KINETIK, Vol. 2, No. 2, Mei 2017, Hal. 131-140
ISSN : 2503-2259
E-ISSN : 2503-226

Analisis Tema Skripsi Mahasiswa Menggunakan Document Clustering Dengan Algoritma LINGO

Dyah Mustikasari
Universitas Muhammadiyah Ponorogo
dyah.mustikasari@gmail.com

Abstrak

Dalam dunia teknologi informasi, banyak sekali topik yang dapat diangkat menjadi tema skripsi. Sayangnya, tidak jarang mahasiswa mengambil tema skripsi yang sama dengan tema skripsi mahasiswa angkatan sebelumnya. Karena memiliki kesamaan tema, akhirnya muncullah judul-judul skripsi yang hampir mirip. Hal ini tentu membuat penelitian di kalangan mahasiswa tidak dapat berkembang dengan cepat. Seharusnya tema yang sudah sering diangkat diganti dengan tema-tema baru yang sesuai dengan perkembangan jaman. Penelitian ini bertujuan untuk menganalisis tema-tema skripsi yang sering digunakan mahasiswa pada rentang waktu tertentu. Tema skripsi dapat dianalisis dari judul dan abstraknya. Karena data yang digunakan pada penelitian ini adalah data tekstual, maka analisis dilakukan dengan metode pada text mining, yakni document clustering menggunakan algoritma clustering LINGO. Document clustering dengan algoritma LINGO diaplikasikan menggunakan perangkat lunak Carrot2 Workbench. Hasil penelitian ini berupa klaster-klaster yang beranggotakan dokumen masukan. Setiap kelompok memiliki label. Label mencerminkan kesamaan anggota pada kelompok tersebut. Dari label klaster ini dapat dianalisis tema skripsi.

Kata kunci: *Skripsi, Clustering, LINGO*

Abstract

In information technology field, there were many topics that could be selected as a theme of thesis. Unfortunately, it was often for student to take same topics as the previous generation student's. Having a similar theme, it caused similar thesis titles. This made the research among college students could not develop quickly. Supposedly, the theme that has often been selected by students should be replaced with new themes deal with new trend. This study aimed to analyze thesis themes frequently used by student. Thesis theme could be analyzed from the title and abstract. Because the data used in this study is textual data, then analyzing were performed on text mining methods, the document clustering, using a clustering algorithm LINGO. Document clustering was applied using software Carrot2 Workbench. The results of this study were clusters consisting of the input documents. Each group has a label. The label reflects member similarity in a group. From this cluster label could be analyzed the thesis theme.

Keywords: *Thesis, Clustering, LINGO*

1. Pendahuluan

Di Indonesia, skripsi adalah penelitian mahasiswa di akhir masa perkuliahan strata satu. Sebagian besar perguruan tinggi menjadikan skripsi sebagai syarat kelulusan. Proses penyusunan skripsi tidak hanya melibatkan mahasiswa, tapi juga dosen. Dosen berperan sebagai pembimbing yang harus mengarahkan mahasiswa untuk menggunakan metode penelitian yang benar. Oleh karena itu, dapat dikatakan, skripsi juga merupakan bagian dari penelitian dosen. Dalam dunia teknologi informasi, banyak sekali topik yang dapat diangkat menjadi tema skripsi. Sayangnya, tidak jarang mahasiswa mengambil tema skripsi yang sama dengan tema skripsi mahasiswa angkatan sebelumnya. Karena memiliki kesamaan tema, akhirnya muncullah judul-judul skripsi yang hampir mirip. Hal ini tentu membuat penelitian di kalangan mahasiswa tidak dapat berkembang dengan cepat.

Tema skripsi yang sudah sering diangkat sebaiknya diganti dengan tema-tema baru yang lebih kekinian agar tidak tertinggal oleh perkembangan teknologi informasi yang sangat cepat.

Mustikasari, D. (2017). Analisis Tema Skripsi Mahasiswa Menggunakan Document Clustering Dengan Algoritma Lingo. Kinetik, 2(2), 131-140. Doi:Http://Dx.Doi.Org/10.22219/Kinetik.V2i2.180
Makalah dikirim 25 Januari 2017; Revisi 22 Maret 2017; Diterima 6 April 2017

Seharusnya tema skripsi dapat mengikuti teknologi informasi *trend* terkini. Oleh karena itu diperlukan kajian mengenai tema-tema apa saja yang sering diangkat dalam skripsi mahasiswa. Penelitian ini bertujuan untuk menganalisis tema-tema apa yang sering digunakan mahasiswa dalam skripsi pada rentang waktu tertentu. Analisis ini dilakukan dengan menggunakan *text mining*.

Text mining adalah cabang dari bidang *data mining*. *Text mining* memiliki kekhasan tersendiri karena menggunakan teks sebagai datanya. *Text mining* biasa digunakan untuk memperoleh informasi yang tidak diketahui sebelumnya, dari sumber tertulis, secara otomatis. Informasi yang digali ini akan menjadi suatu fakta baru yang dapat diteliti lebih lanjut. *Text mining* berbeda dengan *web search*. *Web search* bertujuan untuk menemukan kembali informasi yang telah ditulis oleh seseorang atau telah ada sebelumnya, sedangkan *text mining* bertujuan untuk menggali informasi yang belum diketahui sebelumnya dari sebuah sumber tertulis [1].

Text mining memiliki dua metode dalam praktiknya, yaitu klasifikasi dokumen (*document classification*) dan klasterisasi dokumen (*document clustering*) [2]. Pada klasifikasi, dokumen dikelompokkan ke dalam kategori yang sebelumnya sudah ditentukan terlebih dahulu sehingga metode ini memerlukan sebuah data latih (*training set*). Sedangkan pada *clustering*, dokumen dikelompokkan menurut kesamaannya antara satu dengan yang lain. Penelitian ini akan menggunakan metode *clustering* agar data dapat dikelompokkan sesuai kesamaannya. Proses pengelompokan menjadi lebih alami karena tidak dikendalikan oleh kategori yang sudah ditentukan sebelumnya sebagaimana pada metode klasifikasi dokumen.

Algoritma yang dikembangkan untuk *text mining* sudah cukup banyak. Beberapa algoritma pada *text mining* mengadopsi algoritma bidang *data mining*. Namun, selama satu dekade terakhir, peneliti mulai mengembangkan algoritma khusus untuk *text mining* dengan basis frasa (kata/kumpulan kata). Hal ini mengingat *text mining* menggunakan teks sebagai data. Untuk metode *document clustering*, beberapa algoritma yang telah dikembangkan adalah Fuzzy C-Means, AHC (Agglomerative Hierarchical *Cluster*ing), Neural Network based *Cluster*ing [3], Bisecting K-Means, STC (Suffix Tree *Cluster*ing) [4], dan LINGO (Label Induction Grouping) [5]. Algoritma *text mining* yang berbasis frasa di antaranya adalah STC dan LINGO.

Penelitian ini menggunakan LINGO sebagai algoritma *clustering* karena berbasis frasa yang memang ditujukan untuk bidang *text mining*. Algoritma LINGO juga lebih lengkap karena selain mempertimbangkan frasa juga menggunakan SVD (*Singular Value Decomposition*) untuk menyaring frasa yang memiliki peran paling besar dalam sebuah dokumen. LINGO merupakan algoritma yang cukup baru. LINGO dikembangkan Osinki dalam penelitian doktoralnya. Akan tetapi, LINGO terbukti memberikan hasil cukup yang baik.

Penelitian [6] menggunakan LINGO untuk mengelompokkan data dari ODP (Open Directory Project). ODP adalah direktori yang dikumpulkan secara manual (oleh orang) dan sumbernya berasal dari internet. Bentuknya menyerupai pohon. Cabangnya dinamakan kategori, yang menggambarkan sebuah topik. Kategori ini memiliki beberapa tautan ke sumber internet yang berhubungan dengan topik kategori tersebut. Setiap tautan disertai dengan deskripsi singkat dalam 25-30 kata. Deskripsi inilah yang dijadikan penggalan (*snippet*) dokumen pada penelitian itu. Penggalan (*snippet*) inilah yang dijadikan sebagai data masukan untuk dikelompokkan (*clustering*) menggunakan algoritma LINGO. Hasil *clustering* menunjukkan bahwa 80%-90% label *cluster* yang dihasilkan dari proses *clustering* dapat dipahami dan 70-80% penggalan dokumen yang berada dalam *cluster* cocok dengan topik klasternya [7]. Sayangnya, algoritma ini masih sering menghasilkan klaster “Other Topis”.

LINGO juga digunakan [8] untuk mengelompokkan dokumen berbahasa Cina. Mereka mengganti SVD (*Singular Value Decomposition*) dengan NMF (non-negatif metrix factorization). Penelitian ini ditujukan untuk menyesuaikan algoritma LINGO untuk mengelompokkan (*clustering*) bahasa Cina. Teks bahasa Cina terbentuk dari kalimat. Tidak ada batasan yang jelas antarkata. Oleh karena itu sangat penting untuk melakukan segmentasi pada teks bahasa Cina sebelum dilakukan *clustering*. Penelitian ini [8] menambahkan proses segmentasi kalimat. Dalam percobaannya, [8] membandingkan dengan dua algoritma *clustering* lain yang pernah digunakan untuk bahasa Cina dan sudah teruji, yaitu CTCAUSL (Chinese Text *Cluster*ing Algorithm Using Semantic List) dan K-Means. Hasilnya menunjukkan C-Lingo lebih baik dibanding dengan CTCAUSL dan K-Means. Teknik NMF dinyatakan lebih tepat digunakan untuk bahasa Cina [8].

LINGO juga diteliti untuk mengelompokkan data teks dalam bahasa Marathi [9]. Marathi adalah salah satu bahasa di India. *Clustering* dokumen ini berdasarkan minat pengguna. Minat pengguna diketahui dari riwayat pencarian pengguna. Dalam menentukan tingkat minat

pengguna terhadap konten web, sistem juga dirancang untuk dapat menghitung waktu pengguna membuka suatu konten dan menutupnya. Jika suatu konten dibuka lebih lama maka konten tersebut lebih diminati oleh pengguna, sehingga konten ini mendapat skor lebih tinggi [9]. Sistem menghitung waktu baca (*reading session time*) sebuah jenis konten, bukan banyaknya konten yang dibuka [9]. Data set yang digunakan adalah berita yang diperoleh dari beberapa situs berita berbahasa Marathi. Hasil penelitian menunjukkan akurasinya mencapai angka 91,10% [9].

Algoritma LINGO dibuat untuk meningkatkan kualitas deskripsi *cluster*. Kebanyakan algoritma *clustering* gagal dalam memberikan label yang dapat dengan mudah dibaca dan dipahami oleh penggunanya [10]. Algoritma *clustering* pada umumnya mengelompokkan dokumen berdasarkan konten, dan berdasarkan konten itulah, baru kemudian membuat nama labelnya. Algoritma LINGO membalik proses ini. LINGO membuat label terlebih dahulu dan memastikannya bisa dipahami baru kemudian memasukkan dokumen yang relevan ke dalam *cluster* itu. LINGO diuji secara empiris oleh tujuh pengguna dan empat hasil pencarian, dua berbahasa Polandia dan dua berbahasa Inggris [7]. Pengguna diminta untuk menetapkan mana saja label *cluster* yang dapat dipahami dan dokumen yang memang cocok berada dalam *cluster* itu.

LINGO adalah algoritma *clustering* yang menerapkan lima fase dalam *clustering*, yang meliputi preprocessing, ekstraksi fitur, induksi label klaster, penentuan anggota klaster, dan pembentukan klaster akhir. Keunikan LINGO adalah menerapkan pemunculan label klaster terlebih dahulu baru kemudian menentukan anggota klasternya. Label klaster didapat dari ekstraksi fitur. Lebih lanjut, penggalian informasi diperoleh dari label klaster yang terbentuk. Algoritma LINGO telah diaplikasikan dalam sebuah perangkat lunak bernama Carrot2.

2. Metode Penelitian

2.1 Pra-proses

Data yang digunakan dalam penelitian ini adalah skripsi mahasiswa Teknik Informatika sebanyak 50 sampel data (50 skripsi) dalam rentang tahun akademik 2015/2016. Teks yang diambil sebagai masukan (*input*) *document clustering* adalah judul dan abstrak skripsi. Keduanya dinilai dapat mewakili seluruh isi sebuah skripsi.

Tahapan pra-proses yang dimaksud dalam penelitian ini adalah tokenisasi, penghapusan *stopword*, dan stemming. Tokenisasi adalah memecah teks menjadi kata tunggal. Tokenisasi juga melakukan penyeragaman tulisan menjadi huruf kecil semua. Selain itu, tanda baca dan angka juga dihilangkan. Setelah itu, *stopword* dihapus dari dokumen. Penghilangan *stopword* ini menggunakan daftar stoplist dari Tala [11]. Terakhir, *stemming* dilakukan untuk mengubah kata berimbuhan menjadi kata dasar. Pra-proses ini dilakukan dengan program terpisah dari proses *clustering*, menggunakan penyusunan kode program sendiri.

2.2 Tokenisasi

Tokenisasi adalah langkah pertama dalam pra-proses. Langkah ini berguna untuk memecah sebuah teks utuh menjadi kata-kata tunggal. Kata hasil dari pemecahan ini biasa disebut sebagai token. Token dapat berupa sebuah kata atau sebuah frasa, misalnya “buku cerita” dapat dipisah per kata menjadi “buku” dan “cerita” atau menjadi sebuah frasa “buku cerita”. Hal ini bergantung pada aturan yang digunakan. Tokenisasi juga menyeragamkan penulisan menggunakan huruf kecil seluruhnya atau huruf kapital. Penyeragaman ini agar terhindar dari kesalahan pada proses selanjutnya jika proses tersebut bersifat *case-sensitive*.

2.3 Penghapusan *Stopword*

Stopword adalah proses menghilangkan kata-kata yang tidak memiliki arti khusus terkait konteks dalam teks/dokumen. Kata-kata ini jika dihilangkan tidak mempengaruhi makna atau inti dari teks itu. Hal ini dilakukan untuk memperbesar akurasi dari pembobotan *term* nantinya. Dalam bahasa Indonesia, kata-kata yang termasuk *stopword* misalnya, “dan”, “yang”, “kemudian”, “itu”, dan sebagainya. Proses ini memerlukan kamus *stopwords* atau daftar *stopword*. Daftar *stopwords* untuk bahasa Indonesia telah banyak tersedia baik di internet maupun di jurnal-jurnal. Dalam penelitian ini, daftar *stopwords* yang digunakan merujuk pada penelitian Tala [11].

2.4 *Stemming*

Setelah kata-kata tertentu disaring menggunakan daftar *stopword*, langkah berikutnya adalah s*temming. Stemming* adalah menemukan kata dasar dari kata berimbuhan. Sebagai

contoh, kata dalam bahasa Indonesia “melakukan”, “dilakukan”, dan “berlaku” diubah bentuk dasarnya yaitu “laku”. *Stemming* berguna untuk mengurangi tempat penyimpanan istilah dan memperluas arti dari sebuah istilah.

Setiap bahasa memiliki aturan tertentu dalam pembentukan kata sehingga algoritma stemming untuk setiap bahasa bisa berbeda-beda. Secara umum, susunan kata berimbuhan pada bahasa Indonesia sebagaimana pada Persamaan 1.

Awalan 1 + Awalan 2 + Kata Dasar + Akhiran 3 + Akhiran 2 + Akhiran 1 (1)

Oleh karena itu, untuk memperoleh kata dasar, langkah yang dibutuhkan adalah menghilangkan awalan dan akhiran. Proses *stemming* menentukan kualitas output proses *clustering*. Semakin akurat kata dasar hasil *stemming*, semakin baik keluarannya.

Algoritma yang dikembangkan untuk mencari kata dasar dari kata berimbuhan sudah cukup banyak, diantaranya adalah algoritma Arifin-Setiono, Algoritma Nazief-Andriani, Algoritma Idris-Mustofa, Algoritma Ahmad, Yussof, dan Sembok. Penelitian ini memakai algoritma Nazief-Andriani.

2.5 *Clustering*

Keluaran dari pra-proses menjadi masukan pada langkah *clustering*. *Clustering* termasuk *unsupervised document classification*. Pada klasifikasi menggunakan *clustering*, pengguna tidak memerlukan data latih terlebih dahulu. Hal ini memudahkan bagi pengguna yang belum mempunyai pengetahuan sama sekali mengenai topik atau tema dari data yang akan diklasifikasikan. Data akan terkelompok berdasarkan kedekatan atau kemiripannya (*similarity*). *Clustering* banyak dimanfaatkan untuk menganalisis pola, segmentasi citra, segmentasi pasar, pemetaan wilayah, manajemen pemasaran, membantu membuat keputusan serta *machine learning* dan sebagainya. *Clustering* yang baik harus mampu mengelompokkan data-data yang mirip dan memisahkan data-data yang berbeda (kemiripannya sedikit). Kualitas suatu sistem *clustering* ditentukan oleh seberapa baik sistem dapat menentukan kemiripan antar datanya. Sebuah klaster adalah sekumpulan data yang memiliki kesamaan tertentu dan mempunyai perbedaan dengan klaster lainnya.

Pada penelitian ini proses *clustering* dilakukan dengan bantuan perangkat Carrot2 Workbench. Carrot2 Workbench semisal perangkat *text mining* WEKA, Rapid Miner, Clusty, dan sebagainya. Sebenarnya, Carrot2 sendiri adalah sebuah pustaka dan satu set aplikasi pendukung yang dapat digunakan untuk membangun mesin *clustering*. Carrot2 dapat mengelompokkan hasil pencarian dari mesin pencari semacam Google, Bing, dan PubMed. Selain itu, Carrot2 juga menerima masukan dokumen/teks dari direktori internal dalam format xml. Carrot bisa diintegrasikan dengan program yang ditulis dengan bahasa pemrograman lain. Carrot2 dikembangkan dengan bahasa pemrograman Java. Carrot2 menyediakan API dan JAR untuk bisa diintegrasikan dengan kode program berbasis Java. Untuk selain bahasa Java, Carrot2 menyediakan Carrot2 Document *Cluster*ing Server (DCS) dan dapat dipanggil menggunakan protokol REST.

Workbench disediakan oleh pengembang Carrot2 agar pengguna dapat langsung menggunakan untuk melakukan *clustering*. Carrot2 Workbench menyediakan tiga algoritma, yaitu, Suffix Tree *Cluster*ing, Bisecting K-Means, dan LINGO. Perangkat ini menerima beberapa masukan, diantaranya hasil pencarian dari mesin pencari seperti Google, Bing, dan Pubmed. Di samping itu, perangkat ini juga menerima masukan dari direktori internal dalam bentuk berkas xml dengan format yang ditunjukkan Gambar 1.

Pada penelitian ini, seluruh berkas yang telah melewati tahap pra-proses diubah menjadi bentuk xml berdasarkan susunan di atas. Selanjutnya, berkas dimasukkan dalam Carrot2 Workbench untuk dilakukan *clustering*. Algoritma yang digunakan adalah LINGO.

LINGO merupakan algoritma *clustering* teks berbasis frasa. LINGO memiliki beberapa fase dalam proses *clustering*. Fase pertama adalah *preprocessing*. Fase ini terdiri dari beberapa tahap. Pertama adalah *text filtering*. *Text filtering* adalah tahap penghapusan karakter dan *tag* yang tidak penting. Selanjutnya, LINGO mendeteksi bahasa yang digunakan oleh dokumen. LINGO dapat mendeteksi bahasa secara otomatis sehingga dapat menyesuaikan dengan daftar *stopword* yang akan digunakan. Untuk mendeteksi bahasa, LINGO menggunakan daftar *stopword* dalam basis datanya dan mencocokkannya dengan kata-kata yang ada dalam dokumen, kemudian dipilih daftar *stopword* yang paling sering muncul dalam dokumen tersebut.

Daftar *stopword* itulah yang akan menentukan bahasa dokumen atau teks itu. Misalnya kata yang sering muncul adalah *the*, *and*, *of*, *can*, *will*, maka LINGO akan memutuskan bahwa dokumen/teks input itu berbahasa Inggris. Setelah bahasa terdeteksi, langkah selanjutnya adalah *stemming*. Pada LINGO [7], stemmer yang disediakan adalah Porter stemmer untuk bahasa Inggris dan untuk bahasa Polandia.

```
<searchresult>
    <query>query (optional)</query>
        <document>
                <title>Document 1 Title</title>
                <snippet>Document 1 Content.</snippet>
                <url>http://document.url/1</url>
        </document>
        <document>
                <title>Document 2 Title</title>
                <snippet>Document 2 Content.</snippet>
                <url>http://document.url/2</url>
        </document>
        <document>
                <title>Document 3 Title</title>
                <snippet>Document 3 Content.</snippet>
                <url>http://document.url/3</url>
        </document>
</searchresult>
```

Gambar 1. Format Berkas XML Sesuai Ketentuan Carrort2 Workbench

Fase kedua adalah pemilihan fitur (*feature extraction*). Tujuan dari tahap ini adalah untuk mendaftar semua frasa dan kata tunggal. Frasa dan kata tunggal ini nantinya akan menjadi kandidat label klaster. Kata atau frasa yang potensial untuk dijadikan kandidat label klaster memiliki syarat, yaitu frekuensi frasa minimal n kali kemunculan, tidak melewati batas kalimat/batas frasa, tidak diakhiri atau diawali dengan *stopword* (*stopword* di tengah tidak dihilangkan), dan merupakan frasa lengkap [12] .

LINGO menggunakan algoritma penemuan frasa. Algoritma ini untuk memastikan bahwa sebuah frasa adalah lengkap. Frasa lengkap adalah frasa yang mengandung informasi lengkap, sebaliknya frasa parsial adalah frasa yang katanya tidak lengkap tetapi mengandung informasi yang hampir sama. Cara kerja algoritma ini adalah mengidentifikasi frasa-kiri dan frasa-kanan lalu menggabungkannya menjadi frasa komplit. Frasa-kiri contohnya “buku cerita”, sedangkan frasa-kanannya misalnya “cerita anak”, sehingga frasa komplitnya adalah “buku cerita anak”. Pada tahap akhir pemilihan fitur, term dan frasa yang melebihi batas frekuensi (*frequency threshold*) dipilih untuk tahap selanjutnya. Dalam penelitiannya, batas frekuensi term yang efektif berkisar antara 2-5 [5] .

Fase ketiga merupakan pembentukan label klaster. Label klaster ditentukan terlebih dahulu baru kemudian pada fase selanjutnya anggota tiap klaster ditentukan. Langkah awal fase ketiga ini adalah membuat matriks term-dokumen. Term ini merupakan frasa dan kata tunggal yang telah ditemukan pada fase sebelumnya. Matriks term-dokumen ini disusun dengan TF-IDF. Matriks term-dokumen diurai dengan SVD untuk menemukan konsep abstrak. SVD akan mengurai matriks menjadi tiga matriks U, S, dan V^T [5].

Dari tiga matriks U, S dan V^T, yang paling penting adalah matriks U. Namun tidak semua vektor kolom pada matriks U dipakai. Hanya vektor basis k pertama yang digunakan (kolom k pertama), misalkan nilai k=2, maka hanya dua kolom pertama dari matriks U yang digunakan. Matriks baru ini disebut dengan matriks U_k. Nilai k ditentukan berdasarkan parameter batas kandidat label (*candidate* label *threshold*) [5]. Secara *default*, nilai parameter ini adalah 0,7-0,9, tetapi nilai ini dapat diganti oleh pengguna [5]. Langkah berikutnya adalah menggunakan TF-IDF untuk membuat matriks term-frasa/term. Matriks term-frasa/term disusun dari term terhadap term dan frasa yang ditemukan pada fase pertama. Matriks term-frasa/term kemudian dikalikan dengan matriks U_k. Nilai elemen vektor yang melampaui batas kandidat label di matriks ini menunjukkan frasa/kata yang menjadi kandidat label klaster.

Kandidat label klaster yang terbentuk bisa saja memiliki kemiripan satu dengan yang lain sehingga salah satunya perlu dihilangkan. Tahap ini dilakukan dengan menghitung kemiripan seluruh pasangan kandidat label. Dari satu grup kandidat label yang nilai *cosine similarity*-nya melebihi batas-kemiripan, hanya skor paling tinggilah yang diambil. Batas-kemiripan (*label similarity threshold*) berkisar antara 0,2-0,5 [5].

Fase keempat adalah penentuan konten/anggota dari tiap label klaster. Tahap ini menggunakan VSM (*Vector Space Model*) untuk menentukan dokumen mana saja yang akan dimasukkan ke dalam suatu klaster. VSM digunakan sebagaimana pada sistem temu-balik (*infromation retrieval*). Bedanya adalah, pada LINGO, dokumen (*snippet*) dicocokkan dengan setiap label klaster yang dihasilkan pada fase sebelumnya. Langkah pertama adalah membuat matriks term-label klaster dari semua label klaster yang telah terbentuk (matriksnya seperti matriks term-dokumen). Tranpose matriks ini lalu dikalikan dengan matriks term-dokumen yang dibuat di fase ketiga. Baris dari matriks hasil perkalian tersebut menunjukkan klaster/grup dan kolomnya mewakili dokumen anggota setiap klaster/grup. Sebuah dokumen akan dimasukkan ke dalam suatu *cluster* jika nilai elemen matriksnya melampaui nilai *threshold*, yang dalam penelitian LINGO disebutkan 0,15-0,30 [5]. Skema penentuan menggunakan *cosine similarity* memungkinkan terjadinya klaster yang tumpang-tindih (sebuah dokumen bisa masuk lebih dari satu *cluster*). Terakhir, dokumen yang tidak masuk pada *cluster* mana pun akan dimasukkan dalam label bernama ‘Other topics’.

Fase terakhir dari LINGO adalah menghitung skor klaster. Skor dihitung dengan mengalikan label skor dengan jumlah anggota. LINGO mengurutkan skor dari yang paling tinggi ke yang paling rendah dan menampilkan labelnya.

3. Hasil Penelitian dan Pembahasan

3.1 Pra-proses

Sebelum dilakukan *clustering*, data diproses dahulu melalui tahap pra-proses. Tahap awal pra-proses adalah tokenisasi. Sebelum dilakukan tokenisasi, seluruh dokumen masukan telah diubah menjadi format .txt. Dalam penelitian ini, tokenisasi dilakukan dengan program terpisah, bukan dalam *workbench*. Program ini dinamai program pra-proses. Program ini bekerja dengan cara membuka berkas dokumen kemudian menyeragamkan penulisan teks dalam huruf kecil.

Tahap selanjutnya dari pra-proses adalah penghilangan *stopword* (*stopword removing*). *Stopword* yang digunakan dalam penelitian ini adalah *stoplist* Tala [11]. *Stoplist* atau daftar *stopword* disimpan dalam basis data. Setiap kata pada tiap dokumen dibandingkan dengan kata yang tersimpan pada daftar *stopword.* Jika ada kata yang sama, program akan menghapus kata tersebut dari teks/dokumen. Hasil dari proses ini adalah teks yang hanya mengandung kata-kata inti saja tapi bukan merupakan kata dasar.

Langkah berikutnya dari pra-proses adalah *stemming*. Proses *stemming* diproses dengan program stemming. Masukan untuk program *stemming* adalah berkas *output* program pra-proses*. Stemming* adalah mencari kata dasar dari setiap kata yang terdapat pada teks hasil dari penghilangan *stopword*. Proses ini membutuhkan waktu cukup lama, karena setiap kata harus dibandingkan dengan kata dasar pada kamus. Jika tidak ada, maka akan dicari kata dasarnya dengan cara menghilangkan imbuhan. Stemming ‘Nazief dan Adriani’ memiliki beberapa aturan yang cukup detail sehingga untuk mengolah satu kata menjadi kata dasar cukup memakan waktu. Hasil dari proses *stemming* setiap berkas akan disimpan dalam direktori ‘’*stemming*’’ dengan nama berkas yang sama dan dalam bentuk txt.

Dokumen pada Gambar 2 adalah abstrak salah satu skripsi yang digunakan sebagai dokumen *input* untuk pra-proses. Dokumen tersebut masih mengandung tanda baca, *stopword*, huruf kapital, dan kata berimbuhan. Gambar 3 memperlihatkan dokumen yang sudah melalui tahap pra-proses, yaitu tokenisasi, penghilangan *stopword*, dan *stemming*. Tidak lagi terdapat tanda baca, *stopword*, huruf kapital, serta kata-kata berimbuhannya sudah menjadi kata dasar.

Dalam penelitian ini, pengelompokan dilakukan dengan mesin *clustering* Carrot2 Workbench. Carrot2 Workbench mengharuskan masukan berbentuk xml. Oleh karena itu berkas yang telah preprocessing tadi harus diubah menjadi format xml sesuai aturan Carrot2 Workbench. Untuk membuat masukan xml sesuai aturan Carrot2 Workbench, semua berkas yang telah melalui tahap pra-proses dan *stemming* dimasukkan dalam basis data pada tabel tersendiri kemudian dilakukan penguraian (*parsing*) menjadi xml dengan fungsi DOMDocument. Gambar 4

memperlihatkan sebagian data yang telah diubah menjadi format xml, yang ditampilkan dalam aplikasi Notepad++.

```
$document = new DOMDocument();
$document->formatOutput = true;
```

Selanjutnya, hasil parsing disimpan dalam berkas ‘hasil.xml’.

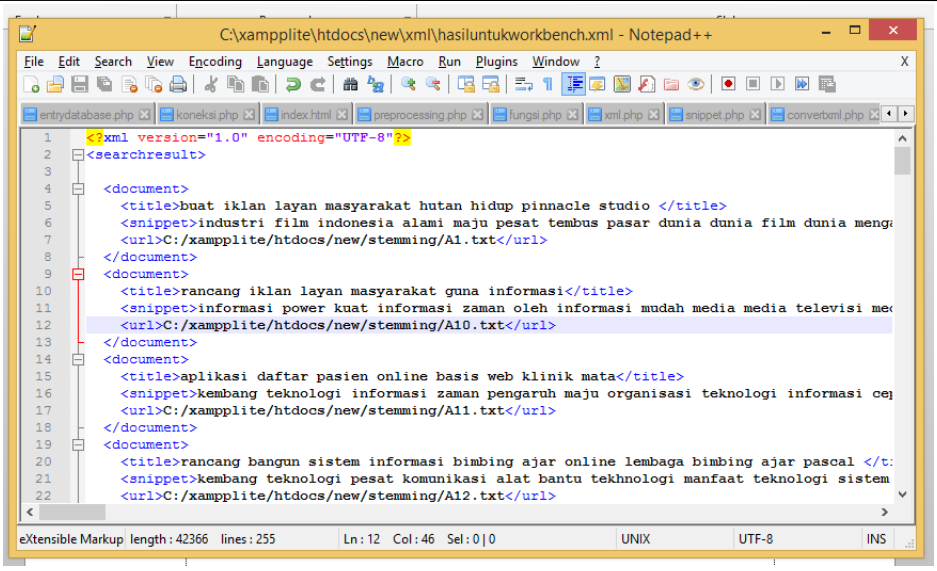
```
$document->save("hasil.xml"
```

Teknologi internet sudah terbukti merupakan salah satu media informasi yang efektif dan efisien dalam penyebaran informasi yang dapat diakses oleh siapa saja, kapan saja dan dimana saja. Teknologi internet mempunyai efek yang sangat besar pada perdagangan atau bisnis. Hanya dari rumah atau kantor, calon pembeli dapat melihat produk-produk pada layar komputer, mengakses informasinya, memesan dan membayar dengan pilihan yang tersedia. Calon pembeli dapat menghemat waktu dan biaya karena tidak perlu datang ke toko atau tempat transaksi sehingga dari tempat duduk ia dapat mengambil keputusan dengan cepat. Transaksi secara online dapat menghubungkan antara penjual dan calon pembeli secara langsung tanpa dibatasi ruang dan waktu. Itu berarti transaksi penjualan secara online mempunyai calon pembeli yang potensial dari seluruh dunia. Sanjaya Jati Mebel Ponorogo adalah satu perusahaan yang bergerak di bidang perdagangan Mebel dan melayani berbagai macam pemesanan barang olahan yang berbahan dasar kayu. Sanjaya Jati Mebel Ponorogo beralamatkan di Jl. Sukowati No. 5 Ngunut Kecamatan Babadan, Kabupaten Ponorogo. Dalam kegiatan produktifnya Sanjaya Jati Mebel Ponorogo memproduksi meja, kursi, dipan, bufet, almari dan berbagai macam furniture yang berbahan dasar kayu jati. Dikerjakan dengan oleh tenaga-tenaga yang profesional dibidangnya, dan senantiasa menjadikan kepuasan pelanggan adalah kewajiban dan motivasi dalam setiap transaksi. Sistem penjualan yang selama ini digunakan oleh Sanjaya Jati Mebel Ponorogo adalah dengan cara memasarkan sendiri produk mebel secara langsung atau harus melalui tatap muka dengan calon pembeli. Hal ini dapat menjadi masalah karena ketidakefisienan dalam proses penjualan produk tersebut. Jika hanya mengandalkan sistem penjualan tersebut, maka pendapatan perusahaan tidak mengalami peningkatan yang signifikan. Selain itu perkembangan perusahaan dinilai agak lambat. Oleh karena itu perlu dirancang suatu sistem pemesanan secara online dengan menggunakan media web atau internet dengan tujuan untuk meminimalkan waktu proses penjualan dengan tujuan dapat meningkatkan volume penjualan, sehingga pendapatan perusahaan dapat meningkat.

Gambar 2. Dokumen Masukan Untuk Pra-proses

teknologi internet bukti salah media informasi efektif efisien sebar informasi akses mana teknologi internet efek dagang bisnis rumah kantor calon beli produk layar komputer akses informasi mesan bayar pilih sedia calon beli hemat biaya toko transaksi duduk ambil putus cepat transaksi online menghubungkan jual calon beli langsung batas ruang transaksi jual online mempunyai calon beli potensial dunia sanjaya jati meubel ponorogo usaha gerak bidang dagang mebel layan mesan barang olah bahan dasar kayu sanjaya jati mebel ponorogo alamat jl sukowati no ngunut camat babad kabupaten ponorogo giat produktif sanjaya jati meubel ponorogo produksi meja kursi dipan bufet almari furniture bahan dasar kayu jati tenaga-tenaga profesional bidang senantiasa jadi puas langgan wajib motivasi transaksi sistem jual sanjaya jati mebel ponorogo pasar produk mebel langsung tatap muka calon beli tidak efisien proses jual produk andal sistem jual dapat usaha alami tingkat signifikan kembang usaha nilai lambat rancang sistem mesan online media web internet tuju minimal

Gambar 3. Dokumen Keluaran dari Pra-proses

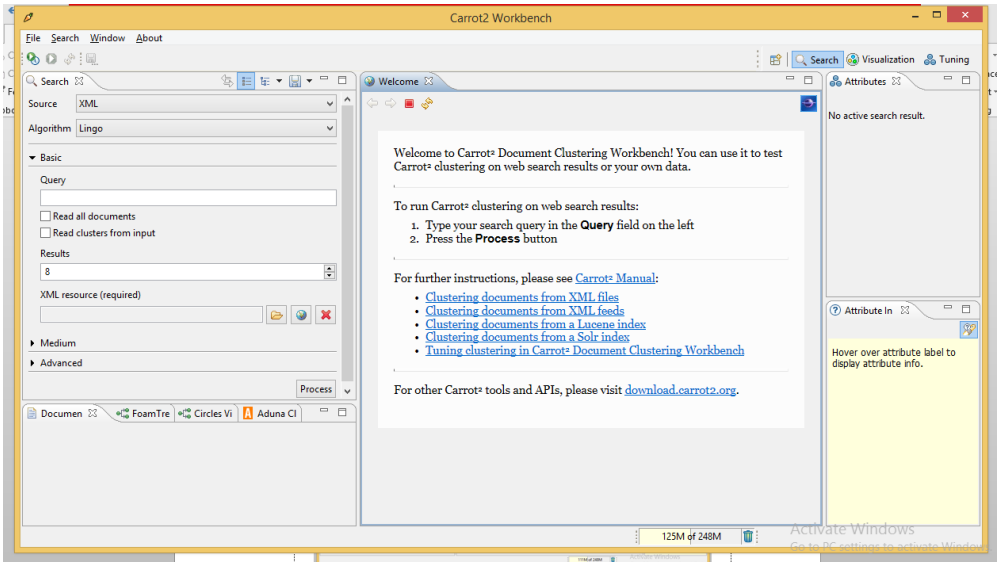


Gambar 4. Data masukan untuk Carrot2 Workbench Berbentuk XML

3.2 *Clustering*

Carrot2 Workbench mengizinkan beberapa *input* diantaranya, berkas xml, Lucene Index, dan Solr Index. Selain data masukan, Carrot2 dapat digunakan untuk mengelompokkan hasil pencarian dari Web [13]. Carrot2 menyediakan tiga algoritma untuk pengelompokan data teks, yaitu Bisecting K-Means, LINGO, dan STC. Masing-masing algoritma dapat diatur parameternya, misal nilai *frequency threshold*, skor *base-cluster*, batas kemiripan (*label similarity threshold*), dan sebagainya.

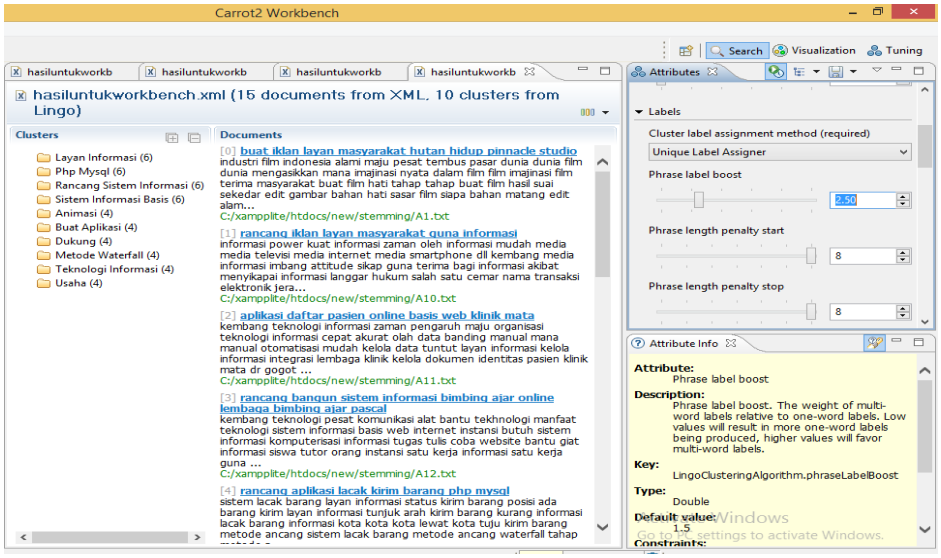
Gambar 5 merupakan tampilan awal dari Carrot2 Workbench. Pada menu source pengguna dapat memilih data masukan (teks) yang ingin dikelompokkan. Untuk masukan berupa berkas xml, pengguna harus memastikan format xml sama sebagaimana yang diminta oleh Carrot2 Workbench. Jika formatnya tidak sama maka data tidak dapat dikelompokkan (*clustering*). Karena Carrot2 Workbench dikembangkan dalam bahasa Polandia dan bahasa Inggris, maka dalam penelitian ini data perlu diolah dahulu dan disesuaikan agar dapat menjadi masukan bagi Carrot2 Workbench. Pengolahan dilakukan dengan program pra-proses dan *stemming*, sebagaimana yang telah dijelaskan sebelumnya. Sebenarnya Carrot2 Workbench telah menyediakan *preprocessor* dan *stemmer*. Namun, keduanya berlaku bukan untuk data teks berbahasa Indonesia.



Gambar 5. Tampilan Awal Carrot2 Workbench

3.3 Klaster Hasil *Clustering*

Carrot2 Workbench dapat menghasilkan klaster yang berbeda-beda tergantung pada pengaturan parameternya. Setiap data teks memiliki sifat tersendiri, misalnya bahasa, pengulangan kata, dan lain-lain. Gambar 6 memperlihatkan klaster yang dihasilkan dari pengelompokan oleh Carrot2 Workbench.



Gambar 6. Hasil Clustering Menggunakan Carrot2 Workbench

Sebagaimana yang tampak pada Gambar 6, terdapat sepuluh klaster. Masing-masing klaster diberi label yang berbeda satu dengan yang lain. Label ini merupakan kata yang sama dari anggotanya. Misalnya klaster ke-5 diberi label “Animasi”, maka semua anggota dalam klaster tersebut mengandung kata/term “Animasi”. Agar lebih jelas, label klaster yang terbentuk sebagaimana pada Gambar 6 dituliskan dalam Tabel 1.

Tabel 1. Klaster Hasil Clustering Carrot2 Workbench

No.	Label klaster	Jumlah Anggota Klaster
0	Layan Informasi	6
1	Php Mysql (	6
2	Rancang Sistem Informasi	6
3	Sistem Informasi Basis	6
4	Animasi	4
5	Buat Aplikasi	4
6	Dukung	4
7	Metode Waterfall	4
8	Teknologi Informasi	4
9	Usaha	4

Setiap klaster memiliki jumlah anggota yang berbeda-beda, tetapi bisa jadi ada anggota yang sama. Misalnya klaster 1 dengan label “Layan Informasi” memiliki anggota enam yaitu, data teks 0, 1, 2, 4, 12, 14. Di sisi lain, klaster 2 dengan label “PHP MySQL” memiliki anggota enam, yaitu data teks 3, 4, 7, 11, 12, 14. Terdapat dua anggota yang sama. Inilah yang dinamakan *cluster* overlap. Hasil pengelompokan pada Gambar 6 menggunakan pengaturan parameter sebagai berikut.

1. *Cluster* count base : 30
2. *Cluster* merging threshold : 0.70
3. Size-score sorting ratio : 0.00
4. Truncated label threshold : 0.8
5. Metode faktorisasi: Nonnegative Matrix Factorization ED Factory

6. Frekuensi maksimal kata dalam dokumen : 0.8
7. Pembobotan term : Linear Tf Idf Term Weigthing
8. Jumlah minimum anggota per *cluster* : 4

Hasil sebagaimana Gambar 6 bisa jadi berbeda dengan pengaturan parameter yang berbeda pula. Hal ini justru memungkinkan pengguna untuk menyesuaikan *clustering* dengan tujuan yang ingin dicapainya.

Dalam penelitian ini, tujuan *clustering* adalah untuk mengetahui tema/topik skripsi yang sering diambil oleh mahasiswa. Hasil *clustering* sebagaimana yang tampak pada Gambar 6 memperlihatkan bahwa dari 50 sampel yang diambil, tema/topik yang diangkat seputar sistem informasi dan pembuatan aplikasi. Selain itu, ada juga tema tentang animasi. Di sisi lain, ada hal yang dapat digali dari pengelompokan ini. Misalnya munculnya label "Metode Waterfall". Klaster dengan label ini memiliki empat anggota yaitu, 1, 4, 5, dan 13. Dari fakta dapat disimpulkan bahwa ternyata metode yang banyak digunakan dalam pengembangan aplikasi adalah metode Waterfall. Selain itu, dari hasil pengelompokan ini juga dapat dilihat bahwa bahasa pemrograman yang sering dipakai adalah PHP dengan basis data MySQL.

4. Kesimpulan

Kesimpulan dari penelitian ini adalah sebagai berikut:

1. Analisis tema skripsi mahasiswa dapat dilakukan dengan algoritma LINGO menggunakan Carrot2 Workbench.
2. Kelompok yang dihasilkan dari *clustering* sejumlah sepuluh klaster dari 50 sampel skripsi yang digunakan.
3. Label klaster yang dihasilkan dapat digunakan untuk menggali tema/topik skripsi yang sering digunakan oleh mahasiswa.
4. Untuk lebih memperluas analisis tentang tema skripsi, maka data yang digunakan dapat lebih banyak. Selain itu, algoritma yang digunakan juga lebih beragam.

Referensi

[1] M. Hearst, "What is text mining?," Japan Advanced Institute of Science and Technology, Pp. 1–3, 2003.
[2] M. W. Kogan, Jacob; Berry, Text Mining Applications and Theory, 1st ed. West Sussex: John Wiley & Sons, Ltd, 2010.
[3] K. Sridevi, R. Umarani, and V. Selvi, "An Analysis of Web Document Clustering Algorithms," Vol. 1, No. 6, 2011.
[4] O. E. Zamir, "Clustering Web Documents : A Phrase-Based Method for Grouping Search Engine Results," University of Washington, 1999.
[5] S. Osi and D. Weiss, "A Concept-Driven Algorithm for Results," IEEE Intell. Syst., Pp. 48–54, 2005.
[6] S. Osi, "Conceptual Clustering Using Lingo Algorithm : Evaluation on Open Directory Project Data."
[7] D. Osinski, Stanislaw; Stefanowski, Jerzy ; Weiss, "Lingo : Search Results Clustering Algorithm Based on Singular Value Decomposition," Poznan, 2005.
[8] X. Lin, Q. Zhang, and G. Wei, "The Clustering Algorithm for Chinese Texts Based on Lingo," in Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), 2011, Pp. 1187–1190.
[9] S. R. Vispute and P. M. A. Potey, "Automatic Text Categorization of Marathi Documents Using Clustering Technique," 2013.
[10] S. Kanthekar, A. Kadam, C. Kunte, and P. Kadam, "Generation," in International Conference on Circuits, Systems, Communication and Information Technology Applications (CSCITA), 2014, Pp. 294–299.
[11] F. Z. Tala, "A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia," Universiteit van Amsterdam.
[12] S. Osinski, "An algorithm for clustering of Web Search Results," Poznan University of Technology, 2003.
[13] J. Stefanowski And D. Weiss, "Carrot 2 And Language Properties In Web Search Results Clustering," Adv. Web Intell., Vol. 1, Pp. 955–955, 2003.