

Sistem Rekomendasi Hasil Pencarian Artikel Menggunakan Metode *Jaccard's Coefficient*

Muhammad Fadelillah, Imam Much Ibnu Subroto, Dedy Kurniadi

Teknik Informatika, Universitas Islam Sultan Agung
Correspondence Author: imam@unissula.ac.id

Abstract

Portal Garuda IPI (Indonesian Publication Index) merupakan situs web yang menyediakan layanan penelusuran, indeksasi, abstraksi, monitoring (pemantauan), dan peningkatan standar kualitas atas berbagai publikasi ilmiah di Indonesia. Portal Garuda IPI menyediakan berbagai fitur layanan untuk memudahkan penelusuran referensi ilmiah Indonesia. Terdapat hal yang sebenarnya kurang optimal dalam sistem ini yaitu, pengguna mencari artikel dengan memasukkan query pencarian, namun sistem seringkali menampilkan hasil yang tidak relevan dengan query yang dimasukkan oleh pengguna. Dengan adanya sistem rekomendasi yang dibangun ini dapat membuat proses pencarian artikel menjadi lebih optimal karena dengan sistem ini pengguna dapat menemukan artikel yang dicari dengan menggunakan metode jaccard's coefficient untuk menghitung kemiripan sehingga sistem dapat menampilkan rekomendasi artikel yang relevan dengan artikel yang dicari.

Kata Kunci : Sistem Temu- Kembali Informasi, Sistem Rekomendasi, *Jaccard's Coefficient*, *Similarity*.

1. PENDAHULUAN

Pencarian informasi saat ini dilakukan dengan menggunakan mesin pencari atau Sistem Temu Kembali Informasi (STKI), *user* menuliskan *query* dan mesin pencari akan menampilkan hasil pencarian. Mesin pencari yang sudah ada dan banyak digunakan saat ini memberikan hasil perolehan pencarian yang banyak (banyak dokumen yang terambil), sehingga diperlukan waktu untuk menentukan hasil pencarian yang relevan. Menentukan hasil yang relevan sesuai dengan keinginan *user* dengan jumlah hasil pencarian yang banyak akan menyulitkan *user*. Hal ini terjadi karena dokumen yang terambil oleh sistem jumlahnya banyak, maka sistem berkemungkinan menampilkan hasil pencarian yang tidak relevan. Banyaknya dokumen hasil pencarian ini membuat waktu yang dibutuhkan dalam pencarian menjadi lebih banyak dari yang diharapkan [1].

Perkembangan penelusuran informasi saat ini menghasilkan *recall* yang tinggi dan *precision* yang rendah. *Recall* yang tinggi diartikan bahwa dokumen yang dihasilkan dalam penelusuran dokumen adalah banyak, sedangkan *precision* rendah dapat diartikan bahwa dokumen yang diharapkan dapat ditemukan sedikit [1].

Portal Garuda IPI (*Indonesian Publication Index*) adalah situs web yang menyediakan layanan penelusuran, indeksasi, abstraksi, *monitoring* (pemantauan), dan peningkatan standar kualitas atas berbagai publikasi ilmiah di Indonesia. Situs ini dibangun oleh IAES (*Institute of Advanced Engineering and Science Indonesia Section*) seksi Indonesia pada tahun 2012.

Dewasa ini kemajuan teknologi berkembang sangat pesat, bahkan hampir seluruh dari kegiatan yang kita lakukan menggunakan teknologi karena sangat bermanfaat dan juga sebagai faktor penunjang kelangsungan hidup manusia. Melalui teknologi informasi, kita bisa berkomunikasi dan mendapatkan informasi yang kita inginkan dengan cepat dan mudah tanpa mengkhawatirkan jarak dan waktu. Perkembangan teknologi informasi dan komputer yang terus berlanjut membawa dampak pada proses pengolahan data dan penyampaian informasi sesuai kebutuhan. Sehingga setiap hal dapat dilakukan dengan lebih mudah dari biasanya. Begitu juga dalam hal mencari artikel jurnal, saat ini banyak yang memanfaatkan internet untuk mencari artikel jurnal sebagai referensi penelitian. Dengan semakin banyaknya permintaan artikel jurnal untuk kebutuhan penelitian, maka diperlukan sistem yang dapat mempermudah pengguna dalam mencari artikel jurnal yang sesuai dengan apa yang dicari. Selama belum ada sistem yang secara efektif memberikan informasi yang akurat tentang pencarian artikel jurnal. Saat ini apabila ada pengguna yang ingin mencari sebuah artikel, sistem akan menampilkan hasil pencarian menurut apa yang dimasukkan pada kolom pencarian. Sebagian besar pengguna sering mengalami masalah ketika artikel yang ditampilkan tidak sesuai dengan apa yang dicari oleh pengguna. Penelitian ini bertujuan mendesain sistem rekomendasi dengan metode *Jaccard's Coefficient*, dan menguji kinerja temu kembali menggunakan *recall* dan *precision*. Selanjutnya dibuat aplikasi dengan algoritma yang efektif dalam

pencarian dokumen dengan metode *Jaccard's Coefficient* pada Portal Garuda IPI (*Indonesian Publication Index*), sehingga pengguna mudah mencari dokumen yang diinginkan. Berdasarkan uraian diatas, penulis tertarik untuk membuat "Sistem Rekomendasi Artikel Jurnal Indonesia Menggunakan Metode *Jaccard's Coefficient*".

2. JACCARD'S COEFFICIENT

Jaccard's Coefficient merupakan metode yang digunakan untuk menghitung tingkat kesamaan (*similarity*) antar dua buah objek. Untuk tujuan klastering dokumen, fungsi yang baik adalah fungsi *Jaccard's Coefficient*[2]

Untuk notasi himpunan dapat digunakan rumus

$$\text{Similarity}(X, Y) = \frac{|X \cap Y|}{|X| + |Y| - |X \cap Y|} \quad (1)$$

dari notasi himpunan di atas dapat dibuat persamaan matematika sebagai berikut :

$$\text{Similarity}(X, Y) = \frac{\sum_{i=1}^t x_i y_i}{\sum_{i=1}^t x_i + \sum_{i=1}^t y_i - \sum_{i=1}^t x_i y_i} \quad (2)$$

Untuk merealisasi konsep ini, setiap dokumen harus dikorelasikan dengan subyek dengan relasi *many to many*, artinya satu subyek bisa memiliki beberapa dokumen, sebaliknya satu dokumen bisa juga memiliki beberapa subyek. Untuk dapat melakukan pengelompokan dokumen terhadap subyek dapat dilakukan dengan 2 cara, yaitu [3] :

1. Memasukkan setiap dokumen secara langsung kedalam subyek.
2. Memasukkan dokumen secara tidak langsung kedalam suatu subyek dengan menggunakan bantuan *term*. Untuk dokumen dalam jumlah yang sangat banyak, tidak dilakukan pengelompokan dengan cara memasukkan satu persatu dokumen kedalam subyek, yaitu dengan memperhitungkan frekuensi kemunculan *term* dalam dokumen tersebut dan jumlah dokumen yang mengandung *term* tersebut.

3. PERANCANGAN SISTEM

Secara umum sistem yang akan dibangun adalah menggunakan metode *Jaccard Coefficient*, yaitu suatu metode yang bertujuan untuk menentukan bobot dari suatu dokumen dan mengurutkannya berdasarkan nilai persamaannya. Sistem ini akan dibangun menjadi sebuah alat bantu yang berkaitan dengan masalah pencarian artikel jurnal pada Portal Garuda IPI. Sistem ini dapat mempermudah proses pencarian artikel yang diinginkan serta untuk mempermudah pengelompokkan dokumen berdasarkan nilai indeksnya.

Contoh sederhana berikut dapat digunakan untuk membantu pemahaman mengenai prosedur kerja dari metode *Jaccard Coefficient*, misalkan terdapat dua buah dokumen yang akan dilakukan pembobotan dan pengindeksan. Isi dari kedua dokumen yang digunakan tersebut dapat dirincikan sebagai berikut:

Dokumen 1 = Hari senin adalah jam makan malam

Dokumen 2 = Makan malam adalah dua hari jumat

Kemudian kita memisahkan dua dokumen di atas menjadi array.

[1]hari, [2]senin, [3]adalah, [4]jam, [5]makan, [6]malam, [7]dua, [8]jumat

Berarti kita memiliki dua set yang berbeda yaitu Dokumen 1 dan Dokumen 2.

Dokumen1 = A, Dokumen2 = B

A= {1,2,3,4,5,6} dan B= {5,6,3,7,1,8}

Kemudian kita mencari union dari kedua dokumen tersebut. Union adalah jumlah kata secara keseluruhan dari dua dokumen yang sedang dihitung. Dari array diatas bisa kita lihat bahwa jumlah kata secara keseluruhan adalah 8 kata. Union dari Dokumen 1 dan 2 adalah sebagai berikut :

$A \cup B = \{1,2,3,4,5,6,7,8\}$

Keterangan :

U = Union

A = Dokumen 1

B = Dokumen 2

Setelah kita berhasil mendapatkan hasil union, selanjutnya adalah mencari intersection diantara dua dokumen tersebut. Intersection adalah jumlah kata yang sama dari dua dokumen yang sedang dihitung. Jika dilihat dari Dokumen A dan Dokumen B, ada beberapa kata yang sama dari kedua dokumen tersebut, antara lain : [1]hari, [3]adalah, [5]makan, dan [6]malam.

Intersection dari Dokumen A dan B adalah : $A \cap B = \{1,3,5,6\}$

Keterangan :

N = Intersection

A = Dokumen 1

B = Dokumen 2

Union = 1,2,3,4,5,6,7,8 = 8 Kata

Intersection = 1,3,5,6 = 4 Kata

Langkah selanjutnya adalah menghitung kemiripan dari kedua dokumen tersebut dengan rumus sebagai berikut :

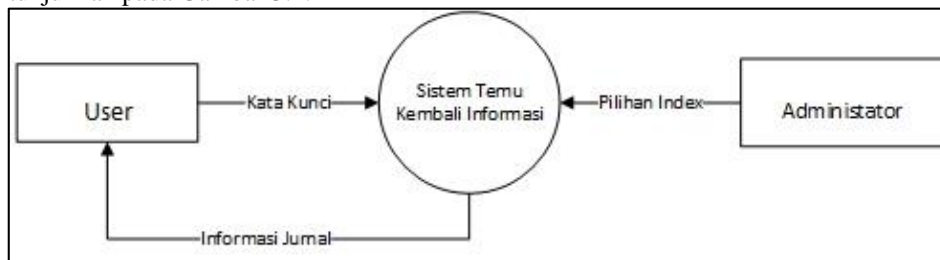
$$Sab = \frac{A \cup B}{A \cap B} = \text{nilai persamaan}$$

$$Sab = \frac{A \cup B}{A \cap B} = \frac{8}{4} = 0,5$$

Berdasarkan nilai persamaan yang diperoleh maka dapat ditetapkan bahwa nilai kemiripan dari dokumen 1 dan dokumen 2 adalah 0,5.

Data flow diagram adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. Berikut adalah gambaran dari data flow diagram Sistem Rekomendasi Pencarian Artikel Jurnal Indonesia.

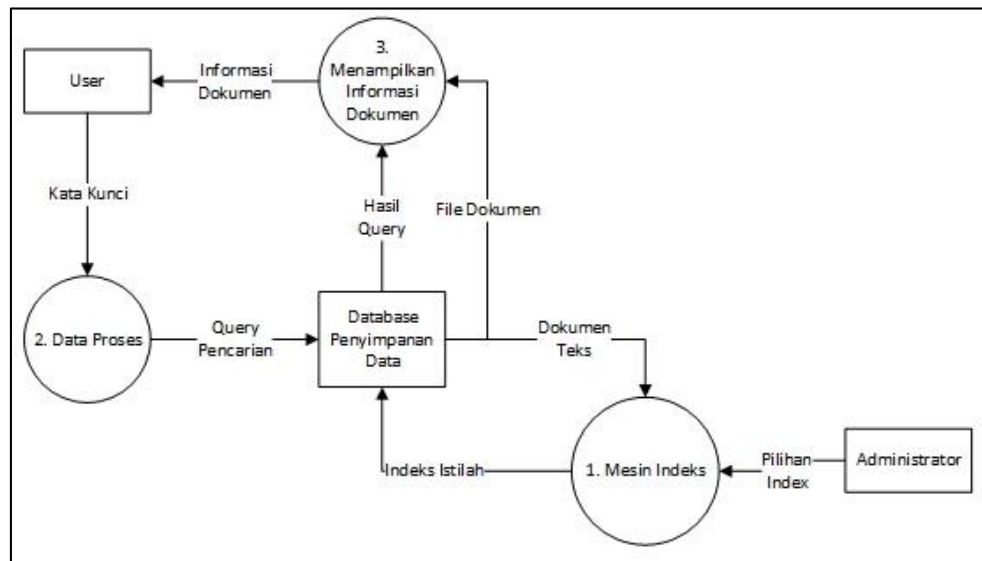
Diagram konteks adalah gambaran awal mengenai sistem rekomendasi pencarian artikel, Konteks diagram ditunjukkan pada Gambar 3.2.



Gambar 3. 2. Diagram konteks Sistem Rekomendasi Pencarian Artikel

Dari diagram konteks maka dapat diturunkan menjadi Data Flow Diagram (DFD) level 1. DFD adalah sebuah teknik grafis yang menggambarkan aliran data yang bergerak dari *input* ke *output*. Selain itu DFD juga menyajikan fungsi-fungsi sistem yang mengolah data *input* dan menghasilkan data *output*. Diagram alur data dapat digunakan untuk menyajikan suatu sistem perangkat lunak pada setiap tingkat abstraksi.

Dari bentuk diagram konteks sistem rekomendasi pencarian artikel, maka bentuk DFD level 1 dari sistem rekomendasi memiliki 2 proses utama dalam sistem rekomendasi pencarian artikel, yaitu proses *Indexing* yang diberi nama dengan mesin indeks dan proses *Query* yang diberi nama dengan Data proses. DFD level 1 ditunjukkan pada gambar 3.3.



Gambar 3. 3 DFD level 1

DFD level 1 memiliki 3 proses yaitu :

1. Mesin Indeks
2. Data Proses
3. Menampilkan Informasi Data

Proses 1 dan 2 merupakan proses utama yang digunakan untuk sistem rekomendasi, sedangkan proses 3 merupakan bagian dari sistem penyimpanan data. Proses 1 merupakan proses *indexing*, sedangkan proses 2 merupakan proses *querying*.

proses DFD Level 1 Proses 1 Mesin Indeks ini dokumen-dokumen yang sudah disimpan pada sistem akan diproses untuk dijadikan indeks istilah yang akan digunakan dalam proses cari. Salah satu *input* dari proses ini merupakan pilihan indeks, yaitu pilihan yang diberikan oleh administrator untuk melaksanakan indeks. Pilihan itu terdiri dari tiga macam, yaitu :

1. Seluruh dokumen, yaitu pilihan bagi administrator untuk mengindeks seluruh dokumen yang disimpan di dalam sistem.
2. Bagian, yaitu pilihan bagi administrator untuk mengindeks seluruh dokumen berdasarkan pada suatu bagian di dalam sistem.
3. Dokumen, yaitu pilihan bagi administrator untuk mengindeks satu atau lebih dokumen tertentu yang terdapat di dalam sistem.

Setelah administrator memilih pilihan indeks, maka sistem akan melakukan pengindeksan berdasarkan pilihan indeks yang dipilih oleh administrator. Selanjutnya dokumen teks sebagai *input* yang kedua akan diproses menjadi istilah indeks serta dihitung nilai bobotnya, lalu disimpan ke dalam basis data. Proses pengolahan teks menjadi indeks istilah akan melalui beberapa tahapan subproses, yaitu pengambilan dokumen teks, *parsing*, penghilangan *stopwords* dan penghitungan nilai bobot pada tiap istilah di dalam indeks. Subproses ini akan dimodelkan pada DFD level 2. Secara umum proses 1 Mesin Indeks ini bertujuan untuk membentuk informasi indeks dari koleksi dokumen sebagai *input* dengan melalui beberapa tahapan.

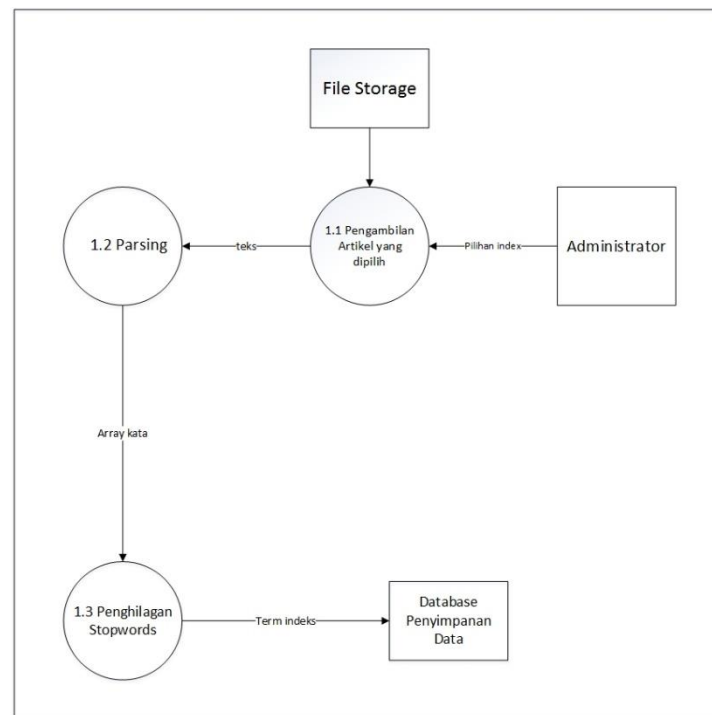
Input pada Proses 1 Proses Cari dari entitas eksternal User adalah kata kunci. Kata kunci ini akan diproses menjadi *query* untuk mendapatkan dokumen sesuai dengan informasi yang ingin dicari oleh pengguna. Proses pada kata kunci untuk diubah menjadi sebuah *query* pada sistem temu-kembali informasi juga memiliki beberapa tahapan yang hampir sama dengan pemrosesan teks pada pengindeksan dokumen yaitu *parsing*, penghilangan *stopwords*, cek frasa serta fungsi kesamaan untuk mengurutkan dokumen dari bobotnya yang paling besar ke yang paling kecil. Tahap pencarian dokumen menggunakan fungsi kesamaan ruang vektor, yang sudah dibahas pada bab sebelumnya untuk mendapatkan kesamaan dokumen dengan *query* yang diberikan oleh pengguna.

Setelah dilakukan *query* pada basis data maka hasil *query* akan diberikan ke Proses 3 Menampilkan Informasi Dokumen yang akan memberikan *output* berupa informasi dokumen dan file kepada pengguna.

DFD Level 2 Proses 1 Mesin Indeks di bagi menjadi 3 sub proses. Akan tetapi proses utama pada mesin indeks ini hanya terdiri dari 2 proses utama yaitu :

1. *Parsing*
2. Penghilangan *Stopwords*

Ketiga proses tersebut yang akan digunakan oleh sistem rekomendasi untuk mengubah teks menjadi *term* indeks untuk digunakan dalam proses pencarian artikel. DFD Level 2 Proses 1 ditunjukkan pada gambar 3.4.



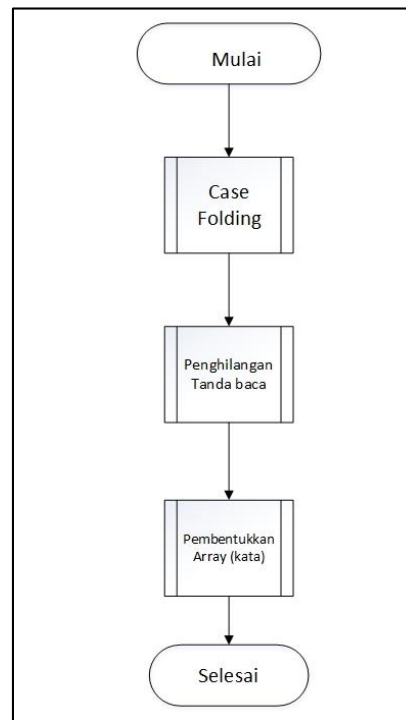
Gambar 3.4 DFD Level 2 Proses 1 Mesin Indeks

Parsing merupakan proses pengenalan kata di dalam rangkaian teks. Input pada proses ini merupakan teks mentah yang masih dalam bentuk paragraf. Nanti teks ini akan diproses dengan menggunakan algoritma *tokenizer* untuk dipecah menjadi unit-unit yang lebih kecil misalnya berupa kata (token).

Tokenizer menerima masukan berupa rangkaian karakter dan memilahnya menjadi token dengan aturan sebagai berikut:

1. Suatu kata dimulai oleh huruf dan angka
2. Kata dipisahkan oleh karakter whitespace (spasi, tabulasi, baris baru, dll)
3. Karakter- karakter khusus yang mengikuti huruf atau angka dianggap bagian dari kata (misalnya tanda persen dalam 125%) namun dianggap sebagai pemisah kata jika tidak.

Awal proses pada *parsing* adalah proses *case folding* yaitu membuat semua huruf pada teks yang akan di-*parsing* menjadi huruf kecil, hal ini dilakukan untuk memperkecil ukuran *database* pada indeks, lalu dilanjutkan dengan pemrosesan penghilangan tanda baca. Selanjutnya akan dilakukan pemisahan teks menjadi array kata. Keluaran yang dihasilkan adalah array kata yang akan digunakan pada pemrosesan selanjutnya. Flow Chart proses parsing ditunjukkan pada gambar 3.5.

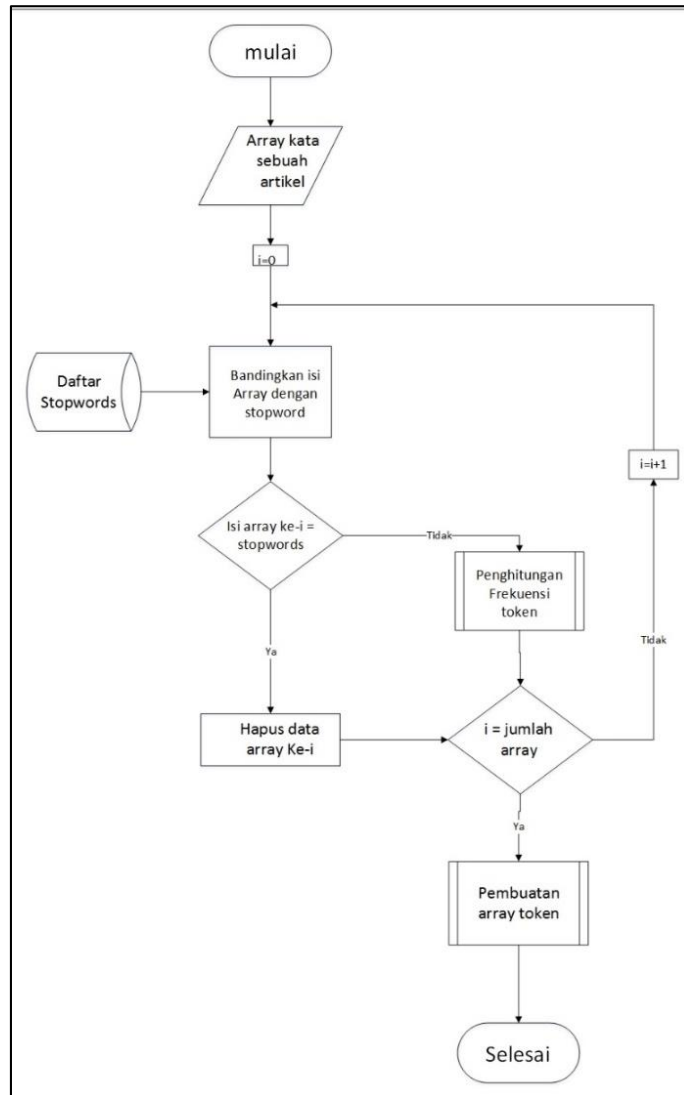


Gambar 3.5 Flow Chart Proses Parsing

Proses penghilangan *stopwords* ini berfungsi sebagai penyaring kata-kata umum yang hampir selalu ada pada dokumen dan tidak signifikan untuk dijadikan indeks suatu dokumen. Kata-kata umum ini telah didefinisikan sebelumnya sehingga pada proses ini hanya membandingkan kata yang merupakan hasil proses *parsing* dengan *stopwords*.

Input pada proses 1.3 Penghilangan *Stopwords* adalah array kata hasil dari proses *parsing*. Algoritmanya adalah sebagai berikut :

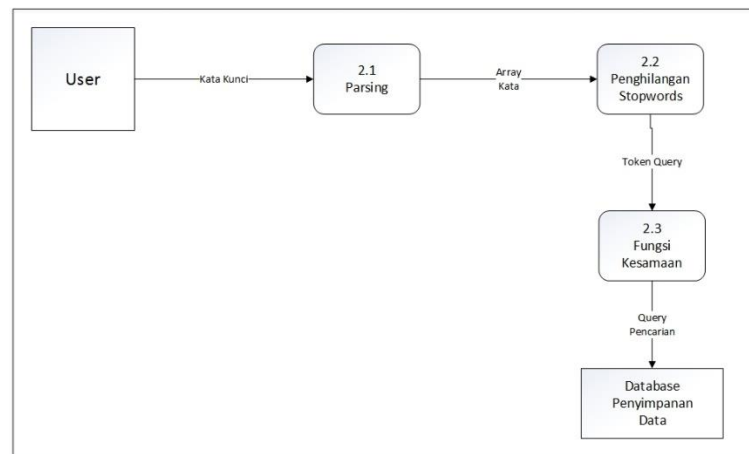
1. Baca array kata yang didapat dari proses *parsing*.
2. Ambil daftar *stopwords*
3. Bandingkan array kata dengan daftar *stopwords*. Jika kata termasuk ke dalam daftar *stopwords*, maka buang kata dari array.
4. Ulangi ke langkah no.3 sampai array kata paling terakhir.



Gambar 3.6 Flowchart Proses 1.3 Penghilangan Stopwords

Hasil dari proses penghilangan ini berupa array token yang nantinya akan digunakan sebagai masukan pada pemrosesan selanjutnya.

Pada DFD Level 2, Proses 2 Proses Cari di pecah menjadi 3 buah subproses. Proses Cari ini merupakan proses pengubahan kata kunci menjadi *query* pencarian yang akan diproses untuk mencari dokumen berdasarkan kata kunci yang dimasukkan oleh pengguna. Dua subproses awal pada proses cari sama dengan subproses pada pengindeksan dokumen yaitu *parsing*, penghilangan *stopwords*, dan fungsi kesamaan DFD Level 2 proses 2 ditunjukkan pada gambar 3.7.

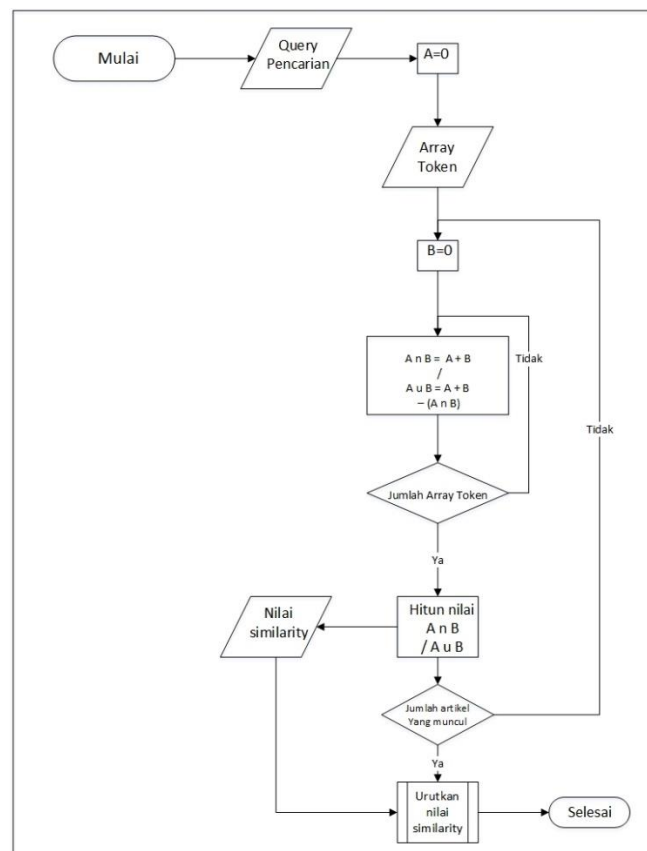


Gambar 3.7 DFD Level 2 Proses 2 Data Proses

Pada DFD Level 2 (Gambar 3.7), Proses 2 Proses Cari di pecah menjadi 4 buah subproses. Proses Cari ini merupakan proses pengubahan kata kunci menjadi *query* pencarian yang akan diproses untuk mencari dokumen berdasarkan kata kunci yang dimasukan oleh pengguna. Dua subproses awal pada proses cari sama dengan subproses pada pengindeksan dokumen yaitu *parsing*, penghilangan *stopwords*, dan fungsi kesamaan.

Algoritma subproses penghilangan *stopwords* dan *parsing* pada Proses Cari ini sama dengan algoritma subproses pada proses Mesin Indeks, sehingga tidak akan dijelaskan lagi.

Setelah parsing dan penghilangan stopwords selesai, maka array tersebut dimasukkan ke dalam table *term_judul* lalu dimasukkan ke dalam rumus fungsi kemiripan untuk dicari nilai kemiripan dari masing- masing artikel berdasarkan *query* pencarian pengguna agar dapat diurutkan. *Flowchart* fungsi kemiripan ditunjukkan pada gambar 3.8.



Gambar 3.8 Proses 2.3 Fungsi Kemiripan

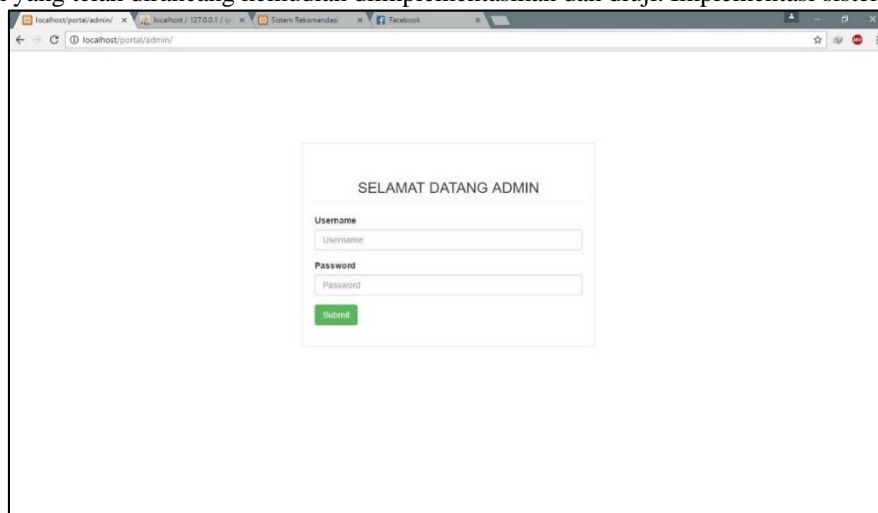
Algoritma dari Proses 2.4 Fungsi Kemiripan adalah sebagai berikut :

1. Ambil array query pencarian.
 2. Ambil array token.
 3. Set $A \cap B = A + B$, Set $A \cup B = A + B - (A \cap B)$
 4. Ulangi dari nomor 2 sampai array token terakhir
 5. Hitung nilai dan simpan nilainya kedalam Nilai *Similarity*
 6. Ulangi dari nomor 1 sampai vektor dokumen terakhir.
 7. Urutkan nilai di dalam nilai *similarity*.
- Fungsi kemiripan disini menggunakan metode *Jaccard Coefficient* dengan rumus yang ditunjukkan pada equation 1.

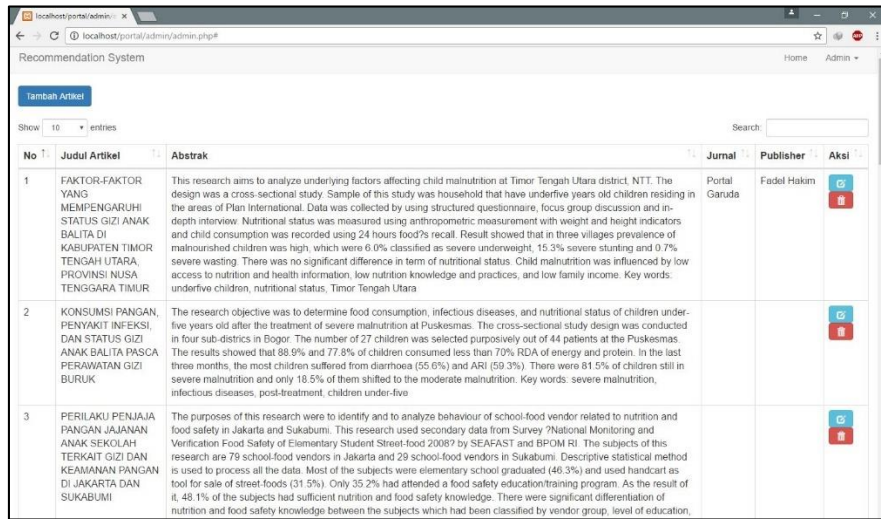
$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \quad (1)$$

4. HASIL DAN ANALISA

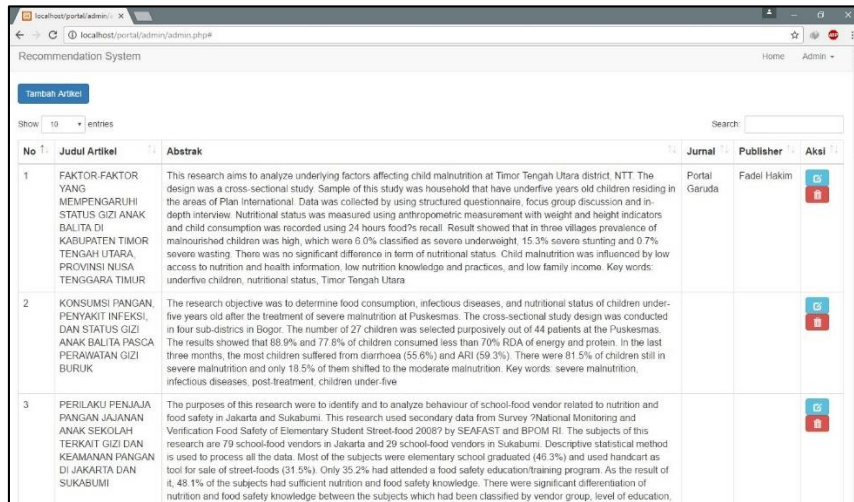
Sistem yang telah dirancang kemudian diimplementasikan dan diuji. Implementasi sistem yaitu :



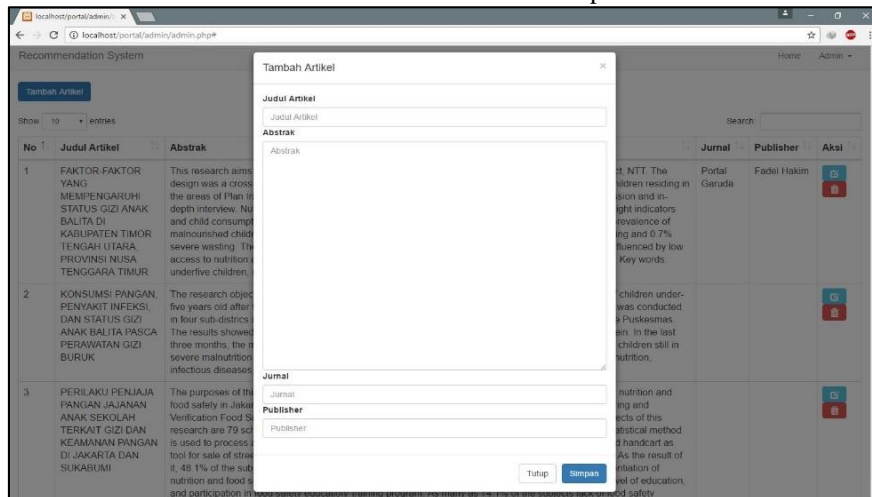
Gambar 4.4 Antarmuka Login Sistem Penyimpanan



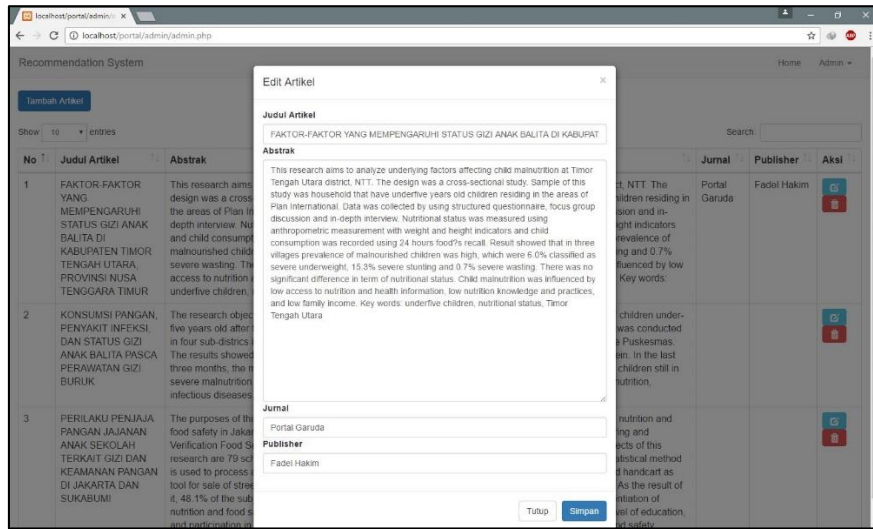
Gambar 4.5 Antarmuka Halaman Admin



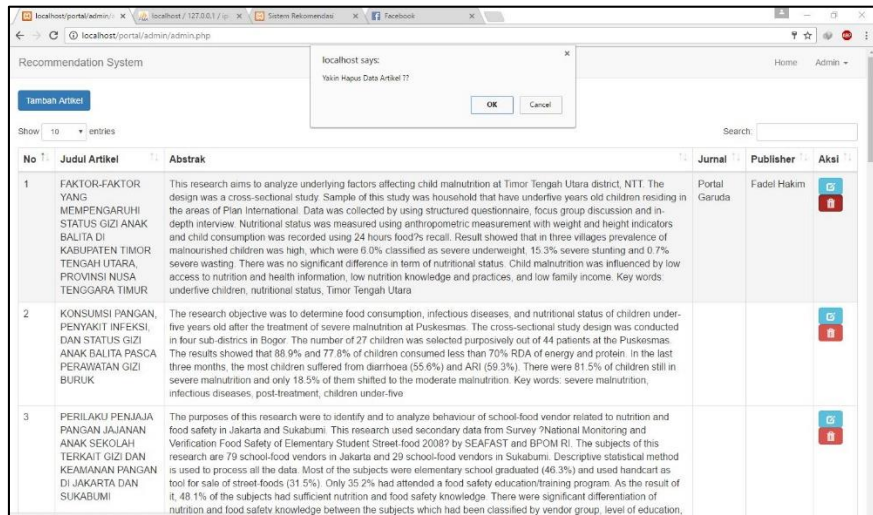
Gambar 4.6 Antarmuka halaman manipulasi data



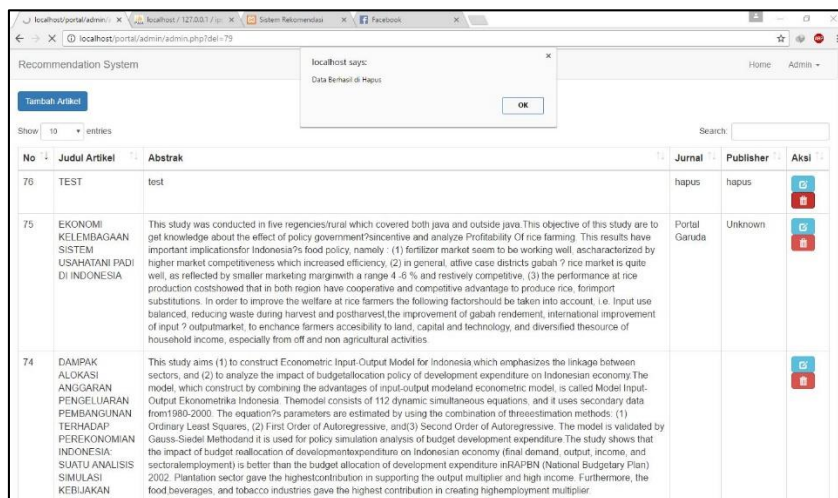
Gambar 4.7 Antarmuka halaman tambah artikel



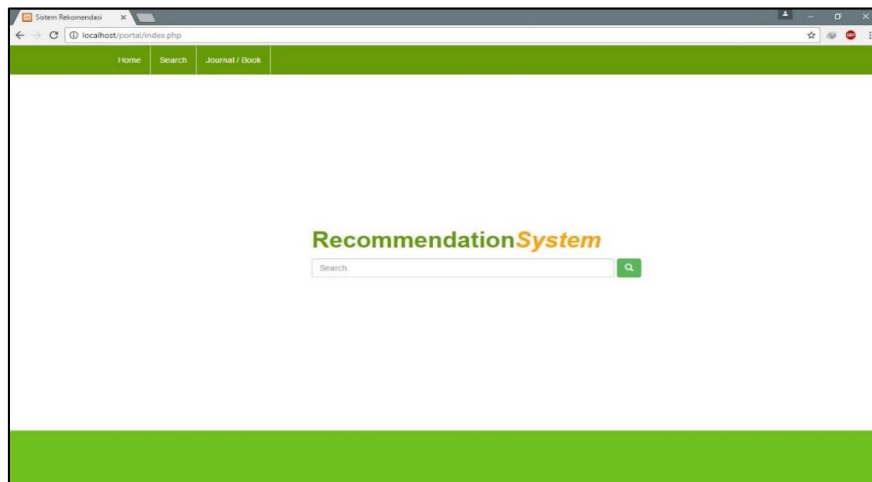
Gambar 4.8 Antarmuka halaman mengubah artikel



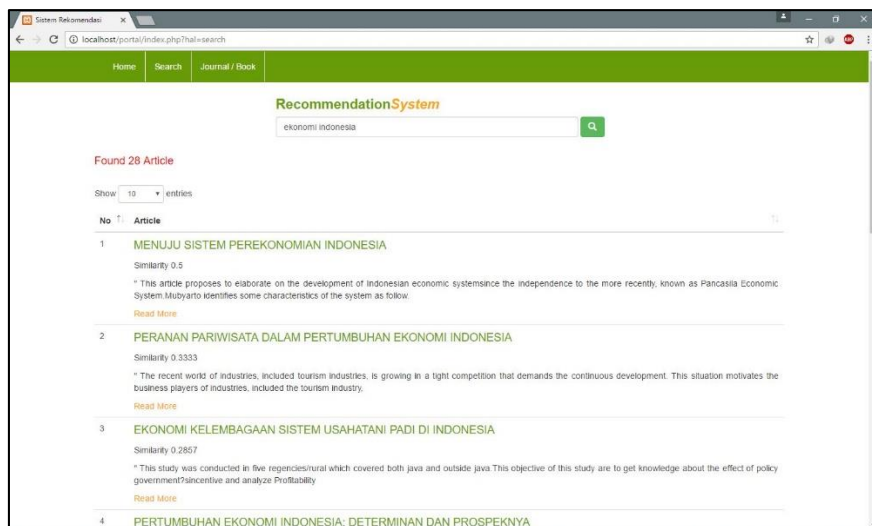
Gambar 4.9 Antarmuka halaman menghapus artikel



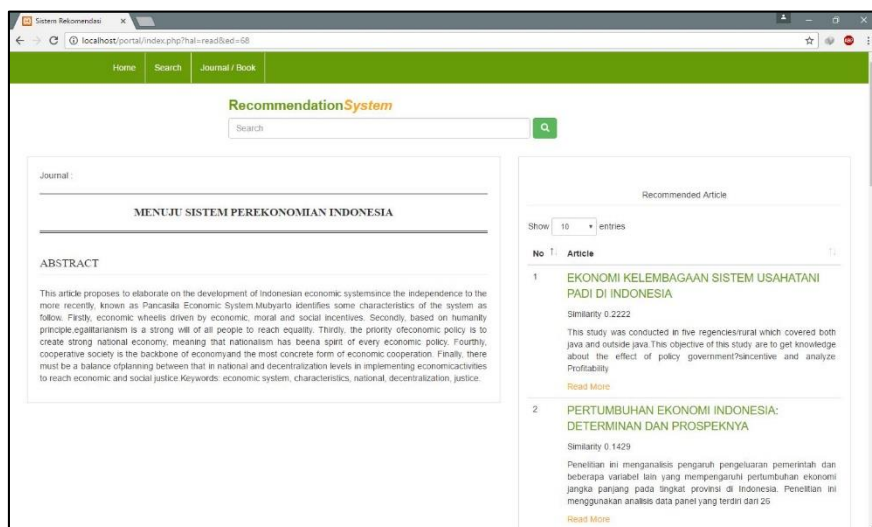
Gambar 4.10 Antarmuka halaman artikel berhasil dihapus



Gambar 4.11 Tampilan halaman beranda



Gambar 4.12 Tampilan halaman hasil pencarian



Gambar 4.13 Tampilan halaman rekomendasi

Dalam tahapan ini, sistem akan dijalankan dan diuji cobakan untuk mengetahui apakah sistem berjalan sesuai dengan hasil analisa dan tujuan yang diharapkan. Untuk mengetahui kemampuan sistem IR yang telah dibangun dalam tugas akhir ini, maka akan dilakukan pengujian dengan mengukur kualitas *retrieval*, yaitu dengan melakukan uji *precision* dan *recall*.

Query yang digunakan adalah:

Tabel 4.2 *Query* yang digunakan dalam pengujian

Ekonomi Indonesia	<i>Query 1</i>
Hukum Laut	<i>Query 2</i>
Gizi Anak	<i>Query 3</i>

Untuk memperoleh nilai *precision* dan *recall*, digunakan Persamaan 2.7, Persamaan 2.8. hasil perhitungan yang dihasilkan oleh metode *Jaccard's Coefficient* berdasarkan *query* yang diinputkan adalah:

1. *Query* : Ekonomi Indonesia

Berdasarkan *Query* yang diinputkan Ditunjukkan bahwa jumlah dokumen yang dikembalikan yang relevan dengan *query* (*tp*) sebanyak 22 dokumen, sedangkan dokumen yang tidak relevan (*fp*) sebanyak 6 dokumen. Dan untuk jumlah dokumen yang tidak dikembalikan yang relevan dengan *query* (*fn*) sebanyak 0 dokumen, sedangkan dokumen yang tidak relevan sebanyak (*tn*) 47 dokumen. Maka, nilai *precision* dan *recall* untuk *query* 1 adalah:

$$\text{Precision} \rightarrow P = tp / (tp + fp) = 22 / (22+6) = 22/28 = 0,79$$

$$\text{Recall} \rightarrow R = tp / (tp + fn) = 22 / (22+0) = 22/22 = 1$$

Tabel 4.3 *Query* "Ekonomi Indonesia"

Keterangan	Relevant	Non Relevant
Retrieved	22	6
Not Retrieved	0	47

2. *Query* : Hukum Laut

Berdasarkan *Query* yang diinputkan ditunjukkan bahwa jumlah dokumen yang dikembalikan yang relevan dengan *query* (*tp*) sebanyak 25 dokumen, sedangkan dokumen yang tidak relevan (*fp*) sebanyak 1 dokumen. Dan untuk jumlah dokumen yang tidak dikembalikan yang relevan dengan *query* (*fn*) sebanyak 0 dokumen, sedangkan dokumen yang tidak relevan sebanyak (*tn*) 49 dokumen. Maka, nilai *precision* dan *recall* untuk *query* 2 adalah:

$$\text{Precision} \rightarrow P = tp / (tp + fp) = 25 / (25+1) = 25/26 = 0,97$$

$$\text{Recall} \rightarrow R = tp / (tp + fn) = 25 / (25+0) = 25/25 = 1$$

Tabel 4.4 *Query* "Hukum Laut"

Keterangan	Relevant	Non Relevant
Retrieved	25	1
Not Retrieved	0	49

3. *Query* : Gizi Anak

Berdasarkan *Query* yang diinputkan, ditunjukkan bahwa jumlah dokumen yang dikembalikan yang relevan dengan *query* (*tp*) sebanyak 25 dokumen, sedangkan dokumen yang tidak relevan (*fp*) sebanyak 2 dokumen. Dan untuk jumlah dokumen yang tidak dikembalikan yang relevan dengan *query* (*fn*) sebanyak 0 dokumen, sedangkan dokumen yang tidak relevan sebanyak (*tn*) 48 dokumen. Maka, nilai *precision* dan *recall* untuk *query* 2 adalah:

$$\text{Precision} \rightarrow P = tp / (tp + fp) = 25 / (25+2) = 25/27 = 0,93$$

$$\text{Recall} \rightarrow R = tp / (tp + fn) = 25 / (25+0) = 25/25 = 1$$

Tabel 4.5 *Query* "Gizi Anak"

Keterangan	Relevant	Non Relevant
Retrieved	25	2
Not Retrieved	0	48

Berdasarkan pengujian yang telah dilakukan, maka dapat diambil kesimpulan bahwa Untuk pengujian *precision* dan *recall* pada Sistem Rekomendasi Menggunakan metode *Jaccard's Coefficient*, persentase kualitas *retrieval* untuk *query* 1 terhadap jumlah dokumen yang berhasil di-*retrieve* oleh sistem, yaitu *precision* 0,79 dan *recall* 1. Sedangkan persentase kualitas *retrieval* untuk *query* 2 berdasarkan jumlah dokumen yang berhasil di-*retrieve*, yaitu *precision* 0,97 dan *recall* 1. Dan persentase kualitas *retrieval* untuk

query 3 berdasarkan jumlah dokumen yang di-*retrieve*, yaitu *precision* 0,93 dan *recall* 1. Dan berdasarkan penelitian yang telah dilakukan, Kerelavan sebuah artikel yang dicari, ditentukan oleh pengguna sendiri. Sistem hanya mengembalikan dokumen yang berhubungan dengan *query* yang diinputkan oleh pengguna.

5. KESIMPULAN

Berdasarkan hasil penelitian yang berupa pengembangan sistem pencarian dengan metode temu-kembali informasi dapat diambil sebuah kesimpulan bahwa Proses pengindeksan artikel didalam sistem yang dikembangkan melalui beberapa tahapan pemrosesan teks, yaitu *parsing*, penghilangan *stopwords* dari setiap judul artikel yang akan dijadikan indeks. Sedangkan untuk .Proses pencariannya juga melalui tahapan proses penghitungan kemiripan antara *query* yang dimasukkan pengguna dengan *term* untuk mendapatkan nilai *similarity* pada tiap artikel dan Sistem Rekomendasi dengan penerapan metode *Jaccard's Coefficient* dapat *retrieve* dokumen yang *relevan* dan tidak *relevan*, sehingga dapat membantu pengguna dalam menemukan dokumen yang dibutuhkan.

DAFTAR PUSTAKA

- [1] F. Amin, *Sistem Temu Kembali Informasi dengan Pemeringkatan Metode Vector Space Model*, vol. 18, no. 2. 2013.
- [2] G. Salton, *Automatic Text Processing, The Transformation, Analysis, and Retrieval of Information by Computer*. Wesley Publishing Company, Inc. All rights reserved, 1989.
- [3] Sugiyamto, B. Surarso, and A. Sugiharto, *Analisa performa metode cosine dan jaccard pada pengujian kesamaan dokumen*, vol. 5. 2014.