MODEL PENCARIAN HADITS YANG RELEVAN MENGGUNAKAN METODE COSINE SIMILARITY

[1] Ahmad Fahrudin Anshori (1010651125), [2] Deni Arifianto, S.ST., M.Kom,
[3] Triawan Adi Cahyanto, M.Kom, [4] Ulya Anisatur R., M.Kom
Jurusan Teknik Informatika Fakultas Teknik Universitas Muhammadiyah Jember
Email : ahfancool@gmail.com

ABSTRAK

Hadits merupakan pedoman seorang muslim dalam menjalankan agamanya setelah Alqur'an. Adapun Jumlah hadist yang diriwayatkan oleh para perawi hadist berjumlah puluhan ribu hadist dengan tema-tema yang sangat banyak. Dengan banyaknya hadist yang diriwayatkan oleh banyak perawi hadits maka orang yang baru mempelajari hadist akan sulit untuk menghafal atau menemukan hadist yang ingin dicari sebagai referensi untuk permasalahan-permasalahan yang ia perlukan. Diperlukan inovasi untuk memudahkan pencarian hadist berjumlah ribuan tersebut dalam suatu sistem temu kembali informasi. Sistem Temu Kembali Informasi dapat dimanfaatkan sebagai solusi karena memberikan alternatif berupa metode *similarity* yang dapat digunakan untuk mencocokkan esai jawaban ujian dan kunci jawaban soal. Metode *similarity* yang paling populer untuk diterapkan pada dokumen teks adalah *cosine similarity*. Kelebihan metode *cosine similarity* adalah tidak terpengaruh pada panjang pendeknya suatu dokumen, karena yang diperhitungkan hanya nilai *term* dari masing-masing dokumen. Pada penelitian ini dibangun sebuah model pencarian hadits yang relevan menggunakan metode *cosine similarity*. Uji coba yang dilakukan adalah pengukuran kemiripan dan penentuan *threshold*. Hasil yang didapat model berhasil menghitung kemiripan dengan jangkauan nilai *cosine* antara 0,05 – 0,25, semakin besar maka semakin mirip esai jawaban tersebut dengan kunci jawaban soal. *Threshold* terbaik yang diperoleh adalah 0,1 dengan *recall* 83% dan *precision* 46%.

Kata kunci: pencarian hadits, relevan, bahasa Indonesia, sistem temu kembali informasi, *cosine similarity*.

1. PENDAHULUAN

Evaluasi hasil belajar merupakan komponen penting dalam proses pembelajaran. Melalui evaluasi dapat diketahui sejauh mana pemahaman peserta didik terhadap materi ajar serta pencapaian tujuan pembelajaran. Kelebihan ujian esai yaitu dapat mencegah timbulnya permainan spekulasi dan dapat mengukur tingkat pemahaman seseorang terhadap materi yang diujikan. Namun ujian esai memiliki kekurangan yaitu kesulitan dalam penilaiannya karena dibutuhkan banyak waktu, tenaga, dan pikiran. Kekurangan tersebut dapat diatasi dengan melakukan otomatisasi penilaian esai. Sistem Temu Kembali Informasi dapat dimanfaatkan sebagai solusi karena memberikan alternatif berupa metode *similarity* yang dapat digunakan untuk mencocokkan esai jawaban ujian dan kunci jawaban soal. Metode *similarity* yang paling populer untuk diterapkan pada dokumen teks adalah *cosine similarity*. Kelebihan metode *cosine similarity* adalah tidak terpengaruh pada panjang pendeknya suatu dokumen, karena yang diperhitungkan hanya nilai *term* dari masing-masing dokumen. Pada penelitian ini dibangun sebuah model penilai esai otomatis jawaban ujian berbahasa Indonesia menggunakan metode *cosine similarity*.

2. TINJAUAN PUSTAKA

Penilai Esai Otomatis

Secara terminologi As-Sunnah berarti :apa saja yang disandarkan kepada Nabi Saw. baik berupa perkataan, perbuatan maupun ketetapan". Pengertian ini jika dikaitkan dengan Ushulfiqh sunnah dibatasi atas perkataan, perbuatan, dan ketetapan Nabi Saw yang berhubungan dengan Adillatul ahkam (Wahyudin).

Sistem Temu Kembali Informasi

Menurut (Cios, 2007) Sistem Temu Kembali Informasi (STKI) adalah bagian dari Ilmu Komputer yang berhubungan dengan tindakan, metode, dan prosedur untuk menemukan kembali data yang tersimpan, kemudian menyediakan informasi mengenai subyek yang dibutuhkan. Tindakan tersebut mencakup *text indexing*, *inquiry analysis*, dan *relevance analysis*. Data mencakup teks, tabel, gambar, ucapan, dan video. Sedangkan informasi mencakup pengetahuan terkait yang dibutuhkan untuk mendukung penyelesaian masalah dan akuisisi pengetahuan.

Tujuan dari STKI adalah memenuhi kebutuhan informasi pengguna dengan menemukan kembali (*retrieve*) semua dokumen yang mungkin relevan, pada waktu yang *sama* menemukan kembali sesedikit mungkin dokumen yang tidak relevan.

Sistem ini menggunakan fungsi *heuristik* untuk mendapatkan dokumen-dokumen yang relevan dengan *query* pengguna (Murad, 2007).

Meskipun pada umumnya STKI digunakan untuk *pencarian* informasi, tapi hal ini juga bisa dimanfaatkan untuk penilai esai otomatis karena konsep dari STKI adalah mencocokan antara dokumen di dalam database dan *query* pengguna. Hal ini bisa dianalogikan sebagai pencocokan antara esai jawaban ujian dan kunci jawaban soal.

Arsitektur Sistem Temu Kembali Informasi

Ada dua pekerjaan yang ditangani oleh STKI, yaitu melakukan *preprocessing* terhadap dokumen dan *query*, kemudian menerapkan metode tertentu untuk menghitung kedekatan (relevansi atau *similarity*) antara dokumen dan *query*.

Pada tahapan *preprocessing*, *query* pengguna dikonversi sesuai aturan tertentu untuk mengekstrak *term-term* penting yang sejalan dengan *term-term* yang sebelumnya telah diekstrak dari dokumen dan menghitung relevansi antara dokumen dan *query* berdasarkan pada *term-term* tersebut. Sebagai hasilnya, sistem mengembalikan suatu daftar dokumen terurut sesuai nilai kemiripannya dengan *query* pengguna (Cios, 2007).

Pembangunan Indek

Pembangunan indek dari koleksi dokumen merupakan tugas pokok pada tahapan *preprocessing* di dalam STKI. Kualitas indek mempengaruhi efektifitas dan efisiensi STKI (Chu et al., 2002). Pembangunan indek harus melibatkan konsep *linguistic processing* yang bertujuan mengekstrak *term-term* penting dari dokumen yang direpresentasikan sebagai *bag-of-words*. Beberapa langkah tahapan *preprocessing* untuk membangun indek yaitu :

1) *Tokenizing*
   Proses pemotongan dokumen menjadi daftar kata yang berdiri sendiri (*token*). Tahapan ini juga menghilangkan karakter-karakter tertentu seperti simbol dan tanda baca serta mengubah semua *token* ke bentuk huruf kecil (*lower case*).
2) *Stopword removal*
   Proses penyaringan (*filtering*) terhadap kata-kata yang tidak penting seperti, kata sambung, kata depan, kata ganti, kata sifat dan lain sebagainya.
3) *Stemming*
   Proses mengubah atau mengembalikan kata menjadi bentuk kata dasarnya (*root word*) dengan menghilangkan imbuhan. Dengan cara ini, diperoleh kelompok kata yang mempunyai makna serupa tetapi berbeda wujud sintaktis satu dengan lainnya. Algoritma *stemming* yang digunakan pada penelitian ini adalah Algoritma Nazief dan Adriani.
4) *Synonym*
   Proses *query expansion* yaitu perluasan kunci jawaban dengan menambahkan sinonim atau kata-kata yang berhubungan dengan kata-kata pada *query* agar sistem dapat mengenali persamaan kata.

Pengukuran Kemiripan

Setelah dilakukan tahapan *prepocessing* akan dilakukan proses pemberian bobot terhadap *term* dan juga proses normalisasi agar dapat di ukur tingkat kemiripan antara kunci jawaban dan esai jawaban. Beberapa langkah tahapan *similarity measure* yang digunakan pada penelitian ini antara lain :

1) Pembobotan TF-IDF
   Term Frequency (*tf*) adalah jumlah kemunculan atau frekuensi (*f*) suatu *term i* di dalam sebuah dokumen *j*.
   $$tf_{ij} = f_{ij}$$
   Inverse Document Frequency (*idf*) adalah logaritma dari jumlah dokumen keseluruhan (*n*) dibagi dengan jumlah dokumen yang memuat *term i* (df_i).
   $$idf_i = log\left(\frac{n}{df_i}\right)$$
   Pembobotan TF-IDF dilakukan dengan mengalikan hasil pembobotan TF dan IDF.
   $$W_{ij} = tf_{ij}.idf_{ij}$$
2) Normalisasi Panjang Vektor
   Normalisasi adalah sebuah cara untuk menormalkan panjang vektor dokumen sehingga vektor tersebut independen terhadap panjangnya.
   $$\frac{1}{\sqrt{w1^2 + w2^2 + \cdots + wm^2}}$$
3) *Cosine Similarity*
   Setelah dilakukan pembobotan, proses selanjutnya adalah pengukuran kemiripan menggunakan rumus *cosine similarity* sebagai berikut :
   $$Cosine \rightarrow sin(d_j, q) = \frac{\vec{d_j} \cdot \vec{q}}{|\vec{d_j}| \cdot |\vec{q}|}$$

Threshold

Untuk memperoleh hasil pencarian dokumen yang maksimal diperlukan sebuah nilai ambang batas (*threshold*) agar sistem dapat memilah mana dokumen yang mirip dan mana yang tidak. Dokumen dengan nilai ≥ *threshold* dapat dinyatakan mirip, sedangkan dokumen dengan nilai < *threshold* dinyatakan tidak mirip. Untuk mendapatkan nilai batas diperlukan suatu data training untuk melakukan uji coba.

Recall* dan *Precision

Evaluasi keakuratan dalam STKI dipengaruhi oleh dua parameter utama yaitu *recall* dan *precision*. *Recall* merupakan parameter yang digunakan untuk melakukan pengukuran terhadap tingkat keberhasilan sistem dalam mengenali suatu dokumen yang relevan terhadap *query*. Sedangkan *precision* merupakan parameter yang digunakan untuk melakukan pengukuran kecocokan antara permintaan informasi dengan respon dari permintaan tersebut. Dalam penelitian ini *precision* dapat diasumsikan sebagai kecocokan antara kunci jawaban soal dengan hasil penilaian esai jawaban.

$$Recall = \frac{Jumlah\ dokumen\ relevan\ terambil}{jumlah\ seluruh\ dokumen\ relevan}$$

$$Precision = \frac{Jumlah\ dokumen\ relevan\ terambil}{Jumlah\ seluruh\ dokumen\ terambil}$$

3. METODOLOGI PENELITIAN

Terdapat empat proses utama pada model ini, yaitu input data ke dalam *database*, data yang di input meliputi status sebagai *query*, dan hadits. Kemudian dilakukan *preprocessing* terhadap status yang telah di input dan hadits, status diasumsikan sebagai *query* dan hadits sebagai dokumen. Setelah itu dilakukan pembobotan dan pengukuran kemiripan menggunakan *cosine similarity*. Hasil dari pengukuran kemiripan ditentukan nilai *threshold*-nya untuk menilai benar dan salahnya esai jawaban ujian tersebut.

4. HASIL DAN PEMBAHASAN

Skenario Uji Coba

Pada uji coba penelitian ini digunakan dua jenis dataset yaitu status (*query*) dan hadits. Status berjumlah 20 status. Sedangkan hadits berjumlah 1000 hadits. Jadi jumlah total dataset adalah 20 *query* dan 1000 dokumen.

Kedua jenis dataset tersebut di input kedalam *database* untuk dilakukan *preprocessing*, pembobotan dan pengukuran kemiripan menggunakan metode *cosine similarity*. Hasil dari pengukuran kemiripan akan ditentukan nilai *threshold*-nya untuk menilai apakah esai jawaban tersebut bernilai benar atau salah. Sebagai evaluasi dari keakuratan sistem dalam melakukan penilaian, akan dihitung nilai *recall* dan *precision*-nya.

Dari kelima uji coba diatas diperoleh:

Tabel 4.1 Hasil Penentuan *Threshold*

No	*Threshold*	*Recall*	*Precision*
0,2	33%	65%	0,2
0,15	33%	65%	0,15
0,1	83%	46%	0,1
0,075	98%	29%	0,075
0,05	100%	23%	0,05

Semakin besar nilai *threshold* maka semakin sempit jangkauan penilaian, sebaliknya jika semakin kecil nilai *threshold* maka semakin luas jangkauan penilaian yang dilakukan. Sebagai contoh, perhitungan kemiripan yang dilakukan oleh sistem menghasilkan nilai *cosine similarity* antara 0,05 – 0,25. Penilaian dengan *threshold* 0,2 dapat diartikan bahwa hanya esai jawaban dengan nilai ≥ 0,25 yang dinilai benar oleh sistem. Hal tersebut berarti *threshold* 0,2 memiliki jangkauan yang sempit yaitu 0,2 – 0,25. Sedangkan penilaian dengan *threshold* 0,1 memiliki jangkauan yang lebih luas yaitu 0,1 – 0,25.

Dari **Tabel 4.1** diketahui bahwa uji coba pertama dengan threshold 0,2 menghasilkan precision sebesar 65%, namun nilai ini masih terlalu buruk sebagai acuan penilaian karena hanya menghasilkan recall sebesar 33%. Nilai ini masih terlalu buruk sebagai acuan penilaian karena hanya menghasilkan recall sebesar 33%. Dalam uji coba dengan threshold 0,15 nilai recall dan precision tidak berubah. Sedangkan pada uji coba selanjutnya dengan menggunakan nilai threshold yang berangsur kecil, nilai recall semakin besar dan nilai precision semakin kecil. Dari kelima uji coba di atas,, uji coba ketiga dengan threshold 0,1 dapat dikatakan yang terbaik karena menghasilkan nilai precision yang tidak terlalu turun sebesar 46% dan nilai recall sebesar 83%.

Pada uji coba keempat dan kelima dengan masing-masing threshold 0,075 dan 0,05 terdapat penurunan pada nilai precision masing-masing sebesar 29% dan 23% meskipun mengalami peningkatan recall dari 98% hingga 100% yang artinya terdapat beberapa error atau kesalahan penilaian oleh sistem. Error atau kesalahan penilaian yang dimaksud disini adalah ada beberapa hasil pencarian yang tidak relevan namun dinilai benar oleh sistem.

5. KESIMPULAN DAN SARAN

Kesimpulan

1. Metode cosine similarity pada Model Aplikasi Pencarian Hadist ini tepat diterapkan dan berhasil dilakukan untuk memunculkan hadits yang relevan. Penentuan hadist yang relevan ini berdasarkan perhitungan bobot kedekatan. Nilai cosine similarity yang dihitung oleh sistem dalam uji coba ini memiliki jangkauan nilai 0,05 – 0,25. Semakin tinggi nilai cosine berarti semakin mirip hadits yang muncul dengan status yang ditulis oleh user.
2. Pada penelitian ini dilakukan uji coba menentukan threshold yang terbaik untuk keakuratan kinerja sistem. Dari kelima uji coba penentuan threshold yang telah dilakukan, diperoleh kesimpulan bahwa semakin kecil nilai threshold, nilai recall akan semakin besar, dan nilai treshold semakin kecil. Hasil uji coba ketiga dengan threshold 0,1 sebagai threshold

terbaik karena menghasilkan nilai recall yang cukup besar (83%) dan precision yang tidak terlalu kecil (46%). Penilaian dengan threshold dibawah 0,1 mengalami penurunan precision yang drastis meskipun nilai recall semakin besar.

3. Dalam penelitian ini penambahan synonim dari term dokumen status menjadi pendukung dalam proses pengukuran kemiripan karena ada kemungkinan pemakaian kata yang semakna oleh user. Namun karena luasnya database synonim yang dipakai dalam aplikasi ini menjadikan konteks kalimat lebih meluas sehingga hasil pencarian lebih banyak yang tidak relevan meskipun dalam perhitungan cosine similarity mempunyai nilai kemiripan pada threshold tertentu.

Saran

Untuk pengembangan lebih lanjut dari sistem ini diberikan saran – saran yang berguna untuk pemikiran maupun implementasinya yaitu :

1. Jangkauan pencarian masih sangat sempit karena dalam penelitian ini hanya digunakan 1000 hadits dengan tema tertentu. Penambahan database hadist yang lebih lengkap dapat menjadi rekomendasi bagi penelitian selanjutnya.
2. Interface yang lebih menarik dan *user friendly*.
3. Penyesuaian database synonim dengan bahasan khusus sesuai kebutuhan program. Sebagai contoh dalam penelitian ini dibutuhkan *synonim* yang mendukung hasil pencarian seperti istilah-istilah serapan bahasa arab, istilah-istilah ilmu hadits dan ilmu fiqih karena dataset yang digunakan adalah terjemah hadits.

4. DAFTAR PUSTAKA

1) Agusta, Ledy (2009). Perbandingan Algoritma Stemming Porter dengan Algoritma Nazief & Adriani untuk Stemming Dokumen Teks Bahasa Indonesia. Bali: Konferensi Nasional Sistem dan Informatika 2009
2) Asian, Jelita. (2007, Maret). - *Effective Techniques For Indonesian Text Retrieval.* Melbourne, Victoria, Australia: *Science, Engineering, and Technology Portfolio* RMIT University.
3) Cios, Krzysztof J, Witold Predrycz, dkk, (2007). Data Mining A Knowledge Discovery Approach. Springer. New York.
4) Hasugian, Jonner. (2006). Penggunaan Bahasa Alamiah dan Kosa Kata Terkontrol dalam Sistem Temu Kembali Informasi Berbasis Teks. Dalam Jurnal Pustaha: Jurnal Studi Perpustakaan dan Informasi, Vol.2, No.2, Desember 2006. USU Press.
5) Kaplan, Ronald M. (2005). *A Method for Tokenizing Text*. In Festschrift in Honor of Kimmo Koskenniemi's 60th anniversary. CSLI Publications.
6) Konchady, M. (2006). *Text Mining Application Programming*, Charles River Media.
7) Murad, M.A.A, dkk.(2009). Malay Document Clustering Algorithm Based on Singular Value Decomposition. Malaysia. Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Putra Malaysia.
8) Salton, Gerard., Christopher Buckley (1988) *Term-Weighting Approaches in Automatic Text Retrieval*. Department of Computer Science, Cornell University, Ithaca, New York, USA
9) Wahyudin, Ahmad, M. Ilyas. Pendidikan Agama Islam untuk Perguruan Tinggi. Jakarta: Grasindo.