Membangun Sistem Repository Berbasis Search Engine Dengan Menggunakan Algoritma Knuth Morris Pratt

Said Mahfud[1], Anwar[2], Muhammad Nasir[3]
Jurusan Teknologi Informasi dan Komputer Politeknik Negeri Lhokseumawe
Jln. Banda Aceh-Medan Km. 280, Buketrata 24301 INDONESIA
saidmahfud02@gmail.com
anwar@pnl.ac.id
masnasir.tmj@gmail.com

Abstrak— **Perkembangan teknologi informasi saat ini sangat pesat, kebutuhan manusia dalam sumber data sangatlah penting. Untuk mengatasi perihal masalah tersebut penulis ingin membangun sistem web server repositoy berbasis search engine, yang merupakan layanan pustaka yang menyediakan beberapa sumber data dan informasi sebagai acuan untuk membuat suatu karya ilmiah. Pencarian pada sistem repository belum menggunakan algoritma pencarian sehingga hasil pencarian dari sistem tersebut belum optimal. Oleh karena itu, perlu adanya implementasi algoritma pencarian yang akan membantu menghasilkan hasil pencarian yang cepat dan optimal. Paper ini bertujuan untuk mengimplementasi Algoritma KMP pada fungsi pencarian dalam sistem repository pada prodi Teknik multimedia dan Jaringan. 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 layanan sistem repository journal dan e-book. Hasil pengujian performa menampilkan bahwa rata-rata performa algoritma KMP dalam menemukan kata di form pencarian adalah 0.0115 detik. Hal ini membuktikan bahwa algoritma KMP sudah cukup cepat dan optimal dalam fungsi pencarian pada layanan sistem repository journal dan e-book.**
Kata Kunci : Web, repository, algoritma KMP, server

Abstract— ***The development of information technology is currently very rapid, human needs in data sources are very important. To overcome this problem the writer wants to build a repositoy web server system based on search engines, which is a library service that provides several sources of data and information as a reference for making a scientific work. Searching on a repository system has not used a search algorithm so that the search results from the system are not optimal. Therefore, it is necessary to implement a search algorithm that will help produce fast and optimal search results. This paper aims to implement the KMP Algorithm in the search function in the repository system in the Multimedia Engineering and Network study programs. The research method that will be used in this study is the System Development Life Cycle stage which consists of analysis, search system design, KMP algorithm implementation and testing. The KMP algorithm is successfully implemented in the journal and e-book system repository service search function. Performance testing results show that the average performance of the KMP algorithm in finding words in the search form is 0.0115 seconds. This proves that the KMP algorithm is fast enough and optimal in the search function on the journal and e-book repository system services.***
Keywords: Web, repository, KMP algorithm, server

I. PENDAHULUAN

Perkembangan teknologi khususnya dibidang teknologi informasi membuat semua pengguna membutuhkan teknologi yang cepat dan dapat diakses dari berbagai tempat dan berbagai sumber daya untuk mendukung segala jenis pekerjaan yang dilakukan sehari hari. Teknologi informasi telah ber-metamorfosis menjadi sebuah basis penting dimana hal hal substansial dari pengguna pribadi didokumentasikan dan disimpan dalam sebuah penyimpanan data [1]. Dalam pelaksanaannya, penyimpanan data beserta aplikasi lainnya seringkali membutuhkan sumberdaya penyimpanan Central Processing Unit (CPU) yang statis atau tidak dapat dibawa kemana –mana, (Aisa, 2014).

Saat ini salah satu masalah yang sudah dapat diatasi adalah seperti masalah penyimpanan file. Proses menyimpan semua file baik foto, film, maupun tugas bahkan laporan pertanggung jawaban kegiatan selalu disimpan dalam media harddisk, flashdisk, ataupun penyimpanan yang lain [2]. Semua orang yang ingin menyalin file pasti harus meminjam harddisk atau flashdisk tersebut lalu dipasangkan dari laptop yang satu ke laptop yang lain dan kembalikan ke pengguna. Hal tersebut tidak sesuai kaidah dan prinsip smart technology for smart life. Untuk itu, dengan adanya teknologi peneliti harus membuat suatu pemanfaatan teknologi baru dengan membuat media penyimpanan sendiri [3].

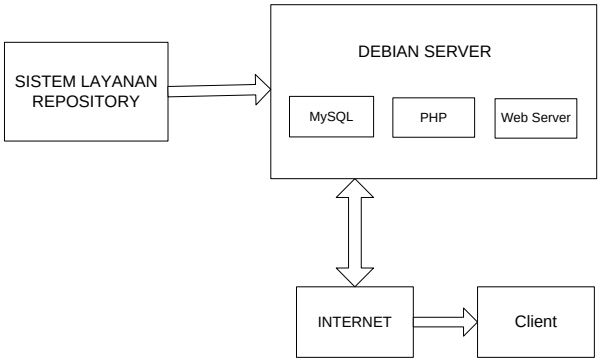
Berdasarkan permasalahan yang telah diuraikan, Maka rumusan masalah yang dapat dirumuskan diantaranya bagaimana merancang sistem untuk layanan *repository*, *b*agaimana menerapkan algoritma *Knuth Morris Pratt* pada *search engine* untuk layanan repository, bagaimana pencocokan string pada pattern secara manual, bagaimana perhitungan kecepatan pencarian string untuk setiap percobaan pada sistem.

Adapun tujuan dari penelitian ini adalah untuk membangun dan mengimplementasikan suatu sistem layanan repository berbasis search engine mahasiswa program studi rekayasa komputer jaringan dengan menggunakan algoritma Knuth Morris Pratt.

II. METODOLOGI PENELITIAN

A. Perancangan sistem

Rancangan sistem merupakan tahap yang dilakukan setelah melalukan analisa perancangan dalam membangun sebuah sistem. Membuat suatu sistem memerlukan persiapan perancangan yang baik dan benar, karena perancangan menyangkut semua elemen yang akan membentuk sebuah sistem.

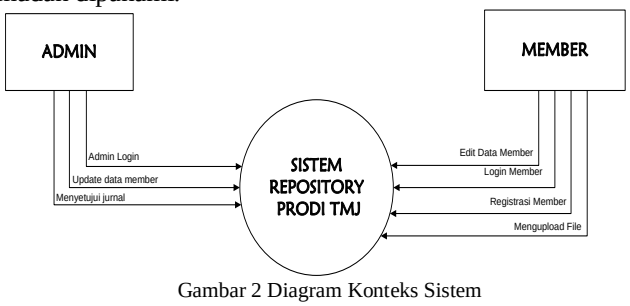


Gambar 1. Block Diagram Sistem Kerja

Berikut ini adalah langkah untuk perancangan sistem, salah satunya perancangan sofware yang mencakup tentang bagaimana membangun sebuah sistem repository sesuai dengan rancangan yang telah dikerjakan. Adapun beberapa hal yang termasuk perancangan software yaitu : pembuatan diagram konteks (Context Diagram) untuk alur pembuatan website, pembuatan DFD (Data Flow Diagram) dan pembuatan ERD (Entity Relationship Diagram) untuk perancangan database. Perancangan sistem ini dilakukan dengan menggunakan bantuan perangkat lunak Microsoft Office Visio 2007.

B. Diagram Konteks (Context Diagram)

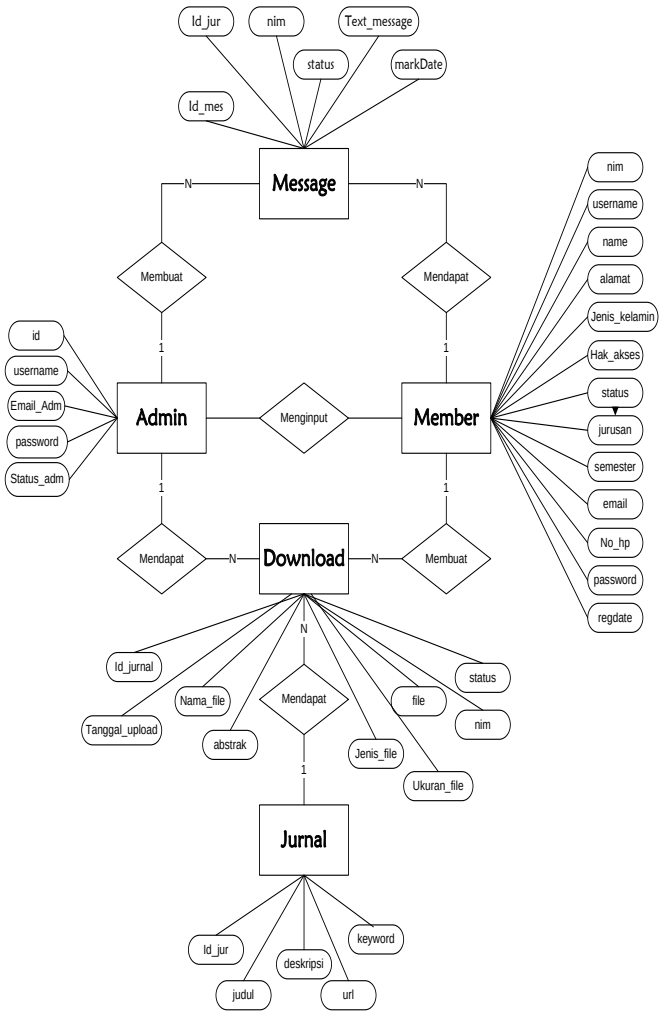
Diagram konteks tedapat entitas-entitas yang menunjukkan pelaku atau aktor sistem repository berbasi search engine menggunakan algoritma Knuth Morris Pratt, serta menggambarkan aliran data apa saja yang dibutuhkan entitas dalam tiap-tiap aktiftas yang dilakukan, agar lebih mudah dipahami.



Gambar 2 Diagram Konteks Sistem

C. Perancangan ERD (Entity Relationship Diagram)

Entity Relation Diagram (ERD) pada sistem layanan repository berbasis search engine menggunakan algoritma knuth morris pratt.



Gambar 3. Entity Relationship Diagram

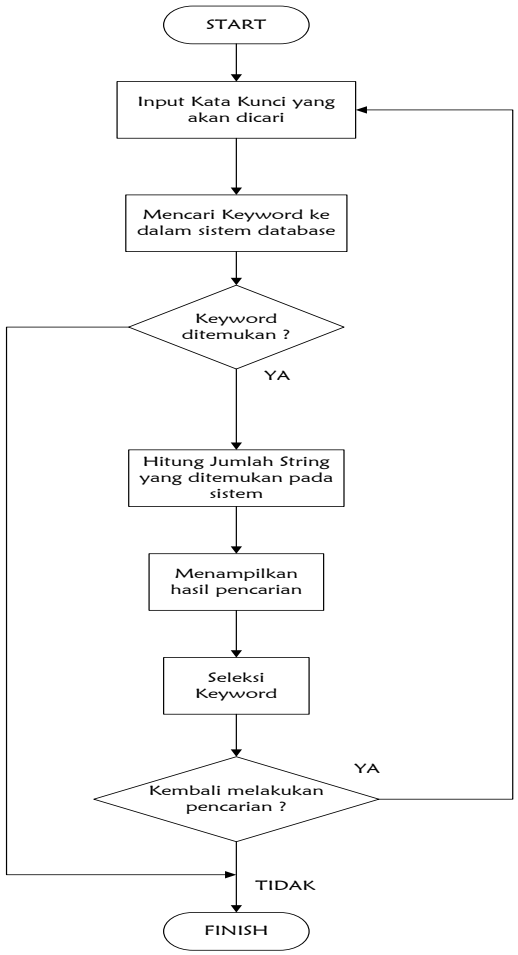
Pada gambar ERD diatas dapat menjelaskan database dari sistem layanan repository berbasis web, yang mendeskripsikan hubungan antara table-tabel yang terjadi didalam database, seperti: many to many, one to many dan many to one.

D. Perancangan Algoritma Knuth Morris Pratt

Dalam peracangan Sistem repository yang utama harus dirancang adalah Flowchart sistem kerja website berbasis search engine menggunakan algoritma KMP. Adapun langkah-langkah perancangan SMS Gateway adalah sebagai berikut :

1) Flowchart Algoritma KMP

Berikut ini merupakan langkah-langkah yang dilakukan untuk menerapkan fungsi search engine pada website, dapat dilihat pada gambar 4.



Gambar 4. Flowchart Algoritma KMP

Adapun langkah-langkah penerapan algoritma Knuth Morris Pratt agar dapat berjalan pada sistem pencarian web adalah sebagai berikut:

a) Membuat tampilan untuk search bar pada tampilan interface website.
b) Membuat file fungsi KMP.php, listing program algoritma Knuth Morris Pratt seperti dibawah ini

```
<?php
error_reporting(0);
class KMP{
  function KMPSearch($p,$t){
    $hasil = array();
    $pattern = str_split($p);
    $text    = str_split($t);
    $lompat = $this->preKMP($pattern);
 //print_r($lompat);
 // perhitungan KMP
 $i = $j = 0;
    $num=0;
    while($j<count($text)){
      while($i>-1                    &&
$pattern[$i]!=$text[$j]){
     // jika tidak cocok, maka lompat
sesuai tabel lompatan
        $i = $lompat[$i];
      }
      $i++;
      $j++;
      if($i>=count($pattern)){
     // jika cocok, tentukan posisi
string yang cocok
  // kemudian lompat ke string
berikutnya
        $hasil[$num++]=$j-
count($pattern);
        $i = $lompat[$i];
      }
    }
  return $hasil;
  }

  /* menetukan tabel lompatan dengan
preKMP
   * input :
   *   $pattern = (string) pattern
   * output :
   *  $lompat = (array int) untuk jumlah
lompatan
   */
  function preKMP($pattern){
    $i = 0;
    $j = $lompat[0] = -1;
    while($i<count($pattern)){
      while($j>-1                    &&
$pattern[$i]!=$pattern[$j]){
        $j = $lompat[$j];
      }
      $i++;
      $j++;
      if($pattern[$i]==$pattern[$j]){
        $lompat[$i]=$lompat[$j];
      }else{
        $lompat[$i]=$j;
      }
    }
    return $lompat;
  }

  /* replace string
   * input :
   *    $str1 = (array string) string
yang akan diganti dengan str2
   *    $str2 = (array string) string
yang akan mengganti str1
   *   $text = (string) text yang akan
dicari
   * output :
   *   $t = teks yang sudah difilter
   */
  function
KMPReplace($str1,$str2,$text){
    $num = 0;
    $location         =         $this-
>KMPSearch($str1,$text);
  $t = '';
  $n = 0; $nn = 0;
  foreach($location as $in){
    $t   .=   substr($text,$n+$nn,$in-$n-
$nn).$str2;
    $nn = strlen($str1);
```

```
      $n = $in;
    }
    $t .= substr($text,$n+$nn);
    return $t;
     }
   }
   ?>
```

c) Kemudian melakukan koneksi dengan file KMP.php agar dapat berjalan pada laman menu utama website.

III. HASIL DAN PEMBAHASAN

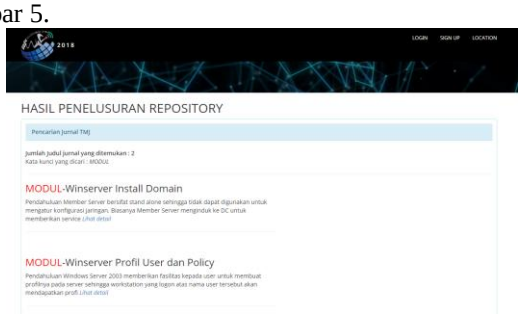
A. Pengujian Algoritma Knuth Morris Pratt (KMP)

Pengujian sistem merupakan proses yang dilakukan untuk mengetahui seberapa jauh Website ini telah berjalan. Pada tahap inilah semua proses akan dikerjakan mulai dari proses pencarian tiap *string*, mencocokan antara kata unik dari kumpulan kata yang diinput atau ada dalam *database*.

Berikut potongan *listing program* untuk koneksi menggunakan pencarian dengan algoritma *Knuth Morris Pratt*.

```
<?php
    $c  = mysql_connect("localhost",
"root", "");
    $db = mysql_selectdb("cloud",$c);

    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'];
      $sql  ="SELECT * FROM download
WHERE nama_file LIKE '%$kata%' AND
status='PUBLISH'";
      $result = mysql_query($sql);
      $jumlah = mysql_num_rows($result);

    if (empty($kata)){
        echo "<script
language='javascript'>alert('Anda Belum
Memasukkan Judul Tugas Akhir Yang Ingin
Dicari!!!');
document.location='koneksi_kmp.php'</scri
pt>";
    }
    else if ($jumlah>0){
        echo "<h2>REPOSITORY TEKNIK
MULTIMEDIA DAN JARINGAN</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,$teks['nama_file']);
        echo "<div
style='width:600px;'>";
        echo
nl2br(str_replace($kata,"<font
color='red'>".$kata."</font>",$teks['nama
_file']));
        echo " <font='blue'><a
href='repositori_lengkap.php?NPM=$teks[ab
strak]'>Baca Selengkapnya</a></font>";
        echo "<br><br/>";
        echo "</div>";
    }
    } else{
        echo "JUDUL YANG ANDA
MAKSUDCARIKAN TIDAK ADA!!";
      }
    }
?>
```

Pada Menu *Homepage* yang berada pada halaman utama website, data akan muncul ketika pencaharian mulai melakukan proses dan semua orang dapat mengaksesnya tanpa harus melakukan tahapan registrasi terlebih dahulu. Pada menu *homepage* ini data akan tampil sesuai dengan inputan kata yang dimasukkan atau dicari. Berikut merupakan proses hasil penelusuran kata, dapat dilihat pada gambar 5.

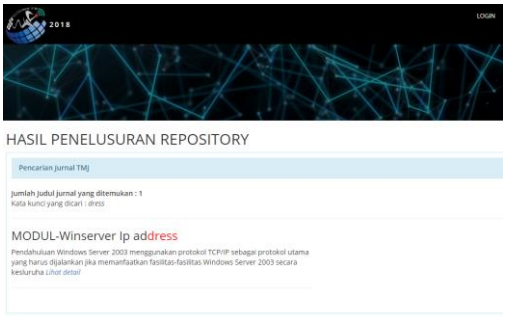


Gambar 5. Tampilan Hasil Penelusuran awal

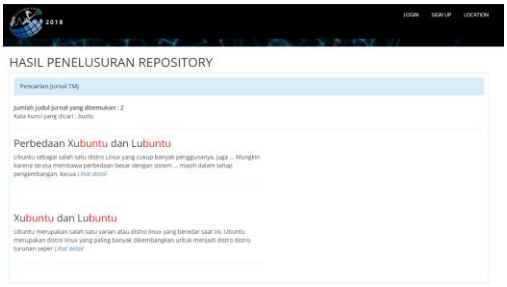
Kemudian jika ingin lihat *info detail* akan tampil seperti gambar 6

Gambar 6. Tampilan Detail Penelusuran

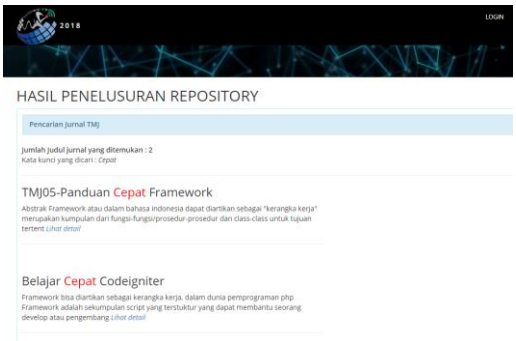
Berikut merupakan percobaan pencarian kata berdasarkan beberapa kriteria yang telah diuji pada system, diantara lain terdapat pada gambar 7, gambar 8 dan gambar 9 dibawah ini



Gambar 7. Hasil Penelusuran karakter kata



Gambar 8. Hasil Penelusuran 2 kata dalam satu judul teks



Gambar 9. Hasil Penelusuran 1 kata dalam dua judul teks berbeda

B. Pembuktian Manual Pencarian Kata

Pada saat melakukan proses pencarian jurnal atau file dokumen, pengunjung atau visitor website hendak ingin mencari jurnal atau dokumen yang hendak akan dibaca. Misalkan ada kasus pengunjung mencari judul, kata kunci yang akan dicari adalah "ADDRESS", dan pattern yang akan dicari adalah "DRESS". Pengguna ingin mencari jurnal secara tepat, dan memilih algoritma untuk pencarian text tersebut.

Penyelesaian untuk kasus diatas jika menggunakan algoritma *Knuth Morris Pratt* adalah sebagai berikut:

1. Terlebih dahulu menentukan *pattern* dan teks yang akan dicari. Dalam kasus ini, pattern dan teks yang hendak dicari adalah sebagai berikut:

   P = DRESS
   T = ADDRESS
2. Selanjutnya menentukan fungsi pinggiran dari pattern dan teks yang akan dicari. Fungsi pinggiran disini didefinisikan sebagai ukuran awalan terpanjang dari pattern. Fungsi pinggiran tersebut telah dijabarkan dalam Tabel I

TABEL I

Fungsi Pinggiran kasus pencarian *string*

J	0	1	2	3	4	5	6
p (j)	A	D	D	R	E	S	S
b (j)	0	1	3	0	0	0	0

3. Memberikan nilai indeks atau lakukan peng-index-an ke *pattern* dan teks. Adapun nilai indeks pattern dan teks akan dipaparkan dalam table II dan III

TABEL II

Nilai indeks untuk Teks

INDEKS	0	1	2	3	4	5	6
t (j)	A	D	D	R	E	S	S

TABEL III

Nilai indeks untuk *Pattern*

INDEKS	0	1	2	3	4
p (j)	D	R	E	S	S

4. Cara perhitungan pergeseran dari algoritma *Knuth-Morris-Pratt* adalah sebagai berikut :
   a. Membandingkan ujung kiri pada teks dan pada pattern.

   Pada ujung kiri teks dan ujung kiri *pattern* terjadi ketidakcocokan, tetapi pada nilai indeks ke 3, terjadi kecocokan. Teks dengan karakter indeks pertama A sedangkan pattern ketiganya yaitu D.
   b. Karena terjadi ketidakcocokan, maka lakukan pergeseran *pattern* P dengan jumlah pergeseran sesuai dengan nilai pinggiran pattern p yang cocok. Pada kasus ini, karakter pada *pattern* dan teks yang menemukan kesamaan ada di indeks 2-6, maka panjang kesesuaian teks adalah 5 (*l=5*).
   c. Nilai pinggiran terpanjang dari *pattern* P yang menemukan kecocokan adalah P[2…6], dimana dalam fungsi pinggiran sebelumnya, nilai *pattern* pada indeks urutan 3 dalam fungsi pinggiran adalah 3 (b(3) =()).
   d. Setelah ditentukan nilai fungsi pinggiran dan panjang dari kecocokan teks dan *pattern*, maka pergeseran karakter dilakukan dengan cara :

Nilai Pergeseran = $l - b$

Keterangan :

Nilai pergeseran : Nilai yang digeser tiap karakternya

I = Panjang Kecocokan karakter antara *pattern* dan teks

b = Nilai dalam fungsi pinggiran

Maka, Nilai Pergeseran = $l - b$

= 5 – 3

= 2karakter

Jadi, pattern P digeser sebanyak 2 karakter ke kanan. Pengeseran tersebut akan terlihat pada table IV

TABEL IV

Nilai indeks untuk *Pattern*

INDEKS	0	1	2	3	4	5	6
TEKS	A	D	D	R	E	S	S
INDEKS P	PERGESER		0	1	2	3	4
PATTERN	AN		D	R	E	S	S

.

IV. KESIMPULAN

Berdasarkan hasil dan pembahasan pada penelitian yang telah dilakukan, maka dapat disimpulkan bahwa:

1. Berdasarkan pengujian yang telah dilakukan sistem *repository* berbasis *search engine* dengan menerapkan algoritma *Knuth Morris Pratt* yang telah berjalan dan sudah mendapatkan hasil pencarian dalam sistem serta dapat mempermudah pengguna untuk menemukan kata pada judul TGA, Jurnal dan karya ilmiah yang berhubungan dengan kata kunci yang sedang menjadi pusat pencarian.
2. Untuk pengujian secara manual hasil pencarian yang dilakukan oleh website berdasarkan judul TGA, jurnal dan karya ilmiah berhasil menampilkan kata yang ingin di cari di dalam database. Dan ketidak cocokan kata terjadi apabila data yang ingin dicari tidak ada dalam database. Pattern akan menyesuaikan dengan teks sebanyak nilai -n pada teks dilakukan pergeseran dari kiri ke kanan setiap karakter kata hingga terjadi kecocokan dari pattern dengan teks.
3. Pencarian kata juga dapat menghitungkan waktu eksekusi dalam satuan detik, untuk setiap pengujian pada sistem dapat dianalisis bahwa setiap 1,1 *milisecond* untuk beberapa pengujian dalam setiap kumpulan kata yang dicari mulai dari pencarian satuan kata, pencarian kata yang sama dalam dua judul berbeda dan pencarian kata yang sama dalam satu judul.

V. REFERENSI

[1] Aisa, S. (2014). Implementasi Private Cloud Menggunakan Raspberry pi Untuk Pengaksesan Data pribadi, 16.

[2] Andre. (2014). https://www.duniailkom.com/ pengertian-dan-fungsi-php-dalampemograman-web/. Retrieved from www.duniailkom.com.

[3] Diartono, D. A. (2010). Integrasi Sistem Presensi Finger Print dan Sistem SMS gateway untuk Monitoring Kehadiran Siswa. Retrieved september 28, 2017, from http://www.unisbank.ac.id/ojs/index.php/fti1/article/view/114

[4] Diartono, D. A. (2010). Integrasi Sistem Presensi Finger Print dan Sistem SMS Gateway untuk Monitoring Kehadiran Siswa. Jurnal Teknologi Informasi DINAMIK, 73.

[5] Felix Andreas Susanto, J. A. (2015). Implementasi Dashboard untuk Sistem Monitoring Bimbingan dan Konseling Siswa. Retrieved 09 27, 2017, from http://www.unisbank.ac.id/ojs/index.php/fti1/article/view

[6] Firmansyah.(2017) .hadribonjay.it.student.pens.ac.id/. Retrieved