Learning Vector Quantization untuk Klasifikasi Abstrak Tesis

Fajar Rohman Hariri*[1], Ema Utami[2], Armadyah Amborowati[3]
[1,2,3]Magister Teknik Informatika STMIK AMIKOM Yogyakarta
E-mail: *[1]**dosendeso@gmail.com**, [2]ema.u@amikom.ac.id, [3]armadyah.a@amikom.ac.id

Abstrak

Data berukuran besar yang sudah disimpan jarang digunakan secara optimal karena manusia seringkali tidak memiliki waktu dan kemampuan yang cukup untuk mengelolanya. Data bervolume besar seperti data teks, jauh melampaui kapasitas pengolahan manusia yang sangat terbatas. Kasus yang disoroti adalah data abstrak tugas akhir mahasiswa jurusan teknik informatika Universitas Trunojoyo Madura. Dokumen tugas akhir oleh mahasiswa terkait hanya diupload pada SIMTAK (Sistem Informasi Tugas Akhir) dan pelabelan bidang minat penelitian dilakukan manual oleh mahasiswa tersebut, sehingga akan ada kemungkian saat mahasiswa mengisi bidang minat tidak sesuai. Untuk menanggulangi hal tersebut, diperlukan adanya mekanisme pelabelan dokumen secara otomatis, untuk meminimalisir kesalahan. Pada penelitian kali ini dilakukan klasifikasi dokumen abstrak tugas akhir menggunakan metode Learning Vector Quantization (LVQ). Data abstrak diklasifikasikan menjadi 3 yaitu SI RPL (Sistem Informasi – Rekayasa Perangkat Lunak), CAI (Computation – Artificial Intelligence) dan Multimedia. Dari berbagai ujicoba yang dilakukan didapatkan hasil metode LVQ berhasil mengenali 90% data abstrak, dengan berhasil mengenali 100% bidang minat SI RPL dan CAI, dan hanya 70% untuk bidang minat Multimedia. Dengan kondisi terbaik didapatkan dengan parameter reduksi dimensi 20% dan nilai learning rate antara 0,1-0,5.

Kata Kunci *— Data Mining, Klasifikasi, Learning Vector Quantization*

Abstract

Huge size of data that have been saved are rarely used optimally because people often do not have enough time and ability to manage. Large volumes of data such as text data, exceed human processing capacity. The case highlighted was the final project abstract data from informatics engineering student Trunojoyo University. Documents abstract just uploaded on SIMTAK (Final Project Information System) and the labeling of the areas of interest of research is done manually by the student, so that there will be a possibility to fill the field of interest while the student is not appropriate. To overcome this, we need a mechanism for labeling a document automatically, to minimize errors. In the present study conducted abstract document classification using Learning Vector Quantization (LVQ). Abstract data classified into three class, SI RPL, CAI and Multimedia. Of the various tests carried out showed that LVQ method successfully recognize 90% of abstract data, to successfully identify 100% interest in the field of RPL SI and CAI, and only 70% for areas of interest Multimedia. With the best conditions obtained with the parameter dimension reduction of 20% and the value of learning rate between 0.1-0.5.

Keywords *— Data Mining, Classification, Learning Vector Quantization*

1. PENDAHULUAN

Perkembangan teknologi telah mengakibatkan meningkatnya data dalam jumlah besar. Data berukuran besar yang sudah disimpan jarang digunakan secara optimal karena manusia seringkali tidak memiliki waktu dan kemampuan yang cukup untuk mengelolanya [1]. Data bervolume besar seperti data teks, jauh melampaui kapasitas pengolahan manusia yang sangat terbatas [2].

Kasus yang disoroti adalah data abstrak tugas akhir mahasiswa jurusan teknik informatika Universitas Trunojoyo Madura. Ada 3 bidang minat yang bisa diambil yaitu SI RPL (Sistem Informasi – Rekayasa Perangkat Lunak), CAI (Computation – Artificial Intelligence) dan Multimedia. Dokumen tugas akhir oleh mahasiswa terkait hanya diupload pada SIMTAK (Sistem Informasi Tugas Akhir) dan pelabelan bidang minat penelitian dilakukan manual oleh mahasiswa tersebut, sehingga akan ada kemungkian saat mahasiswa mengisi bidang minat tidak sesuai. Untuk menanggulangi hal tersebut, diperlukan adanya mekanisme pelabelan dokumen secara otomatis, untuk meminimalisir kesalahan.

Data mining sangat sesuai untuk diterapkan pada data berukuran besar [3]. Metode data mining yang akan diterapkan dalam penelitian ini adalah klasifikasi. Klasifikasi dokumen adalah proses melabeli dokumen sesuai dengan kategori yang dimilikinya.

Adapun beberapa penelitian mengenai klasifikasi dokumen yang menginspirasi dilakukan penelitian pada tesis kali ini adalah penelitian Klasifikasi Dokument Teks Menggunakan Algoritma Naive Bayes dengan Bahasa Pemograman Java oleh Silfia Andini pada tahun 2013, menggunakan artikel berita, dengan metode naïve bayes bisa membantu user dalam memilih user dalam memilih dan mengkategorikan dokumen, pada penelitian ini diadapatkan bahwa metode Naïve Bayes cukup baik dalam mengklasifikasikan dokumen, namun tidak dijabarkan berapa % akurasinya [4].

Penelitian tetang klasifikasi text pernah dilakukan oleh Jeevanandam Jotheeswaran untuk mengklasifikasikan opini pada tahun 2013 dengan judul *Opinion Mining Using Decision Tree Based Feature Selection Through Manhattan Hierarchical Cluster Measure,* penelitian ini meneliti metode seleksi fitur PCA untuk klasifikasi opini dengan menggunakan IMDb data set. Klasifikasi dengan menggunakan ekstraksi fitur menggunakan PCA akurasinya lebih baik sekitar 5%, namun masih perlu adanya penelitian dan skenario untuk metode PCA [5].

Oleh karena permasalahan diatas maka pada penelitian ini akan dibahas mengenai klasifikasi dokumen teks menggunakan metode LVQ (*Learning Vector Quantization*) algoritma ini dikenal dengan kemampuannya dalam klasifikasi yang mempunyai tinggkat akurasi tinggi dan kecepatan dalam hal proses[6], penelitian ini fokus dalam melihat performa LVQ dalam mengklasifikasikan dokumen abstrak tugas akhir, dan untuk stemming dokumennya menggunakan algoritma nazief & andriani karena algoritma ini memiliki akurasi yang lebih baik dibandingkan algoritma porter [7], dan akan digunakan untuk mengklasifikasi dokumen abstrak tugas akhir mahasiswa teknik informatika Universitas Trunojoyo Madura. Berbeda dengan peneltiian sebelumnya, penelitian ini menggunakan dokumen teks abstrak tugas akhir bahasa Indonesia dan mengklafisikasikan berdasarkan bidang minat penelitian. Penelitian kali ini menggunakan berbagai macam sekenario, dari jumlah data learning dan training, juga dalam hal reduksi dimensinya, sehingga bisa diketahui kondisi optimal dalam klasifikasi menggunakkan metode LVQ.

2. METODE PENELITIAN

2.1. Metode Penelitian

Penelitian ini merupakan penelitian eksperimental, karena untuk mendapatkan performa terbaik algortima LVQ dalam mengklasifikasikan, dilakukan beberapa kali percobaan dengan parameter yang berbeda-beda. Data didapatkan dari abstrak tugas akhir mahasiswa jurusan teknik informatika Universitas Trunojoyo Madura dari 3 bidang minat yaitu SI RPL, CAI dan Multimedia. Ada 120 data file dokumen abstrak dengan ekstensi pdf.

2.2. Metode Analisis Data

Untuk analisis data, dari dokumen abstrak yang diperoleh, akan dilakukan percobaan dengan beberapa scenario ujicoba yang dijelaskan pada tabel 1 di bawah:

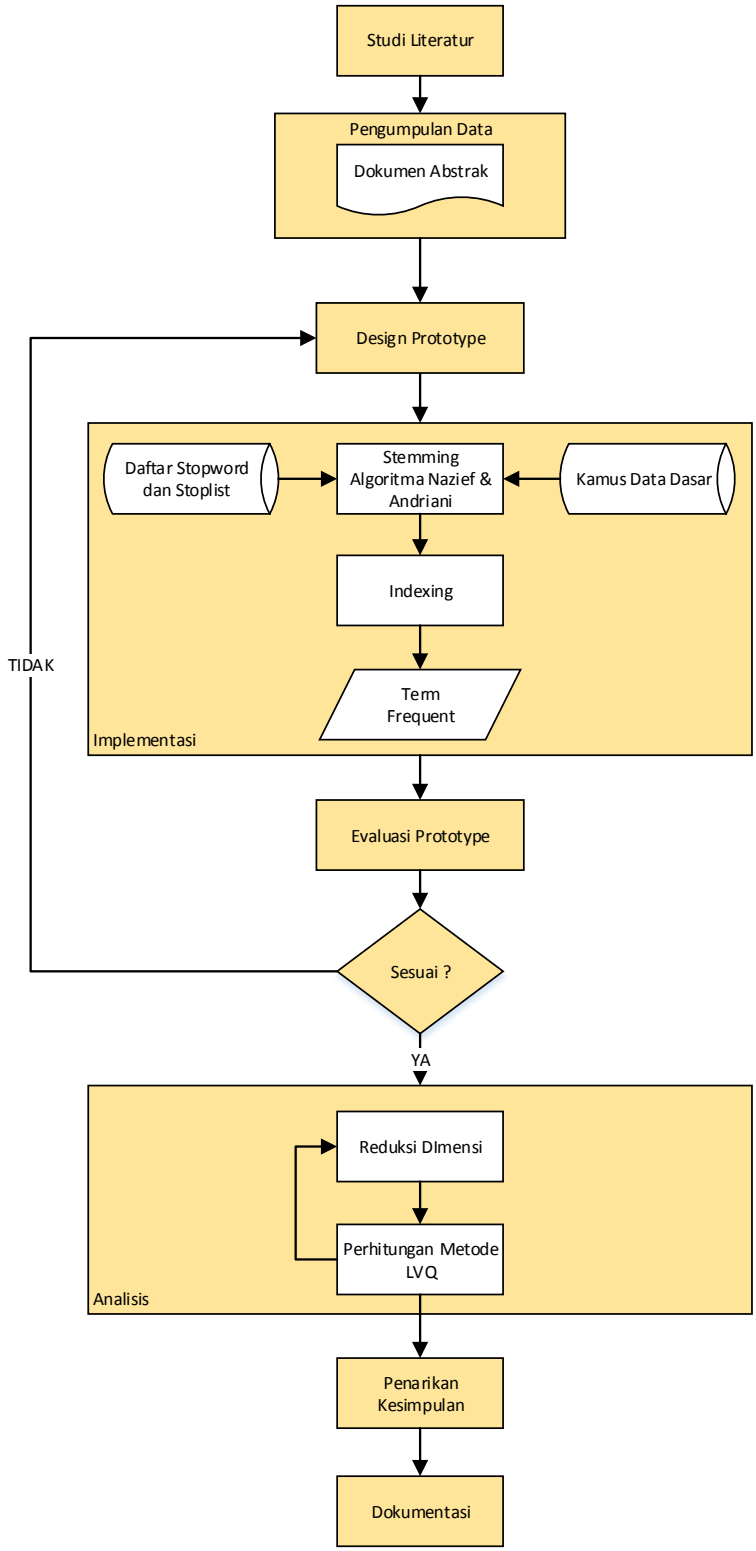
Tabel 1. Skenario Ujicoba

No	Jumlah Data Learning	Jumlah Data Testing	Keterangan
1	100%	100%	Keseluruhan data akan menjadi data learning dan juga data testing.
2	75%	25%	75% data akan digunakan sebagai data learning, dan 25% sisanya dijadikan data testing.
3	50%	50%	75% data akan digunakan sebagai data learning, dan 25% sisanya dijadikan data testing.
4	25%	75%	75% data akan digunakan sebagai data learning, dan 25% sisanya dijadikan data testing.

Selain itu juga akan dilakukan analisis dengan reduksi dimensi yang berbeda-beda. Dengan proses reduksi dimensi, jumlah data learning dan data testing yang berbeda-beda diharapkan bisa diketahui hasil akurasi terbaik dari metode LVQ dalam mengklasifikasikan dokumen.

2.3. Alur Penelitian

Secara umum, penelitian dilakukan dengan beberapa tahapan seperti pada gambar 1:



Gambar 1. Alur Penelitian

Alur penelitian dari penelitian ini adalah:

1. Studi Literatur
   Pada tahap ini dilakukan pengumpulan informasi yang diperlukan untuk penelitian. Informasi yang diperlukan diperoleh dengan mempelajari literatur- literatur mengenai permasalahan klasifikasi dokumen, metode *Learning Vector Quantization* dan agoritma stemming untuk bahasa Indonesia.
2. Pengumpulan Data
   Mencari data-data yang terkait dengan sistem, antara lain mengumpulkan data tugas akhir yang berupa abstrak penelitian mahasiswa jurusan teknik informatika Universitas Trunojoyo Madura.
3. Design Prototipe
   Membuat design prototype untuk proses stemming, pembentukan term frequent.
4. Implementasi
   a. Steeming Algoritma Nazief & Andriani
      Proses pemecahan kata dan mengambil kata dasar dari dokumen abstrak menggunakan Algoritma Nazief & Andriani.
   b. Indexing
      Dari data kata dasar masing-masing dokumen akan dilist dan dilihat frekuanesi masing-masing kata untuk selanjutnya dibentuk sebuah matrix *term frequent.*
5. Evaluasi Prototype
   Dilakukan analisa terhadap prototype yang ada, jika sudah sesuai lanjut ke proses selanjutnya, jika masih belum, dilakukan perbaikan pada prototype sampai sesuai
6. Analisis
   a. Reduksi Dimensi
      Dokumen – dokumen abstrak yang ada akan menghasilkan banyak kata dasar, dan ada beberapa kata yang hanya akan muncul di satu atau sedikit dokumen, sehingga untuk selanjutnya akan dilakukan proses reduksi dimensi dengan membuang kata-kata itu.
   b. Perhitungan Metode LVQ
      Dari data setelah di reduksi dimensi, akan diolah menggunakan metode LVQ untuk menghasilkkan bobot yang akan digunakan untuk proses klasifikasi.
7. Penarikan Kesimpulan
   Setelah dilakukan pengamatan dari beberapa skenario percobaan, dilihat bagaimana akurasi dari metode LVQ, dan dilihat bagaimana kondisi terbaik yang menghasilkan akurasi terbaik.
8. Dokumentasi
   Pada tahap terakhir ini dilakukan penyusunan laporan dan dokumentasi dari pelaksanaan secara keseluruhan.

2.4. Learning Vector Quantization

Jaringan saraf tiruan *Learning Vector Quantization* (LVQ) telah banyak dimanfaatkan untuk pengenalan pola baik berupa citra, suara, dan lain-lain. Jaringan LVQ sering pula digunakan untuk ekstraksi ciri (*feature*) pada proses awal pengenalan pola. Metode Jaringan Syaraf LVQ termasuk dengan *Supervised Learning* dalam penentuan bobot / model pembelajarannya, dimana pada metode LVQ ditentukan hasil seperti apa selama proses pembelajaran. Selama proses pembelajaran nilai bobot disusun dalam suatu range tertentu tergantung pada nilai input yang diberikan. Tujuan pembelajaran ini adalah pengelompokan unit-unit yang hampir sama dalam satu area tertentu. Pembelajaran seperti ini sangat cocok untuk pengelompokan (klasifikasi) pola.

Prinsip kerja dari algoritma LVQ adalah pengurangan node-node tetangganya (*neighbour*), sehingga pada akhirnya hanya ada satu node *output* yang terpilih (*winner node*). Pertama kali yang dilakukan adalah melakukan inisialisasi bobot untuk tiaptiap *class.* Setelah diberikan bobot, maka jaringan diberi input sejumlah dimensi node/neuron input. Setelah input diterima jaringan, maka jaringan mulai melakukan perhitungan jarak vektor yang didapatkan

dengan menjumlah selisih/jarak antara vektor input dengan vektor bobot menggunakan *Euclidean distance*. Secara matematis *Ecluidean Distance* dapat dirumuskan [8]:

$$d_j^2 = \sum_{i=0}^{n-1} (X1(t) - W_{ij})^2 \quad (1)$$

Dimana: d_j^2 = *distance*
X_t = Node data input
W_{ij} = Bobot ke-ij

Setelah diketahui tiap-tiap jarak antara *nodeoutput* dengan *input* maka dilakukan perhitungan jumlah jarak selisih *minimum.* Dimana *node* yang terpilih (*winner*) berjarak minimum akan di update bobot, update bobot *node winner* yang dirumuskan sebagai berikut: Jika sesuai target memakai rumus:

$$W_{ij}(t+1) = W_{ij}(t) + \alpha(t).\left(x_i(t) - W_{ij}(t)\right), j \in Ne \quad (2)$$

Dan jika tidak

$$W_{ij}(t+1) = W_{ij}(t) - \alpha(t).\left(x_i(t) - W_{ij}(t)\right), j \in Ne \quad (3)$$

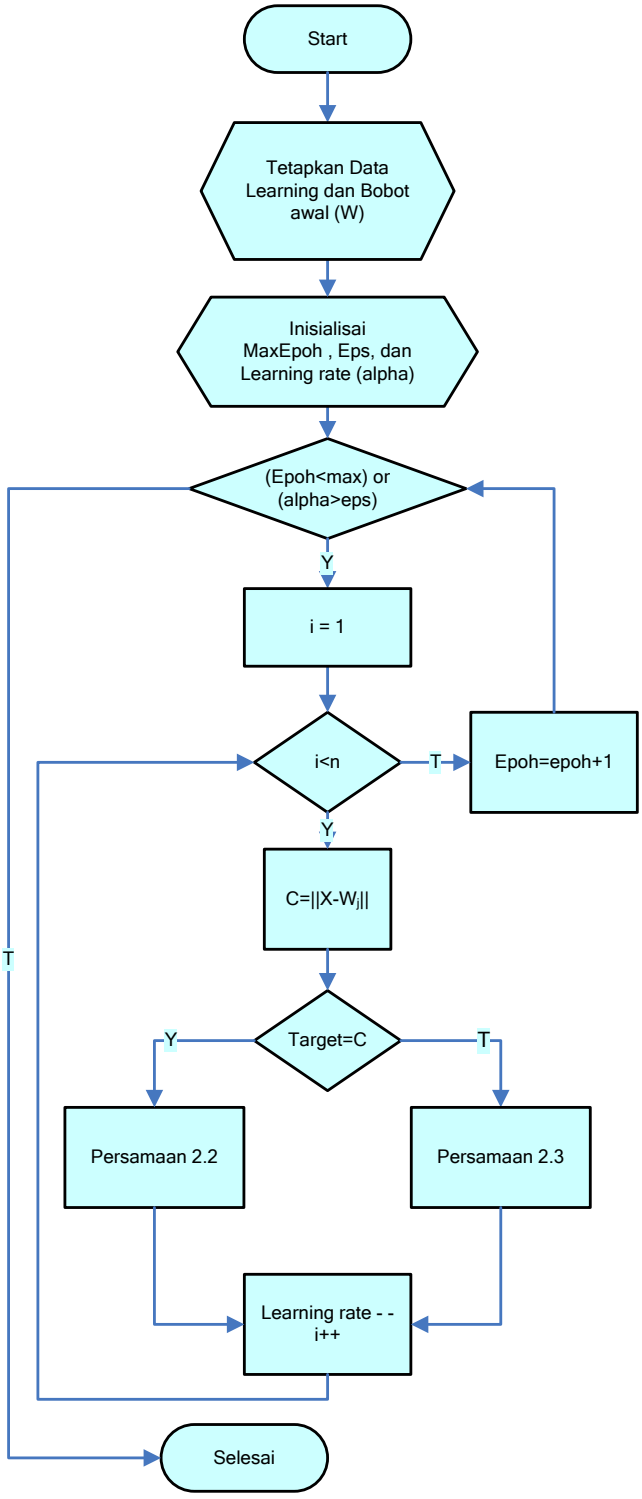
$$0 < (t) < 1$$

Dimana:
x = Input pixel
w = bobot
Ne = Nilai *neighborhood*
t = waktu
i = index node input
j = index node output
α = alpha learning rate

α(t) merupakan *alpha/learning rate* yaitu faktor pengali pada perubahan bobot yang berubah terhadap perubahan *error*. Perubahan *alpha* ini sesuai dengan banyaknya *input* yang masuk. Faktor pengali *alpha*/*learningrate* ini akan selalu berkurang bila tidak ada perubahan *error.* Dalam penelitian ini *alphalearningrate* akan berubah berkurang secara geometris sebagai berikut [8]:

$$(t+1) = 0.1 * \alpha(t) \quad (4)$$

Secara garis besar algoritma *Learning Vector Quantization (LVQ)* sebagai berikut [14]:



Gambar 2. *Flowchart LVQ*

Algoritma dari metode LVQ adalah sebagai berikut:

1. Siapkan data learning, x (m,n) dan target T (1,n)
2. Inisialisasi bobot (W), maksimum epoh (Max Epoh), error minimum yang diharapkan (Eps), *learning rate* (α). Max Epoh dan *learning rate* digunakan untuk menentukan batas ambang komputasi
3. Melakukan proses sebagai berikut selama (epoh<makEpoh) atau (α>eps)
   a. epoh = epoh+1
   b. Kerjakan untuk i=1 sampai n
      1) Tentukan j sedemikian rupa sehingga ||X-Wj|| minimum (Sebut sebagai Cj)
      2) Perbaiki Wj dengan ketentuan
         a) Jika T=Cj maka:
            Wj(baru)=Wj(lama) +α(X-Wj(lama))
         b) Jika T≠Cj maka:
            Wj(baru)=Wj(lama) -α(X-Wj(lama))
   c. Kurangi nilai α
4. Kembali ke langkah ke-3, jika (epoh<makEpoh) atau (α>eps) tidak terpenuhi, selesai.

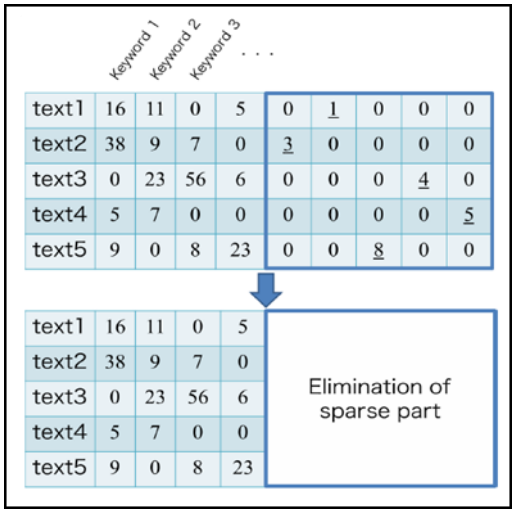
Setelah dilakukan pelatihan, akan diperoleh bobot akhir (W). Bobot-bobot ini nantinya akan digunakan untuk melakukan klasifikasi terhadap data baru.

2.5. Reduksi Dimensi Berdasarkan Karakteristik Data Artikel

Proses reduksi dimensi dilakukan untuk mengecilkan dimensi data sehingga waktu komputasi dibutuhkan lebih sedikit. Namun proses reduksi dimensi harus memperhatikan karakteristik data, karena dimensi yang hilang bisa jadi juga menghilangkan karakteristik data [9]. Oleh karena itu dalam penelitian ini dipilih reduksi dimensi dengan menghilangkan *sparse part*. Term yang memiliki frekuensi yang kebanyakan bernilai nol akan dihilangkan, karena term tersebut merepresentasikan term yang sangat jarang dijumpai dalam data atau dengan kata lain term boleh diabaikan.

Besarnya pengurangan dimensi term sesuai dengan parameter yang ditentukan. Karena ditakutkan proses ini akan menghilangkan karakteristik data, maka dibuat parameter nilai yang berfungsi untuk membatasi reduksi dimensi. Sehingga meskipun dalam suatu term kebanyakan nilainya nol, term tersebut tidak akan dihilangkan jika ada salah satu dari nilai frekuensinya melebihi parameter yang telah ditentukan [2].

Data TF yang sudah masuk dalam database memiliki nilai yang berbeda-beda. Term yang memilik TF yang kebanyankan bernilai nol dapat diabaikan, dengan catatan tidak ada nilai TF yang melebihi parameter yang sudah ditentukan. Dengan kata lain proses reduksi dimensi adalah dengan cara menghilangkan fitur atau term yang yang frekuensinya semua/kebanyakan bernilai 0. Jika ada beberapa yang tidak bernilai 0 maka TF tersebut nilainya harus melebihi parameter tertentu. Hal ini dimaksudkan agar data tidak kehilangan karekteristiknya. Sehingga proses dapat berjalan dengan baik [2].



Gambar 3. Ilustrasi reduksi Dimensi

Dalam gambar 3, ditunjukan misal parameter adalah 40%. Maka reduksi dimensi dilakukan dengan menghilangkan kolom yang nilainya bukan 0 kurang dari 40% data yang ada. Kolom ke 5-9 akan dihilangkan karena kolom tersebut tidak memenuhi syarat.

2.6. Algoritma Nazief & Adriani

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 (kata dasar) dari sebuah kata. Pada umumnya kata dasar pada bahasa Indonesia terdiri dari kombinasi: [10]

Prefiks 1 + Prefiks 2 + Kata dasar + Sufiks 3 + Sufiks 2 + Sufiks 1

Algoritma Nazief & Adriani yang dibuat oleh Bobby Nazief dan Mirna Adriani ini memiliki tahap -tahap sebagai berikut: [10]

1. Pertama cari kata yang akan diistem dalam kamus kata dasar. Jika ditemukan maka diasumsikan kata adalah root word, maka algoritma berhenti.
2. Inflection Suffixes (“-lah”, “-kah”, “-ku”, “-mu”, atau “-nya”) dibuang. Jika berupa particles (“-lah”, “ -kah”, “-tah” atau “-pun”) maka langkah ini diulangi lagi untuk menghapus Possesive Pronouns (“-ku”, “-mu”, atau “-nya”) jika ada.
3. Hapus Derivation Suffixes (“-i”, “ -an” atau “-kan”). Jika kata ditemukan di kamus, maka algoritma berhenti. Jika tidak maka ke langkah 3a
   a. Jika “-an” telah dihapus dan huruf terakhir dari kata tersebut adalah “-k”, maka “k” juga ikut dihapus. Jika kata tersebut ditemukan dalam kamus maka algoritma berhenti. Jika tidak ditemukan maka lakukan langkah 3b.
   b. Akhiran yang dihapus (“-i”, “-an” atau “-kan”) dikembalikan, lanjut ke langkah 4
4. Hilangkan derivation prefixes DP {“di-”,“ke-”,“se-”,“me-”,“be-”,“pe”, “te-”} dengan iterasi maksimum adalah 3 kali:
   a. Langkah 4 berhenti jika:
      1) Terjadi kombinasi awalan dan akhiran yang terlarang seperti padaTabel 2.
      2) Awalan yang dideteksi saat ini sama dengan awalan yang dihilangkan sebelumnya.
      3) Tiga awalan telah dihilangkan

Tabel 2. Kombinasi Awalan dan Akhiran yang tidak diijinkan

Awalan	Akhiran yang tidak diijinkan
be-	-i
di-	-an
ke-	-i , -kan
me-	-an
se-	-i , -kan
te-	-an

b. Identifikasikan tipe awalan dan hilangkan. Awalan ada tipe:
   1) Standar: “di-”, “ke-”, “se-” yang dapat langsung dihilangkan dari kata
   2) Kompleks: “me-”, “be-”, “pe”, “te-” adalah tipe-tipe awalan yang dapat bermorfologi sesuai kata dasar yang mengikutinya. Oleh karena itu, gunakan aturan pada Tabel 3 untuk mendapatkan pemenggalan yang tepat.

Tabel 3. Aturan awalan me-, be-, pe-, te-

Aturan	Format Kata	Pemenggalan
1	berV..	Ber-V | ber-rV
2	berCAP…	Ber-CAP dimana C !=’r’ & P!=’er’
3	berCAerV..	berCaerV dimana C !=’r’
4	Belajar	Bel-ajar
5	beCerC	Be-CerC dimana C !=’r’|’l’
6	terV	Ter-V | te-rV
7	terCerV	Ter-CerV dimana C !=’r’
8	terCP..	Ter-CP dimana C !=’r’ & P!=’er’
9	teCerC..	Te-CerC dimana C !=’r’
10	Me{l|r|w|y}V..	Me-{l|r|w|y}V..
11	Mem{b|f|v}..	Mem-{b|f|v}..
12	Mempe{l|r}	Mem-pe{l|r}
13	Mem{rV|V}	Me-m{rV|V} | Me-p{rV|V}
14	Men{c|d|j|z}..	Men-{c|d|j|z}..
15	MenV	Me-nV | me-tV
16	Meng{g|h|q}	Meng-{g|h|q}
17	mengV	Meng-V | Meng-kV
18	menyV	Meny-sV
19	mempV	Mem-pV dimana V!=’e’
20	Pe{w|y}V	Pe-{w|y}V
21	perV	Per-V | pe-rV
22	perCAP	Per-CAP dimana C !=’r’ & P!=’er’
23	perCaerV	Per-CaerV dimana C !=’r’
24	pem{b|f|v}..	Pem-{b|f|v}..
25	pem{rV|V}	Pe-m{rV|V} | Pe-p{rV|V}
26	pen{c|d|j|z}..	Pen-{c|d|j|z}..
27	penV	Pe-nV | Pe-tV
28	Peng{g|h|q}	Peng-{g|h|q}
29	pengV	Peng-V | peng-kV
30	penyV	Peny-sV
31	pelV	Pe-lV kecuali ajar yang menghasilkan ajar
32	peCerV	peC-erV dimana C!={r|w|y|l|m|n}
33	peCP	Pe-CP dimana C!={r|w|y|l|m|n} dan P!=’er’

c. Cari kata yang telah dihilangkan awalannya ini di dalam kamus. Apabila tidak ditemukan, maka langkah 4 diulangi kembali. Apabila ditemukan, maka keseluruhan proses dihentikan.

5. Apabila setelah langkah 4 kata dasar masih belum ditemukan, maka proses recoding dilakukan dengan mengacu pada aturan pada Tabel 3. Recoding dilakukan dengan menambahkan karakter recoding di awal kata yang dipenggal. Pada Tabel 3, karakter recoding adalah huruf kecil setelah tanda hubung (‘-’) dan terkadang berada sebelum tanda kurung. Sebagai contoh, kata “menangkap” (aturan 15), setelah dipenggal menjadi “nangkap”. Karena tidak valid, maka recoding dilakukan dan menghasilkan kata “tangkap”.
6. Jika semua langkah telah selesai tetapi tidak juga berhasil maka kata awal diasumsikan sebagai root word dan proses selesai.

Tipe awalan ditentukan melalui langkah-langkah berikut:

a. Jika awalannya adalah: “di-”, “ke-”, atau “se-” maka tipe awalannya secara berturut-turut adalah “di-”, “ke-”, atau “se-”.
b. Jika awalannya adalah “te-”, “me-”, “be-”, atau “pe-” maka dibutuhkan sebuah proses tambahan untuk menentukan tipe awalannya.
c. Jika dua karakter pertama bukan “di-”, “ke-”, “se-”, “te-”, “be-”, “me-”, atau “pe-” maka berhenti.
d. Jika tipe awalan adalah “none” maka berhenti. Jika tipe awalan adalah bukan “none” maka awalan dapat dilihat pada Tabel 4. Hapus awalan jika ditemukan.

Untuk aturan penghapusan awalan ada pada tabel 4 dan tabel 5.

Tabel 4. Cara menentukan tipe awalan “te-“

Following Character				Tipe Awalan
Set 1	Set 2	Set 3	Set 4	
“-r-”	“-r-”	-	-	None
“-r-”	-	-	-	Ter-luluh
“-r-”		“-er-”	Vowel	Ter
“-r-”		“-er-”	Not vowel	Ter-
“-r-”		Not “-er-”	-	Ter
Not (vowel or “r”)	“-er-”	Vowel	-	none
Not (vowel or “r”)	“-er-”	Not vowel	-	None

Tabel 5. Cara menentukan tipe awalan “te-“

Tipe awalan	Awalan yang harus dihapus
Di-	Di-
Ke-	Ke-
Se-	Se-
Te-	Te-
Ter-	Ter-
Ter-luluh	Ter-

Untuk mengatasi keterbatasan pada algoritma di atas, maka ditambahkan aturan-aturan di bawah ini:

1. Aturan untuk reduplikasi
   a. Jika kedua kata yang dihubungkan oleh kata penghubung adalah kata yang sama maka root word adalah bentuk tunggalnya, contoh: “buku-buku” root word-nya adalah “buku”.
   b. Kata lain, misalnya “bolak-balik”, “berbalas-balasan, dan ”seolaholah”. Untuk mendapatkan root word-nya, kedua kata diartikan secara terpisah. Jika keduanya

memiliki root word yang sama maka diubah menjadi bentuk tunggal, contoh: kata "berbalasbalasan", "berbalas" dan "balasan" memiliki root word yang sama yaitu "balas", maka root word "berbalas-balasan" adalah "balas". Sebaliknya, pada kata "bolak-balik", "bolak" dan "balik" memiliki root word yang berbeda, maka root word-nya adalah "bolak-balik".

2. Tambahan bentuik awalan dan akhiran serta aturannya
   a. Untuk tipe awalan "mem-", kata yang diawali dengan awala"memp" memiliki tipe awalan "mem-".
   b. Tipe awalan "meng-", kata yang diawali dengan awalan"mengk-" memiliki tipe awalan "meng-".

3. HASIL DAN PEMBAHASAN

3.1. Implementasi

3.1.1. Preprocessing

Tahapan preprocessing:

1. Tokenisasi
   Untuk proses tokenisasi, dalam membaca isi file pdf pada penelitian ini menggunakan library itextsharp, library ini cukup bagus dalam merubah isi file pdf menjadi text. Setelah dilakukan ujicoba dengan beberapa file pdf yang ada, keberhasilan library itextsharp ditunjukkan oleh tabel 6 berikut:

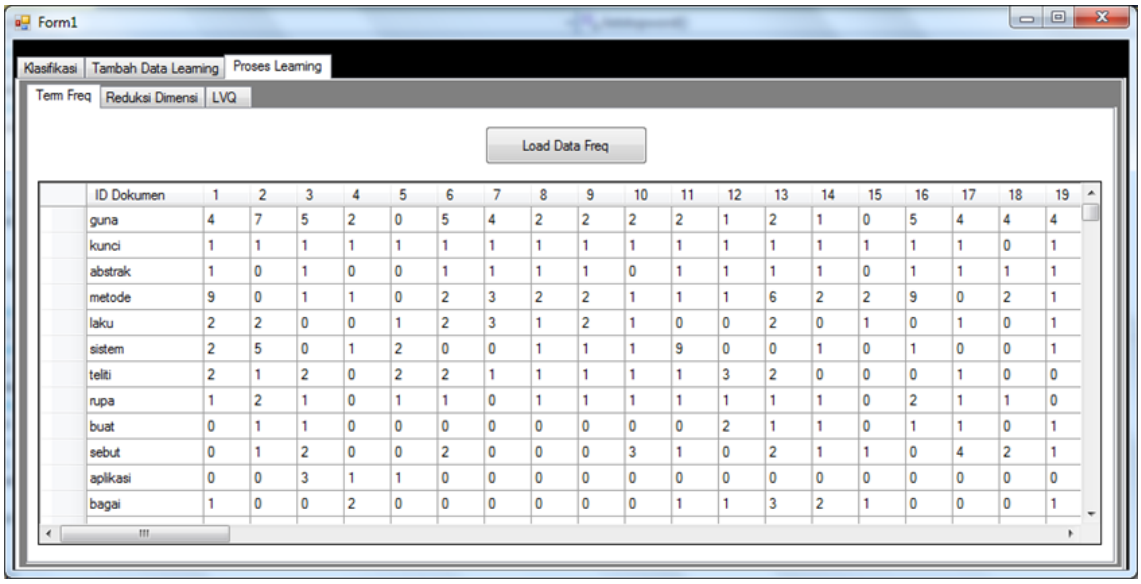
Tabel 6. Keberhasilan library itextsharp

Jenis file PDF	Hasil
Dibuat lewat Ms.Office	Berhasil
Dibuat lewat pdf creator	Berhasil
Dibuat dari Latex	Kurang sempurna (spasi hilang)
Berpassword	Tidak Berhasil
Diproteksi, tidak bisa dicopy	Tidak Berhasil
Terdapat Rumus Matematika	Berhasil

   Untuk file pdf biasa yang dibuat lewat aplikasi pdf creator atau lewat Ms. Word library itextsharp berhasil membaca isi file pdf dengan sempurna, namun kurang sempurna dalam membaca file pdf yang dibuat dari latex, pdf dari latex terbaca, namun text yang dihasilkan tidak ada spasinya, tulisan yang ada menjadi satu. Abstrak yang mengandung rumus matematika berhasil dibaca isinya.
   Dilakukan ujicoba dengan menggunakan file pdf yang terproteksi password dan juga yang tidak bisa dicopy, hasilnya library ini tidak berhasil membaca isinya.
2. Stopword and Stoplist Removal
   Term yang dihasilkan pada proses sebelumnya bermacam-macam, diantara term yang ada, ada beberapa term yang dianggap tidak berguna atau tidak bernilai, kata-kata tersebut contohnya seperti kata sambung, kata depan, dll. Ada total 694 kata yang termasuk dalam daftar kata tersebut. Yang selanjutnya dapat dihilangkan untuk dilakukan proses selanjutnya.
3. Stemming
   Metode stemming yang dipakai adalah Algoritma Nazief & Adriani. Dari 120 data abstrak yang ada, setelah dilakukan proses tokenisasi, stopword removal, selanjutnya untuk setiap kata, diambil kata dasarnya dengan algoritma tersebut dengan menerapkan *regular expression*.

3.1.2. Indexing

Proses selanjutnya setelah preprosesing adalah indexing. Dari 120 data abstrak yang ada, dilakukan preprocessing, dan merubahnya menjadi matrix term frequent, dihasilkan 7171 kata (term), hasil term frquent yang dihasilkan seperti gambar 4 dibawah.



ID Dokumen	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
guna	4	7	5	2	0	5	4	2	2	2	2	1	2	1	0	5	4	4	4
kunci	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
abstrak	1	0	1	0	0	1	1	1	1	0	1	1	1	1	0	1	1	1	1
metode	9	0	1	1	0	2	3	2	2	1	1	1	6	2	2	9	0	2	1
laku	2	2	0	0	1	2	3	1	2	1	0	0	2	0	1	0	1	0	1
sistem	2	5	0	1	2	0	0	1	1	1	9	0	0	1	0	1	0	0	1
teliti	2	1	2	0	2	2	1	1	1	1	1	3	2	0	0	0	1	0	0
rupa	1	2	1	0	1	1	0	1	1	1	1	1	1	1	0	2	1	1	0
buat	0	1	1	0	0	0	0	0	0	0	0	2	1	1	0	1	1	0	1
sebut	0	1	2	0	0	2	0	0	0	3	1	0	2	1	1	0	4	2	1
aplikasi	0	0	3	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
bagai	1	0	0	2	0	0	0	0	0	0	1	1	3	2	1	0	0	0	1

Gambar 4. Hasil Term Frequet

3.1.3. Reduksi Dimensi

Dari 120 data dokumen abstrak setelah diproses menghasilkan 7171 term (kata). Dari 7171 kata yang ada, ada beberapa kata yang hanya muncul di satu atau dua dokumen saja, sehingga kata tersebut dapat dihiraukan. Oleh karena itu, untuk menghilangkan kata-kata tersebut, dan juga untuk mempercepat proses perhitungan, dilakukan proses reduksi dimensi.

Hasil jumlah term yang dihasilkan setelah dilakukan proses reduksi dimensi ada pada Tabel 7 berikut:

Tabel 7. Hasil Reduksi Dimensi

Nilai Reduksi Dimensi	Jumlah Term
10 %	91
20 %	37
30 %	20
40 %	12
50 %	8
60 %	3
70 %	3
80 %	2
90 %	1

Nilai % diatas berarti kata-kata yang diambil adalah kata yang muncul di lebih dari berapa % dokumen. Semakin sedikit parameter, maka akan semakin sedikit pula term yang dihasilkan.

3.2. Hasil Ujicoba

Setelah dilakukan proses reduksi dimensi, selanjutnya dilakukan proses pelatihan menggunakan Learning Vector Quantization. Dari bobot akhir yang dihasilkan, kemudian dihitung akurasinya dengan cara membandingkan hasil asli (nyata) dengan hasil dari metode LVQ. Dengan 100% data digunakan sebagai data testing dan juga data training dan dengan skenario reduksi dimensi yang berbeda, menghasilkan akurasi seperti pada tabel 8 berikut:

Tabel 8. Hasil Akurasi skenario Reduksi Dimensi

Nilai Reduksi Dimensi	Akurasi (%)
10 %	75.56
20 %	84.17
30 %	74.14
40 %	66.67
50 %	56.67
60 %	33.33
70 %	33.33
80 %	33.33
90 %	33.33

Dari tabel dan grafik diatas, pada reduksi dimensi 60%-90% hanya menghasilkan akurasi 33.33%, semua data abstrak terklasifikasi masuk dalam bidang minat multimedia, dan dari tabel 3.8 diketahui yang menghasilkan nilai akurasi terbaik adalah saat dilakukan reduksi dimensi 20% dengan nilai akurasi mencapai 84.17% . Dengan rincian pengenalan untuk masing-masing bidang minat ada pada tabel 9 berikut:

Tabel 9. Rincian akurasi pengenalan reduksi dimensi 20%

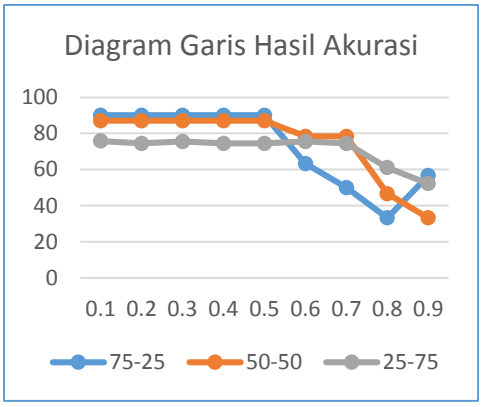
Bidang Minat	Jumlah Dokumen	Jumlah Berhasil Dikenali	Akurasi
SI RPL	40	40	100 %
CAI	40	35	87.5 %
MULTIMEDIA	40	26	65 %

Dari tabel diatas diketahui, metode LVQ paling baik dalam mengklasifikasikan abstrak untuk bidang minat SI RPL dengan akurasi mencapai 100% , dan paling jelek dalam mengklasifikasikan abstrak untuk bidang minat Multimedia yang hanya berhasil mengenali 26 data dari 40 data dan menghasilkan nilai akurasi sebesar 65%.

Setalah itu, dilakukan skenario ujicoba dengan merubah data learning dan testing. Akurasi yang dihasilkan seperti pada tabel 10 dibawah.

Tabel 10**.** Hasil Akurasi Skenario Jumlah Data Training

Learning Rate	75% Training 25% Testing	50% Training 50% Testing	25% Training 75% Testing
0.1	90 %	87 %	75.76 %
0.2	90 %	87 %	74.44 %
0.3	90 %	87 %	75.56 %
0.4	90 %	87 %	74.44 %
0.5	90 %	87 %	74.44 %
0.6	63.33 %	78.33 %	75.56 %
0.7	50 %	78.33 %	74.44 %
0.8	33.33 %	46.67 %	61.11 %
0.9	56.67 %	33.33 %	52.22 %



Gambar 5. Diagram Garis Hasil Akurasi

Dari tabel 10 dan gambar 5, diketahui nilai akurasi terbaik adalah 90%, dengan skenario 75% Data Training – 25% data testing dan dengan nilai learning rate atara 0.1-0.5. Dengan rincian pengenalan untuk masing-masing bidang minat pada tabel 11 berikut:

Tabel 11. Rincian akurasi 75-25 dan learning rate 0.1-0.5

Bidang Minat	Jumlah Dokumen	Jumlah Berhasil Dikenali	Akurasi
SI RPL	40	40	100 %
CAI	40	40	100 %
MULTIMEDIA	40	28	70 %

Dari tabel diatas diketahui, metode LVQ paling baik dalam mengklasifikasikan abstrak untuk bidang minat SI RPL dan juga CAI dengan akurasi mencapai 100% , dan kurang baik dalam mengklasifikasikan abstrak untuk bidang minat Multimedia yang hanya menghasilkan nilai akurasi sebesar 70%.

Dari beberapa sekenario ujicoba, dengan variasi nilai reduksi dimensi, jumlah data learning dan ujicoba, menghasilkan akurasi yang berbeda pula. Dalam penelitian ini, nilai reduksi dimensi sebesar 20% menghasilkan akurasi yang lebih baik. Pada skenario jumlah data learning dan ujicoba, semakin banyak data learning, semakin baik akurasinya.

Hasil terbaik diperoleh dengan reduksi dimensi 20%, menggunakan 75% data learning, dan dari 25% data ujicoba, metode LVQ berhasil mengenali 90% data. Dengan didapatkannya hasil terbaik menggunakan learning rate antara 0,1 – 0,5 , dengan dapat mengenali 100% data untuk bidang minat SI/RPL dan CAI , namun kurang bagus dalam mengenali data abstrak multimedia, dengan hanya dapat mengenali 70% data.

Metode LVQ cukup baik dalam mengklasifikasikan dokumen abstrak, namun dalam penelitian ini metode LVQ kurang bagus dalam mengklasifikasikan abstrak bidang minat multimedia, dikarenakan metode LVQ kurang bagus dalam mengolah data yang beragam, dan pada penelitian kali ini abstrak-abstrak bidang minat multimedia yang ada terlalu beragam apabila dibandingkan dengan 2 bidang minat lainnya, sehingga term yang dihasilkan saat direduksi dimensi kurang bisa memperlihatkan ciri dari bidang minat multimedia.

4. KESIMPULAN

Dari hasil penelitian didapatkan bahwa:

1. Akurasi yang dihasilkan oleh metode LVQ dalam mengklasifikasi dokumen abstrak tugas akhir mahasiswa jurusan teknik informatika Universitas Trunojoyo Madura mencapai 90% untuk keseluruhan data, dengan berhasil mengenali abstrak tugas akhir bidang minat SI RPL dan CAI dengan akurasi 100% , namun untuk bidang minat multimedia hanya menghasilkan akurasi sebesar 70%.

2. Akurasi terbaik sebesar 90% didapatkan dengan kondisi:
   a. Reduksi dimensi dengan parameter 20%.
   b. Perbandingan data testing dan training sebesar 75% data testing dan 25% data training.
   c. Learning rate antara 0,1-0,5.

5. SARAN

Saran dari peneliti untuk penelitian selanjutnya:

1. Untuk proses reduksi dimensi bisa dilakukan dengan metode yang berbeda supaya bisa diketahui mana metode reduksi dimensi yang baik dalam hal klasifikasi abstrak.
2. Perlu adanya perbandingan dengan metode klasifikasi yang lain supaya bisa diketahui mana metode kalsifikasi yang cocok dalam klasifikasi dokemen abstrak.

DAFTAR PUSTAKA

[1] Sitanggang, I. S., Hermadi, I., Edward, 2007, Clustering Menggunakan Self-Organizing Maps (Studi Kasus: Data PPMB IPB), *Jurnal Ilmiah Ilmu Komputer*, Vol 5, No 2.

[2] Toyota, T., Nobuhara, H., 2012, Visualization of the Internet News Based on Efficient Self-Organizing Map Using Restricted Region Search and Dimensionality Reduction, *Journal of Advanced Computational Intelligence and Intelligent Informatics,* Vol 16, No 2, hal 219-226.

[3] Turban, E., 2005, *Decision Support System and Intelligent System*, edisi Bahasa Indonesia Jilid I, Andi, Yogyakarta.

[4] Andini, S., 2013, Klasifikasi Dokument Teks Menggunakan Algoritma Naive Bayes dengan Bahasa Pemograman Java, Jurnal Teknologi Informasi & Pendidikan, Vol 6, No 2, hal 140-147.

[5] Jotheeswaran, J., Kumaraswamy, Y. S., 2013, Opinion Mining Using Decision Tree Based Feature Selection Through Manhattan Hierarchical Cluster Measure, *Journal of Theoretical and Applied Information Technology*, Vol 58, No 1, hal 72-80.

[6] Hadnanto, M. A., 1996, Perbandingan Beberapa Metode Algoritma JST untuk Pengenalan Pola Gambar, *Tugas Akhir*, Teknik Elektronika, ITS Surabaya, Surabaya.

[7] Agusta, L., 2009, Perbandingan Algoritma Stemming Porter dengan Algoritma Nazief & Adriani untuk Stemming Dokumen Teks Bahasa Indonesia, *Konferensi Nasional Sistem dan Informatika*, Bali, 4 November 209.

[8] Kusumadewi, S., 2003, *Artificial Intelligence: Teknik & Aplikasinya*, Graha Ilmu, Yogyakarta.

[9] Yang, Y., Pedersen, J. O., 1997, A Comparative Study on Feature Selection in Text Categorization, *Proceedings of the Fourteenth International Conference on Machine Learning (ICML'97)*, Nashville, Tennessee, USA, 8-12 Juli 1997.

[10] Nazief, B. A. A., Adriani, M., 1996, Confix-Stripping: Approach to Stemming Algorithm for Bahasa Indonesia, *Laporan Penelitian*, Fakultas Ilmu Komputer, Universitas Indonesia.