

Analisa Penerapan Algoritma *Brute Force* Dalam Pencocokan String

1st* Amin Siddiq Sumi
Fakultas Ilmu Komputer
Universitas Muslim Indonesia
Makassar, Indonesia
aminsiddiqsumi@gmail.com

2nd Purnawansyah
Fakultas Ilmu Komputer
Universitas Muslim Indonesia
Makassar, Indonesia
purnawansyah@umi.ac.id

3rd Lukman Syafie
Fakultas Ilmu Komputer
Universitas Muslim Indonesia
Makassar, Indonesia
lukman.syafie@umi.ac.id

Abstrak—Kebutuhan untuk menemukan informasi yang berguna dan cepat dalam suatu data yang besar sangat dibutuhkan. Karena kompleksitas data yang begitu banyak maka diperlukan suatu metode atau cara untuk dapat mencari suatu informasi yang diperlukan. Untuk melakukan pencarian sebuah data atau informasi tidak terlepas dari pencocokan *string* dimana dari hasil pencocokan inilah akan ditemukan pola kalimat yang dicari. Dalam penelitian ini membahas tentang penerapan algoritma *brute force* dalam melakukan pencocokan sebuah *string*. Algoritma ini melakukan pencocokan string dengan menggeser satu persatu *pattern* dan menyesuaikannya dengan teks hingga antara *pattern* dan teks memiliki pola yang sama. Hasil analisis dari penelitian ini berupa uji coba pencocokan *string* dengan algoritma *brute force* dengan studi kasus menggunakan mesin pencarian (*search engine*) dengan bahasa pemrograman PHP untuk pencocokan *string*.

Kata kunci—*brute force*; algoritma; *search engine*; *string matching*

I. PENDAHULUAN

Teks pada saat ini sudah dalam format digital, hal inilah yang semakin mempermudah penggandaan teks, sehingga jumlah teks semakin banyak. Dengan banyaknya jumlah teks ini menyebabkan adanya suatu kebutuhan untuk mencari suatu kata atau kalimat dalam teks tersebut. Untuk memenuhi kebutuhan tersebut digunakanlah Algoritma pencocokan string untuk mencari kata / kalimat dalam pola berbasis teks. algoritma pencocokan *string* digunakan untuk memperoleh informasi secara cepat dan efisien [1].

Secara defenisi algoritma adalah alur pemikiran yang logis yang dapat dituangkan ke dalam bentuk tulisan [2]. Kebutuhan untuk mencari informasi yang berguna dalam suatu data yang besar sangat dibutuhkan pada saat ini. Selain untuk mencari informasi yang tepat, pencarian *string* juga diharapkan untuk dapat menuntaskan pekerjaannya dengan cepat. Sehingga hasil dari pencarian itu, dapat digunakan tepat, cepat, dan efisien. Contoh dari penggunaan pencarian *string*, antara lain adalah *search engine* untuk *website*, pencarian kata di suatu dokumen, pembuatan program kamus, dan data mining [3].

Untuk memecahkan masalah ini, dibuatlah suatu algoritma yang memanfaatkan kemampuan komputer untuk menjalankannya. Algoritma ini melakukan pencarian dengan suatu jalan tertentu yang terurut, tidak ambigu, dan dapat dijalankan dengan baik oleh komputer.

Algoritma *string matching* biasa digunakan penerapannya dalam *search engine* (mesin pencarian), pencarian data dalam kamus dan lain-lain. Ada banyak algoritma yang ditemukan saat ini yang umum digunakan adalah *brute force* dan algoritma *knuth morris pratt* (KMP).

Dalam penelitian ini, penerapan algoritma *brute force* dalam pencocokan *string* dengan mesin pencarian (*search engine*) akan diimplementasikan. Adapun, bahasa pemrograman PHP akan digunakan sebagai alat bantu untuk pencocokan string.

II. METODOLOGI

A. *String Matching*

Secara umum, istilah yang terdapat dalam pencocokan *string* antara lain teks dan *pattern*. teks (*text*) adalah (long) *string* yang panjangnya n . *pattern* yaitu *string* dengan panjang m karakter ($m < n$) yang akan dicari di dalam teks [4]. Teks (*text*) yaitu *long string* yang panjangnya n karakter. Contoh implementasi *String matching* adalah pencocokan string pada *Microsoft word*, editor atau pencocokan website dengan memasukkan kata kunci sebagaimana yang telah diterapkan pada *search engine* seperti *yahoo*, atau *Google*. *String Matching* adalah proses pencarian semua kemunculan query yang selanjutnya disebut *pattern* ke dalam *string* yang lebih panjang atau teks. *String Matching* dirumuskan seperti pada persamaan (1) [5]:

$$\begin{aligned} x &= x [0 \dots m-1] \\ y &= y [0 \dots n-1] \end{aligned} \quad (1)$$

Dimana : x = *pattern* y = teks
 m = panjang *pattern* n = panjang teks

Kedua *string* terdiri dari sekumpulan karakter yang disebut alfabet yang dilambangkan dengan Σ (*sigma*) dan mempunyai ukuran σ (*tao*). *String matching* dibagi menjadi dua, yakni *exact matching* dan *heuristic* atau *statistical matching*. *Exact Matching* digunakan untuk menemukan.

Pattern yang berasal dari satu teks. Contoh pencarian *exact matching* adalah pencarian kata “pelajar” dalam kalimat “saya seorang pelajar” atau “saya seorang siswa”. Sistem akan memberikan hasil bahwa kalimat pertama mengandung kata “pelajar” sedangkan kalimat kedua tidak, meskipun kenyataannya pelajar dan siswa adalah kata yang bersinonim. Algoritma *exact matching* diklasifikasi menjadi tiga bagian menurut arah pencariannya.

B. Algoritma Brute Force

Brute force adalah sebuah pendekatan yang langsung (*straight forward*) untuk memecahkan suatu masalah, biasanya didasarkan pada pernyataan masalah (*problem statement*) dan definisi konsep yang dilibatkan. Algoritma *brute force* memecahkan masalah dengan sangat sederhana, langsung dan dengan cara yang jelas (*obvious way*). Di dalam pencocokan *string*, terdapat istilah teks dan *pattern*. Teks merupakan kata yang dicari dan dicocokkan dengan *pattern* [6]. Sedangkan *pattern* merupakan kata yang diinputkan untuk dicocokkan.. *Brute Force* merupakan algoritma pencarian *string* termudah. Dengan asumsi bahwa teks berada di dalam array T [1..n] dan *pattern* berada di dalam array P [1..m] maka algoritma *Brute Force* pencocokan *string* adalah sebagai berikut [7].

- Mula-mula *string* dicocokkan pada awal teks.
- Dengan bergerak dari kiri ke kanan, dibandingkan setiap karakter di dalam *string* dengan karakter yang bersesuaian di dalam teks, jika sesuai dibandingkan tersebut mengeluarkan hasil.
- Jika *string* belum ditemukan kecocokan dari teks belum habis, maka geser *string* satu karakter ke kanan dan berulang langkah ke 2.

C. C. MySQL

Database yang digunakan dalam pembuatan program ini adalah MySQL, MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU General *Public License* (GPL), tetapi mereka juga menjual di bawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem database (DBMS) dapat diketahui dari cara kerja *optimizer*-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program-program aplikasinya. Sebagai database

server, MySQL dapat dikatakan lebih unggul dibandingkan database server lainnya dalam query data. Hal ini terbukti untuk query yang dilakukan oleh single user, kecepatan query MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan Interbase.

D. Hypertext Preprocessor (PHP)

PHP (*Hypertext Preprocessor*) adalah kode/skrip yang akan dieksekusi pada server side [8]. Selain itu, PHP juga bisa digunakan sebagai bahasa pemrograman umum. PHP disebut bahasa pemrograman *server side* karena diproses pada komputer server. Hal ini berbeda dibandingkan dengan bahasa pemrograman *client-side* seperti JavaScript yang diproses pada *web browser (client)*.

Dalam beberapa tahun perkembangannya, PHP menjadi bahasa pemrograman web yang *powerful* dan tidak hanya digunakan untuk membuat halaman web sederhana, tetapi juga website populer yang digunakan oleh jutaan orang seperti wikipedia, wordpress, joomla, dan lain-lain. Saat ini PHP adalah singkatan dari PHP: *Hypertext Preprocessor*, sebuah kepanjangan rekursif, yakni permainan kata dimana kepanjangannya terdiri dari singkatan itu sendiri PHP: *Hypertext Preprocessor*.

III. HASIL DAN PEMBAHASAN

Perancangan dimulai dengan pembuatan database sebagai penampung data yang kemudian akan dijadikan sebagai data *training* ketika melakukan proses pencarian. Algoritma *brute force* akan melakukan pencocokan *string* dengan data yang telah tersimpan dalam sebuah database dengan bantuan bahasa pemrograman PHP. Berikut rancangan databasenya.

A. Tabel Database

Rancangan tabel database terdiri dari 4 record, ditunjukkan pada Tabel I.

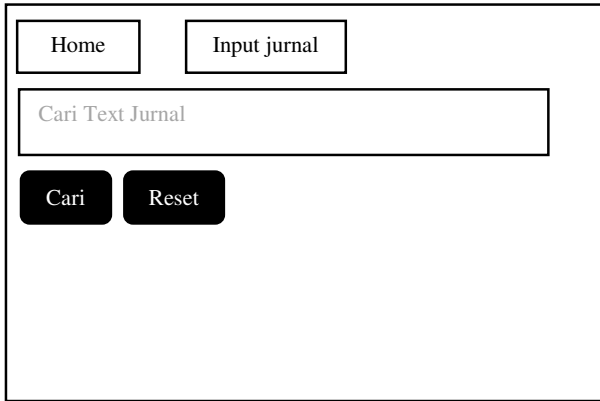
TABEL I. DATABASE

Nama Field	Type	Size	Ket
Id	Int	100	Primary Key
Nama_Jurnal	Varchar	100	-
Nama_Mahasiswa	Varchar	50	-
Abstrak	Text	-	-

Tabel 1, field terdiri dari id, nama jurnal, nama mahasiswa dan abstrak. id sebagai *primary key* yang berfungsi sebagai pembeda setiap *field* data yang tersimpan nama jurnal berisi judul jurnal terkait, nama mahasiswa berisi nama penulis dari sebuah jurnal dan abstrak berisi informasi singkat terkait sebuah jurnal.

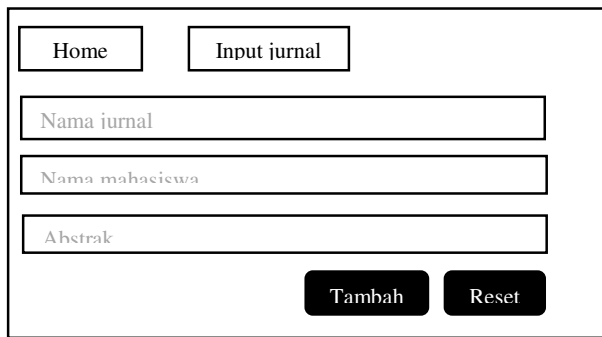
B. Antarmuka Tampilan

Selanjutnya setelah selesai merancang database, yaitu membuat desain antarmuka tampilan dari aplikasi pencocokan *string* dimana tampilan terdiri dari 2 bagian yaitu home, dan input jurnal.



Gambar. 1. Antarmuka menu home

Pada halaman menu home digunakan untuk melakukan penelusuran data yang diinputkan melalui text field yang disediakan.



Gambar. 2. Antarmuka menu input

Pada halaman menu input digunakan untuk melakukan input data baru yang kemudian akan tersimpan dalam database.

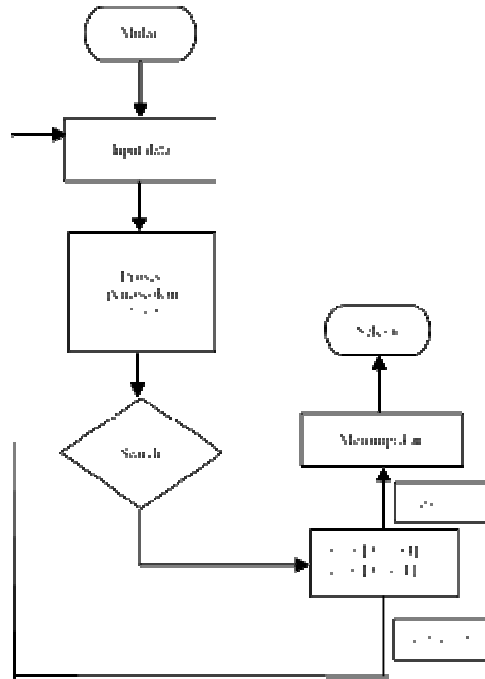
C. Flowchart Diagram

Metodologi analisis yang digunakan dalam tahapan penelitian ini, akan diuraikan dalam diagram alir (Flowchart). Gambar 3 menunjukkan flowchart analisa kinerja dari algoritma brute force

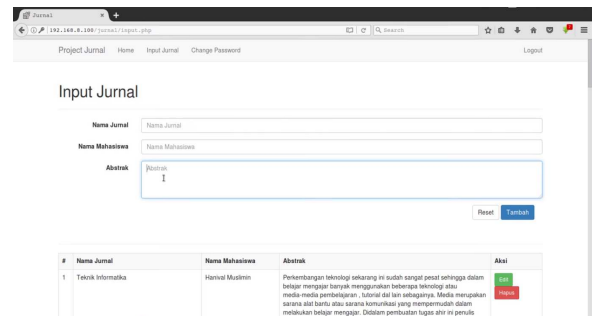
$$\begin{aligned} \text{Dimana : } x &= \text{pattern} & y &= \text{teks} \\ m &= \text{panjang pattern} & n &= \text{panjang teks} \end{aligned}$$

D. Hasil Implementasi

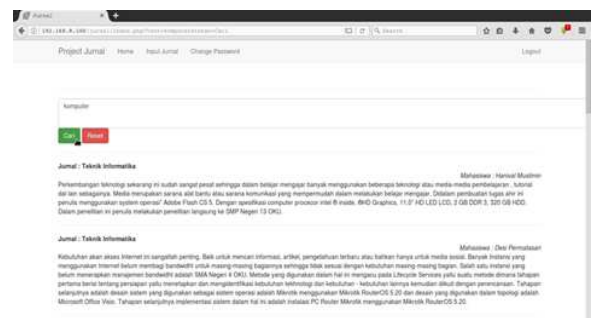
Hasil input data setelah berhasil tersimpan ke dalam database maka data yang telah dimasukkan akan ditampilkan kembali. Hasil input data dan input string ditunjukkan pada Gambar 4 dan Gambar 5.



Gambar. 3. Flowchart Diagram

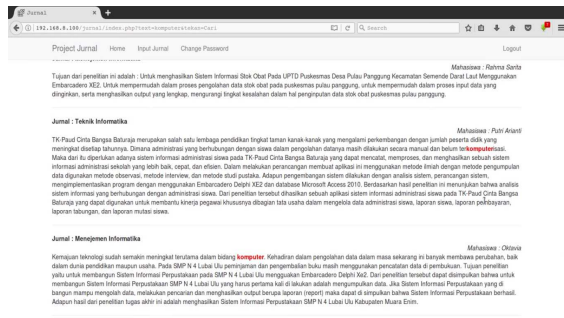


Gambar. 4. Input data



Gambar. 5. Input string pencarian

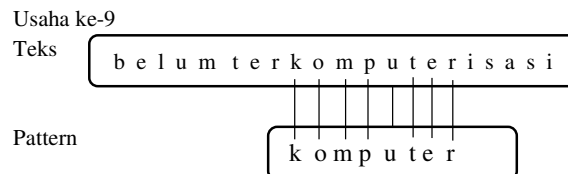
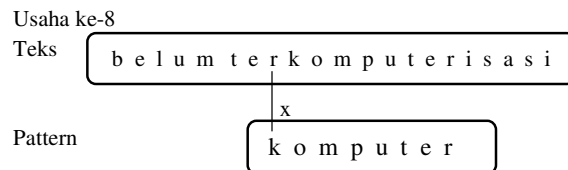
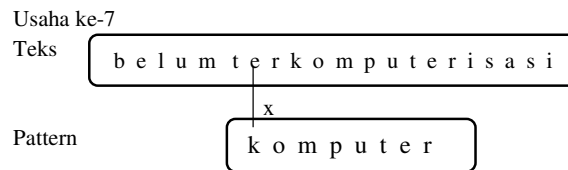
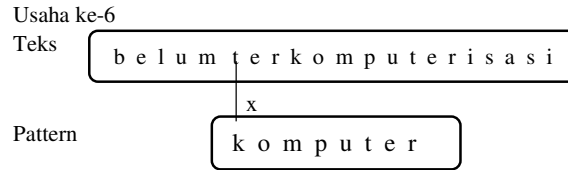
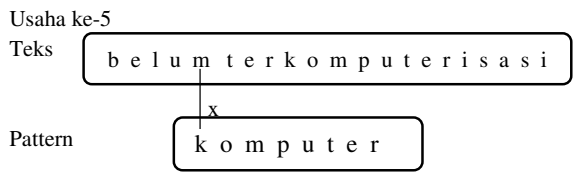
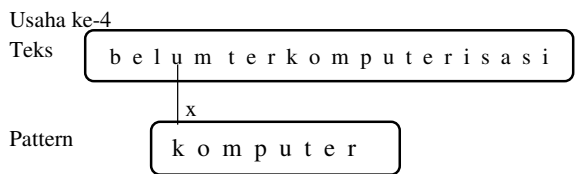
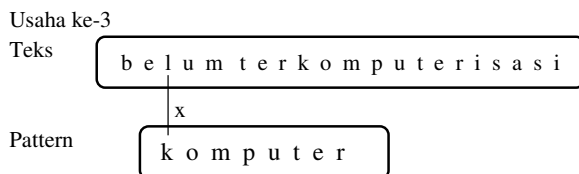
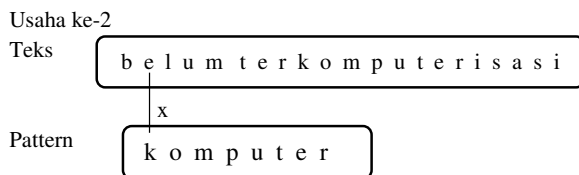
Contoh hasil pencocokan *string* yang ditelusuri adalah “KOMPUTER” ditunjukkan pada Gambar 6.



Gambar. 6. hasil pencocokan string

Hasil pencocokan *string* ditandai dengan *text* berwarna merah

E. Analisis Hasil Uji Coba



Gambar. 7. Iterasi uji coba

Ket : x = tidak cocok

Dari hasil analisis diatas menunjukkan bahwa algoritma *brute force* dalam melakukan proses pencarian dimulai dari teks paling kiri dengan indeks dimulai dari [1...n] selanjutnya jika *pattern* tidak cocok maka geser *pattern* sebanyak satu langkah kekanan menuju indeks berikutnya. Dari usaha yang dilakukan pattern cocok ditemukan pada usaha ke-9 dan indeks ke-9 jika pattern cocok pencarian dihentikan.

IV. KESIMPULAN

Dari hasil analisis pencocokan string, maka dapat diambil kesimpulan Algoritma *brute force* dapat digunakan untuk memecahkan hampir sebagian besar masalah karena alur yang sederhana dan mudah dimengerti dan menghasilkan algoritma yang layak untuk beberapa masalah penting seperti pencarian, pengurutan, pencocokan string, perkalian matriks serta menghasilkan algoritma baku (standar) untuk tugas-tugas komputasi seperti penjumlahan/perkalian *N* buah bilangan, dan menentukan elemen minimum atau maksimum ditabel.

DAFTAR PUSTAKA

- [1] M. Syarif, "Implementasi Algoritma String Matching Dalam Pencarian Surah Dan Ayat Dalam Al-Quran Berbasis Web," vol. 6, no. 2, pp. 70–76, 2017.
- [2] H. Pratiwi et al., "IMPLEMENTASI ALGORITMA BRUTE FORCE DALAM," vol. II, no. 2, pp. 119–125, 2016.
- [3] M. Akbar, I. Fuadina, and J. Pramadesa, "Algoritma Pencarian String dengan Menggunakan metode Brute Force yang diperkaya," pp. 1–3.
- [4] H. Fernando and A. Boyer-, "Perbandingan dan Pengujian Beberapa Algoritma Pencocokan String," no. 10, 2009.
- [5] Mesran, "Implementasi Algoritma Brute Force Dalam Pencarian," Inf. dan Teknol. Ilm., 2014.
- [6] U. S. Utara, "Universitas Sumatera Utara," 2017.
- [7] R. B. Abuhari, J. T. Informatika, F. Sains, and D. A. N. Teknologi, "Rancang bangun aplikasi penelusuran skripsi menggunakan algoritma brute force," 2016.
- [8] A. Vironica, "Rancang Bangun Aplikasi Pengelolaan Surat Masuk Dan Surat Keluar Pada Sekolah Menengah Pertama Negeri 2 Nawangan," vol. 11, no. 1, pp. 34–41, 2014.