Pemanfaatan *Vector Space* Model pada Penerapan Algoritma Nazief Adriani, KNN dan Fungsi *Similarity Cosine* untuk Pembobotan IDF dan WIDF pada *Prototipe* Sistem Klasifikasi Teks Bahasa Indonesia

Diki Susandi[1], Usep Sholahudin[2]
Program Studi Teknik Informatika – Universitas Serang Raya
Jln. Raya Cilegon Serang – Drangong Kota Serang
[1]unsera.diky@gmail.com
[2]s.usep@yahoo.com

Abstrak - ***Vector space model (VSM)* adalah suatu model yang digunakan untuk mengukur kemiripan antara suatu dokumen dengan suatu *query*. Pada model ini, *query* dan dokumen dianggap sebagai vektor-vektor pada ruang n-dimensi, dimana n adalah jumlah dari seluruh *term* yang ada di dalam daftar. Teknologi informasi khususnya internet sangat mendukung terjadinya pertukaran informasi dengan sangat cepat. Kondisi tersebut memunculkan masalah untuk mengakses informasi yang diinginkan secara akurat dan cepat. Untuk mengatasi masalah tersebut, salah satu teknik yang dapat digunakan adalah dengan mengklasifikasikan teks dokumen tersebut sesuai dengan karakteristik, fitur, maupun kelasnya berdasarkan aturan baku bahasa yang akan diolah. Dalam penelitian ini Bahasa Indonesia adalah bahasa yang digunakan sebagai sumber acuan. Jenis penelitian ini termasuk kepada penelitian terapan (*Applied Research*). Objek dalam penelitian ini adalah dokumen Teks Berbahasa Indonesia. Tujuan dari penelitian ini menganalisis efektifitas model sistem klasifikasi / kategorisasi dokumen dalam penerapan *vector space model* berdasarkan pembobotan *term* dokumen dan *query*, juga menerapkan metode *stemming* Bahasa Indonesia dengan algoritma *nazief adriani*, menghasilkan nilai *similarity* dengan fungsi *cosine* yang berpengaruh pada pemeringkatan hasil kategorisasi dokumen yang relevan.**

Kata Kunci: ***Vector Space Model*, Algoritma *Nazief Adriani*, Fungsi *Similarity Cosine*, Algoritma *K-Nearest Neighbor*, Klasifikasi Dokumen Bahasa Indonesia, Pembobotan *term* Dokumen**

I. PENDAHULUAN

Informasi saat ini sangat mudah didapatkan oleh setiap orang dimanapun berada. Teknologi informasi khususnya internet sangat mendukung terjadinya pertukaran informasi dengan sangat cepat. Internet menjadi media informasi dan komunikasi yang telah dimanfaatkan banyak orang dengan banyak kepentingan. Informasi yang berkualitas dipengaruhi oleh relevansi, keakuratan dan tepat waktu. Meskipun demikian, belum banyak tersedia mesin pencari yang efektif bagi pengguna Bahasa Indonesia untuk menggali informasi dari halaman-halaman web Berbahasa Indonesia. Portal-portal Internet yang menyediakan sarana pencarian dokumen pada umumnya menggunakan teknologi komersial yang tersedia di pasaran dan diperuntukkan bagi Bahasa Inggris. Tanpa disadari, tidak adanya mesin pencari yang dapat mengolah dokumen dan informasi Berbahasa Indonesia secara efektif yang telah menimbulkan dampak tersendiri bagi pengguna Internet yang tidak pasif Berbahasa Inggris.

Begitupun dalam penyimpanan dokumen secara digital semakin meningkat. Kondisi tersebut memunculkan masalah untuk mengakses informasi yang diinginkan secara akurat dan cepat. Oleh karena itu, walaupun sebagian besar dokumen digital tersimpan dalam bentuk teks dan berbagai algoritma yang efisien untuk pencarian teks telah dikembangkan, namun kesulitan menemukan suatu dokumen yang berhubungan dengan suatu kata kunci tertentu dengan hasil yang tepat dan akurat masih terjadi. Pencarian terhadap seluruh isi dokumen yang tersimpan bukanlah solusi yang tepat mengingat pertumbuhan ukuran data yang tersimpan umumnya.

Dengan semakin banyak dan beragamnya informasi yang tersedia, kebutuhan pengguna internet telah bergeser dari arah kuantitatif ke arah kualitatif. Kebutuhan yang semula berupa informasi sebanyak-banyaknya telah bergeser menjadi informasi secukupnya sejauh hasil yang dihasilkan oleh sistem merupakan hasil yang relevan dengan keperluan. Kebutuhan akan suatu mekanisme pencarian dokumen yang lebih efektif dan relevan dirasakan semakin mendesak.

Seringkali pada web, dimana kita mencari suatu informasi tertentu, banyak hal yang penting justru terlewatkan, malah yang tidak penting banyak terserap. Untuk

mengatasi masalah tersebut, salah satu teknik yang dapat digunakan adalah dengan mengklasifikasikan teks tersebut sesuai dengan karakteristik, fitur, maupun kelasnya berdasarkan aturan baku bahasa yang akan diolah, dalam penelitian ini Bahasa Indonesia yang digunakan sebagai sumber acuan.

Didasari alternatif tersebut, maka dalam penelitian ini akan dibangun suatu aplikasi perangkat lunak yang dapat melakukan klasifikasi data teks terhadap sumber informasi teks elektronik yang diunggah secara terpandu dan selektif. Metode yang digunakan untuk mendukung proses klasifikasi ini adalah *Algoritma Nazief Adriani*, *K-Nearest Neighbor* dan Fungsi *Similarity Cosine*.

Tujuan hasil dari penelitian ini adalah membuat model sistem untuk mengklasifikasikan dokumen teks dalam Bahasa Indonesia yang relevan sesuai dengan kebutuhan pengguna sistem, serta mendapatkan model pembobotan *query* yang tepat untuk penggunaan sistem dalam upaya untuk membetuk sistem pengklasifikasian dokumen yang efektif.

II. LANDASAN TEORI

A. *Data Mining*

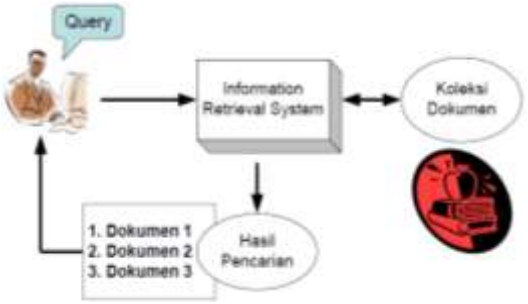
Text mining disebut juga dengan *text data mining* adalah suatu proses ekstraksi pola berupa informasi dan pengetahuan yang berguna dari sejumlah besar sumber data teks, seperti dokumen word, pdf, dan kutipan teks. *Text mining* mencari pola-pola yang ada diteks dalam bahasa natural yang tidak terstuktur seperti buku, email, artikel, halaman web. Kegiatan yang biasa dilakukan oleh text mining adalah *text categorization*, *text clustering*, *conception / entityextraction*.[1]

Dalam memberikan solusi, *text mining* mengadopsi dan mengembangkan banyak teknik dari bidang lain, seperti *Data Mining*, *Information Retrieval* (IR), *Statistic and Mathematic*, *Machine Learning*, *Linguistic*, *Natural Languange Processing* (NLP) dan *Visualization*. Kegiatan riset untuk *text mining* antara lain ekstraksi dan penyimpanan teks, *preprocessing* akan konten teks, pengumpulan data statistik dan *indexing* serta analisa konten.

B. *Information Retrieval (IR)*

Information retieval adalah ilmu untuk menemukan material yang umumnya merupakan dokumen-dokumen yang ditujukan untuk memenuhi kebutuhan informasi dari pemakai (*user*).[3]

Information retrieval (IR) *system* digunakan untuk menemukan kembali (*retrieve*) informasi-informasi yang relevan terhadap kebutuhan pengguna dari suatu kumpulan informasi secara otomatis.

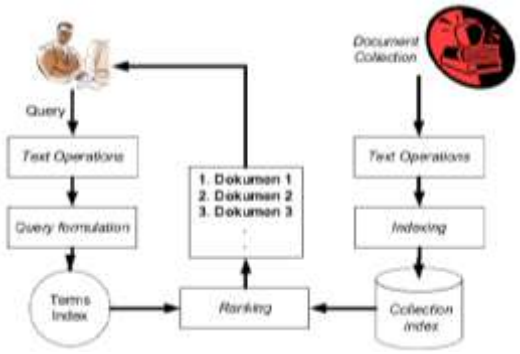


Gambar 1. Ilustrasi IR *System*

Gambar 1 menjelaskan tentang proses yang terjadi di dalam *information retrieval* system terdiri dari 2 bagian utama, yaitu *indexing subsystem* dan *searching subsystem* (*matching system*). Proses *indexing* dilakukan untuk membentuk basis data terhadap koleksi dokumen yang dimasukkan atau dengan kata lain, *indexing* merupakan proses persiapan yang dilakukan terhadap dokumen sehingga dokumen siap untuk diproses. Proses *indexing* sendiri meliputi 2 proses, yaitu dokumen *indexing* dan *term indexing*. Dari term *indexing* akan dihasilkan koleksi kata yang akan digunakan untuk meningkatkan performance pencarian pada tahap selanjutnya.

Information retrieval system terutama berhubungan dengan pencarian informasi yang isinya tidak memiliki struktur. Demikian pula ekspresi kebutuhan pengguna yang disebut *query*, juga tidak memiliki struktur. Hal ini yang membedakan *information retrieval system* dengan sistem basis data. Dokumen adalah contoh informasi yang tidak terstruktur. Isi dari suatu dokumen sangat tergantung pada pembuat dokumen tersebut.

Sebagai suatu sistem, *information retrieval system* memiliki beberapa bagian yang membangun sistem secara keseluruhan. Gambaran bagian-bagian yang terdapat pada suatu *information retrieval system* digambarkan sebagai berikut:



Gambar 2. Skema IR *System*

Gambar 2 memperlihatkan bahwa terdapat dua buah alur operasi pada *information retrieval system*. Alur pertama dimulai dari koleksi dokumen dan alur kedua dimulai dari *query* pengguna. Alur pertama yaitu pemrosesan terhadap koleksi dokumen menjadi basis data indeks tidak tergantung pada alur kedua. Sedangkan alur kedua tergantung dari

keberadaan basis data indeks yang dihasilkan pada alur pertama.

Bagian-bagian dari information retrieval system menurut gambar 2 meliputi:

1. *Text Operations* (operasi terhadap teks) yang meliputi pemilihan kata-kata dalam query maupun dokumen (*termselection*) dalam pentransformasian dokumen atau *query* menjadi *term index* (indeks dari kata-kata).
2. *Query formulation* (formulasi terhadap *query*) yaitu memberi bobot pada indeks kata-kata *query*.
3. *Ranking* (perangkingan), mencari dokumen-dokumen yang relevan terhadap *query* dan mengurutkan dokumen tersebut berdasarkan kesesuaiannya dengan *query*.
4. *Indexing* (pengindeksan), membangun basis data indeks dari koleksi dokumen. Dilakukan terlebih dahulu sebelum pencarian dokumen dilakukan.

C. Klasifikasi Teks

Klasifikasi adalah proses penemuan model (atau fungsi) yang menggambarkan dan membedakan kelas data atau konsep yang bertujuan agar bisa digunakan untuk memprediksi kelas dari objek yang label kelasnya tidak diketahui.[6]

Algoritma klasifikasi yang banyak digunakan secara luas, yaitu *Decision / Classification Trees*, *Bayesian Classifiers / Naive Bayes Classifiers*, *Neural Networks*, Analisa Statistik, Algoritma Genetika, *Rough Sets*, *K-Nearest Neighbor*, *Metode Rule Based*, *Memory Based Reasoning* dan *Support Vector Machines* (SVM).

Pengklasifikasian teks sangat dibutuhkan dalam berbagai macam aplikasi, terutama aplikasi yang jumlah dokumennya bertambah dengan cepat. Ada dua cara dalam penggolongan teks, yaitu *clustering teks* dan klasifikasi teks. *Clustering teks* berhubungan dengan menemukan sebuah struktur kelompok yang belum kelihatan (tak terpandu atau *unsupervised*) dari sekumpulan dokumen. Sedangkan pengklasifikasian teks dapat dianggap sebagai proses untuk membentuk golongan golongan (kelas-kelas) dari dokumen berdasarkan pada kelas kelompok yang sudah diketahui sebelumnya (terpandu atau *supervised*).

D. Vector Space Model

Salah satu model matematika yang digunakan pada sistem temu-kembali informasi untuk menentukan bahwa sebuah dokumen itu relevan terhadap sebuah informasi adalah *vector space model* (VSM). Model ini akan menghitung derajat kesamaan antara setiap dokumen yang disimpan di dalam sistem dengan *query* yang diberikan oleh pengguna. Model ini pertama kali diperkenalkan oleh Salton.[10]

Vector space model adalah suatu model yang digunakan untuk mengukur kemiripan antara suatu dokumen dengan suatu *query*. Pada model ini, *query* dan dokumen dianggap sebagai vektor-vektor pada ruang n-dimensi, dimana n adalah jumlah dari seluruh term yang ada dalam *leksikon*. *Leksikon* adalah daftar semua term yang ada dalam indeks. Salah satu cara untuk mengatasi hal tersebut dalam *model vector space* adalah dengan cara melakukan perluasan vektor. Proses perluasan dapat dilakukan pada vektor *query*, vektor dokumen atau pada kedua vektor tersebut.

Pada VSM, setiap dokumen dan *query* dari pengguna direpresentasikan sebagai ruang vektor berdimensi n. Biasanya digunakan nilai bobot istilah (*term weigthing*) sebagai nilai dari vektor pada dokumen nilai 1 untuk setiap istilah yang muncul pada vektor *query*.

E. Algoritma K-Nearest Neighbor

Algoritma *K-Nearest Neighbor* adalah suatu metode yang menggunakan algoritma *supervised*. Perbedaan antara *supervised learning* dengan *unsupervised learning* adalah pada *supervised learning* bertujuan untuk menemukan pola baru dalam data dengan menghubungkan pola data yang sudah ada dengan data yang baru. Sedangkan pada *unsupervised learning*, data belum memiliki pola apapun, dan tujuan *unsupervised learning* untuk menemukan pola dalam sebuah data.

Tujuan dari algoritma *K-Nearest Neighbor* adalah untuk mengklasifikasi objek baru berdasarkan atribut dan *training samples*. Dimana hasil dari sampel uji yang baru diklasifikasikan berdasarkan mayoritas dari kategori pada KNN. Pada proses pengklasifikasian, algoritma ini tidak menggunakan model apapun untuk dicocokkan dan hanya berdasarkan pada memori. Algoritma KNN menggunakan klasifikasi ketetanggaan sebagai nilai prediksi dari sampel uji yang baru.

F. Stemming dan Algoritma Nazief Adriani

Stemming merupakan bagian yang tidak terpisahkan dalam *information retrieval* (IR). Tidak banyak algoritma yang dikhususkan untuk stemming Bahasa Indonesia dengan berbagai keterbatasan didalamnya. Algoritma *porter* salah satunya, algoritma ini membutuhkan waktu yang lebih singkat dibandingkan dengan *stemming* menggunakan algoritma *nazief adriani*, namun proses *stemming* menggunakan algoritma *porter* memiliki presentase keakuratan (presisi) lebih kecil dibandingkan dengan *stemming* menggunakan algoritma *nazief adriani*. Algoritma *nazief adriani* sebagai algoritma *stemming* untuk teks Berbahasa Indonesia yang memiliki kemampuan presentase keakuratan (presisi) lebih baik dari algoritma lainnya. Algoritma ini sangat dibutuhkan dan menentukan dalam proses IR dalam dokumen Indonesia.[2]

Stemming adalah salah satu cara yang digunakan untuk meningkatkan performa IR dengan cara mentransformasi kata-kata dalam sebuah dokumen teks ke bentuk kata dasarnya. Proses *stemming* pada teks Berbahasa Indonesia lebih rumit / kompleks karena terdapat variasi imbuhan yang harus dibuang untuk mendapatkan *rootwiord* (kata dasar) dari sebuah kata. Pada umumnya kata dasar pada Bahasa Indonesia terdiri dari kombinasi:

$$prefiks1 + prefiks2 + KataDasar + sufiks1 + sufiks2 \quad (1)$$

Algoritma *nazief adriani* merupakan sebuah algoritma untuk mencari sebuah kata dasar atau lebih dikenal dengan

istilah *stemming*. Perlu diketahui sebelumnya, bahwa untuk membuat algoritma *nazief adriani* ini membutuhkan sebuah list kata dasar, sehingga bisa menggunakan bantuan *database* atau *array* pada sebuah program, dimana list kata dasar Bahasa Indonesia disimpan di dalam *database*.

Algoritma *nazief adriani* ini memiliki beberapa function utama seperti dibawah ini:

1. *Function* Cek Kata Dasar (*string*)
2. *Function* Hapus Akhiran (*string*)
3. *Function* Hapus Akhiran Kepunyaan (*string*)
4. *Function* Hapus_derivation_prefix (*string*)
5. *Function* Stemming (*string*)

G. Tokenisasi

Tokenisasi adalah pemotongan *string input* berdasarkan tiap kata yang menyusunnya. Pemecahan kalimat menjadi kata-kata tunggal dilakukan dengan menscan kalimat dengan pemisah (*delimiter*) *whitespace* (spasi, tab, dan *new line*).[9]

Secara garis besar *tokenisasi* adalah tahap memecah sekumpulan karakter dalam suatu teks kedalam satuan kata. Sekumpulan karakter tersebut dapat berupa karakter *whitespace*, seperti enter, tabulasi, spasi. Namun untuk karakter petik tunggal („), titik (.), semikolon (;), titik dua (:) atau lainnya, juga dapat memiliki peran yang cukup banyak sebagai pemisah kata. Sebuah titik (.) biasanya untuk tanda akhir kalimat, tapi dapat juga muncul dalam singkatan, inisial orang, alamat internet, dll. Kemudian tanda *hypen* (-) biasanya muncul untuk menggabungkan dua token yang berbeda untuk membentuk token tunggal. Tapi dapat pula ditemukan untuk menyatakan rentang nilai, kata berulang, dsb. Atau karakter *slash* (/) sebagai pemisah file atau direktori atau *url* ataupun untuk menyatakan "dan atau".

Tokenisasi merupakan proses pemotongan kumpulan karakter menjadi sebuah kata tunggal atau token.

H. Filtering

Filtering adalah mengambil kata-kata penting dari hasil token. Bisa menggunakan algoritma *stoplist* (membuang kata yang kurang penting) atau *wordlist* (menyimpan kata penting). *Stoplist* / *stopword* adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan *bag-of-words*. Contoh *stopwords* adalah "yang", "dan", "di", "dari" dan seterusnya.[9]

Filtering yaitu proses pembuangan *stopword* yang dimaksudkan untuk mengetahui suatu kata masuk kedalam *stopword* atau tidak. Pembuangan *stopword* adalah proses pembuangan *term* yang tidak memiliki arti atau tidak relevan. *Term* yang diperoleh dari tahap tokenisasi dicek dalam suatu daftar *stopword*, apabila sebuah kata masuk didalam daftar *stopword* maka kata tesebut akan masuk keproses berikutnya.

I. Stopword

Proses pembuangan *stopword* dimaksudkan untuk mengetahui suatu kata masuk ke dalam *stopword* atau tidak. Pembuangan *stopword* adalah proses pembuangan *term* yang tidak memiliki arti atau tidak relevan. Term yang diperoleh dari tahap tokenisasi dicek dalam suatu daftar *stopword*, apabila sebuah kata masuk di dalam daftar *stopword* maka kata tersebut tidak akan diproses lebih lanjut. Sebaliknya apabila sebuah kata tidak termasuk di dalam daftar *stopword* maka kata tersebut akan masuk ke proses berikutnya. Daftar *stopword* tersimpan dalam suatu tabel, dalam penelitian ini menggunakan daftar *stopword* yang merupakan *stopword* Bahasa Indonesia yang berisi kata-kata seperti ; ini, itu, yang, ke, di, dalam, kepada, dan seterusnya.

Contoh *stopword* yang lain dalam Bahasa Indonesia yaitu : juga, dari, dia, kami, kamu, aku, saya, dan, tersebut, pada, dengan, adalah, yaitu, tak, tidak, pada, jika, maka, ada, pun, lain, saja, hanya, namun, seperti, kemudian, dll.

J. Pembobotan Istilah (Term Weighting)

1) Term Frequency dan Inverse Document Frequency (TF-IDF):

Term Frequency merupakan salah satu metode untuk menghitung bobot tiap *term* dalam text. Dalam metode ini, tiap *term* diasumsikan memiliki nilai kepentingan yang sebanding dengan jumlah kemunculan *term* tersebut pada text. Bobot sebuah *term* t pada sebuah text d dirumuskan dalam persamaan berikut:

$$W(d,t) = TF(d,t) \quad (2)$$

Dimana TF (d,t) adalah *term frequency* dari *term* t di text d. *Term frequency* dapat memperbaiki nilai *recall* pada *information retrieval*, tetapi tidak selalu memperbaiki nilai *precision*. Hal ini disebabkan *term* yang *frequent* cenderung muncul di banyak text, sehingga *term-term* tersebut memiliki kekuatan *diskriminatif* / keunikan yang kecil. Untuk memperbaiki permasalahan ini, *term* dengan nilai frekuensi yang tinggi sebaiknya dibuang dari *set term*.[4]

Jika *term frequency* fokus pada kemunculan *term* dalam sebuah text, *Inverse Document Frequency* (IDF) fokus pada kemunculan *term* pada keseluruhan koleksi text. Pada IDF, *term* yang jarang muncul pada keseluruhan koleksi *term* dinilai lebih berharga. Nilai kepentingan tiap *term* diasumsikan berbanding terbalik dengan jumlah text yang mengandung *term* tersebut. Nilai IDF sebuah *term* t dirumuskan dalam persamaan berikut:

$$IDF(t) = \log(\frac{N}{df(t)}) \quad (3)$$

Di mana N adalah total jumlah text / dokumen pada koleksi dan df(t) adalah jumlah dokumen yang mengandung *term* t. Persamaan ini mengacu pada definisi Salton. IDF dapat memperbaiki nilai *precision*, karena mengkhususkan fokus pada sebuah *term* dalam keseluruhan dokumen. Penelitian belakangan ini telah mengkombinasikan TF dan IDF untuk menghitung bobot *term* dan menunjukkan bahwa gabungan keduanya menghasilkan performansi yang lebih baik. Kombinasi bobot dari sebuah *term* t pada text d didefinisikan sebagai berikut:

$$TF - IDF(d,t) = TF(d,t).IDF(t) \quad (4)$$

2) Weighted Inverse Document Frequency (WIDF):

Metode pembobotan kata yang digunakan selain TF-IDF adalah *Weighted Inverse Document Frequency* (WIDF). WIDF merupakan sebuah metode pengembangan dari metode *Inverse Document Frequency* (IDF).

Tabel I. *Example of Collection*.

	d1	d2	d3	d4	d5
.	.	.	.	.	.
.	.	.	.	.	.
t_x	3	5	2	1	2
t_y	2	2	1	5	3
.	.	.	.	.	.
.	.	.	.	.	.

Metode WIDF menghitung faktor 1/df(t) dengan *term frekuensi*. Sebagai contoh, 1/df(t) dari tabel I diatas menjadi berikut untuk semua text:

$$\frac{1}{1+1+1+1+1} \quad (5)$$

Kemudian angka "1" diganti dengan frekuensi dari masing-masing *term*, menjadi berikut pada kasus d2.

$$\frac{5}{3+5+2+1+1} \quad (6)$$

Faktor inilah yang disebut WIDF, tidak seperti IDF, WIDF dapat membedakan masing-masing text d1……..d5. Sehingga WIDF dengan *term* t dalam *text* d dapat dituliskan sebagai persamaan berikut:

$$WIDF(d,t) = \frac{TF(d,t)}{\sum_{i \in D} TF(i,t)} \quad (7)$$

Dimana TF(d,t) adalah *term frequency* dari *term* t didalam text d dan i menyatakan jumlah text. WIDF dari *term* t menjumlahkan semua *term frequency* dari semua kumpulan text. Dengan kata lain WIDF adalah bentuk normalisasi *term frequency* dari semua kumpulan text.[13]

K. Fungsi Similarity Cosine

Kesamaan antara dokumen Di dengan dokumen Dj dapat diukur dengan fungsi similaritas (mengukur kesamaan) atau fungsi jarak (mengukur ketidaksamaan). Beberapa fungsi similaritas dan fungsi jarak yang dapat dijumpai antara lain adalah *Dice, Jaccard, Euclidean Distance, Pearson Correlation* dan *Cosine Similarity*.[5]

Pada penelitian ini akan digunakan fungsi similaritas yaitu fungsi *Cosine* yang selanjutnya disebut *Cosine Coefficient*. *Cosine Coefficient* merupakan metode yang digunakan untuk menghitung tingkat kesamaan (*similarity*) antar dua buah objek. Untuk tujuan klastering dokumen, fungsi yang baik adalah fungsi *Cosine Coefficient*.[10]

Untuk notasi himpunan dapat digunakan rumus sebagai berikut:

$$sim(D_i, D_j) = \frac{\sum_{k=1}^{d} D_{ik} . D_{jk}}{\sqrt{\sum_{k=1}^{d} D_{ik}^2 . \sum_{k=1}^{d} D_{jk}^2}} \quad (8)$$

III. PEMBAHASAN

A. Pengkategorian Data

Pada penelitian ini percobaan dilakukan terhadap 9 dokumen sampel Berbahasa Indonesia yang terbagi ke dalam 3 bentuk kategori, masing-masing kategori terdapat 3 dokumen yang telah dikonversi dalam bentuk file .pdf yang selanjutnya dilakukan proses *stemming* sehingga menghasilkan satu *dataset* yang di dalamnya terdapat beberapa *term* kata.

Adapun kategori yang telah ditentukan beserta masing-masing dokumen sampelnya adalah sebagai berikut:

1. Kategori Hukum, dokumen sampel yang terkait sebagai berikut:
   a. Implementasi UU Tentang Tindak Pidana Korupsi Terhadap Para Pejabat Negara. (d1)
   b. Permasalahan Keadilan Terhadap Tindak Pidana Pada Anak Di Bawah Umur. (d2)
   c. Tinjauan Kriminologis Tindak Pidana Penyalahgunaan Narkotika Oleh Remaja. (d3)
2. Kategori Kesehatan, dokumen sampel yang tekait sebagai berikut:
   a. Gambaran Pelayanan Kesehatan Di Wilayah Kerja Puskesmas. (d4)
   b. Hubungan Pemberian Imunisasi BCG Dengan Kejadian Tuberkulosis Paru Pada Anak Balita. (d5)
   c. Pengaruh Pendidikan Kesehatan Tentang Hipertensi Kehamilan. (d6)
3. Kategori Pendidikan, dokumen sampel yang terkait sebagai berikut:
   a. Pendekatan Jenis Dan Metode Penelitian Pendidikan. (d7)
   b. Pengaruh Kebijakan Sekolah Gratis Terhadap Prestasi Belajar. (d8)
   c. Pengaruh Kualitas Pembelajaran Guru Terhadap Prestasi Belajar. (d9)

B. Prototipe Sistem Klasifikasi Dokumen

1) Tampilan Halaman Login:

Gambar 3. Halaman *Login*

2) Tampilan Halaman Data Admin:

Gambar 4. Halaman Data *Admin*

3) Tampilan Halaman Data Kategori:

Gambar 5. Halaman Data Kategori

4) Tampilan Halaman Learning Document:

Gambar 6. Halaman *Learning Document*

5) Tampilan Halaman Klasifikasi Dokumen Dengan TF.IDF:

Gambar 7. Halaman Klasifikasi By TF.IDF

6) Tampilan Halaman Klasifikasi Dokumen Dengan WIDF:

Gambar 8. Halaman Klasifikasi By WIDF

C. Hasil Perhitungan Dengan Fungsi Cosine

1) Hasil Perhitungan Berdasarkan Pembobotan TF-IDF (K= 3):

Tabel II. Hasil Perangkingan Q1

Hasil Perangkingan	Kategori	Relevan
Q1,d1	Hukum	Ya
Q1,d2	Hukum	Ya
Q1,d8	Pendidikan	Tidak

Tabel III. Hasil Perangkingan Q2

Hasil Perangkingan	Kategori	Relevan
Q2,d2	Hukum	Ya
Q2,d8	Pendidikan	Tidak
Q2,d3	Hukum	Ya

Tabel IV. Hasil Perangkingan Q3

Hasil Perangkingan	Kategori	Relevan
Q3,d3	Hukum	Ya
Q3,d2	Hukum	Ya
Q3,d6	Kesehatan	Tidak

Tabel V. Hasil Perangkingan Q4

Hasil Perangkingan	Kategori	Relevan
Q4,d4	Kesehatan	Ya
Q4,d5	Kesehatan	Ya
Q4,d6	Kesehatan	Ya

Tabel VI. Hasil Perangkingan Q5

Hasil Perangkingan	Kategori	Relevan
Q5,d5	Kesehatan	Ya
Q5,d6	Kesehatan	Ya
Q5,d7	Pendidikan	Tidak

Tabel VII. Hasil Perangkingan Q6

Hasil Perangkingan	Kategori	Relevan
Q6,d6	Kesehatan	Ya
Q6,d7	Pendidikan	Tidak
Q6,d5	Kesehatan	Ya

Tabel VIII. Hasil Perangkingan Q7

Hasil Perangkingan	Kategori	Relevan
Q7,d7	Pendidikan	Ya
Q7,d6	Kesehatan	Tidak
Q7,d8	Pendidikan	Ya

Tabel IX. Hasil Perangkingan Q8

Hasil Perangkingan	Kategori	Relevan
Q8,d8	Pendidikan	Ya
Q8,d9	Pendidikan	Ya
Q8,d2	Hukum	Tidak

Tabel X. Hasil Perangkingan Q9

Hasil Perangkingan	Kategori	Relevan
Q9,d9	Pendidikan	Ya
Q9,d8	Pendidikan	Ya
Q9,d2	Hukum	Tidak

2) Hasil Perhitungan Berdasarkan Pembobotan WIDF (K= 3):

Tabel XI. Hasil Perangkingan Q1

Hasil Perangkingan	Kategori	Relevan
Q1,d1	Hukum	Ya
Q1,d2	Hukum	Ya
Q1,d8	Pendidikan	Tidak

Tabel XII. Hasil Perangkingan Q2

Hasil Perangkingan	Kategori	Relevan
Q2,d2	Hukum	Ya
Q2,d1	Hukum	Ya
Q2,d3	Hukum	Ya

Tabel XIII. Hasil Perangkingan Q3

Hasil Perangkingan	Kategori	Relevan
Q3,d3	Hukum	Ya
Q3,d2	Hukum	Ya
Q3,d7	Kesehatan	Tidak

Tabel XIV. Hasil Perangkingan Q4

Hasil Perangkingan	Kategori	Relevan
Q4,d4	Kesehatan	Ya
Q4,d7	Pendidikan	Tidak
Q4,d8	Pendidikan	Tidak

Tabel XV. Hasil Perangkingan Q5

Hasil Perangkingan	Kategori	Relevan
Q5,d5	Kesehatan	Ya
Q5,d6	Kesehatan	Ya
Q5,d9	Pendidikan	Tidak

Tabel XVI. Hasil Perangkingan Q6

Hasil Perangkingan	Kategori	Relevan
Q6,d6	Kesehatan	Ya
Q6,d5	Kesehatan	Ya
Q6,d9	Pendidikan	Tidak

Tabel XVII. Hasil Perangkingan Q7

Hasil Perangkingan	Kategori	Relevan
Q7,d7	Pendidikan	Ya
Q7,d5	Kesehatan	Tidak
Q7,d6	Kesehatan	Tidak

Tabel XVIII. Hasil Perangkingan Q8

Hasil Perangkingan	Kategori	Relevan
Q8,d8	Pendidikan	Ya
Q8,d1	Hukum	Tidak
Q8,d9	Pendidikan	Ya

Tabel XIX. Hasil Perangkingan Q9

Hasil Perangkingan	Kategori	Relevan
Q9,d9	Pendidikan	Ya
Q9,d8	Pendidikan	Ya
Q9,d6	Kesehatan	Tidak

D. Pengujian Precision dan Recall

Untuk mengetahui pembobotan mana yang lebih baik dalam melakukan pengkategorian dokumen berdasarkan fungsi *similarity cosine* dilakukan dengan cara pengujian ketepatan dan kelengkapan menggunakan perhitungan nilai *precision* dan *recall*. Dimana persamaan untuk nilai *precision* dan *recall* sebagai berikut:

$$precision = \frac{jlhdokumenrelevanterambil}{jlhdokumenterambildalampencarian} \quad (9)$$

$$recall = \frac{jlhdokumenrelevanterambil}{jlhdokumenterambildalamdatabase} \quad (10)$$

Bila dilihat berdasarkan hasil perangkingan pada pembobotan TF-IDF dan pada pembobotan WIDF, dengan menggunakan nilai KNN sama dengan 3, maka didapatkan perhitungan sebagai berikut:

Tabel XX. *Precision* dan *Recall*

Query	*Precision*		*Recall*	
	TF-IDF	WIDF	TF-IDF	WIDF
Q1	67%	67%	67%	67%
Q2	67%	100%	67%	100%
Q3	67%	67%	67%	67%
Q4	100%	33%	100%	33%
Q5	67%	67%	67%	67%
Q6	67%	67%	67%	67%
Q7	67%	33%	67%	33%
Q8	67%	67%	67%	67%
Q9	67%	67%	67%	67%

Dari hasil pengujian dokumen *query* dengan masing-masing dokumen sampel dengan menggunakan pembobotan TF-IDF dan WIDF didapatkan rata-rata perhitungan nilai *precision* dan *recall* adalah sebagai berikut:

Tabel XXI. Rata-Rata *Precision* dan *Recall*

	TF-IDF	WIDF
Precision	70,7%	63,1%
Recall	70,7%	63,1%

Pada tabel di atas menunjukkan bahwa untuk mengklasifikasikan dokumen Berbahasa Indonesia dengan menggunakan pembobotan TF-IDF memiliki rata-rata ketepatan (*precision*) sebesar 70,7%. Sedangkan dengan menggunakan pembobotan WIDF memiliki rata-rata ketepatan (*precision*) sebesar 63,1%.

Berdasarkan hasil pengujian dengan menggunakan 9 dokumen sampel dapat disimpulkan bahwa untuk mengklasifikasikan dokumen Berbahasa Indonesia dengan menggunakan pembobotan TF-IDF lebih baik daripada dengan menggunakan pembobotan WIDF.

IV. KESIMPULAN

Dari hasil penelitian yang telah dilakukan, maka dapat disimpulkan bahwa sistem yang dibuat dapat melakukan klasifikasi dokumen / teks Berbahasa Indonesia sesuai dengan kategori yang ditentukan. Dari hasil pengujian *precision* dan *recall* didapat bahwa dengan menggunakan pembobotan TF-IDF ketepatan dan kelengkapan sistem dalam melakukan klasifikasi dokumen sebesar 70,7%. Sedangkan dengan menggunakan pembobotan WIDF ketepatan dan kelengkapan sistem dalam melakukan klasifikasi dokumen sebesar 70,6%.

REFERENSI

[1] Andika, Ari. 2015. *Perancangan Aplikasi Pengukuran Similaritas Pada Dokumen Dengan Metode Semantic*. STMIK Budi Darma: Medan

[2] Agusta, L.2009. *Perbandingan Algoritma Stemming Porter Dengan Algoritma Nazief dan Adriani Untuk Stemming Dokumen Teks Bahasa Indonesia*. Universitas Kristen Satya Wacana.Bali.2009

[3] Christopher, D. Manning, Prabhakar Raghavan, Hinrich Schütze. 2009]. *An Introduction to Information Retrieval*. Cambrigde: Cambridge UP.

[4] Diah Pudi Langgeni, ZK.Abdurahman Baizal dan Yanuar Firdaus A.W. *Clustering Artikel Berita Berbahasa Indonesia Menggunakan Unsupervised Feature Selection*. Institut Teknologi Telkom : Bandung.2010

[5] Hamzah A., F.Soesianto, Adhi Susanto & Jazi Eko Istiyanto. *Studi Kinerja Fungsi-fungsi Jarak dan Similaritas Dalam Clustering Dokumen Teks Berbahasa Indonesia*. Seminar UPN Veteran. Yogyakarta.2008

[6] Han, J., Kamber, M. 2006. *Data Mining Concept and Tehniques*. San Fransisco: Morgan Kauffman.

[7] Henny Leidiyana. 2013. *Penerapan Algoritma K-Nearest Neighbor Untuk Penentuan Resiko Kredit Kepemilikan Kendaraan Bermotor*. STMIK Nusa Mandiri: Jakarta

[8] Lasarus, P. Malese. 2015. *Model Mesin Pencari Dokumen Bahasa Indonesia Studi Efektifitas pada Vektor Space Model Algoritma Stemming Poter Pembobotan Frekuensi Term Berbanding Frekuensi Term Dalam Pencarian dan Fungsi Kesamaan Cosine*. Magister Komputer, Universitas Budi Luhur Jakarta

[9] Marlinda, Linda dan Rianto, Harsih. *Pembelajaran Bahasa Indonesia Berbasis Web Menggunakan Metode Maximum Marginal Relevance*. Jurnal Seminar Nasional. AMIK Bina Sarana Informatika.Jakarta Pusat.2013

[10] Salton, G., 1989, *Automatic Text Processing: The Transformation, Analysis, And Retrieval Information by Computer*, Massachusetts, Addison-Wesley.

[11] Soesianto F., Adhi Susanto & Jazi Eko Istiyanto. *Studi Kinerja Fungsi-fungsi Jarak dan Similaritas Dalam Clustering Dokumen Teks Berbahasa Indonesia*. Seminar UPN Veteran. Yogyakarta.2008

[12] Tala, F. Z. A *Study of Stemming Effects on Information Retrieval in Bahasa Indonesia*. Amsterdam: Universitet van Amsterdam.2003

[13] Tokunaga, Takenobu & Iwayana, Makoto. *Text Categorization Based On Weighted Inverse Document Frequency*. Tokyo: Department Of Computer Science Tokyo Institute Of Technology.1994

[14] Triawati, Candra. 2009. *Metode Pembobotan Statistical Concept Based untuk Klastering dan Kategorisasi Dokumen Berbahasa Indonesia*. IT TELKOM Bandung.