

SISTEM APLIKASI PENCARIAN DOKUMEN BERBASIS JUDUL DOKUMEN DENGAN PENERAPAN PROFIL *DIGRAM* PADA *VECTOR SPACE MODEL* DENGAN PEMBOBOTAN BINER SERTA KESETARAAN *OCHIAI*

Dewi Kusumaningsih¹, Ryanda Rachman², Asep Ahmadi³, Indra Nugroho⁴

^{1,2}Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur
Jl. Ciledug Raya, Pertukangan Utara, Jakarta Selatan, 12260, Indonesia
¹dewi.kusumaningsih@gmail.com, ²ryandar@gmail.com

^{3,4}Program Studi Magister Ilmu Komputer, Universitas Budi Luhur
Jl. Ciledug Raya, Pertukangan Utara, Jakarta Selatan, 12260, Indonesia
³asep.ahmadi@explorindo.com, ⁴indnugroho@yahoo.com

ABSTRAK

Perkembangan teknologi pada era globalisasi saat ini, membawa pengaruh besar dalam proses pencarian informasi, kebutuhan akan informasi juga harus cepat dan dapat memudahkan user tanpa harus menyetik secara persis query judul dokumen yang akan dicari. Sistem aplikasi pencarian dokumen ini dibuat penulis bertujuan untuk mengevaluasi efektifitas dari suatu sistem perolehan informasi (*information retrieval*) yang menggunakan model pendekatan pengukuran kesamaan digram dengan cara membandingkan satu model pendekatan model lain. Dengan menerapkan tahap pembentukan token, pembobotan profil (*digram*), dan fungsi *similarity*. Dari hasil penghitungan rata-rata *precision* dan *recall* akan didapatkan kumpulan titik efektifitas yang pada akhirnya penelitian ini akan digambarkan dalam bentuk tabel. Selanjutnya tabel tersebut akan disatukan dengan satu tabel yang menampilkan rata-rata *precision* dan *recall* pada penelitian yang terkait, untuk dilakukan perbandingan dalam memperoleh kesimpulan akhir. Pada *Vector Space Model*, setiap dokumen di dalam database dan query pengguna direpresentasikan oleh suatu *vector* multi-dimensi sesuai dengan jumlah profil dalam dokumen yang terlibat pada penelitian ini, metode yang digunakan untuk melakukan pemprofilan teks ialah dengan menggunakan metode *digram*. Dimana teks akan dibaca perhuruf yang nantinya akan dibuat *digram* dari huruf-huruf tersebut dan pembobotan biner dimana setiap *digram* yang ada pada sebuah string diberi bobot 1. Fungsi dari kesetaraan *Ochiai* adalah untuk mengetahui *similarity* atau tingkat kesetaraan antara query yang di-input dengan dokumen yang berada di dalam database, jika *similarity* dokumen yang di-input melebihi dari *threshold* yang telah dipilih, maka dokumen tersebut adalah dokumen yang relevan untuk ditampilkan di tabel hasil pencarian.

Kata kunci : pencarian dokumen, *digram*, pembobotan biner, *ochiai*

I. PENDAHULUAN

1.1 Latar Belakang

Perpustakaan merupakan penyedia informasi dan pelestarian yang berperan untuk keperluan pendidikan, penelitian, pengembangan ilmu dan pembangunan nasional. Sehingga pada saat ini perpustakaan dituntut untuk mengikuti perkembangan kemajuan dunia teknologi dalam hal kecepatan memperoleh suatu informasi. Fungsi utama setiap perpustakaan atau pusat informasi adalah mengadakan, mengolah, menyediakan, dan menyebarkan informasi kepada pengguna. Untuk melaksanakan fungsi tersebut, maka perpustakaan harus mengolah dan mengatur koleksinya sedemikian rupa sehingga informasi tersebut dapat disimpan dan ditemukan kembali secara mudah, cepat, dan tepat jika diperlukan oleh pengguna. Dengan kata lain, di dalam perpustakaan diperlukan suatu sistem perolehan informasi (*Information Retrieval System*) yang baik untuk mempermudah penggunaannya dalam mencari suatu buku atau referensi yang diharapkan.

1.2 Masalah

Dengan melihat pada latar belakang diatas, maka didapatkan beberapa poin permasalahan sebagai berikut:

- Pencarian buku pada sistem di perpustakaan Universitas Budi Luhur saat ini sangat *sensitive* terhadap kata-kata atau kalimat yang di-input dengan judul-judul buku yang ada di *database*, sehingga jika ada buku yang judulnya mirip namun menyimpang (salah ketik), maka sistem tidak dapat menampilkan judul buku yang diharapkan.
- Mesin pencari yang ada kurang efisien, pengguna harus masuk ke beberapa *form* berbeda terlebih dahulu dan melakukan proses pengaturan ketika ingin melakukan pencarian dokumen.
- Mesin pencari yang ada sulit untuk mencari dokumen jika pengguna memasukkan kata kunci yang beragam

1.3 Tujuan Penulisan

Adapun tujuan penulisan dari penelitian ini adalah:

- Terbentuknya sebuah sistem pencarian dokumen yang toleran terhadap kesalahan pengetikan oleh pengguna, dimana ketika pengguna salah atau kurang tepat dalam memasukkan kata kunci, maka sistem tetap dapat menampilkan dokumen yang dicari.
- Membuat sebuah mesin pencari yang lebih efisien, dimana pengguna tidak perlu lagi melakukan pengaturan ketika melakukan pencarian, cukup dengan mengetikkan kata kunci dari dokumen yang dicari.
- Membuat sebuah mesin pencari yang dapat menebak kata kunci yang beragam dari *query* yang dimasukkan oleh pengguna.

1.4 Batasan Masalah

Untuk mencapai tujuan diatas, maka peneliti memfokuskan batasan masalah hanya pada peneitian judul dokumen yang ada dan terdaftar di perpustakaan Universitas Budi Luhur. Dokumen-dokumen yang dimaksudkan berupa jurnal, buku, paper KKP, paper TA dan dokumen lainnya yang memiliki kode buku perpustakaan.

1.5 Metode Pengembangan

Beberapa metode pengembangan yang digunakan dalam mengembangkan program aplikasi pencarian judul dokumen ini meliputi :

a. Metode Kepustakaan

Metode ini dilakukan untuk mengumpulkan data dengan mencari dan membaca buku-buku referensi yang dapat menunjang penyusunan skripsi ini. Serta mencari informasi dengan cara memanfaatkan situs *search engine* seperti www.google.com, serta melihat pembahasan yang sudah ada yang berkaitan di internet.

b. Metode Wawancara

Dengan cara langsung datang ke perpustakaan Universitas Budi Luhur dan melakukan wawancara dengan pengelola perpustakaan, untuk mengetahui permasalahan yang dihadapi dan mendiskusikan aplikasi yang diharapkan dapat memecahkan masalah tersebut.

c. Observasi

Melihat langsung alur proses pada dokumen serta mempelajari dokumen yang digunakan dan laporan-laporan yang dihasilkan untuk mengumpulkan data.

d. Membuat Rancangan Aplikasi

Membuat rancangan aplikasi sesuai hasil yang dilakukan dengan membuat *activity diagram*, *statechart diagram*, *class diagram*, rancangan layar, rancangan *database*, dan *flowchart*.

e. Implementasi

Rancangan aplikasi yang telah dibuat dalam bentuk *web* kemudian diimplementasikan sesuai dengan hasil analisa.

f. Uji Coba Aplikasi

Setelah aplikasi setelah dibuat, maka dilakukanlah uji coba terhadap aplikasi untuk mendapatkan evaluasi program.

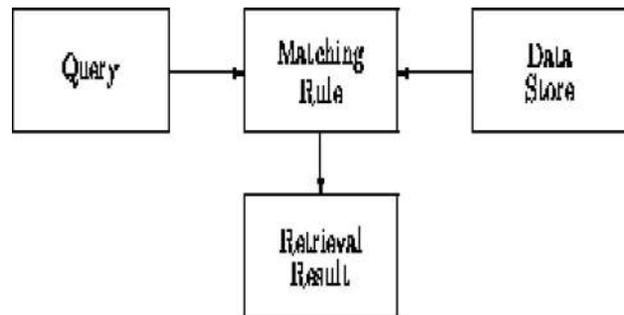
II. LANDASAN TEORI

2.1 Konsep Umum Manajemen Perpustakaan

Jogiyanto H.M (2005) mendefinisikan informasi adalah data yang dapat diolah yang lebih berguna dan berarti bagi yang menerimanya[1]. Jo Bryson dalam Lasa (2007) menyatakan bahwa manajemen perpustakaan merupakan upaya pencapaian tujuan dengan memanfaatkan sumber daya manusia, informasi, sistem dan sumber dana dengan tetap memperhatikan fungsi manajemen, peran dan keahlian[2]. Dari pengertian ini, ditekankan bahwa untuk mencapai tujuan, diperlukan sumber daya manusia, dan sumber-sumber non-manusia yang berupa sumber dana, teknik atau sistem, fisik, perlengkapan, informasi, ide atau gagasan, dan teknologi. Elemen-elemen tersebut dikelola melalui proses manajemen yang meliputi perencanaan, pengorganisasian, kepemimpinan, dan pengendalian, yang diharapkan mampu menghasilkan produk berupa barang atau jasa yang dapat dimanfaatkan oleh masyarakat pengguna.

2.2 Konsep Umum Perolehan Informasi

Gordon B. Davis (1999) menyatakan bahwa informasi adalah data yang dapat diolah menjadi suatu bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau mendatang[3]. Hasugian (2006) mengemukakan bahwa *Information Retrieval System* (Sistem Temu Balik Informasi) merupakan suatu proses untuk mengidentifikasi, kemudian memanggil (*retrieve*) suatu dokumen dari simpanan (*file*), sebagai jawaban atas permintaan informasi[4]. Salton (1983) menjelaskan bahwa secara sederhana perolehan sistem informasi merupakan suatu sistem yang menyimpan informasi dan memperoleh kembali informasi tersebut[5].



Gambar 1 : Proses Information Retrieval

2.3 Pemrofilan Text

a. Konsep Pemrofilan Teks

Sebastiani (2002) menyatakan bahwa pemrofilan teks adalah sebuah pekerjaan dari bahasa dokumen yang alamiah untuk kategorisasi standar sesuai dengan kontennya. Pada penelitian ini metode yang digunakan adalah metode *n-gram*. Dimana teks tersebut akan dibaca kata per kata[6].

b. Representasi Profil

Pada *Vector Space Model* setiap dokumen di dalam *database* dan *query* pengguna direpresentasikan oleh suatu vektor *multi-dimensi* sesuai dengan jumlah *term* dalam dokumen yang terlibat. Pada model ini :

- 1) *Vocabulary*, kumpulan *term* yang membentuk suatu ruang vektor,
- 2) Setiap *term* di dalam dokumen diberikan bobot *real* bernilai w_{ij} ,
- 3) Dokumen dan *query* diekspresikan sebagai vektor t dimensi $d_j=(W_1, W_2, \dots, W_{ij})$ dan terdapat n dokumen didalam koneksi, yaitu $j = 1, 2, \dots, n$.

	T_1	T_2	T_3	T_n
D_1	w_{11}	w_{12}	w_{13}	w_{1n}
D_2	w_{21}	w_{22}	w_{23}	w_{2n}
D_3	w_{31}	w_{32}	w_{33}	w_{3n}
...
D_m	w_{m1}	w_{m2}	w_{m3}	w_{mn}
Q	b_1	b_2	b_3	b_n

Gambar 2 : Representasi Kata Matriks-Dokumen

2.4 Pencocokan Profil Text

A. Konsep Pencocokan Profil

Pada model ruang *vector*, prinsip dasar *vector* model adalah:

- 1) Dokumen dipresentasikan dengan menggunakan *keyterm*,
- 2) Ruang dimensi ditentukan oleh *keyterm*,
- 3) *Query* dipresentasikan dengan menggunakan *vector keyterm*,
- 4) Kesamaan *document-keyterm* dihitung berdasarkan jarak *vector*.

B. Fungsi Kesetaraan

Jika *similarity* dokumen yang diinputkan mendekati *similarity* dokumen yang telah ditetapkan maka dokumen tersebut adalah dokumen yang cocok untuk masuk kedalam salah satu kategori yang paling mendekati *similarity* dari dokumen yang berada dalam *database*. Formula *Ochiai* dimana nilai bobot seluruh term yang telah diperoleh akan dikalkulasikan dengan nilai seluruh query yang juga telah dihitung sebelumnya, sebagai berikut:

$$S_{OCHIAI} = \frac{a}{\sqrt{(a+b)(a+c)}}$$

Keterangan:

Sim (Q, D) = ukuran kesamaan (*similarity*) antara *query input* dokumen

- a = jumlah nilai bobot dengan kondisi *term query* bernilai 1 dan *term* dokumen bernilai 1
- b = jumlah nilai bobot dengan kondisi *term query* bernilai 1 dan *term* dokumen bernilai 0
- c = jumlah nilai bobot dengan kondisi *term query* bernilai 0 dan *term* dokumen bernilai 1

C. Pengukuran Keefektifan Perolehan Informasi

1) Pengujian Ketepatan (*Precision*)

Precision adalah suatu tingkat ketepatan hasil terhadap dokumen yang *relevan* dengan dokumen yang ditemukan. Rumus untuk menghitung *precision* adalah :

$$Precision = \frac{\text{Jumlah dokumen relevan yang terambil}}{\text{Jumlah seluruh dokumen yang terambil dalam pencarian}}$$

2) Kelengkapan (*Recall*)

Recall adalah suatu tingkat keberhasilan untuk mengenali suatu dokumen dari seluruh dokumen yang seharusnya dikenali. Rumus untuk menghitung *recall* adalah:

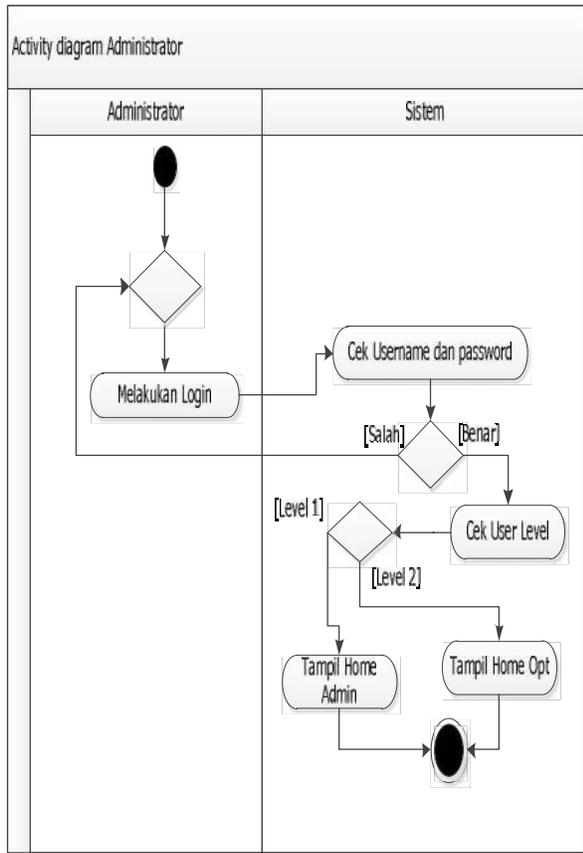
$$Recall = \frac{\text{Jumlah dokumen relevan yang terambil}}{\text{Jumlah dokumen relevan yang ada dalam database}}$$

2.5 Metodologi Pengembangan Sistem

Rekayasa perangkat lunak adalah disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal spesifikasi sistem sampai pemeliharaan sistem setelah digunakan.

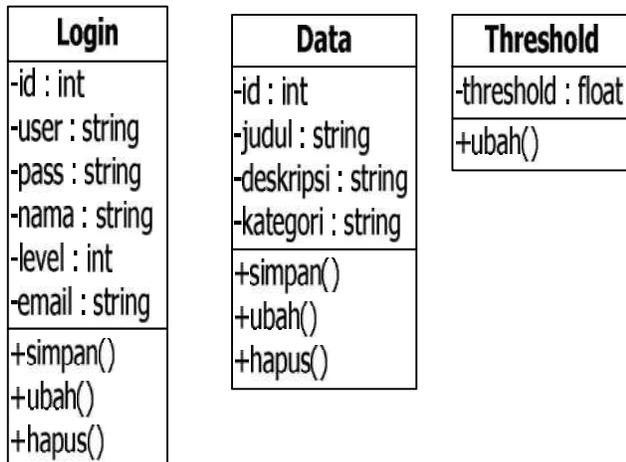
III. RANCANGAN SISTEM DAN APLIKASI

3.1 Aktifitas



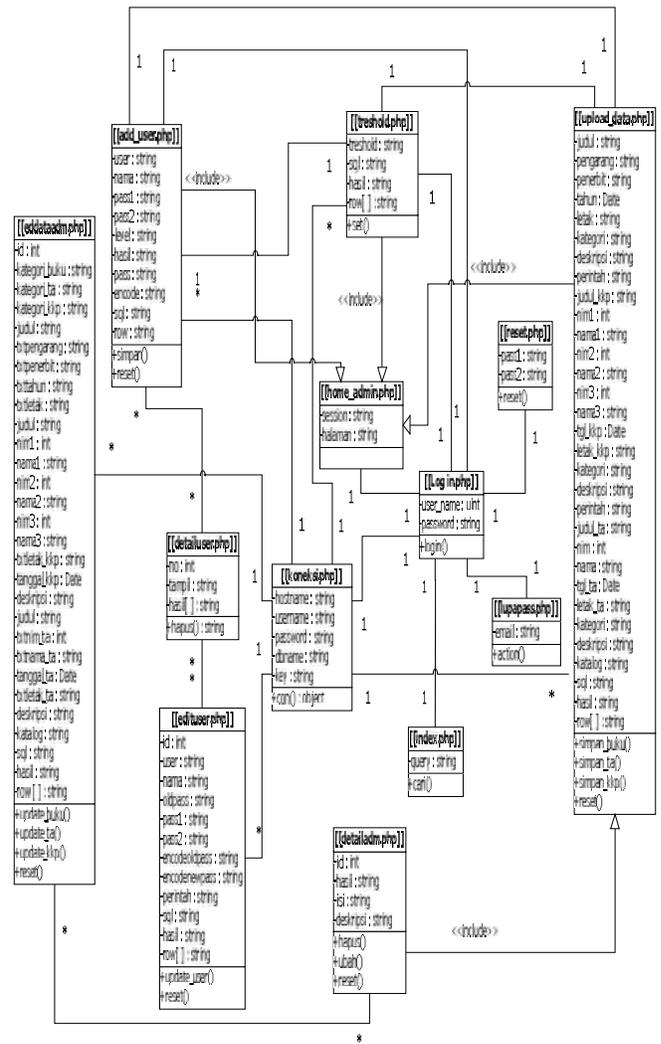
Gambar 3 : Aktifitas Login

3.2 Class Diagram



Gambar 4 : Class Diagram

3.3 Interaksi Komponen



Gambar 5 : Interaksi Komponen Administrator

Tabel 1 : Algoritma Method Index

1.	Ekstrak hasil input menjadi digram
2.	Memasukkan digram kedalam array
3.	Ambil data dari database
4.	If record != end of record
5.	Ekstrak data dari database menjadi digram Then
6.	Memasukkan array ke dalam array Then
7.	Normalisasi array input dan database Then
8.	Pembobotan array Then
9.	Identifikasi nilai a, b, c Then
10.	Hitung simillarity Then
11.	Ambil nilai threshold dari database Then
12.	Bandingkan hasil simillarity dengan threshold Then
13.	Tampil hasil search Then
14.	Kembali ke ambil data dari database
15.	Else
16.	Selesai
17.	End if

IV. HASIL DAN PEMBAHASAN

4.1. Implementasi

Pada bab ini menjelaskan tentang bagaimana mengimplementasikan dan analisa program pembelajaran implementasi sebagai mesin pencari menggunakan sistem aplikasi pencarian dokumen berbasis judul dokumen dengan penerapan profil *digram* pada *vector space model* dengan pembobotan biner serta kesetaraan *Ochiai*.

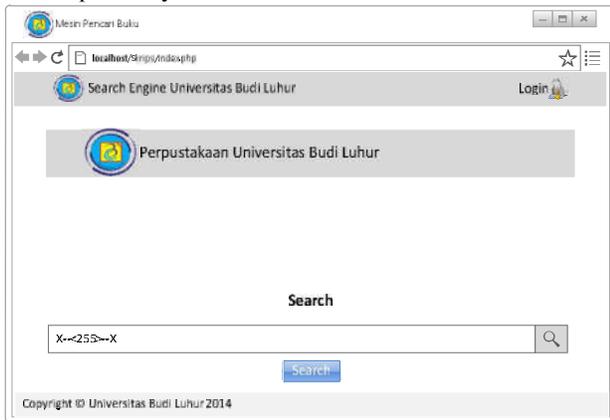
a. Spesifikasi perangkat keras minimum

Prosesor Intel Dual-Core yang terpasang prosesor grafik Intel GMA dengan kapasitas RAM 1 GB dan penyimpanan sebesar 250 GB.

b. Metode pengukuran

- 1) Pengujian Ketepatan (Precision)
- 2) Pengujian Kelengkapan (Recall)

c. Tampilan Layar



Gambar 10 : Tampilan Layar Form Index

4.2. Hasil Uji Coba

a. Pengukuran kinerja Aplikasi

Pengukuran kinerja aplikasi dilakukan untuk melakukan evaluasi tingkat keberhasilan terhadap teknik yang digunakan. Dalam hal ini sudah dipersiapkan sebanyak 30 judul dokumen, terdiri dari 10 judul buku, 10 judul KKP, dan 10 judul tugas akhir yang nantinya akan dicari berdasarkan judul yang *diinput* pada *textbox* pencarian.

Query yang dimasukkan dalam uji coba program antara

lain :

- 1) Instalasi Jaringan Komputer
- 2) Belajar *Administrasi Database*
- 3) Aplikasi Jaringan Komputer

b. Hasil Uji Coba

1) Precision

Dengan pengukuran secara precision didapatkan hasil sebagai berikut :

- a) Instalasi Jaringan Komputer
 $6/7 = 0,86 = 86\%$
- b) Belajar administrasi database mysql
 $4/4 = 1 = 100\%$
- c) Aplikasi jaringan komputer
 $3/4 = 0,75 = 75\%$

2) Recall

- a) Instalasi jaringan komputer
 $6/10 = 0,6 = 60\%$
- b) Belajar administrasi database mysql
 $4/10 = 0,4 = 40\%$
- c) Aplikasi jaringan komputer
 $3/10 = 0,3 = 30\%$

c. Evaluasi Program

Evaluasi program merupakan salah satu hal yang perlu dilakukan dalam setiap pengembangan aplikasi untuk menganalisa dan mengetahui hasil yang telah dicapai oleh aplikasi yang telah dibuat dalam penelitian ini.

Dalam aplikasi ini ditemukan beberapa kelebihan dan kekurangan antara lain :

1) Kelebihan Program

- a) Mampu mencari teks yang tidak terstruktur.
- b) Dapat memunculkan dokumen yang dicari *user* sesuai dengan *query* yang dimasukkan meskipun terdapat kesalahan dalam pengetikan judul.
- c) Dapat menampilkan judul-judul dokumen lain yang berkaitan dengan *query* yang dimasukkan.

2) Kekurangan Program

- a) Dari segi ketepatan didapatkan nilai rata-rata 1 yang berarti bahwa fungsi kesetaraan yang digunakan memiliki nilai similarity yang tinggi.
- b) Dari segi kelengkapan program ini belum memadai karena hanya terdapat 30 record judul buku.
- c) Waktu tanggap pencarian dokumen yang kurang responsif saat dibandingkan dengan menggunakan perintah SQL '%LIKE%'.

V. PENUTUP

5.1. Kesimpulan

Kesimpulan yang diambil dari aplikasi ini adalah sebagai berikut :

- a. Semakin tinggi nilai dari *threshold* yang ditetapkan maka nilai dari *precision* (ketepatan) akan semakin tinggi, tetapi nilai dari *recall* (kelengkapan) akan semakin rendah.
- b. Penggunaan pemprofilan *text* menggunakan *digram* dapat memberikan kemudahan bagi *user* yaitu *query* data yang ditulis oleh *user* tidak mesti sama persis dengan *query* dokumen yang dicari.
- c. Keluaran akhir dari aplikasi ini berupa daftar data dari buku/KKP/TA yang direkomendasikan karena memiliki kemiripan dengan *query* yang di-input oleh *user*.

5.2. Saran

Perlu dilakukan juga percobaan dengan mencari alternatif pemprofilan *text* lain seperti *uni-gram*, *bi-gram*, *quad-gram* dan *quint-gram* untuk mengetahui kinerja yang lebih baik.

Demikian kesimpulan dan saran ini penulis sampaikan, semoga sistem yang penulis buat dapat bermanfaat dan memberikan kemudahan dalam proses pencarian buku di Perpustakaan Universitas Budi Luhur.

DAFTAR PUSTAKA

- [1] Jogiyanto, H.M. 2005, *Analisis dan Desain Sistem Informasi*, Yogyakarta, Andi.
- [2] Lasa, H.S. 2007, *Manajemen Perpustakaan Sekolah*, Yogyakarta, Andi.
- [3] Davis, Gordon B. 1999, *Kerangka Dasar Sistem Informasi Manajemen, alih bahasa Andreas S. Adiwardana*, Jakarta, PT. Ikrar Mandiri Abadi.
- [4] Hasugian, Jonner 2006, *Penggunaan Bahasa Ilmiah dan Kosa Kata Terkendali dalam Sistem Temu Balik Informasi Berbasis Teks*. Pustaka: (Jurnal Studi Perpustakaan dan Informasi) ; Vol.II, No.2. p.72-75.
- [5] Salton, G., McGill, Michael J. 1983, *Introduction To Modern Information Retrieval*, New York, McGraw-Hill.
- [6] Sebastiani, F. 2002, *Machine Learning In Automated Text*