IMPLEMENTASI EKSTRAKSI FITUR PADA PENGOLAHAN DOKUMEN BERBAHASA INDONESIA

Putu Manik Prihatini
Jurusan Teknik Elektro, Politeknik Negeri Bali
Bukit Jimbaran, P.O.Box 1064 Tuban Badung – Bali. Phone:+62-361-701981, Fax:+62-361-701128
E-mail: manikprihatini@pnb.ac.id

Abstrak: Ekstraksi fitur merupakan proses untuk mencari nilai-nilai fitur yang terkandung dalam dokumen untuk proses *text mining*. Ekstraksi fitur menjadi bagian yang sangat penting dalam pengolahan dokumen pada mesin pencari karena sangat menentukan keberhasilan proses *text mining*. Salah satu metode ekstraksi fitur yang banyak digunakan dan populer adalah TF-IDF. Pada penelitian ini, metode TF-IDF telah diimplementasikan dengan membuat aplikasi menggunakan perangkat lunak Matlab. Dokumen untuk aplikasi diperoleh melalui media digital Detikcom dan disimpan dalam bentuk file teks. Proses pengolahan dokumen melibatkan *case folding*, *tokenization*, *filtering*, *stemming* dan ekstraksi fitur. Hasil ekstraksi fitur berupa matriks yang berisi urutan kata-kata unik dari seluruh dokumen dan nilai-nilai fitur TF-IDF dari setiap kata pada seluruh dokumen.

Kata Kunci: *ekstraksi fitur, TF-IDF, dokumen berbahasa Indonesia*

The Implementation of Extraction Feature on Indonesian Documents' Processing

Abstract: *The extraction feature is a process to look for the values of the features contained in the document for text mining process. Extraction feature becomes a very important part in the processing of documents on the search engines because it determines the success of the process of text mining. One of the feature extraction methods is widely used and popular is the TF-IDF. In this study, the TF-IDF method has been implemented by creating application using Matlab software. Documents that used for application was obtained through digital media Detik.com and stored in text files. The processing of documents involves folding case, tokenization, filtering, stemming, and feature extraction. Feature extraction results in the form of a matrix which contains a sequence of unique words from all the documents and the values of TF-IDF feature of every word in the entire document.*

Key words*: extraction feature, TF-IDF, Indonesian documents*

I. PENDAHULUAN

Di era digital seperti saat ini, perkembangan informasi yang sangat cepat telah diimbangi dengan kemajuan teknologi sebagai media penyajian informasi yang dibutuhkan oleh pengguna. Teknologi mesin pencari informasi yang muncul seiring dengan perkembangan internet telah memberikan kemudahan bagi pengguna dalam memperoleh informasi. Mesin pencari menampilkan puluhan ribu informasi sesuai dengan kriteria yang diinginkan oleh pengguna. Informasi yang ditampilkan tidak hanya dalam bentuk teks, melainkan juga dalam bentuk visual seperti gambar dan video.

Informasi yang ditampilkan oleh mesin pencari dihasilkan melalui proses pengolahan dokumen yang dimiliki oleh server dari mesin pencari terhadap kriteria yang diketikkan oleh pengguna pada kotak teks. Proses pengolahan dokumen dan kriteria melibatkan beberapa proses seperti *case folding*, *tokenization*, *filtering*, *stemming*, ekstraksi fitur dan *text mining*. *Case folding* merupakan proses untuk mengubah seluruh teks dokumen menjadi huruf kecil. *Tokenization* merupakan proses untuk memecah teks dokumen menjadi kata serta menghilangkan angka, tanda baca dan spasi. *Filtering* merupakan proses untuk menghilangkan kata-kata yang tidak bermakna. *Stemming* merupakan proses untuk mencari kata dasar dari setiap kata dengan menghilangkan imbuhan baik awalan maupun akhiran. Ekstraksi fitur merupakan proses untuk mencari nilai-nilai fitur yang terkandung dalam dokumen. *Text mining* merupakan proses untuk menggali informasi yang terkandung dalam dokumen berdasarkan nilai-nilai fitur hasil ekstraksi fitur.

Ekstraksi fitur menjadi bagian yang sangat penting dalam pengolahan dokumen pada mesin pencari karena sangat menentukan keberhasilan proses *text mining.* Jika nilai fitur yang dihasilkan tidak tepat, maka informasi yang digali dalam *text mining* tidak bisa memenuhi kriteria yang diinginkan. Akibatnya, informasi yang ditampilkan oleh mesin pencari tidak akan memenuhi keinginan pengguna. Sampai saat ini, mesin pencari informasi

yang ada masih belum mampu memenuhi kebutuhan pengguna sistem terhadap hasil pencarian yang didapatkan (Chali et al., 2015).

Pada pengolahan dokumen, suatu teks atau dokumen direpresentasikan dalam bentuk *bag-of-words* (BOW), yang mampu memunculkan ruang kata berdimensi tinggi (Zhao et al., 2015). Salah satu metode ekstraksi fitur dengan konsep BOW yang banyak digunakan dan populer adalah TF-IDF (Huang et al., 2006, Li et al., 2007, Liu et al., 2007, Liu et al., 2011, Haddi et al., 2013, Malandrakis et al., 2013, Ceci et al., 2014, Corrêa et al., 2014, Hai et al., 2014, Lizhen et al., 2014, Noh et al., 2015, Tutkan et al., 2016).

Untuk itu, melalui penelitian ini, penulis ingin membuat implementasi dari metode ekstraksi fitur TF-IDF terhadap dokumen berbahasa Indonesia. Adapun tujuan penelitian ini adalah untuk mengetahui fitur-fitur apa saja yang dihasilkan oleh metode TF-IDF terhadap dokumen berbahasa Indonesia, sehingga hasilnya bisa digunakan untuk tahapan selanjutnya dalam *text mining*.

II. METODE PENELITIAN

Pada penelitian sebelumnya, penulis telah menghasilkan implementasi terhadap proses pra pengolahan dokumen yang melibatkan *case folding*, *tokenization*, *filtering* dan *stemming*. Hasil penelitian tersebut digunakan pada penelitian ini untuk melakukan ekstraksi fitur dengan metode TF-IDF terhadap dokumen berbahasa Indonesia. Adapun rancangan arsitektur yang digunakan pada penelitian ini dapat dilihat pada Gambar 1.

Case Folding

Case folding merupakan proses untuk mengubah semua teks dokumen menjadi huruf kecil.

Tokenization

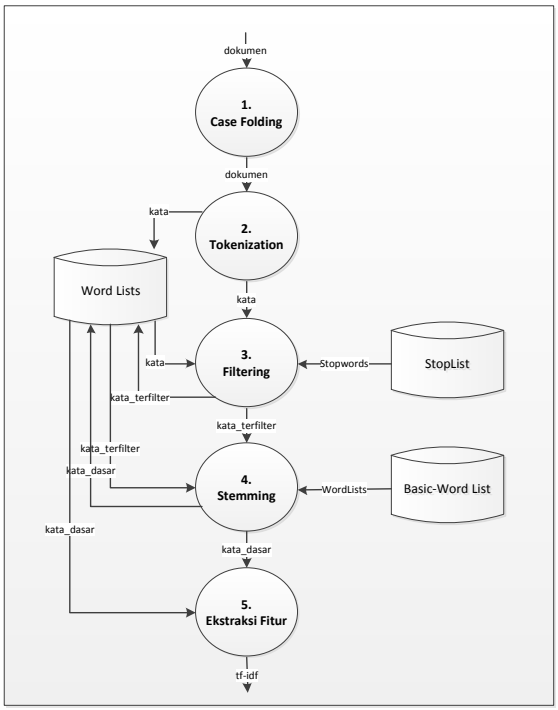
Tokenization merupakan proses untuk memecah teks dokumen menjadi kalimat, kemudian memecahnya menjadi kata-kata. Proses ini juga dilakukan untuk menghilangkan angka, tanda baca dan spasi.

Filtering

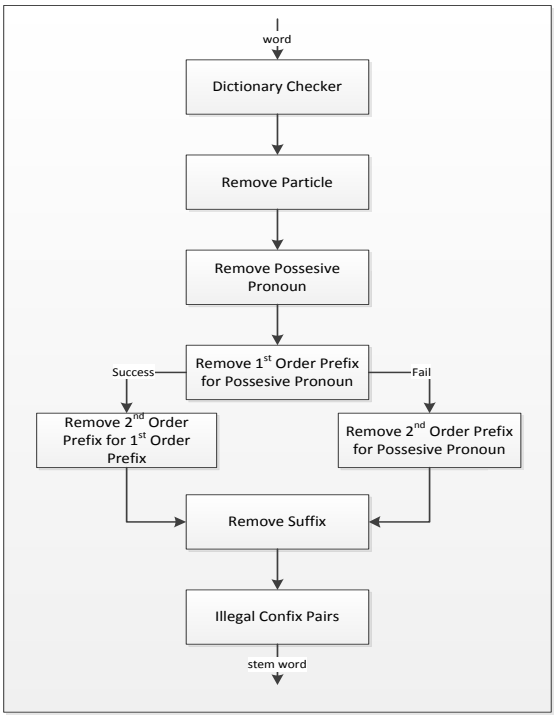
Filtering merupakan proses untuk membuang kata-kata yang tidak bermakna dalam dokumen. Daftar kata-kata yang tidak bermakna disimpan dalam sebuah basis pengetahuan bernama stoplist. Pada penelitian sebelumnya, stoplist yang digunakan terdiri dari 906 kata, yang merupakan gabungan dari stop list dan most common words yang dipublikasikan oleh (Tala, 2003), serta stop list yang dihasilkan oleh penulis.

Stemming

Stemming merupakan proses untuk mencari kata dasar dari setiap kata dalam dokumen dengan membuang imbuhan, baik awalan maupun akhiran. Proses ini menggunakan basis pengetahuan bernama *basic-words list* sebagai kamus kata dasar. Pada penelitian sebelumnya, *basic-words list* terdiri dari 30.342 kata. Algoritma *stemming* yang digunakan adalah dengan memodifikasi algoritma Porter-like Stemmer for Indonesian yang dipublikasikan oleh Tala (2003). Bagan alir dari algoritma *stemming* yang digunakan dapat dilihat pada Gambar 2.



Gambar 1.Arsitektur Umum Penelitian



Gambar 2.Proses Stemming

Ekstraksi Fitur dengan TF-IDF

Pada skema Term Frequency-Inverse Document Frequency (TF-IDF), TF dihitung berdasarkan jumlah kemunculan setiap kata dalam tiap dokumen, dan IDF dihitung berdasarkan jumlah kemunculan kata dalam keseluruhan dokumen (Blei et al., 2003). Setelah melalui proses normalisasi, nilai TF dibandingkan terhadap nilai IDF (pada umumnya berbentuk skala logaritma). Hasil akhirnya berupa matriks term-document *X*, dimana kolom-kolomnya berisi nilai TF-IDF untuk setiap dokumen. Oleh karena itu, skema TF-IDF mengurangi ukuran panjang dokumen yang bervariasi menjadi dokumen dengan ukuran yang tetap.

Rancangan arsitektur untuk melakukan ekstraksi fitur dengan TF-IDF dapat dilihat pada Gambar 3.

Perhitungan untuk mencari nilai TF-IDF menggunakan rumus 1 dan 2 berikut (Manning et al., 2008).

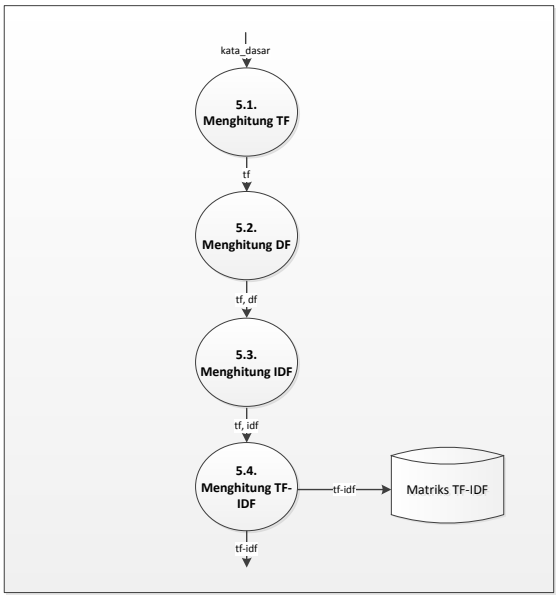
$$tf - idf_{t,d} = tf_{t,d} \; x \; idf_t \qquad (1)$$

$$idf_t = \log \frac{N}{df_t} \qquad (2)$$

III. HASIL DAN PEMBAHASAN

Pra Implementasi

Penelitian ini menggunakan dokumen berupa berita-berita yang diperoleh melalui media digital Detikcom. Berita yang dikumpulkan berasal dari beberapa kategori dan sub kategori seperti pada Tabel 1. Setiap berita disimpan ke dalam bentuk file teks (*.txt) dengan menggunakan aplikasi Notepad.



Gambar 3.Arsitektur TF-IDF

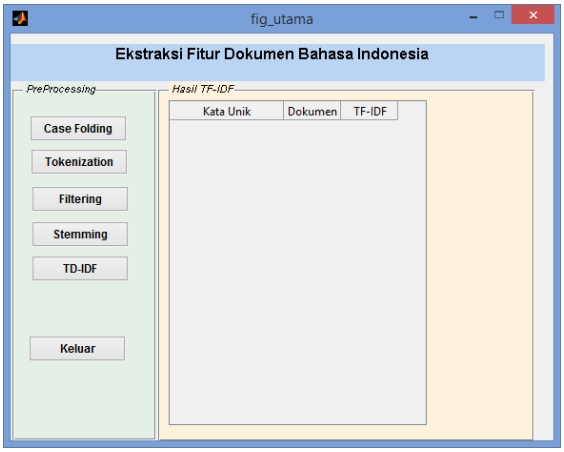
Tabel 1.Kategori Data Masukan

Kategori	Sub Kategori
Finance	Industri, Jasa Keuangan, Kebijakan Moneter, Peluang Usaha, Properti
Hiburan	*Fashion and Style*, *Movie*, *Music*, Selebriti Internasional, Selebriti Lokal
News	Kriminalitas, Pemerintahan, Pertahanan dan Keamanan, Politik, Sosial Budaya
Olahraga	Angkat Besi, Balap Sepeda, Bulu Tangkis, MotoGP, Sepakbola
Teknologi	Fotografi, Gagdet, Games, Media Sosial, OS dan Software

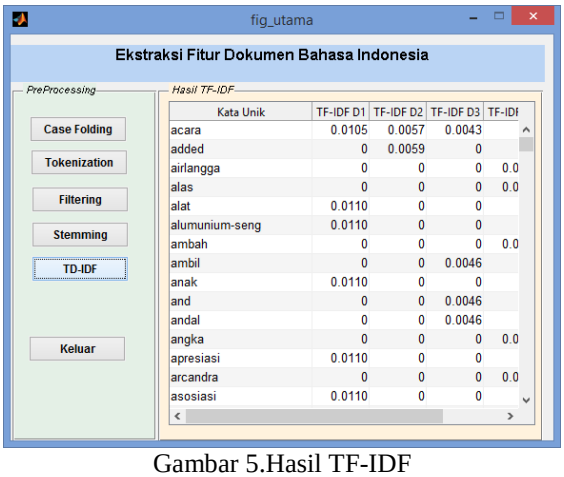
Implementasi

Arsitektur penelitian ini diimplementasikan menggunakan perangkat lunak Matlab seperti pada Gambar 4.

Hasil penelitian ini berupa daftar kata yang terdapat dalam dokumen beserta nilai TF-IDF nya, seperti ditampilkan pada Gambar 5.



Gambar 4.Desain Tampilan Penelitian



Gambar 5.Hasil TF-IDF

Pembahasan

Hasil penelitian yang ditampilkan pada Gambar 5 diperoleh melalui beberapa langkah seperti berikut.

Langkah 1. *Case Folding*

Case folding dilakukan dengan mengimplementasikan kode program seperti berikut.

```
%proses tokenization
%menghilangkan tanda baca
data=regexprep(data,'[<>.,?!"+=\/:[];_12345678910@#$%£^&*(
){}|]','');
data=strtrim(data);
%menghilangkan cell yang kosong
data(strcmp('',data))=[];
data_token=sort(data);
```

Langkah 2. *Filtering*

Filtering dilakukan dengan mengimplementasikan kode program seperti berikut.

```
for j=1:size(data_token,1)
        temp=data_token{j};
        k=1;
        while (k<=m_hubung)
           cek=strcmp(temp,kamushubung{k});
           if cek==1
              k=m_hubung+1;
              ya=1;
           else
              k=k+1;
              ya=0;
           end
        end
        if ya==0
          hasil_filter{z}=temp;
          z=z+1;
        end
end
```

Langkah 3. *Stemming*

Stemming dilakukan dengan mengimplementasikan kode program seperti berikut.

```
for j=1:size(data_filter,1)
        temp=data_filter{j};
        if isempty(regexp(temp,'[a-z]\-[a-z]','once'))==0
           [tok,rem]=strtok(temp,'-');
           [tok2,rem]=strtok(rem,'-');
           kata1=tok;
           kata2=tok2;
           %panggil stemming
           katastem1=stemming(kata1,katadasar,m_dasar);
           %katastem1=perl('stem.pl',kata1);
           %katastem2=perl('stem.pl',kata2);
           katastem2=stemming(kata2,katadasar,m_dasar);
           if strcmp(katastem1,katastem2)==1
                 hasilstem{z}=katastem1;
                 z=z+1;
           else
                 hasilstem{z}=temp;
                 z=z+1;
           end
        elseif isempty(regexp(temp,'[a-z]\-[a-z]','once'))==1
           %panggil stemming
           katastem=stemming(temp,katadasar,m_dasar);
           %katastem=perl('stem.pl',temp);
           if strcmp(katastem,'')==0
              hasilstem{z}=katastem;
              z=z+1;
           end
        end
    end
```

Langkah 4. *TF-IDF*

TF-IDF dilakukan dengan mengimplementasikan kode program seperti berikut.

```
%menghitung jumlah kemunculan kata
z=1;
idy=0;
for i=1:N
   for j=1:numel(data_gabung)
      idx=ismember(data_stem{i},data_gabung(j));
      out(j)=sum(idx,1);
      wordcount{z,j}=out(j);
   end
   z=z+1;
end
%menghitung tf
idy=0;
for i=1:N
   max=size(wordcount,2);
   for j=1:max
      idy=idy+wordcount{i,j};
   end
   jum{i,1}=idy;
   for j=1:max
      tf{i,j}=wordcount{i,j}/jum{i,1};
   end
end
%menghitung df
maks_m=size(data_gabung,1);
for i=1:maks_m
   jum=0;
   for j=1:N
     cek=ismember(data_gabung{i},data_stem{j});
     if isequal(cek,1)
        jum=jum+1;
     end
   end
   idf{i}=jum*log(N/jum);
end
%menghitung tf idf
for i=1:size(tf,1)
   for j=1:size(idf,2)
      hasil=tf{i,j}*idf{1,j};
      tfidf{i,j}=hasil;
   end
end
```

Hasil TF-IDF yang ditampilkan pada Gambar 5 menunjukkan:

1. Kolom Kata Unik berisi kata-kata hasil proses stemming.
2. Kolom TF-IDF D1 berisi nilai TF-IDF dari setiap kata unik yang ada di dokumen D1. Nilai fitur bernilai 0 menunjukkan bahwa kata tersebut tidak terdapat dalam dokumen D1. Nilai fitur yang dimiliki

setiap kata unik berbeda-beda pada setiap dokumen D_i karena dipengaruhi oleh jumlah kemunculan kata tersebut dalam dokumen.

3. Kolom TF-IDF D2 dan seterusnya memiliki penjelasan yang sama dengan kolom TF-IDF D1.

IV. SIMPULAN

Ekstraksi fitur pada pengolahan dokumen berbahasa Indonesia pada penelitian ini telah berhasil dilakukan dengan mengimplementasikan metode TF-IDF. Implementasi dilakukan dengan membuat aplikasi menggunakan perangkat lunak Matlab. Dokumen untuk aplikasi berupa berita yang diperoleh melalui media digital Detikcom dan disimpan dalam bentuk file teks. Proses pengolahan dokumen melibatkan *case folding*, *tokenization*, *filtering*, *stemming* dan ekstraksi fitur. Hasil ekstraksi fitur berupa matriks yang berisi urutan kata-kata unik dari seluruh dokumen dan nilai-nilai fitur TF-IDF dari setiap kata pada seluruh dokumen.

Pada penelitian berikutnya, tahapan pengolahan dokumen berbahasa Indonesia ini akan dilanjutkan dengan tahap klustering untuk mengelompokan dokumen berdasarkan nilai-nilai fitur TF-IDF yang diperoleh dari proses ekstraksi fitur.

DAFTAR PUSTAKA

[1] BLEI, D. M., NG, A. Y. & JORDAN, M. I., "Latent Dirichlet Allocation", *Journal of Machine Learning Research*, 3**,** 993-1022, 2003.

[2] CECI, M., LOGLISCI, C. & MACCHIA, L., "Ranking Sentences for Keyphrase Extraction: A Relational Data Mining Approach", *Procedia Computer Science, Elsevier*, 38, 2014.

[3] CHALI, Y., HASAN, S. A. & MOJAHID, M., "A reinforcement learning formulation to the complex question answering problem", *Information Processing and Management*, 51**,** 252-272, 2015.

[4] CORRÊA, G. N., MARCACINI, R. M., HRUSCHKA, E. R. & REZENDE, S. O., "Interactive textual feature selection for consensus clustering", *Pattern Recognition Letters*, 52, 2014.

[5] HADDI, E., LIU, X. & SHI, Y., "The Role of Text Pre-processing in Sentiment Analysis", *Procedia Computer Science, Elsevier*, 17, 2013.

[6] HAI, Z., CHANG, K., KIM, J.-J. & YANG, C. C., "Identifying Features in Opinion Mining via Intrinsic and Extrinsic Domain Relevance", *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, 26, 2014.

[7] HUANG, S., CHEN, Z., YU, Y. & MA, W.-Y., "Multitype Features Coselection for Web Document Clustering", *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, 18, 2006.

[8] LI, Y., ALGARNI, A., ALBATHAN, M., SHEN, Y. & BIJAKSANA, M. A., "Relevance Feature Discovery for Text Mining", *JOURNAL OF LATEX CLASS FILES*, 6, 2007.

[9] LIU, F., LIU, F. & LIU, Y., "A Supervised Framework for Keyword Extraction From Meeting Transcripts", *IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*, 19, 2011.

[10] LIU, K., XU, L. & ZHAO, J., "Co-extracting Opinion Targets and Opinion Words from Online Reviews Based on the Word Alignment Model", *JOURNAL OF LATEX CLASS FILES*, 6, 2007.

[11] LIZHEN, L., WEI, S., HANSHI, W., CHUCHU, L. & JINGLI, L., "A Novel Feature-based Method for Sentiment Analysis of Chinese Product Reviews", *China Communications*, 2014.

[12] MALANDRAKIS, N., POTAMIANOS, A., IOSIF, E. & NARAYANAN, S., "Distributional Semantic Models for Affective Text Analysis", *IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*, 21, 2013.

[13] MANNING, C. D., RAGHAVAN, P. & SCHÜTZE, H., "Introduction to Information Retrieval 1st Edition", Cambridge University Press, 2008.

[14] NOH, H., JO, Y. & LEE, S., "Keyword selection and processing strategy for applying text mining to patent analysis", *Expert Systems with Applications*, 2015.

[15] TALA, F. Z. 2003. *A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia.*

[16] TUTKAN, M., GANIZ, M. C. & AKYOKU, S., "Helmholtz principle based supervised and unsupervised feature selection methods for text mining", *Information Processing and Management*, 2016.

[17] ZHAO, Z., HE, X., ZHANG, L., NG, W. & ZHUANG, Y., "Graph Regularized Feature Selection with Data Reconstruction", 2015.