JURNAL KOMPUTASI, Vol.14, No.1, January 2017, pp. 115~124
ISSN: 1693-7554

Implementasi Algoritma *Knuth-Morris-Pratt* Pada Fungsi Pencarian Judul Tugas Akhir *Repository*

Halimah Tus Sa'diah
Manajemen Informatika, Universitas Pakuan
Jl. Ciheuleut Pakuan PO Box 452 Bogor 16143 Jawa Barat Indonesia
e-mail: ha5.tussadiah47@gmail.com

Abstrak

Pencarian pada sistem repository belum menggunakan algoritma pencarian sehingga hasil pencarian dari sistem tersebut belum optimal. Oleh karena itu, perlu adanya implementasi algorima pencarian yang akan membantu menghasilkan hasil pencarian yang cepat dan optimal. Paper ini bertujuan untuk mengimplementasi Algoritma KMP pada fungsi pencarian dalam sistem repository tugas akhir. Metode penelitian yang akan digunakan dalam penelitian ini yaitu tahap System Development Life Cycle yang terdiri atas analisis, rancangan sistem pencarian, implementasi algoritma KMP dan pengujian. Algoritma KMP berhasil diimplementasikan pada fungsi pencarian aplikasi repository tugas akhir. Hasil pengujian performa menunjukan bahwa rata-rata performa algoritma KMP dalam menemukan kata di form pencarian adalah 0.0138 detik. Hal ini menunjukan bahwa algoritma KMP sudah cukup cepat dan optimal dalam fungsi pencarian pada aplikasi repository tugas akhir Diploma Manajemen Informatika

Kata kunci: *Algoritma KMP, Algoritma Pencarian, Fungsi Pencarian, Knuth-Moriss-Pratt, Repository*

1. Pendahuluan

Seiring dengan pesatnya pertumbuhan teknologi informasi dalam bentuk format digital menyebabkan sejumlah arsip atau dokumen pada suatu instansi, dikonversi dalam bentuk digitalisasi. Bentuk digitalisasi tersebut akan disimpan dalam sistem repository. *Repository* merupakan ruang fisik yang digunakan untuk penyimpanan dokumen atau arsip [1]. Dokumen yang digitalisasi biasanya dokumen yang sangat penting dan mempunyai nilai guna yang tinggi. Salah satu dokumen yang penting untuk digitalisasi adalah dokumen tugas akhir mahasiswa.

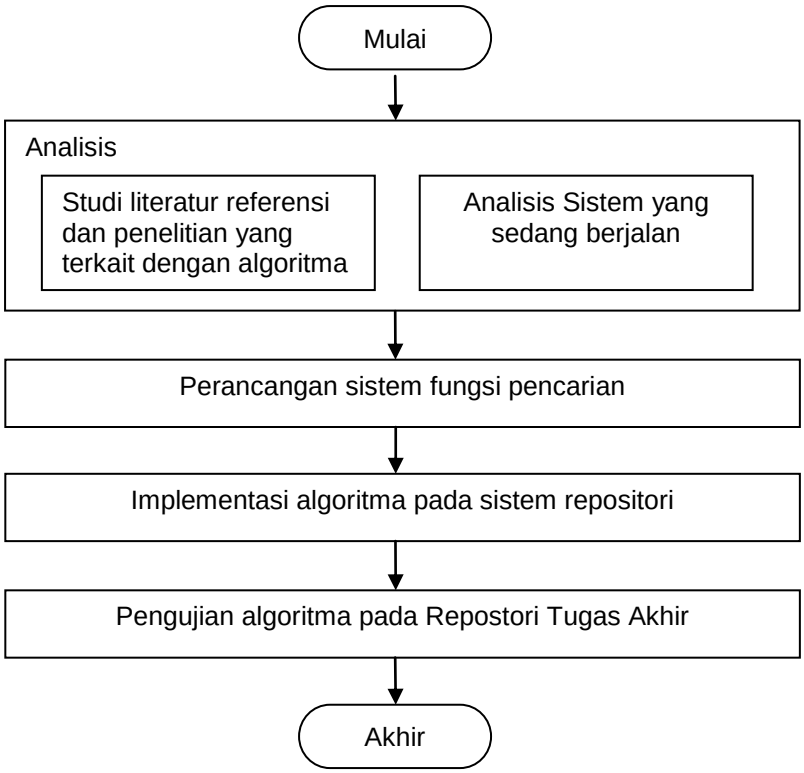
Dokumen tugas akhir mahasiswa Diploma manajemen informatika mulai akhir tahun 2015 akan digitalisasi pada sistem repository tugas akhir. Tujuan pendigitalisasi tugas akhir, yaitu agar mahasiswa mudah dalam mencari dokumen tugas akhir, dan pengelolaan dokumen tugas akhir menjadi lebih mudah bagi jurusan. Sistem Repository tugas akhir sudah dibangun pada akhir 2015, [2] namun masih perlu penyempurnaan salah satunya pada bagian pencarian.

Pencarian pada sistem *repository* belum menggunakan algoritma pencarian sehingga hasil pencarian dari sistem tersebut belum optimal. Oleh karena itu, perlu adanya implementasi algorima pencarian yang akan membantu menghasilkan hasil pencarian yang tepat dan optimal. Salah satu algoritma pencarian yang dapat digunakan adalah algoritma *Knuth-Morris-Pratt* (KMP). Algoritma KMP merupakan algoritma pencarian hasil pengembangan dari algoritma pencarian *Brute Force*. Algoritma ini memiliki keunggulan pencarian kecocokan pada file yang berukuran besar. Algoritma KMP mencari teks berdasarkan urutan dari kiri ke kanan pada awal teks dan kemudian menggeser susunan kata sampai pada ujung teks [3]. Penelitian mengenai implementasi algoritma algoritma *KMP* sudah pernah dilakukan oleh Rossaria *et al* (2015), yaitu implementasi algoritma KMP dalam aplikasi dokumen berbasis android [3]. Selain itu, Zaky (2015) mengimplementasikan algoritma KMP dalam perancangan *game* HANACARAKA [4]. *Paper* ini bertujuan untuk mengimplementasi Algoritma KMP pada fungsi pencarian dalam sistem *repository* tugas akhir Diploma Manajemen Informatika.

JURNAL KOMPUTASI, Vol.14, No.1, January 2017, pp. 115~124
ISSN: 1693-7554

2. Metode Penelitian

Tahap penelitian terdiri dari beberapa tahap (Gambar 1), yaitu analisis, perancangan sistem fungsi pencarian, implementasi algoritma pada sistem repositori, dan pengujian algoritma. Analisis yang dilakukan, yaitu studi literatur dengan melakukan pengumpulan berbagai referensi yang relevan mengenai algoritma pencarian KMP. Selain itu, pada tahap ini dilakukan analisis sistem yang sedang berjalan, khususnya pada fungsi pencarian. Tahap selanjutnya setelah analisis adalah merancang sistem pencarian dengan menggambar diagram alur fungsi pencarian pada sistem *repository* berdasarkan algortitma KMP. Pada tahap Implementasi algortima dilakukan implementasi algoritma dengan memodifikasi kueri PHP MYSQL fungsi pencarian yang terdapat pada sistem *repository* Diploma Manajemen Informatika. Setelah tahap implementasi, dilakukan pengujian algoritma dengan cara membaca posisi string yang ditemukan berdasarkan kata yang dicari dan pengujian performa algoritma KMP pada fungsi pencarian aplikasi *repository* tugas akhir.



Gambar 1. Tahapan Penelitian

3. Analisis dan Hasil

3.1. Algoritma KMP dan *Repository* Tugas Akhir

3.1.1 Penelitian Terkait

Beberapa penelitian terkait mengenai algoritma KMP ditunjukan pada Tabel 1. Pada Tabel 1, dapat dilihat bahwa sistem repositori belum menggunakan algoritma untuk mendukung optimalnya hasil pencarian sehingga pada penelitian ini diimplementasikan algoritma KMP. Algoritma KMP sudah diimplementasikan pada program Android untuk mendukung pengoptimalan pencarian kata (Tabel 1). Algoritma tersebut pada penelitian ini akan diimplementasikan pada sistem berbasis web.

JURNAL KOMPUTASI, Vol.14, No.1, January 2017, pp. 115~124
ISSN: 1693-7554

Tabel 1. Penelitian Terkait

Penulis (Tahun)	Algoritma	Basis Media	Keterangan
Zuraiyah *et al* (2015)	-	Web	Sistem Repository pada fungsi pencarian belum menggunakan algoritma
Rossaria *et al* (2015)	KMP	Android	Implementasi algoritma KMP pada android untuk pencarian kata
Zaky (2015)	KMP	Andoid	Implementasi algoritma KMP pada android untuk Game

3.1.2 Algoritma KMP

Algoritma KMP dikembangkan oleh D.E. Knuth, J.H.Morris, dan V.R. Pratt [3]. Algoritma ini merupakan jenis *extract string matching algorithm* yang melakukan pencocokan string secara tepat sesuai dengan susunan karakter [5]. Berikut merupakan algoritma dari KMP [6] :

```
procedure HitungPinggiran(input m :integer, P : array [1..m] of
char,output b : array [1..m] of integer)
{
menghitung nilai b[1..m] untuk pattern
P[1..m }
Kamus
k,q : integer
Algoritma
b[1] ← 0 ; q ← 2 ; k ← 0;
for q ← 2 to m do
       while ((k>0) and (P[q]≠P[k+1])) do
              k ← b[k]
       endwhile
       if P[q]=P[k+1] then
              k ← k + 1
       endif
b[q] = k
endfor

procedure KMPsearch(input m,n :integer, input P : array [1..m] of
char, input T : array [1..n] of char, output idx : integer){
mencari kecocokan pattern P di dalam teks T dengan algoritma KMP.
Jika ditemukan P di dalam T, maka lokasi awal kecocokan disimpan di
dalam peubah idx
}
Kamus
i,j : integer
ketemu : boolean
b : array[1..m] of integer

procedure HitungPinggiran(input m : integer, P : array [1..m] of
char,output b : array [1..m] of integer)
```

```
Algoritma
HitungPinggiran(m, P, b)
j ← 0 ; i ← 1 ; ketemu ← false;
while (i ≤ n and not ketemu)
do
       while ((j>0) and (P[j+1]≠T[i]))
       do j ← b[j] endwhile
       if P[j+1] = T[i] thenj ← j + 1 endif
       if j = m then ketemu ← true
       elsei ← i + 1
       endif
endwhile
if ketemu then idx ← i - m + 1 else idx ← -999 {tidak ketemu}
```

JURNAL KOMPUTASI, Vol.14, No.1, January 2017, pp. 115~124
ISSN: 1693-7554

3.1.3 *Repository* Tugas Akhir

Repository tugas akhir merupakan sistem yang dibangun untuk mendigitalisasi tugas akhir Diploma Manajemen Informatika. Sistem ini dibangun untuk memudahkan mahasiswa dalam pencarian tugas akhir dan mempermudah staff dalam pengelolaan tugas akhir. Selain itu, sistem ini dibangun sebagai wadah dalam dokumentasi bentuk digital tugas akhir D3 Manajemen Informatika [2]. Sistem Repository tugas akhir ditunjukan Gambar 2.

Gambar 2 Sistem *Repository* Tugas Akhir

3.2. Perancangan Alur Fungsi Pencarian pada Sistem *Repository* Tugas Akhir dengan algoritma KMP

Perancangan alur fungsi pencarian algoritma KMP dalam PHP-MYSQL ditunjukan pada Gambar 3.

Implementasi Algoritma KMP pada *Repository* Tugas Akhir

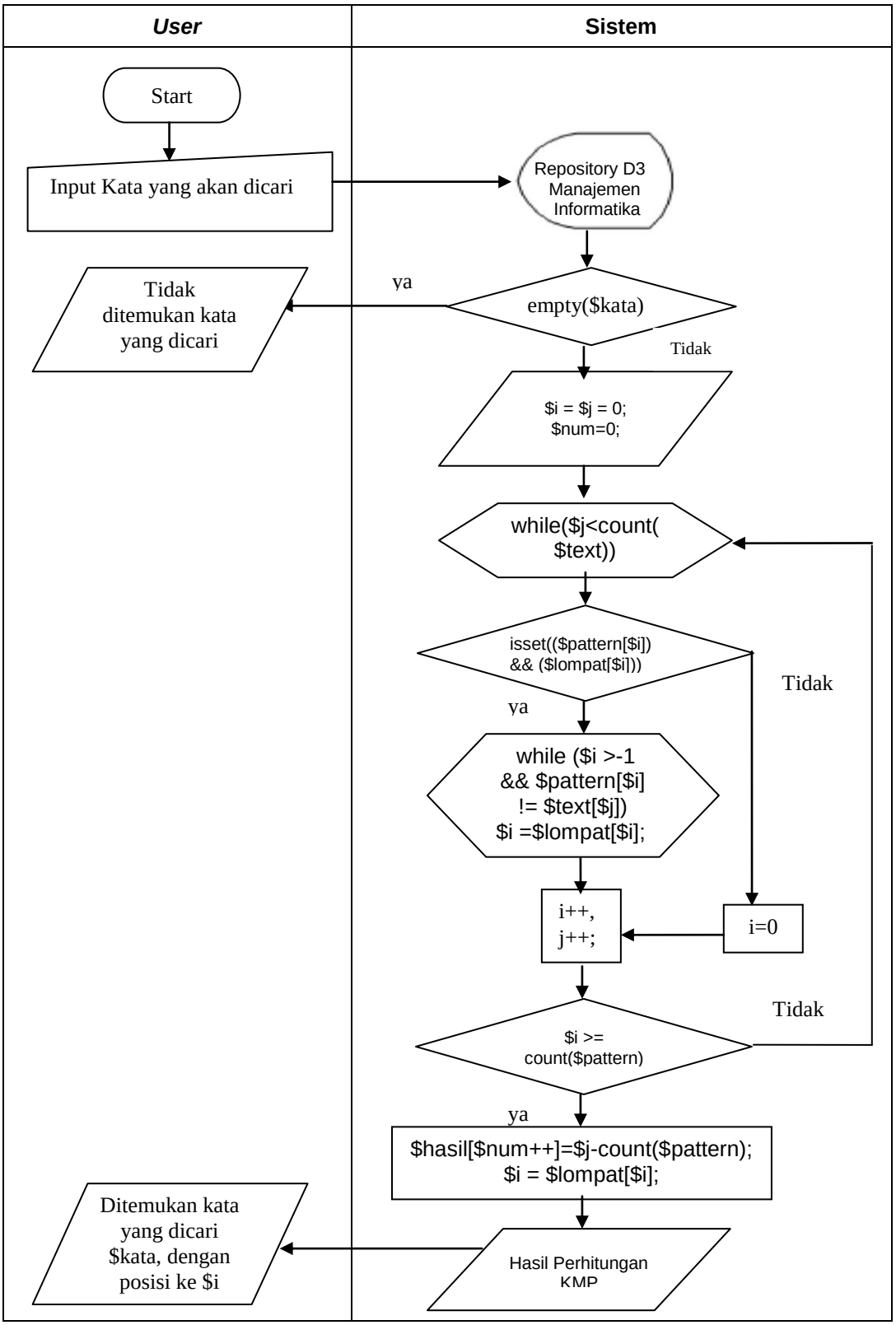
Tahap implementasi, yaitu tahap mengimplementasikan algoritma KMP dalam bentuk PHP dan selanjutnya fungsi algoritma tersebut *included* ke dalam script PHP-MySQL fungsi pencarian. Langkah-langkah implementasi algoritma KMP dengan menggunakan PHP-MYSQL, yaitu

1. Memanggil *script* fungsi koneksi PHP-MYSQL
2. Memanggil *script* fungsi KMP
   Algoritma KMP dikonversi ke dalam bentuk PHP. Alur kerja *script* PHP KMP ditunjukan pada Gambar 3.
3. Mengecek variabel kata.
   Variabel kata merupakan tempat menampung kata yang akan dicari.

JURNAL KOMPUTASI, Vol.14, No.1, January 2017, pp. 115~124
ISSN: 1693-7554

a. Jika variabel kata kosong, maka dicetak output “Anda belum memasukan judul tugas akhir yang ingin dicari” (Gambar 4).
b. Jika variabel kata ada, dan terdapat sejumlah data yang ditemukan sesuai dengan database maka dicetak jumlah judul, dan judulnya (Gambar 5).
c. jika variabel kata ada, tetapi tidak ada yang sesuai dengan data yang terdapat di database maka dicetak output “Tidak ditemukan kata ” (Gambar 6)

JURNAL KOMPUTASI, Vol.14, No.1, January 2017, pp. 115~124
ISSN: 1693-7554



. Gambar 3 Alur fungsi pencarian KMP dalam PHP

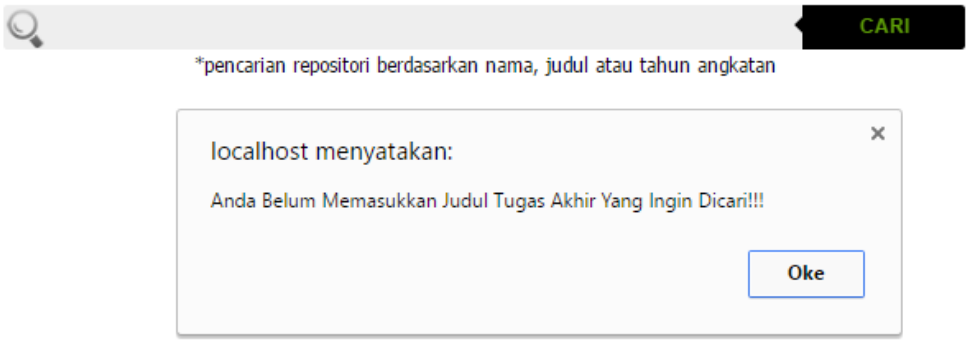
JURNAL KOMPUTASI, Vol.14, No.1, January 2017, pp. 115~124
ISSN: 1693-7554

3.3

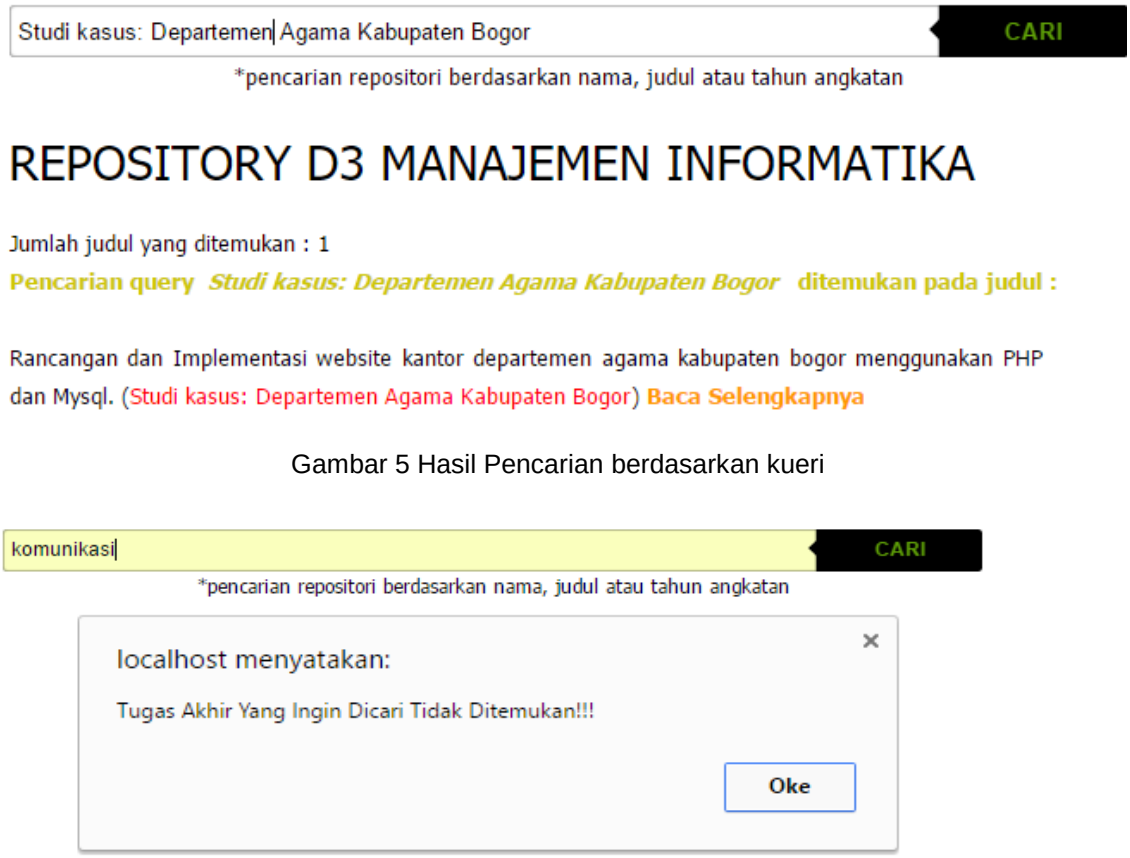
Script PHP-MYSQL fungsi pencarian algoritma KMP

```
<?php
include "koneksi.php";
include_once("kmp.php");
$kata = '';
if(isset($_GET['kata'])) $kata = $_GET['kata'];

echo " <form class='form-wrapper cf' action='' method='post'>
              <input type='text'  name='kata' value=$kata >
              <button type='submit' name='cari'><b>Cari</b></button>
      </form>”;

$KMP = new KMP();
if(isset($_POST['cari'])){
       $kata = $_POST['kata'];
       $sql="SELECT * FROM repositori_mhs where judul LIKE '%$kata%'";
       $result = mysql_query($sql);
       $jumlah = mysql_num_rows($result);
       if (empty($kata)){
              echo "<script language='javascript'>
              alert('Anda Belum Memasukkan Judul Tugas Akhir Yang Ingin
Dicari!!!');
              document.location='index.php'</script>";
                     }
       else if ($jumlah>0){
              echo "<h2>REPOSITORY D3 MANAJEMEN INFORMATIKA</h2>";
              echo "Jumlah judul yang ditemukan : ".$jumlah."<br>";
              echo "<b><font color='#cec708'>Pencarian query
 <i>$kata</i>
                 ditemukan pada judul : </font></b><br><br>";

       while ($teks = mysql_fetch_array($result)){
              $hasil = $KMP->KMPSearch($kata,$teks['judul']);

              echo "<div style='width:600px;'>";

              echo nl2br(str_replace($kata,"<font
              color='red'>".$kata."</font>",$teks['judul']));
              echo " <font='blue'><a
              href='repositori_lengkap.php?NPM=$teks[npm]'>Baca
              Selengkapnya</a></font>";
              echo "<br><br/>";
              echo "</div>";
       }
```

JURNAL KOMPUTASI, Vol.14, No.1, January 2017, pp. 115~124
ISSN: 1693-7554



Gambar 4 Notifikasi jika belum memasukan kata pencarian



Gambar 5 Hasil Pencarian berdasarkan kueri

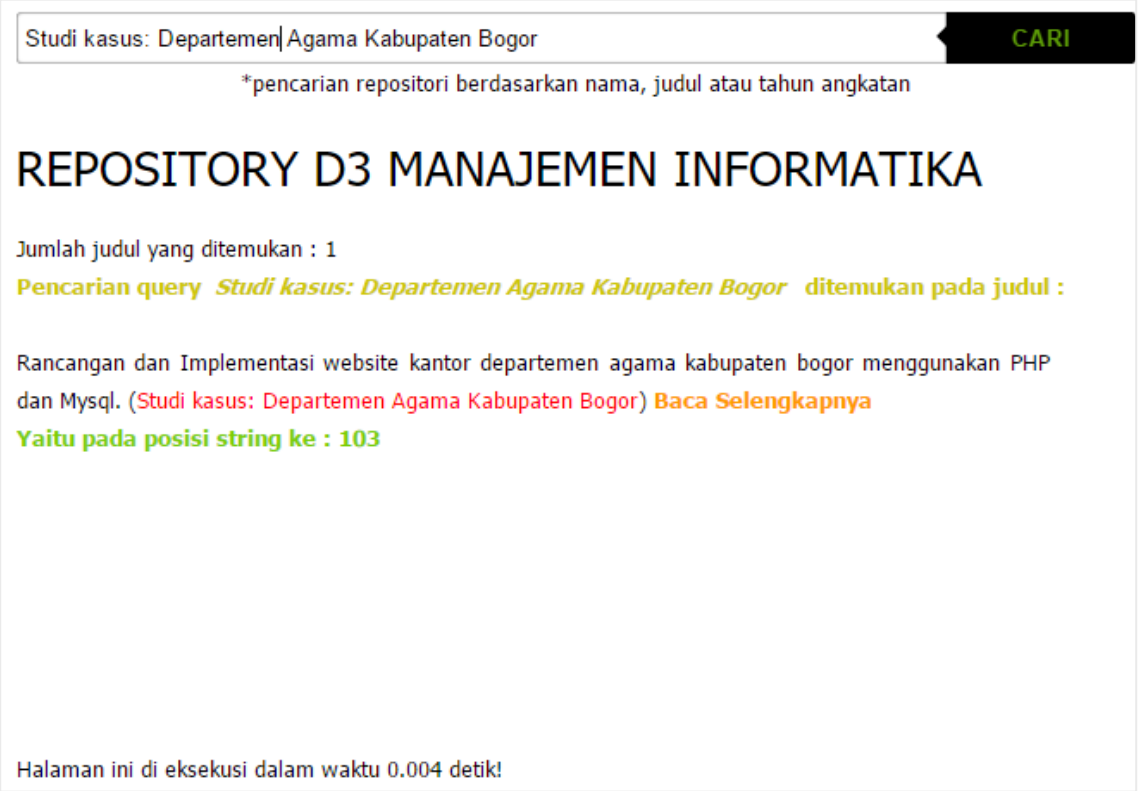
Gambar 6 Notifikasi jika kata yang dicari tidak ditemukan

3.4 Pengujian Algoritma KMP

Pengujian dilakukan dengan menguji posisi string saat penemuan hasil pencarian dan menguji performa pencarian algoritma KMP . Hasil pencarian dengan kueri “Studi kasus departemen agama kabupaten bogor ” dihasilkan 1 judul pada *repository* tugas akhir dengan waktu eksekusi 0.004 detik (Gambar 7). Pada hasil pencarian judul terdapat keterangan posisi kata web yang ditemukan, yaitu pada posisi ke 103 dengan judul tugas akhir “Rancangan dan Implementasi Website kantor departemen agama kabupaten bogor menggunakan PHP MYSQL” (Gambar 7). Pengujian lainnya yang dilakukan

JURNAL KOMPUTASI, Vol.14, No.1, January 2017, pp. 115~124
ISSN: 1693-7554

adalah pengujian performa eksekusi fungsi pencarian menggunakan algoritma KMP. Tabel 1 merupakan hasil pengujian performa algoritma KMP pada fungsi pencarian *repository* tugas akhir.



Gambar 7 Posisi String pada Pengujian KMP

Tabel 2 Performa Eksekusi Algoritma KMP pada fungsi pencarian

No	Kata yang dicari	Jumlah judul yang ditemukan	Waktu Eksekusi Fungsi Pencarian dengan Algoritma KMP (detik)
1	Web	44	0.028
2	Algoritma	13	0.011
3	Mobile	12	0.005
4	PHP	14	0.007
5	Mikrocontroler	10	0.008
6	Jaringan	6	0.005
7	Server	32	0.017
8	Visual	35	0.025
9	Aplikasi	37	0.021
10	Multimedia	25	0.011
		Jumlah	0.138 detik
		Rata-rata	0.0138 detik

JURNAL KOMPUTASI, Vol.14, No.1, January 2017, pp. 115~124
ISSN: 1693-7554

Pengujian performa algoritma KMP diuji dengan cara memasukan input kata yang ingin dicari ke dalam *form* pencarian *repository* tugas akhir. Sistem akan menghitung waktu eksekusi algoritma KMP dalam menghasilkan hasil pencarian (Gambar 7). Data yang diuji berjumlah 100 data. Berdasarkan Tabel 2, waktu eksekusi algoritma KMP yang paling cepat sebesar 0.005 detik, yaitu pada pencarian kata '*mobile*'. Adapun waktu eksekusi algoritma KMP yang paling lambat sebesar 0.028 detik, yaitu pada pencarian kata 'web. Rata-rata performa algoritma KMP pada sistem *repository* tugas akhir berdasarkan 100 data adalah 0.0138 detik. Hal ini menunjukan bahwa algoritma KMP sudah cukup cepat dan optimal dalam mengeksekusi kueri pada fungsi pencarian *repository* tugas akhir. Algoritma KMP membutuhkan waktu O(m) untuk mengitung fungsi pinggiran pada fungsi pencarian. Adapun proses pencarian teks algoritma KMP pada fungsi pencarian membutuhkan waktu O(n), sehingga kompleksitas algortima KMP sebesar O(m+n) [8].

4. Kesimpulan

Pencarian pada sistem *repository* tugas akhir belum menggunakan algoritma pencarian sehingga dibutuhkan implementasi algoritma. Algoritma pencarian yang diimplementasikan, yaitu algoritma Knuth-Morris-Pratt. Algoritma KMP berhasil diimplementasikan pada fungsi pencarian aplikasi *repository* tugas akhir. Hasil pengujian performa menunjukan bahwa rata-rata performa algoritma KMP dalam menemukan kata di *form* pencarian adalah 0.0138 detik. Hal ini menunjukan bahwa algoritma KMP sudah cukup cepat dan optimal dalam menemukan hasil pencarian kata pada aplikasi *repository* tugas akhir.

5. Saran

Untuk penyempurnaan penelitian, sistem pada hasil pencarian perlu ditambahkan konsep *information retrievel* dan Algoritma *Levenshtein Distance* untuk fitur *autocomplete* dan *autocorrect* dalam fungsi pencarian.

JURNAL KOMPUTASI, Vol.14, No.1, January 2017, pp. 115~124
ISSN: 1693-7554

References

[1] Reitz,J.M. *Dictionary for Library and Information Science*. Westport, Connecticut: Libraries Unlimited.2004
[2] Zuraiyah TA, Sadiah HT, Utami DA, Ramadhan NA. *Repository Tugas Akhir (RTA) Online.* Prosiding Seminar Nasional Teknik Informarika dan Komputer. Depok. 2015.
[3] Rossaria M, Susilo B, Ernawati. *Implementasi Algoritma Pencocokan String Knuth-Morriss-Pratt Dalam Aplikasi Pencarian Dokumen Digital Berbasis Android*. 2015. 3 (2) : 183-195.
[4] Zaky,A.M.Implementasi Algoritma Knuth Morris Pratt Pada Perancangan Game Hanacaraka. Skripsi. Universitas Negeri Semarang. 2015.
[5] http://www.cs.cmu.edu/afs/andrew.cmu.edu/course/15/354/www/postscript/kmp.pdf. diakses pada tanggal 15 desember 2016
[6] Ekaputri G.H. Sinaga Y.H. Aplikasi Algoritma Pencarian String Knuth-Morris-Pratt dalam Permainan Word Search. http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2005-2006/Makalah2006/MakalahStmik2006-07.pdf diakses tanggal 10 Desember 2016.
[7] Sulun HS. Penerapan Algoritma Knuth-Morris-Pratt pada Aplikasi Pencarian Berkas di Komputer. http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2006-2007/Makalah_2007/MakalahSTMIK2007-124.pdf. diakses tanggal 12 Desember 2016.
[8] Basee, S. Computer Algorithms : Introduction To Design and Analysis.