

# J-INTECH

Journal of Information and Technology

Volume 05 Nomor 02, Desember Tahun 2017

J-INTECH

Volume 05 Nomor 02, Desember Tahun 2017



**SEKOLAH TINGGI INFORMATIKA & KOMPUTER INDONESIA**

Jl. Raya Tidar 100 Malang, 65146

Telp. (0341)560823, Fax (0341)562525

**STIKI**

ISSN: 2303-1425 E-ISSN: 2580-720X

# J-INTTECH

Journal of Information and Technology  
Volume 05 Nomor 02, Desember 2017



LEMBAGA PENELITIAN DAN PENGABDIAN KEPADA MASYARAKAT

**STIKI**

**SEKOLAH TINGGI INFORMATIKA & KOMPUTER INDONESIA**  
Jl. Raya Tidar 100, Malang; Phone: 0341-560823; Fax: 0341-562525; <http://www.stiki.ac.id>; [mail@stiki.ac.id](mailto:mail@stiki.ac.id)

## **PENGANTAR REDAKSI**

J-INTECH merupakan jurnal yang diterbitkan oleh Sekolah Tinggi Informatika dan Komputer Indonesia Malang guna mengakomodasi kebutuhan akan perkembangan Teknologi Informasi serta guna mensukseskan salah satu program DIKTI yang mewajibkan seluruh Perguruan Tinggi untuk menerbitkan dan mengunggah karya ilmiah mahasiswanya dalam bentuk terbitan maupun jurnal online.

Pada edisi ini, redaksi menampilkan beberapa karya ilmiah mahasiswa yang mewakili beberapa mahasiswa yang lain, yang dianggap cukup baik sebagai media pembelajaran bagi para lulusan selanjutnya.

Tentu saja diharapkan pada setiap penerbitan memiliki nilai lebih dari karya ilmiah yang dihasilkan sebelumnya sehingga merupakan nilai tambah bagi para adik kelas maupun pihak-pihak yang ingin studi atau memanfaatkan karya tersebut selanjutnya.

Pada kesempatan ini kami juga mengundang pihak-pihak dari PTN/PTS lain sebagai kontributor karya ilmiah terhadap jurnal J-INTECH, sehingga Perkembangan IPTEK dapat dikuasai secara bersama-sama dan membawa manfaat bagi institusi masing-masing.

Akhir redaksi berharap semoga dengan terbitnya jurnal ini membawa manfaat bagi para mahasiswa, dosen pembimbing, pihak yang bekerja pada bidang Teknologi Informasi serta untuk perkembangan IPTEK di masa depan.

**REDAKSI**

# J-INTTECH

Journal of Information and Technology  
Volume 05 Nomor 02, Desember 2017

---

## DAFTAR ISI

Sistem Penunjang Keputusan Pemilihan Beasiswa dengan Metode <i>Decision Tree</i> ID3 pada SMAK Kalam Kudus Malang..... <i>Erwin Prasetya Chrisnata</i>	01-12
Sistem Informasi Logistik Berbasis Web di Unit Donor Darah PMI Kota Malang..... <i>Anjang Wijaya</i>	13-16
Sistem Pendukung Keputusan Diagnosa Penyakit Paru-Paru dengan Metode <i>Weighted Product</i> guna Membantu Proses Anamnesa Berbasis <i>Mobile</i> ..... <i>Devi Tri Wahyuningtyas</i>	17-24
Penerapan Metode Bayes <i>Classifier</i> untuk Pradiagnosa Penyakit Tuberculosis ..... <i>Andhika Dwi Indra Irawan</i>	25-31
Sistem Informasi <i>Positioning</i> Samsat Keliling Berbasis Android..... <i>Yosia Prabowo</i>	32-39
Sistem Pendukung Keputusan Penilaian Kinerja Karyawan Menggunakan Metode <i>Weighted Product</i> di PT Makmur Jaya Kharisma ..... <i>Yehezkiel Fernando</i>	40-43
Sistem Penunjang Keputusan Mekanisme Pemilihan Hasil Pertanian dengan Metode Topsis Berbasis Webgis di Dinas Pertanian Kabupaten Malang..... <i>RB. Dandy Raga Utama</i>	44-47
Kontrol Suhu dan Kelembaban pada <i>Green House</i> ..... <i>Rizka Septiandoyo Nugroho</i>	48-53
Aplikasi Pendeteksi Kelayakan Telur Menggunakan Metode <i>Backpropagation</i> dan <i>Thresholding</i> ..... <i>Harman Tunggorono</i>	54-63

Sistem Penunjang Keputusan Penggolongan Keluarga Melalui Posdaya dengan Metode <i>Decision Table</i> Berbasis Webgis.....	64-70
<i>Sephira Elliandini Widodo</i>	
Pemanfaatan <i>Engine</i> Vuforia untuk Implementasi Teknologi <i>Augmented Reality</i> dalam Metode Pembelajaran Sholat Berbasis <i>Mobile</i> .....	71-81
<i>Dawang Mahendra Sudirman Putra</i>	
<i>Prototype</i> Alat Bantu Tuna Netra Berupa Tongkat Menggunakan Arduino dan Sensor Ultrasonik .....	82-90
<i>Charles Setiawan</i>	
Pemanfaatan Corona SDK dalam Perancangan <i>Game</i> Edukasi Matematika Berbasis Android.....	91-103
<i>Rindang Raharjo Rozak</i>	
Optimasi Penjadwalan Kegiatan Belajar Mengajar menggunakan Algoritma Genetika (Studi Kasus: SMKN 8 Malang).....	104-109
<i>Gusti Dani Arianto</i>	
Sistem Pakar Identifikasi Hama dan Penyakit Buah Mangga Menggunakan Metode Inferensi <i>Forward Chaining</i> Berbasis Web.....	110-118
<i>Muhammad Zaidi Efendi</i>	
Implementasi Corona <i>Game Engine</i> untuk <i>Game</i> Edukasi “ <i>Galaxy of Science</i> ” Berbasis Android.....	119-126
<i>Albert Ferento</i>	
<i>Game</i> Tutorial Pengenalan Rambu Rambu Lalu Lintas untuk Anak Sekolah Dasar .....	127-134
<i>L. Danny Adventus Rufus</i>	
Aplikasi Kompetisi Bola Basket Berbasis <i>Mobile</i> (Studi Kasus: STIKI <i>Basketball League</i> ) .....	135-138
<i>Sendi Kurniawaty</i>	
Sistem Penunjang Keputusan untuk Menentukan Barang Terlaris dengan Algoritma Apriori pada CV Calosa Global Indonesia .....	139-146
<i>Septian Widjaya</i>	
Pemanfaatan Sistem Temu Kembali Informasi dalam Pencarian Dokumen Menggunakan Metode <i>Vector Space Model</i> .....	147-153
<i>Ferry Sanjaya</i>	



ISSN: 2303-1425 E-ISSN: 2580-720X

# J-INTECH

Journal of Information and Technology  
Volume 05 Nomor 02, Desember 2017

---

- Pelindung** : Ketua STIKI
- Penasehat** : Puket I, II, III
- Pembina** : Ka. LPPM
- Editor** : Subari, S.Kom, M.Kom
- Section Editor** : Daniel Rudiaman S.,ST, M.Kom
- Reviewer** : Dr. Eva Handriyantini, S.Kom, M.MT.  
Evi Poerbaningtyas, S.Si, M.T.  
Laila Isyriyah, S.Kom, M.Kom  
Anita, S.Kom, M.T.
- Layout Editor** : Nira Radita, S.Pd., M.Pd  
Muh. Bima Indra Kusuma

# Pemanfaatan Sistem Temu Kembali Informasi dalam Pencarian Dokumen Menggunakan Metode *Vector Space Model*

Ferry Sanjaya

Program Studi Teknik Informatika, Sekolah Tinggi Informatika & Komputer Indonesia (STIKI)  
Malang

Email: ferrysanjaya99@gmail.com

## ABSTRAK

Seiring banyaknya dokumen menyebabkan berkembangnya teknologi, penyimpanan dokumen dalam bentuk file semakin banyak digunakan. Akan tetapi, file-file dokumen tersebut akan terus bertambah setiap harinya dan untuk mencari informasi dari isi file-file tersebut akan menjadi sulit. Untuk itu perlu diterapkannya metode ilmu pencarian yang dikenal dengan temu kembali informasi (*Information Retrieval*). Salah satunya metode dalam sistem temu kembali adalah *Vector Space Model*. Pada metode VSM sebelum melakukan pencarian dokumen akan dilakukan *indexing* dengan memecah isi teks dari dokumen-dokumen tersebut menjadi *index term*. *Index term* ini yang akan digunakan untuk proses pencarian. Proses pembentukan *index term* dari teks yang terdapat di dalam dokumen akan melalui beberapa tahapan yaitu *parsing*, *text preprocessing*, penghitungan bobot, dan juga pengukuran kesamaan (*Similarity Measure*).

**Kata Kunci:** *pencarian dokumen, information retrieval, vector space model*

## 1. PENDAHULUAN

Perkembangan jaman hingga saat ini sangat berdampak besar terhadap kehidupan sekarang. Salah satu hal yang berubah adalah cara menggunakan data sebagai informasi pada era *big data* sekarang ini. Dengan semakin banyaknya jumlah dokumen yang beredar saat ini menimbulkan sebuah masalah untuk melakukan pencarian yang diinginkan dengan cepat dan akurat baik pada media online melalui internet (*search engine, e-library*) ataupun offline (sistem penyimpanan pada komputer). meskipun saat ini beberapa *e-library* sudah menerapkan algoritma untuk pencariannya seperti algoritma *boolean search* namun hal itu belum cukup akurat untuk proses pencarian karena tidak dapat mengenali dokumen-dokumen yang paling relevan atau cukup relevan (*pure exact match*).

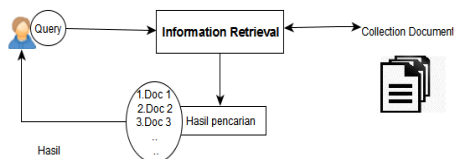
Salah satu metode dalam temu balik informasi (*Information Retrieval*) untuk mendapatkan dokumen yang relevan berdasarkan *query* adalah *Vector Space Model*. *Vector Space Model (VSM)* adalah metode untuk mengetahui tingkat kedekatan atau kesamaan (*similarity*) term dengan cara pembobotan *term*. Dokumen dipandang sebagai sebuah vektor yang memiliki *magnitude* (jarak) dan *direction* (arah). Metode *Vector Space Model* ini merepresentasikan suatu dokumen dan *query* dalam sebuah bentuk vektor. Relevansi

sebuah dokumen ke sebuah *query* didasarkan pada kesamaan di antara vektor dokumen dan vektor *query*. Dalam merepresentasikan dalam sebuah vektor dibutuhkan bobot *term* dari dokumen ataupun *query*. *Term* dapat berupa kata, frase, atau unit hasil *indexing* lain dalam suatu dokumen sebagai gambaran dari isi setiap dokumen tersebut.

Dalam penentuan *term* pada suatu dokumen ataupun *query* diperlukan beberapa tahapan antara lain *filtering*, *stemming* dan *tokenizing*. Setiap *term* tentunya memiliki tingkat kepentingan yang berbeda dalam dokumen untuk itulah diperlukan *term weighting* (pembobotan *term*). Metode pembobotan yang umumnya digunakan dalam *Vector Space Model* yaitu (TF-IDF) *Term Frequency Inverse Document Frequency*. Metode TF-IDF merupakan suatu cara untuk memberikan bobot hubungan suatu kata (*term*) terhadap dokumen. Dari hasil pembobotan dengan metode TF-IDF ini nantinya sebuah dokumen dan *query* bisa direpresentasikan dalam sebuah ruang vektor yang kemudian akan dicari tingkat kedekatannya dengan menggunakan pengukuran *cosine similarity* sehingga mendapatkan dokumen yang paling relevan dengan suatu *query*.

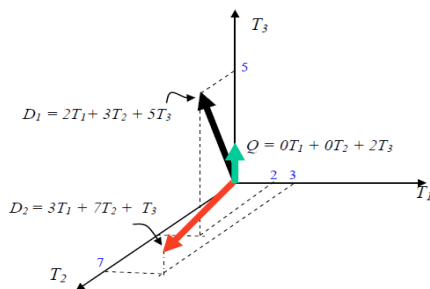


*Information Retrieval* adalah aktifitas utama yang dilakukan oleh sebuah penyedia informasi atau pusat pelayanan informasi, termasuk perpustakaan dan jenis dari layanan lainnya yang menyediakan informasi. Sistem temu kembali informasi tidak memberitahu (yakni tidak mengubah pengetahuan) pengguna mengenai masalah yang ditanyakannya. Sistem tersebut hanya memberitahukan keberadaan (atau ketidakberadaan) dan keterangan dokumen-dokumen yang berhubungan dengan permintaannya.



**Gambar 1. Sistem Temu Kembali Informasi**

Pada *Information Retrieval System* terdapat beberapa metode yang digunakan dalam proses pencarian salah satunya adalah dengan merepresentasikan proses pencarian adalah menggunakan model ruang vektor. Model ruang vektor dibuat berdasarkan pemikiran bahwa isi dari dokumen ditentukan oleh kata-kata yang digunakan dalam dokumen tersebut. Model ini menentukan kemiripan (*similarity*) antara dokumen dengan *query* dengan cara merepresentasikan dokumen dan *query* masing-masing ke dalam bentuk vektor. Tiap kata yang ditemukan pada dokumen dan *query* diberi bobot dan disimpan sebagai salah satu elemen vektor.



**Gambar 2. Ilustrasi Dokumen dan Query dalam ruang vektor**

*Text preprocessing* atau sering disebut juga proses *indexing*, merupakan tahapan awal pada proses merepresentasikan koleksi dokumen kedalam bentuk tertentu untuk

memudahkan dan mempercepat proses pencarian dan penemuan kembali dokumen yang relevan. Pembangunan index dari koleksi dokumen merupakan tugas pokok pada tahapan *preprocessing* di dalam IR. Kualitas index mempengaruhi efektifitas dan efisiensi sistem IR. Terdapat beberapa tahapan pada fase ini antara lain:

- *Case Folding dan Tokenization, Case folding* dilakukan untuk merubah huruf besar dari setiap kata diubah menjadi huruf kecil dan menghilangkan karakter selain huruf seperti angka dan tanda baca (*delimiter*). Sedangkan *tokenization* yaitu memecah semua kalimat pada isi dokumen menjadi kata per kata.
- *Filtering*, dilakukan dengan metode stop words yaitu menghilangkan semua kata yaitu kata sambung, kata depan, kata ganti, dan lain-lain. Sebagian contoh dalam bahasa Indonesia: yang, juga, dari, dia, kami, kamu, aku, saya, ini, itu, atau, dan, tersebut, pada, dengan, adalah, yaitu, ke, tak, tidak, di, pada, jika, maka, ada, pun, lain, saja, hanya, namun, seperti, kemudian, karena, untuk, dan lain-lain.
- *Stemming*, digunakan untuk mengubah term yang masih melekat dalam term tersebut awalan, sisipan, dan akhiran. Proses stemming dilakukan dengan cara menghilangkan semua imbuhan (*affixes*) baik yang terdiri dari awalan (*prefixes*), sisipan (*infixes*), akhiran (*suffixes*) dan *confixes* (kombinasi dari awalan dan akhiran) pada kata turunan. stemming digunakan untuk mengganti bentuk dari suatu kata menjadi kata dasar dari kata tersebut yang sesuai dengan struktur morfologi bahasa Indonesia.

Metode TF-IDF merupakan metode pembobotan *term* yang banyak digunakan sebagai metode pembandingan terhadap metode pembobotan baru. Pada metode ini, perhitungan bobot *term t* dalam sebuah dokumen dilakukan dengan mengalikan nilai *Term Frequency* dengan *Inverse Document Frequency*.

$$W = tf_{ij} \times idf_j$$

$$W = tf_{ij} \times \log\left(\frac{D}{df_i}\right)$$

Keterangan:

$W$  adalah bobot term  $t_j$  terhadap dokumen  $d_i$

$tf_{ij}$  adalah jumlah kemunculan term  $t_j$  dalam dokumen  $d_i$

$D$  adalah jumlah semua dokumen yang ada

$df_j$  adalah jumlah dokumen yang mengandung term  $t_j$  (minimal ada satu kata yaitu term  $t_j$ )

*Vector Space Model* dan pembobotan TF-IDF digunakan untuk merepresentasikan nilai numerik dokumen sehingga kemudian dapat dihitung kedekatan antar dokumen. Semakin dekat dua vektor di dalam suatu VSM maka semakin mirip dua dokumen yang diwakili oleh vektor tersebut. Kemiripan antar dokumen dihitung menggunakan suatu fungsi ukuran kemiripan (*similarity measure*). Ukuran ini memungkinkan perankingan dokumen sesuai dengan kemiripan relevansinya terhadap *query*.

$$Sim(\vec{d}_j, \vec{q}) = \frac{\sum_{i=1}^t (W_{ij} \times W_{iq})}{\sqrt{\sum_{i=1}^t (W_{ij})^2 \times \sum_{i=1}^t (W_{iq})^2}}$$

Keterangan:

$D_j$  :Dokumen ke  $j$

$Q$  :*query* user

$\sum_{i=1}^t W_{ij}$  :jumlah bobot kata  $i$  pada dokumen  $j$

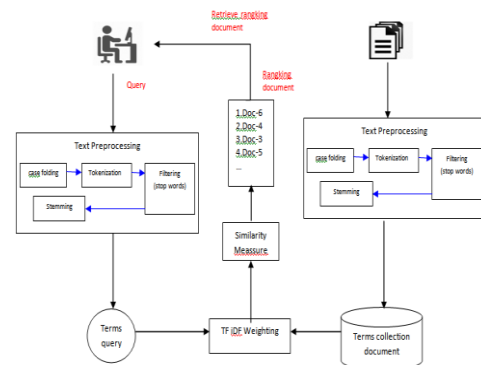
$\sum_{i=1}^t W_{iq}$  :jumlah bobot kata  $i$  pada *query*

## 2. RANCANGAN SISTEM

### A. Analisa Data

Proses pencarian dalam sistem temu kembali informasi pada penelitian ini diterapkan pada sistem penyimpanan dokumen \*.pdf berupa jurnal dengan topik Teknologi Informasi. Dari masing-masing dokumen akan dilakukan *indexing* berdasarkan pada semua isi dokumen.

## Arsitektur sistem temu kembali informasi

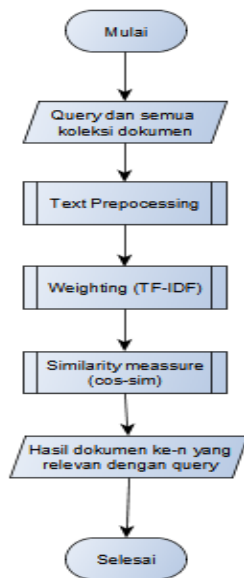


**Gambar 3. Arsitektur Sistem Temu Kembali Informasi**

Pada sistem temu kembali dokumen dan juga *query* user akan dilakukan *text preprocessing* dan juga pembobotan (*Weighting*) untuk memperoleh nilai kemiripannya. *Text preprocessing* pada dokumen akan disimpan dalam database yang berupa *index term*. Hal ini dilakukan untuk membuat proses pencarian kembali lebih efisien.

### Perancangan Proses

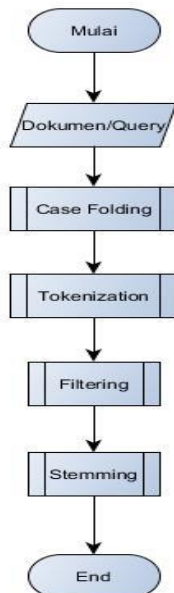
Pada perancangan proses dilakukan untuk menjelaskan bagaimana proses yang dikerjakan sistem dalam melakukan pencarian dokumen terhadap *query* dari *user*. Data yang digunakan dokumen file digital dengan ekstensi \*.pdf. Proses awal dalam pencarian dokumen adalah melakukan *indexing (text preprocessing)* untuk mendapatkan bobot dari koleksi dokumen dan juga *query* pencarian dari *user*. Gambar 3.2. menjelaskan rancangan alur sistem secara umum.



Gambar 4. Flowchart Perancangan Sistem

### B. Text Preprocessing

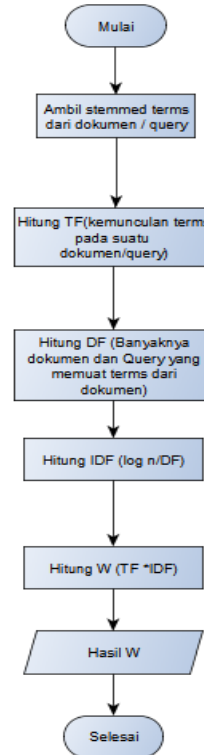
Pada tahapan *text preprocessing* ini dilakukan untuk memperoleh *index terms* dari dokumen maupun *query* yang nantinya akan digunakan untuk pembobotan. Langkah-langkah dari tahapan preprocessing adalah *case folding*, *tokenization*, *filtering*, *stemming*. Tahap *preprocessing* dijelaskan pada gambar 5.



Gambar 5. Flowchart Prosedur Text Preprocessing

### C. Pembobotan (Weighting)

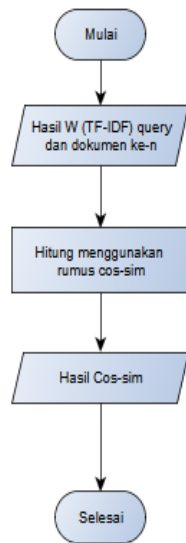
Tahapan pembobotan (*term weighting*) ini dilakukan dengan metode TF-IDF. Dengan menghitung *Term Frequency* (TF), yaitu frekuensi kemunculan suatu kata muncul dalam satu dokumen dan juga *Inverse Document Frequency* (IDF), frekuensi kemunculan suatu kata di dalam seluruh dokumen. Penjelasan mengenai proses pembobotan (*term weighting*) TF-IDF dijelaskan pada gambar 6.



Gambar 6. Flowchart Pembobotan

### Similarity Measure

Ukuran kemiripan (*similarity measure*) ini menggunakan algoritma *Cosine similarity*. *Cosine similarity* merupakan algoritma digunakan untuk mengukur kedekatan antara dua vektor yaitu vektor *query* dan dokumen. Tahapan ini merupakan tahap akhir untuk memperoleh hasil dokumen yang relevan dengan *query*. Alur proses perhitungan ukuran kemiripan adalah sebagai berikut:



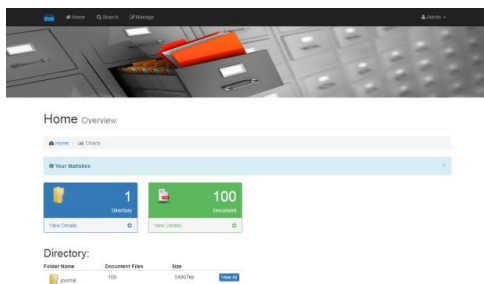
Gambar 7. Flowchart Similarity Measure

### 3. IMPLEMENTASI DAN PEMBAHASAN

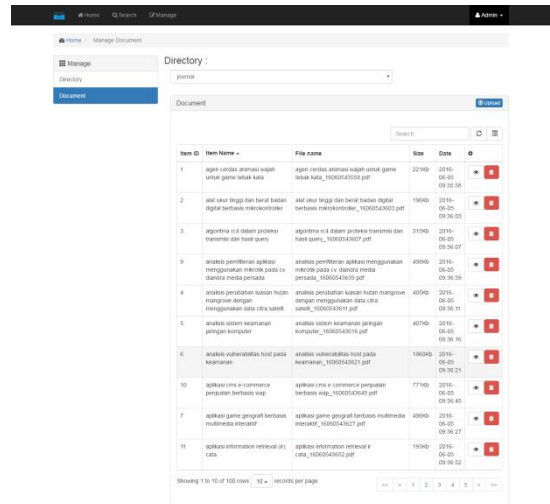
#### A. Implementasi

Dalam implementasi pembuatan program ini menggunakan bahasa pemrograman interpreter PHP versi 5.4. Selain itu juga dibutuhkan library pendukung lain untuk implementasi VSM seperti *pdfparser*, *stemming sastrawi*(Nazief and Andriani). Untuk mendukung tampilan dan interaksi user digunakan *bootstrap* dan juga *jQuery*.

Pada sistem ini terdapat role admin sebagai pengelola koleksi dokumen. Berikut merupakan beberapa halaman dari admin.

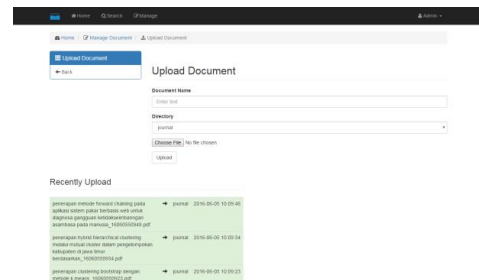


Gambar 8. Halaman Home Admin



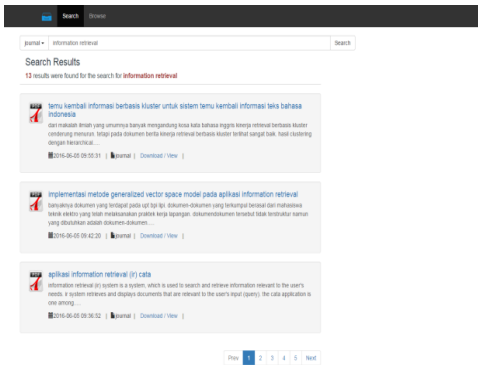
Gambar 9. Halaman Manage Dokumen

Pada sistem temu kembali terdapat proses *indexing* untuk menghasilkan *index term* hal ini dilakukan sistem saat admin melakukan penambahan dokumen pada halaman upload dokumen. Pada proses *indexing* inilah fase *text preprocessing* dilakukan.



Gambar 10. Halaman Upload Dokumen

Sedangkan proses *weighting* dan *similarity measure* dilakukan saat pengunjung melakukan pencarian dokumen. Dimana dari pencarian tersebut sistem dapat mengenali dokumen berdasarkan tingkat relevansinya dan dikembalikan kepada pengunjung. Dokumen yang dianggap paling relevan akan berada di urutan paling atas dari hasil pencarian dokumen.



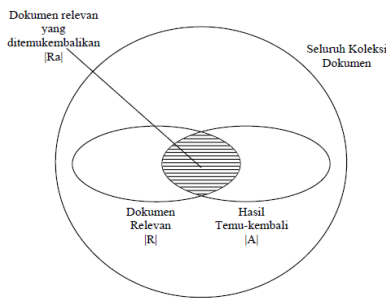
**Gambar 11. Halaman Pencarian Dokumen Oleh Pengunjung**

**B. Pembahasan**

Untuk pembahasan dalam proses pencarian dengan menerapkan *information retrieval* perlu dilakukan uji coba. Uji coba dilakukan untuk mengetahui kelebihan dan kelemahan dari metode *VSM*.

**Recall dan Precision**

Salah satu pengujian sistem temu kembali adalah dengan pengujian *recall* dan *precision*. *Precision* dapat dianggap sebagai ukuran ketepatan atau ketelitian, sedangkan *recall* adalah kesempurnaan. Nilai *precision* adalah proporsi dokumen yang terambil oleh sistem adalah relevan. Nilai *recall* adalah proporsi dokumen relevan yang terambil oleh sistem



**Gambar 12. Gambaran Recall dan Precision**

Berikut ini merupakan rumus dari perhitungan recall dan precision:

$$R = \frac{\text{number of relevant items retrieved} \text{ } |RA|}{\text{Total number relevant items in collection} \text{ } |R|}$$

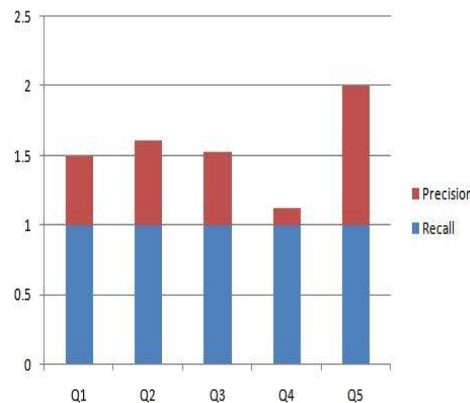
$$P = \frac{\text{number of relevant items retrieved} \text{ } |RA|}{\text{Total number of items retrieved} \text{ } |A|}$$

**Hasil Uji Recall dan Precision**

Untuk mengetahui hasil uji *recall* dan *precision* sistem akan dilakukan uji terhadap 6 *query* acak. Data dokumen pada sistem berjumlah 100 dokumen dan setelah dilakukan proses indexing menghasilkan 13103 *terms* pada database. Berikut ini merupakan daftar *query* yang dilakukan dalam pengujian.

**Tabel 1. Hasil Uji Recall dan Precision**

<i>Query</i>	Recall	Precision
Metode fuzzy (Q1)	1	0.5
Fuzzy (Q2)	1	0.61
Information retrieval (Q3)	1	0.53
Sistem informasi geografis (Q4)	1	0.125
Implementasi mikrokontroller arduino (Q5)	1	1



**Gambar 13. Hasil Pengujian Recall dan Precision**

Terhadap beberapa hasil uji *recall* dan *precision* terhadap 5 *query* dapat dilihat bahwa sistem dapat mengembalikan semua dokumen yang dianggap relevan (*recall*) tetapi mendapatkan hasil ketepatan rata-rata (*average precision*) 0.44.

Nilai *recall* dan *precision* bernilai antara 0 sd 1. Sistem temu kembali informasi diharapkan untuk dapat memberikan nilai *recall* dan *precision* mendekati 1. Pengguna rata-rata ingin mencapai nilai *recall* tinggi dan *precision* tinggi, pada kenyataannya hal itu harus dikompromikan karena sulit dicapai.

Tentunya nilai *recall* dan *precision* diatas bergantung pada dokumen dan juga *query*. Dalam penentuan nilai dari recall dan

precision harus didapatkan jumlah dokumen yang relevan terhadap suatu topik informasi. Satu satunya cara untuk mendapatkannya yaitu dengan membaca dokumen itu satu per satu.

#### 4. KESIMPULAN

Pemanfaatan sistem temu kembali pada proses pencarian yang telah dibuat telah dapat mencari file dokumen berdasarkan isi dokumen.

1. Hasil pencarian dokumen pada sistem dapat mengembalikan hampir semua dokumen yang relevan jika dokumen tersebut ada. ( $recall = 1$ )
2. Hasil pengujian didapatkan nilai *precision* yang rendah hal ini dipengaruhi oleh banyaknya variasi term dalam data karena sistem melakukan *indexing* dari semua kata yang ada dalam dokumen. Semakin banyak term yang dihasilkan sebuah dokumen dalam proses indexing maka semakin sering dokumen tersebut dikembalikan oleh sistem terhadap *query* yang sebenarnya kurang relevan (*low precision*). Tetapi hasil tersebut telah diatasi pada sistem dengan memberi nilai *threshold* pada nilai perhitungan kemiripan (*similarity*).
3. *Term frequency* dan *Inverse document frequency* berperan penting dalam hasil perangkaian dokumen yang dikembalikan. Dengan kata lain hasil rangking dokumen sangat dipengaruhi oleh data term pada dokumen dan juga *query*.

#### 5. REFERENSI

- [1] Baeza, Yates., Ribeiro, Neto. (1999). Modern Information Retrieval, Harlow, Addison-Wesley.
- [2] Fatkhul, A. (2012). Sistem Temu Kembali Informasi dengan Metode Vector Space Model. Jurnal Fakultas Teknologi Informasi. Universitas Stikubank. Semarang.
- [3] Handojo, A, dkk. (2014). Document Searching Engine Using Term Similarity Vector Space Model on English and Indonesian Document. Jurnal Informatics Engineering Department Faculty of Industrial Technology, Petra Christian University Surabaya, Indonesia.
- [4] Manwar, A.B. (2005). A Vector Space Model For Information Retrieval: A Matlab Approach, Indian Journal of Computer Science and Engineering (IJCSE), Department of Computer Science S.G.B. Amravati University, Amravati MS, India.
- [5] Rijsbergen, C.J. van. (1979). Information Retrieval, Second Edition. Butterworths, London.
- [6] Salton, G. (1969). Automatic Text Analysis. Technical Report No. 69-36. Department of Computer Science. Cornell University, Ithaca, New York.