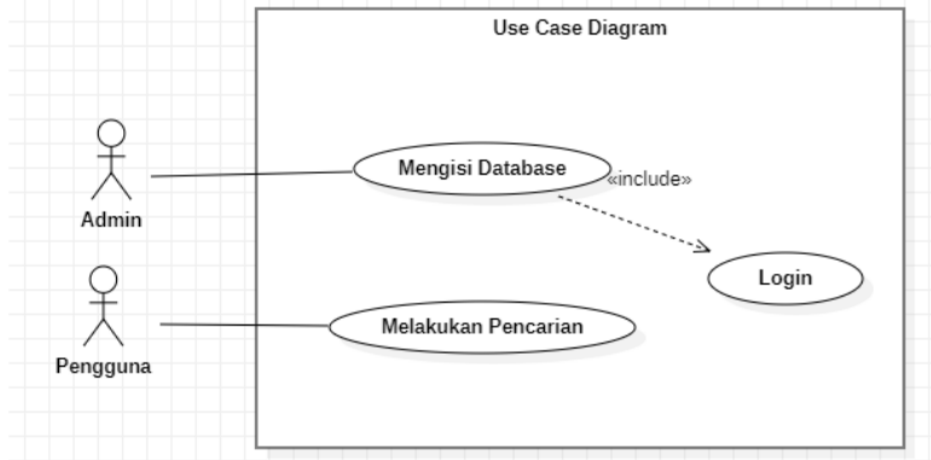
¹ Terdapat 2 (dua) teknik utama yang dapat digunakan dalam algoritma *string matching* yaitu *exact matching*, dimana hasil pencocokannya berisi *string* yang tidak memiliki perbedaan dengan *string* yang diinput atau sama persis, seperti yang terdapat pada algoritma *Smith Waterman*, *Needleman Wunsch*, *algoritma Boyer-Moore-Horspool*, *algoritma Knuth-Morris-Pratt*. Kedua adalah *approximate matching* berbeda dengan *exact matching*, teknik ini menghasilkan pencocokan mengandung *string* yang memiliki sedikit perbedaan atau bisa dikatakan tidak harus persis dengan *string* yang diinput, seperti yang terjadi pada pada algoritma *algoritma Brute Force*, *Fuzzy String Searching*, *algoritma Rabin Karp* [20].

III. HASIL DAN PEMBAHASAN

A. Rancangan Sistem

Rancangan sistem yang digunakan pada penelitian ini menggunakan pemodelan *Unified Modeling Language*

(UML) yang digambarkan pada diagram Use Case seperti pada gambar 3.



Gambar 3 Use Case Diagram

Gambar 3 merupakan gambaran sistem mengenai aplikasi yang akan dibangun. Terdapat 2 (dua) aktor utama yaitu admin yang berfungsi untuk menginputkan data-data dari pencarian dan juga pengguna yang berfungsi untuk melakukan pencarian.

Dalam penelitian ini, penulis membangun sebuah aplikasi pencarian kamus istilah database (DBMS)

dibangun dengan menggunakan bahasa pemrograman PHP dan algoritma yang digunakan ialah brute force, mendapatkan hasil pencarian efisien sesuai dengan kata kunci yang dicari oleh pengguna dikamus aplikasi. Beberapa data istilah dan artinya [1] yang diambil untuk data sampel penelitian dijabarkan pada tabel 1.

TABEL 1
DATA SAMPEL ISTILAH DBMS

No	Istilah	Penjelasan
1	DBMS	Database Management System.
2	Database	Sekumpulan data yang disimpan dalam perangkat komputer
3	Database Management System	sebuah perangkat lunak yang dirancang khusus dalam menghubungkan setiap basis data
4	Data Definition Language	Sebuah perintah yang digunakan untuk mengelola kerangka dari sebuah database maupun tabel
5	Data Manipulation Language	Sebuah perintah yang digunakan untuk memanipulasi data
6	Data	Data adalah sekumpulan fakta
7	Informasi	Sekumpulan dari data yang telah diolah
8	Komponen Utama DBMS	Perangkat Keras, Perangkat Lunak dan Pengguna
9	Model Relasional	Sebuah model yang menggambarkan relasi antar tabel

B. Pengkodean Sistem

Gambar 4 merupakan *source code* dalam penerapan algoritma *brute force* pada aplikasi kamus pencarian DBMS. Data sampel ditampilkan semua, kemudian data sampel diubah menjadi array. Pattern atau pola yang akan dicari diubah menjadi array. Proses dilakukan menggunakan perulangan dari teks kemudian

didalamnya dimasukan dengan perulangan dari pola atau pattern, lalu ditambahkan pengkondisian jika teks index 0 sama dengan pola index 0 maka berhenti dan dilanjut dengan teks index 0 sama dengan pola index 1, apabila ada yang tidak sama maka perulangan pada teks akan dilanjutkan ke index berikutnya hingga pola dan teks sama, maka proses algoritma akan berhenti.

```

var datas = "";
var textspl = "";
var temp = [];

for (let a = 0; a < data.length; a++) {
  datas = Object.values(data[a].title)
  textspl = datas;
  for (let i = 0; i < textspl.length; i++) {
    for (let j = 0; j < patternspl.length; j++) {
      if(textspl[i + j] != patternspl[j]) {
        console.log(textspl[i + j]);
        break;
      } else {
        temp.push(textspl[i + j])
        console.log(temp)
      }
    }
  }
}

```

Gambar 4 Source Code Algoritma Brute Force

C. Gambaran Sistem

Gambar 5 merupakan *screenshot* dari sampel data istilah dan pengertian dari materi DBMS terdapat pada

[1]. Sampel data yang dicari yaitu dengan kata kunci DBMS sehingga dapat menghasilkan *output* seperti pada Gambar 5.



Gambar 5 Tampilan Pencarian Istilah DBMS

Gambar 6 merupakan *screenshot* dari hasil pencocokan pola dari teks basis data dan hasil yang dimunculkan adalah teks istilah dan pengertian mengenai model basis data. Pada pencarian dimulai dari pola pada index ke 0 dan teks index 0 kemudian

ditemukan sebuah kecocokan huruf B pada pola diindex 6 dan huruf B pada teks diindex ke 6 diteruskan dengan huruf A pada index ke 7 dan seterusnya hingga huruf dari pola tersebut berakhir.



Gambar 6 Hasil Pencocokan Pola dengan Teks

D. Pengujian *String Matching*

Algoritma *brute force* untuk mencocokkan pola dengan teks yang diproses mulai mencocokkan setiap karakter pola dengan teks dari kiri, kekanan. Setiap perubahan dalam proses pengecekan pola dengan teks dilakukan secara bertahap, yaitu satu langkah untuk setiap perubahan. Ini memungkinkan untuk mendapatkan lebih banyak informasi dari setiap proses pencocokan teks yang berjalan.

Analisis pencocokan menggunakan algoritma *brute*

force untuk mencari istilah DBMS pada penelitian kali ini dijabarkan pada tabel 2. Pada Tabel 2 menjabarkan proses tahap 1 yaitu analisis pencocokan terjadi ketidaksesuaian antara huruf yang ada pada pattern B dengan huruf text M pada index 0, maka pattern B bergeser sebanyak 1 langkah ke kanan.

TABEL 2
TAHAP 1 ANALISIS PENCOCOKAN

INDEKS	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
TEXT	M	O	D	E	L		B	A	S	I	S		D	A	T	A
PATTERN	B	A	S	I	S		D	A	T	A						

Selanjutnya pada tabel 3 menjelaskan tahap 2 analisis pencocokan terjadi ketidaksesuaian antara huruf yang

ada pada pattern B dengan huruf text O pada index 1, maka pattern B bergeser sebanyak 1 langkah ke kanan.

TABEL 3
TAHAP 2 ANALISIS PENCOCOKAN

INDEKS	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
TEXT	M	O	D	E	L		B	A	S	I	S		D	A	T	A
PATTERN		B	A	S	I	S		D	A	T	A					

Tahap selanjutnya pada tabel 4 di tahap 3 ini yaitu analisis pencocokan terjadi ketidaksesuaian antara huruf yang ada pada pattern B dengan huruf text D pada index

2, maka pattern B bergeser sebanyak 1 langkah ke kanan.

TABEL 4
TAHAP 3 ANALISIS PENCOCOKAN

INDEKS	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
TEXT	M	O	D	E	L		B	A	S	I	S		D	A	T	A
PATTERN			B	A	S	I	S		D	A	T	A				

Tahap selanjutnya pada tabel 5 tahap 4 analisis pencocokan terjadi ketidaksesuaian antara huruf yang

ada pada pattern B dengan huruf text E pada index 3, maka pattern B bergeser sebanyak 1 langkah ke kanan.

TABEL 5
TAHAP 4 ANALISIS PENCOCOKAN

INDEKS	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
TEXT	M	O	D	E	L		B	A	S	I	S		D	A	T	A
PATTERN				B	A	S	I	S		D	A	T	A			

PATTERN	B	A	S	I	S	D	A	T	A
---------	---	---	---	---	---	---	---	---	---

Tahap selanjutnya pada tabel 6, tahap 5 analisis pencocokan terjadi ketidaksamaan antara huruf yang ada pada pattern B dengan huruf text L pada index 4, maka pattern B bergeser sebanyak 1 langkah ke kanan.

TABEL 6
TAHAP 5 ANALISIS PENCOCOKAN

INDEKS	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
TEXT	M	O	D	E	L		B	A	S	I	S		D	A	T	A
PATTERN					B	A	S	I	S		D	A	T	A		

Pada tabel 7 tahap 6 analisis pencocokan terjadi ketidaksamaan antara huruf yang ada pada pattern B dengan huruf text kosong pada index 5, maka pattern B bergeser sebanyak 1 langkah ke kanan.

TABEL 7
TAHAP 6 ANALISIS PENCOCOKAN

INDEKS	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
TEXT	M	O	D	E	L		B	A	S	I	S		D	A	T	A
PATTERN						B	A	S	I	S		D	A	T	A	

Terakhir, pada tabel 8 tahap 7 analisis pencocokan terjadi sebuah persamaan antara huruf yang dipattern B dengan huruf text B pada index 6, maka pergeseran proses pencocokan dihentikan. text over text” tidak terpilih.

TABEL 8
TAHAP 7 ANALISIS PENCOCOKAN

INDEKS	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
TEXT	M	O	D	E	L		B	A	S	I	S		D	A	T	A
PATTERN							B	A	S	I	S		D	A	T	A

Berdasarkan dari hasil pengujian dengan menggunakan pencocokan string dan aplikasi yang digunakan dengan menggunakan Algoritma Brute Force pada aplikasi DBMS dengan jumlah data yang didapat dalam waktu yang dibutuhkan untuk melakukan pencarian dapat dilihat pada tabel 9.

TABEL 9
Hasil Waktu Pengujian

Keyword	Jumlah Data	Pencarian
Data	7	0.0165 Detik
DBMS	3	0.0177 Detik
Sistem	4	0.0154 Detik
Database	6	0.0174 Detik
Rata-Rata Waktu		0.01675 Detik

IV. KESIMPULAN

Dari hasil analisa penggunaan brute force untuk melakukan pencocokan string pada sebuah pola dapat disimpulkan bahwa algoritma ini cocok diterapkan dalam menyelesaikan masalah pencarian karena alur yang sangat sederhana juga mudah dimengerti dalam pengimplementasiannya, lebih efektif untuk data yang sangat terbatas, penggunaan code program yang dihasilkan juga tidak terlalu sulit serta jika dilihat dari hasil pengujian diperoleh hasil rata-rata waktu yang dibutuhkan dalam melakukan pencarian sebesar 0.01675 Detik. Hal tersebut selaras dengan tujuan dari penelitian ini dimana dengan menerapkan algoritma ini dapat membantu menyelesaikan permasalahan pencarian yang sederhana. Selain itu algoritma yang bersifat linear dapat memeriksa setiap indeks setiap string di temukan secara alfabet.

DAFTAR PUSTAKA

- [1] E. Setyawati, Sarwani, H. Wijoyo, and N. Soeharmoko, *Relational Database Management System (RDBMS)*. 2020. doi: 10.1201/9781420064636.ch4.
- [2] D. P. Rakhmadani, Y. D. Putra, and I. L. Dianti, “Transformasi Digital Pada Bisnis Umkm Dengan Penerapan Dbms,” *J. Eksek.*, vol. 17, no. 2, pp. 258–279, 2020.
- [3] I. Warman and R. Ramdaniansyah, “Analisis Perbandingan Kinerja Query Database Management System (DBMS) Antara Mysql 5.7.16 Dan Mariadb 10.1,” *J. Teknoif*, vol. 6, no. 1, pp. 32–41, 2018, doi:

- 10.21063/jtif.2018.v6.1.32-41.
- [4] T. Tematis, “Kamus Istilah,” *Badan Pengembangan dan Pembinaan Bahasa, Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Republik Indonesia*, 2016. https://kbki.kemdikbud.go.id/entri/kamus_istilah
- [5] R. Rismayani, N. Sambo Layuk, S. Wahyuni, H. Wali, and N. K. Marselina, “Pencarian Kata Pada Aplikasi Kamus Istilah Komputer dan Informatika Menggunakan Algoritma Brute Force Berbasis Android,” *Komputika J. Sist. Komput.*, vol. 10, no. 1, pp. 43–52, 2021, doi: 10.34010/komputika.v10i1.3644.
- [6] Yusmaida, Neneng, and A. Ambarwari, “Sistem Informasi Pencarian Kos Berbasis Web Dengan Menggunakan Metode Hill Climbing,” *J. Teknol. dan Sist. Inf.*, vol. 1, no. 1, pp. 68–74, 2020, [Online]. Available: <http://jim.teknokrat.ac.id/index.php/sisteminformasi>
- [7] A. W. Pratama and A. Dores, “Sistem Informasi Pencarian Pengepul Barang Bekas Di Kota Tangerang Berbasis Website,” *JUST IT J. Sist. Informatika, Teknol. Inf. dan Komput.*, vol. 9, no. 1, pp. 72–80, 2018, [Online]. Available: <https://jurnal.umj.ac.id/index.php/just-it/article/view/3229>
- [8] M. R. Azis, I. Fitri, and B. Rahman, “Penggunaan Algoritma Brute Force String Matching Dalam Pencarian Orang Hilang Pada Website Temukandia.Com,” *JIPi (Jurnal Ilm. Penelit. dan Pembelajaran Inform.)*, vol. 6, no. 2, pp. 205–212, 2021, doi: 10.29100/jipi.v6i2.1979.
- [9] N. Novianti, N. Marbun, M. Zarlis, and D. Hartama, “Analisis Pencarian Arti Istilah Statistika Menerapkan Algoritma Brute Force,” *Semin. Nas. Mat. dan Terap.*, vol. 1, pp. 881–886, 2019.
- [10] A. S. Sumi, P. Purnawansyah, and L. Syafie, “Analisa Penerapan Algoritma Brute Force Dalam Pencocokan String,” *Pros. SAKTI (Seminar Ilmu Komput. dan Teknol. Informasi)*, vol. 3, no. 2, pp. 88–92, 2018, [Online]. Available: <http://e-journals.unmul.ac.id/index.php/SAKTI/article/view/1836>
- [11] W. Anggraini, H. Meileni, S. Kom, D. Khairunissa, and M. Si, “Sistem Repository Tugas Akhir Mahasiswa Di Jurusan Manajemen Informatika Menggunakan Algoritma Brute Force Berbasis Website,” *Politek. Negeri Sriwij.*, pp. 1–7, 2018.
- [12] D. Indrawan and S. R. Jalilah, “Metode Kombinasi/Campuran Bentuk Integrasi Dalam Penelitian,” *J. Stud. Guru dan Pembelajaran*, vol. 4, no. 3, pp. 735–739, 2021, [Online]. Available: <https://e-journal.my.id/jsgp/article/view/1452>
- [13] I. P. Pratiwi, F. Ferdinandus, and A. D. Limantara, “Sistem Pendukung Keputusan Penerima Program Keluarga Harapan (PKH) Menggunakan Metode Simple Additive Weighting,” *J. Tek. Inform. Sitem Inf. dan Ilmu Komput.*, vol. 8, no. 2, 2019.
- [14] G. F. H. Nainggolan, S. Andryana, and A. Gunaryati, “Pencarian Berita Pada Web Portal Menggunakan Algoritma Brute Force String Matching,” *JIPi (Jurnal Ilm. Penelit. dan Pembelajaran Inform.)*, vol. 6, no. 1, pp. 1–10, 2021, doi: 10.29100/jipi.v6i1.1824.
- [15] T. Zebua, N. Silalahi, L. Batu, and S. Utara, “Aplikasi saran buku bacaan bagi pengunjung perpustakaan Amik Stiekom Sumatera Utara,” *J. Ris. Sist. Inf. Dan Tek. Inform.*, vol. 3, no. 3, pp. 66–72, 2018.
- [16] E. I. E. Putri and Y. N. Marlim, “Analisis Algoritma Brute Force dalam Aplikasi Correspondence Information System,” *Semin. Nas. Inform. Pros. Senat. 2021*, pp. 421–427, 2021.
- [17] B. Erdani, F. D. Aditia, S. Rodiah, Ciptyasih, and I. H. Santi, “Application System Dictionary of PHP Programming Language Terms Using Brute Force Algorithm,” *Multimed. Artif. Intell.*, vol. 3, no. 1, pp. 1–8, 2019, [Online]. Available: https://jurnal.itats.ac.id/wp-content/uploads/2015/10/6.-Anita-T.-Kumiawati-Afrilyan-Ruli-Dwi-Rama_EDITED.pdf
- [18] A. Sinaga and N. Nuraisana, “Implementasi Algoritma Brute Force Dalam Pencarian Menu Pada Aplikasi Pemesanan Coffee (Studi Kasus : Tanamera Coffee),” *J. Ilmu Komput. dan Sist. Inf.*, vol. 3, no. 3, pp. 303–313, 2021.
- [19] F. Firmansyah, Fauziah, and N. Hayati, “Analisis Perbandingan Dan Implementasi String Matching Dan Sql Query Pada Sistem Informasi Persediaan Obat Berbasis Web Apotek Erha Farma,” *J. Ilm. Teknol. dan*

Rekayasa, vol. 27, no. 2, pp. 154–168, 2022, doi:
10.35760/tr.2022.v27i2.7079.

- [20] A. A. Nababan and M. Jannah, “Algoritma String Matching Brute Force Dan Knuth-Morris-Pratt Sebagai Search Engine Berbasis Web Pada Kamus Istilah Jaringan Komputer,” *J. Mantik Penusa*, vol. 3, no. Desember, pp. 87–94, 2019.

PENERAPAN ALGORITMA BRUTE FORCE UNTUK PENCARIAN KAMUS ISTILAH SISTEM MANAJEMEN BASIS DATA.pdf

ORIGINALITY REPORT

5%

SIMILARITY INDEX

5%

INTERNET SOURCES

2%

PUBLICATIONS

%

STUDENT PAPERS

PRIMARY SOURCES

1

e-jurnal.pelitanusantara.ac.id

Internet Source

4%

2

ejournal.pelitaindonesia.ac.id

Internet Source

2%

Exclude quotes Off

Exclude bibliography On

Exclude matches < 2%