ISSN print 2087-1716
ISSN online 2548-7779

ILKOM Jurnal Ilmiah Volume 9 Nomor 2 Agustus 2017

ANALISIS *STRING MATCHING* PADA JUDUL SKRIPSI DENGAN ALGORITMA *KNUTH-MORRIS PRATT* (KMP)

Wistiani Astuti
whistieruslank@gmail.com
Teknik Informatika, Universitas Muslim Indonesia

Abstrak

Skripsi adalah suatu karya ilmiah mahasiswa yang disusun dalam rangka memenuhi sebagian syarat penyelesaian studi pada program strata satu (S-1) di setiap Perguruan Tinggi Negeri maupun Swasta yang ada di Indonesia. Salah satu upaya yang dapat dilakukan dalam menentukan kesamaan judul skripsi ialah dengan melakukan pencocokan *string* (*string matching* atau *pattern matching*) dalam teks yang terdapat pada judul skripsi yang diusulkan. Pada penelitian ini menggunakan Algoritma *Knuth-Morris-Pratt* (KMP) untuk menganalisis bagaimana proses pencocokan *string* yang dihasilkan dan membandingkan sejauh mana nilai kemiripan dari beberapa judul yang sama dan serupa sehingga dapat memberikan suatu informasi yang efektif bagi mahasiswa.

Kata Kunci : Skripsi, *string matching*, algoritma *Knuth-Morris Pratt* (KMP)

1. Pendahuluan

Skripsi adalah suatu karya ilmiah mahasiswa yang disusun dalam rangka memenuhi sebagian syarat penyelesaian studi pada program strata satu (S-1) di setiap Perguruan Tinggi Negeri maupun Swasta yang ada di Indonesia [7]. Skripsi merupakan tugas akhir mahasiswa untuk menyelesaikan studi S1 yang bersifat mandiri dan wajib untuk mendapatkan gelar sarjana Strata Satu (S-1). Setiap mahasiswa harus mencari topik penelitian terlebih dahulu untuk menentukan fokus penelitiannya. Mahasiswa melakukan pencarian topik tugas akhir, seperti membaca jurnal penelitian baik lokal, nasional maupun internasional, mengikuti penelitian yang dilakukan dosen, membaca kumpulan tugas akhir yang pernah dibuat oleh mahasiswa sebelumnya, serta melakukan observasi masalah yang terjadi baik di lingkup perguruan tinggi maupun diluar perguruan tinggi. Ada beberapa cara mendapatkan topik tugas akhir, akan tetapi mencari topik tugas akhir bukanlah hal yang mudah, hal ini terbukti dengan banyak mahasiswa yang kesulitan untuk memulai mengerjakan tugas akhir karena belum mendapatkan topik penelitian yang sesuai.

Dalam pencarian topik penelitian atau judul skripsi beberapa mahasiswa sering kali mengalami kesulitan dan kendala sehingga beberapa kali harus mengganti dan merubah judul skripsi. Pengajuan judul skripsi oleh mahasiswa terkadang diterima ataupun ditolak berdasarkan pertimbangan yang salah satunya adalah judul skripsi sudah ada dan kendala utamanya adalah menghindari indikasi adanya kesamaan judul dan adanya indikasi plagiat. Adanya kesamaan judul skripsi antara mahasiswa satu dengan lain tanpa dilakuan pengecekan dapat terjadi *redundancy* judul yang tentunya dapat dihindari apabila ada kontrol dari dosen atau jurusan yang dapat melakukan pengecekan sebelum judul tersebut di setujui [3]. Salah satu upaya yang dapat dilakukan dalam menentukan judul skripsi ialah dengan melakukan pencocokan *string* (*string matching* atau *pattern matching*) dalam teks yang terdapat pada judul skripsi yang diusulkan.

Pencocokan *string (string matching)* adalah pencarian suatu kata di dalam dokumen misalnya menu *Find* di dalam *Microsoft Word* [1]. Pencocokan *string* diperlukan sebagai media atau acuan bagi mahasiswa dalam mengembangkan wawasan dan mengetahui judul skripsi atau topic penelitian yang akan ditentukan dan dibuat oleh mahasiswa. Pada peneliti sebelumnya menggunakan algoritma *Booyer Moore* dalam melakukan pencocokan *string* dan hasilnya kurang optimal untuk digunakan dalam mengambil suatu informasi [2]. Oleh karena itu, pada penelitian ini peneliti mencoba menggunakan Algoritma *Knuth-Morris-Pratt* (KMP) untuk menganalisis bagaimana proses pencocokan *string* yang dihasilkan dan membandingkan sejauh mana nilai kemiripan dari beberapa judul yang sama dan serupa sehingga dapat memberikan suatu informasi yang efektif bagi mahasiswa.

Berdasarkan latar belakang diatas maka penulis ingin membahas atau melakukan penelitian tentang **Analisis *String Matching* Pada Judul Skripsi dengan Algoritma *Knuth-Morris-Pratt* (KMP)**.

ISSN print 2087-1716
ISSN online 2548-7779

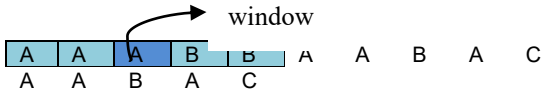
2. Metode

2.1 Skripsi

Skripsi adalah istilah yang digunakan di Indonesia untuk mengilustrasikan suatu karya tulis ilmiah berupa paparan tulisan hasil penelitian sarjana (S1) yang membahas suatu permasalahan/fenomena dalam bidang ilmu tertentu dengan menggunakan kaidah-kaidah yang berlaku. Skripsi bertujuan agar mahasiswa mampu menyusun dan menulis suatu karya ilmiah sesuai dengan bidang ilmunya. Skripsi merupakan persyaratan untuk mendapatkan status sarjana (S1) di setiap Perguruan Tinggi Negeri maupun Swasta yang ada di Indonesia. Istilah skripsi sebagai tugas akhir sarjana hanya digunakan di Indonesia. Di negara lain, seperti Australia menggunakan istilah thesis untuk penyebutan tugas akhir dengan riset untuk jenjang *undergraduate* (S1), *postgraduate* (S2), Ph.D dengan riset (S3) dan *dissertation* [2].

2.2 Dasar Teori *String Searching*

String adalah urutan dari karakter, yang mana karakter ini dapat terdiri dari beberapa *alphabet*. Misalnya adalah *string biner* yang terdiri dari dua *alphabet*, yaitu 0 dan 1, jadi string biner merupakan suatu urutan karakter 0 maupun 1. Contoh yang lain adalah *string* ASCII yang terdiri dari 256 *alphabet*. String searching pada dasarnya adalah mencari pola P yang memiliki panjang m dalam suatu teks T yang memiliki panjang n. Baik pola maupun teks dinyatakan dalam array, pola dinyatakan dengan P[0..m-1] dan teks dinyatakan dengan T[0..n-1]. Kecocokan adalah apabila karakter pada teks dan karakter pada pola yang dibandingkan adalah sama, sedangkan ketidakcocokan adalah sebaliknya.



Ket : Biru Muda menunjukkan kecocokan
Biru Tua menunjukkan ketidakcocokan

2.3 Algoritma *String Matching*

Algoritma *string matching* dalam bahasa Indonesia dikenal dengan istilah algoritma pencocokan *string* [6]. Persoalan pencarian *string* dirumuskan sebagai berikut:

Diberikan:

1. Sebuah teks (*text*), yaitu sebuah (*long*) *string* yang panjangnya n karakter.
2. *Pattern,* yaitu sebuah *string* dengan panjang m karakter (m < n) yang akan dicari dalam teks.

Carilah lokasi pertama di dalam teks yang bersesuaian dengan *pattern*. sebuah kata dalam dokumen misalnya menu *Find* dalam Microsoft Word.

Contoh :

Pattern : not

Teks : nobody **no**ticed him

target

2.4 Algoritma *Knuth-Morris-Pratt* (KMP)

Algoritma *Knuth-Morris-Pratt* (KMP) adalah salah satu algoritma pencarian string, dikembangkan secara terpisah oleh Donald E. Knuth pada tahun 1967 dan James H. Morris bersama Vaughan R. Pratt pada tahun 1966, namun keduanya mempublikasikannya secara bersamaan pada tahun 1977 [1]. Jika kita melihat algoritma *brute force* lebih mendalam, kita mengetahui bahwa dengan mengingat beberapa perbandingan yang dilakukan sebelumnya kita dapat meningkatkan besar pergeseran yang dilakukan. Hal ini akan menghemat perbandingan, yang selanjutnya akan meningkatkan kecepatan pencarian.

Perhitungan pergeseran pada algoritma ini adalah sebagai berikut, bila terjadi ketidakcocokan pada saat *pattern* sejajar dengan teks [*i..i* + *n*–1], kita bisa menganggap ketidakcocokan pertama terjadi di antara teks [*i* + *j*] dan pattern [j], dengan 0 < *j*<*n*. Berarti, teks [*i..i* + *j*–1] = *pattern* [0..*j*–1] dan a = teks [*i* +*j*] tidak sama dengan *b* = *pattern*[*j*]. Ketika kita menggeser, sangat beralasan bila ada sebuah awalan *v* dari pattern akan sama dengan sebagian akhiran *u* dari sebagian teks. Sehingga kita bisa menggeser pattern agar awalan *v* tersebut sejajar dengan akhiran dari *u* [1].

3. Hasil dan Pembahasan

3.1 Analisis algoritma *Knuth-Morris Pratt* (KMP)

ISSN print 2087-1716
ISSN online 2548-7779

Pada algoritma KMP, kita memelihara informasi yang digunakan untuk melakukan jumlah pergeseran. Algoritma menggunakan informasi tersebut untuk membuat pergeseran yang lebih jauh, tidak hanya satu karakter [11]. Pada Algoritma *Knuth-Morris Pratt* (KMP) informasi ketidakcocokan *pattern* dengan teks disimpan untuk menentukan jumlah pergeseran. Algoritma KMP melakukan pergeseran lebih jauh sesuai dengan informasi yang disimpan yang menyebabkan waktu pencarian dapat dikurangi secara signifikan. Selain itu, dari penyimpanan informasi pada pencocokan *string* yang dilakukan algoritma *Knuth-Morris Pratt* dapat diambil keuntungan lain yaitu dari informasi yang disimpan dapat digunakan untuk menentukan berapa persen kedekatan *string* yang akan dicocokkan dengan *pattern* yang ada.

Secara sistematis, langkah-langkah yang dilakukan algoritma *Knuth-Morris-Pratt* (KMP) pada saat mencocokkan *string* :

1. Algoritma *Knuth-Morris-Pratt* (KMP) mulai mencocokkan *pattern* pada awal teks.
2. Dari kiri ke kanan, algoritma ini akan mencocokkan karakter per karakter *pattern* dengan karakter di teks yang bersesuaian, sampai salah satu kondisi berikut dipenuhi :
   a. Karakter di *pattern* dan di teks yang dibandingkan tidak cocok (*mismatch*).
   b. Semua karakter di *pattern* cocok. Kemudian algoritma akan memberitahukan penemuan di posisi ini.
3. Algoritma kemudian menggeser *pattern* berdasarkan tabel *next*, lalu mengulangi langkah 2 sampai *pattern* berada di ujung teks.

3.2 Pembahasan

Penerapan algoritma *Knuth Morris Pratt* pada judul skripsi mahasiswa terdiri dari beberapa langkah sebagai berikut :

String S :

A	A	S	L	P	I	A	S	T	E	R	M	Q

Pattern P **:**

S	I	S	T	E	M

Langkah 1: Bandingkan *pattern* [1] dengan *string* [1]

A	A	S	L	P	I	A	S	T	E	R	M	Q

S	I	S	T	E	M

pattern [1] tidak cocok dengan *string* [1] maka *pattern* akan bergeser satu posisi ke kanan.

Langkah 2: Bandingkan *pattern* [1] dengan *string* [2]

A	A	S	L	P	I	A	S	T	E	R	M	Q

S	I	S	T	E	M

pattern [1] tidak cocok dengan *string* [2] maka *pattern* akan bergeser satu posisi ke kanan.

Langkah 3: Bandingkan *pattern* [1] dengan *string* [3]

A	A	S	L	P	I	A	S	T	E	R	M	Q

S	I	S	T	E	M

pattern [1] cocok dengan *string* [3]. Karena ada kecocokan, maka algoritma *Knuth Morris Pratt* (KMP) akan menyimpan informasi ini, dan *pattern* tidak akan melakukan pergeseran dan melanjutkan pencocokan *pattern* [2] dengan *string* [4].

Langkah 4: Bandingkan *pattern* [2] dengan *string* [4]

A	A	S	L	P	I	A	S	T	E	R	M	Q

S	I	S	T	E	M

pattern [2] tidak cocok dengan *string* [4] maka *pattern* akan bergeser satu posisi ke kanan.

Langkah 5: Bandingkan *pattern* [2] dengan *string* [5]

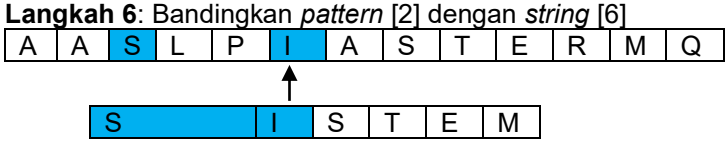
A	A	S	L	P	I	A	S	T	E	R	M	Q

S	I	S	T	E	M

ISSN print 2087-1716
ISSN online 2548-7779

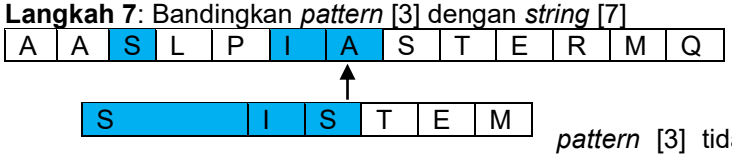
pattern [2] tidak cocok dengan *string* [5] maka *pattern* akan bergeser satu posisi ke kanan.

Langkah 6: Bandingkan *pattern* [2] dengan *string* [6]



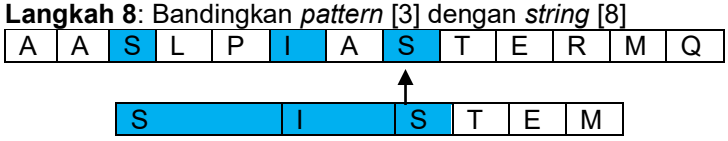
pattern [2] cocok dengan *string* [6]. Karena ada kecocokan, maka algoritma *Knuth Morris Pratt* (KMP) akan menyimpan informasi ini, dan *pattern* tidak akan melakukan pergeseran dan melanjutkan pencocokan *pattern* [3] dengan *string* [7].

Langkah 7: Bandingkan *pattern* [3] dengan *string* [7]



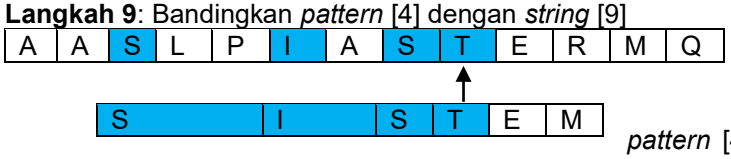
pattern [3] tidak cocok dengan *string* [7] maka *pattern* akan bergeser satu posisi ke kanan.

Langkah 8: Bandingkan *pattern* [3] dengan *string* [8]



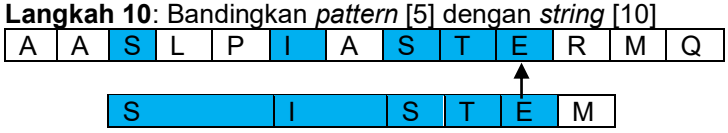
pattern [3] cocok dengan *string* [8]. Karena ada kecocokan, maka algoritma *Knuth Morris Pratt* (KMP) akan menyimpan informasi ini, dan *pattern* tidak akan melakukan pergeseran dan melanjutkan pencocokan *pattern* [4] dengan *string* [9].

Langkah 9: Bandingkan *pattern* [4] dengan *string* [9]



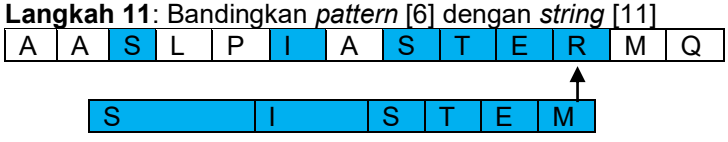
pattern [4] cocok dengan *string* [9]. Karena ada kecocokan, maka algoritma *Knuth Morris Pratt* (KMP) akan menyimpan informasi ini, dan *pattern* tidak akan melakukan pergeseran dan melanjutkan pencocokan *pattern* [5] dengan *string* [10].

Langkah 10: Bandingkan *pattern* [5] dengan *string* [10]



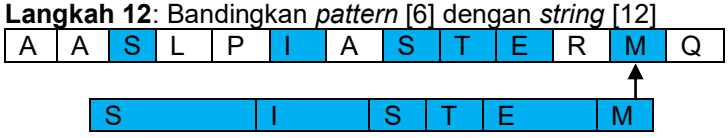
pattern [5] cocok dengan *string* [10]. Karena ada kecocokan, maka algoritma *Knuth Morris Pratt* (KMP) akan menyimpan informasi ini, dan *pattern* tidak akan melakukan pergeseran dan melanjutkan pencocokan *pattern* [6] dengan *string* [11].

Langkah 11: Bandingkan *pattern* [6] dengan *string* [11]



pattern [6] tidak cocok dengan *string* [11] maka *pattern* akan bergeser satu posisi ke kanan.

Langkah 12: Bandingkan *pattern* [6] dengan *string* [12]

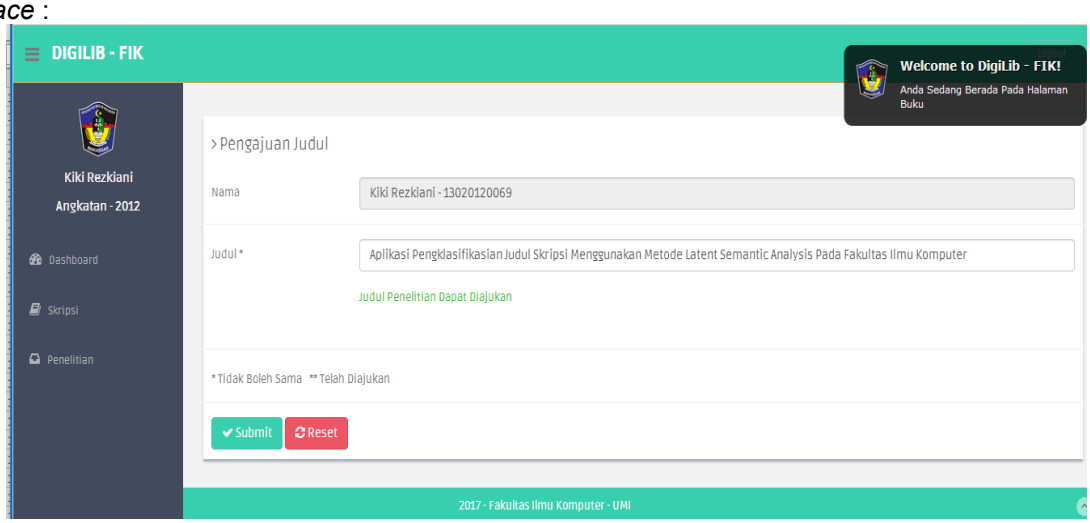


ISSN print 2087-1716
ISSN online 2548-7779

pattern [6] cocok dengan *string* [12]. Karena ada kecocokan, maka algoritma *Knuth Morris Pratt* (KMP) akan menyimpan informasi ini, dan *pattern* tidak akan melakukan pergeseran dan melanjutkan pencocokan *pattern* [7] dengan *string* [13]. Namun karena jumlah *pattern* hanya 6 huruf maka pencarian akan dihentikan dan diperoleh hasil bahwa *pattern* P terdapat kecocokan dengan string S sebesar 100%.

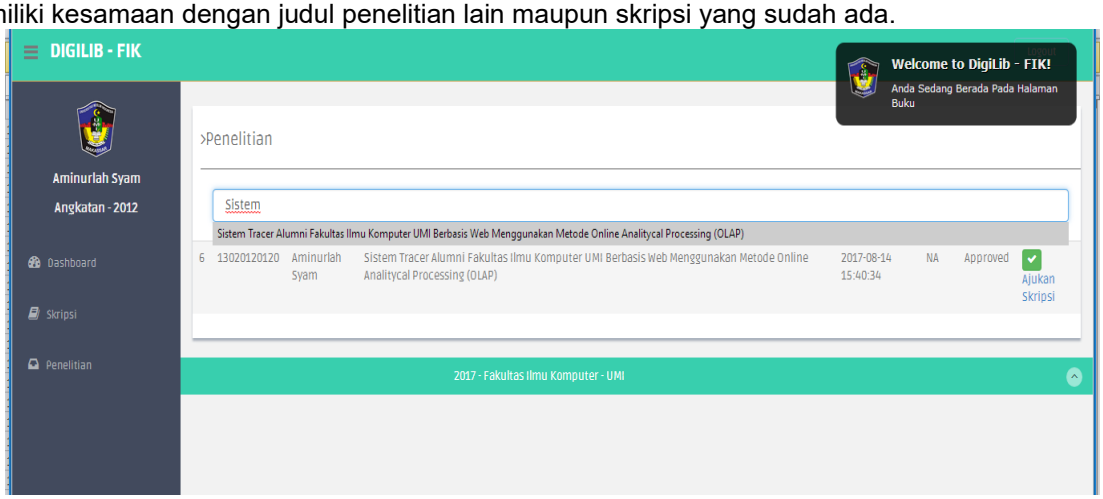
3.3 Implementasi dan Pengujian Sistem

Tahap ini dilakukan pengujian sistem menggunakan metode *black box*. Pada pengujian *black box* tidak memperdulikan mekanisme internal pada sebuah sistem dan hanya berfokus pada keluaran yang dihasilkan sebagai respon dari pelaksanaan sebuah kondisi yang diinginkan. Berikut hasil *user interface* :



Gambar 1. Halaman Pengajuan Judul Penelitian

Gambar 1 menunjukkan informasi pengajuan judul penelitian yang diajukan oleh mahasiswa dan pada halaman ini pula mahasiswa mendapatkan informasi bahwa judul penelitian yang diajukan tidak memiliki kesamaan dengan judul penelitian lain maupun skripsi yang sudah ada.



Gambar 2 menunjukkan informasi judul penelitian yang sudah ada sebelumnya sehingga mahasiswa yang akan mengajukan judul penelitian tidak dapat mengusulkan judul penelitian yang sama kecuali judul penelitian yang diajukan mahasiswa berbeda objek dan metodenya dengan judul penelitian sebelumnya.

4. Kesimpulan dan Saran

4.1 Kesimpulan

Berdasarkan hasil penelitian yang dilakukan maka dapat diambil kesimpulan bahwa hasil analisis algoritma *knuth-morris pratt* (KMP) dapat diketahui kualitas pencocokan string pada judul skripsi atau topik penelitian yang diajukan mahasiswa sehingga setiap mahasiswa bisa mengetahui judul-judul skripsi yang sudah ada dan yang belum ada pada sistem yang dibangun.

ISSN print 2087-1716
ISSN online 2548-7779

4.2 Saran

Adapun saran pada penelitian ini sebagai berikut :

1. Pada penelitian ini perlu adanya pengembangan lebih lanjut oleh peneliti lain terkait pengembangan perangkat lunak.
2. Penelitian ini dapat diterapkan metode dan algoritma lain untuk pengembangan analisis.

Daftar Pustaka

[1] Ekaputri Gahayu Handari, dan Yulie Anneria Sinaga, 2006. **Aplikasi Algoritma Pencarian String Knuth-Morris-Pratt dalam Permainan Word Search.** Departemen Teknik Informatika, Institut Teknologi Bandung, Bandung

[2] Ginting Guidio Leonaerde, 2014. **Penereapan Algoritma Boyer Moore Pada Aplikasi Pengajuan Judul Skripsi Berbasis Web**, Seminar Informasi dan Teknologi Ilmiah (INTI 2014).

[3] Heriyanto, 2012. **Pencarian Kemiripan Judul Skripsi dan Abstrak dengan Metode Exact Match (Studi Kasus Program Studi Teknik Informatika UPN "Veteran" Yogyakarta)**, Seminar Nasional Informatika 2012 (semnasIF 2012), Yogyakarta.

[4] Kusrini, 2006. **Sistem Pakar Teori dan Aplikasi**, Penerbit ANDI Yogyakarta hal : 1.

[5] Lestari Sri, Djaya Amin, 2011, **Aplikasi Search Engine Menggunakan Algoritma Knuth-Morris-Pratt (Kmp)**, Prosiding Seminar Nasional Manajemen Teknologi XIII.

[6] Munir, Rinaldi, 2007. **Diktat Kuliah IF2251 *Strategi Algoritmik***. Institut Teknologi Bandung.

[7] Pedoman Penulisan Skripsi Universitas Negeri Surabaya 2014

[8] Pressman, Roger S. 2002. **Rekayasa Perangkat Lunak : Pendekatan Praktisi (Buku 1).** Yogyakarta : Andi

[9] Rama Aulia, 2008, **Analisa Algoritma Knuth-Morris-Pratt dan Algoritma Boyer Moore dalam Proses Pencarian String**.

[10] Sianipar. R.H, Mangiri. H.S, I.K, 2013, **Matlab untuk Pemrosesan Citra Digital**, Informatika, Bandung.

[11] Sunni, I, 2010. ***Music Finder* Menggunakan Algoritma *KMP Extension*.** Diambil 01 Desember 2012 *dari* http://www.in*Form*atika.org/~rinaldi/Stmik/20102011/Makalah2010/MakalahStima2010-096.pdf

[12] Whitten, J. L.& L. D. Bentley. 2007. ***Systems Analysis and Design Methods***. New York : McGrawHill.

