PERBANDINGAN METODE *COSINE SIMILARITY* DENGAN METODE *JACCARD SIMILARITY* PADA APLIKASI PENCARIAN TERJEMAH AL-QUR'AN DALAM BAHASA INDONESIA

Ogie Nurdiana[1], Jumadi[2], Dian Nursantika[3]
[1,2,3]Jurusan Teknik Informatika, Fakultas Sains dan Teknologi
Universitas Islam Negeri Sunan Gunung Djati Bandung
Jl. A.H. Nasution 105, Bandung 40614 Indonesia
[1]ogie.nurdiana447@gmail.com, [2]jumadi@uinsgd.ac.id, [3]dianursantika@gmail.com

Abstract—**Todays there are more applications supporting Alqurán to facilitate such a study, which could be called digital AL-Quran. But when using applications digital AL-Quran, which has many applications users experience difficulties when searching for a word that users want. This occurs when users misspell a word you want to search and applications that are not yet able to identify or justify the wrong word. In this thesis made the information retrieval system that is used to find information that is relevant to the needs of its users automatically based on conformity to the query of a collection of information. Algoritma used to determine the similarity (degree of similarity) or relevant similarity algoritma, cosine, Jaccard, and nearest neighbor (k-nn) for comparing algoritma that are more relevant to the translation application alquran. The test result proves that the cosine similarity algoritma has the highest value with the percentage of 41% compared with Jaccard 19% algoritma and nearest neighbor (k-nn) 40% on translation of AL-Quran as much 6326 document and 33 query different experiments.**

Keyword - Alqur'an, Relevan, Cosine, Jaccard, Nearest Neighbor (K-NN)

I. PENDAHULUAN

A. Latar Belakang

Al-Quran merupakan wahyu Allah atau kalam Ilahi yang diturunkan sebagai mukjizat kepada Nabi Muhammad SAW. Al-Quran diturunkan untuk menjadi pegangan bagi umat manusia yang ingin mencapai kebahagiaan, baik di dunia maupun di akhirat. Al-Quran mengandung nilai-nilai yang berhubungan dengan keimanan, syariah, akhlak serta peraturan-peraturan yang mengatur tingkah laku dan tata cara hidup manusia, baik sebagai makhluk individu maupun makhluk sosial. Al-Quran juga mengandung falsafah, kisah-kisah dan sumber ilmu pengetahuan, sebagai pelajaran, nasihat dan pencerdasan bagi umat manusia. Al-Quran dengan susunan kata yang indah, kalimat yang baik dan terang serta gaya bahasa yang mengagumkan, memberikan inspirasi yang tidak pernah kering.

Dengan kemajuan teknologi yang sangat pesat ini sudah banyak aplikasi penunjang Al-Quran untuk memudahkan untuk mempelajarinya, yang bisa disebut atau di artikan dengan adanya Al-Quran Digital. Akan tetapi, ketika menggunakan aplikasi Al-Quran Digital yang sudah banyak beredar di dunia maya, pengguna aplikasi mengalami kesulitan pada saat mencari dari sebuat kata yang pengguna inginkan dengan salah satunya yaitu dengan kesalahan pengguna dalam penulisan dari suatu kata yang dicarinya dan aplikasi pada saat ini masih kurang atau belum menangani atau membenarkan kata yang salah. Maka dilakukan perancangan dan pembuatan aplikasi yang memudahkan seseorang dalam menemmukan padanan terjemah al-qur'an, untuk mencari padanan yang sesuai dilakukan dengan mengukur kemiripan dokumen terkait (*document similarity*) tetapi dalam pengaplikasian banyak yang beredar penggunaan pengukuran kemiripan tanpa melihat keakuratan dari metode atau algoritma yang digunakan, pada permasalahan tersebut maka dibuatkan perbandingan untuk memilih metode yang lebih menunjang keakuratan dari pengukuran kemiripan.

B. Tujuan

Mengimplementasikan *text mining* menggunakan perbandingan algoritma *cosine similarity* dengan algoritma *jaccard similarity* dan metode tambahan *k-nearest neighbor (K-NN)* untuk mendukung pencocokan kata yang lebih akurat dalam terjemah Al-Qur'an.

II. LANDASAN TEORI

A. *Text Mining*

Data *mining* sebagai proses untuk mendapatkan informasi yang berguna dari gudang basis data yang benar. Data *mining* juga dapat diartikan sebagai pengekstrakan informasi baru yang diambil dari bongkahan data besar yang

membantu dalam pengambilan keputusan. Istilah *data mining* kadang disebut juga *knowledge discovery*.[1]

Data teks akan diproses menjadi data *numerik* agar dapat dilakukan proses lebih lanjut. Sehingga dalam *text mining* ada istilah *preprocessing* data, yaitu proses pendahulu yang diterapkan terhadap data teks yang bertujuan untuk menghasilkan data *numerik*. Pada proses *preprosesing* merupakan tahap dimana deskripsi di tangani untuk dapat siap diproses memasuki tahap *text mining*.[2] Tahap-tahap tersebut adalah :

1. *Parsing/ Tokenizing*

   Parsing yaitu sebuah proses yang dilakukan seseorang untuk menjadikan sebuah kalimat menjadi lebih bermakna atau berada dengan cara memecah kalimat tersebut menjadi kata-kata atau *frase-frase ("Parsing").*
2. *Stopwords Removal/ Filtering*

   Stopwords removal merupakan proses penghilangan kata tidak penting pada deskripsi melalui pengecekan kata-kata hasil *parsing* deskripsi apakah termasuk di dalam daftar kata tidak penting (*stoplist*) atau tidak. Jika termasuk di dalam *stoplist* maka kata-kata tersebut akan di-*remove* dari deskripsi sehingga kata-kata yang tersisa di dalam deskripsi di anggap sebagai kata-kata penting atau *keywords*.
3. *Stemming*

   stemming digunakan untuk mengurangi ukuran dari suatu ukuran *index file*. Misalnya dalam suatu deskripsi terdapat *variant* kata "memberikan", "diberikan", "memberi" dan "diberi" hanya memiliki akar kata (*stem*) yaitu "beri". Ukuran file daftar *index* yang semula berjumlah lima *record* akan di-*reduce* sehingga menjadi satu *record* saja.
4. *Tagging*

   Tahap *tagging* merupakan tahap mencari bentuk awal atau *root* dari tiap kata lampau atau kata hasil *stemming* yang bukan bahasa indonesia. Berikut contoh *tagging* dalam bahasa *inggris*.
5. *Anayizing*

   Tahap *analyzing* merupakan tahap penentuan seberapa jauh keterhubungan antara kata-kata dengan dokumen yang ada.

B. Pembobotan *Term Frequency-Invers Document Frequency*

Tf-Idf yaitu perhitungan yang menggambarkan seberapa pentingnya kata (*term*) dalam sebuah dokumen. Proses ini digunakan untuk menilai bobot relevansi *term* dari sebuah dokumen terhadap seluruh dokumen. *Term frequency* adalah ukuran seringnya kemunculan sebuah *term* dalam sebuah dokumen. *IDF* merupakan banyaknya istilah tertentu dalam keseluruhan dokumen, dapat dihitung dengan persamaan (1) :[3]

$$idf_{j=log\frac{n}{n_j}} \quad \textbf{(1)}$$

Dimana *n* merupakan jumlah dokumen yang di gunakan, n_j merupakan hasil *df* (*document frequency*), *log* digunakan untuk memperkecil pengaruhnya *relative* terhadap *tf*. Bobot dari term dihitung menggunakan ukuran *tf-idf* dalam persamaan (2) :

$$w = tf \text{ x } idf \quad \textbf{(2)}$$

Dimana *tf* merupakan kemunculan term dari setiap dokumen, dan *w* merupakan bobot dokumen terhadap kata atau bobot dari *key* terhadap dokumen.

C. *Cosine Siimilarity*

Metode *Cosine Similarity* merupakan metode yang digunakan untuk menghitung *similarity* (tingkat kesamaan) antar dua buah objek. Secara umum penghitungan metode ini didasarkan pada *vector space similarity measure*. Metode *cosine similarity* ini menghitung *similarity* antara dua buah objek (misalkan D1 dan D2) yang dinyatakan dalam dua buah *vector* dengan menggunakan *keywords* (kata kunci) dari sebuah dokumen sebagai ukuran. [4]

$$CosSim(d_i, q_i) = \frac{q_i.d_i}{|q_i||d_i|} = \frac{\sum_{j=1}^{t}(q_{ij}.d_{ij})}{\sqrt{\sum_{j=1}^{t}(q_{ij})^2.\sum_{j=1}^{t}(d_{ij})^2}} \quad \textbf{(3)}$$

Keterangan :

q_{ij} = bobot istilah *j* pada dokumen *i*=$tf_{ij}\ .idf_j$

d_{ij} = bobot istilah *j* pada dokumen *i*=$tf_{ij}\ .idf_j$

D. *Jaccard Similarity*

Jaccard Coeficient adalah salah satu metode yang dipakai untuk menghitung *similarity* antara dua *objects* (*items*). Seperti halnya *cosine distance* dan *matching coefficient*, secara umum perhitungan metode ini didasarkan pada *vector space similarity measure*.[5]

$$J(X,Y) = \frac{\sum_{i=1}^{p} x_i y_i}{\sum_{j=1}^{p} {x_i}^2 + \sum_{j=1}^{p} {y_i}^2 - \sum_{i=1}^{p} x_i y_i} \quad \textbf{(4)}$$

Dimana x merupakan nilai dari key dan y nilai dari dokumen.

E. *K-NN (K-Nearest Neighbor)*

K-Nearest Neighbor merupakan sebuah algoritma yang sering digunakan untuk klasifikasi teks dan data. Penggunaan *K-Nearest Neighbor* mempunyai sifat *self-learning* dimana jika semakin banyak dokumen, maka makin banyak pula sumber yang dapat digunakan untuk dibandingkan. *K Nearest Neighbor* berarti mencari tetangga yang paling dekat dengan sets yang akan di klasifikasi.[6]

$$D_{euc}(P,Q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \quad \textbf{(5)}$$

Keterangan :

P dan *Q* = titik pada ruang *vector n*-dimensi.

p_i dan q_i = besaran *scalar* untuk dimensi ke-*i* dalam ruang *vector n*-dimensi

Untuk mengukur jarak antara p_i dan q_i maka di lakukan nilai 1 untuk nilai maksimum karena akar kuadrat jika jarak biasanya lebih besar dari 1, dan jika jarak kurang dari 1 akan menghasilkan nilai yang sangat penting untuk similarity, maka rumusnya :

$$D_{euc}(P,Q) = \frac{1}{1+\sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}} \quad \textbf{(6)}$$

III. ANALISIS DAN PERANCANGAN

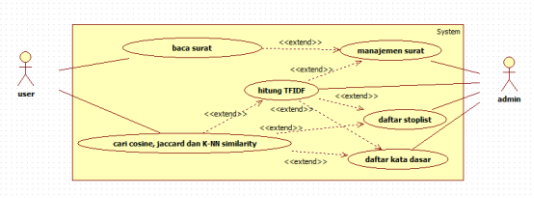
A. Analisis

Analisis kebutuhan yaitu tahapan untuk mengidentifikasi permasalahan serta proses yang terjadi dalam membangun sebuah sistem. Analisis dilakukan terhadap kebutuhan *website* yang akan dibangun, agar permanfaatan *website* yang digunakan memperoleh hasil yang optimal. Kemudian dilakukan analisis terhadap pengguna *website*, yang digunakan sebagai pembagian otoritas penggunaan *website*. Pada analisis kebutuhan bertujuan untuk mengidentifikasi jalannya suatu sistem dan permasalahan-permasalahan yang terjadi pada sistem yang dibuat.

B. Perancangan Sistem

Dalam pembahasan perancangan sistem akan menjelaskan tentang perancangan sistem. Penelitian aplikasi yang dirancang yaitu aplikasi berbasis *web* yang terdiri dari *admin*istrator yang berperan untuk mengelola aplikasi meliputi data sepenuhnya dan *user* yang berperan untuk melakukan pencarian berkaitan informasi.

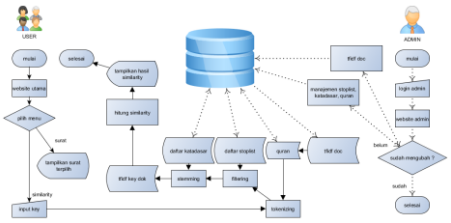
1. *Use Case Diagram*



Gambar 3.1 *Use Case Diagram*

Dalam *representasi* Gambar 3.1 terdapat dua *actor* yakni *administrator* dan *user*. *Administrator* berperan sebagai pengelola utama dari aplikasi yang dapat mengakses semua fasilitas yang ada dalam aplikasi. Sedangkan *user* adalah pengguna secara umum dari membaca surat Al-Qur'an dan melakukan proses pencarian terjemah.

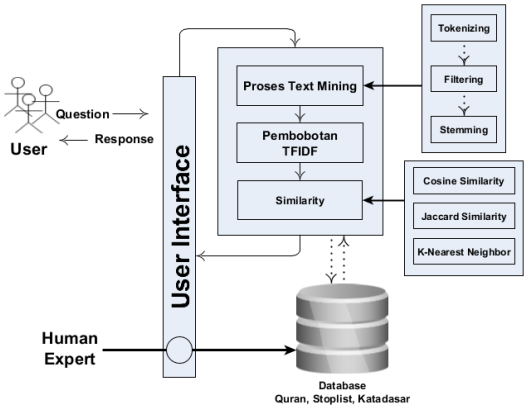
2. Skema Aplikasi



Gambar 3.2 Skema Aplikasi

Skema proses aplikasi terdapat 2 pengguna yaitu *user* dan *admin*, *admin* bisa melakukan manajemen dan penghitungan *tfidf* akan tetapi terdapat pemilihan setelah *login admin* yaitu pemilihan apakah *admin* akan mengedit atau sudah memenej, apabila sudah maka proses *admin* akan selesai, jika belum maka ada pemilihan untuk memenej atau penghitungan *tfidf*, alur dari *tfidf admin* yaitu proses yang dilakukan di database berupa data dari quran dan di *proses text mining* yang terdiri dari *tokenizing*, *filtering* dan *stemming* dari data yang sudah ada di *database*, apabila *admin* memilih menejemen yang terdiri daftar *stoplist*, daftar kata dasar dan data quran akan otomatis terpanggil di *database* ketika pemilihan berlangsung. Pada proses *user* terdapat 2 pemilihan yaitu surat untuk melihat data quran dan *similarity* untuk menghitung kemiripan yang di awali dengan *input key* dan di proses ke tahap *text mining* dan menghitung keiripan dengan metode yang digunakan.

3. Arsitektur Sistem



Gambar 3.3 Arsitektur Sistem

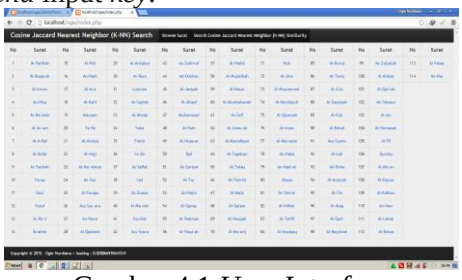
Pada Gambar 3.3 sistem aplikasi terdapat 2 pengguna yaitu *user* dan *human expert* atau *admin*, *human expert* bisa melakukan apapun yang berada di *database*, sedangkan user dari *input* pertanyaan akan menghasilkan hasil akhir dari pertanyaan itu dengan proses *text mining* kemudian pembobotan *tfidf* dan penghitungan kemiripan atau *similarity* dengan metode yang di gunakan.

IV. IMPLEMENTASI DAN PENGUJIAN

A. Implementasi

1. Implementasi *User Interface*

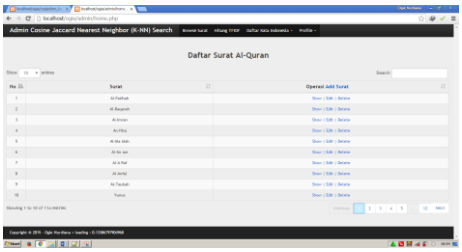
Pada tampilan *menu* utama *user* terdapat dari 2 *menu* yaitu *browse* surat yaitu *menu* utama *user* dan *menu* search *cosine jaccard k-nearest neighbor (K-NN) similarity* merupakn *menu* input *key*.



Gambar 4.1 *User Interface*

2. Implementasi Admin *Interface*

Halaman utama *admin* terdapat 3 menu yaitu browse surat merupakan tampilan utama *admin*, menu hitung *tfidf* merupakan halaman proses penghitungan *tfidf* seluruh dokumen, dan menu *logout* merupakan proses keluar akses sebagai *admin*



Gambar 4.2 Admin *Interface*

3. Pengujian

Dari hasil uji coba penghitungan *probabilitas* atau kemunculan dan menghitung kemiripan dokumen *text* dari algoritma *cosine similarity, jaccard* dan *nearest neighbor (K-NN) similarity* dari seluruh dokumen Al-Quran yaitu 6236 dokumen dengan pengujian dilakukan 30 kali dengan *key* yang berbeda. Maka hasil persentasi dan hasil rata-rata dari pengujian tersebut yaitu :

Tabel 4.1 Hasil Pengujian

Kode	***Cosine***	***Jaccard***	***KNN***
K1	0,139872	0,06099	0,095979
K2	0,02405	0,010935	0,128219
K3	0,061694	0,028895	0,125886
K4	0,133569	0,055498	0,105569
K5	0,081169	0,03594	0,119472
K6	0,070034	0,03223	0,112465
K7	0,277538	0,146228	0,062198
K8	0,295603	0,145796	0,090548
K9	0,148879	0,06319	0,056636
K10	0,198596	0,086176	0,073927
K11	0,140768	0,059974	0,101357
K12	0,103904	0,050617	0,141535
K13	0,206611	0,104605	0,108184
K14	0,103681	0,052328	0,139716
K15	0,332589	0,148003	0,117515
K16	0,203543	0,071302	0,066478
K17	0,116736	0,040292	0,08266
K18	0,14797	0,072231	0,130688
K19	0,064945	0,030819	0,147392
K20	0,094797	0,043095	0,143484
K21	0,205474	0,074278	0,066845
K22	0,057496	0,026982	0,128787
K23	0,114819	0,053021	0,147111
K24	0,202506	0,074082	0,04579
K25	0,241771	0,097114	0,08082
K26	0,090073	0,035277	0,085085
K27	0,188607	0,094438	0,095274
K28	0,238873	0,103154	0,035005
K29	0,1786	0,079652	0,103013
K30	0,084209	0,039216	0,13612
K31	0	0	0
K32	0,127077	0,057255	0,073326
K33	0	0	0

Dari hasil setiap uji coba dengan 33 percobaan *keyword* yang berbeda maka di dapat hasil perbandingan dari setiap metode yang di gunakan, untuk mendapatkan hasil perbandingan yang di lakukan dalam *persentase* maka pada setiap metode harus di dapatkan hasil rata-rata terlebih dahulu. Kemudian dilakukan penghitungan hasil *persentase* dari setiap metode. Hasil perbandingan dari setiap metode terdapat pada Tabel 4.2

Tabel 4.46 Hasil Perbandingan

	Cosine	*Jaccard*	*KNN*
Rata-rata	0,141699	0,062837	0,095366
Persentase	47%	21%	32%

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan Hasil Pengujian yang telah dilakukan didapat beberapa kesimpulan :

1. Metode *cosine*, *jaccard* dan *k-nearest neighbor (K-NN)* yang digunakan pada proses klasifikasi dokumen teks dengan hasil akhir dari percobaan 33 kali dengan *key* yang berbeda dan total 6326 dokumen di dapat metode *cosine* yang nilai kemiripannya tertinggi yaitu 41% dari metode *jaccard* 19% dan *k-nearest neighbor (K-NN)* 40%, karena metode *cosine similarity* mempunyai konsep normalisasi panjang *vektor* data dengan membandingkan *N-gram* yang sejajar satu sama lain dari 2 pembanding. Sedangkan pada metode *jaccard* hanya membandingkan isi *N-gram* dengan *eksak* dan hanya melihat apakah ada suatu *N-gram* tertentu pada pembanding tanpa melihat posisi penulisan yang berbeda. Pada *euclidean distance* yang diterapkan di metode *k-nearest neighbor (K-NN)* tidak mempunyai konsep *normalisasi* panjang *vektor* data, sehingga nilai akurasi metode dipengaruhi oleh panjang 2 data pembanding dan harus menentukan nilai dari pamameter *K* (jumlah dari tetangga terdekat). Maka metode *cosine similarity* menjadi usulan alternatif untuk mencari kemiripan dari teks *mining*.
2. Keakuratan hasil yang dicari sangat berpengaruh pada kata kunci yang di cari, agar hasil yang ingin dicari lebih *relevan* maka kata kunci harus sesuai dengan aturan penulisan bahasa Indonesia.
3. Keakuratan pengelompokan dokumen teks dan pemilihan kata untuk dijadikan sebagai *term* sangat terpengaruh oleh kelengkapan daftar *stoplist* dan hasil *root* kata dari *stemming* untuk teks bahasa Indonesia.

B. Saran

Hasil dari proyek akhir ini belumlah sempurna, untuk meningkatkan hasil yang dicapai dapat dilakukan :

1. Penambahan metode yang biasa digunakan untuk jenis teks *mining* seperti metode *Inner Similarity* dan *Dice Similarity*
2. Untuk penelitian selanjutnya disarankan untuk perbaikan dalam pengindeksan seluruh

dokumen sehingga dalam pemrosesan *Term Frequency-Invers Document Frequency* (*TF-IDF)* tidak memakan waktu lama.

DAFTAR PUSTAKA

[1] E. Prasetyo, *Data Mining – Konsep Dan Aplikasi Manggunakan Maltab*. Yogyakarta : ANDI , 2012.

[2] S. Nurhayati, "Text Mining", *Implementasi Text Mining Untuk Klasifikasi Kesenian Tradisional Dengan Metode Nbc (Naïve Bayes Classifier)*, Fakultas Teknik dan Ilmu Komputer Universitas Komputer Indonrsia. Bandung, 2010, pp. 1-5.

[3] M. Fitri, Kombinasi Tf-Idf, *Perancangan Sistem Temu Balik Informasi Dengan Metode Pembobotan Kombinasi Tf-Idf Untuk Pencarian Dokumen Berbahasa Indonesia*, Tanjungpura, 2013, pp. 1-6

[4] G. A. Pradnyana dan N. A. Sanjaya, "Cosine Similarity", *Perancangan Dan Implementasi Automated Document Integration Dengan Menggunakan Algoritma Complete Linkage Agglomerative Hierarchical Clustering*, vol. 5, (2), pp. 1-10, September 2012.

[5] S. S. S. Purwandari, *Rancang Bangun Search Engine Tafsir Al-Quran Yang Mampu Memproses Teks Bahasa Indonesia Menggunakan Metode Jaccard Similarity*, Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang, 2012, pp. 9-27.

[6] N. Krisandi, Helmi, dan B. Prihandono, "Klasifikasi Data", *Algoritma K-Nearest Neighbor Dalam Klasifikasi Data Hasil Produksi Kelapa Sawit Pada PT.Minamas Kecamatan Parindu*, vol. 2, (1), pp. 33-38, 2013.