

## IMPLEMENTASI *VECTOR SPACE MODEL* UNTUK MENINGKATKAN KUALITAS PADA SISTEM Pencarian BUKU PERPUSTAKAAN

Dedi Leman<sup>1</sup>, Khusaeri Andesa<sup>2</sup>

Teknik Informasi, Magister Komputer, Universitas Putra Indonesia "YPTK" Padang  
Universitas Potensi Utama  
Jl. K.L.Yos Sudarso Km 6.5 No. 3A - Medan  
Lemhan28@yahoo.com

### Abstrak

Buku merupakan media informasi yang berguna di dalam perkembangan ilmu pengetahuan, dengan membaca buku akan banyak ilmu yang kita dapat serta informasi yang beragam. Pada perkembangan era informasi saat ini segala jenis buku sangatlah mudah didapat dan beragam jenisnya tanpa membatasi usia. Pada saat ini perpustakaan STMIK-AMIK Riau telah menyediakan beragam buku yang tersedia khususnya bagi mahasiswa. Semakin banyak dan beragamnya buku yang tersedia, mahasiswa STMIK-AMIK Riau yang ingin membaca ataupun meminjam buku memiliki kesulitan di dalam menemukan buku yang sesuai karena harus mencari pada tumpukan rak-rak yang begitu banyak sehingga buku yang mereka butuhkan tidak ditemukan. Atas dasar permasalahan tersebut timbul sebuah gagasan untuk membangun sebuah sistem mesin pencari yang dapat memberikan pembobotan dari masing-masing judul buku yang sesuai dengan kata kunci yang dicari, sehingga akan dapat memilih yang mana buku yang menjadi paling diprioritaskan. Metode *Vector Space Model* merupakan salah satu alternatif yang dapat diimplementasikan untuk memecahkan masalah ini. Dengan metode *Vector Space Model* dapat dilihat tingkat kedekatan atau kesamaan dengan cara pembobotan *term*. Ini adalah suatu metode pembobotan data dengan menghitung jarak antar dokumen. Hasil yang diberikan dari pengujian sistem dengan metode *Vector Space Model* memberikan pembobotan pada judul buku, memberikan rekomendasi atau buku yang paling diprioritaskan sehingga sesuai dengan kebutuhan.

**Kata Kunci :** Vector Space Model, Mesin pencari, Perpustakaan

### 1. PENDAHULUAN

#### 1.1 Latar Belakang Masalah

Buku merupakan kumpulan kertas yang merupakan media informasi yang berguna di dalam perkembangan ilmu pengetahuan, dengan membaca buku akan banyak ilmu yang kita dapat serta informasi yang beragam, baik pengetahuan maupun hiburan. Hal inilah yang membuat buku memiliki peranan penting dalam perkembangan ilmu pengetahuan baik di lembaga pendidikan formal maupun nonformal. Pada perkembangan era informasi saat ini segala jenis buku-buku sangatlah mudah didapat dan beragam jenisnya tanpa membatasi usia. Buku tidak hanya untuk orang dewasa, buku juga tersedia untuk anak-anak bahkan untuk orang lanjut usia. Segala keanekaragaman buku-buku dengan beragam jenis dan kategori umumnya tersedia di perpustakaan atau di toko buku.

Dengan berkembangnya era informasi pada saat ini perpustakaan STMIK-AMIK Riau telah menyediakan beragam buku-buku yang semakin banyak, yang tersedia khususnya bagi mahasiswa. Biasanya mereka berada di perpustakaan untuk mencari referensi untuk membuat tugas akhir atau mengerjakan tugas yang diberikan oleh dosen

mereka masing-masing. Semakin banyak dan beragamnya buku yang tersedia, mahasiswa STMIK-AMIK Riau yang ingin membaca ataupun meminjam buku memiliki kesulitan di dalam menemukan buku yang sesuai karena harus mencari pada tumpukan rak-rak yang begitu banyak sehingga buku yang mereka butuhkan tidak ditemukan.

Dari kendala diatas, secara langsung maupun tidak langsung ikut berpengaruh terhadap minat baca mahasiswa STMIK-AMIK Riau. Berdasarkan hal diatas, timbul sebuah gagasan untuk membangun sebuah sistem mesin pencari yang dapat memberikan pembobotan dari masing-masing judul buku yang sesuai dengan kata kunci yang dicari, sehingga akan dapat memilih yang mana buku yang menjadi paling diprioritaskan.

Metode *Vector Space Model* merupakan salah satu alternatif yang dapat diimplementasikan untuk memecahkan masalah ini. *Vector space model* ( *VSM* ) atau model ruang vektor adalah suatu metode untuk merepresentasikan dokumen dan *query* dalam bentuk vektor pada ruang multidimensional (Singh, Dwivedi, 2012). Dengan metode *Vector*

*Space Model* dapat dilihat tingkat kedekatan atau kesamaan dengan cara pembobotan *term*.

Pada proses *stemming* atau mencari kata dasar pada kata, sistem akan menggunakan algoritma tala. Menurut hasil penelitian sebelumnya yang telah dilakukan oleh (Mardi Siswo Utomo, 2013) menyatakan bahwa *Stemmer* tala merupakan adopsi dari algoritma *stemmer* bahasa inggris terkenal *porter stemmer*. *Stemmer* ini menggunakan *rule base* analisis untuk mencari *root* sebuah kata. Pada sistem pencarian buku akan menggunakan *stemming* algoritma tala karna algoritma tala merupakan pengembangan dari algoritma porter.

Sistem pencarian buku dengan menggunakan *Vector space model* ini akan menerima masukan *query* pengguna yang akan dicocokkan dengan judul buku yang dimiliki sistem. Sehingga sistem dapat mengurutkan yang nama buku yang diprioritaskan yang sesuai dengan keinginan mereka dan sistem juga akan menunjukkan dimana posisi rak dari masing-masing buku tersebut. Salah satunya seperti penggunaan sistem rekomendasi pencarian yang mulai dimanfaatkan oleh banyak perpustakaan. Selain itu juga sangat membantu mahasiswa yang sedang melakukan penelitian atau untuk membuat tugas. Dari latar belakang di atas maka diajukan judul tesis dengan judul "IMPLEMENTASI *VECTOR SPACE MODEL* UNTUK MENINGKATKAN KUALITAS PADA SISTEM Pencarian BUKU PERPUSTAKAAN (STUDI KASUS PERPUSTAKAAN STMIK-AMIK RIAU)".

## 1.2 Perumusan Masalah

Berdasarkan latar belakang masalah di atas, maka perumusan masalahnya yaitu :

1. Bagaimana metode *Vector Space Model* dapat diterapkan dalam sebuah sistem atau aplikasi pencarian buku perpustakaan berdasarkan judul buku ?
2. Bagaimana sistem atau aplikasi pencarian buku dengan metode *Vector Space Model* dapat memberikan pembobotan pada judul buku, memberikan rekomendasi atau buku yang paling diprioritaskan yang sesuai dengan kata kunci yang dicari ?

## 1.3 Batasan Masalah

Agar penelitian ini terarah dan permasalahan yang dihadapi tidak terlalu luas serta sesuai dengan tujuan penulis, maka ditetapkan batasan-batasan terhadap masalah yang sedang diteliti, hal ini dimaksudkan agar langkah-langkah keseluruhan masalah tersebut tidak terjadi

penyimpangan. Adapun batasan masalahnya yaitu :

1. Metode *Vector Space Model* digunakan hanya untuk proses pembobotan buku yang sesuai dengan kata kunci yang dicari.
2. Untuk pembobotan buku, Metode *Vector Space Model* membandingkan antara judul buku dengan kata kunci yang dicari pada sistem.
3. Pada proses *stemming*, sistem menggunakan algoritma tala.
4. Pengujian sistem akan dilakukan dengan dengan beberapa sampel judul buku.

## 1.4 Tujuan Penelitian

Tujuan penelitian ini adalah :

1. Memahami metode *Vector Space Model* didalam proses pencarian buku berdasarkan kata kunci.
2. Menerapkan metode *Vector Space Model* dalam pencarian buku perpustakaan.
3. Membuat sebuah model pencarian buku yang memanfaatkan metode *Vector Space Model* dengan bahasa pemrograman PHP.
4. Menguji hasil rancangan *sistem yang telah dibangun*.

## 1.5 Manfaat Penelitian

Manfaat penelitian ini adalah :

1. Dengan adanya sistem ini staf perpustakaan STMIK AMIK Riau akan sangat terbantu dalam pengolahan data buku.
2. Mahasiswa juga terbantu dengan pencarian buku yang sesuai keinginan mereka.
3. Memudahkan mahasiswa menemukan buku yang paling prioritas atau relevan dengan pencarian yang dilakukan oleh sistem.

## 2. LANDASAN TEORI

### 2.1. Sistem Temu Kembali Informasi

Sistem temu kembali informasi (*information retrieval system*) adalah bagian dari ilmu komputer yang berkaitan dengan pengambilan informasi dari dokumen-dokumen berdasarkan isi dari dokumen-dokumen itu sendiri. Sistem Temu Kembali Informasi merupakan sistem yang berfungsi untuk menemukan informasi yang relevan dengan kebutuhan pengguna. Informasi yang diproses terkandung dalam sebuah dokumen yang bersifat tekstual. Dokumen yang diambil merupakan dokumen yang relevan dan membuang dokumen yang tidak relevan.

Sistem temu kembali informasi merupakan suatu sistem yang menemukan *retrieve* informasi yang sesuai dengan kebutuhan user dari kumpulan informasi secara otomatis. Prinsip kerja sistem

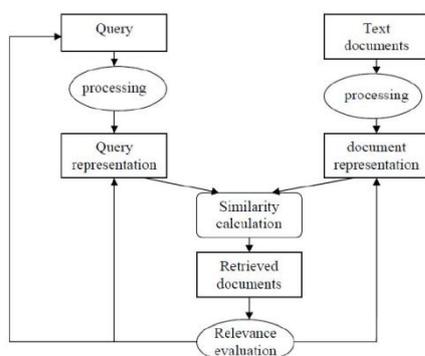
temu kembali informasi jika ada sebuah kumpulan dokumen dan seorang user yang memformulasikan sebuah pertanyaan *request* atau *query* (Fatkhul Amin, 2012).

Sistem Temu Kembali Informasi bertujuan untuk menjawab kebutuhan informasi user dengan sumber informasi yang tersedia dalam kondisi seperti sebagai berikut (Fatkhul Amin, 2012) :

1. Mempresentasikan sekumpulan ide dalam sebuah dokumen menggunakan sekumpulan konsep.
2. Terdapat beberapa pengguna yang memerlukan ide, tapi tidak dapat mengidentifikasi dan menemukannya dengan baik.
3. Sistem temu kembali informasi bertujuan untuk mempertemukan ide yang dikemukakan oleh penulis dalam dokumen dengan kebutuhan informasi pengguna yang dinyatakan dalam bentuk *key word query*/istilah penelusuran.

## 2.2. Arsitektur Sistem Temu Kembali Informasi

Sistem memiliki dua pekerjaan yaitu, yaitu melakukan *pre-processing* terhadap *database* dan kemudian menerapkan metode tertentu untuk menghitung kedekatan *relevansi* atau *similarity* antara dokumen di dalam *database* yang telah dipreprocess dengan *query* pengguna. *Query* yang dimasukkan pengguna dikonversi sesuai aturan tertentu untuk mengekstrak *term-term* penting yang sejalan dengan *term-term* yang sebelumnya telah diekstrak dari dokumen dan menghitung relevansi antara *query* dan dokumen berdasarkan pada *term-term* tersebut. Sebagai hasilnya, sistem mengembalikan suatu daftar dokumen terurut sesuai nilai kemiripannya dengan *query* pengguna (Giat Karyono, Fandy Setyo Utomo, 2012).



Gambar 2.1 Proses Temu Kembali Informasi

## 2.3. Text Mining

*Text mining* adalah proses mengolah data yang berupa teks yang didapatkan dari dokumen untuk mencari kata-kata yang dapat mewakili isi dari dokumen sehingga dapat dilakukan analisa hubungan antar dokumen. Proses penganalisan teks guna menyarikan informasi yang bermanfaat untuk tujuan tertentu.

*Text mining* dapat didefinisikan sebagai suatu proses menggali informasi dimana seorang user berinteraksi dengan sekumpulan dokumen menggunakan *tools* analisis yang merupakan komponen-komponen dalam *data mining* yang salah satunya adalah kategorisasi (Fatkhul Amin, 2011).

### 2.4. Case Folding

Adalah proses yang digunakan untuk mengubah huruf besar menjadi huruf kecil. *Case-folding* adalah proses penyamaan case dalam sebuah dokumen. Ini dilakukan untuk mempermudah pencarian dokumen.

### 2.5. Cleaning Data

Adalah proses yang digunakan untuk melakukan pembersihan text dari karakter-karakter selain huruf, tanda baca dan *tag* yang tidak digunakan nantinya didalam proses *information retrieval*.

### 2.6. Tokenizing

Tokenisasi secara garis besar dapat diartikan sebagai peoses pememecah sekumpulan karakter dalam suatu teks ke dalam satuan kata. Proses untuk membagi teks yang dapat berupa kalimat, paragraf atau dokumen, menjadi *token-token* atau bagian-bagian tertentu.

Tokenisasi merupakan proses pemisahan suatu rangkaian karakter berdasarkan karakter spasi, dan mungkin pada waktu yang bersamaan dilakukan juga proses penghapusan karakter tertentu, seperti tanda baca. Token seringkali disebut sebagai istilah *term* atau kata, sebagai contoh sebuah token merupakan suatu urutan karakter dari dokumen tertentu yang dikelompokkan sebagai unit semantik yang berguna untuk diproses (Fatkhul Amin, 2012).

### 2.7. Filtering

Filtering adalah proses membuang kata-kata yang dianggap sebagai *nois* atau kata yang dianggap tidak penting dan tidak berpengaruh terhadap makna kata.

Tahap filtering adalah tahap pengambilan kata-kata yang penting dari hasil tokenizing. Tahap filtering ini menggunakan daftar *stoplist* atau *wordlist*. *Stoplist* yaitu penyaringan *filtering* terhadap kata-kata yang tidak layak untuk dijadikan sebagai pembeda atau sebagai kata

kunci dalam pencarian dokumen sehingga kata-kata tersebut dapat dihilangkan dari dokumen. Sedangkan *wordlist* adalah daftar kata yang mungkin digunakan sebagai kata kunci dalam pencarian dokumen, dengan demikian maka tentu jumlah kata yang termasuk dalam *wordlist* akan lebih banyak daripada *stoplist* (Hervilorra Eldira, 2010).

## 2.8. Stemming

*Stemming* merupakan suatu proses untuk menemukan kata dasar dari sebuah kata dengan menghilangkan semua imbuhan. *Stemming* digunakan

untuk mengganti bentuk dari suatu kata menjadi kata dasar dari kata tersebut yang sesuai dengan struktur morfologi bahasa Indonesia yang baik dan benar.

*Stemming* adalah salah satu cara yang digunakan untuk meningkatkan performa *Information Retrieval* dengan cara mentransformasi katakata dalam sebuah dokumen teks ke kata dasarnya. Algoritma *Stemming* untuk bahasa yang satu berbeda dengan algoritma *stemming* untuk bahasa lainnya. Sebagai contoh Bahasa Inggris memiliki morfologi yang berbeda dengan Bahasa Indonesia sehingga algoritma *stemming* untuk kedua bahasa tersebut juga berbeda. Proses *stemming* pada teks berbahasa Indonesia lebih rumit/kompleks karena terdapat variasi imbuhan yang harus dibuang untuk mendapatkan root word dari sebuah kata (Ledy Agusta, 2009).

## 2.9. Indexing

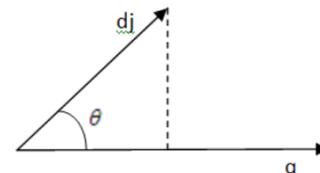
Proses *Indexing* adalah tahap pengindeksan kata dari koleksi teks yang digunakan untuk mempercepat proses pencarian. Seluruh dokumen dalam koleksi disimpan dalam satu file dengan format tertentu sehingga antara dokumen satu dengan dokumen yang lain bisa dibedakan. Setelah kata telah dikembalikan dalam bentuk kata dasar, kemudian disimpan dalam tabel. Proses *indexing* menghasilkan database index (Fatkhul Amin, 2012).

## 2.10. Vector Space Model

*Vector Space Model* (VSM) adalah metode untuk melihat tingkat kedekatan atau kesamaan *similarity term* dengan cara pembobotan *term*. Dokumen dipandang sebagai sebuah vektor yang memiliki *magnitude* (jarak) dan *direction* (arah). Pada *Vector Space Model*, sebuah istilah direpresentasikan dengan sebuah dimensi dari ruang vektor. Relevansi sebuah dokumen ke sebuah *query* didasarkan pada similaritas diantara vektor dokumen dan vektor *query* (Fatkhul Amin, 2012).

*Vector space model* digunakan untuk mengukur kemiripan antara suatu dokumen dengan suatu *query*. Konsep dasar dari *Vector Space Model* adalah menghitung jarak antar dokumen kemudian mengurutkan berdasarkan tingkat kedekatannya. Cara kerja *Vector Space Model* dimulai dengan *case folding*, *cleaning data*, *indexing*, *filtering*, *stemming*, dan *tokenisasi* yaitu tahap pemotongan string input berdasarkan tiap kata yang menyusunnya dan memecah dokumen ke dalam tabel frekuensi kata. Seluruh kata dalam dokumen dibentuk menjadi satu yang disebut sebagai *term*. Tiap dokumen ditampilkan sebagai vektor yang akan dibandingkan dengan *term* yang telah dibentuk. *Similarity Analysis* untuk mengukur kemiripan dokumen dilakukan dengan menghitung *cosinus* jarak antara dokumen tersebut.

Sebuah dokumen  $d_j$  dan sebuah *query*  $q$  direpresentasikan sebagai vektor dimensi seperti pada gambar 2.7.



Gambar 2.2 Representasi Dokumen dan Query pada Ruang Vektor

Proses perhitungan *Vector space model* melalui tahapan perhitungan *term frequency* ( $tf$ ) menggunakan persamaan (1) (Fatkhul Amin, 2012).

$$tf = tf_{ij} \quad (1)$$

Dengan  $tf$  adalah *term frequency*, dan  $tf_{ij}$  adalah banyaknya kemunculan *term*  $t_i$  dalam dokumen  $d_j$ . *Term frequency* ( $tf$ ) dihitung dengan menghitung banyaknya kemunculan *term*  $t_i$  dalam dokumen  $d_j$ .

Perhitungan *Inverse Document Frequency* ( $idf$ ), menggunakan persamaan (2)

$$idf_i = \log \frac{N}{df_i} \quad (2)$$

Dengan  $idf_i$  adalah *inverse document frequency*,  $N$  adalah jumlah dokumen yang terambil oleh sistem, dan  $df_i$  adalah banyaknya dokumen dalam koleksi dimana *term*  $t_i$  muncul di dalamnya, maka Perhitungan  $idf_i$  digunakan untuk mengetahui banyaknya *term* yang dicari ( $df_i$ ) yang muncul dalam dokumen lain yang ada pada *database*.

Perhitungan *term frequency Inverse Document Frequency* ( $tfidf$ ), menggunakan persamaan (3)

$$W_{ij} = tf_{ij} \cdot \log \frac{N}{df_i} \quad (3)$$

Dengan  $W_{ij}$  adalah bobot dokumen,  $N$  adalah Jumlah dokumen yang terambil oleh sistem,  $tf_{ij}$  adalah banyaknya kemunculan term  $t_i$  pada dokumen  $d_j$ , dan  $df_i$  adalah banyaknya dokumen dalam koleksi dimana term  $t_i$  muncul di dalamnya. Bobot dokumen ( $W_{ij}$ ) dihitung untuk didapatkannya suatu bobot hasil perkalian atau kombinasi antara *term frequency* ( $tf_{ij}$ ) dan *Inverse Document Frequency* ( $idf$ ).

Perhitungan Jarak query, menggunakan persamaan (4)

$$|q| = \sqrt{\sum_{i=1}^t (W_{iq})^2} \quad (4)$$

Dengan  $|q|$  adalah Jarak query, dan  $W_{iq}$  adalah bobot query dokumen ke- $i$ , maka Jarak query ( $|q|$ ) dihitung untuk didapatkan jarak query dari bobot query dokumen ( $W_{iq}$ ) yang terambil oleh sistem. Jarak query bisa dihitung dengan persamaan akar jumlah kuadrat dari query.

Perhitungan Jarak Dokumen, menggunakan persamaan (5)

$$|d_j| = \sqrt{\sum_{i=1}^t (W_{ij})^2} \quad (5)$$

Dengan  $|d_j|$  adalah jarak dokumen, dan  $W_{ij}$  adalah bobot dokumen ke- $i$ , maka Jarak dokumen ( $|d_j|$ ) dihitung untuk didapatkan jarak dokumen dari bobot dokumen dokumen ( $W_{ij}$ ) yang terambil oleh sistem. Jarak dokumen bisa dihitung dengan persamaan akar jumlah kuadrat dari dokumen.

Menghitung *index terms* dari dokumen dan query ( $q, d_j$ ). menggunakan persamaan (6)

$$q, d_j = \sum_{i=1}^t W_{iq} \cdot W_{ij} \quad (6)$$

Dengan  $W_{ij}$  adalah bobot term dalam dokumen,  $W_{iq}$  adalah bobot query.

Pengukuran *Cosine Similarity* menghitung nilai *kosinus* sudut antara dua *vector* menggunakan persamaan (7)

$$Sim(q, d_j) = \frac{q \cdot d_j}{|q| * |d_j|} \quad (7)$$

Similaritas antara query dan dokumen atau  $Sim(q, d_j)$  berbanding lurus terhadap jumlah bobot query ( $q$ ) dikali bobot dokumen ( $d_j$ ) dan berbanding terbalik terhadap akar jumlah kuadrat

$q$  ( $|q|$ ) dikali akar jumlah kuadrat dokumen ( $|d_j|$ ). Perhitungan similaritas menghasilkan bobot dokumen yang mendekati nilai 1 atau menghasilkan bobot dokumen yang lebih besar dibandingkan dengan nilai yang dihasilkan dari perhitungan *inner product*.

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Vector Space Model

Untuk kepentingan analisis data di dalam melakukan penelitian ini dibutuhkan beberapa sampel data yang diambil dari tiga buah judul buku yaitu :

D1 : Membuat Website Canggih dengan jQuery untuk Pemula.

D2 : Panduan Praktis Belajar Internet Untuk Pemula.

D3 : Panduan Membuat Aplikasi Inventory Barang Dengan Visual Basic.

Jadi total dokumen ada 3. Apabila dilakukan pencarian dokumen dengan kata kunci :

Q : cara membuat web untuk pemula  
dokumen manakah yang paling relevan ?

Untuk menjawab pertanyaan di atas perlu dilakukan suatu proses perhitungan pembobotan atau perankingan dari masing-masing judul buku tersebut sehingga dari hasil pembobotan atau perankingan tersebut akan terlihat judul buku yang mana yang relevan atau yang paling diprioritaskan. Di dalam penerapan metode *Vector Space Model* ada beberapa tahapan proses pengolahan data terlebih dahulu antara lain :

Dari tahapan pembobotan atau perankingan dengan *Vector Space Model* agar mempermudah di dalam proses perhitungan dengan tahapan-tahapan persamaan yang telah dijelaskan sebelumnya pada Sub Bab 2.10, dibuat sebuah tabel ilustrasi perhitungan *Vector Space Model* seperti pada tabel 4.8 di bawah ini.

Tabel 3.1 Ilustrasi Perhitungan Vector Space Model

Trens	Jumlah $tf = tf_{ij}$				
	Q	D1	D2	D3	$df_i$
buat	1	1	0	1	2
website	0	1	0	0	1
canggih	0	1	0	0	1
jquery	0	1	0	0	1
pla	1	1	1	0	2
pandu	0	0	1	1	2
praktis	0	0	1	0	1
ajar	0	0	1	0	1
internet	0	0	1	0	1

Tremis	Jumlah $tf = tf_{ij}$				$df_i$
	Q	D1	D2	D3	
aplikas	0	0	0	1	1
inventory	0	0	0	1	1
barang	0	0	0	1	1
visual	0	0	0	1	1
basic	0	0	0	1	1

Tabel 3.2 Ilustrasi Perhitungan Vector Space Mode (Lanjutan)

Tremis	$N/df_i$	$idf_i = \log \frac{N}{df_i}$
buat	3/2=1,5	0,1761
website	3/1=3	0,4771
canggih	3/1=3	0,4771
jquery	3/1=3	0,4771
pla	3/2=1,5	0,1761
pandu	3/2=1,5	0,1761
praktis	3/1=3	0,4771
ajar	3/1=3	0,4771
internet	3/1=3	0,4771
aplikas	3/1=3	0,4771
inventory	3/1=3	0,4771
barang	3/1=3	0,4771
visual	3/1=3	0,4771
basic	3/1=3	0,4771

Tabel 3.3 Ilustrasi Perhitungan Vector Space Model (Lanjutan)

Tremis	Q	D1	D2	D3
buat	0,1761	0,1761	0	0,1761
website	0	0,4771	0	0
canggih	0	0,4771	0	0
jquery	0	0,4771	0	0
pla	0,1761	0,1761	0,1761	0
pandu	0	0	0,1761	0,1761
praktis	0	0	0,4771	0
ajar	0	0	0,4771	0
internet	0	0	0,4771	0
aplikas	0	0	0	0,4771

Tremis	Q	D1	D2	D3
inventory	0	0	0	0,4771
barang	0	0	0	0,4771
visual	0	0	0	0,4771
basic	0	0	0	0,4771

Keterangan : *Tremis* adalah hasil indexing, *Q* adalah jumlah kemunculan *tremis* pada *Query*, (*D1,D2 dan D3*) jumlah kemunculan *tremis*,  $df_i$  adalah banyaknya dokumen dalam koleksi di mana term  $t_i$  muncul di dalamnya,  $N/df_i$  total dokumen dibagi banyaknya dokumen dalam koleksi di mana term  $t_i$  muncul di dalamnya,  $idf_i$  adalah nilai dari  $\log \frac{N}{df_i}$  dan (*Q, D1,D2 dan D3*) kolom terakhir adalah bobot dokumen dan bobot *query*.

Dari tabel di atas persamaan (1), persamaan (2) dan persamaan (3):

Similarity Analysis :

Perhitungan Jarak Dokumen dengan menggunakan persamaan (5):

$$/ D1 / = \sqrt{0,1761^2+0,4771^2+0,4771^2+0,4771^2+0,1761^2} = \sqrt{0,744896} = 0,8631$$

$$/ D2 / = \sqrt{0,1761^2+0,1761^2+0,4771^2+0,4771^2+0,4771^2} = \sqrt{0,744896} = 0,8631$$

$$/ D3 / = \sqrt{0,1761^2+0,1761^2+0,4771^2+0,4771^2+0,4771^2+0,4771^2+0,4771^2} = \sqrt{1,200144} = 1,0955$$

Perhitungan Jarak *Query* dengan menggunakan persamaan (4):

$$/ Q / = \sqrt{0,1761^2+0,1761^2} = \sqrt{0,062022} = 0,249$$

Menghitung *index terms* dari dokumen dan *query* menggunakan persamaan (6).

$$Q * D1 = 0,1761 * 0,1761 + 0,1761 * 0,1761 = 0,0620$$

$$Q * D2 = 0,1761 * 0,1761 = 0,0310$$

$$Q * D3 = 0,1761 * 0,1761 = 0,0310$$

Menghitung similaritas dokumen dan meranking menggunakan persamaan (7) pada Sub Bab 2.10.

$$\frac{Q * D1}{|Q|*|D1|} = \frac{0,0620}{0,249 * 0,8631} = 0,2885$$

$$\frac{Q * D2}{|Q|*|D2|} = \frac{0,0310}{0,249 * 0,8631} = 0,1442$$

$$\frac{Q * D3}{|Q| * |D3|} = \frac{0,0310}{0,249 * 1,0955} = 0,1136$$

Dari proses similaritas di atas dapat diambil ranking dari setiap dokumen yaitu :

Rank 1 : Dokumen 1 = 0,2885

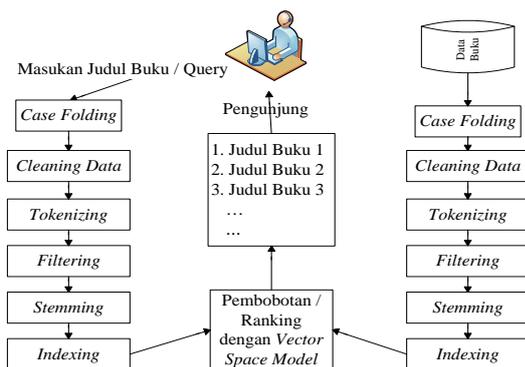
Rank 2 : Dokumen 2 = 0,1442

Rank 3 : Dokumen 3 = 0,1136

Jadi dokumen yang paling relevan dengan kata kunci adalah dokumen 1.

### 3.2 Skema Pencarian Data Buku

Skema rancangan pencarian data buku menjelaskan bagaimana alur pemrosesan sistem pencarian data buku yang nantinya akan dibangun. Dari gambar tersebut dapat dilihat proses diawali dari memasukkan judul buku atau *query* pada sistem oleh Pengunjung, kemudian query tersebut dilakukan beberapa proses yaitu *case folding*, *cleaning data*, *tokenizing*, *filtering*, *stemming* dan *indexing* sehingga akan dapat menghasilkan bobot atau ranking dari judul buku yang relevan. Pengunjung akan mendapatkan informasi berupa sejumlah buku yang relevan dan dapat dilihat berapa bobot untuk masing masing buku yang direkomendasikan oleh sistem.



Gambar 3.1 Skema Pencarian Data Buku

### 3.3 Hasil

#### 3.1 Halaman Utama

Halaman utama pada sistem merupakan halaman yang akan diakses oleh pengunjung perpustakaan untuk mulai melakukan pencarian buku sesuai dengan kebutuhan mereka. Halaman ini berisi informasi tentang semua buku yang ada pada perpustakaan STMIK-AMIK Riau.



Gambar 3.2 Tampilan Halaman Utama

Di halaman utama pada sistem pencarian terdiri dari No, Judul buku, Pengarang, Penerbit, Tahun buku, Jumlah Halaman, Rak dan bobot. Pada bagian bobot awalnya bernilai “0”, nilai untuk masing-masing bobot akan muncul apabila mulai dilakukan proses pencarian buku.

#### 3.2 Pengujian Pembobotan Dengan Vector Space Model

Prose pembobotan dilakukan Setelah proses *case folding*, proses *cleaning data*, proses *tokenizing*, proses *filtering*, proses *stemming* dan *indexing* selesai dilakukan. Tahap pembobotan dilakukan dengan mengambil hasil indexing menjadi *term* di dalam tabel ilustrasi perhitungan *Vector Space Model* kemudian dilakukan perhitungan-perhitungan dengan persamaan-persamaan yang telah ditentukan sebelumnya.

Proses Pembobotan dengan Menggunakan Vector Space Model

Terms	Q1	D1	D2	D3	q1	q2	q3	Q	D1	D2	D3
judul	1	1	0	1	2	1,5	0,1761	0,1761	0,1761	0	0,1761
website	0	1	0	0	1	0	0,4771	0	0,4771	0	0
canggih	0	1	0	1	0	0	0,4771	0	0,4771	0	0
query	0	1	0	0	1	0	0,4771	0	0,4771	0	0
pda	1	1	1	0	2	1,5	0,1761	0,1761	0,1761	0,1761	0
pandu	0	0	1	1	2	1,5	0,1761	0	0	0,1761	0,1761
praktis	0	0	1	0	1	0	0,4771	0	0	0,4771	0
ajar	0	0	1	0	1	0	0,4771	0	0	0,4771	0
internet	0	0	1	0	1	0	0,4771	0	0	0,4771	0
aplikas	0	0	0	1	1	0	0,4771	0	0	0	0,4771
inventory	0	0	0	1	1	0	0,4771	0	0	0	0,4771
barang	0	0	0	1	1	0	0,4771	0	0	0	0,4771
visual	0	0	0	1	1	0	0,4771	0	0	0	0,4771
basic	0	0	0	1	1	0	0,4771	0	0	0	0,4771
Q1 x D144 = 0,062						D144 = 0,0631	Q  0,249		H = 144 <b>0,2885</b>		
Q1 x D145 = 0,031						D145 = 0,0631	Q  0,249		H = 145 <b>0,1442</b>		
Q1 x D146 = 0,031						D146 = 1,0955	Q  0,249		H = 146 <b>0,1136</b>		

Gambar 3.3 Pengujian Proses Pembobotan dengan Vector Space Model

Dari hasil perhitungan pembobotan di atas maka sistem memberikan jawaban atau hasil perhitungan yang menunjukkan judul buku manakah yang paling relevan dan yang mana yang paling diprioritaskan.



No	Judul	Pengarang	Penerbit	Tahun	Jumlah Halaman
1	Membuat website canggih dengan Query untuk pemula	Toni Kun	Medakita	2010	115
2	Panduan Praktis Belajar Internet Untuk Pemula	Andika Pratama	Klik Media	2009	167
3	Panduan Membuat Aplikasi Inventory Barang Dengan Visual Basic	Bursaff Nugroho	RumpuTelno.com	2014	240

**Gambar 3.4 Hasil Pembobotan Judul Buku**

Pada gambar 3.4 hasil pencarian buku dengan kata kunci “Cara membuat web untuk pemula” menghasilkan bobot judul buku yang tertinggi bernilai “0,2885” dengan judul buku “Membuat website canggih dengan jquery untuk pemula”, selanjutnya untuk bobot tertinggi kedua bernilai “0,1442” dengan judul buku “Panduan Praktis Belajar Internet Untuk Pemula” dan yang terakhir bernilai bobot “0.1136” dengan judul buku “Panduan Membuat Aplikasi Inventory Barang Dengan Visual Basic”.

#### 4. KESIMPULAN

Dari analisa dan pembahasan yang penulis lakukan pada bab-bab sebelumnya dapat diambil suatu kesimpulan yaitu:

1. Penerapan *Vector Space Model* pada sistem pencarian buku perpustakaan dapat mempermudah pengunjung dalam melakukan pencarian informasi buku yang relevan dan sesuai dengan kebutuhan pengunjung.
2. Metode *Vector Space Model* dapat diterapkan pada sistem pencarian buku yang dinamis.
3. Dengan sistem pencarian buku perpustakaan ini dapat dilihat nilai bobot dari masing-masing buku yang relevan.

#### DAFTAR PUSTAKA

- Mohammed Said Abual-Rub, Rosni Abdullah dan Nur Aini Abdul Rashid “ A Modified Vector Space Model for Protein Retrieval.” IJCSNS Vol.7 No.9, Penang, 2007.
- Jitendra Nath Singh dan Sanjay Kumar Dwivedi “ Analysis of Vector Space Model in Information Retrieval.” CTNGC , Lucknow, 2012.
- Giat Karyono dan Fandy Setyo Utomo “ Temu Balik Informasi pada Dokumen Teks Berbahasa Indonesia dengan Metode Vector Space Retrieval Model.” Semantik, Semarang, 2012.
- Fatkhul Amin “ Sistem Temu Kembali Informasi dengan Metode Vector Space Model.” JSIB, Semarang, 2012.
- Suciadi Ciptono Rahayu, Ema Rachmawati dan Ade Romadhony “ Implementasi Generalized Vector Space Model pada Pencarian Produk Berbasis Opini Produk.” 2012
- Fatkhul Amin “ Implemantasi Search Engine ( Mesin Pencarian ) Menggunakan Metode Vector Space Model.” Dinamika Teknik Vol.5 No.1, Semarang, 2011.
- Ledy Agusta “ Perbandingan Algoritma Stemming Porter dengan Algoritma Nazief dan Adriani Untuk Stemming Dokumen Teks Berbahasa Indonesia.” IJCSNS Vol.7 No.9, Penang, 2007.
- Mardi Siswo Utomo “Implementasi Stemmer Tala pada Aplikasi Berbasis Web.” ISSN Vol.18 No.1, Stikubank, 2013.
- Sugiarti Yuni (2013). “Analisis dan Perancangan UML [ Unified Modeling Language ] Generated VB.6.” 1. Yogyakarta: Graha Ilmu. 33-82.