JURNAL DEVICE, VOL. 10 NO 1, 1-8
ISSN (print) : 0216-9185

ANALISIS PERBANDINGAN ALGORITMA TF-IDF DENGAN SQL QUERY UNTUK KASUS PENCARIAN PADA SISTEM INFORMASI DOKUMENTASI ARSIP (SIDOKAR)

Dian Asmarajati [1)]

[1)] Teknik Informatika, Universitas Sains Al-Qur'an
Email : dianaj@fastikom-unsiq.ac.id

ABSTRAK

Arsip merupakan salah satu sumber informasi yang berfungsi penting untuk menunjang proses kegiatan administrasi dan manajemen pada sebuah instansi dalam berbagai bentuk dan media. Tidak hanya pengelolaan arsip tetapi diperlukan juga fitur pencarian. Proses pencarian yang biasanya digunakan adalah SQL query dengan susunan dari beberapa perintah dasar SQL untuk menghasilkan hasil pencarian yang sesuai. Namun dengan SQL query tersebut belum cukup untuk melakukan seleksi dan untuk menemukan data yang relevan dikombinasikan dengan Algoritma TF-IDF. Dengan demikian maka perlu dibuatnya aplikasi perbandingan antara Algoritma TF-IDF dengan SQL query untuk mengetahui efisiensi waktu proses dan keefektifan dalam kesesuaian hasil pencarian data pada suatu data arsip. Hasil yang diperoleh dari penelitian ini adalah dalam hal kecepatan proses atau running time SQL query lebih efisien dari pada Algoritma TF-IDF sedangkan dalam hal kesesuaian hasil pencarian data relevan Algoritma TF-IDF lebih efektif dari pada SQL query.

Kata Kunci : Arsip, Sistem Pencarian, Perbandingan, SQL query, Algoritma TF-IDF

ABSTRACT

Archive is one source of information that functions is important to support the process of administrative and management activities in an agency in various forms and media. Not only archive management, but also a search feature is needed. The search process that is usually used is SQL query with an arrangement of some basic SQL commands to produce the appropriate search results. However, the SQL query is not enough to make a selection and to find relevant data combined with the TF-IDF algorithm. Thus it is necessary to make a comparison application between the TF-IDF Algorithm with SQL query to find out the efficiency of processing time and effectiveness in the suitability of data search results on an archive data. The results obtained from this research are that the speed of processing or running time of SQL queries is more efficient than the TF-IDF algorithm while in terms of the suitability of relevant data search results the TF-IDF algorithm is more effective than SQL queries.

Keywords: Archive, Search System, Comparison, SQL query, TF-IDF Algorithm

1. PENDAHULUAN

Arsip merupakan rekaman kegiatan atau peristiwa dalam berbagai bentuk dan media dengan perkembangan teknologi informasi dan komunikasi yang dibuat dan diterima oleh lembaga atau instansi pemerintah maupun swasta.(Simangunsong, 2018) Tidak hanya pengelolaan suatu surat tetapi juga pada pencarian suatu dokumen arsip tersebut. Terdapat ratusan atau ribuan arsip pada setiap instansi yang masing-masing dokumen memiliki banyak informasi didalamnya. Apalagi jika terjadi penambahan dokumen pada arsip setiap saat. Begitu banyak dokumen yang tertumpuk dan menyebabkan kesulitan untuk mencari dokumen tersebut.

Apabila pada Kecamatan Mojotengah setiap tahunnya memiliki 2000 data arsip yang terdiri dari surat masuk dan surat keluar. Maka dengan kurun waktu 5 tahun sudah menjadi 10000 data arsip. Dan data setiap tahunnya tidak dapat dipastikan dalam bentuk angka karena pasti berbeda-beda. Pencarian yang manual sangat memakan waktu lama untuk menemukan dokumentasi arsip surat yang ingin dicari. Dengan demikian, bagaimana cara mendapatkan suatu informasi dengan mudah, cepat, akurat dan sesuai dalam pencarian arsip. Salah satu solusi dari permasalahan ini adalah dibuatnya sistem pencarian.

Sistem pencarian adalah salah satu alat bantu untuk mencari suatu informasi yang sesuai dengan apa yang diinginkan. Dengan sistem pencarian suatu proses pencarian akan memberikan efisiensi waktu dan kesesuaian hasil pencarian pada data arsip. Sistem pencarian pada arsip sangat dibutuhkan untuk mempermudah bagi penguna. Proses pencarian yang biasa digunakan adalah SQL query dengan susunan dari beberapa perintah dasar SQL untuk menghasilkan hasil pencarian yang sesuai. Pada pencarian di sistem pencarian sering kali pengguna lupa dan tidak mengetahui kata yang dicari dengan pasti dan untuk menemukan data yang relevan dengan kebutuhan dari penggunanya secara otomatis di perlukannya algoritma dalam query pencarian. Salah satu algoritma yang digunakan untuk mencari data yang relevan pada pencarian adalah algoritma TF-IDF.

SQL query bekerja dengan memanggil satu persatu tabel data yang dicari dalam proses pencarian sedangkan algoritma TF-IDF memiliki keunggulan dalam memfilter. Algoritma TF-IDF memerlukan script yang panjang tetapi memiliki tahap-tahap proses dengan menghilangkan sesuatu yang tidak diperlukan dapat membuat pencarian dengan cepat. Proses pencarian yang sama-sama menggunakan query tetapi mengkombinasikan beberapa langkah dari algoritma TF-IDF seperti proses penghilangan tanda baca, merubah kata menjadi huruf kecil, meghilangkan kata umum, mencari kata dasar, mengindeks dokumen, menghitung jumlah kata menggunakan metode TF-IDF, perangkingan suatu data berdasarkan tingkat relevansinya.

Pencarian menggunakan fitur pencarian dari SQL query dengan yang menggunakan algoritma TF-IDF apakah akan berpengaruh terhadap kecepatan proses pencarian dan kesesuaian hasil pencarian pada data yang dicari. Dikarenakan belum diketahui perbandingan kedua proses pencarian maka perlu dibuatnya aplikasi perbandingan algoritma TF-IDF dan SQL query untuk mengetahui kecepatan dan kesesuaian data pada masing-masing proses pencarian tersebut. Kemudian dilakukan analisa perbandingan efisiensi waktu pemprosesan dan keefektifan pencarian dari dua cara yang berbeda. Peneliti hanya membandingkan proses pencarian menggunakan algoritma TF-IDF dengan query sql pada MySQL dan bagaimanakah pengaruhnya terhadap efisiensi waktu proses dan keefektifan pada kesesuaian hasil dari pencarian data pada suatu data arsip. (Safitri, 2013)

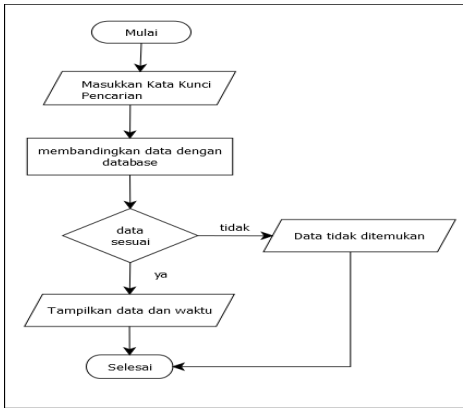
2. METODE

Metode pengumpulan data dalam penelitian ini dilakukan untuk memperoleh data yang nantinya data tersebut dapat dianalisa dan diolah oleh penulis, sehingga penulis mengetahui penyelesaian dari masalah yang dihadapi.

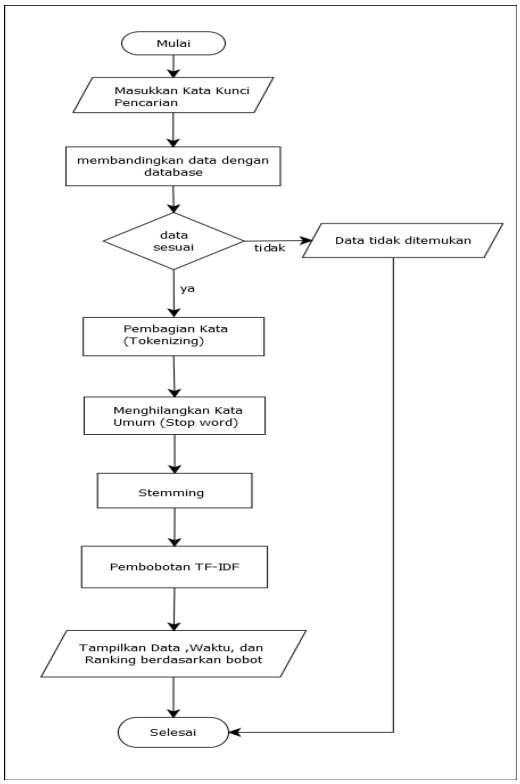
a. Observasi
   Merupakan teknik pengumpulan data dengan cara melakukan pengamatan terhadap Sistem Arsip (SIDOKAR) pada Kecamatan Mojotengah secara langsung, kemudian menarik kesimpulan dari beberapa kendala yang ada.. Pada penelitian ini, yang dijadikan sebagai data utama arsip dari SIDOKAR yang berjumlah 50 data terdiri dari 25 arsip surat masuk dan 25 arsip surat keluar.
b. Wawancara
   Merupakan teknik pengumpulan data dengan melakukan tanya jawab secara langsung terhadap bagian kepegawaian yang bertanggung jawab mengelola surat menyurat di Kecamatan Mojotengah. Wawancara untuk memperoleh data-data yang dibutuhkan oleh penulis seperti prosedur dan mekanisme pengarsipan surat.
c. Studi Literatur
   Merupakan teknik pengumpulan data pengetahuan dengan mempelajari buku-buku, jurnal, artikel dan koleksi perpusatakaan yang berkaitan dengan penulisan tugas akhir yang berhubungan tentang arsip, sistem temu kembali informasi, studi komparasi dan algoritma TF-IDF.

3. HASIL DAN PEMBAHASAN

Tahap pertama dalam merancang aplikasi perbandingan pada pencarian dengan merancang alur proses yang dikerjakan oleh aplikasi. Alur proses yang akan digambarkan meliputi dua proses yaitu pencarian menggunakan SQL Query dan pencarian menggunakan Algoritma TF-IDF sebagai berikut:



Gambar 1. Diagram Alir Proses Pencarian menggunakan SQL Query



Gambar 2. Diagram Alir Proses Pencarian menggunakan Algoritma TF-IDF

Pada Gambar 1. Terdiri dari proses-proses berikut : Masukkan kata kunci pencarian, Pengecekan Kata Kunci, Perbandingan dengan database dokumen, Hasil pencarian

Sedangkan pada Gambar 2. Terdiri dari proses-proses berikut: Memasukkan kata kunci pencarian, Pengecekan kata kunci, Menghilangkan kata-kata umum (filtration), Stemming, Perbandingan dengan database dokumen, Perhitungan bobot TF-IDF, Perangkingan, Hasil pencarian

Pada Tahap kedua yaitu Analisis Pencarian Sistem Temu Kembali Informasi / Information Retrieval System (IRS) dengan Pembobotan Kata menggunakan Algoritma Term Frequency – Inverse Document Frequency (TF-IDF).

Sebelum dilakukan pembobotan kata (TF-IDF) dilakukan indexing. Pembangunan index dari koleksi dokumen merupakan tugas pokok pada tahapan prepocessing di dalam Information Retrieval. Tahap- tahap pada preprocessing adalah :

- Tahap Tokenizing adalah pengambilan kata-kata (term) dengan cara menghapus karakter tanda baca yang terdapat pada dokumen dan mengubah kumpulan kata menjadi huruf kecil (lowercas). Kata Kunci (KK) : Rakor Persiapan Pelatihan Pengelolaan Keuangan (Aziz, Abdul. (2015).
- Setelah data arsip dijasikan token (per kata), langkah selanjutnya adalah proses pengecekan adanya kata umum pada data tersebut. Apabila ditemukan kata umum (misalnya yang, dari, atau, dan, dengan, sedangkan, mungkin dan sebagainya) maka akan dihapus. Hal tersebut akan meningkatkan akurasi dari pencarian dimana suatu yang dicari berbeda kata satu dengan lainnya.
- Langkah berikutnya yaitu stemming (mencari kata dasar). Proses stemming seperti kata membaca dan baca akan diperoleh bentuk kata dasar yang sama yaitu baca.
- Perhitungan bobot TF-IDF
  Contoh :
  Kata Kunci (KK) : Rakor Persiapan Pelatihan Pengelolaan Keuangan
  - Menghitung *Term Frequency (TF) Term Frequency (tf)* merupakan frekuensi kemunculan *term (t)* pada dokumen (d)

Tabel 1. Perhitungan TF

Term	TF													
	KK	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13
rakor	1	0	0	0	0	0	0	0	0	0	0	0	0	0
siap	1	0	0	0	0	0	0	0	1	0	0	0	0	0
latih	1	0	0	0	0	0	0	0	1	0	0	1	0	1
kelola	1	0	0	0	0	0	0	0	0	0	0	1	0	0
uang	1	0	0	0	0	0	0	0	0	0	0	1	0	0

Tabel 2. Lanjutan Perhitungan TG

Term	TF											
	D14	D15	D16	D17	D18	D19	D20	D21	D22	D23	D24	D25
rakor	1	0	0	0	0	0	0	0	0	0	1	0
siap	0	0	0	0	0	0	0	0	0	0	0	0
latih	0	0	0	0	0	0	0	0	0	1	0	0
kelola	0	0	0	0	0	0	0	0	0	1	0	0
uang	0	0	0	0	0	0	0	0	0	0	0	0

  - Menghitung *Document Frequency (DF) Document Frequency (df)* adalah banyaknya dokumen dimana suatu *term* (t) muncul

Tabel 3. Perhitungan *Document Frequency* (DF)

DF
2
1
4
2
1

  - Menghitung Inverse Document Frequency (IDF)

Tabel 4 Perhitungan *Inverse Document Frequency* (IDF)

IDF
log (d/df)
Log 25/2 = 1,097
Log 25/1 = 1,398
Log 25/4 = 0,796
Log 25/2 = 1,097
Log 25/1 = 1,398

- Perhitungan Pembobotan TF-IDF Term Query Dalam setiap Dokumen

Tabel 5 Perhitungan Pembobotan TF-IDF

Term	W = TF * IDF													
	KK	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13
rakor	1,097	0	0	0	0	0	0	0	0	0	0	0	0	0
siap	1,398	0	0	0	0	0	0	0	1,398	0	0	0	0	0
latih	0,796	0	0	0	0	0	0	0	0,796	0	0	0,796	0	0,796
kelola	1,097	0	0	0	0	0	0	0	0	0	0	1,097	0	0
uang	1,398	0	0	0	0	0	0	0	0	0	0	1,398	0	0

Tabel 6. Lanjutan Perhitungan Pembobotan TF-IDF

Term	W = TF * IDF											
	D14	D15	D16	D17	D18	D19	D20	D21	D22	D23	D24	D25
rakor	1,097	0	0	0	0	0	0	0	0	0	1,097	0
siap	0	0	0	0	0	0	0	0	0	0	0	0
latih	0	0	0	0	0	0	0	0	0	0,796	0	0
kelola	0	0	0	0	0	0	0	0	0	1,097	0	0
uang	0	0	0	0	0	0	0	0	0	0	0	0

Dikarenakan terdapat kata kunci putri memiliki nilai idf=0 maka perhitungan bobot nya menggunakan rumus:

$$W_{ij} = tf_{ij} \times (\log (D/df_j)+1)$$

Berhubung nilai idf≠0 maka lanjut pembobotan masing-masing dokumen. Dan semisal memiliki nilai W=0 berarti tidak termasuk data yang dicari karena tidak memiliki bobot.

Bobot (W) untuk D8 = 0 + 1,398 + 0,796 + 0 + 0 = **2,194**
Bobot (W) untuk D11 = 0 + 0 + 0,796 + 1,097 + 1,398 = **3,291**
Bobot (W) untuk D13 = 0 + 0 + 0,796 + 0 + 0 = **0,796**
Bobot (W) untuk D14 = 1,097 + 0 + 0 + 0 + 0 = **1,097**
Bobot (W) untuk D23 = 0 + 0 + 0,796 + 1,097 + 0 = **1,893**
Bobot (W) untuk D24 = 1,097 + 0 + 0 + 0 + 0 = **1,097**

Setelah bobot masing-masing dokumen diketahui, maka dilakukan proses pemeringkatan atau perankingan dokumen berdasarkan besarnya tingkat kerelevan (kesesuaian) dokumen terhadap *query*, dimana semakin besar nilai bobot dokumen terhadap *query* maka semakin besar tingkat similaritas dokumen tersebut terhadap query yang dicari.

- Hasil dan Perankingan Dokumen terhadap *Query*

Tabel 7. Hasil dan Perankingan Dokumen terhadap query

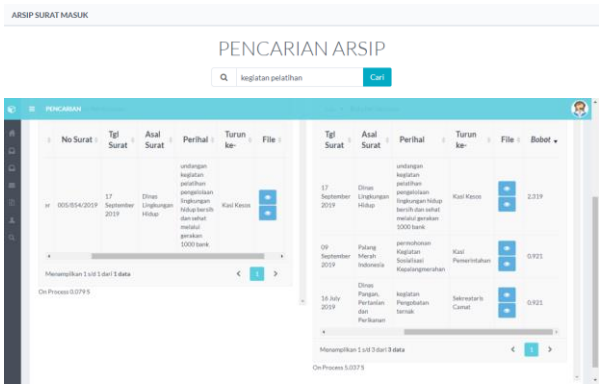
	D8	D11	D13	D14	D23	D24
W	**2,194**	**3,291**	**0,796**	**1,097**	**1,893**	**1,097**
Rank	II	I	VI	IV	III	V

Dengan demikian dapat dihasilkan daftar dokumen teranking berdasarkan nilai kesesuaian (similarity) antara dokumen dan query masukan yang kemudian akan diberikan kepada pengguna. Dari hasil pembobotan dan perankingan dapat diketahui bahwa dokumen 11 memiliki tingkat relevansi tertinggi disusul dokumen 8 lalu dokumen 23, dokumen 14, dokumen 24 dan terkecil adalah dokumen 13.

- Analisis Pencarian menggunakan SQL query

  Memasukkan keyword (kata kunci) pada sistem pencarian. Bahwa pada sql query biasa pencarian tidak dilakukannya *prepocessing* seperti pencarian pada sistem temu kembali informasi (IRS). Tahap pencariannya hanya dilakukan pengecekkan query pada data yang sesuai dengan database. Pencarian menggunakan SQL query ataupun TF-IDF sama-sama menggunakan query hanya saja pada TF-IDF menambahkan penghitungan bobot term dan dilakukannya *prepocessing* untuk mencari kemiripan kata kunci pada suatu data yang dicari.

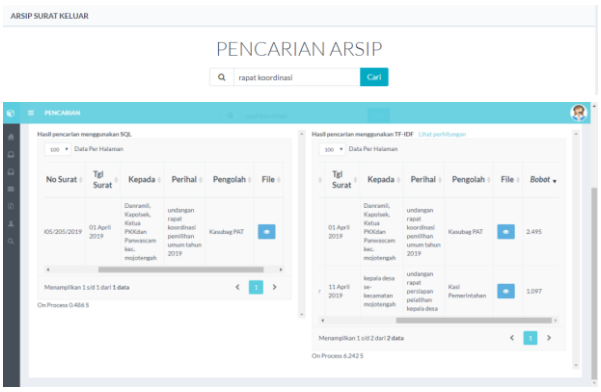
Tahap Selanjutnya yaitu Implementasi dan Pengujian



Gambar 3. Pengujian Asip Surat Masuk dengan Kata Kunci Normal

Arsip Surat Masuk

Pengujian dengan Kata Kunci : Kegiatan Pelatihan

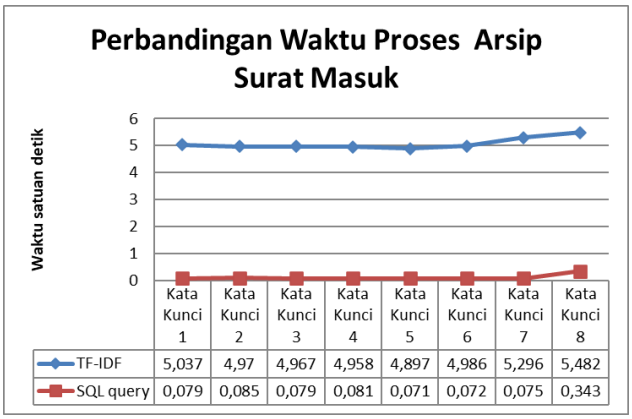


Gambar 4. Pengujian Arsip Surat Keluar dengan Kata Kunci Normal

Arsip Surat Keluar

Pengujian dengan Kata Kunci : Rapat Koordinasi

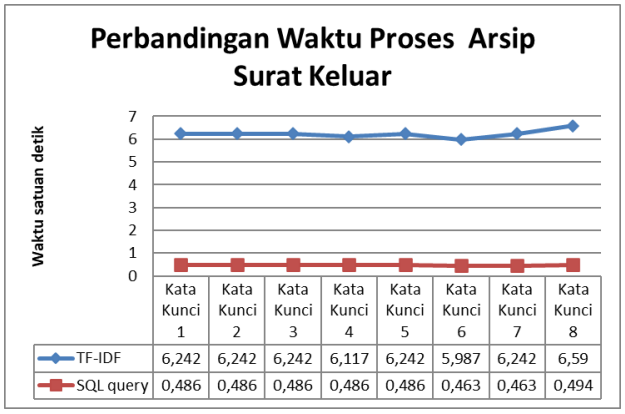
Analisa Hasil Pengujian Perbandingan



Gambar 5. Grafik Perbandingan Waktu Proses Arsip Surat Masuk

Dari grafik diatas menunjukan bahwa pada pada pencarian menggunakan Algoritma TF-IDF lebih memakan waktu proses lebih lama dari SQL query. Dapat disimpulkan bahwa pencarian menggunakan SQL query memiliki tingkat efisiensi dari Algoritma TF-IDF

Analisa Pengujian Perbandingan Waktu Proses pada Arsip Surat Keluar



Gambar 6. Grafik Perbandingan Waktu Proses Arsip Surat Keluar

Sama dengan pencarian Arsip Surat Masuk bahwa pada pencarian Arsip Surat Keluar ini Algoritma TF-IDF lebih memakan waktu proses lebih lama dari SQL query. Dapat disimpulkan bahwa pencarian menggunakan SQL query memiliki tingkat efisiensi dari Algoritma TF-IDF.

4. PENUTUP

4.1. Kesimpulan

Berdasarkan uraian analisis, perancangan, implementasi dan pengujian yang telah dilakukan pada Analisis perbandingan Algoritma TF-IDF dengan SQL query untuk kasus pencarian pada Sistem Informasi Dokumentasi Arsip (SIDOKAR) yang penulis teliti, maka dapat diambil kesimpulan:

1. Mengetahui perbandingan antara Algoritma TF-IDF dengan SQL query pada SIDOKAR dalam hal efisiensi waktu proses dan keefktifan dalam kesesuaian hasil proses pencarian data pada data arsip.
2. Perbandingan waktu proses arsip surat masuk maupun arsip surat keluar bahwa SQL query lebih efisien dibandingkan Algoritma TF-IDF pada kecepatan waktu prosesnya. Karena pada penelitian arsip surat masuk waktu proses pada SQL query memiliki rata-rata 0,110 S sedangkan Algoritma TF-IDF adalah 4,452 S. Begitu pula dengan penelitian arsip surat keluar waktu proses pada SQL query memiliki rata-rata 0,481 S sedangkan Algoritma TF-IDF adalah 6,238 S.
3. Dengan melakukan analisa menggunakan hardware yang berbeda dan memiliki spesifikasi lebih tinggi. Berpengaruh pada pencarian bahwa yang menggunakan hardware spesifikasi lebih tinggi menyatakan lebih cepat dari pada hardware yang lebih rendah. Karena pada penelitian arsip surat masuk waktu proses pada SQL query memiliki rata-rata 0,108 S sedangkan Algoritma TF-IDF adalah 2,767S. Begitu pula dengan penelitian arsip surat keluar waktu proses pada SQL query memiliki rata-rata 0,212 S sedangkan Algoritma TF-IDF adalah 3,179 S.
4. Perbandingan kesesuaian hasil pada arsip masuk maupun arsip surat keluar bahwa Algoritma TF-IDF lebih efektif dibandingkan SQL query.

Karena pada penelitian arsip surat masuk yang menggunakan Algoritma TF-IDF memiliki persentase 100% sedangkan SQL query hanya 12,5%. Begitu pula dengan penelitian arsip surat keluar Algoritma TF-IDF memiliki persentase 100% sedangkan SQL query hanya 22,22%.

4.2. Saran

1. Apabila akan menerapkan Algoritma TF-IDF untuk pencarian data pada arsip surat disarankan melakukan perbandingan dengan algoritma-algoritma yang dapat mempercepat waktu proses pencarian. Agar memperoleh pencarian yang efisien dan efektif salah satunya adalah penggunaan Algoritma Hamming Distance.
2. Optimalisasi query TF-IDF dan proses pencarian agar waktu pencarian lebih cepat.
3. Untuk pengembangan sistem sehingga pada pencarian dapat mencari tidak berdasarkan “perihal” dan “asal surat” maupun “kepada” tetapi dapat seluruhnya.

5. DAFTAR PUSTAKA

Aziz, Abdul. (2015). *Penerapan Metode Term Frequency Inverse Document Frequency (TF-IDF) Dan Cosine Similarity Pada Sistem Temu Kembali Informasi Untuk Mengetahui Syarah Hadits Berbasis Web*

(Studi Kasus: Syarah Umdatil Ahkam). Jurnal Teknik Informatika, 11(2), 151.

Safitri, Rima Noer. 2013 *Penerapan Metode Term Frequency Inverse Docment Frequency (TF-IDF) dan Cosine Similarity pada Sistem Temu Kembali Informasi untuk Mengetahui Syarah Hadits Berbasis Web (Studi Kasus: Syarah Umdatil Ahkam). Jurnal Teknik Informatika* 11(2), 151.

Simangunsong. (2018). Sistem Informasi Pengarsipan Dokumen Berbasis Web. Jurnal Mantik Penusa, (1)