

Penggunaan Algoritma Boyer-Moore dalam Penentuan Artikel yang Aman bagi Anak

Sashi Novitasari/13514027

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

13514027@std.stei.itb.ac.id

Abstract—Di zaman ini, media informasi berkembang cukup pesat. Semua orang dapat mendapatkan berbagai informasi dari media tersebut, termasuk anak-anak. Anak-anak dapat dengan mudah mengakses internet untuk mencari berbagai informasi. Walau pun internet memudahkan anak untuk mencari informasi, akses informasi yang bebas dapat merusak moral anak karena tidak semua informasi mengandung konten yang positif. Agar anak tidak dapat mengakses informasi seperti itu, suatu sistem berupa filter diperlukan. Filter akan menentukan apakah informasi aman atau tidak, dan penentuan tersebut dapat dilakukan dengan basis pencocokan *string*, yaitu pencocokan *string* dengan algoritma Boyer-Moore.

Keywords—*string matching*, algoritma Boyer-Moore, filter informasi, anak-anak.

I. PENDAHULUAN

Anak-anak merupakan anggota termuda muda dalam masyarakat dunia, yang merupakan generasi penerus masyarakat tersebut. Setiap orang mendapatkan pendidikan nilai atau pun norma yang penting pada saat masih anak-anak. Pendidikan tersebut berpengaruh besar terhadap pembentukan dan perkembangan karakter anak. Bila terdapat kesalahan dalam penyampaian nilai, antara lain cara mau pun konten, maka karakter anak dapat berkembang ke arah yang tidak baik. Seluruh orang tua tentunya menghindari hal tersebut terjadi dan mengusahakan yang terbaik agar anak mereka dapat tumbuh berkembang menjadi pribadi yang baik.

Pada zaman ini, media informasi digital, yaitu internet, berkembang sangat pesat. Kini internet sangat mudah diakses dan dapat digunakan oleh siapa pun, termasuk anak-anak. Internet sangat membantu anak dalam menambah wawasan dan pengetahuan akan informasi. Informasi yang terdapat di internet sangat banyak dan mencakup berbagai topik.

Tidak semua topik informasi baik untuk anak-anak. Dikarenakan akses internet yang sangat bebas, pembatasan atau pemilahan informasi yang aman dan tidak aman untuk anak perlu dilakukan. Contoh dari informasi yang tidak aman untuk

anak adalah informasi menyangkut pornografi dan kekerasan. Rasa keingintahuan anak terhadap suatu hal sangatlah besar. Bila mereka melihat atau mendapatkan sedikit dari informasi tersebut, mereka akan terus mencari informasi tersebut. Dikarenakan umur, pengalaman, dan pengetahuan yang masih belia, anak-anak cenderung tidak dapat membedakan baik dan tidaknya suatu informasi. Oleh karena itu, bila mereka sempat mendapatkan sedikit saja mengenai informasi dengan topik yang tidak baik, mereka mungkin akan terus melakukan pencarian informasi terkait topik tersebut dan mengadopsi nilai yang terdapat pada informasi tersebut tanpa disadari bahwa nilai tersebut merupakan nilai yang tidak baik. Sudah bukan hal yang mengagetkan bila kita melihat suatu berita yang memberitakan suatu kasus kekerasan dan pornografi yang dilakukan oleh anak karena mendapatkan informasi mengenai hal tersebut di internet.

Penyaringan informasi yang tepat bagi anak-anak perlu dilakukan. Agar karakter anak tidak rusak dengan mengakses internet, pembatasan informasi merupakan hal yang sangat krusial untuk dilakukan. Dengan melakukan penyaringan, anak-anak tidak akan mendapatkan informasi yang tidak baik, tetap berada dalam kondisi ‘tidak tahu’ sehingga tidak merasa penasaran akan konten dari informasi yang kurang baik.

Penyaringan informasi dalam suatu bacaan dari internet dapat dilakukan dengan melakukan pemeriksaan pada konten. Bila bacaan tersebut mengandung suatu kata yang buruk atau menjadi ciri dari informasi yang tidak tepat bagi anak, maka bacaan akan dikategorikan sebagai bacaan yang tidak tepat bagi anak. Pada saat ini, terdapat banyak aplikasi internet atau berupa *search engine* yang ditargetkan untuk pengguna anak-anak, contohnya adalah *KidRex.org* dan *Gogooligans.com*. Bila pengguna melakukan pencarian informasi melalui aplikasi tersebut, aplikasi hanya akan menampilkan hasil pencarian yang dikategorikan aman bagi anak. Informasi yang tidak aman tidak akan ditampilkan.



Gambar 1. Tampilan *KidRex.org*, browser untuk anak-anak. (Sumber: <http://www.123ict.co.uk/searches-children/> diakses pada 8 Mei 2016)

Pemeriksaan konten informasi yang berupa bacaan dapat dilakukan dengan metode pencocokan kata, dalam hal ini menggunakan metode pencocokan *string* dengan algoritma Boyer-More. Metode tersebut digunakan untuk melakukan pencarian kata yang tidak baik untuk anak.

II. DASAR TEORI

2.1 Pencarian String

Algoritma pencarian *string* merupakan suatu jenis algoritma yang digunakan untuk mencari dan menentukan keterkandung suatu kata atau *pattern* pada sebuah teks yang lebih panjang dibandingkan dengan *pattern* tersebut. Algoritma pencarian *string* dapat dibagi menjadi dua kategori¹, yaitu:

1. Exact String Matching

Dimisalkan terdapat teks $T = t_1t_2t_3...t_n$ dan pola atau kata $P = p_1p_2...p_m$. Pada *exact string matching* atau pencocokan *string* yang tepat, pencarian kata P pada teks T akan dilakukan¹. Algoritma ini melakukan pemeriksaan apakah kata yang terdapat pada teks T sama dengan P . Berikut adalah contoh ilustrasi dari *exact string matching*:

$P = ababaaac$

$T = ahsuhd \underline{ababaa}csad$

Bila menggunakan metode ini, keluran dari hasil pencarian adalah pola ditemukan di dalam teks. Contoh algoritma merupakan algoritma *exact string matching* adalah algoritma Knuth-Morris-Pratt dan algoritma Boyer-Moore².

2. Approximate String Matching

Approximate String Matching merupakan algoritma pencarian *string* yang berdasarkan pada perkiraan. Dimisalkan terdapat dua buah *string* yang berbeda, yaitu S_1 dan S_2 . Dengan menghirung fungsi jarak terlebih dahulu untuk menentukan kemiripan dari kedua *string* ini, S_2 dapat ditransformasikan menjadi S_1 dengan melalui 3 operasi, yaitu operasi penghapusan, operasi penyisipan, operasi substitusi¹. Contoh algoritma yang dikategorikan sebagai algoritma *approximate string matching* adalah algoritma Levenshtein Distance³.

2.2 Algoritma Boyer-Moore

Algoritma Boyer-Moore merupakan algoritma *exact string matching* yang dibuat oleh R.M. Boyer dan J.S Moore pada tahun 1997⁴. Seperti algoritma *exact string matching* lainnya, algoritma ini digunakan untuk mencari suatu *string* di dalam suatu teks. Algoritma ini dikatakan sebagai algoritma pencarian *string* terbaik secara rata-rata bila dibandingkan dengan algoritma pencarian *string* lainnya, contohnya algoritma Knuth-Morris-Pratt. Oleh karena itu, algoritma Boyer-Moore menjadi algoritma yang paling banyak digunakan dalam berbagai aplikasi yang menyediakan fasilitas pencarian berbasis teks⁴.

Algoritma pencocokan *string Boyer-Moore* (BM) didasarkan oleh 2 teknik yaitu *looking-glass technique* dan *character-jump technique*⁵. *Looking-glass technique* merupakan suatu teknik pergeseran *pattern* ke arah belakang ketika pencarian *pattern* pada teks dilakukan. Sedangkan *character-jump technique* merupakan teknik pergeseran *pattern*, yang akan digunakan pada saat terjadi ketidakcocokan (*mismatch*) antara karakter yang sedang dicocokkan pada *pattern* dan teks.

2.2.1 Langkah Pencarian

Langkah pencarian yang dilakukan oleh algoritma Boyer-Moore terdiri atas 2 tahap, yaitu tahap *preprocessing* dan tahap pencocokan.

1. Tahap Preprocessing

Pada tahap *preprocessing*, fungsi atau tabel pinggiran yang berdasarkan pada *pattern* akan ditentukan. Fungsi pinggiran digunakan untuk menentukan pergeran *pattern* pada *character-jump technique*, yaitu bila terjadi ketidakcocokan karakter atau *mismatch* pada saat tahap pencocokan, *pattern* dapat digeser sehingga karakter yang menyebabkan *mismatch* pada teks bercocokan dengan karakter yang sama dan berada di posisi terkanan pada *pattern* ($T[i]=P[j]$). Teknik ini disebut juga sebagai pergeseran *bad-character*⁶. Tabel tersebut akan menyimpan informasi mengenai indeks kemunculan terakhir setiap karakter pada *pattern* ($L(x)$)⁶. Contoh dari tabel fungsi pinggiran adalah sebagai berikut:

Pattern = aaabdbdeaccii

Tabel =

x	a	b	d	e	C	I
L(x)	8	5	6	7	10	11

Algoritma (dalam bahasa C++) pada tahap ini adalah sebagai berikut⁷:

```
int* buildLast(string pattern){
    int* last = new int[256];
```

```

for (int i = 0; i < 256; i++)
    last[i] = -1;

for (int i = 0; i < pattern.size(); i++){
    int dummy = (int)pattern[i];
    if (dummy < 0) 1 += 128;
    last[dummy] = i;
}
return last;
}

```

2. Tahap Pencocokan

Berikut adalah langkah-langkah yang dilakukan dalam tahap pencocokan.

- 1) Pencocokan *string* dimulai dari bagian awal dari teks.
- 2) Dari kanan ke kiri, algoritma akan mencocokkan karakter per karakter pada *pattern* dengan karakter di teks yang posisinya bersesuaian, sampai salah satu kondisi berikut dipenuhi:
 - a) Karakter pada *pattern* dan pada teks yang sedang dibandingkan tidak cocok
 - b) Semua karakter pada *pattern* cocok. Pada titik ini, algoritma akan mengembalikan suatu nilai yang menandakan bahwa teks mengandung *pattern*.
- 3) Bila $T[i] = x$ sedangkan karakter $P[j] \neq T[i]$, maka terdapat 3 kemungkinan kasus, yaitu⁵:
 - a) Jika P mengandung karakter x, maka akan dilakukan pergeseran ke kanan, di mana letak x terdapat pada P dan diposisikan pada $T[i]$.
 - b) Jika P mengandung karakter x tetapi x tersebut tidak berada pada sebelah kiri $P[j]$, maka akan dilakukan pergeseran ke arah kanan sebanyak 1 karakter, yaitu $P[j]$ sekarang berada pada $T[i+1]$.
 - c) Jika kasus 1 dan 2 tidak dapat dilakukan, maka geser $P[1]$ ke $T[i+1]$.

Sesudah pergeseran dilakukan, langkah kedua akan kembali dilakukan sampai salah satu kondisi pada langkah 2 terpenuhi atau *pattern* sudah berada di ujung teks sehingga pergeseran sudah tidak dapat dilakukan (menandakan bahwa teks tidak mengandung *pattern*)

Algoritma proses ini dalam bahasa C++ adalah sebagai berikut⁷:

```

int bmMatch(string text, string pattern){
    int* last = buildLast(pattern);
}

```

```

int n = text.length();
int m = pattern.size();
int i = m - 1;

if (i > n - 1) return -1;
int j = m - 1;
do{
    if (pattern[j] == text[i]) {
        if (j == 0) return i;
        else{
            i--;
            j--;
        }
    }
    else{
        int lastidx = text[i];
        if (lastidx < 0) lastidx += 128;
        int lo = last[lastidx];
        i = i + m - std::min(j, 1 + lo);
        j = m - 1;
    }
} while (i <= n - 1);
return -1;
}

```

Algoritma akan mengembalikan posisi karakter pertama *pattern* pada teks bila *pattern* terdapat pada teks atau mengembalikan nilai -1 bila teks tidak mengandung *pattern*.

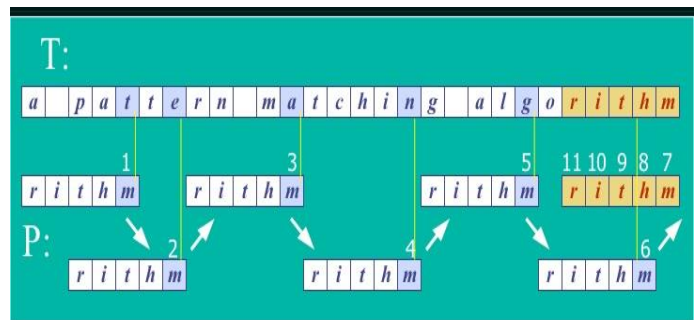
Berikut adalah contoh pencocokan *string* dengan menggunakan algoritma Boyer-Moore:

Pattern : rithm
Teks : a pattern matching algorithm

- Langkah *preprocessing*

x	r	i	t	h	m
L(x)	1	2	3	4	5

- Langkah pencarian:



Gambar 2. Ilustrasi Langkah pencarian dengan algoritma Boyer-Moore⁷.

2.2.2 Kompleksitas Waktu

Diberikan sebuah *pattern* P dengan panjang m karakter dan teks T dengan panjang n . Maka, kompleksitas waktu untuk menentukan fungsi pinggiran adalah $O(m + \mathcal{O})$, dengan \mathcal{O} adalah ukuran tabel fungsi pinggiran. Dalam keadaan terburuk, kompleksitas waktu pada langkah pencocokan adalah $O(mn)$. Sehingga, bila dimisalkan $A = m + \mathcal{O}$, maka total kompleksitas waktu pada keadaan terburuk (*worst case*) dari algoritma Boyer-Moore adalah $O(mn + A)^7$. Sedangkan dalam keadaan terbaik, algoritma ini akan memiliki kompleksitas waktu $O(n/m)^8$.

2.3 Webfilter

Webfilter merupakan suatu subsistem perangkat lunak yang digunakan oleh situs atau *browser*, untuk menyaring situs dengan konten tertentu. *Webfilter* hanya akan menampilkan situs yang lolos dari proses penyerangan, dengan kata lain konten situs yang aman untuk ditampilkan pada pengguna⁹. Pada umumnya, browser yang ditujukan untuk pengguna anak-anak memanfaatkan sistem ini. Pada saat pengguna melakukan pencarian dengan *browser* tersebut, *browser* akan melakukan filter terhadap situs yang terdapat pada hasil pencarian sistem, dan hanya akan menampilkan situs dengan konten yang aman bagi anak-anak.

2.4 Dampak Internet Terhadap Moral Anak

Pada saat ini, internet penuh dengan konten yang tidak tepat bagi anak¹⁰. Contoh dari informasi pada internet yang tidak tepat bagi anak adalah informasi mengenai kekesaran serta pornografi. Informasi-informasi seperti itu dapat mengakibatkan perilaku menyimpang bagi anak. Pencarian informasi pada internet terkadang menyertakan hasil pencarian yang berisi konten yang telah disebutkan sebelumnya. Karena anak-anak memiliki rasa keingintahuan yang cukup besar, bila konten tersebut dibuka, anak bisa menjadi tertarik untuk mencari informasi lainnya terkait konten, atau bahkan menirukan hal-hal terkait konten negatif tersebut¹¹. Hal tersebut akan menimbulkan sikap asusila, amoral, serta sikap agresif pada anak. Agar hal seperti ini terhindar, salah satu cara yang dapat dilakukan untuk mengurangi akses anak terhadap informasi negatif adalah dengan menggunakan aplikasi *webfilter* pada browser. Pada saat ini, terdapat banyak *browser* yang ditujukan untuk pengguna anak, dan *browser-browser* tersebut menggunakan *webfilter* agar situs dengan konten yang kurang baik bagi anak tidak ditampilkan pada hasil pencarian.

III. PENGGUNAAN ALGORITMA BOYER-MOORE DALAM MENENTUKAN ARTIKEL YANG AMAN BAGI ANAK

Algoritma Boyer-Moore digunakan untuk menemukan suatu kata tertentu pada suatu teks. Oleh karena itu, algoritma

ini dapat digunakan dalam proses penentuan artikel yang aman bagi anak, yaitu pada proses filter atau penyaringan. Pertama-tama program atau algoritma akan mencari keberadaan seluruh kata negatif pada teks artikel dengan menggunakan algoritma Boyer-Moore. Bila program menemukan salah satu atau beberapa kata tersebut pada teks artikel, maka artikel digolongkan sebagai artikel yang mengandung konten berbahaya bagi anak. Namun bila tidak terdapat satu pun kata negatif tersebut, maka artikel dinyatakan aman bagi pembaca anak-anak.

3.1 Implementasi dan Pengujian

Sebelum proses pemeriksaan atau penyaringan dilakukan, daftar kata yang negatif atau kata yang menjadi ciri dari artikel yang tidak aman bagi anak disimpan dalam suatu larik. Pengujian dilakukan dengan menggunakan program pemeriksa yang merepresentasikan fungsi filter pada *webfilter* (oleh karena itu program tidak mencakup tahap pencarian artikel di internet). Pada program pemeriksa, kata-kata yang dimasukkan dalam larik adalah 'bunuh', 'pelecehan', 'pukul', 'kekerasan', 'porno', 'tinju', dan 'smack down'.

Setelah program menerima masukan *file* yang berisikan artikel yang akan diperiksa, pemeriksaan pun dilakukan. Pemeriksaan akan dilakukan dengan mencari keberadaan setiap *pattern*, yang terdapat pada larik, pada teks atau artikel. Bila artikel mengandung salah satu dari kata tersebut, maka artikel akan dikategorikan sebagai artikel yang memiliki konten berbahaya bagi anak dan tidak menampilkan artikel tersebut. Namun bila artikel tidak mengandung satu pun *pattern* yang terdapat pada larik, maka artikel dikategorikan aman dan ditampilkan oleh program. Berikut adalah contoh hasil pengujian program penentu artikel yang aman bagi anak, yang menggunakan algoritma Boyer-Moore:

1. Pengujian 1

- Teks artikel:

<p>Pembunuhan sadis yang menimpa dosen Universitas Muhammadiyah Sumatera Utara (UMSU) Nur'ain Lubis (63) yang terjadi Senin 2 Mei 2016 kemarin, akhirnya terkuak. Aksi itu dipicu dendam Roymardo Sah Siregar karena acap dimarahi korban. "Tersangka memiliki dendam terhadap korban yang sudah terjadi sekitar satu bulan. Tersangka merasa sakit hati karena korban selalu memarahi tersangka ketika tidak memperhatikan korban mengajar," kata Kapolresta Medan Kombes Pol Mardiaz Kusin Dwihananto, Medan, Selasa (3/5/2016). Sakit hati tersangka semakin menjadi-jadi ketika korban juga sering mengancam akan memberikan nilai jelek kepadanya. Roymardo dinilai tidak serius mengikuti mata kuliah yang diajarkan korban.</p>
--

- Hasil:

```

=====
FILTER ARTIKEL
yang aman bagi anak
=====
Nama file artikel : teks
Hasil: Artikel ini mengandung konten
yang berbahaya bagi anak.

Waktu eksekusi : 5 microseconds

```

Gambar 3. Hasil pengujian 1 dengan menggunakan program pemeriksa

2. Pengujian 2

- Teks artikel:

Dampak Positif Mengikuti Lomba Anak

Lomba-lomba anak tersebut memang banyak manfaatnya, dari mengembangkan potensi anak sampai memupuk rasa percaya diri mereka. Seperti pada lomba mewarnai, anak belajar untuk percaya pada kemampuannya, mencocokkan warna, mengambil keputusan warna yang digunakan, memahami media dan alat yang digunakan, serta mempraktekkan tehnik mewarnai yang sudah dipelajarinya. Dengan ikut lomba si anak akan makin mengasah kemampuannya menjadi lebih baik lagi.

- Hasil:

```

=====
FILTER ARTIKEL
yang aman bagi anak
=====
Nama file artikel : aman
Hasil: Artikel ini aman bagi anak.

Dampak Positif Mengikuti Lomba Anak

Lomba-lomba anak tersebut memang banyak ma
nfaatnya, dari mengembangkan potensi anak
sampai memupuk rasa percaya diri mereka. S
eperti pada lomba mewarnai, anak belajar u
ntuk percaya pada kemampuannya, mencocokk
an warna, mengambil keputusan warna yang dig
unakan, memahami media dan alat yang digun
akan, serta mempraktekkan tehnik mewarnai
yang sudah dipelajarinya. Dengan ikut lomb
a si anak akan makin mengasah kemampuannya
menjadi lebih baik lagi.

Waktu eksekusi : 26 microseconds

```

Gambar 4. Hasil pengujian 2 dengan menggunakan program pemeriksa

3.2 Pembahasan

Pengujian dilakukan pada dua buah artikel yang salah satunya mengandung kata yang negatif dan lainnya tidak. Teks yang digunakan pada pengujian 1 mengandung kata 'bunuh' yang merupakan salah satu kata yang terdapat pada larik yang berisi kata-kata negatif. Oleh karena itu, teks artikel tersebut ditentukan sebagai artikel yang tidak aman bagi anak-anak dan teks tidak ditampilkan pada layar. Berbeda dengan teks pada pengujian 1, teks artikel pada pengujian 2 tidak mengandung satu pun kata yang terdapat pada larik. Maka dari itu, hasil dari pemeriksaan menunjukkan bahwa artikel aman dibaca oleh anak-anak dan teks ditampilkan pada layar program.

Waktu yang dibutuhkan untuk melakukan pemeriksaan pada kedua pengujian berjalan kurang dari 1 milidetik. Pada pengujian 1, panjang teks yang digunakan adalah 703 karakter. Walau pun pada larik terdapat 7 buah *string* dengan panjang rata-rata 7 karakter, pemeriksaan dengan algoritma Boyer-Moore tidak dilakukan sebanyak 7 kali. Pemeriksaan dihentikan bila algoritma menemukan *pattern* pada larik pada teks. Waktu pemeriksaan yang dibutuhkan pengujian 1 adalah 5 mikrodetik. Pada pengujian 2, panjang teks yang digunakan adalah 480 karakter. Teks tersebut tidak mengandung kata yang terdapat pada larik, maka pemeriksaan dilakukan sebanyak 7 kali (sesuai jumlah *string* pada larik). Waktu yang dibutuhkan pengujian 2 untuk menyelesaikan seluruh pemeriksaan adalah 26 mikrodetik, sehingga setiap pemeriksaan rata-rata membutuhkan waktu $(26/7)$ mikrodetik.

IV. KESIMPULAN

Pembatasan informasi-informasi negatif harus dilakukan agar anak-anak tidak mengakses atau pun mendapatkan informasi tersebut. Dikarenakan pengetahuannya yang masih belia dan belum bisa membedakan hal yang baik/buruk dengan benar, anak-anak bisa saja mengadopsi nilai dari informasi yang memiliki konten yang tidak tepat bagi anak. Bentuk pembatasan dapat dilakukan dengan menggunakan sistem penyaring informasi yang menggunakan algoritma Boyer-Moore.

Algoritma Boyer-Moore dapat digunakan untuk menentukan apakah suatu artikel aman atau tidak dibaca oleh anak. Algoritma tersebut berperan sebagai filter, yaitu memeriksa keberadaan suatu kata negatif pada teks artikel. Dikarenakan algoritma ini hanya dapat melakukan pencocokan *string* yang sudah tersedia sebelumnya, pendefinisan daftar kata yang berbahaya harus dilakukan oleh pengguna. Namun secara keseluruhan, algoritma cukup berjalan dengan baik karena hasil yang dikeluarkan sesuai dengan kategori artikel dan waktu eksekusi tidak lama (kasus terburuk dengan teks 480 karakter memerlukan waktu $(26/7)$ detik untuk melakukan pemeriksaan).

UCAPAN TERIMA KASIH

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas segala rahmat-Nya sehingga makalah ini dapat diselesaikan dengan baik. Penulis mengucapkan terima kasih kepada Dr. Ir. Rinaldi Munir, M.T dan Dr. Nur Ulfa Maulidevi, S.T, M.Sc. selaku dosen pengajar mata kuliah Strategi Algoritma atas segala bimbingan dan ilmu yang telah diberikan kepada penulis. Selanjutnya, penulis mengucapkan terima kasih kepada semua sumber serta pihak yang telah memberikan bantuan secara langsung mau pun tidak langsung dalam proses pembuatan makalah. Semoga dengan dibuatnya makalah ini, makalah ini dapat berguna bagi seluruh pembaca.

REFERENSI

- [1] R.C.T. Lee, dkk. "Exact and Approximate String Matching Algorithms". http://t2.ecp168.net/webs@73/cyberhood/StringMatching/StringMatchingAlgorithm/Chapter_1_Definitions_of_String_Matching_Algorithms.doc (diakses pada 6 Mei 2016)
- [2] Christian Charras, Thierry Lecroq. 1997. "Exact String Matching Algorithm". <http://www-igm.univ-mlv.fr/~lecroq/string/index.html> (diakses pada 6 Mei 2016).
- [3] Muhammad Bahari Ilmy, Nitia Rahmi, Roland L Bu'ulölö. 2005. "Penerapan Algoritma Leveshtein Distance untuk Mengoreksi Kesalahan Pengejaan pada Editor Text". <http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2005-2006/Makalah2006/MakalahStmik2006-30.pdf>. (diakses pada 6 Mei 2016)
- [4] Hendra. 2011. "Kasus Pada Algoritma Boyer-Moore". <http://technoprast.blogspot.co.id/2011/04/kasus-pada-algoritma-boyer-moore.html>. (diakses pada 6 Mei 2016)
- [5] H.W. Lang. 2001. "Boyer Moore Algorithm". <http://www.inf.fh-flensburg.de/lang/algorithmen/pattern/bmen.htm>. (diakses pada 6 Mei 2016)
- [6] Muhtar Hartopo. 2014. "Algoritma Pencocokan String untuk Mendeteksi Penyakit Turunan". https://www.academia.edu/12327660/Algoritma_Pencocokan_String_untuk_Mendeteksi_Penyakit_Turunan. (diakses pada 6 Mei 2016)
- [7] Andrew Davidson. 2006. "Pattern Matching" dalam Munir, Rinaldi (E.d.). [http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2014-2015/Pencocokan%20String%20\(2015\).ppt](http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2014-2015/Pencocokan%20String%20(2015).ppt) (diakses pada 7 Mei 2016)
- [8] Christian Charras, Thierry Lecroq. 1997. "Boyer-Moore algorithm". <http://www-igm.univ-mlv.fr/~lecroq/string/node14.html#SECTION00140>. (diakses pada 7 Mei 2016).
- [9] Marget Rouse. 2005. "Web filter". <http://searchsecurity.techtarget.com/definition/Web-filter>. (diakses pada 7 Mei 2016)
- [10] Eric Geier. 2013. "How to child-proof the Internet". <http://www.pcworld.com/article/2042233/how-to-child-proof-the-internet.html>. (diakses pada 7 Mei 2016)
- [11] Muhammad Najib. 2016. "Menyelamatkan Anak dari Dampak Negatif Internet". <http://harian.analisadaily.com/opini/news/menyelamatkan-anak-dari-dampak-negatif-internet/221836/2016/03/15>. (diakses pada 7 Mei 2016)

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 9 Mei 2016



Sashi Novitasari/13514027