

RANCANG BANGUN SISTEM INFORMASI HADITS MENGUNAKAN TEKNIK TEMU KEMBALI INFORMASI MODEL RUANG VEKTOR

Nesdi E. Rozanda, Arif Marsal, Kiki Iswanti

Jurusan Sistem Informasi, Fakultas Sains dan Teknologi,
Universitas Islam Negeri Sultan Syarif Kasim Riau

Jl.HR.Soebratas KM.15 no 155 Simpang Baru- Panam Pekanbaru - Riau

e-mail : nesdi.er@uin-suska.ac.id, nesdiev@yahoo.com, arif.marsal@gmail.com,
kiki.iswanti@yahoo.com

ABSTRAK

Paper ini membahas tentang cara lain yang bisa digunakan untuk mendapatkan informasi hadits yaitu dengan menggunakan suatu sistem aplikasi yang menggunakan teknik Temu Kembali Informasi (Information Retrieval) dengan Model Ruang Vektor. Model digunakan untuk mencari kemiripan antara kata kunci yang dimasukkan dengan dokumen yang tersimpan didalam basis data. Basis data terdiri dari dokumen hadits berjumlah 536 dokumen. Sistem yang dihasilkan dapat mempercepat proses pencarian dokumen hadits, mempermudah mendapatkan dokumen dan dapat menemukan dokumen hadits atau informasi yang relevan.

Kata Kunci : aplikasi, hadits, Temu Kembali Informasi, relevan

ABSTRACT

This paper discusses the different ways that can be used to obtain information hadith by using an application system that uses Vector Space Model in Information Retrieval (IR) technique. The model is used to find the similarity between the keywords with the documents in the database. The database consists of 536 hadiths documents. The application system was produced was indicated that give increasing speed on process to search, retrieve and find the relevant documents

Keyword: application, hadith, information retrieval, relevant

PENDAHULUAN

Saat ini, komputer merupakan salah satu teknologi yang dapat membantu pekerjaan manusia seperti dalam bidang perniagaan, perkantoran, dan pendidikan. Komputer tidak hanya digunakan sebagai alat mengetik atau untuk menghitung saja, akan tetapi dapat menghasilkan suatu informasi yang bermanfaat bagi orang yang memerlukannya.

Manfaat dari sistem informasi yang dapat dirasakan saat ini seperti sistem pencarian (*search engine*). *The American Heritage Dictionary* mendefinisikan *search engine* sebagai sebuah program perangkat lunak (*software*) yang menelusur, menjanging, dan menampilkan informasi dari pangkalan data (Anam, 2008). Contoh dari sistem pencari yang sering digunakan yaitu OPAC (*Online Public Access Catalogue*), google, dan yahoo.

Penggunaan mesin pencari dapat juga digunakan dalam bidang keislaman, misalnya pencarian ayat al-qur'an, pencarian do'a sehari-hari, dan sistem pencarian hadits.

Hadits diperlukan sebagai pedoman hidup dalam ajaran umat muslim. Hadits menurut istilah ahli hadits adalah apa yang disandarkan kepada Nabi Shalallahu 'alaihi wa sallam, baik berupa ucapan, perbuatan, penetapan, sifat, atau sirah beliau, baik sebelum kenabian atau sesudahnya. Hadits diklasifikasikan menjadi tiga bagian, yaitu hadits *shahih*, hadits *hasan*, dan hadits *dhaif*.

Ada beberapa cara yang bisa digunakan untuk dapat memperoleh informasi hadits yaitu dengan bertanya kepada orang, mencari dalam buku, dan pencarian menggunakan sistem. Dalam mendapatkan informasi hadits tersebut, terkadang kita mengalami kendala yaitu jika

bertanya kepada orang, tetapi tidak semua orang dapat menghafal hadits, walaupun ada orang yang bisa menghafal semua hadits, akan tetapi sulit untuk menemuinya karena jarak dan waktu yang terbatas. Hadits bisa didapat dalam buku atau kitab, tapi tidak semua buku tersedia dipustakaannya maupun dipasaran kita harus membelinya dan akan mengeluarkan biaya. Cara ketiga dalam mencari hadits adalah menggunakan sistem. Salah satu aplikasi yang dapat mengakses hadits digital terdapat di situs <http://id.lidwa.com/app/>, merupakan software kitab hadits digital online terjemah Indonesia yang cukup lengkap, namun pada aplikasi ini menyediakan strategi pencarian *query* berbasis kata (*word-matching*) yang menampilkan daftar hasil (*search result*) ayat berdasarkan faktor sinonim antara *query* dan terjemahan hadits dalam *database* yang mengandung kata yang ada dalam *query*. Hal ini menyebabkan adanya hadits relevan dalam koleksi yang tidak dipanggil karena tidak memuat kata yang ada dalam *query*.

Untuk mengatasi hal tersebut, maka dengan salah satu cara yang cepat dan tepat sesuai menurut kriteria yang dicari, dapat digunakan sistem informasi hadits berbasis teknik Temu Kembali Informasi (*Information Retrieval*).

Temu kembali Informasi merupakan teknik yang digunakan dalam pencarian atau penyelusuran informasi yang lebih relevan. Dalam teknik Temu Kembali Informasi, terdapat beberapa pemodelan yang biasa digunakan, yaitu model *boolean* dan *extended boolean*, model ruang vektor, dan model *probabilistic inference*. Dalam pencarian hadits, Temu Kembali Informasi digunakan untuk menampilkan beberapa hadits yang berhubungan kata kunci yang dicari sesuai dengan kriteria tertentu. Namun, sistem yang didapat tidak menampilkan hasil berdasarkan ranking dari kata kunci yang dimasukkan.

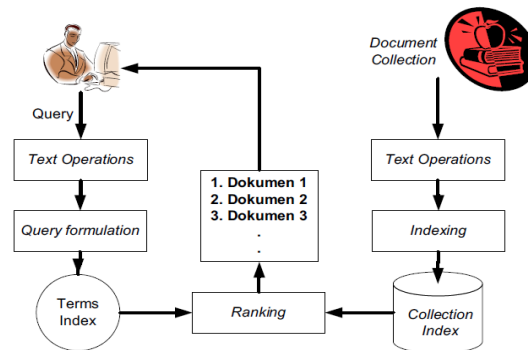
Berdasarkan hal tersebut, maka dilakukan penelitian dengan membuat suatu sistem informasi hadits menggunakan teknik temu kembali informasi dengan pemodelan ruang vektor. Pemodelan ini digunakan untuk menemukan kemiripan antara dokumen yang tersimpan didalam *database* dengan kata kunci yang dimasukkan, sehingga dapat

mempermudah pengguna sistem untuk menemukan dokumen yang dicari.

LANDASAN TEORI

Information retrieval (Temu Kembali Informasi)

Information retrieval (IR) *system* digunakan untuk menemukan kembali (*retrieve*) informasi-informasi yang relevan terhadap kebutuhan pengguna dari suatu kumpulan informasi secara otomatis (Bunyamin, 2005). Salah satu aplikasi umum dari IR *system* adalah *search engine* atau mesin pencarian yang terdapat pada jaringan internet. Sebagai suatu sistem, IR *system* memiliki beberapa bagian yang membangun sistem secara keseluruhan. Bagian-bagian yang terdapat pada IR *system* digambarkan pada Gambar 1 :



Gambar 1. Ilustrasi Temu Kembali Informasi

(Sumber: Bunyamin, Jurnal Informatika UKM, Vol. I, 2005)

Alur pertama dimulai dari koleksi dokumen dan alur kedua dimulai dari *query* pengguna. Alur pertama yaitu pemrosesan terhadap koleksi dokumen menjadi basis data indeks tidak tergantung pada alur kedua. Sedangkan alur kedua tergantung dari keberadaan basis data tergantung dari keberadaan basis data indeks yang dihasilkan pada alur pertama.

Bagian-bagian dari IR *system* menurut gambar 2 meliputi:

- 1) *Text Operations* (operasi terhadap teks) yang meliputi pemilihan kata-kata dalam *query* maupun dokumen (*term selection*) dalam pentransformasian dokumen atau *query* menjadi *term index* (indeks dari kata-kata).

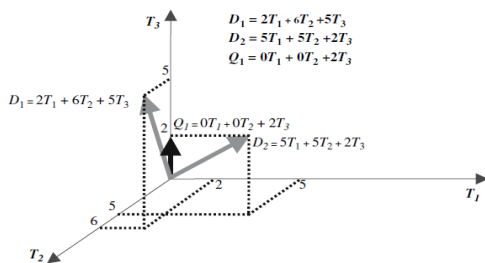
- 2) *Query formulation* (formulasi terhadap *query*) yaitu memberi bobot pada indeks kata-kata *query*.
- 3) *Ranking* (perangkingan), mencari dokumen-dokumen yang relevan terhadap *query* dan mengurutkan dokumen tersebut berdasarkan kesesuaiannya dengan *query*.
- 4) *Indexing* (pengindeksan), membangun basis data indeks dari koleksi dokumen. Dilakukan terlebih dahulu sebelum pencarian dokumen dilakukan.

Model Ruang Vektor

Model ruang vektor adalah model sistem temu balik informasi yang mengibaratkan masing-masing *query* dan dokumen sebagai sebuah vektor n-dimensi (Agus, 2011). Model ini menentukan kemiripan (*similarity*) antara dokumen dengan *query* dengan cara merepresentasikan dokumen dan *query* masing-masing ke dalam bentuk vektor. Pada model ruang vektor, setiap dokumen di dalam database dan *query* pengguna direpresentasikan oleh suatu vektor multi-dimensi. Dimensi yang sesuai dengan jumlah term pada dokumen yang terlibat pada model ini adalah :

- 1) *Vocabulary* merupakan kumpulan semua *term* berbeda yang tersisa dari dokumen setelah *preprocessing* dan mengandung *t term index*. *Term-term* ini membentuk suatu ruang vektor.
- 2) Setiap *term i* di dalam dokumen atau *query j*, diberikan suatu bobot (*weight*) bernilai *real* W_{ij} .
- 3) Dokumen dan *query* diekspresikan sebagai vektor *t* dimensi $d_j = (W_1, W_2, \dots, W_{ij})$ dan terdapat *n* dokumen di dalam koleksi, yaitu $j = 1, 2, \dots, n$.

Contoh dari perhitungan kemiripan yaitu :



Gambar 2. Contoh Model Ruang Vektor dengan dua dokumen D_1 dan D_2 , serta *query* Q_1 (Sumber : Cios, 2007)

Penentuan relevansi dokumen dengan *query* dipandang sebagai pengukuran kesamaan (*similarity measure*) antara vektor dokumen dengan vektor *query*. Semakin “sama” suatu vektor dokumen dengan vektor *query* maka dokumen dapat dipandang semakin relevan dengan *query* (Bunyamin, 2005).

Keberhasilan dari model ruang vektor ini ditentukan oleh skema pembobotan terhadap suatu *term* baik untuk cakupan lokal maupun global, dan faktor normalisasi. Pembobotan lokal hanya berpedoman pada frekuensi munculnya *term* dalam suatu dokumen dan tidak melihat kemunculan *term* tersebut di dalam dokumen lainnya. Faktor yang memegang peranan penting dalam pembobotan kata, yaitu :

- 1) *Term Frequency (tf)*

Pendekatan dalam pembobotan lokal yang paling banyak diterapkan adalah *term frequency (tf)*. Faktor ini menyatakan banyaknya kemunculan suatu kata dalam suatu dokumen. Semakin sering suatu kata muncul dalam sebuah dokumen, berarti semakin penting kata tersebut.

- 2) *Inverse Document Frequency (idf)*

Pembobotan global digunakan untuk memberikan tekanan terhadap *term* yang mengakibatkan perbedaan dan berdasarkan pada penyebaran dari *term* tertentu di seluruh dokumen. Banyak skema didasarkan pada pertimbangan bahwa semakin jarang suatu *term* muncul di dalam total koleksi maka *term* tersebut menjadi semakin berbeda. Bobot global dari suatu *term i* pada pendekatan *inverse document frequency (idf)* dapat dirumuskan sebagai berikut :

$$Idf(i) = \log \left(\frac{N}{Df(i)} \right)$$

Dimana :

N = jumlah artikel dalam koleksi dokumen

$Df(i)$ = jumlah dokumen koleksi yang mengandung *term i*

Bobot *term i* di dalam *Information Retrieval System* (W_{ij}) dihitung menggunakan *tf-idf* yaitu :

$$W_{i,j} = tf_{i,j} \times idf_i$$

- 3) Normalisasi panjang dokumen
Untuk mengurangi pengaruh perbedaan panjang dokumen, pada pembobotan kata digunakan satu faktor lagi yang disebut sebagai normalisasi panjang dokumen. Normalisasi yang digunakan adalah normalisasi kosinus. Rumus normalisasi kosinus yaitu :

1. Normalisasi panjang *term*, merupakan jumlah perbandingan antara bobot *term* terhadap panjang *term*.

$$\bar{w}_{i,j} = \frac{w_{i,j}}{\sqrt{\sum_{i=1}^t w_{i,j}^2}}$$

2. Pembobotan masing-masing *query*

$$w_{i,q} = \left(0.5 + \frac{0.5 \text{freq}_{i,q}}{\text{maxtfraq}_{i,q}} \right) \times \text{idf}(i)$$

3. Normalisasi *query*, merupakan jumlah perbandingan antara bobot *query* terhadap panjang *query*.

$$\bar{w}_{i,q} = \frac{w_{i,q}}{\sqrt{\sum_{i=1}^t w_{i,q}^2}}$$

$$\text{sim} \left(\begin{matrix} \rightarrow \\ \vec{d}_j \end{matrix} \rightarrow \begin{matrix} \rightarrow \\ \vec{q} \end{matrix} \right) = \frac{\vec{d}_j \cdot \vec{q}}{|\vec{d}_j| \cdot |\vec{q}|} = \frac{\sum_{i=1}^t (w_{i,j} \cdot w_{i,q})}{\sqrt{\sum_{i=1}^t w_{i,j}^2} \cdot \sqrt{\sum_{i=1}^t w_{i,q}^2}}$$

Dimana:

d_j = bobot dokumen

q = bobot *query*

$|d_j|$ = panjang dokumen

$|q|$ = panjang *query*

Metode pengukuran kesesuaian ini memiliki beberapa keuntungan, yaitu adanya normalisasi terhadap panjang dokumen. Hal ini memperkecil pengaruh panjang dokumen.

Precision dan Recall

Untuk mengukur kualitas dari dokumen yang terambil maka diperlukan perhitungan penilaian dengan cara *precision* dan *recall*.

- 1) *Precision* merupakan mengevaluasi kemampuan sistem IR untuk menemukan kembali *top-ranked* yang paling relevan, dan didefinisikan sebagai presentase dokumen yang di-*retrieve* yang benar – benar relevan terhadap *query* pengguna.

$$\text{Precision} = \frac{\text{relevant document}}{\text{retrieved document}}$$

- 2) Sedangkan *Recall* adalah mengevaluasi kemampuan sistem IR untuk menemukan semua *item* yang relevan dari dalam koleksi dokumen dan didefinisikan sebagai

presentase dokumen yang relevan terhadap *query* pengguna dan yang diterima.

$$\text{Recall} = \frac{\text{relevant document}}{x}$$

Dimana: x = dokumen yang relevan yang terambil

Retrieved_document = rangking dari dokumen yang terambil

Relevant_document = jumlah dokumen relevan

METODOLOGI

Langkah-langkah untuk membuat Sistem Informasi Hadits Menggunakan Temu Kembali Informasi Model Ruang Vektor digunakan dengan model prototipe. Berikut langkah-langkah dari proses pemodelan prototipe:



Gambar 3. Metodologi Model Prototipe

Dari gambar 3, maka dapat dijelaskan bahwa:

1. Pengumpulan Kebutuhan

Kebutuhan yang diperlukan dalam pembuatan sistem ini adalah pengumpulan dokumen-dokumen hadits yang akan diolah nantinya. Dokumen-dokumen yang akan diolah merupakan dokumen yang *download* di internet. Dan kemudian menentukan kebutuhan modul-modul apa saja yang dipergunakan. Modul-modul yang digunakan adalah Beranda/ halaman utama, Pencarian, *About*, *Upload* Hadits, Daftar (Kamus, *Tokenlist*, *Stoplist*, *Stemming*, dan Hasil *Upload*), *Indexing*, *Retrieval*, *Precision* dan *Recall*.

2. Perancangan Sistem

Setelah kebutuhan sudah ditentukan, maka dilakukan perancangan Sistem Informasi Hadits. Perancangan ini dilakukan dengan membuat alur kerja sistem menggunakan Diagram UML, merancang antarmuka sistem menggunakan *Microsof Visio*, dan membuat sistem menggunakan bahasa pemrograman PHP.

3. Evaluasi Sistem

Setelah sistem sudah dirancang dan bisa digunakan, maka dilakukan evaluasi yaitu uji coba dari sistem tersebut. Uji coba sistem ini dilakukan oleh administrator dan pengguna. Evaluasi ini juga menguji bagaimana kinerja sistem dengan menggunakan *precision* dan *recall*.

HASIL

Analisis Sistem Usulan

Proses IR, terbagi dalam 2 proses, yaitu proses *indexing* dan proses perangkaian. Untuk proses *indexing*, terdapat beberapa tahap yang perlu dilakukan, yaitu

a. Tokenisasi dokumen

Tokenisasi merupakan tahap untuk memisahkan kata-kata dari kalimat dan menghilangkan tanda baca, seperti tanda koma, tanda titik, tanda petik dan karakter lainnya, serta mengubah semua *token* bentuk huruf kecil (*lower case*).

b. Penghapusan *Stoplist*

Proses *stoplist* merupakan proses pembuangan kata penghubung atau kata dalam bahasa indonesia, seperti dan, maka, lalu, dalam, yang, akan, dari, pada, itu, telah, dengan, dan sebagainya. *Stoplist* adalah proses yang dilakukan setelah proses tokenisasi. Dimana proses ini hanya menyaring kata/*term* yang dianggap penting. Tiap element yang terdapat pada *tokenTerm* dicocokkan dengan daftar *stoplist* di *database* yang menyimpan kumpulan kata yang dianggap tidak penting (*stopword*), yang disebut kamus kata.

c. *Stemming*

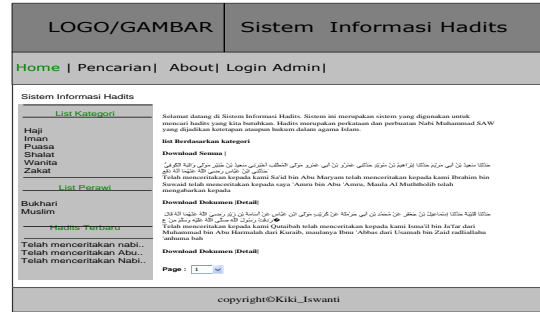
Proses *stemming* ini dilakukan untuk mengubah kata yang memiliki kata awalan dan kata akhiran (imbuhan) menjadi bentuk kata dasar. Dari dokumen yang diambil, pencocokan kata dilakukan menggunakan kamus *stemming* apakah kata yang diambil sudah cocok atau belum.

d. *Weighting*

Weighting adalah proses pembobotan *term* dengan melihat frekuensi kemunculan kata pada setiap dokumen. Pembobotan ini dilakukan menggunakan perhitungan *tf*, *df*, *idf*, dan normalisasi kata.

Perancangan Sistem

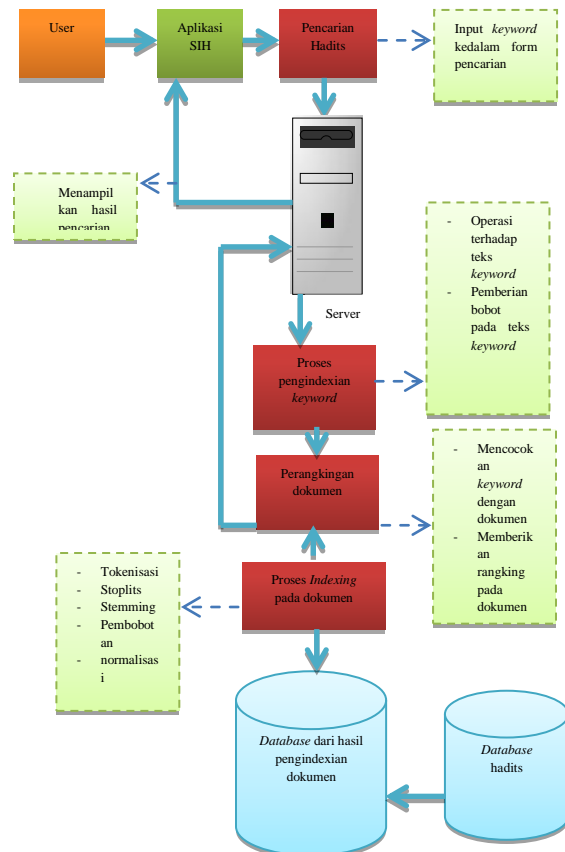
1. Antarmuka Sistem



Gambar 5. Antarmuka Halaman Utama

Modul utama ini digunakan pada halaman utama pada aplikasi sistem informasi hadits ini. Modul utama ini berisi menu home atau halaman utama, menu pencarian digunakan untuk menginputkan kata kunci, *about* digunakan untuk informasi tentang aplikasi ini, dan *login admin* hanya digunakan untuk admin saja sebagai pengolah dokumen. Dan isi dari halaman utama ini berupa informasi-informasi tentang hadits.

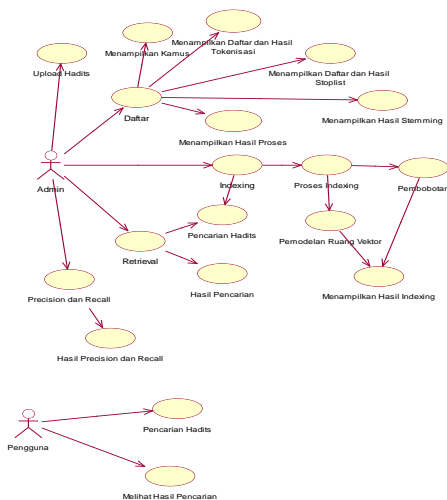
2. Arsitektur Sistem Usulan



Gambar 6. Arsitektur Sistem usulan

Berdasarkan gambar 6 dijelaskan *user* (pengguna) menggunakan aplikasi SIH dalam pencarian dokumen hadits. Setelah *user* menginputkan *keyword* atau *query*, maka *server* akan merespon dan memproses *query* yang sudah di *inputkan*. Kemudian *server* mengoperasikan *query* tersebut. Dalam proses pengoperasian *query* tersebut, terdapat proses pengindexian *query* yaitu mengoperasikan teks yang ada pada *query* dan memberikan bobot pada *query*. Lalu, untuk memanggil dokumen yang akan didapatkan, sebelumnya terdapat *database* dari hasil *indexing* dari dokumen yang sudah diproses dengan menggunakan teknik temu kembali informasi yang melalui tokenisasi, stoplist, stemming, pembobotan dan normalisasi. Setelah itu, hasil dari pengindexian dokumen diberi rangking dan nilai kemiripan dengan *query*. Dari hasil perangkingan tersebut kemudian diterima oleh *server* dan *server* memberikan kepada *user*. Dan *user* dapat melihat hasil dari pencarian didalam aplikasi SIH.

3. Usecase Diagram



Gambar 7. Usecase diagram aplikasi

Pada gambar Usecase diagram terdapat dua aktor, yaitu *admin* dan pengguna. Dua aktor ini berperan sebagai objek yang menggunakan aplikasi. Administrator berperan sebagai mengontrol aplikasi sistem informasi hadits dengan mengakses semua fasilitas yang ada dalam aplikasi. Sedangkan pengguna merupakan objek yang berperan sebagai pengguna secara umum dari proses melakukan pencarian hadits. Berikut penjelasan dari setiap *usecase* tersebut:

- a) *Upload* hadits
Pada proses ini, administrator melakukan proses penginputan dokumen yang akan di olah. Dan diproses dengan menggunakan tokenisasi, *stoplist*, dan *stemming* yang akan di simpan kedalam corpus hadits.
- b) *Daftar*
Pada *usecase* proses ini administrator dapat menampilkan daftar kamus, daftar tandabaca dan menampilkan hasil dari proses tokenisasi, menampilkan daftar *stopword* dan hasil dari *stoplist*, menampilkan hasil proses *stemming*, serta menampilkan seluruh dokumen yang telah diproses dari *tokenlist* sampai ke *stemming*.
- c) *Indexing*
Usecase indexing terdapat proses pembobotan *term* dan menormalisasikan *term* terhadap dokumen sehingga akan dapat dijumpai mana dokumen yang sangat mirip dengan kata kunci yang dimasukkan.
- d) *Retrieval*
Proses ini dilakukan untuk menguji aplikasi dengan cara memasukkan kata kunci kedalam form yang sudah disediakan dan kemudian akan menampilkan hasil dari pencarian tersebut.
- e) *Pencarian* hadits
Pengguna melakukan pencarian hadits dengan aplikasi sistem informasi hadits secara umum dengan memasukkan kata kunci yang diinginkan.
- f) *Melihat* hasil pencarian
Pengguna melihat secara detail dari hasil pencarian hadits yang telah dilakukan.

Uji Coba dan Evaluasi Kinerja Sistem

Uji coba untuk mengukur kemampuan sistem dilakukan dengan 7 *query* (kata kunci) dari masing-masing kategori. Dimana hasil dari pencarian tersebut dievaluasi dengan perhitungan *Precision* dan *Recall*. Berikut daftar *query* yang digunakan dalam uji coba:

Tabel 1. Daftar *Query*

No	Kasus	Quey
1	a. 1 Kata dasar	1. iman
	b. 2 Kata dasar	2. haji mabrur
		3. puasa ramadhan

		4. wanita shalihah
2	kata mengandung tokenlits/ tandabaca	5. zakat fitrah, puasa
3	kata mengandung stopword/stoplits	6. mahar dan hukum nikah
4	Kata mengandung imbuhan	7. makanan yang dihalkalkan dan diharamkan

Dari tabel *query*, maka di ambil salah satu contoh dari *query* tersebut yaitu pada *query* Iman. Maka pada tabel 2, terdapat jumlah rank 23, jumlah dokumen yang relevan terdapat 14 dokumen, maka didapatkan *precision* rata-rata 0,73 dan *recall* 1.

Tabel 2. Hasil *Precision* dan *Recall*

Ran k	Id_ dok	Dok ditem ukan	Relevan (ya/ tidak)	<i>Precision</i> (dok_ ditemuka n / Rank)	<i>Recall</i> (dok ditemuk an / Jumlah Relevan)
1	151	1	Ya	1	0.0714286
2	162	0	Tidak	0	0
3	362	2	Ya	0.666667	0.142857
4	360	3	Ya	0.75	0.214286
5	164	4	Ya	0.8	0.285714
6	368	0	Tidak	0	0
7	156	5	Ya	0.714286	0.357143
8	232	6	Ya	0.75	0.428571
9	366	7	Ya	0.777778	0,5
10	382	0	Tidak	0	0
11	155	8	Ya	0.727273	0.571429
12	6	0	Tidak	0	0
13	376	9	Ya	0.692308	0.642857
14	359	10	Ya	0.714286	0.714286
15	165	11	Ya	0.733333	0.785714
16	222	0	Tidak	0	0
17	233	0	Tidak	0	0
18	375	0	Tidak	0	0
19	361	12	Ya	0.631579	0.857143
20	377	13	Ya	0.65	0.928571
21	447	0	Tidak	0	0
22	55	0	Tidak	0	0
23	192	14	Ya	0.608696	1
				Rata-rata= 0.73	Max= 1

Tabel 3. Hasil Rata- rata dan maximal *Precision* dan *Recall*

Query	Jlh Dok Ditemukan	Jlh dok Relevan	<i>Precision</i> (Rata-rata)	<i>Recall</i> (Maximal)
Haji	23	14	0.73	1
Haji mabrur	39	28	0.71	1
Puasa ramadhan	44	20	0.67	0.909
wanita shalihah	47	21	0.52	0.954
zakat fitrah, puasa	52	30	0.80	0.937
Mahar dan hukum nikah	21	16	0.86	1
Makanan yang dihalkalkan dan diharamkan	48	12	0.30	1
Rata- rata			0.65	0.97

Dari uji coba yang telah dilakukan, maka didapatkan hasil dari 7 *query* yang di inputkan maka didapatkan rata-rata pada *precision* 0,65 dengan keakuratan persentase 65% dan perhitungan rata-rata *recall* terdapat 0,97 dengan keakuratan persentase 97% . Dari perhitungan yang dilakukan maka hampir semua dokumen relevan terambil oleh sistem.

Ada beberapa dokumen yang hilang dari kata kunci yang dimasukkan, yaitu pada Q3, Q4, dan Q. Pada Q3 terdapat 20 dokumen relevan dari 44 dokumen yang terambil dengan kata kunci “puasa ramadhan” dan dokumen yang dianggap relevan ada 2 dokumen yang hilang, sehingga dapat diperhitungkan bahwa dokumen yang relevan terdapat 22 dokumen, maka hasil perhitungan rata-rata dokumen terhadap *precision* 0,67 dan *recall* 0,909 . Begitu juga dengan Q4 dan Q5.

KESIMPULAN

Berdasarkan pembahasan dan pengujian yang telah dilakukan pada Sistem Informasi Hadits, maka didapatkan kesimpulan sebagai berikut :

- 1) sistem sudah termasuk sistem yang efektif dalam pencarian hadits. Berdasarkan hasil uji dari tujuh *query* dengan perhitungan *precision* dan *recall*, maka didapatkan hasil rata-rata *precision* 65% dan menghasilkan *recall* rata-rata 0,97 yaitu hampir semua dokumen yang relevan terambil oleh sistem.
- 2) Sistem Informasi Hadits yang dihasilkan menggunakan Model Ruang Vektor ternyata dapat memudahkan pengguna dalam mencari hadits dan mendapatkan hadits yang diinginkan.

SARAN

Saran yang diberikan untuk menjadikan Sistem Temu Kembali Informasi ini lebih baik, yaitu:

- 1) Database Sistem informasi hadits ini masih belum lengkap dan beragam, oleh karena itu perlu ditambah dan selalu *diupdate* secara berkala.
- 2) Sistem Informasi Hadits ini dapat dikembangkan lagi dengan *fitur-fitur* baru, seperti pencarian ayat Al-qur'an dengan artinya dan pencarian berdasarkan suara.

DAFTAR PUSTAKA

- Adi, Heru Darmawan, dkk** (2011). *Rancang Bangun Aplikasi Search Engine Tafsir Al-Qur'an Menggunakan Teknik Text Mining Dengan Algoritma VSM (Vector Space Model)*. Hal 3. ppta.stikom.edu/upload/upload/file/05410100342makalah.pdf. diakses 20 Mei 2012.
- Agusta, Ledy** (2009). *Perbandingan Algoritma Stemming Porter Dengan Algoritma Nazief & Adriani Untuk Stemming Dokumen Teks Bahasa Indonesia*. 2009. [Online] Available. <http://yudiagusta.files.wordpress.com/2009/11/196-201-knsi09-036-perbandinganalgoritma-stemming-porter-dengan-algoritma-nazief-adriani-untuk-stemming-dokumenteks-bahasa-indonesia.pdf>. diakses 6 Oktober 2012.
- Anam, Syamsul**, (2008). *Analisa Kinerja Search Engine Dengan Menilai Precision And Recall Untuk Informasi Marketing Dan Advertising (Studi Kasus : Google, Yahoo, MSN, Ask)*. Hal 3. Universitas Narotama Surabaya. Surabaya..
- Bunjamin, Hendra**, (2005). *Algoritma Umum Pencarian Informasi Dalam Sistem Temu Kembali Informasi Berbasis Metode Vektorisasi Kata dan Dokumen*. Jurnal Informatika UKM, No. 2, Vol. I, hal 85-87.
- Cios, Krzysztof J, Witold Predrycz, dkk**, (2007). *Data Mining A Knowledge Discovery Approach*. Springer. New York..
- Dwi, Ratih Puspita**, (2009). *Analisis Usability Pada Search Engine Dengan Dokumen Teks Terstruktur*. [Online] Available. http://digilib.itelkom.ac.id/index.php?option=com_content&view=article&id=544:search-engine-&catid=6:internet&Itemid=14. Diakses 10 Mei 2012.
- Hasibuan, Zainal A, Yofi Andri**, (2001). *Penerapan Berbagai Teknik Sistem Temu-Kembali Informasi Berbasis Hiperteks*. Fakultas Ilmu Komputer Universitas Indonesia..
- Irfani, Ahmad**, (2007). *Perancangan Prototipe Sistem Temu Kembali Informasi Menggunakan Algoritma Fuzzy Clustering*. Hal 11 dan 18. Instuti pertanian Bogor. Bogor.
- Jogiyanto**, (2005), *Pengenalan Komputer*. Edisi 5, hal 711. Andi Offset, Yogyakarta..
- Kadir, Abdul**, (2006). *Dasar Aplikasi Database MySQL Delphi*. Edisi II, hal 2-3. Andi Offset, Yogyakarta..
- Manning, Christopher D., Prabhakar Raghavan, dkk**, (2009). *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, England.
- Maulana, Agus Yusuf Noor**. *Analisis Information Retrieval System Dengan Model Ruang Vektor*. <http://elib.unikom.ac.id/gdl.php?mod=browse&op=read&id=jbptunikompp-gdl-agusmaulan-26077>. Diakses 04 Mei 2012.
- Mulyanto, Agus**, (2009). *Sistem Informasi Konsep & Aplikasi*. Cetakan 1, hal 1-35. Pustaka Pelajar, Yogyakarta..
- Pressman, Roger S**, (2002). *Rekayasa Perangkat Lunak*. Pendekatan Praktisi (Buku I), hal 39. Penerbit Andi, Yogyakarta.
- Proboyekti, Umi. *Software Process Model I*. [Online] Available <http://lecturer.ukdw.ac.id/othie/softwareprocess.pdf>. Diakses 26 Juni 2012
- Sakur, Stendy B**. *PHP 5 Pemrograman Berorientasi Objek Konsep & Implementasi*. Edisi 1, hal 3. Andi Offset, Yogyakarta. 2010
- Sasongko, Jati. *Aplikasi Untuk Membangun Corpus Dari Data Hasil Crawling Dengan Berbagai Format Data Secara Otomatis*. 2010. [online] Available. www.unisbank.ac.id/ojs/index.php/fti1/article/download/107/102. Diakses 04 November 2012.
- Satria, Romi Wahono dan Sri Dharwiyanti**, (2003). *Pengantar Unified Modeling Language (UML)*. [Online] Available. <http://ebookpp.com/ke/keuntungan-menggunakan-uml-pdf.html>. Diakses 06 Mei 2012.
- Solahudin, Agus dan Agus Suyadi**, (2009). *Ulumul hadits*. Cet. Ke 1, hal 13, 33. Pustaka Setia, Bandung.
- Sommerville, Ian**, (2003). *Software Engineering (Rekayasa Perangkat Lunak)*. Edisi 6, Jilid 1, hal 164. Penerbit Erlangga, Jakarta..
- Suparta, Munzier**, (2008). *Ilmu Hadits*. Edisi 5, hal 1. PT. Raja Grafindo Persada, Jakarta..
- Suwanto, Sri Ati**. *Temu Kembali Informasi Dari Sudut Pandang Pendekatan Berorientasi Pemakai*. [Online] Available. <http://www.eprints.undip.ac.id/3714/1/RETRIEval-new.doc>. diakses 14 April 2012