

Mesin Pencari Dokumen Karya Ilmiah pada Lingkungan Akademik

(*Scientific Documents Search Engine in Academic Environment*)

B.Very Christioko
Fakultas Teknologi Informasi dan Komunikasi Universitas Semarang

Abstract

Currently, search engines have become common programs for internet users, especially for academics, both students and lecturers. The purpose of this paper is to build a scientific document search engine in an academic environment using internet technology (PHP & MySQL), so the system can be accessed anywhere. With this search engine, academics can find some existing scientific documents easily to use them as references.

Keywords: search engine, scientific document, PHP, MySQL

1. PENDAHULUAN

Kemajuan teknologi informasi (TI) telah membawa perubahan dalam berbagai aspek kehidupan manusia, khususnya dalam dunia pendidikan. Bermunculannya mesin pencari di internet mendorong pengguna internet untuk menggali lebih dalam potensi-potensi yang terdapat dalam jaringan internet. Dengan mesin pencari Google, kita dimudahkan untuk melakukan pencarian informasi baik halaman web, gambar, video, berita yang terkait dengan kata kunci yang kita berikan. Dengan mudahnya akademisi mengakses internet untuk mencari referensi dari karya ilmiah yang sedang dibuat, dapat juga menimbulkan beberapa permasalahan, yaitu budaya menjiplak. Yang mungkin sering terjadi yaitu judul atau topik karya ilmiah yang mempunyai kemiripan dengan karya-karya ilmiah yang sudah ada. Oleh karena itu perlu untuk membangun sebuah mesin pencari dan membangun database untuk pendataan karya ilmiah pada lingkungan Universitas. Mesin pencari dibuat dengan teknologi web dimana dapat diakses pada lingkungan intranet menggunakan protokol HTTP. Pengguna dari mesin pencari ini adalah Dosen pembimbing, namun tidak menutup kemungkinan bagi mahasiswa yang bermaksud untuk mencari referensi sebagai studi pustaka.

2. METODE

2.1. Membangun Database

Dalam tahap ini dipersiapkan sebuah database untuk menampung data karya ilmiah dan indek kata kunci untuk pencarian. Database terdiri dari tiga tabel yaitu tabel content untuk menampung daftar karya ilmiah yang terdiri dari *contid*, *title* dan *abstract*, tabel *keyword* untuk menampung daftar kata kunci yang terdiri dari *keyid* dan *keyword*, dan tabel *link* untuk menampung daftar relasi antara tabel *content* dan *keyword* yang terdiri dari *keyid* dan *contid*.

2.2. Membuat Kata Buangan

Menyusun kata buangan yaitu kata sambung yang biasa digunakan pada bahasa indonesia, tujuannya untuk menyaring kata kunci dengan kata buangan sehingga terbentuk kata kunci yang akan disimpan pada tabel *Keyword*. Kata sambung antara lain :

- a. Kata sambung asal, misalnya : dan, maka, sedang, hingga, meski, lalu, bila, sambil, atau, serta, karena, jika, dll.
- b. Kata sambung jadian / bentukan:
 - kata ulang, misalny: jangan-jangan, seakan-akan, kalau-kalau, dll.

- Kata sambung majemuk, misalnya: apabila, lagi pula, karena itu, andaikata, sebab itu, dll.
- Kata sambung berimbuhan, misalnya: sebelum, selama, sehingga, seandainya, sekiranya, melainkan, semenjak, andaikan, bagaikan, asalkan, sedangkan, jangankan, walaupun, meskipun, kendatipun, bermula, sebermula, dll.

2.3. Menyiapkan Database

Untuk menyiapkan isi tabel pada *database*, dibuat sebuah form input data karya ilmiah yang secara otomatis akan disimpan pada tabel *content* dan akan melakukan *parsing* terhadap isi dari abstrak karya ilmiah tersebut untuk disimpan pada tabel *keyword* dan *link*.

Dalam tahap ini terdapat 4 fungsi yang disebut *UPLOAD ENGINE*, yaitu :

- Fungsi mengambil semua kata kunci yang terdapat pada tabel *Keyword*, tugas ini didefinisikan pada fungsi *LoadCurrentWords()*.
- Fungsi untuk memecah kata-kata pada abstrak, tugas ini didefinisikan pada fungsi *ExtractWords()*.
- Fungsi untuk menyaring kata hasil fungsi *ExtractWords()* dengan kata buangan dan duplikasi data, tugas ini didefinisikan pada fungsi *FilterCommonAndDuplicateWords()*.
- Fungsi untuk memproses penyiapan *database* pada tabel *Keyword* dan tabel *link*, tugas ini didefinisikan pada fungsi *ProcessForm()*.

2.4. Membuat Mesin Pencari

Dalam tahap ini dibuat sebuah antarmuka untuk pengguna berupa mesin pencari menggunakan HTML, yang terdiri dari *textbox* isian kata kunci dan tombol pencarian. Dalam tahap inilah proses pencarian terjadi dari mulai pengguna memberikan kata pencarian, kemudian

SEARCH ENGINE melakukan pencarian pada *database* untuk kemudian ditampilkan ke browser apabila *link* ditemukan.

3. PEMBAHASAN

3.1. Membangun Database

Untuk menampung judul dan abstrak karya ilmiah dibuat sebuah tabel *Content* dengan *field* (*Contid int(9)*, *Title text*, *Abstract Longtext*), dan untuk menampung daftar kunci dibuat sebuah tabel *Keyword* (*Keyid int(9)*, *Keyword varchar(100)*). Relasi antar tabel *Content* dan *Keyword* dibuat sebuah tabel *Link* (*Keyid (int(9)*, *Contid int(9)*). Skema dari tabel diatas sbb:

Tabel Content :

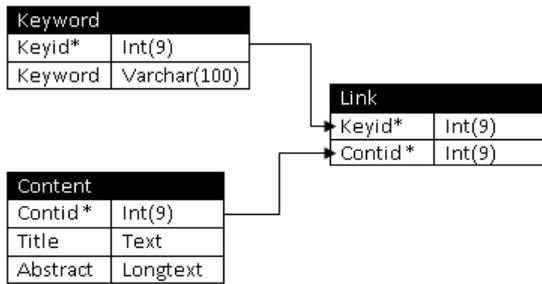
```
CREATE TABLE content (
  contid mediumint(9) NOT NULL
  auto_increment,
  title text,
  abstract longtext,
  PRIMARY KEY (contid) ) TYPE=MyISAM;
```

Tabel Keyword :

```
CREATE TABLE keyword (
  keyid mediumint NOT NULL auto_increment,
  keyword varchar(100) default NULL,
  PRIMARY KEY (keyid),
  KEY keyword (keyword) ) TYPE=MyISAM;
```

Tabel Link :

```
CREATE TABLE link (
  keyid mediumint NOT NULL,
  contid mediumint NOT NULL)
  TYPE=MyISAM
```



Gambar 1. Skema Database karya_ilmiah

3.2. Membuat Kata Buangan

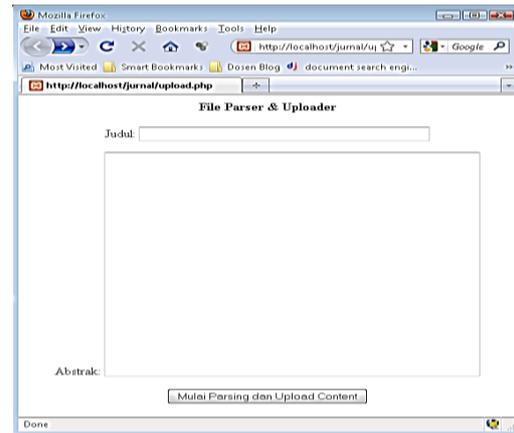
Kata kunci harus sudah dipisahkan dari kata buangan dalam hal ini adalah kata sambung, dibuat sekumpulan kata buangan yang terdiri dari kata sambung ke dalam sebuah variabel array. Untuk mendapatkan kata kunci, abstrak disaring dengan kata buangan kemudian disimpan dalam tabel *Keyword*.

```
$COMMON_WORDS = array("dan"=>1,
"maka"=>1, "sedang"=>1, "hingga"=>1,
"meski"=>1, "lalu"=>1, "bila"=>1, "sambil"=>1,
"atau"=>1, "serta"=>1,"karena"=>1, "jika"=>1)
```

3.3. Menyiapkan Database

Untuk menyiapkan database, dibuat sebuah form HTML dengan isian judul dan abstrak karya ilmiah. Setelah form diisi dan disimpan, judul dan abstrak disimpan pada tabel *Content*. Kemudian pada tahap 'Upload Engine', seluruh isi abstrak dihilangkan dari kata-kata buangan kemudian disimpan pada variabel array *wordmap* dimana setiap kata kunci hanya terdapat satu.

Setiap kata kunci dalam variabel *array wordmap* akan diparsing dengan tabel *Keyword*. Jika terdapat kata kunci yang sama maka akan dibuat *record* baru pada tabel *link* yang berisi *Contid* dan *Keyid*, jika tidak maka kata kunci baru akan di tambahkann pada tabel *Keyword*. Dengan ditambahkan record-record baru pada tabel *Content* dan *Keyword*, tabel *Link* akan diupdate.



Gambar 2. Form Parsing & Upload Engine

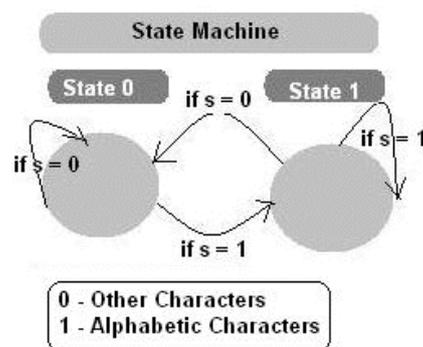
Fungsi yang terdapat dalam Upload Engine, yaitu

a. LoadCurrentWords()

Pada fungsi ini kata kunci yang terdapat pada tabel *Keyword* akan diambil dan disimpan pada variabel Array *\$Allword*, dengan nilai *\$Allword[Katakunci]=>id*.

b. ExtractWords()

Pada fungsi ini seluruh kata akan dipecah dan hanya huruf saja yang diperbolehkan disimpan dalam variabel array *\$wordlist*. Untuk mengimplementasikan ini maka dibuat 2 kondisi status, yaitu status1 jika kata tersusun dari huruf, jika tidak maka kondisi status0.



Gambar 3. Bagan State Machine

c. FilterCommonAndDuplicateWords()

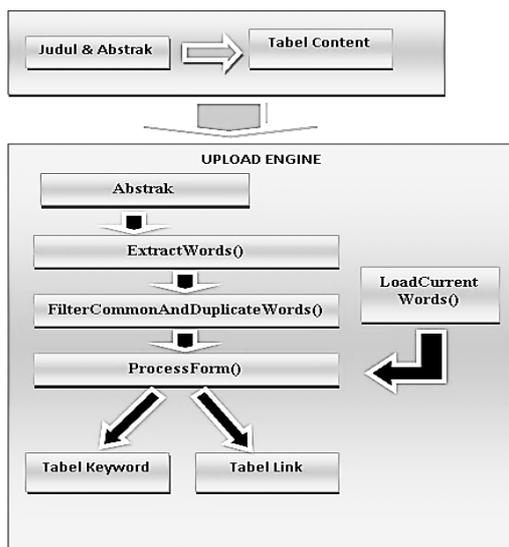
Fungsi ini dipanggil setelah proses pemisahan kata dari fungsi *ExtractWord()*. Fungsi ini akan menyaring seluruh kata dengan menghilangkan kata buangan yang

sudah dibangun. Kemudian fungsi ini akan menghilangkan kata-kata yang ganda, sehingga hanya ada satu record saja setiap kata kunci. Hasil parsing ini kemudian disimpan dalam variabel array \$wordmap.

d. ProcessForm()

Fungsi ini adalah inti dari *Upload Engine*. Setelah kata kunci diparsing, dihilangkan dari kata buangan, maka fungsi ini akan dipanggil. Setelah proses insert data judul dan abstrak ke dalam tabel Content, content id baru akan dibuat dan disimpan dalam variabel \$contentid kemudian mengupdate tabel *Keyword* dan tabel *link*.

Untuk setiap kata kunci dalam variabel array \$wordmap akan di *query* pada tabel *Keyword*, jika kata kunci sudah ada maka *record* baru *Keyid* dan *Contentid* akan ditambahkan pada tabel *Link*, jika kata kunci belum ada maka kata kunci baru akan ditambahkan pada tabel *Keyword* dan *Keyid* akan terbentuk untuk kemudian melakukan update pada tabel *link* untuk menambahkan *Keyid* dan *Contentid*.

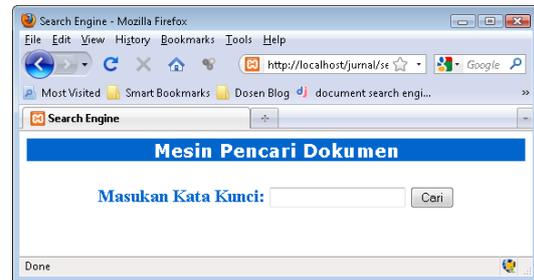


Gambar 4. Bagan Upload Engine

3.4. Membuat Mesin pencari

Mesin pencari dibuat menggunakan script PHP menggunakan form HTML. Sama seperti mesin pencari lainnya, pengguna memasukkan kata kunci pada *textbox* isian,

kemudian tekan enter, kemudian hasil pencarian akan ditampilkan dengan *link* menuju ke karya ilmiah yang dimaksud.



Gambar 5. Mesin pencari dokumen

Langkah-langkah dari proses pencarian sebagai berikut :

- Kata pencarian di konversi ke dalam huruf kecil dengan fungsi `strtolower()`, kemudian disimpan dalam variabel `$search_keyword`.

```
$search_keywords=strtolower(trim($keywords));
```

- `$search_keyword` dipecah untuk kemudian disimpan pada variabel Array `$arrWord`, menggunakan fungsi `explode()`.

```
$arrWords = explode(" ", $search_keywords);
```

- Hilangkan duplikasi kata pada variabel array `$arrWord` menggunakan fungsi `array_unique()`.

```
$arrWords = array_unique($arrWords);
```

- Hilangkan kata-kata buangan pada variabel array kata pencarian `$arrWord`.

- Cari kata kunci variabel array `$search_keyword` pada tabel *Keyword*, jika ditemukan maka simpan *Keyid* pada variabel `$kid`.

- Berdasarkan id *Keyword* `$kid` yang ditemukan, cari *Keyword* id pada tabel *link*,

dan jika ditemukan, simpan pada variabel array \$contArray.

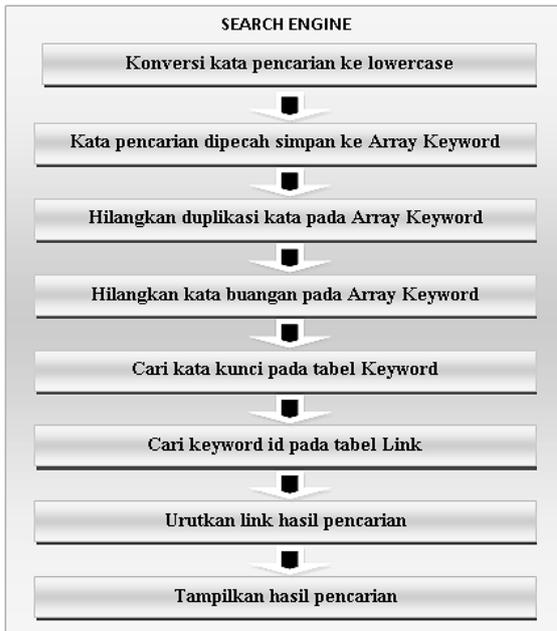
- g. Urutkan hasil pencarian pada tabel link dengan urutan Descending, misalkan terdapat tiga hasil pencarian maka link ditampilkan dari urutan 3, 2, kemudian 1.
- h. Tahap terakhir adalah menampilkan hasil pencarian pada browser menggunakan tag HTML.

Table	Action	Records ¹	Type
<input type="checkbox"/> content		1	MyISAM
<input type="checkbox"/> keytable		145	MyISAM
<input type="checkbox"/> link		145	MyISAM
3 table(s) Sum		291	MyISAM

Gambar 6. Hasil upload dokumen

content	title	abstrak
1	ANALISIS BERBAGAI FAKTOR PENENTU DALAM PENYELARASA	Saat ini, hampir semua perusahaan kecil, menengah...
2	E-LEARNING SEBAGAI PENDUKUNG PENGAJARAN DAN RISET	Tulisan ini bertujuan untuk menunjukkan feasibility...

Gambar 7. Browse tabel Content



Gambar 5. Bagan Search Engine

	keyid	keyword
<input type="checkbox"/>	1	saat
<input type="checkbox"/>	2	ini
<input type="checkbox"/>	3	hampir
<input type="checkbox"/>	4	semua
<input type="checkbox"/>	5	perusahaan
<input type="checkbox"/>	6	kecil
<input type="checkbox"/>	7	menengah
<input type="checkbox"/>	8	maupun
<input type="checkbox"/>	9	enterprise

Gambar 8. Browse tabel Keyword

	keyid	contid
<input type="checkbox"/>	1	1
<input type="checkbox"/>	2	1
<input type="checkbox"/>	3	1
<input type="checkbox"/>	4	1
<input type="checkbox"/>	5	1
<input type="checkbox"/>	6	1
<input type="checkbox"/>	7	1
<input type="checkbox"/>	8	1
<input type="checkbox"/>	9	1

Gambar 9. Browse tabel Link

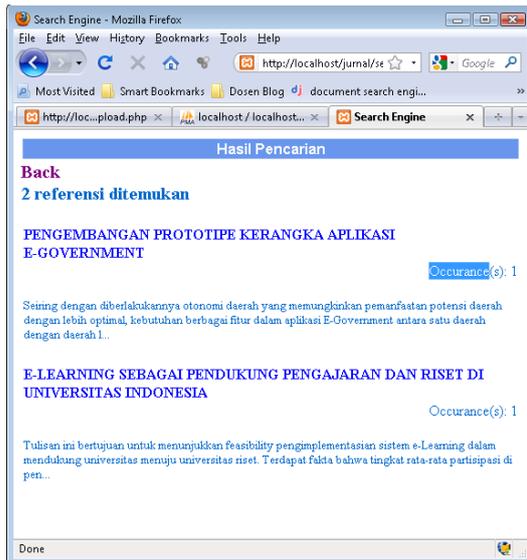
4. PENGUJIAN

Berikut beberapa pengujian terhadap mesin pencari dokumen :

- a. Pengujian terhadap fungsi upload karya ilmiah dengan mengupload sebuah karya ilmiah yaitu judul dan abstrak, hasil kata kunci dan link yang dihasilkan sebagai berikut :

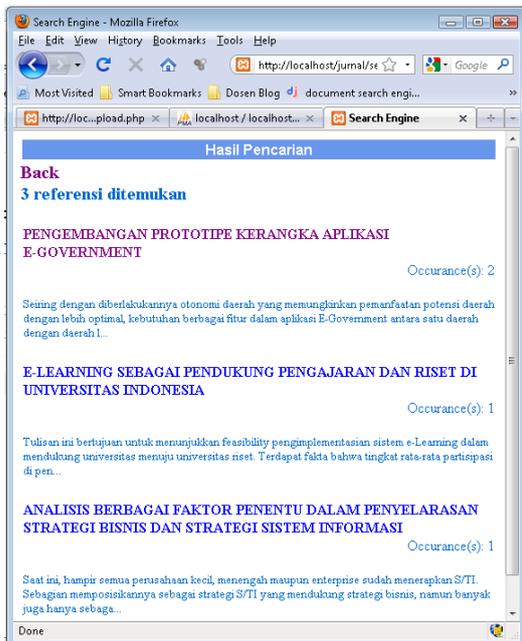
Dengan mengupload sebuah karya ilmiah dapat diperoleh katakunci : 145, dan link : 145.

- b. Pengujian terhadap fungsi mesin pencari dengan memasukkan kata kunci pencarian pada kotak text box mesin pencari. Dengan satu kata pencarian : **sistem**, ditemukan dua buah dokumen karya ilmiah seperti gambar dibawah ini.



Gambar 10. Hasil Pencarian satu kata

Dengan dua kata pencarian : **informasi bisnis**, ditemukan tiga buah dokumen karya ilmiah seperti gambar dibawah ini.



Gambar 11. Hasil Pencarian dua kata

5. KESIMPULAN

Mesin pencari dokumen karya ilmiah hanya mencari kata-kata yang terdapat dalam abstrak karya ilmiah. Kata kunci diperbolehkan lebih dari satu kata, apabila kata kunci lebih dari satu kata maka hasil pencarian dapat ditampilkan jika salah satu dari kata kunci ditemukan pada dokumen. Dalam pengembangan berikutnya tidak ditutup kemungkinan untuk melakukan pencarian pada judul, nama pembimbing, nama penguji, bahkan isi dari dokumen karya ilmiah tersebut.

DAFTAR PUSTAKA

1. M.Murali Dharan, "Dynamic Document Search Engine – Part1", <http://www.phpbuilder.com>, 2003.
2. M.Murali Dharan, "Dynamic Document Search Engine – Part2", <http://www.phpbuilder.com>, 2003.
3. Mitchell Harper, "Developing A Site Search Engine With PHP And MySQL", <http://www.devarticles.com>, 2002.
4. Thony Mansda, "Kata sambung", <http://wacana-bahasa.blogspot.com>