Positif : Jurnal Sistem dan Teknologi Informasi
Volume 4, No.2, 2018, pp. 103 - 112

E-ISSN 2460-9552
P-ISSN 2620-3227

Article history
Received July 12, 2018
Accepted Nov 15, 2018

IMPLEMENTASI TEKS MINING UNTUK KLASIFIKASI BUKU BERDASARKAN DEWEY DECIMAL CLASIFICATION (DDC) DI PERPUSTAKAAN STMIK ASIA MALANG BERBASIS VEKTOR SPACE MODEL

Sunu Jatmika[1], Maria Theresia Indriastuti[2], Gibran Adna Wafdulloh[3]
[1,2,3] STMIK ASIA Malang
Email : [1]sunu.srg@gmail.com,[2]maria_thi@yahoo.com, [3]gibranadna@gmail.com

Abstract

ASIA University Library has a problem in the grouping of new books in large numbers, because the DDC is a parameter for the grouping of the book has not been entered into the library information system, so that when the librarian input some books, they should be viewed DDC before entering into the system and it was very reducing the work time efficiently. Those problems can be solved by adding the new intelligent system in the application, namely the pre-processing algorithms that has tokenizing phase, filtering, and stemming use Nazief Andriani and then compared to find similarities documents or title with Vector Space Model algorithm. From the test that results using the algorithm above, eventually the system could resolve the problems in the library of ASIA that is this system has a recall value by 63% and the value of Precision by 72%. These results meet on the effectiveness of information retrieval system that the accuracy of a value is at least 50%.

Keywords: *Data Teks Mining, DDC, Pre-Processing, Vector Space Model (VSM)*

Abstrak

Permasalahan yang dihadapi perpustakaan STMIK ASIA Malang sampai saat ini adalah dalam hal sistem pengklasifikasi buku , hal ini disebabkan d*ewey decimal clasification* (DDC) yang menjadi parameter untuk pengelompokkan buku belum masuk ke dalam sistem informasi perpustakaan, sehingga ketika pustakawan menginputkan buku harus melihat DDC terlebih dahulu sebelum memasukkan ke dalam sistem dan itu sangat mengurangi efesien waktu kerja. Permasalahan tersebut dapat diselesaikan dengan menambah sistem cerdas baru didalam aplikasi, yaitu dengan algoritma *pre-processing* yang mempunyai tahapan tokenizing, filtering, dan stemming menggunakan Nazief Andriani dan kemudian dibandingkan untuk mencari kemiripan dokumen/judul dengan algoritma *Vector Space Model*. Dari hasil pengujian dengan menggunakan algoritma diatas sistem akhirnya bisa membantu masalah yang dimiliki perpustakaan ASIA yaitu sistem ini memiliki nilai *Recall* sebesar 63% dan nilai *Precision* sebesar 72%. Sehingga bisa meningkatkan kinerja dari pustakwaan untuk pengelompokan buku perpustakaan sebesar 50%.

Kata kunci: *Teks Mining, DDC, Pre-Processing, Vector Space Model (VSM)*

E-ISSN 2460-9552
P-ISSN 2620-3227

1. PENDAHULUAN

Sebagaimana kondisi perkembangan internal yang dialami kampus STMIK ASIA Malang dalam bidang pengajaran dan bidang lainnya, tentunya perpustakaan STMIK ASIA pun ikut terus berkembang. Hal ini dapat dilihat melalui semakin bertambahnya jumlah buku di perpustakaan STMIK ASIA. Akan tetapi dengan bertambahnya buku yang ada di perpustakaan masalah timbul adalah dalam hal klasifikasi buku hal ini disebabkan pustakawan yang akan mengelompokkan buku harus melihat DDC terlebih dahulu sebelum dimasukkan ke sistem sehingga hal ini sangat menyita waktu dalam pengerjaannya.

Dengan adanya aplikasi berbasis web yang disebut SLiMS masih terdapat kekurangan, yaitu dalam menentukan nomor klasifikasi koleksi buku yang pada umumnya masih dilakukan secara manual, yang mana hal ini relatif rumit terutama jika buku yang dikelompokkan dalam jumah besar, keterbatasan pustakawan dan wawasan tentang isi buku kurang luas. Tujuan penentuan nomor klasifikasi adalah untuk menempatkan koleksi perpustakaan sesuai dengan jenisnya, agar rapi dan terstruktur, sehingga mempermudah pustakawan maupun pengunjung untuk mencari buku. Adapun untuk mengembangkan aplikasi senayan ini, perlu beberapa algoritma untuk menyusunnya, yaitu tahap pertama dengan pre-processing yang kemudian dibandingkan dengan algoritma Vector Space Model.

2. LANDASAN TEORI

2.1 *Vector Space Model*

Metode yang diaplikasikan dalam proses pencarian informasi atau pertambangan teks, adalah dengan menggunakan metode frekuensi jangka –frekuensi dokumen terbalik (*tf-idf*), yang sekaligus digunakan mengevaluasi adalah dengan menggunakan kata dalam dokumen. *Tf-idf* merupakan metode untuk mengkonversi representasi tekstual dari informasi ke *Space Model Vector (VSM)*, maupun mengkonversi ke fitur jarang.

Mengutip dari Gerard Salton yang menyatakan bahwa VSM jika ditinjau dari sejarahnya memiliki konteks yang masih belum layak, Gerard Salton menjelaskan bahwa VSM sendiri sebenarnya tidak berhasil. *VSM* adalah model aljabar yang mewakili informasi tekstual dalam bentuk vektor, komponen vektor tersebut dapat mewakili istilah dari *tf-idf* atau ketersediaan maupun tidak tersedianya bagian kata yang dimaksud di dalam dokumen tesebut.

Sebagai catatan,VSM klasik yang dihasilkan oleh Salton yaitu dengan menggabungkan parameter/informasi lokal dan global dalam artian bahwa Salton menggunakan istilah terisolasi yang dianalisis dan atau seluruh koleksi dokumen.

VSM menurut Sensu Lato adalah ruang di mana teks dipresentasikan tidak dalam bentuk vektor dari angka representasi string tekstual aslinya, jika menurut Lato VSM adalah presentasi dari fitur yang diekstrak dari dokumen.

Istilah “adalah” dan “ “ diabaikan seperti yang telah dikutip sebelumnya. Jika terdapat kosakata indeks maka dapat mengkonversi dokumen uji yang nantinya ditetapkan menjadi ruang vektor di mana untuk setiap istilah vektor diindeks sebagai kosakata indeks, sehingga istilah pertama dari vektor adalah “biru”, yang kedua “matahari” dan seterusnya. Setelah itu digunakan istilah-frekuensi untuk mewakili setiap istilah dalam ruang vektor. Istilah-frekuensi tidak lebih dari ukuran berapa kali istilah dalam kosa kata $^{E(t)}$ yang muncul dalam dokumen *d3* atau *d4*, jika didefinisikan istilah-frekuensi sebagai fungsi menghitung adalah sebagai berikut:

$$tf(t,d) = \sum_{x \in d} fr\,(x,t)$$

Dimana fr (x,t) adalah sebuah fungsi sederhana yg didefinisikan sebagai

$$fr(x,t) = \begin{cases} 1, if\ x = t \\ 0, \end{cases}$$

Jadi, tf (t,d) adalah berapa kali t jangka hadir dalam dokumen d sebagai contoh tf (“*sun”,* d4) = 2 sehingga hanya terdapat dua kejadian dari istilah "sun" di d4 dokumen. Sedangkan pencapaian vektor dokumen yang diwakili oleh:

$$(tf(t1,dn), tf(t2,dn), tf(t3,dn), \dots, tf(tn,dn))$$

Setiap dimensi dari vektor dokumen tersebut dipresentasikan oleh istilah dari kosakata tf (t1, d2) yang merupakan jangka dari frekuensi istilah 1 atau *t1* sebagai kosakata dari “biru” pada d2 dokumen.

$$\overrightarrow{Vds} = (tf(t_1,d_3), tf(t_2,d_3), tf(t_3,d_3), \dots, tf(t_n,d_3))$$

E-ISSN 2460-9552
P-ISSN 2620-3227

$$\overrightarrow{Vds} = (tf(t_1, d_4), tf(t_2, d_4), tf(t_3, d_4), \ldots, tf(t_n, d_4))$$

Yang mana mengevaluasi kepada :

$$v_{\overrightarrow{ds}} = (0,1,1,1)$$
$$v_{\overrightarrow{ds}} = (0,2,1,0)$$

Hasil dari $v\overrightarrow{d_3}$ menunjukkan bahwa terdapat, dalam rangka, 0 kejadian istilah "biru", 1 terjadinya istilah "matahari", dan sebagainya $v\overrightarrow{d_3}$,kami memiliki 0 kejadian dari istilah "biru", 2 kejadian istilah "matahari", dll. Selain presentasi dalam bentuk vektor juga dapat dipresentasikan ke dalam bentuk matriks dengan $|D| \times F$, dimana |D| adalah kardinalitas dari ruang dokumen, atau berapa banyak dokumen yang dimiliki dan F adalah jumlah fitur yang dipresentasikan dalam ukuran kosakata. Representasi matriks vektor yang dijelaskan di atas adalah:

$$M_{|D|} \times F = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 2 & 1 & 0 \end{bmatrix}$$

Matriks tersebut adalah presentasi dari frekuensi dengan jangka yang cenderung jarang, maka yang akan ditampilkan nantinya adalah representasi umum dari matriks tersebut.

Kelemahan dari metode pendekatan jangka frekuensi adalah yang muncul di bagian atas atau bagian hasil pencarian yang tertinggi adalah kosakata yang paling sering frekuensinya, sedangkan kosakata yang memiliki frekuensi rendah atau jarang muncul, maka akan berada di urutan bawah dari hasil pencarian dengan menggunakan metode pendekatan jangka frekuensi. Walaupun istilah yang langka atau istilah yang jarang muncul tersebut lebih informatif dan sesuai dari pada istilah yang berada di urutan tertinggi.

Umumnya para pengguna menerapkan istilah yang hampir muncul diseluruh dokumen yang dicari. Untuk mengerucutkan hasil pencarian maka diterapkan tf-idf. Cara kerja tf-idf adalah dengan menitik beratkan kata dalam koleksi dokumen yang terkait dengan cara menggabungkan perimeterlokal dan global sehingga hasil pencarian istilah tersebut masi berada dalam konteks koleksi dokumen yang diharapkan. Tf-idf juga menggunakan skala logaritmik dalam pengklasifikasian istilah.

Definisi \ mathrm {tf} (t, d) yang sebenarnya adalah hitungan jangka waktu t istilah dalam dokumen d. Frekuensi penggunaan istilah sederhana dapat memunculkan kembali masalah spamming kata kunci, dalam artian jika istilah tersebut terulang dalam dokumen dengan tujuan untuk meningkatkan peringkat klasifikasi dalam IR (*Information Retrieval*) sistem atau membuat bias dari dokumen yang panjang, pengguna menemukan istilah penting dengan jalan tingginya frekuensi istilah dalam dokumen.Untuk itu vektor tersebut perlu dinormalkan kembali.

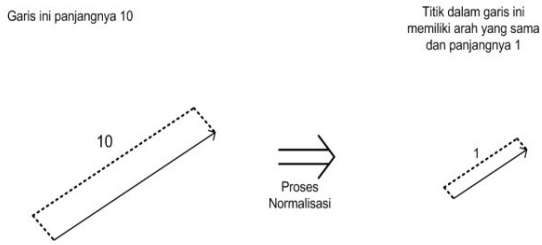
2.2 Vektor Normalisasi

Untuk menormalkan vektor, adalah sama dengan menghitung Unit Vector vektor, dan mereka dilambangkan menggunakan "topi" notasi $\vec{v}$. Definisi unit vektor $\vec{v}$ dari vektor $\vec{v}$ adalah:

$$\hat{v} = \frac{\vec{v}}{\|\vec{v}\|p}$$

Dimana adalah vektor satuan, atau vektor normal, $\hat{v}$ adalah vektor akan menjadi normal dan $\|\vec{v}\|p$ adalah norma (magnitude, panjang) vektor $\vec{v}$ di L^p.

Vektor satuan sebenarnya tidak lebih dari versi normal vektor, adalah vektor yang panjangnya 1.



Gambar 2.1. Gambar Vektor

Akan tetapi terdapat pertanyaan penting tentang hal tersebut, yaitu bagaimana panjang vektor dihitung dan memahami hal ini harus dengan memahami motivasi ruang, juga disebut ruang Lebesgue.

Biasanya panjang vektor $\|\vec{u}\|1 = (|u_1|+|u_2|+|u_3|+\ldots+|u_n|)$ dihitung dengan menggunakan norma Euclidean - norma adalah fungsi yang memberikan panjang ketat positif atau ukuran untuk semua vektor dalam ruang vektor -, yang didefinisikan oleh:

$$\|\vec{u}\| = \sqrt{u_1^2 + u_2^2 + u_3^2 + \ldots + u_n^2}$$

Akan tetapi definisi tersebut bukan satu-satunya cara untuk menentukan panjang, dan oleh karena itulah terkadang sejumlah p

E-ISSN 2460-9552
P-ISSN 2620-3227

bersama-sama dengan notasi norma, seperti di $\|u\|_p$ karena bisa digeneralisasi sebagai:

$$\|u\|_p = (\sum_{i=1}^{n} |\overrightarrow{u_i}|^p)^{\frac{1}{p}}$$

Sehingga, ketika membaca tentang L2-norma, Anda membaca tentang norma Euclidean, norma dengan p = 2, norma paling umum digunakan untuk mengukur panjang vektor, biasanya disebut "besarnya"; sebenarnya, ketika Anda memiliki panjang ukuran wajar tanpa pengecualian (tanpa nomor p),Anda memiliki L2-norma (Euclidean norm).

Ketika membaca tentang L1-norma, Anda membaca tentang norma dengan p = 1, didefinisikan sebagai:

$$\|\vec{u}\|1 = (|u_1|+|u_2|+|u_3|+\ldots+|u_n|)$$

Yang tidak lebih dari jumlah sederhana dari komponen-komponen vektor, juga dikenal sebagai Taksi jarak, juga disebut Manhattan jarak. Perhatikan bahwa dapat juga menggunakan norma apapun untuk menormalkan vektor, tetapi akan digunakan norma yang paling umum, L2-Norm, yang juga default dalam 0,9 rilis scikits.learn tersebut. Juga dapat ditemukan kertas membandingkan kinerja dari dua pendekatan antara metode lain untuk menormalkan vektor dokumen, sebenarnya dapat menggunakan metode lain, tetapi harus diringkas, setelah menggunakan norma, maka harus digunakan untuk seluruh proses langsung yang melibatkan norma (vektor satuan yang digunakan L1-norma tidak akan memiliki panjang 1 jika mengambil L2-norma nanti).

Sekarang proses vektor normalisasi adalah, dapat dicoba contoh konkret, proses menggunakan L2-norma (akan digunakan istilah sekarang) untuk menormalkan vektor kami $\vec{v}_{d4} = (0,2,1,0)$ untuk mendapatkan vektor satuan yang $\hat{v}_{d4}$ Untuk melakukan itu, kami akan plug sederhana ke dalam definisi vektor satuan untuk mengevaluasi:

$$\hat{v} = \frac{\vec{v}}{\|\vec{v}\|p}$$

2.3 *Term Frequency-Invers Document Frequency* (TF-IDF)

Ruang dokumen dapat didefinisikan kemudian $D = \{d_1, d_2, \ldots, d_n\}$ di mana n adalah jumlah dokumen dalam corpus, dan dalam kasus masalah ini sebagai $D_{train} = \{d_1, d_2\}$ dan $D_{test} = \{d_3, d_4\}$. Kardinalitas ruang dokumen ditentukan oleh $|D_{train}| = 2$ dan $|D_{test}| = 2$ karena hanya dimiliki 2 dua dokumen untuk pelatihan dan pengujian, akan tetapi tidak perlu memiliki kardinalitas yang sama.

Idf (*invers document frequency*) dapat didefinisikan:

$$\text{Idf}(t_1) = log\frac{|D|}{1+|\{d:t_{1\in d}\}|}$$

Dimana $|\{d: t \in d\}|$ adalah jumlah dokumen dimana t jangka muncul, ketika fungsi memenuhi jangka frekuensi $tf(t,d) \neq 0$ hanya menambahkan 1 ke dalam rumus untuk menghindari nol-divisi. Rumus untuk tf-idf kemudian:

tf-idf (t) = tf (t,d) x idf (t)

dan formula ini memiliki konsekuensi penting, berat tinggi perhitungan tf-idf tercapai ketika memiliki frekuensi jangka tinggi (tf) dalam dokumen yang diberikan (parameter lokal) dan frekuensi dokumen rendah istilah dalam seluruh koleksi (parameter global).

Sekarang menghitung idf untuk setiap fitur yang ada di matriks fitur dengan frekuensi istilah yang telah dihitung di tutorial pertama:

$$M_{train} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 2 & 1 & 0 \end{bmatrix}$$

Karena ada 4 fitur, maka harus menghitung idf(t_1), idf(t_2), idf(t_3), idf(t_4) :

$$\text{Idf}(t_1) = log\frac{|D|}{1+|\{d:t_{1\in d}\}|} = \log\frac{2}{1} = 0.69314718$$

Sekarang telah dimiliki matriks dengan frekuensi istilah (M_{train}) dan vektor yang mewakili idf untuk setiap fitur dari matriks kami ($\overrightarrow{idf}_{train}$), dapat dihitung bobot tf-idf kami. Yang harus dilakukan adalah perkalian sederhana dari setiap kolom dari matriks:

dan kemudian kalikan dengan matriks frekuensi istilah, sehingga hasil akhir dapat didefinisikan kemudian sebagai:

$$M_{tf-idf} = M_{train} \text{ x } M_{idf}$$

Harap dicatat bahwa perkalian matriks tidak komutatif, hasil A x B akan berbeda dengan hasil dari B x A, dan ini adalah mengapa M_{idf} adalah di sisi kanan dari perkalian, untuk mencapai efek yang diinginkan dari mengalikan setiap nilai idf untuk fitur yang sesuai:

E-ISSN 2460-9552
P-ISSN 2620-3227

normalisasi L2 kami ke M_{tf-id} matriks. Harap dicatat bahwa normalisasi ini adalah "baris-bijaksana" karena akan ditangani setiap baris dari matriks sebagai vektor dipisahkan untuk dinormalisasi, dan tidak matriks secara keseluruhan:

$$M_{tf-id} = \frac{M_{tf-id}}{\|M_{tf-idf}\|2} = \begin{matrix} 0 & -0.70710678 & -0.70710678 & 0 \\ 0 & -0.89442719 & -0.4472136 & 0 \end{matrix}$$

Dan itu adalah titik berat tf-idf cukup normal yang relevan dengan set dokumen pengujian, yang sebenarnya merupakan kumpulan unit vektor. Jika mengambil L2-norma setiap baris dari matriks, maka akan terlihat bahwa semuanya memiliki L2-norma 1.

definisi produk dot untuk dua vektor $\vec{a} = (a_1, a_2, a_3, \ldots.)$ $\vec{b} = (b_1, b_2, b_3, \ldots.)$ dimana a_n and b_n adalah komponen dari vektor (fitur dokumen, atau nilai-nilai TF-IDF untuk setiap kata dari dokumen) dan n adalah dimensi dari vektor:

$$\vec{a}.\vec{b} = \sum_{i=1}^{n} a_i\, b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$$

definisi titik produk adalah perkalian sederhana dari masing-masing komponen dari vektor kedua ditambahkan bersama-sama. Lihat contoh produk dot untuk dua vektor dengan 2 dimensi masing-masing (2D):

$$\vec{a} = (0,3)$$
$$\vec{b} = (0,4)$$
$$\vec{a}.\vec{b} = 0 * 4 + 3 * 0 = 0$$

Hal pertama yang diperhatikan adalah bahwa hasil dari *dot product* antara dua vektor bukan vektor lain tetapi nilai tunggal, skalar.

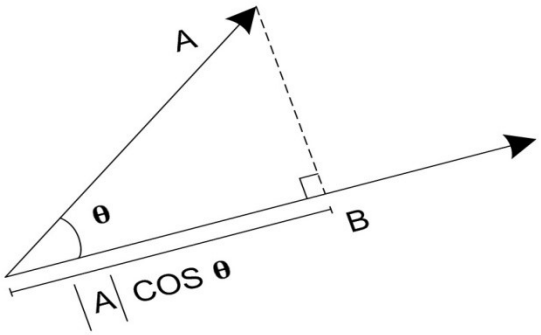
Ini semua sangat sederhana dan mudah dimengerti, tapi apa adalah titik produk? Apa ide intuitif di balik itu? Apa artinya memiliki produk titik nol? Untuk memahami itu, memahami apa definisi geometris dari *dot product*:

$$\vec{a}.\vec{b} = \|\vec{a}\|\|\vec{b}\| \cos\theta$$

Menata ulang persamaan untuk memahami lebih baik menggunakan properti komutatif:

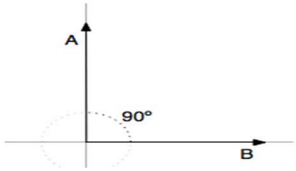
$$\vec{a}.\vec{b} = \|\vec{b}\|\|\vec{a}\| \cos\theta$$

Jadi, apa istilah $\|\vec{a}\| \cos\theta$ Istilah ini adalah proyeksi dari vektor $\vec{a}$ ke vektor $\vec{b}$ seperti yang ditunjukkan pada gambar di bawah ini:



Gambar 2.2 proyeksi vektor

Sekarang, apa yang terjadi ketika vektor $\vec{a}$ ortogonal (dengan sudut 90 derajat) ke vektor $\vec{b}$ seperti pada gambar di bawah?



Gambar 2.3 Sudut 90 Derajat

Tidak akan ada sisi yang berdekatan pada segitiga, maka akan setara dengan nol, istilah $\|\vec{a}\| \cos\theta$ akan menjadi nol dan perkalian yang dihasilkan dengan besarnya vektor $\vec{b}$ juga akan menjadi nol. ketika dot produk antara dua vektor yang berbeda adalah nol, mereka ortogonal satu sama lain (mereka memiliki sudut 90 derajat), ini adalah cara yang sangat rapi untuk memeriksa orthogonality vektor yang berbeda. Hal ini juga penting untuk dicatat bahwa menggunakan contoh 2D, tapi fakta paling menakjubkan tentang itu adalah bisa menghitung sudut dan kesamaan antara vektor dalam ruang dimensi yang lebih tinggi.

2.4 *Cosine Similarity*

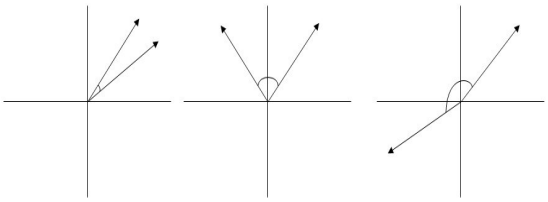
Cosinus kesamaan antara dua vektor (atau dua dokumen pada Space Vector) adalah ukuran yang menghitung kosinus sudut. metrik ini adalah pengukuran orientasi dan tidak besarnya, dapat dilihat sebagai perbandingan antara dokumen di ruang dinormalisasi karena tidak mempertimbangkan hanya besarnya

E-ISSN 2460-9552
P-ISSN 2620-3227

masing-masing jumlah kata (tf-idf) dari setiap dokumen, tetapi sudut antara dokumen. Untuk membangun persamaan *cosine* similarity adalah untuk memecahkan persamaan dot produk untuk $\cos\theta$:

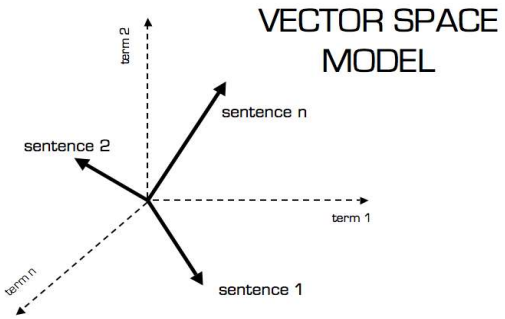
$$\vec{a}.\vec{b} = \|\vec{a}\|\|\vec{b}\| \cos\theta$$

$$\cos\theta = \frac{\vec{a}.\vec{b}}{\|\vec{a}\|\|\vec{b}\|}$$



Gambar 2.4 Sudut Cosine Similarity

Perhatikan bahwa vektor menunjuk ke titik jauh dari vektor lain, masih bisa memiliki sudut kecil dan itu adalah titik pusat pada penggunaan Cosine Similarity, pengukuran cenderung mengabaikan hitungan jangka yang lebih tinggi pada dokumen. Misalkan memiliki dokumen dengan kata "langit" muncul 200 kali dan dokumen lain dengan kata "langit" yang muncul 50, jarak Euclidean antara mereka akan lebih tinggi tapi sudut masih akan kecil karena mereka menunjuk ke arah yang sama. Model Ruang Vektor dokumen (seperti pada gambar di bawah) dimodelkan sebagai vektor (dengan TF-IDF penting) dan juga memiliki rumus untuk menghitung kesamaan antara dokumen yang berbeda dalam ruang vektor. Adapun gambar *vector space model*:



Gambar 2.5 Gambar VSM

Untuk memecahkan Kesamaan Cosine untuk sudut antara vektor dengan rumus:

$$\cos\theta = \frac{\vec{a}.\vec{b}}{\|\vec{a}\|\|\vec{b}\|}$$

hanya perlu mengisolasi sudut (θ) dan memindahkan cos ke kanan dari persamaan:

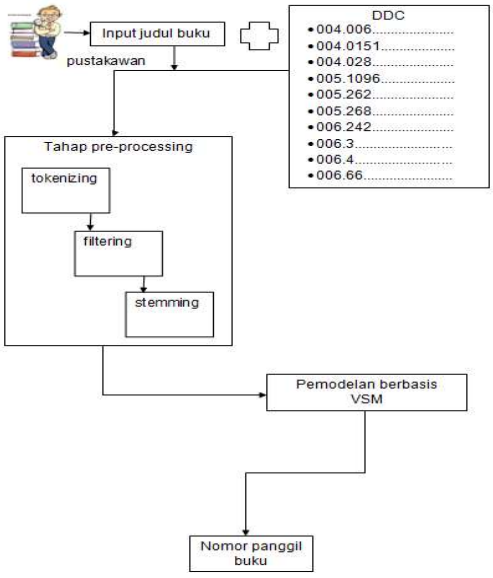
$$\theta = arccos\frac{\vec{a}.\vec{b}}{\|\vec{a}\|.\|\vec{b}\|}$$

$arccos$ adalah sama dengan kebalikan dari kosinus $(cos^{-}1)$.

3. METODOLOGI PENELITIAN

3.1. Blok Sistem

Arsitektur sistem digunakan untuk menggambarkan sistem kerja yang digunakan pada proses analisa dan implementasi. Dengan adanya arsitektur sistem dapat dilihat alur sistem secara lengkap, adapun arsitektur sistem secara keseluruhan ditunjukkan pada gambar berikut:



Gambar 3.1 Block Diagram

Dan berikut adalah penjelasan dari arsitektur diatas:

1. Proses awal sistem dimulai dari pustakawan yang menginputkan judul buku baru. Judul buku baru ini nantinya sebagai query dari sistem yang akan diolah. Tahapan pre-processing dibutuhkan untuk membentuk kata yang siap untuk di mining. Tahapan yang dilakukan yaitu tokenizing, filtering, stemming. Selain itu di dalam sistem juga ada DDC, yaitu kumpulan-kumpulan nomor panggil buku yang nantinya buku yang diinputan oleh pustakawan akan mendapat

E-ISSN 2460-9552
P-ISSN 2620-3227

nomer panggil sesuai dengan ketentuan dari perpustakaan. Nomor panggil yang ada di dalam DDC juga akan diolah dengan pre-processing sehingga membentuk kata yang siap untuk di mining.

2. Selanjutnya dari judul yang diolah dan menghasilkan query, dibandingkan dengan DDC yang diolah sehingga menghasilkan nomor panggil judul buku yang relevan dengan DDC yang ada. Perbandingan sistem diatas tentunya menggunakan pemodelan berbasi vector space model (VSM). Nomor panggil judul buku yang dihasilkan akan ditampilkan kepada pustakawan sehingga pustakawan dengan mudah memberi nomor panggil pada buku tersebut.

3.2. Pre-Processing

Dalam hal ini contoh kasusnya yang diambil dari salah satu judul buku yang ada di perpustakaan STMIK ASIA yang mana judul ini menjadi query untuk membandingkan dokumen-dokumen yang ada di dalam DDC sehingga menghasilkan nomor panggil untuk buku tersebut. Tahapan-tahapan dalam pre-processing data meliputi tokenizing, filtering, dan stemming. Dan supaya hasil dari pre-processing ini dapat dilihat hasilnya maka pada tahap-tahap tersebut akan disimpan. Dalam sistem yang dirancang proses pre-processing ini membutuhkan waktu yang cukup lama dikarenakan pemrosesan dilakukan terhadap seluruh dokumen.

Berikut ini tahapan-tahapan dalam pre-processing yang dilakukan.

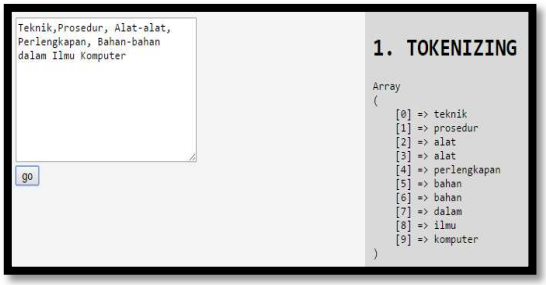
1. Tokenizing
   Tokenizing merupakan tahapan memecah kalimat menjadi kata-kata yang berdiri sendiri-sendiri, yaitu dengan cara melakukan pemisahan masing-masing kata dengan cara memotong semua kata berdasarkan spasi ‘’ ‘’.
2. Filtering
   Filtering merupakan tahapan dimana kata-kata yang dianggap tidak memiliki makna dihilangkan. Pada tahapan filtering diperlukan sebuah kamus stopword yang berisikan list dari kata-kata yang dianggap tidak memiliki makna.
3. Stemming
   Setelah penghapusan semua kata-kata yang tidak memiliki makna dan sering muncul maka hanya akan tersisa kata-kata yang penting saja. Kata-kata inilah yang menunjukkan isi dari dokumen. Akan tetapi kata-kata ini harus diproses lagi untuk mencari bentuk kata dasarnya. Proses pembentukan kata dasar inilah yang disebut dengan stemming. Sebagai contoh kata “menggambar” dan “menggambarkan” adalah dua kata yang memiliki makna berbeda tetapi memiliki kata dasar yang sama yaitu “gambar”. Kata “menggambar” dan “menggambarkan” harus diubah ke bentuk dasar yaitu “gambar” sehingga memiliki makna yang sama dan dihitung memiliki dua kali kemunculan yaitu dari kata “menggambar” dan “menggambarkan”. Salah satu fungsi dari proses stemming ini adalah untuk mengurangi jumlah kata yang akan diproses pada proses pecarian. Sistem yang dibangun tersebut akan menggunakan algoritma stemming Nazief Andriani. Terdapat beberapa alasan pemilihan algoritma ini dipilih dalam penelitian ini, diantaranya adalah:

a. Algoritma Nazief Andriani memiliki akurasi yang tinggi dibanding dengan algoritma-algoritma yang lain.
b. Proses stemming hanya dilakukan sekali pada penelitian ini dan hasilnya pun disimpan sehingga, walaupun algoritma Nazief Andriani memerlukan waktu yang lama, hal ini tidak menjadi pertimbangan utama.
c. Dataset bersifat tetap.
d. Dokumen yang dijadikan studi kasus adalah dokumen-dokumen yang berbahasa indonesia, sehingga algoritma Nazief Andriani sangat tepat digunakan disini.

4. HASIL DAN PEMBAHSAN

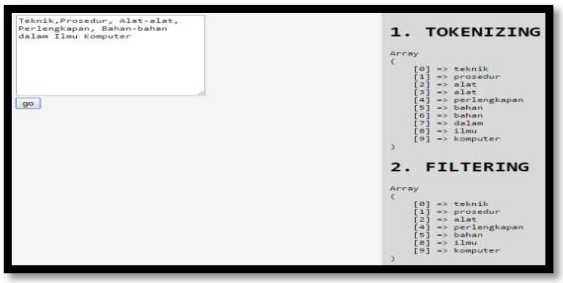
Secara umum bab ini berisi tentang desain dan implementasi Information Retrieval System yang telah dibangun. Pembangunan sistem di dalam program ini juga mengarah pada apa yang telah dibahas di bab – bab sebelumnya. Yang mana di dalam bab ini berisi tampilan – tampilan tentang program yang telah dibangun beserta data yang ada dan disertai dengan hitungan dari data kongkrit sehingga menghasilkan penerapan yang relevan. Adapun untuk hasil -hasilnya adalah sebagai berikut.

E-ISSN 2460-9552
P-ISSN 2620-3227



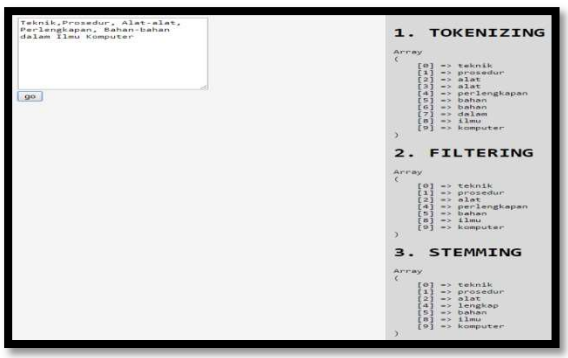
Gambar 4.1 Implementasi Tokenizing

Tampilan program untuk posisi ini berguna untuk memecah kata yang diinputkan sehingga kata – kata tersebut berdiri sendiri. Adapun source code nya adalah sebagai berikut.



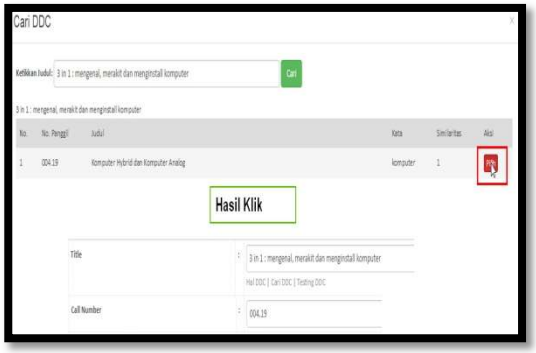
Gambar 4.2 Implementasi Filtering

Setelah diproses melalui tokenizing dilanjutkan ke dalam proses filtering, yang mana menghilangkan kata – kata yang termasuk dalam stop words. Adapun source code nya sebagai berikut.



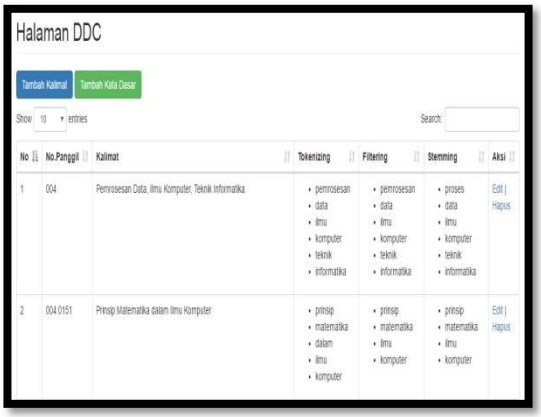
Gambar 4.3 Implementasi Stemming

Proses terakhir pengolahan kata yang diinputkan adalah tahap stemming, dimana pada tahap ini kata dari hasil filtering yang mengandung imbuhan akan dihilangkan dan menjadi kata dasar.



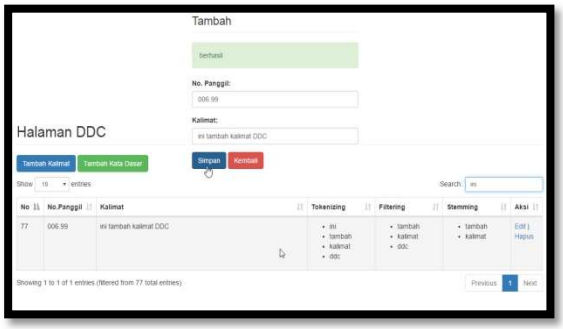
Gambar 4.4 Implementasi cari DDC

Pada tahap ini adalah perbandingan dari judul yang sudah di stemming dengan DDC yang ada di dalam database yang sudah diproses melalui stemming juga. Hasil keduanya kemudian diolah sehingga menghasilkan kalimat dan nomor DDC yang relevan. Kemudian di klik tombol pilih, maka akan langsung masuk ke dalam event judul dan nomor DDC.
Pada event halaman DDC terdapat data – data DDC yang sudah diinputkan dan juga sudah diolah dengan algoritma pre-processing. Di dalamnya juga terdapat tombol tambah kalimat dan tambah kata dasar. Fungsi dari tambah kalimat adalah untuk menambah DDC baru yang belum masuk ke dalam data – data DDC yang sudah ada. Adapun fungsi dari tombol tambah kata dasar adalah untuk menambah atau pembaruan pembendaharaan kata dasar yang sudah ada sehingga menjadikan hasil dari filtering lebih akurat. Adapun untuk implementasinya adalah sebagai berikut.

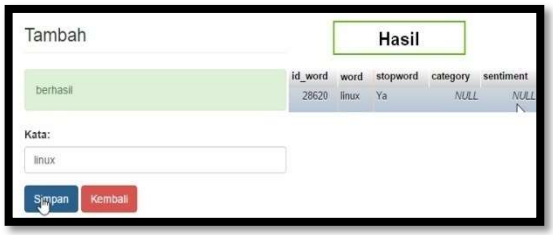


Gambar 4.5 Implementasi Halaman DDC

E-ISSN 2460-9552
P-ISSN 2620-3227

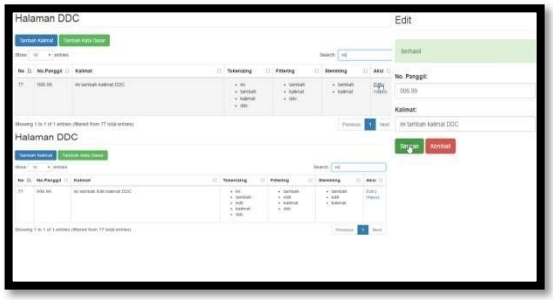


Gambar 4.6 Implementasi Tambah Kalimat



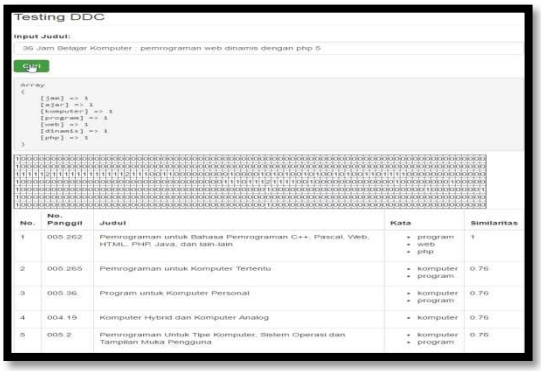
Gambar 4.7 Implementasi Tambah Kata Dasar

Di dalam halaman DDC ada tombol edit dan tombol hapus, yang mana tombol edit berfungsi untuk mengedit data – data DDC yang mungkin terjadi kesalahan saat penginputan atau ada update dari sumber DDC yang sudah resmi. Sedangkan untuk tombol hapus berfungsi untuk menghapus data – data DDC yang ada yang mungkin terjadi kesalahan penginputan atau sudah tidak dipakai lagi nomor DDC tersebut. Adapun untuk implementasi dari tombol – tombol tersebut adalah sebagai berikut.



Gambar 4.8 Implementasi Tombol Edit

Selanjutnya adalah event testing DDC, yang mana event ini berfungsi untuk menampilkan hasil perbandingan antara judul yang diinputkan dengan data – data DDC yang ada beserta nilai dari perbandingan tersebut sehingga menghasilkan urutan – urutan DDC yang mana yang lebih relevan dengan judul yang diinputkan. Untuk hasilnya adalah sebagai berikut.



Gambar 4.9 Implementasi Testing DDC

5. KESIMPULAN DAN PENUTUP

5.1 Kesimpulan

Beberapa kesimpulan dapat diambil selama proses penelitian ini yang dapat diterapkan dan bisa membantu dalam perpustakaan STMIK ASIA sebagai berikut:

1. Algoritma vector space model telah berhasil menemukan nilai yang relevan untuk referensi sebuah penomoran panggil buku.
2. Dengan menggunakan algoritma tersebut yang ditambah dengan akurasi berdasarkan recall dan precision dihasilkan nilai akurasi dengan recall sebanyak 63% dan untuk nilai precision sebanyak 72%. Kedua nilai akurasi ini memenuhi efektivitas sistem temu kembali informasi minimal 50%.

5.2 Saran

Dari serangkaian perhitungan dan pengujian pada bab sebelumnya, diharapkan sistem peringkasan ini dapat dikembangkan hingga menghasilkan ringkasan yang lebih baik, seperti:

1. Dapat melakukan analisa lebih lanjut untuk menentukan stopword yang akan digunakan, karena pada pengembangan sistem peringkasan penentuan kata yang akan di filtering juga sangat menentukan dalam pembobotan masing-masing kalimat.
2. Diharapkan penelitian ini dapat dilanjutkan dan dikembangkan terutama untuk sinonim kata, karena pada sistem ini kata yang memiliki arti sama tetapi berbeda penulisan tidak terseleksi sehingga kata – kata tersebut dihapus oleh sistem.

6. DAFTAR PUSTAKA

Cahyono, J. T., Purnama, B. E., & Sukadi.Pembuatan Sistem Informasi

E-ISSN 2460-9552
P-ISSN 2620-3227

Rental Mobil Purnama Rent Car Ploso Pacitan Berbasis Web.Pacitan. IJNS Accepted Paper.2013.

Chowdhury Roy A.,Bhattacharyya K.A., and Chattopadhyay P. Study On Functional Properties Of Raw And Blended Jackfruit Seed Flour (A Nonconventional Source) For Food Application. Indian Journal of Natural Products and Resources.2012

Even, Yahir dan Zohar. Introductions to text mining. University of Illions. Automeated Learning Group National Center for Supercomputing aplications.2002

Feldman, R., dan Sanger, J. The Text Mining Handbook :. Cambridge University Press.Advanced Approach in Analyzing Unstructured Data.2007.

Han, J & Kamber, M.Data Mining : Concepts and Techniques.2001.

Hasugian, Jonner. Dasar-Dasar Ilmu Perpustakaan dan Informasi. Medan: USU Press. 2009.

Kent, A. Information Analysis and Retrieval, 3 rd Edition.New York.Becker and Heys.1971.

Lancaster, F.W. 1979. Information Retrieval Systems: Characteristics, Testing, and Evaluation, 2 nd Edition.New York.John Wiley.1979.

Reitz, Joan M.Dictionary for Library for Library and Information Science. Westport: Libraries Unlimited.2004.

Sutabri, Tata. Analisa Sistem Informasi. Edisi Pertama. Yogyakarta: Andi.2004.

Zaenab, Ratu Siti. Efektivitas Temu Kembali Informasi Dengan Menggunakan Bahasa Alami Pada CD-ROM AGRIS dan CAB ABSTRACT, dalam Jurnal Pustakawan Pertanian Vol. 11.2002