BAB I

PENDAHULUAN

1.1 . LATAR BELAKANG

Terlalu banyak untuk menyebut jenis bencana alam yang datang silih berganti menghampiri Indonesia. Kondisi ini membuat pemerintah harus berpikir ekstra keras untuk mengatasi dampak buruk bencana alam tersebut. Memang berbagai pihak telah berupaya sekuat tenaga untuk mengatasinya namun hasil yang didapatkan belum seperti yang diharapkan. Korban yang timbul akibat bencana alam masih saja tinggi. Belum lagi kerusakan lingkungan yang makin parah. Manusia memang tidak punya kekuatan untuk menghalau bencana alam apabila bencana itu datang. Yang bisa manusia lakukan adalah mengambil langkah penyelamatan diri dan harta benda sedini mungkin agar terhindar dari malapetaka yang berasal dari proses alamiah tersebut. Langkah penyelamatan ini bisa dilakukan dengan dukungan teknologi canggih yang mampu mendeteksi gejala alam baik itu yang bersumber dari laut maupun dari darat. Hal yang terpenting di sini adalah bagaimana meminimalkan korban sekecil mungkin atau tidak ada korban jiwa sama sekali.

Dalam konteks ini, teknologi informasi (TI) hadir memainkan peran yang cukup penting. Sebagaimana manusia, eksistensi TI tidak untuk menghalau suatu bencana alam yang datang secara tiba-tiba melainkan untuk menyampaikan informasi sebelum dan sesudah bencana alam itu terjadi. Teknologi informasi tersebut tentu tidak berdiri sendiri melainkan terintegrasi dengan berbagai

perangkat teknologi canggih lainnya yang dapat memberikan peringatan dini secara sistimatis kepada warga yang berdomisili di sekitar kawasan rawan bencana alam. Sistem ini yang kemudian dikenal dengan nama sistem peringatan dini.

Mengantisipasi bencana dalam waktu singkat dapat dilakukan dengan menerapkan sistem peringatan dini. Sistem itu bekerja dengan memanfaatkan basis data dari berbagai situs di Internet.

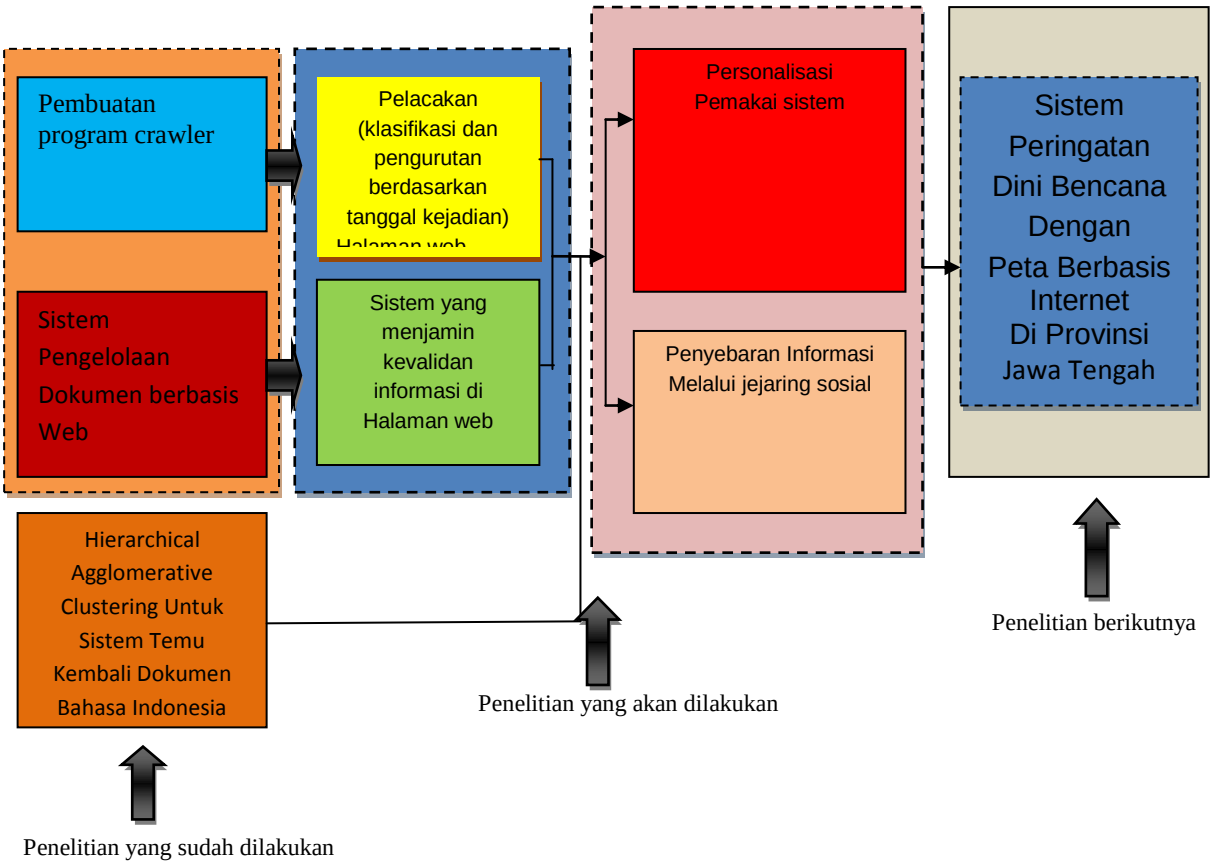
BAB II

STUDI PUSTAKA/ HASIL YANG SUDAH DICAPAI DAN STUDI PENDAHULUAN YANG SUDAH DILAKSANAKAN

2.1 STATE OF THE ART

Penelitian yang berkenaan dengan Sistem Peringatan dini terhadap bencana sudah banyak dilakukan di hampir belahan dunia karena memang diperlukan untuk mencegah jatuhnya korban jiwa yang sangat banyak. (Utami,2008) Meneliti penggunaan SMS Gateway untuk sistem peringatan dini pada bencana banjir. Susetyo (2008) meneliti prediksi coverage jaringan wireless HF untuk sistem peringatan dini di Indonesia.

Pada saat ini beberapa situs internet digunakan untuk memberitahukan atau memberitakan adanya bencana. Sebagai contoh, Badan Meteorologi , Klimatologi dan GeoFisika mengumumkan adanya gempa dan tsunami melalui website.Ataupun www.detik.com, akan memuat berita-berita kejadian yang sedang terjadi. Dengan demikian apabila seseorang dapat mendapat informasi dalam waktu yang singkat. Penelitian ini memanfaatkan informasi-informasi yang telah tersedia sekiranya terjadi suatu bencanatersebut untuk disebar luaskan ke masyarakat.



Gambar 2.1 State of The Art Penelitian Yang Sudah dikerjakan dan Rencana Pengembangan

2.2 CRAWLER

Web Crawler, juga sering dikenal sebagai Web Spider atau Web Robot adalah salah satu komponen penting dalam sebuah mesin pencari modern. Fungsi utama Web Crawler adalah untuk melakukan penjelajahan dan pengambilan halaman-halaman Web yang ada di Internet. Hasil pengumpulan situs Web selanjutnya akan diindeks oleh mesin pencari sehingga mempermudah pencarian informasi di Internet.

Mendesain sebuah crawler yang baik saat ini menemui banyak tantangan. Secara eksternal, crawler harus mengatasi besarnya situs Web dan link jaringan.

Secara internal , crawler harus mengatasi besarnya volume data. Sehubungan dengan terbatasnya sumber daya komputasi dan keterbatasan waktu, maka harus hati-hati memutuskan URL apa yang harus di scan dan bagaimana urutannya. Crawler tidak dapat mengunduh semua halaman web. Penting bagi crawler untuk memilih halaman dan mengunjungi halaman yang penting dulu dengan memprioritaskan URL yang penting tersebut dalam antrian. Crawler juga harus memutuskan berapa frekuensi untuk merevisi halaman yang pernah dilihat, untuk memberikan informasi ke client perubahan yang terjadi di Web. Zuliarso E. dan Mustofa K (2009a) telah menguji algoritma kunjungan crawler berdasarkan isi halaman web. Dalam Zuliarso E. dan Mustofa K (2009b) telah menguji algoritma penelusuran berdasarkan breadth first search, banyaknya backlink, dan ontologi.

2.3. PENGINDEKAN

Inverted file atau index inverted adalah mekanisme untuk pengindeksan kata dari koleksi teks yang digunakan untuk mempercepat proses pencarian. Struktur inverted file terdiri dari dua elemen, yaitu: kata (*vocabulary*) dan kemunculan (*occurences*). Kata-kata tersebut adalah himpunan dari kata-kata yang ada pada teks, atau merupakan ekstraksi dari kumpulan teks yang ada. Februariyanti H., 2010 melakukan penelitian menggunakan algoritma indeks inverted untuk proses indeks kata (*term*), cosine similaritas untuk menghitung kesamaan kata dalam dokumen.

2.4. PENDETEKSIAN KEJADIAN

Tujuan dari pendeteksi kejadian adalah untuk mengidentifikasi cerita yang membicarakan kejadian yang sebelumnya tidak dilaporkan. Masalah ini berkaitan dengan sistem yang mampu membuat keputusan ya/tidak tanpa campur tangan pemakai (Allan,1998)(Papka, 1999).

Dari perspektif jurnalis, sebuah cerita tentang kejadian secara khusus akan berisi : 1. kapan kejadian terjadi; 2. siapa saja yang terlibat; 3. dimana kejadian terjadi; 4. bagaimana kejadiannya; dan 5. pengaruh atau dampaknya. Namun demikian tidak semua berita memuat informasi tersebut. Oleh karena itu diperlukan dokumen terkait untuk meyakinkan bahwa informasi tersebut benar.

2.5. KLASTERING

Penyelesian masalah pendeteksian kejadian yang baru berhubungan dengan masalah klastering kejadian secara on-line. Dalam masalah ini, dokumen-dokumen berita dikelompokkan bersama jika membicarakan berita kejadian yang sama. Februariyanti H.,2010, melakukan penelitian ini untuk mengklaster dokumen dengan menggunakan Algoritma Hierarchical Agglomerative Clustering. Klastering ini ditekankan untuk dokumen berbahasa Indonesia. Keterkaitan antar dokumen diukur berdasarkan kemiripan antar dokumen (similarity).

2.6. PELACAKAN KEJADIAN

Tujuan dari pelacakan adalah untuk mengambil bagian dari deretan berita-berita yang membahas suatu kejadian tertentu. Hal ini dilakukan dengan melakukan klasifikasi dan selajutnya melakukan pengurutan kejadian (Allan,1998) (Papka,1999).

2.7. VALIDASI

Tujuan dari validasi adalah menjamin bahwa informasi yang didapat kemudian disampaikan mengandung nilai kebenaran. Validasi dilakukan dengan melakukan pengujian berdasarkan basisdata dokumen yang telah diklaster dan basisdata dokumen yang telah dilacak (Vert,2010).

2.8. PERSONALISASI

Personalisasi Web adalah suatu proses mengumpulkan dan menyimpan informasi tentang lokasi pengunjung, meneliti informasi, dan berdasarkan pada analisa, mengirimkan informasi yang tepat kepada masing-masing pengunjung di waktu yang tepat, jadi personalisasi merupakan suatu upaya untuk memberikan layanan dalam bentuk aplikasi dan informasi yang disesuaikan dengan minat, peran, dan kebutuhan pengunjung web (Nasraoui,2005).

Personalisasi Web dikategorikan ke dalam beberapa model, dari yang paling sederhana sampai yang kompleks (JYH, P. ,2006), yaitu:

1. Memorization

Seperti namanya model ini merupakan sebuah bentuk personalisasi yang digunakan untuk mengingat atau menyimpan informasi dari pengunjung.

2. Customization

   Customization merupakan sebuah bentuk personalisasi dengan mengambil input dari informasi pengunjung pada saat melakukan registrasi, kemudian data tersebut digunakan untuk melakukan customisasi struktur dan isi halaman web.

3. Guidence or Recommender System

   Model personalisasi ini memberikan rekomendasi kepada pengunjung sesuai dengan minat dan selera pengunjung.

4. Task Performance Support

   Model ini melibatkan client-side system dengan bantuan perangkat lunak yang akan membantu pengunjung dalam menelusuri halaman web sesuai dengan informasi yang diinginkan.

2.9. PENYEBARAN INFORMASI MELALUI JEJARING SOSIAL

Jejaring sosial adalah struktur sosial yang terdiri dari elemen-elemen individual atau organisasi. Jejaring ini menunjukan jalan dimana mereka berhubungan karena kesamaan sosialitas, mulai dari mereka yang dikenal sehari-hari sampai dengan keluarga. Istilah ini diperkenalkan oleh profesor J.A. Barnes di tahun 1954.

Facebook (atau facebook) adalah sebuah layanan jejaring sosial dan situs web yang diluncurkan pada Februari 2004 yang dioperasikan dan dimiliki oleh

Facebook, Inc. Pada Januari 2011, Facebook memiliki lebih dari 600 juta pengguna aktif Pengguna dapat membuat profil pribadi, menambahkan pengguna lain sebagai teman dan bertukar pesan, termasuk pemberitahuan otomatis ketika mereka memperbarui profilnya. Selain itu, pengguna dapat bergabung dengan grup pengguna yang memiliki tujuan tertentu, diurutkan berdasarkan tempat kerja, sekolah, perguruan tinggi, atau karakteristik lainnyaa.

Konsep yang diusung oleh Twitter adalah menyebarkan informasi pesan secara singkat, padat dan real time di dalam kalimat kurang dari 140 karakter kepada pembacanya diseluruh dunia.

Pengguna Twitter dapat menyebarkan informasi pesan singkat melalui beberapa cara, bisa melalui situs Twitter sendiri, melalui SMS, atau melalui aplikasi Twitter lainnya seperti Twirl, Snitter, atau Twitterfox yang merupakan aplikasi tambahan untuk browser Firefox. Karena kandungan pesan yang singkat, Twitter dimasukkan dalam kategori mikroblog, yaitu sebuah media online yang memungkinkan penggunanya menuliskan informasi pesan secara singkat. Panjang pesan tersebut biasanya kurang dari 200 karakter.

BAB III

TUJUAN PENELITIAN

3.1. TUJUAN PENELITIAN

Tidak berlebihan jika saat ini Indonesia disebut sebagai negara yang cukup rawan terhadap bencana alam. Dalam dekade terakhir, perjalanan Indonesia diwarnai berbagai bencana alam yang melanda berbagai daerah. Mulai dari tanah longsor, banjir, gempa bumi, gunung meletus, bahkan tsunami. Berita-berita terkait bencana sudah menjadi hal yang biasa kita dengar setiap harinya. Kondisi ini seolah tak terelakkan bagi Indonesia. Selalu ada saja berita bencana dari waktu ke waktu.

Hal yang saat ini mendesak untuk diusahakan adalah meminimalisir dampak bencana dengan adanya sistem peringatan dini bencana. Bagi bangsa sebesar Indonesia dengan tingkat kerawanan bencana yang tinggi, sistem peringatan dini benar-benar menjadi kebutuhan. Pengalaman bangsa Indonesia terhadap bencana sudah menjadi bukti betapa bangsa Indonesia tidak siap dan tidak memiliki persiapan untuk menghadapi bencana yang sewaktu-waktu bisa datang tanpa diundang. Bahkan bisa dikatakan bangsa Indonesia menjadi pasrah terhadap bencana. Bangsa Indonesia mungkin paham tindakan kuratif yang akan dilakukan setelah bencana datang, tetapi tidak begitu baik dalam hal tindakan preventif.

Untuk itu, penelitian ini akan mengeksplorasi pertanyaan ‘Bagaimana membangun sistem peringatan dini bencana berdasarkan situs berita di web melalui jejaring sosial?’.

Dengan berdasarkan hal tersebut diatas maka tujuan dari penelitan ini adalah :

1. Membuat sistem crawler yang mampu menelusuri dan mendeteksi adanya kejadian bencana di situs web.
2. Membangun modul program untuk mengidentifikasi bahwa halaman situs memuat informasi tentang bencana.
3. Membangun modul program yang mampu mengelompokkan koleksi dokumen berita dalam klaster-klaster.
4. Membangun modul program yang mampu melacak berita sehingga dokumen berita yang sudah diklaster dapat diurutkan berdasarkan urutan tanggal kejadian.
5. Membuat modul program yang mampu memvalidasi kebenaran isi berita tentang terjadinya bencana berdasarkan basis data dokumen yang tersedia.
6. Membangun modul personalisasi untuk pemakai.
7. Merancang dan membuat sebuah perangkat lunak menyampaikan informasi melalui Facebook dan Twitter pada para pemakai sesuai dengan kriteria yang telah ditentukan sebelumnya.

3.2. MANFAAT PENELITIAN

Penelitian ini bertujuan untuk membangun basisdata halaman web yang berkaitan dengan bencana, mengelompokkan halaman web ke dalam klaster-

klaster dan selanjutnya mengurutkan kejadian berdasarkan urutan waktu. Dengan pengetahuan yang tersedia ini dapat digunakan untuk memprediksi bencana yang mungkin terjadi ataupun memvalidasi apakah suatu informasi tentang bencana yang dimuat disuatu situs berita merupakan informasi yang valid.

Keutamaan dari penelitian ini adalah :

1. Pada saat ini di Indonesia belum ada sistem peringatan dini yang terintegrasi. Misalnya untuk gempa dan tsunami, maka informasi akan disediakan oleh Badan Meteorologi dan Geofisika. Sedang informasi bencana seperti banjir dan tanah longsor banyak tersedia di situs berita. Sistem yang diusulkan diharapkan mampu mengintegrasikan berbagai informasi tentang bencana dalam satu sistem.
2. Saat ini banyak situs berita yang segera memuat peristiwa yang sedang terjadi dalam waktu yang tidak lama . Untuk itu diperlukan sistem crawler yang akan memantau perkembangan situs-situs berita sehingga kalau ada peristiwa / kejadian akan segera terdeteksi.
3. Sistem mampu memberikan peringatan secara dini tentang bencana sesegera mungkin pada para pengguna Facebook dan Twitter. Pada saat ini handphone yang memiliki fasilitas Facebook dan Twitter sangat banyak dan layanan aksesnyapun gratis. Sehingga diharapkan para pemakai handphone akan sangat mudah mendapatkan informasi tentang bencana kapanpun dan dimanapun. Selanjutnya para pemakai dapat segera menyebarkan ke teman yang lain. Untuk informasi selanjutnya para pemakai dapat menggunakan fasilitas replay (balas).

4. Sistem juga mampu memberikan informasi tentang bencana apa saja dan kapan saja yang telah terjadi disuatu tempat. Hal ini dikarenakan sistem akan mengelompokkan bencana berdasarkan klaster dan diurutkan berdasarkan urutan kejadian. Pengetahuan ini digunakan untuk memvalidasi suatu kejadian yang baru dan dapat juga digunakan untuk memprediksi kejadian apa yang mungkin terjadi di daerah tersebut.
5. Karena sistem mempunyai modul personalisasi, maka maka para pemakai dapat meminta informasi tentang bencana di satu tempat sesuai dengan keinginan pemakai..Misalkan pemakai dapat memilih informasi di daerah tertentu saja.
6. Memanfaatkan fasilitas yang disediakan di internet untuk dipakai oleh pengguna untuk peringatan dini terhadap bencana. Sehingga seseorang dapat mengantisipasi dengan baik. Disamping itu juga memberikan manfaat lebih dalam penggunaan Facebook dan Twitter, yang selama ini hanya untuk pertemanan dan pemasaran.

BAB IV

METODE PENELITIAN

4.1. LANGKAH-LANGKAH PENELITIAN

1. Obyek Penelitian

   Obyek penelitian adalah situs-situs berita ataupun situs lembaga pemerintah yang memberikan informasi tentang bencana.

2. Metode pengumpulan data :

   Pengumpulan data dilakukan dengan mengumpulkan data dari berbagai situs berita maupun situs lembaga pemerintah yang memberitakan kejadian bencana.

3. Data dan alat

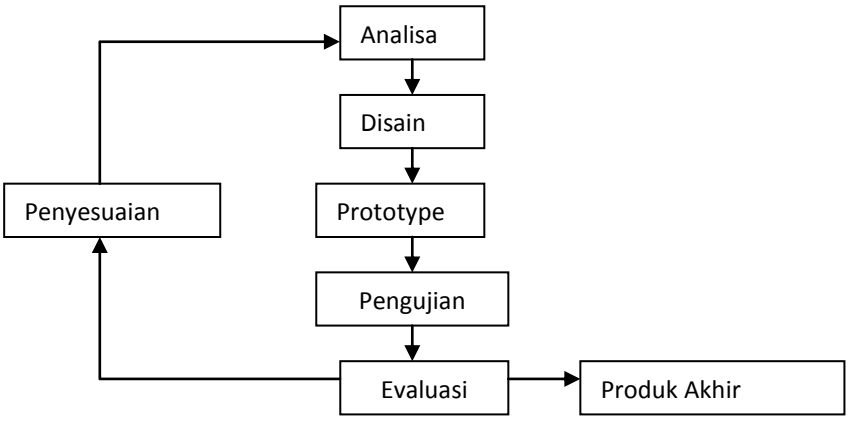
   Data Yang Dibutuhkan Untuk Penelitian Ini adalah:

   a. Halaman situs berita.

   b. Kamus Elektronik Bahasa Indonesia.

4.2. METODE PENGEMBANGAN

Penelitian ini menggunakan model prototyping. Di dalam model ini sistem dirancang dan dibangun secara bertahap dan untuk setiap tahap pengembangan dilakukan percobaan-percobaan untuk melihat apakah sistem sudah bekerja sesuai dengan yang diinginkan. Sistematika model prototyping terdapat pada gambar 4.1. memperlihatkan tahapan pada prototyping.



Gambar 4.1 Tahapan Prototyping (Pressman, 1997)

Berikut adalah tahapan yang dilakukan pada penelitian ini dengan metode pengembangan prototyping :

1. Analisa

   Pada tahap ini dilakukan analisa tentang masalah penelitian dan menentukan pemecahan masalah yang tepat untuk menyelesaikannya.

2. Disain

   Pada tahap ini dibangun rancangan sistem dengan menggunakan tools pengembangan sistem informasi yaitu DFD, ERD, Class Diagram dan flowchart.

3. Prototype

   Pada tahap ini dibangun aplikasi sistem temu kembali informasi bahasa Indonesia dengan metode Hierarchical Agglomerative Clustering sesuai dengan disain dan kebutuhan sistem.

4. Pengujian

   Pada tahap ini dilakukan pengujian pada aplikasi yang sudah dibangun, pengujian dilakukan dengan validasi dengan menggunakan input query

dalam bentuk teks dan kesesuaian query dengan hasil simmilaritas dan hasil klaster yang di dapat dari aplikasi.

5. Evaluasi

   Pada tahap ini dilakukan evaluasi apakah performa aplikasi sudah sesuai dengan yang diharapkan, apabila belum maka dilakukan penyesuaian – penyesuaian sesuai kebutuhan.

6. Penyesuaian

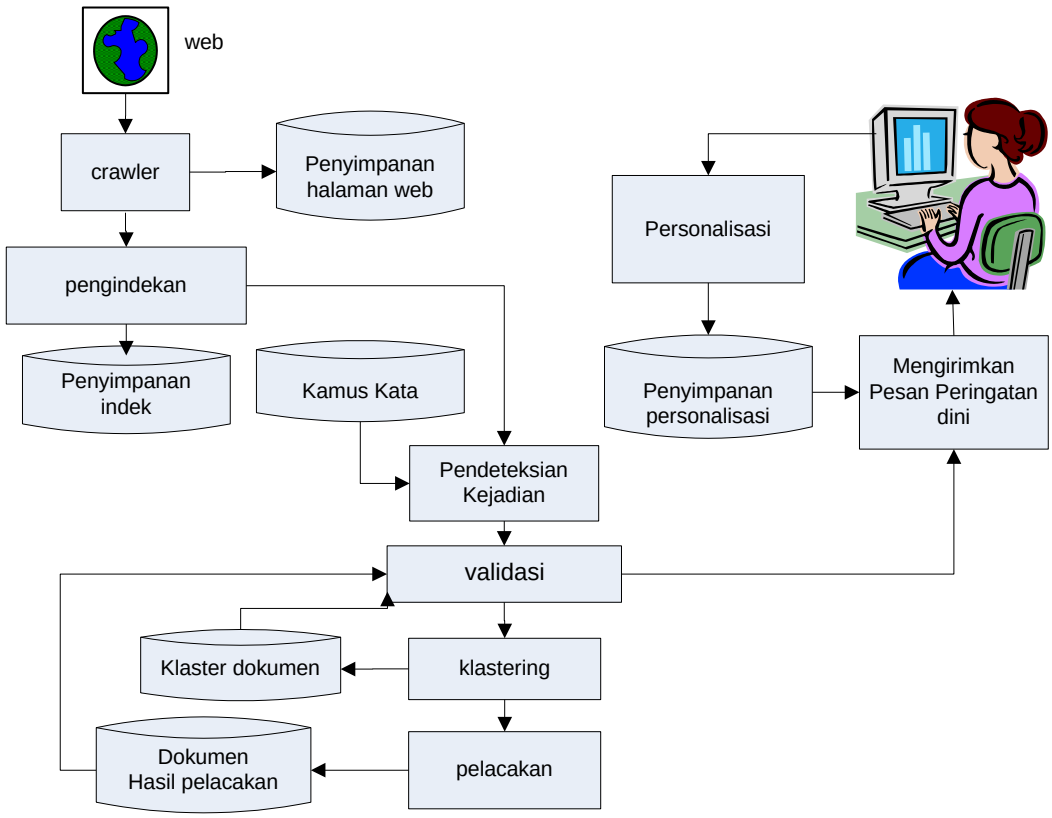
   Tahap ini dilakukan apabila pada evaluasi performa aplikasi kurang memadai dan dibutuhkan perbaikan, tahap ini melakukan penyesuaian dan perbaikan pada aplikasi sesuai dengan kebutuhan.

BAB V

HASIL DAN PEMBAHASAN

5. 1. KONSTRUKSI DESAIN SISTEM

Modul indexer mengekstrak semua kata dalam tiap halaman, dan menyimpan URL dimana tiap kata muncul. Hasilnya adalah "*lookup table*" yang sangat besar yang menyediakan semua URL yang menunjuk ke halaman-halaman dimana sebuah kata yang diberikan muncul.



Gambar 5. 1 Arsitektur Sistem Peringatan Dini Berbasis Situs Berita Melalui Jejaring Sosial

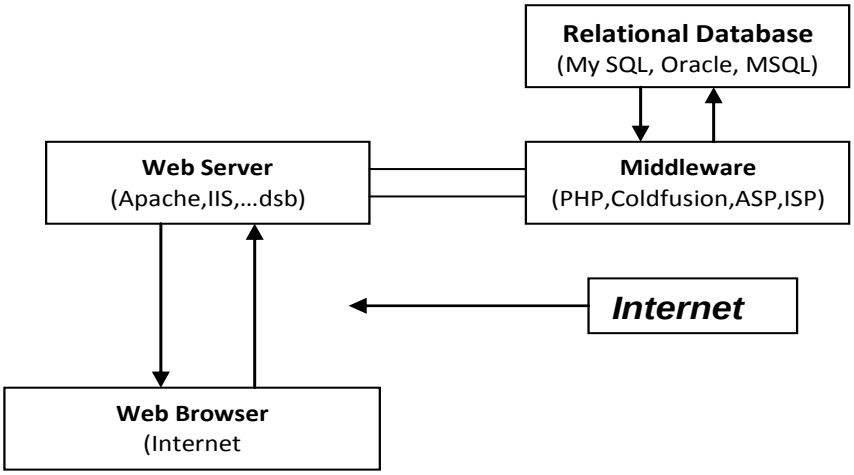
Identifikasi kejadian dillakukan kombinasi dari satu kata sampai satu kalimat untuk mencari kata-kata yang sesuai dengan kamus yang berisi kata-kata

yang berkaitan dengan bencana. Selanjutnya dilakukan validasi informasi berdasarkan basis data klaster dokumen dan basis data dokumen yang telah dilacak (diklasifikasi dan diurutkan berdasarkan kejadian).

Apabila informasi memang valid maka sistem melakukan pembacaan basisdata personal. Berdasarkan basisdata personal, maka dilakukan penyebaran informasi melalui Facebook dan Twitter.

5.2. ARSITEKTUR APLIKASI WEB

Pada tingkat yang paling rendah, web bekerja pada arsitektur client server yang berarti antara keduanya baik sentral server dan aplikasi client bertanggung jawab pada sejumlah proses, secara detail arsitektur aplikasi web digambarkan pada Gambar 5.2 Berbeda dengan program lain yang dapat berjalan tanpa bantuan server.



Gambar 5.2 Arsitektur Aplikasi Web

1. Client

   Aplikasi client tunggal yang dapat dikembangkan melalui MySQL dan PHP adalah aplikasi we browser dan bahasa utama dari web browser adalah HTML yang menyediakan sekumpulan teks yang menjelaskan bagaimana teks ditampilkan.

2. Server

   Kebanyakan dari seluruh pekerjaan aplikasi web terletak di server, aplikasi tersebut disebut sebagai web server yang akan bertanggung jawab untuk berkomunikasi dengan browser yang ada pada client.

3. Sistem Operasi

   Web server, bahasa pemograman, database server harus bekerja dengan sistem operasi. Data base server adalah server yang menangani database. Banyak sekali sistem operasi yang populer saat ini, seperti Windows 98, Windows NT/2000, Macintosh, Unix, Linux dan masih banyak lagi lainnya.

4. Web Server

   Web server merupakan penyimpan data web. Misalnya komputer A menyimpan data web miliknya sendiri. Ia memberikan service kepada komputer lain yang ingin mengakses data web tersebut. (tentunya bila sudah ada koneksi antara komputer A dengan komputer lainnya tersebut). Hampir semua pekerjaan dari aplikasi web berada di server. Aplikasi web server tersebut bisa berupa Apache (web server yang bekerja di lingkungan

unix dan juga di Windows OS), IIS (web server yang bekerja pada Windows OS dan merupakan komponen kunci dari Microsoft ASP) dsb.

5. Middleware

   PHP termasuk dalam class bahasa middleware. Bahasa ini bekerja pada web server sebagai interpreter permintaan dari client, memproses permintaan, menghubungkan dengan program–program lain di server untuk memenuhi permintaan dan kemudian dikirimkan kembali ke browser client.

6. Relational Database

   Relational Database Management System (RDBMS) menyediakan cara yang terbaik untuk menyimpan dan mengakses suatu informasi yang komplek. Beberapa RDBMS komersial yang populer antara lain : Oracle, MSSQL Server, IBM db2 sebagai tambahan untuk MySQL pada saat ini terdapat dua open source RDBMS yang besar yaitu : PostgressQL dan MySQL.

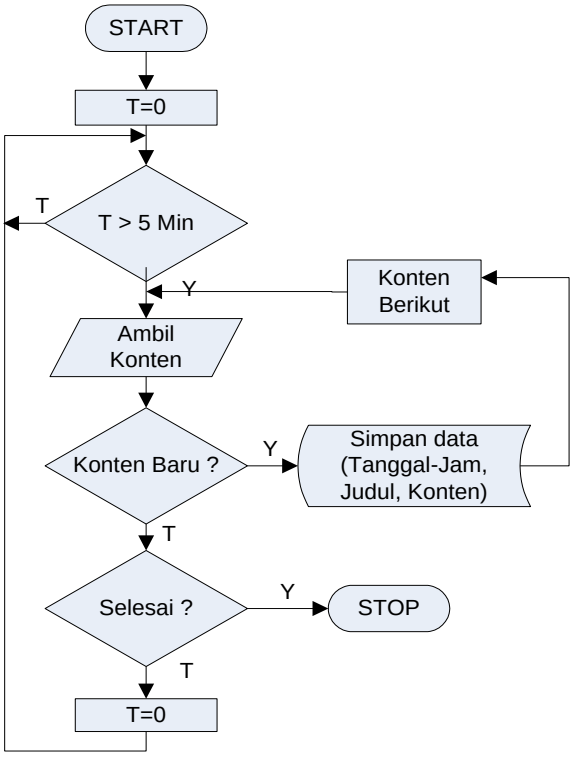
5.3. PEMBUATAN PROGRAM CRAWLER DAN KORPUS DOKUMEN BERITA

Web Crawler, juga sering dikenal sebagai Web Spider atau Web Robot adalah salah satu komponen penting dalam sebuah mesin pencari modern. Fungsi utama Web Crawler adalah untuk melakukan penjelajahan dan pengambilan halaman-halaman Web yang ada di Internet. Hasil pengumpulan situs Web

selanjutnya akan diindeks oleh mesin pencari sehingga mempermudah pencarian informasi di Internet.

5.3.1. Perencanaan

Crawl digunakan untuk mengambil konten pada situs target secara berkala. Proses dilakukan terus-menerus dan dilakukan secara otomatis. Program ini ditempatkan pada server yang terkoneksi internet 24 jam, sehingga memperkecil kehilangan informasi / berita.



Gambar 5.3. Flowchart Program Crawler

Pada gambar 5.3 diperlihatkan alur program dari bagian crawl. Secara umum program akan aktif setiap 5 menit, sehingga setiap 5 menit sekali program akan melakukan pengambilan konten pada situs target melakukan, sekaligus

melakukan pemeriksaan apakah konten tersebut baru atau sudah pernah diproses. Apabila konten tersebut baru maka konten tersebut akan disimpan di basis data, apabila konten sudah pernah disimpan maka dilanjutkan ke konten berikutnya.

5.3.2. Analisis Kebutuhan

Mendesain sebuah crawler yang baik saat ini menemui banyak tantangan. Secara eksternal, crawler harus mengatasi besarnya situs Web dan link jaringan. Secara internal , crawler harus mengatasi besarnya volume data. Sehubungan dengan terbatasnya sumber daya komputasi dan keterbatasan waktu, maka harus hati-hati memutuskan URL apa yang harus di scan dan bagaimana urutannya. Crawler tidak dapat mengunduh semua halaman web. Penting bagi crawler untuk memilih halaman dan mengunjungi halaman yang penting dulu dengan memprioritaskan URL yang penting tersebut dalam antrian. Crawler juga harus memutuskan berapa frekuensi untuk merevisi halaman yang pernah dilihat, untuk memberikan informasi ke client perubahan yang terjadi di Web.

5.3.3. Perancangan

Walaupun banyak aplikasi untuk Web crawler, pada intinya semuanya secara fundamental sama. Berikut ini proses yang dilakukan Web crawler pada saat bekerja :

1. Mengunduh halaman Web.
2. Memparsing halaman yang didownload dan mengambil semua link.
3. Untuk setiap link yang diambil, ulangi proses.

Perancangan Basis data

Setelah konten baru berhasil didownload maka segera konten tersebut disimpan ke dalam tabel terstruktur. Dalam penelitian ini aplikasi basis data yang digunakan adalah mysql dengan pertimbangan gratis dan mudah diinstall pada server berbasis linux.

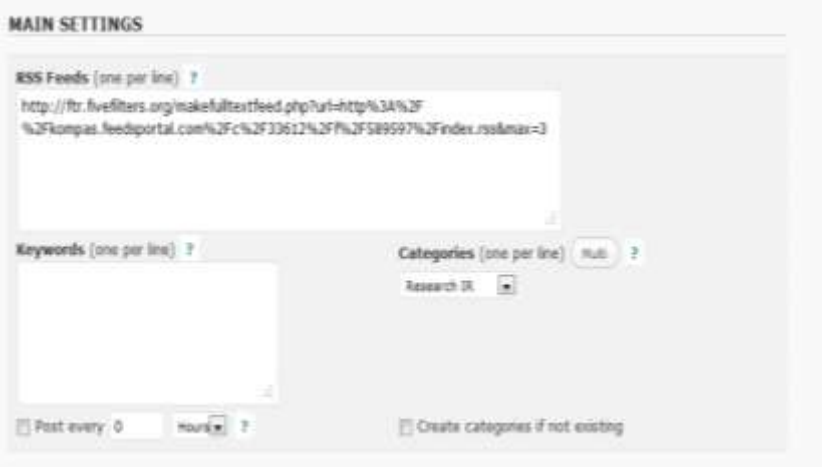
Pada tabel 5.1 diperlihatkan struktur inti dari tabel yang digunakan untuk menyimpan hasil pengambilan konten. Pada tabel tersebut ID menjadi kunci primer, sementara attribut lain yang disimpan adalah waktu pengambilan, judul dan isi artikel.

Tabel 5.1 Tabel Konten

NO	Nama	Tipe	Kunci
1	Id	BIGINT	Primer
2	post_date	Time Stamp	
3	Post_title	Text	
4	Post_content	Text	
5	Status	Varchar(15)	index

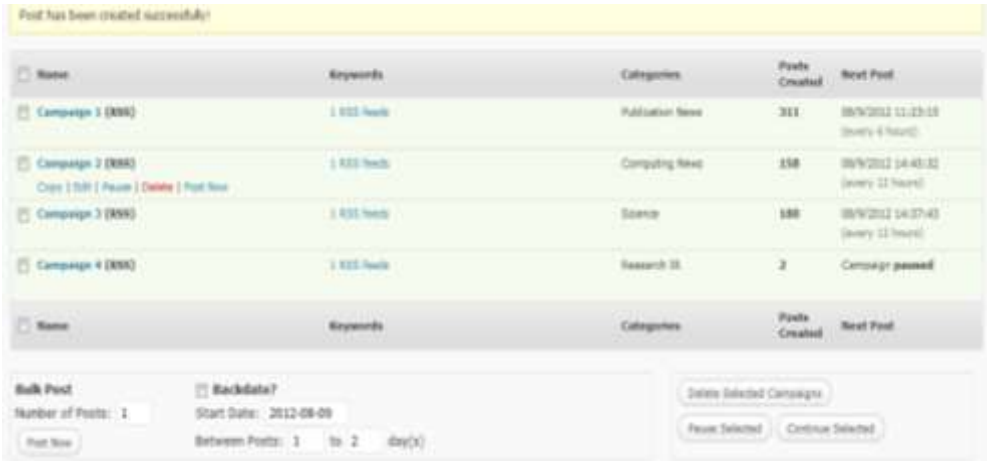
5.3.4. Implementasi

Beberapa pengaturan yang diperlukan untuk menjalankan crawl diantaranya adalah alamat situs target, dalam penelitian ini target situs menggunakan alamat rss sehingga mempermudah pencarian artikel atau berita terbaru. Pada penelitian ini juga menggunakan situs bantu fivefilters.org untuk mengambil konten penuh dari suatu feed rss.



Gambar 5.4. Tampilan Layar Pengaturan Parameter Crawler

Pada gambar 5.4 merupakan tampilan layar untuk pengaturan parameter-parameter yang diperlukan pada saat crawling. 1) Alamat situs rss target 2) Kategori dimana hasil pengambilan konten akan disimpan 3) Kata kunci yang digunakan untuk menyaring.



Gambar 5.5 Tampilan Layar untuk Pengambilan Konten

Pada Gambar 5.5 merupakan tampilan layar pada saat sebuah konten dari situs target berhasil diambil dan disimpan di tabel. Program crawl ini dapat mentargetkan lebih dari satu situs dan dengan interval pengambilan yang berbeda-beda.

1. Prosedur Ambil Konten

Implementasi prosedur ambil konten menggunakan bahasa pemrograman php, metode pengambilan menggunakan CURL. Pemilihan metode ini dikarenakan metode ini dapat terdeksi oleh web server selayaknya browser dengan operator manusia, sehingga web server akan memberikan isi halaman web selengkap-lengkapnya. Sementara dengan metode lain crawl akan terdeteksi sebagai *bot* yang mungkin saja konten yang di berikan tidak selengkap apabila di deteksi sebagai manusia. Pada kode sumber 1 diperlihatkan kelas pada program php yang digunakan untuk mengambil target situs target.

2. Kode Sumber 1

Berikut ini adalah implementasi dari proses crowler. Proses crawler disimpan dalam class *mycurl.*

Algoritma 5.1 Algoritma Crawler

```
<?php
 class mycurl {
    protected $_useragent = 'Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1';
    protected $_url;
    protected $_followlocation;
    protected $_timeout;
    protected $_maxRedirects;
    protected $_cookieFileLocation = './cookie.txt';
    protected $_post;
    protected $_postFields;
    protected $_referer ="http://www.google.com";
    public function createCurl($url = 'nul')
    {
       curl_setopt($s,CURLOPT_URL,$this->_url);
       curl_setopt($s,CURLOPT_HTTPHEADER,array('Expect:'));
       curl_setopt($s,CURLOPT_TIMEOUT,$this->_timeout);
       curl_setopt($s,CURLOPT_MAXREDIRS,$this->_maxRedirects);
       curl_setopt($s,CURLOPT_RETURNTRANSFER,true);
       curl_setopt($s,CURLOPT_FOLLOWLOCATION,$this->_followlocation);
       curl_setopt($s,CURLOPT_COOKIEJAR,$this->_cookieFileLocation);
       curl_setopt($s,CURLOPT_COOKIEFILE,$this->_cookieFileLocation);
       if($this->authentication == 1){
         curl_setopt($s, CURLOPT_USERPWD, $this->auth_name.':'.$this->auth_pass);
```

```
        }
        if($this->_post) {
           curl_setopt($s,CURLOPT_POST,true);
           curl_setopt($s,CURLOPT_POSTFIELDS,$this->_postFields);
        }
    }
}?>
```

Dari hasil proses korpus akan dihasilkan dokumen berita hasil crawler yang akan disimpan dalam tabel corpus. Tabel corpus dapat dilihat pada gambar 5.2

Tabel 5. 2 Tabel Corpus

Nama Field	Type Field	Keterangan
situs	varchar(200)	Alamat situs halaman web
Isi	Text	Isi halaman web
tanggal	Date	Tanggal melakukan unduh halaman web

5.4. PEMBUATAN PROGRAM KLASTER

Tujuan klastering dokumen adalah untuk memisahkan dokumen yang relevan dari dokumen yang tidak relevan (Zhang J., et all, 2001). Atau dengan kata lain, dokumen-dokumen yang relevan dengan suatu *query* cenderung memiliki kemiripan satu sama lain dari pada dokumen yang tidak relevan, sehingga dapat dikelompokkan ke dalam suatu klaster.

Klastering dokumen dapat dilakukan sebelum atau sesudah proses temu kembali (Zhang J., dkk., 2001). Pada klastering dokumen yang dilakukan sebelum proses temu kembali informasi, koleksi dokumen dikelompokkan ke dalam klaster berdasarkan kemiripan (*similarity*) antar dokumen. Selanjutnya dalam proses temu kembali informasi, apabila suatu dokumen ditemukan maka seluruh dokumen yang berada dalam klaster yang sama dengan dokumen tersebut juga dapat ditemukan.

Klastering dokumen memberikan beberapa manfaat, antara lain:

1. Mempercepat pemrosesan *query* dengan menelusur hanya pada sejumlah kecil anggota atau wakil klaster, sehingga dapat mempercepat proses temu kembali informasi
2. Membantu melokalisasi dokumen yang relevan
3. Membentuk kelas-kelas dokumen sehingga mempermudah penjelajahan dan pemberian interpretasi terhadap hasil penelusuran
4. Meningkatkan efektivitas dan efisiensi temu kembali informasi dan memberikan alternatif metode penelusuran

5.4.1. Perencanaan

Setelah proses crawler selesai dikerjakan dilanjutkan proses klastering. Proses klastering ini dimaksudkan agar dokumen yang tidak relevan dengan topik maka aka dipisahkan dimana dokumen yang memiliki kemiripan satu sama lain akan mengelompok menjadi sebuah klaster. Pada tahap proses klastering dokumen berita dibutuhkan dokumen yang tersimpan dalam korpus hasil crawler.

Pada penelitian ini proses klaster menggunakan algoritma *single pass clustering.* Single Pass Clustering merupakan suatu tipe clustering yang berusaha melakukan pengelompokan data satu demi satu dan pembentukan kelompok dilakukan seiring dengan pengevaluasian setiap data yang dimasukkan ke dalam proses klaster. Pengevaluasian tingkat kesamaan antar data dan klaster dilakukan dengan berbagai macam cara termasuk menggunakan fungsi jarak, vectors similarity, dan lain-lain.

5.4.2. Analisis Kebutuhan

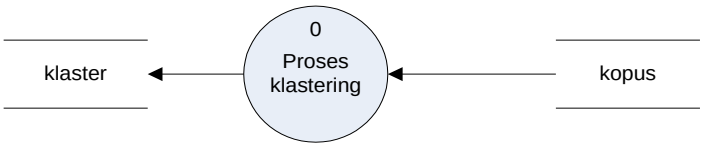
Dalam menggunakan algoritma ini, dua hal yang perlu menjadi perhatian adalah penentuan objective function dan penentuan threshold value. Objective function yang ditentukan haruslah sebisa mungkin mencerminkan keadaan data yang dimodel dan dapat memberikan nilai tingkat kesamaan atau perbedaan yang terkandung di dalam data tersebut. Penentuan threshold value juga merupakan hal yang subjektif, makin besar nilai threshold, makin mudah suatu data untuk bergabung ke dalam suatu klaster, dan demikian juga sebaliknya.

Pada modul klastering dokumen dengan algoritma *Clustering Single Pass* yang dibuat diharapkan memiliki kemampuan :

1. Modul harus mampu membuat database indexing dalam bentuk tabel indeks.
2. Modul harus mampu membuat filter terhadap daftar kata umum (stop word) dan tidak menyimpannya dalam database.
3. Modul harus mampu membuat klaster untuk dokumen yang akan disimpan dalam database.

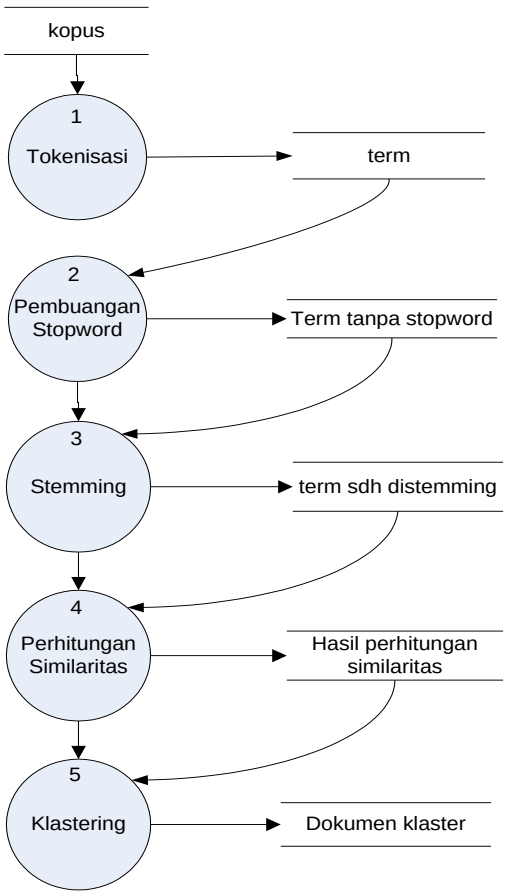
5.4.3. Perancangan Menggunakan DFD dan Flowchart

Pada perancangan program klaster menggunakan algoritma *single pass clustering* digambarkan dengan diagram alir data (DFD). Pada Gambar 5.6 adalah konteks diagram dari program klaster. Program klaster akan dikerjakan dari dokumen yang tersimpan dalam data store korpus. Hasil program klaster akan disimpan dalam data store klaster.



Gambar 5.6 Kontek Diagram Program Klaster

Dari diagram konteks pada gambar 5.6 dilanjutkan dengan perancangan DFD level 0 yang menggambarkan proses lebih rinci dari program klaster. DFD level 0 program klaster dapat dilihat pada Gambar 5.7



Gambar 5.7 DFD Level 0 Program Klaster

Dapat dilihat pada Gambar 5.7 DFD level 0 program klaster. Program dimulai dari proses preprosesing. Proses preprosesing terdiri dari proses tokenisasi, pembuangan stopword, stemming. Preprosesing akan dimulai dari

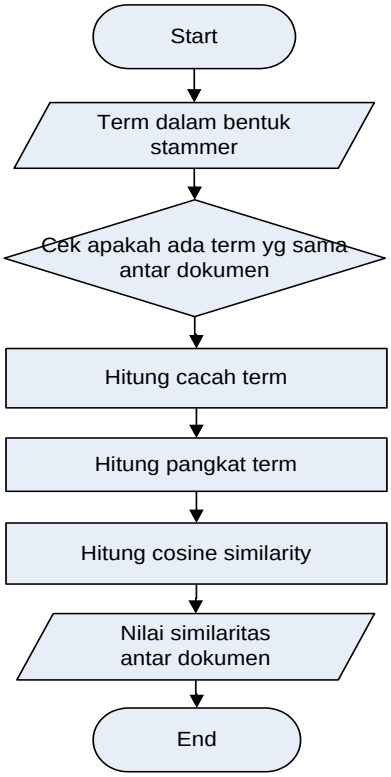
proses tokenisasi yang mengambil dokumen dari data store korpus hasil crawler. Hasil proses tokenisasi disimpan dalam data store term. Dari data store term akan dilakukan proses pembuangan stopword, yaitu proses untuk membuang kata-kata yang tidak bermakna. Hasil dari proses pembuangan stopword akan disimpan dalam data store term tanpa stopword. Kemudian proses akan dilanjutkan dengan stemming, yaitu proses pembentukan kata dasar dan hasilnya akan disimpan dalam data store term sudah distemming. Dari hasil proses stemming akan dilanjutkan dengan proses hitung simmilaritas antar dokumen. Dari hasil perhitungan simmilaritas baru dilakukan proses klastering. Proses klaster akan menggunakan algoritma *single pass* clustering. Hasil proses klaster akan disimpan dalam data store klaster.

Masing-masing proses preprosesing dan proses klaster dapat dijelaskan sebagai berikut :

1. Modul Tokenisasi

Sebelum kata dipisahkan dari kalimatnya, terlebih dahulu dibersihkan dari tanda baca, tag html dan angka. Pada penelitian ini untuk membersihkan tanda baca dapat digunakan perintah yang disediakan oleh Java. Pembersihan dilakukan sebelum proses tokenisasi (*tokenizations)* dimaksudkan untuk memperkecil hasil dari tokenisasi. Pada proses tokenisasi akan dibaca dokumen abstrak dalam format teks akan dilakukan proses pemotongan string input berdasarkan tiap kata yang menyusunnya. Pada umumnya setiap kata terindentifikasi atau terpisahkan dengan kata yang lain oleh karakter spasi, sehingga proses tokenisasi mengandalkan karakter spasi pada dokumen untuk melakukan pemisahan kata.

Seperti yang terlihat pada Gambar 5.8 pada proses preprosesing untuk tokenisasi, semua term dalam dokumen yang dibaca diganti dengan huruf kecil. Setelah itu tiap term akan dicek apakah tanda baca atau tidak. Jika tanda baca maka akan dihapus/dibuang. Proses akan dilanjutkan untuk membuat term menjadi token-token yang terpisah.



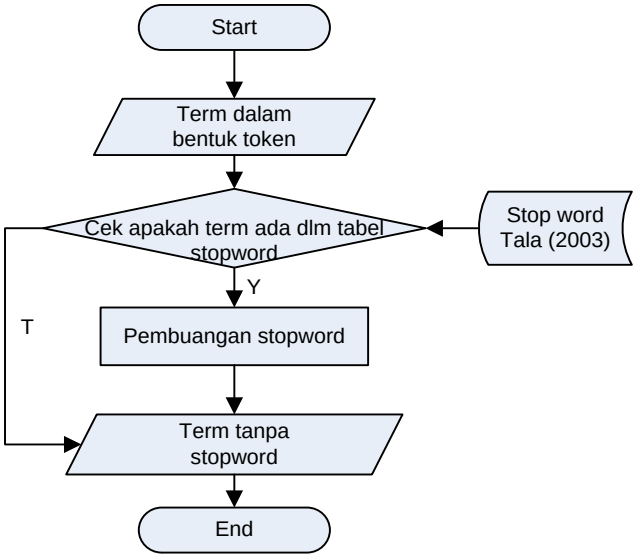
Gambar 5.8 Flowchart Proses Tokenisasi

2. Modul Pembuangan Stop Word

Proses pembuangan stop word dimaksudkan untuk mengetahui suatu kata masuk ke dalam stop word atau tidak. Pembuangan stopword adalah proses pembuangan term yang tidak memiliki arti atau tidak relevan. Term yang diperoleh dari tahap tokenisasi dicek dalam suatu daftar *stopword*, apabila sebuah kata masuk di dalam daftar stopword maka kata tersebut tidak akan diproses lebih

lanjut. Sebaliknya apabila sebuah kata tidak termasuk di dalam daftar stopword maka kata tersebut akan masuk keproses berikutnya. Daftar stop word tersimpan dalam suatu tabel, dalam penelitian ini menggunakan daftar stop word yang digunakan oleh Tala (2003), yang merupakan stop word Bahasa Indonesia yang berisi kata-kata seperti ; ini, itu, yang, ke, di, dalam, kepada, dan seterusnya sebanyak 780 kata.

Seperti terlihat pada Gambar 5.9 pembuangan stop word dilakukan dengan mengecek pada tabel stop word. Bila term cocok dengan salah satu isi tabel stop word, maka term tersebut dianggap sebagai stop word akan dibuang dan tidak akan diikutkan pada proses *stemming*. Dari proses pembuangan stop word akan menghasilkan term tanpa stop word.



Gambar 5.9 Flowchart Proses Pembuangan Stop Word

3. Modul Stemming

Proses *stemming* adalah proses pembentukan kata dasar. Term yang diperoleh dari tahap pembuangan stop word akan dilakukan proses stemming.

Algoritma stemming yang digunakan adalah modifikasi Porter stemmer dari (Tala, 2003). Stemming digunakan untuk mereduksi bentuk term untuk menghindari ketidak cocokan yang dapat mengurangi recall, di mana term-term yang berbeda namun memiliki makna dasar yang sama direduksi menjadi satu bentuk.

Proses stemming adalah bagian dari proses filtering, tujuan utama dari proses stemming adalah mengembalikan kata dalam bentuk dasarnya. Dengan kata dasar dapat mereduksi bentuk term untuk menghindari ketidak cocokan yang dapat mengurangi recall, di mana term-term yang berbeda namun memiliki makna dasar yang sama direduksi menjadi satu bentuk.

Struktur pembentukan kata dalam Bahasa Indonesia adalah sebagai berikut:

[awalan-1] + [awalan-2] + dasar + [akhiran] + [kepunyaan] + [sandang]

Masing-masing bagian tersebut (yang dalam kotak bisa ada atau tidak), digabungkan dengan kata dasar membentuk kata berimbuhan.

Penggunaan algoritma stemming Tala bertujuan untuk mempercepat waktu implementasi dan diharapkan performa yang stabil walaupun data dokumen bertambah terus. Algoritma Tala menggunakan algoritma rule based stemming seperti halnya dengan algoritma porter pada stemming bahasa Inggris.

Pada stemmer Tala terdapat 5 langkah utama dengan 3 langkah awal dan 2 langkah pilihan, langkah-langkah tersebut sbb:

a. Menghilangkan partikel
b. Menghilangkan kata sandang dan kepunyaan.

c. Menghilangkan awalan 1

d. Jika suatu aturan terpenuhi jalankan sbb :

   - Hilangkan Akhiran
   - Jika suatu aturan terpenuhi, hilangkan awalan 2. Jika tidak proses stemming selesai

e. Jika tidak ada aturan yang terpenuhi jalankan sbb :

   - Hilangkan awalan 2.
   - Hilangkan Akhiran
   - Proses stemming selesai.

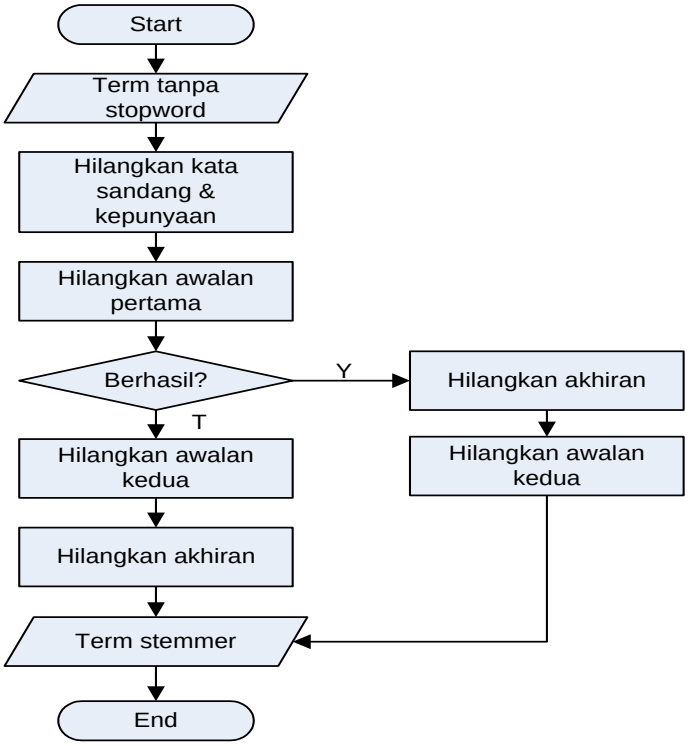
Selain itu tala juga membagi imbuhan menjadi 5 cluster yang nantinya digunakan untuk menghilangkan imbuhan pada setiap tahapnya.

Seperti terlihat pada Gambar 5.10 proses stemming dimulai dari input term tanpa stop word. Term yang dibaca akan dilakukan proses stemmer, pada penelitian ini digunakan Stemmer Tala (2003) yang mengadopsi stemmer Porter.

Tahap pertama proses stemming adalah menghilangkan partikel kata. Proses dilanjutkan dengan menghilangkan kata sandang dan kepunyaan, kemudian dilanjutkan dengan menghilangkan awalan kata pertama. Jika berhasil maka proses selanjutnya adalah menghilangkan akhiran dan awalan kedua. Jika tidak berhasil maka dilakukan adalah menghilangkan awalan kedua dan akhiran.

Output dari proses stemmer adalah term dalam bentuk kata dasar. Hasil proses akan dilakukan perhitungan cacah term, jika ketemu term yang sama maka cacah term akan bertambah jumlahnya. Kata dasar ini akan diindeks oleh sistem

dan akan disimpan dalam tabel koleksi. Tabel koleksi akan menyimpan term-term dari masing-masing dokumen.



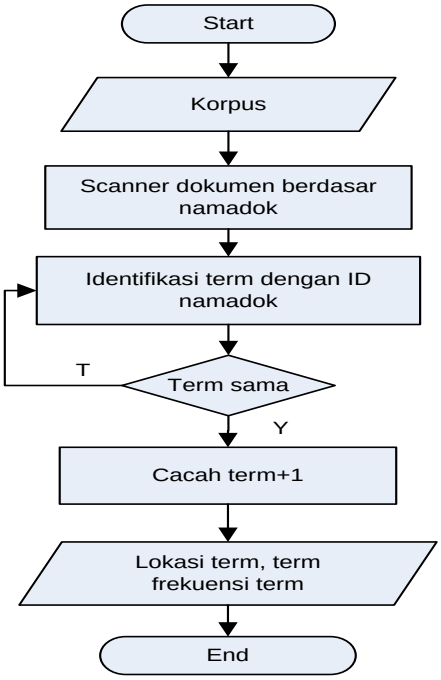
Gambar 5.10 Flowchart Proses Stemming

4. Modul Indexing

Proses *indexing* merupakan tahapan preprocessing yang sangat penting dalam sistem temu kembali informasi sebelum pemrosesan query. Pada proses ini seluruh dokumen dalam koleksi disimpan dalam suatu file dengan format sedemikian sehingga dokumen satu dengan dokumen yang lain dapat dibedakan. Setelah kata telah dikembalikan dalam bentuk asal (kata dasar), kata-kata tersebut disimpan kedalam tabel basis data. Penelitian ini menggunakan metode *Inverted Index*, dengan struktur terdiri dari: kata (*term*) dan kemunculan. Kata-kata tersebut adalah himpunan dari kata-kata yang ada pada dokumen, merupakan ekstraksi dari

kumpulan dokumen yang ada. Setiap term akan ditunjukkan informasi mengenai semua posisi kemunculannya secara rinci.

Setelah kata telah dikembalikan dalam bentuk asal (kata dasar), kata-kata tersebut disimpan kedalam tabel basis data.. Pada proses ini seluruh dokumen akan disimpan dalam suatu file dengan format sedemikian sehingga dokumen satu dengan dokumen yang lain dapat dibedakan. Proses indek menggunakan metode inverted indexing. Yaitu metode indeks dengan membedakan letak term (kata) tiap-tiap term dalam dokumen. Untuk membedakan letak term digunakan nama dokumen sebagai indeks. Tiap term yang terbaca akan disimpan dengan indeks nama dokumen. Flowchart untuk proses indexing dapat dilihat pada Gambar 5.11



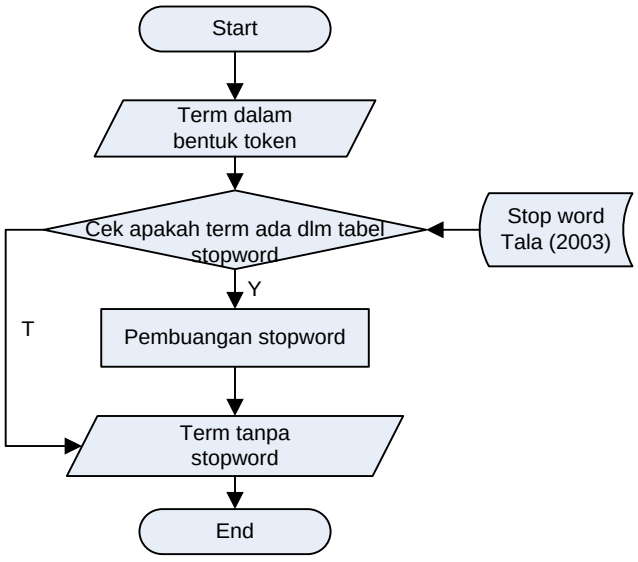
Gambar 5.11 Flowchart Proses Indexing

5. Modul Hitung Similaritas

Relevansi sebuah dokumen ke sebuah *query* didasarkan pada *similarity* (similaritas) diantara vektor dokumen dan vektor *query*. Koordinat dari bobot

istilah secara dasarnya diturunkan dari frekuensi kemunculan dari istilah. Pada modul ini akan dihitung presentase kemunculan tiap kata (term) dan presentase kesamaan antar dua term. Metode yang digunakan untuk menghitung adalah metode *cosine simmilarity*. Masing-masing dokumen akan dihitung cacah term yang sama antara dokumen yang satu dengan dokumen yang lain. Hasil dari hitung cacah akan dihasilkan dokumen dengan nilai similaritas dokumen. Nilai similaritas dokumen yang tertinggi dapat dianggap bahwa dokumen tersebut paling simmilar, yaitu memiliki banyak kesamaan.

Seperti terlihat pada Gambar 5.12 proses hitung similaritas dokumen beberapa proses telah dilakukan pada proses indexing. Proses indexing menghasilkan dokumen yang telah teridentifikasi lokasi dan jumlah cacah term, dan pangkat frekuensi term. Cacah term dan pangkat term akan digunakan untuk hitung *cosine similarity*. Output adalah nilai similaritas dokumen yang satu dengan dokumen yang lain.



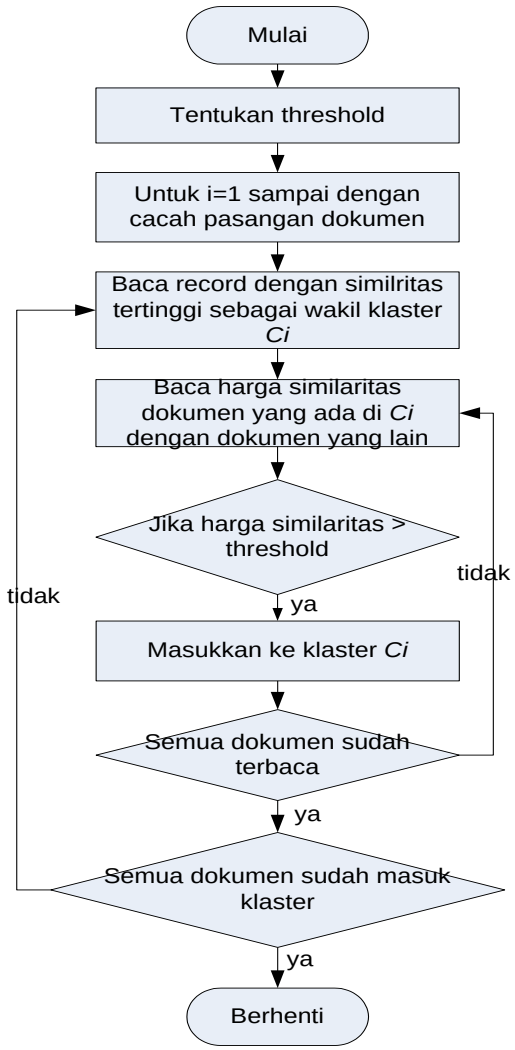
Gambar 5.12 Flowchart Proses Hitung Similaritas

6. Modul Klastering

Pada penelitian ini dokumen akan dibuat klaster dengan menggunakan metode *Clustering Single Pass*. Metode ini berawal dari objek-objek individual. Jadi pada awalnya banyaknya klaster sama dengan banyaknya objek. Pertama-tama objek-objek yang paling mirip dikelompokkan, dan kelompok-kelompok awal ini digabungkan sesuai dengan kemiripannya (similaritas). Akhirnya, sewaktu kemiripan berkurang, semua subkelompok digabungkan menjadi satu klaster tunggal. Begitu seterusnya dari hasil similaritas yang tertinggi akan dibandingkan dengan dokumen yang satu dengan dokumen yang lain, sehingga didapat similaritas terendah. Hasil similaritas terendah menyatakan bahwa dokumen tersebut merupakan klaster yang berbeda.

Proses klastering pada penelitian ini digunakan algoritma Clustering Single Pass. Seperti terlihat pada Gambar 5.13 proses klastering akan dilakukan dari hasil output proses hitung similaritas, yaitu nilai similaritas antar dokumen. Proses pertama adalah mencari similaritas tertinggi (maksimal). Dokumen dengan similaritas tertinggi akan menjadi klaster *C1*. Selanjutnya dicari dokumen yang memiliki similaritas diatas threshold.

Proses akan berulang dengan mencari dokumen yang lain untuk dimasukkan ke klaster yang berbeda.



Gambar 5.13 Flowchart Proses Klastering

5.4.4. Perancangan Database Klaster

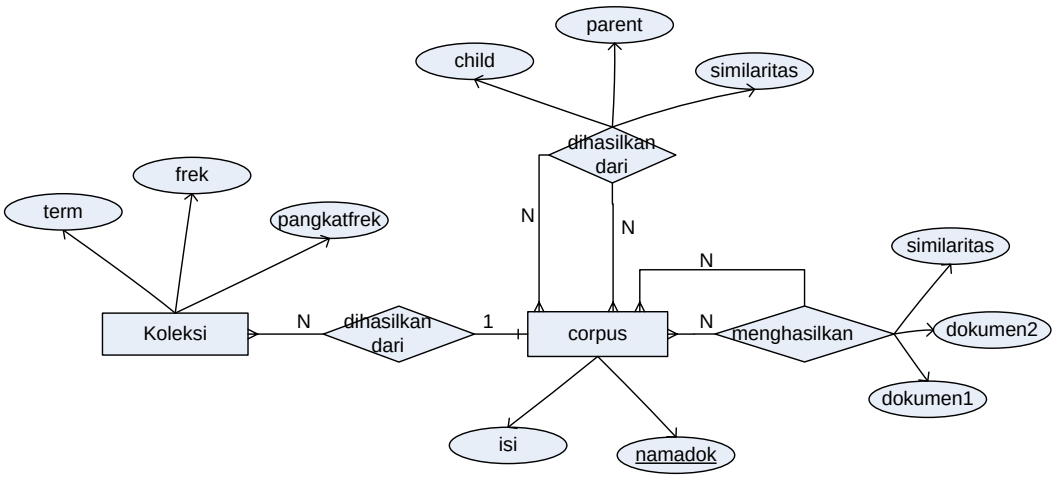
Untuk menjawab pertanyaan tentang pemrosesan data, metode pemodelan data menggunakan ERD (*Entity Relationshyp Diagram*) atau Diagram Hubungan Entitas yang memungkinkan perekayasaan perangkat lunak untuk mengidentifikasi objek data dan hubungannya dengan menggunakan notasi grafis.

Entity Relationshyp Diagram digunakan untuk memudahkan struktur data dan hubungan antar data, karena hal ini relatif kompleks. Dengan *Entity*

Relationshyp Diagram dapat melakukan pengujian model dengan mengabaikan proses yang harus dilakukan.

Dalam rancangan sistem basis data untuk klaster digunakan *Entity Relationshyp Diagram* atau Diagram Hubungan Entitas dan desain tabel untuk menggambarkan atribut-atributnya yang ditunjukkan pada Gambar 5.14.

Terlihat pada gambar 5.14 bahwa entity corpus memiliki hubungan *one to many* (satu ke banyak) dengan entity koleksi, karena satu corpus dapat menghasilkan banyak koleksi dengan atribut namadok sebagai *primary key* untuk relasi. Hal ini juga sama bahwa entity corpus berelasi *recursive,* dengan relasi *many to many* (banyak ke banyak) yang akan menghasilkan nilai similaritas sebagai hasil relasi. Nilai similaritas akan disimpan dalam tabel cosin. Entity corpus juga berelasi *recursive* dengan relasi *many to many* (banyak ke banyak) dengan nilai similaritas akan menghasilkan klaster dokumen. Klaster dokumen akan disimpan dalam tabel klaster.



Gambar 5.14 ER-D Proses Klastering Dokumen

Nilai similaritas yang dihasilkan dari relasi recursive entity corpus akan menghasilkan 2 (dua) tabel yaitu tabel cosin dan tabel klaster. Tabel cosin terdiri dari *field* dokumen1, dokumen2 dan similaritas.

Tabel klaster menyimpan *field* parent, child dan nilai similaritas. Parent adalah dokumen yang memiliki nilai similaritas lebih tinggi dibandingkan dengan dokumen yang lain. Child adalah dokumen yang memiliki nilai similaritas lebih rendah dibandingkan dengan dokumen yang lain.

Berikut adalah transformasi Entity Relationshyp Diagram ke tabel yang digunakan beserta tipe datanya:

1. Tabel stopword yang dibuat tidak ada dalam rancangan *Entity Relationshyp Diagram.* Tabel stop word berisi daftar stop word yang digunakan untuk pengecekan stopword yang ada di korpus. Tabel stop word digunakan untuk pengecekan term hasil proses tokenisasi. Term sebagai stop word dalam tabel disimpan dengan nama *field* term.

Tabel 5.3 Tabel stopword.dbo

Nama Field	**Type Field**	**Keterangan**
Term	varchar(200)	Term stop word

2. Tabel corpus akan menyimpan dokumen berupa data abstrak skripsi yang diambil dari dokumen dalam format file teks. Dokumen abstrak akan tercatat di *field* isi dan nama dokumen tercatat dengan *field* namadok.

Tabel 5.4 Tabel corpus.dbo

Nama Field	Type Field	Keterangan
situs	varchar(200)	nama file dokumen
Isi	Text	Isi dari dokumen

3. Tabel koleksi akan menyimpan term dari masing-masing dokumen, frekuensi tiap term tercatat di *field* frek dan pangkat frekuensi term dicatat dalam *field* pangkatfrek.

Tabel 5.5 Tabel koleksi.dbo

Nama Field	Type Field	Keterangan
Namadok	varchar(50)	Nama file dokumen
Term	varchar(100)	term-term dari dokumen
Frek	Integer	Cacah frekuensi tiap-tiap term
Pangkatfrek	Integer	pangkat tiap-tiap term

4. Tabel cosin berisi daftar term dari setiap dokumen disimpan dalam *filed* term dan similaritas antar dokumen akan disimpan di *field* similaritas.

Tabel 5.6 Tabel cosin.dbo

Nama Field	Type Field	Keterangan
Dokumen1	varchar(150)	nama dokumen pertama yang akan dihitung similaritas dengan dokumen2
Dokumen2	varchar(150)	nama dokumen kedua yang akan dihitung similaritas dengan dokumen1
Similaritas	Float	Nilai similaritas hasil dari hitung similaritas

5. Tabel klaster berisi daftar klaster dari tiap dokumen, nama dokumen dalam tabel klaster tercatat dengan *filed* child dan parent, nilai similaritas antara dokumen dengan klaster sejenis disimpan dengan nama *field* similaritas.

Tabel 5.7 Tabel klaster.dbo

Nama Field	**Type Field**	**Keterangan**
Situs	Text	Dokumen anak
klaster	Text	No klaster

5.4.5. Implementasi Preprosesing

Dalam text preprocessing ada beberapa langkah yang perlu dilakukan untuk mendapatkan teks yang bebas derau (noise) atau bebas kata-kata yang tidak bermakna. Selain membebaskan dari derau, text preprocessing juga mengembalikan kata menjadi kata dasar atau root word.

Langkah-langkah dalam Text preprocessing dalam bahasa Indonesia adalah :

a. Proses Filtering.
b. Proses Tokenizing
c. Proses Stemming.

Proses Filtering diawalai dengan menghilangkan tag html kemudian dilanjutkan hanya melewatkan huruf a sampai z dan spasi. Hal ini dilakukan untuk menghemat waktu ekseskusi setiap dokumennya. Pada Algoritma 5.2 diperlihatkan urutan proses indek dokumen berita secara keseluruhan.

Proses text preprocessing dilakukan pada semua data yang ada, untuk data yang besar dibutuhkan waktu yang lama juga. Pada aplikasi berbasis web masalah akan timbul jika waktu eksekusi lebih dari 30 detik, karena sebagian besar web

server membatasi waktu eksekusi permintaan layanan maksimal 30 detik. Walaupun waktu eksekusi dapat diperlama tetapi capaian proses tidak dapat diinformasikan secara cepat dan akurat oleh sistem, karena biasanya server baru akan memberikan informasi setelah proses 100% selesai. Selain itu memperbesar waktu eksekusi akan menggangu kompaktibiltas dengan web hosting yang ada.

Algoritma 5.2 Keseluruhan proses indek dan pengukuran similaritas

```
// Proses Indek
Untuk Setiap Dokumen {
        Hilangkan Tag_html;
        Hanya gunakan huruf a s/d z serta spasi
        Tokenizing Judul;
        Untuk setiap kata dalam judul dokumen {
                Jika Kata bukan Stopword{
                        Stemming;
                        Simpan dalam master kata;
                        Simpan dalam transaksi Judulkata;
                }
        }
        Tokenizing Abstrak;
        Untuk setiap kata dalam abstrak dokumen {
                Jika Kata bukan Stopword{
                        Stemming;
                        Simpan dalam master kata;
                        Simpan dalam transaksi abskata;
                }
        }
}

// Proses pengukuran
Untuk Setiap Pasangan Dokumen {
        Ukur jarak cosine (dokumen1,dokumen2);
        Simpan dalam tabel cosine;
}
```

Untuk mengatasi hal tersebut maka proses text preprocessing dilakukan tiap satu persatu dokumen, dengan demikian hanya 1 dokumen yang akan diproses setiap waktunya oleh web server setiap kali url program di muat / dipanggil. Kemudian menggunakan mekanisme variabel session untuk menyimpan data

pointer posisi dokumen terakhir diproses. Sehingga setiap kali url dimuat maka pointer akan bergeser ke dokumen selanjutnya sampai pointer menunjuk pada dokumen terakhir.

Mekanisme pemanggilan / pemuatan ulang url program secara otomatis dapat menggunakan bantuan javascript autoreload. Setiap kali script autoreload dipanggil maka browser secara otomatis memanggil / memuat ulang halaman tersebut, demikian seterusnya sampai semua dokumen selesai diproses.

Pada algoritma 5.3 diperlihatkan pseudo code proses implementasi mekanisme autoreload text preprocessing pada aplikasi berbasis web, dimana variabel $_SESSION['id'] digunakan untuk menyimpan array id jurnal, variabel $_SESSION['dok-POS'] menyimpan pointer posisi dokumen dan variable $_SESSION['dok-MAX'] untuk menyimpan jumlah dari dokumen berita yang diproses.

Algoritma 5.3 Pseudo code proses implementasi mekanisme autoreload untuk memuat proses text preprocessing

```
session_start();
$_SESSION[dok-pos]:=0;
$_SESSION[id];= array of idjurnal;
$_SESSION[dok-max];=sizeof($id);
number_of_record(data)
<iframe autoreload >
        $pos=$_SESSION[pos];
        $max=$_SESSION[max];
        $id=$_SESSION[id];
        while (($pos<$max) and ($pos<$pos+5)) {
                TextPreprocessing($id[$pos]);
                $pos++;
        } else {
                Halt("Proses Selesai");

        }
        $_SESSION[pos]:=$pos;
        echo "<script type='text/javascript'> window.onload=
        setTimeout('window.location.reload()',1) ;</script>";
</iframe>
```

Pada pseudo code diperlihatkan setiap pemanggilan program hanya akan diproses dokumen sebanyak 5 buah saja, pembatasan ini untuk memotong proses menjadi lebih kecil. Setelah program selesai dijalankan, program akan dipanggil ulang oleh javascript autoreload pada bagian bawah.

Kemudian pada algoritma 5.4 diperlihatkan pseudo code proses implementasi mekanisme autoreload pengukuran similaritas pada aplikasi berbasis web, dimana variabel $_SESSION['id'] digunakan untuk menyimpan array id jurnal, variabel $_SESSION['dok-POS'] menyimpan pointer posisi dokumen dan variable $_SESSION['dok-MAX'] untuk menyimpan jumlah dari dokumen berita yang diproses. Kemudian pada proses pengukuran similaritas digunakan tambahan variabel disamping tiga variabel di atas, yaitu variabel $_SESSION['ids'] berisi array dari jurnalid yang akan diukur similaritasnya, variabel $_SESSION['sim-MAX'] berisi jumlah dokumen berita yang akan diukur, variabel $_SESSION['sim-POS'] berisi pointer dokumen yang diproses dan variabel $_SESSION['sim-POS1'] berisi pointer dokumen 2 yang diukur similaritasnya dengan dokumen yang pertama.

Pada pseudo code diperlihatkan setiap pemanggilan program diproses sebanyak 100 kali pengukuran similaritas, pembatasan ini untuk memotong proses menjadi lebih kecil. Setelah program selesai dijalankan, program akan dipanggil ulang oleh javascript autoreload pada bagian bawah. Variabel count digunakan untuk membatasi proses sebanyak 100 kali.

Perlu diperhatikan bahwa untuk menjalankan program ini, javascript dan cookies pada browser komputer klien harus diaktifkan. Javascript digunakan

untuk mengeksekusi mekanisme autoreload dan cookies untuk menyimpan variabel session di komputer lokal.

Algoritma 5.4 Pseudo code proses implementasi mekanisme autoreload untuk memuat proses pengukuran similaritas

```
session_start();
$_SESSION[sim-pos]:=0;
$_SESSION[id];= array of idjurnal;
$_SESSION[max];=sizeof($id);
$_SESSION[sim-pos1]:=0;
$_SESSION[ids];= array of idjurnal;
$_SESSION[sim-max];=sizeof($ids);
number_of_record(data)
<iframe autoreload >
        $pos=$_SESSION[sim-pos];
        $max=$_SESSION[max];
        $id=$_SESSION[id];
        $pos1=$_SESSION[sim-pos1];
        $maxp=$_SESSION[sim-max];
        $ids=$_SESSION[ids];
        $count=0;
        if(($pos<$max) and ($count<100)){
                if($pos1<$maxp){
                        ukurcosine($id[$pos],$ids[$pos1]);
                        $pos1++;
                } else {
                        $pos1=0;
                        $pos++;
                }
                $count++;
        } else {
                Halt("Proses Selesai");
        }
        $_SESSION[pos]:=$pos;
        $_SESSION[pos1]:=$pos1;
        echo "<script type='text/javascript'> window.onload=
        setTimeout('window.location.reload()',1) ;</script>";
</iframe>
```

1. Implementasi Text Filtering

Sebelum kata dipisahkan dari kalimatnya, terlebih dahulu dibersihkan dari tanda baca, tag html dan angka. Untuk membersihkan dapat digunakan perintah ekspresi regular yang ada pada bahasa pemrograman PHP. Pembersihan dilakukan

sebelum proses tokenizing dimaksudkan untuk memperkecil hasil dari tokenizing. Dengan demikian diharapkan keluaran dari tokenizing berupa kata-kata yang bersih dari tanda baca, tag html dan angka.

Proses pembersihan tanda baca dan angka diperlihatkan pada pseudo code pada algoritma 5.5

Algoritma 5.5 Pseudo code proses pembersihan tanda baca dan angka

```
$tmp = "";
$str=trim($str);
while (ereg("<(/?[[:alpha:]]*)[[:space:]]*([^>]*)>",$str,$reg)) {
        $i = strpos($str,$reg[0]);
        $l = strlen($reg[0]);
        $tag = "";
        $tmp .= substr($str,0,$i) . $tag;
        $str = substr($str,$i+$l);
}
$str = $tmp . $str;
$str=eregi_replace("[^a-z]"," ",$str);
return $str;
```

2. Implementasi Text Tokenizing

Pada kalimat, pemisah antar kata adalah karakter spasi. Sehingga proses deteksi token dapat dilakukan dengan melihat keberadaan karakter spasi. Proses deteksi dapat menggunakan perulangan melakukan pembacaan setiap karakter. Tetapi pada pemrograman PHP terdapat perintah untuk mengubah string menjadi array dengan pemisah karakter tertentu, yaitu perintah *explode([separator],[teks]).* Dengan mengisi [teks] dengan variabel string dan [separator] diisi dengan karakter spasi, maka setelah perintah dieksekusi, semua kata akan terpisah dari string dan tersusun dalam suatu array.

Setelah token dideteksi maka array hasil dari deteksi tersebut diolah oleh proses berikutnya. Pemrosesan pada proses berikutnya dilakukan kata-perkata untuk meringankan proses.

3. Implementasi Deteksi Stopword

Proses deteksi stopword dimaksudkan untuk mengetahui suatu kata masuk ke dalam stop word atau tidak. Apabila sebuah kata masuk di dalam daftar stopword maka kata tersebut tidak akan diproses lebih lanjut. Sebaliknya apabila sebuah kata tidak termasuk di dalam daftar stopword maka kata tersebut akan masuk keproses berikutnya.

Daftar Stopword tersimpan dalam suatu tabel, dalam penelitian ini menggunakan daftar stop word yang digunakan oleh Tala, 2003. Untuk mempercepat proses deteksi, terlebih dahulu tabel stopword di muat kedalam array memori, kemudian dilakukan proses perbandingan kata yang akan dideteksi dengan array stopword. Pada pemrograman PHP proses pembandingan kata dengan array dapat dilakukan dengan 1 perintah *in_array($str,$Stopword).* Perintah tersebut akan menghasilkan nilai True jika kata pada variabel $str terdapat dalam salah satu elemen array $Stopword.

4. Implementasi Stemming Tala

Proses stemming adalah bagian dari proses filtering, tujuan utama dari proses stemming adalah mengembalikan kata dalam bentuk dasarnya. Struktur pembentukan kata dalam Bahasa Indonesia adalah sebagai berikut:

[awalan-1] + [awalan-2] + dasar + [akhiran] + [kepunyaan] + [sandang]

Masing-masing bagian tersebut (yang dalam kotak bisa ada atau tidak), digabungkan dengan kata dasar membentuk kata berimbuhan.

Penggunaan algoritma stemming Tala bertujuan untuk mempercepat waktu implementasi dan diharapkan performa yang stabil walaupun data dokumen bertambah terus. Algoritma Tala menggunakan algoritma rule based stemming seperti halnya dengan algoritma porter pada stemming bahasa Inggris.

Pada stemmer Tala terdapat 5 langkah utama dengan 3 langkah awal dan 2 langkah pilihan, langkah-langkah tersebut sbb:

a. Menghilangkan partikel
b. Menghilangkan kata sandang dan kepunyaan.
c. Menghilangkan awalan 1
d. Jika suatu aturan terpenuhi jalankan sbb :
    - Hilangkan Akhiran
    - Jika suatu aturan terpenuhi, hilangkan awalan 2. Jika tidak proses stemming selesai
e. Jika tidak ada aturan yang terpenuhi jalankan sbb :
    - Hilangkan awalan 2.
    - Hilangkan Akhiran
    - Proses stemming selesai.

Selain itu tala juga membagi imbuhan menjadi 5 kluster yang nantinya digunakan untuk menghilangkan imbuhan pada setiap tahapnya.

a. Proses menghilangkan partikel

Pada proses ini dokumen dibersihkan dari partikel / tanda baca. Selain tanda baca dalam proses ini juga dihilangkan semua angka serta kata-kata yang tidak bermakna (stopword). Stopword yang diketahui disimpan dalam tabel basis data stopword kemudian untuk semua kata yang ada dalam tabel tersebut akan dihilangkan. Isi tabel basis data stopword diambil dari daftar stopword Tala (Tala 2003).

Masukan untuk proses stemming adalah kata hasil dari tokenizing. Tanda baca dan angka sudah dihilangkan sebelum dilakukan tokenizing. Kemudian data stopword tersimpan dalam tabel basis data, proses menghilangkan stopword akan lebih cepat dilakukan sekaligus melalui perintah Query. Sehingga Stopword akan dihilangkan setelah proses stemming selesai dilaksanakan pada semua dokumen. Proses menghilangkan stopword dibahas pada pemrosesan indek jurnal.

b. Proses menghilangkan kata sandang dan kepunyaan

Pada proses ini dokumen melalui perlakuan untuk menghilangkan kata sandang dan kepunyaan. Proses ini dibagi dalam 2 cluster proses yang harus diproses secara urut. Algoritma 5.6 adalah algoritma yang digunakan untuk menghilangkan kata sandang dan kepunyaan.

Algoritma 5.6 Pseudo code untuk menghilangkan kata sandang

```
// Aturan cluster 1
$str= ganti("lah "," ") pada $str;
$str= ganti("kah "," ") pada $str;
$str= ganti("pun "," " ) pada $str;
// Aturan cluster 2
$str= ganti("nya "," ") pada $str;
$str= ganti("ku "," ") pada $str;
$str= ganti("mu "," " ) pada $str;
```

c. Menghilangkan awalan 1

Pada proses ini dokumen melalui perlakuan untuk menghilangkan awalan, stemmer Tala melokalisasi awalan 1 dalam 1 cluster proses yang harus diproses secara urut. Algoritma 5.7 adalah algoritma yang digunakan untuk menghilangkan awalan 1.

Algoritma 5.7 Algoritma untuk menghilangkan awalan 1.

```
// Aturan cluster 3
$str= ganti(" meng'',"  ") pada $str;
$str= ganti(" menya"," s") pada $str;
$str= ganti(" menyi"," s") pada $str;
$str= ganti(" menyu"," s") pada $str;
$str= ganti(" menye"," s) pada $str;
$str= ganti(" menyo"," s") pada $str;
$str= ganti(" meny"," s") pada $str;
$str= ganti(‘‘ men","  " ) pada $str;
$str= ganti(" mema"," p") pada $str;
$str= ganti(" memi"," p") pada $str;
$str= ganti(" memu"," p) pada $str;
$str= ganti(" meme"," p) pada $str;
$str= ganti(" memo"," p) pada $str;
$str= ganti(" mem","  " ) pada $str;
$str= ganti(" me","  " ) pada $str;
$str= ganti(" peng","  " ) pada $str;
$str= ganti(" penya"," s") pada $str;
$str= ganti(" penyi"," s") pada $str;
$str= ganti(" penyu"," s") pada $str;
$str= ganti(" penye"," s) pada $str;
$str= ganti(" penyo"," s") pada $str;
$str= ganti(" peny"," s) pada $str;
$str= ganti(" pen","  " ) pada $str;
$str= ganti(" pema"," p") pada $str;
$str= ganti(" pemi"," p") pada $str;
$str= ganti(" pemu"," p) pada $str;
$str= ganti(" peme"," p) pada $str;
$str= ganti(" pemo"," p) pada $str;
$str= ganti(" pem","  " ) pada $str;
$str= ganti(" di","  ") pada $str;
$str= ganti(" ter","  " ) pada $str;
$str= ganti(" ke","  " ) pada $str;
```

d. Menghilangkan awalan 2.

Pada proses ini dokumen melalui perlakuan untuk menghilangkan awalan, stemmer Tala melokalisasi awalan 2 dalam 1 cluster proses yang harus diproses secara urut. Algoritma 5.8 adalah Algoritma yang digunakan untuk menghilangkan awalan 2.

Algoritma 5.8 Algoritma untuk menghilangkan awalan 2.

```
// Aturan cluster 4
$str= ganti(" ber"," ") pada $str;
$str= ganti(" bel"," ") pada $str;
$str= ganti(" be"," ") pada $str;
$str= ganti(" per"," " ) pada $str;
$str= ganti(" pel"," ") pada $str;
$str= ganti(" pe"," " ) pada $str;
```

e. Menghilangkan akhiran.

Pada proses ini dokumen melalui perlakuan untuk menghilangkan awalan, stemmer Tala melokalisasi akhiran dalam 1 cluster proses yang harus diproses secara urut. Algoritma 5.9 adalah algoritma yang digunakan untuk menghilangkan akhiran.

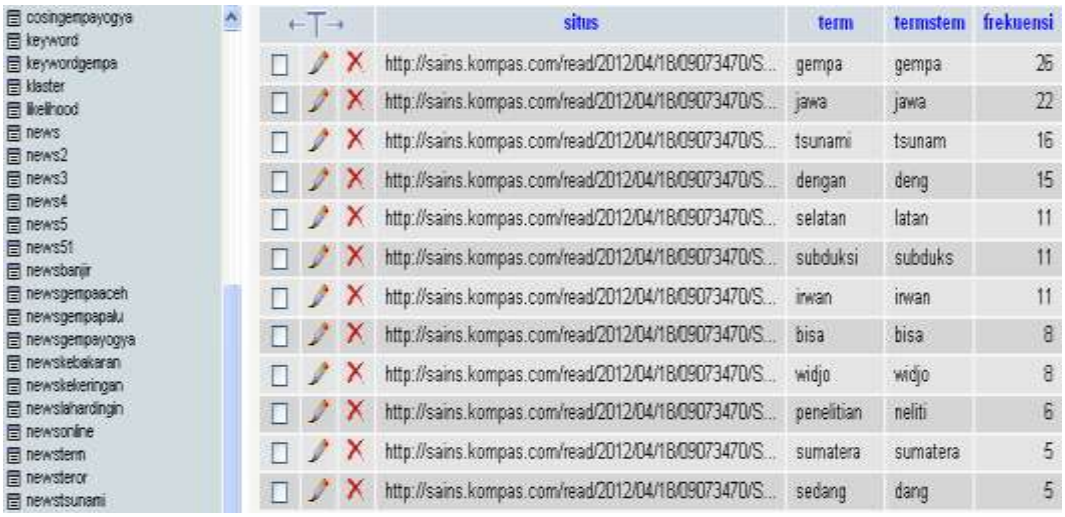
Algoritma 5.9 Algoritma untuk menghilangkan akhiran.

```
// Aturan cluster 5
$str= ganti("kan "," ") pada $str;
$str= ganti("an "," ") pada $str;
$str= ganti("i "," ") pada $str;
```

Setelah 5 tahap dilalui maka kata sudah dianggap telah menjadi root atau kata dasar. Menurut Tala kata dasar pada bahasa Indonesia terdiri paling sedikit 2 kata, sehingga sebelum dilakukan penggantian / penghilangan awalan, akhiran ataupun partikel diperhatikan panjang huruf yang tersisa. Jumlah huruf yang akan diproses minimal 2 + (panjang imbuhan yang akan dihilangkan) + 2 (spasi, untuk depan dan belakang kata).

f. Implementasi Proses Indek

Setelah kata telah dikembalikan dalam bentuk asal (kata dasar), kata-kata tersebut disimpan kedalam tabel basis data. Proses indek menggunakan metode Inverse Document Frequency (IDF) dimana setiap kata akan diketahui frekuensi kemunculan di seluruh koleksi dokumen dan setiap dokumen. Tabel proses indeks dapat dilihat pada gambar 5.15. Pada gambar 5.15 dalam tabel disimpan alamat situs, term yang muncul, term yang telah distemming dan frekuensi kemunculan dari masing-masing term.

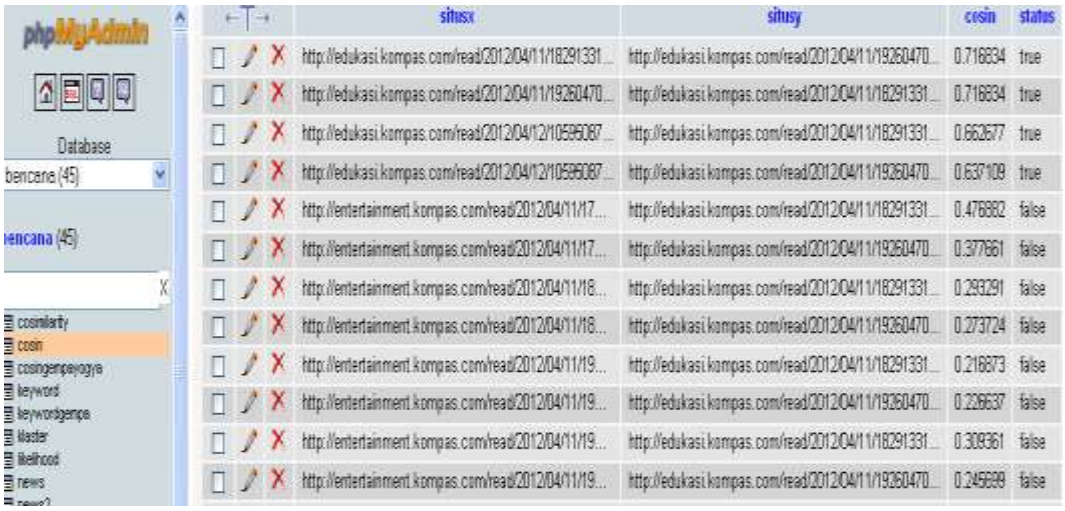


	situs	term	termstem	frekuensi
	http://sains.kompas.com/read/2012/04/18/09073470/S...	gempa	gempa	26
	http://sains.kompas.com/read/2012/04/18/09073470/S...	jawa	jawa	22
	http://sains.kompas.com/read/2012/04/18/09073470/S...	tsunami	tsunam	16
	http://sains.kompas.com/read/2012/04/18/09073470/S...	dengan	deng	15
	http://sains.kompas.com/read/2012/04/18/09073470/S...	selatan	latan	11
	http://sains.kompas.com/read/2012/04/18/09073470/S...	subduksi	subduks	11
	http://sains.kompas.com/read/2012/04/18/09073470/S...	irwan	irwan	11
	http://sains.kompas.com/read/2012/04/18/09073470/S...	bisa	bisa	8
	http://sains.kompas.com/read/2012/04/18/09073470/S...	widjo	widjo	8
	http://sains.kompas.com/read/2012/04/18/09073470/S...	penelitian	neliti	6
	http://sains.kompas.com/read/2012/04/18/09073470/S...	sumatera	sumatera	5
	http://sains.kompas.com/read/2012/04/18/09073470/S...	sedang	dang	5

Gambar. 5.15 Gambar Tabel Hasil Proses Indeks

g. Implementasi algoritma cosine coefficient

Pada implementasi cosine coefisien adalah melakukan proses perhitungan simmilaritas antar dokumen yang satu dengan dokumen yang lain. Pada Gambar 5.16 terlihat tabel hasil proses hitung similaritas antar dokumen. Dalam tabel disimpan alamat dokumen x dan dihitung similaritas dengan dokumen y. Hasil hitung similaritas disimpan dalam kolom cosin.



Gambar 5.16 Gambar Tabel Hasil Proses Cosine Coefficient

6. Modul klastering

Klaster dokumen akan diproses dari nilai similaritas yang dihasilkan oleh *method* Cosine(). Seperti pada Gambar 5.16 class Klaster akan melakukan proses klaster dengan *program prosesklaster.php*.

Nilai similaritas hasil dari proses hitung similaritas akan digunakan untuk melakukan proses klastering. Proses klastering dilakukan dengan mencari nilai similaritas yang maksimum dari table cosin. Implementasi dalam SQL adalah sbb:

> *SELECT situsx,situsy,cosin FROM `cosin` where status = 'false' order by cosin desc limit 1*

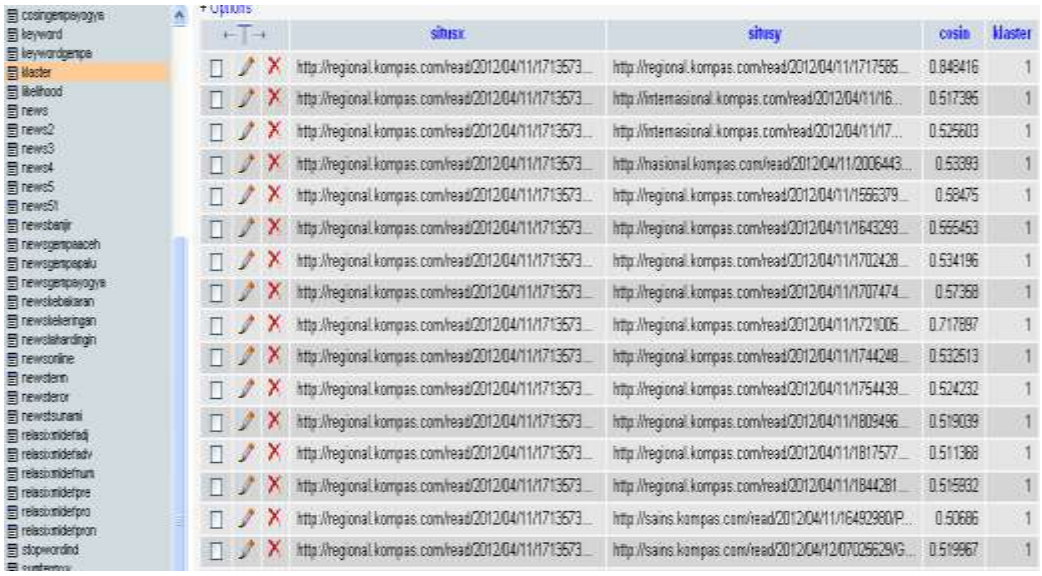
Hasil dari SQL adalah nilai similaritas yang telah dirangking dari nilai similaritas maksimum. Maka dokumen satu pasang situs ini menjadi kandidat dari klaster pertama (*C1).*

Tahap kedua proses klastering adalah mencari kadindat yang lain dalam klaster *C1* dengan mencari nilai similaritas maksimum dari satu pasang dokumen.

SELECT situsx,situsy,cosin FROM `cosin` where (status = 'false') and (situsx ='$situs1') and (cosin >=0.5)
go

Hasil dari SQL adalah dokumen dengan nilai similaritas yang lebih besar dari nilai threshold akan masuk dalam satu klaster *C1*.

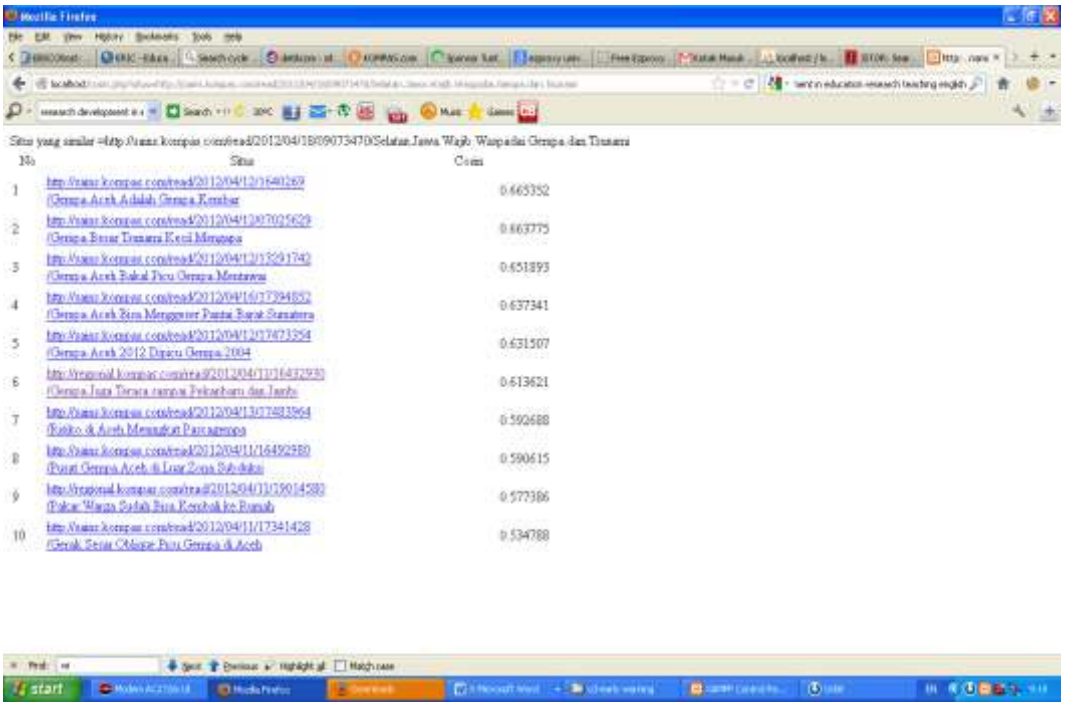
Proses akan dilanjutkan dengan mencari kandidat untuk klaster yang lain. Implementasi proses klaster akan dilakukan tahap yang sama mulai tahap kedua sampai dokumen dalam korpus habis dibuat klaster. Tabel hasil proses klaster dapat dilihat pada gambar 5.17



Gambar 5.17 Gambar Tabel Hasil Proses Klatering

Dari tabel klastering yang telah dihasilkan proses selanjutnya dibuat tampilan untuk user interface. Tampilan hasil klaster dibuat dengan dengan menggunakan PHP seperti terlihat pada Gambar 5.18 dan Gambar 5.19 menunjukkan hasil similaritas dengan dokumen lain jika salah satu klaster pada Gambar 5.18 diklik (dipilih).

Gambar 5. 18 Tampilan Klaster Hierarki



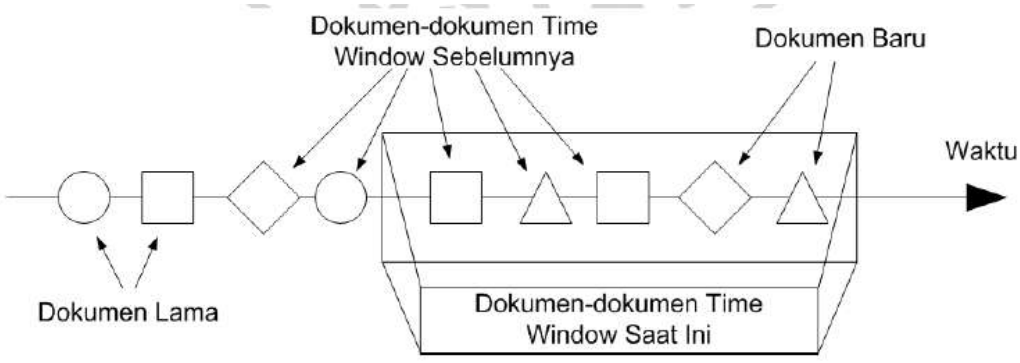
Gambar 5.19 Gambar Tampilan Cosine

5.5. TOPIC DETECTION AND TRACKING

Topic Detection dan Tracking digunakan untuk mendeteksi kemunculan topik-topik baru dan menelusuri kemunculan ulang dan evolusi dari topik-topik tersebut. Pada studi ini, metode yang diteliti adalah teknik penanganan dokumen-dokumen pada aliran berita dan pendeteksian topik berita.

5.5.1 *Time Window*

Time window adalah suatu metode yang digunakan pada studi TDT untuk menangani aliran berita. *Time window* memandang sebuah aliran dokumen berita melalui "jendela" dengan jumlah dokumen atau interval waktu tertentu.



Gambar 5.20 Ilustrasi Time Window pada Aliran Dokumen Berita

Penggunaan *time window* menyebabkan koleksi dokumen yang diacu berisi dokumen-dokumen pada jendela tersebut saja. Dengan sifat aliran berita yang selalu berubah setiap waktu, koleksi dokumen harus selalu diperbaharui seiring dengan datangnya dokumen-dokumen berita baru. Pembaharuan koleksi dokumen tersebut dilakukan dengan menambakan dokumen berita baru dan membuang dokumen yang dianggap telah kadaluarsa. Proses ini diilustrasikan pada Gambar 5.20

Secara umum ada dua cara dalam memperbaharui koleksi dokumen pada *time window*, yaitu secara langsung dan secara tertunda. Pembaharuan koleksi dokumen secara langsung memperbaharui koleksi dokumen segera setelah dokumen baru datang. Cara pembaharuan koleksi ini dapat menyebabkan pemrosesan yang sangat banyak, karena harus dilakukan setiap saat dokumen baru datang.

Pembaharuan koleksi dokumen secara tertuda adalah pembaharuan koleksi dokumen yang dilakukan dengan menunda pembaharuan sampai waktu atau jumlah dokumen tertentu. Sebagai contoh, pembaharuan koleksi dokumen dapat dilakukan dengan mengumpulkan dokumen-dokumen dalam interval waktu tiap satu jam. Cara pembaharuan koleksi dokumen ini efektif pada keadaan dimana kedatangan dokumen baru hanya diselingi oleh waktu yang singkat.

5.5.2 Pendeteksian Topik Baru

Pendeteksian topik baru atau *First Story Detection* (FSD) atau *New Event Detection* adalah metode yang digunakan pada studi TDT untuk mengidentifikasi kejadian atau event baru dari aliran dokumen berita dan menentukan apakah berita pada dokumen tersebut adalah berita baru atau kelanjutan dari berita yang telah ada sebelumnya.

Pendeteksian topik baru terdiri dari dua bagian: *retrospective detection* dan *on-line detection*. *Restrospective detection* adalah penelusuran dokumen-dokumen berita masa lampau untuk mendapatkan topik yang sebelumnya belum

teridentifikasi. *On-line detection* adalah usaha untuk mengidentifikasi awal dari topik baru dari aliran berita secara *real-time*.

Dalam pendeteksian topik, digunakan pendekatan konvensional dengan *vector space model* untuk merepresentasikan dokumen-dokumen berita. Sebuah dokumen berita direpresentasikan sebagai sebuah vektor dengan *term* sebagai dimensinya.

5.5.3. Klasifikasi Kategori

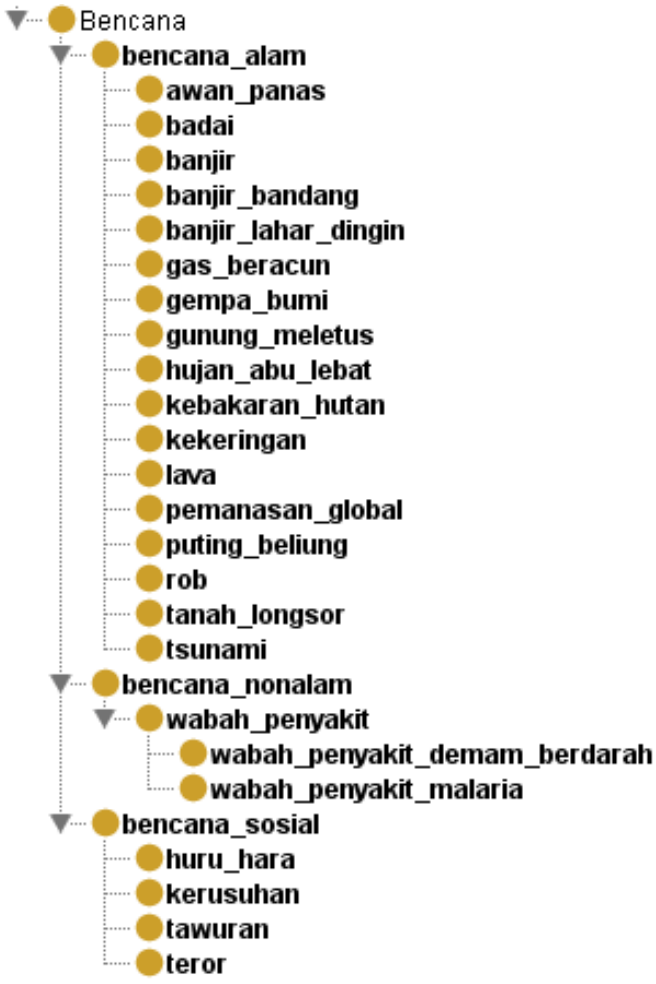
Klasifikasi kategori, untuk berita, adalah sebuah masalah klasifikasi teks multi-label. Tujuannya adalah untuk memberikan satu atau lebih kategori ke sebuah artikel berita. Tekhnik standar dalam klasifikasi teks mutli-label adalah menggunakan himpunan pengklasifikasi biner. Untuk tiap kategori, sebuah pengklasifikasi digunakan untuk memberikan jawaban “ya” atau “tidak” pada sebuah kategori yang diberikan ke sebuah teks.

Klasifikasi kategori berkaitan dengan pemberian satu atau lebih label kategori ke artikel-artikel berita. Dalam penelitian ini kategori bencana didasarkan pada UU No. 24 tahun 2007 tentang Penanggulangan Bencana ada 3 (tiga) jenis bencana yaitu :

1. **Bencana Alam** adalah bencana yang diakibatkan oleh peristiwa atau serangkaian peristiwa yang disebabkan oleh alam, antara lain berupa gempa bumi, tsunami, gunung meletus, banjir, kekeringan, angin topan, dan tanah longsor.

2. **Bencana non alam** bencana yang diakibatkat oleh peristiwa atau serangkaian peristiwa non alam yang antara lain berupa kegagalan teknologi, gagal modernisasi, epidemic dan wabah penyakit.
3. **Bencana sosial** adalah bencana yang diakibatkan oleh peristiwa atau serangkaian peristiwa yang diakibatkan manusia yang meliputi konflik social antar kelompok dan antar komunitas masyarakat serta terror.

Kombinasi topik dan kategori akan membuat struktur hirarki. Sebagai contoh hirarki untuk bencana dapat dilihat pada Gambar 5.21 berikut ini:



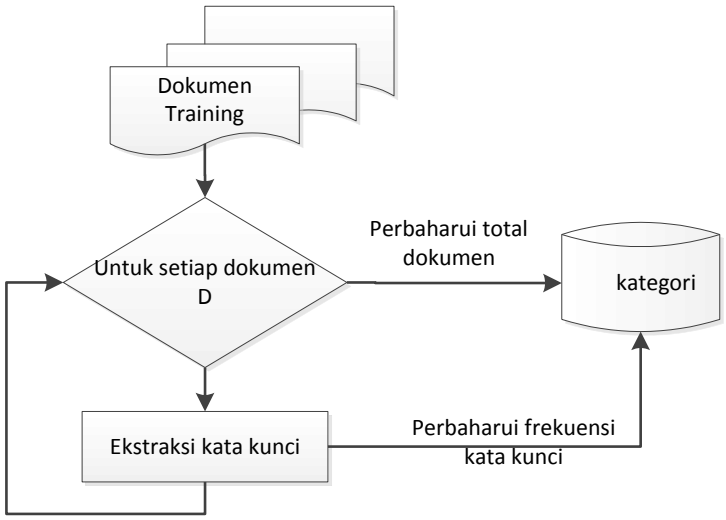
Gambar 5.21 Kategori bencana berdasarkan UU No. 24 tahun 2007

Algoritma klasifikasi untuk berita, selain diinginkan mempunyai presisi dan recall yang tinggi juga mudah diperbaharui. Hal ini dikarenakan perubahan dunia yang sangat cepat sehingga dimungkinkan munculnya kategori yang baru. Mudah diperbaharui, dimaksudkan bahwa pengklasifikasi membutuhkan cara pengujian ulang yang sederhana (tidak perlu melakukan training secara keseluruhan) atau tidak perlu dilakukan pengujian secara keseluruhan. Hal ini dikarenakan jumlah berita yang sangat banyak sehingga kalau dilakukan pengulangan pelatihan data akan menimbulkan masalah. Untuk itu diperlukan suatu algoritma yang tidak memerlukan pelatihan sebelumnya.

Langkah pertama sebelum melakukan klasifikasi kategori adalah penentuan kategori primitif. Kategori primitif merupakan kategori yang telah ditentukan sebelumnya. Dalam penulisan penelitian ini, kategori yang digunakan merupakan hasil pengamatan terhadap kategori – kategori yang terdapat pada situs berita seperti www.kompas.com, www.antaranews.com dan www.tempo.co.id Klasifikasi kategori dibagi menjadi dua proses besar: proses *training* dan proses klasifikasi.

Training

Pada proses *training*, model kategori untuk setiap kategori dibangun. Model ini berisi nama kategori, jumlah dokumen dan daftar *keywords* (kata kunci). Untuk melatih sebuah *classifier* untuk kategori, diperlukan sekumpulan dokumen *training*. Sebuah metode otomatis untuk mendapatkan himpunan training dibuat, hal ini mencakup pembuatan korpus domain. Dari artikel-artikel ini kata kunci diekstrak dan disimpan.



Gambar 5.22 Menjelaskan Alur Kerja Proses *Training*.

Pada gambar 5.22 diperlihatkan alur kerja dari proses training. Berbeda dengan algoritma *training* pada umumnya, proses *training* hanya berfungsi untuk mengambil kata kunci, tanpa mengambil parameter – parameter tertentu.

Setiap kali sebuah artikel ditambahkan sebagai data pelatihan, "banyaknya dokumen" untuk suatu kategori diperbaharui. Banyaknya dokumen memberitahukan berapa banyak dokumen pelatihan yang telah dilihat untuk kategori ini.

Kata kunci diekstrak dari artikel. Tiap kata kunci dicari dalam himpunan kata kunci suatu kategori. Jika kata kunci ditemukan maka himpunan kata kunci diperbaharui, dengan menambah banyaknya kata kunci di "In-Document". Jika kata kunci tidak ditemukan maka kata kunci ditambahkan ke himpunan kata kunci dengan nilai awal banyak kata kunci di "In-Document" adalah 1.

Pembuatan model kategori dengan cara ini membolehkan model untuk mudah di perbaharui. Probabilitas dari kata kunci ke berapa yang diberikan ke

sebuah kategori dapat secara mudah di perbaharui menggunakan banyaknya kata kunci “In-Document” dan banyaknya “total number of documents.

Algoritma 5.10 Algoritma proses training

```
$result = mysql_query("SELECT alamat,judul,isi FROM  `newsgempayogya` ");
while($row = mysql_fetch_array($result))
  {  $kategori='gempa Yogyakarta';
$situs=$row['alamat'];  $string = $row['isi'];
$word = str_word_count(strtolower($string ),1);
$jumlah = count($word);  $word_count = array_count_values($word);
arsort($word_count);
$sql2="insert into totaldokumen (situs,kategori) values ('$situs','$kategori')";
$sql2="update keyword set counter=counter+1
where term in (SELECT term  FROM newsterm  WHERE situs = '$situs')";
$sql2="insert  into `keyword` (term,termstem,kategori,counter)
(select distinct term,termstem,'$kategori',1 from newsterm where (situs='$situs') and
(term not in (SELECT term FROM keyword)))";
}
```

Dari algoritma 5.10 akan dihasilkan dokumen training bencana. Hasil proses traning akan disimpan dalam tabel training dan dihasilkan tabel kategori seperti terlihat pada tabel 5.23


Sort by key: None
+ Options

term	id	status	counter	kategori	termstem
gempa	80		146	gempa Aceh	gempa
aceh	512		129	gempa Aceh	aceh
rabu	589		117	gempa Aceh	rabu
tsunami	296		93	gempa Aceh	tsunam
berkekuatan	36		83	gempa Aceh	kuat
warga	308		82	gempa Aceh	warga
-	1		80	gempa Aceh	-
skala	260		77	gempa Aceh	skala
richter	231		75	gempa Aceh	richter
barat	27		75	gempa Aceh	barat
jakarta	96		72	gempa Aceh	jakarta
sumatera	266		70	gempa Aceh	sumatera
badan	17		65	gempa Aceh	badan
mengguncang	168		62	gempa Aceh	guncang
tersebut	287		59	gempa Aceh	but
wilayah	310		56	gempa Aceh	wilayah
pusat	588		55	gempa Aceh	pusat
sekitar	593		55	gempa Aceh	kitar
sr	1677		55	gempa Aceh	sr
dalam	46		53	gempa Aceh	dalam
seperti	254		50	gempa Aceh	rti
sementara	251		49	gempa Aceh	ttara
wib	1638		48	gempa Aceh	wib
daerah	532		47	gempa Aceh	daerah

Gambar 5.23 Gambar Tabel Katagori

Proses berikutnya adalah klasifikasi, yaitu penetapan kategori untuk dokumen uji yang diujikan pada aplikasi. Proses klasifikasi meliputi empat tahap besar:

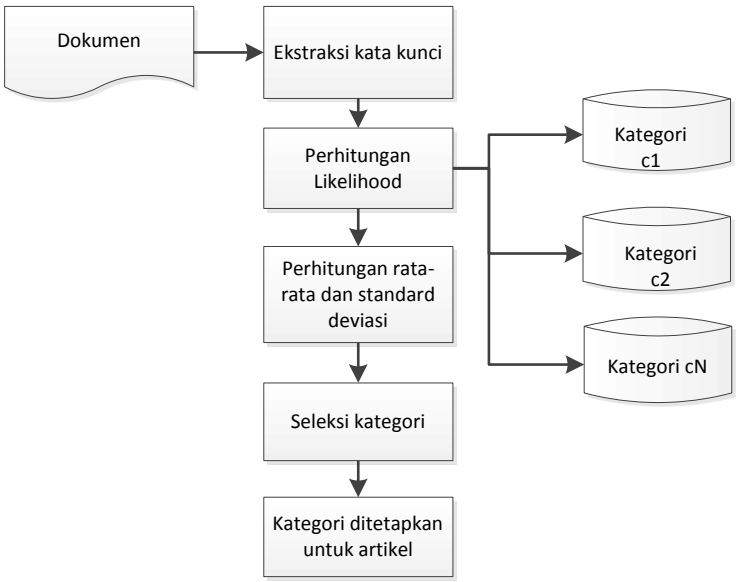
1. Ekstraksi kata kunci dokumen uji
2. Perhitungan *likelihood*
3. Perhitungan rata – rata dan standard deviasi
4. Seleksi kategori

Ekstraksi kata kunci untuk dokumen uji menggunakan algoritma yang sama dengan dokumen *training*. Perhitungan *likelihood* untuk sebuah kategori dijelaskan pada rumus 5.1. Dalam persamaan tersebut, cj adalah kategori, A adalah artikel dokumen uji, dan $P(k_i|c_j)$ dihitung menggunakan "*In-Document*" dan perhitungan "jumlah total dokumen".

$$Likelihood(c_j \mid A = \{k_1, k_2, ..., k_n\}) = -\sum_{i=1}^{n} P(k_i \mid c_i) \log(P(k_i \mid c_j)) \quad (5.1)$$

Setelah seluruh *likelihood* untuk semua kategori telah dihitung, nilai ambang batas bisa didapatkan. Nilai ambang (*threshold*), seperti yang ditunjukkan pada rumus 5.2, berguna untuk menentukan apakah sebuah kategori bisa ditetapkan untuk artikel uji atau tidak. Nilai ini didapatkan dari standar deviasi dan rata – rata. *L* adalah jumlah banyaknya *likelihood*, sementara l_i adalah *likelihood* untuk kategori ke – *i*. Asumsinya adalah kategori – kategori yang tepat akan memiliki nilai yang besarnya jauh berbeda dibandingkan kategori – kategori lainnya. Secara formal, klasifikasi kategori dijelaskan pada Gambar 5.24

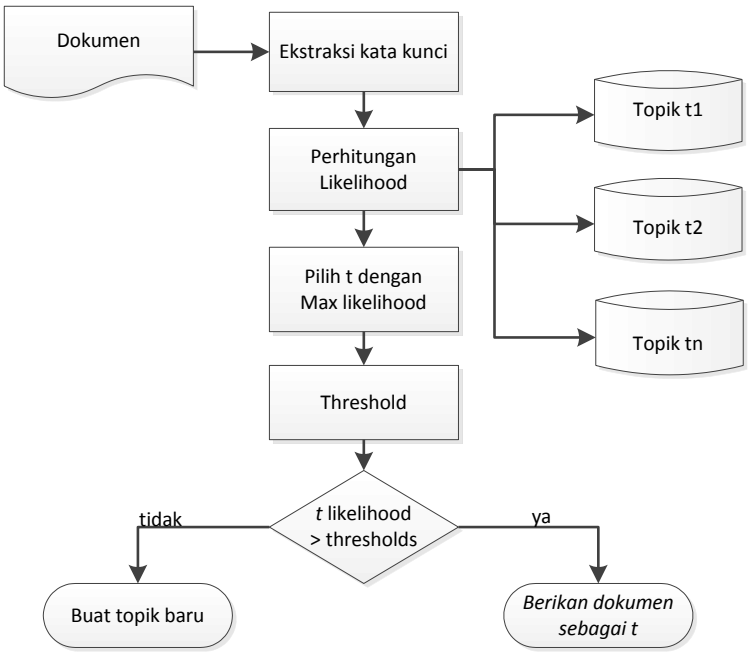
$$Threshold = \frac{\sum_1^{|L|} l_i}{|L|} + \sqrt{\frac{\sum (l_i - \frac{\sum_1^{|L|} l_i}{|L|})^2}{|L|}} \quad (5.2)$$



Gambar 5.24 Alur Proses Katagori

Identifikasi Topik

Algoritma identifikasi topik tidak memerlukan *corpus training* khusus untuk melatih aplikasi terlebih dahulu. Topik baru muncul setiap hari, oleh karena itu, diperlukan sebuah algoritma yang dapat mengetahui apakah topik baru harus ditentukan untuk artikel tersebut. Algoritma identifikasi topik dibagi menjadi dua proses besar, yaitu klasifikasi dan *dynamic thresholding*. Konsep dasar identifikasi topik dijelaskan pada Gambar 5.25.



Gambar 5.25 Flowchart Proses Identifikasi Topik

Algoritma penemuan topic dan klasifikasi menggunakan klastering single-pass untuk menentukan topic dari sebuah artikel. Klasifikasi dilakukan dengan menemukan topic yang paling serupa dengan artikel. Tetapi karena topic yang baru muncul tiap hari dibutuhkan suatu mekanisme jika suatu kondisi yang diberikan ke topic adalah sebuah pilihan yang baik.

Algoritma ini menghitung *similarity* antara kata kunci topik yang sebelumnya telah diketahui dengan kata kunci artikel uji. Setelah itu, nilai yang memiliki *similarity* paling tinggi ditetapkan untuk artikel sebagai *conditionally assigned topic*. Sebagaimana yang telah dijelaskan mengenai *vector-space model*, kata kunci dokumen dan topik juga direpresentasikan dalam bentuk vektor. Nilai isi vektor merupakan skor kata kunci.

Untuk membandingkan antara vektor kata kunci dengan vektor topik, keduanya ditransformasikan ke dalam *vector-space* yang sama. Bila sebuah kata kunci terdapat dalam vektor artikel saja, maka vektor topik juga ditambahkan tempat untuk kata kunci tersebut namun dengan nilai kata kunci 0, karena kata kunci tersebut tidak ditemukan di dalam topik. Begitu juga sebaliknya apabila kata kunci ditemukan hanya pada vektor topik, maka disediakan tempat untuk kata kunci pada vektor artikel dengan nilai sebesar 0. Contoh transformasi vektor dapat dilihat pada Gambar 5.26

	gempa	berkekuatan	Aceh	Richter		gempa	berkekuatan	Aceh	Richter	korban
Topik:	2	5	4	1	➔	2	5	4	1	0
Artikel:	gempa	Aceh	korban		➔	gempa	berkekuatan	Aceh	Richter	korban
	1	3	1			1	0	3	0	1

Gambar 5.26 Contoh Transformasi Vektor

Setelah kedua vektor dinormalisasi, maka *CosSim* untuk keduanya dihitung menggunakan rumus 5.3. Pada rumus tersebut, t_i adalah vektor topik ke – *i*, dan *A* adalah artikel uji A. $|t_i|$ dan $|A|$ beruturut – turut merupakan panjang vektor topik ke – *i* dan panjang vektor Artikel A. *CosSim* tertinggi dipilih sebagai *conditionally assigned topic*, atau topik awal yang ditentukan. Topik ini nantinya akan diuji kembali menggunakan nilai ambang batas dinamis (*dynamic thresholding*).

$$CosSim(t_i, A) = \frac{t_i \bullet A}{|t_i||A|} \quad (5.3)$$

Dengan perkembangan berita yang ada di dunia, tidaklah mungkin topik – topik yang sebelumnya pernah muncul dapat mencakup seluruh topik yang diprediksi akan muncul di masa depan. Oleh karena itu, algoritma identifikasi topik juga harus dapat mendeteksi, apakah topik awal yang ditentukan sudah merupakan topik yang tepat atau topik baru harus diberikan. Hal ini dapat dilakukan dengan perhitungan nilai ambang yang dinamis. Nilai ambang atau *threshold* akan membandingkan antara nilai topik awal yang ditentukan dengan nilai topik baru yang mungkin terbentuk *NewTSim* menggunakan rumus 5.4

$$NewTSim(t_c, A) = \frac{(0.05 \times (Mean(A) - StdDev.(A)) \times Mean(t_c)}{(|A| \times (Mean(A))^2) \times (|t_c| \times (Mean(t_c))^2)} \quad (5.4)$$

NewTSim menghitung topik baru secara hipotetis. Pada rumus t_c merupakan topik awal yang telah ditentukan, yaitu hasil perhitungan *CosSim* terbesar, *Mean(A)* merupakan rata – rata vektor dokumen A, *StdDev.(A)* adalah standar deviasi vektor dokumen A, dan *Mean(t_c)* adalah rata – rata topik awal yang telah ditentukan. |*A*| adalah panjang vektor dokumen A, dan |t_c| adalah panjang vektor topik. Bila nilai *NewTSim* telah ditemukan, maka langkah berikutnya adalah menggunakan nilai tersebut sebagai salah satu komponen dalam *dynamic thresholding* untuk dibandingkan dengan nilai topik awal yang telah ditentukan. Terdapat dua nilai ambang batas yang harus dipenuhi agar sebuah topik awal dapat ditetapkan untuk artikel. Nilai ambang pertama dijelaskan pada persamaan 5.5 dan nilai ambang kedua dijelaskan pada persamaan ke 5.6.

$$CosSim(t_c, A) > 0.1 \wedge CosSim(t_c.A) > NewTSim(t_c, A) \quad (5.5)$$

$$NumTopics > 10 \wedge CosSim(t_c, A) > (2 \times StdDev(AllTopicSims) + Mean(AllTopicSims)) \quad (5.6)$$

$CosSim(t_c,A)$ adalah hasil perhitungan *Cosine Similarity* terbesar yang didapatkan melalui rumus 5.3. dan selanjutnya dianggap sebagai topik awal yang ditentukan, sementara *NumTopics* merupakan jumlah keseluruhan topik yang telah diketahui sebelumnya, *StdDev*(*AllTopicSims*) dan *Mean*(*AllTopicSims*) adalah berturut – turut standard deviasi dan rata – rata seluruh *similarity* topik yang telah dihitung pada tahap klasifikasi topik. Persamaan 5.5 membandingkan antara *similarity* topik yang telah ditentukan dengan konstanta dam dengan nilai topik hipotetis baru yang didapatkan melalui rumus 5.4. Persamaan 5.6 berguna jika jumlah topik yang telah diketahui sebelumnya telah mencukupi. Berdasarkan hasil eksperimen, jumlah topik yang harus dipenuhi adalah sepuluh. Selain itu, persamaan 5.5. juga memeriksa apakah *CosSim* topik awal tersebut mempunyai beda besar yang jauh dibandingkan dengan topik – topik lainnya. Apabila nilai *similarity* topik memenuhi kedua nilai ambang, maka topik yang telah ditentukan sebelumnya ditetapkan sebagai topik untuk dokumen. Sebaliknya, apabila nilai tersebut tidak memenuhi kedua nilai ambang, berarti topik baru harus diberikan dan topik baru tersebut kemudian menjadi sumber pertama untuk data *training*.

Algoritma 5.11. Identifikasi Topic

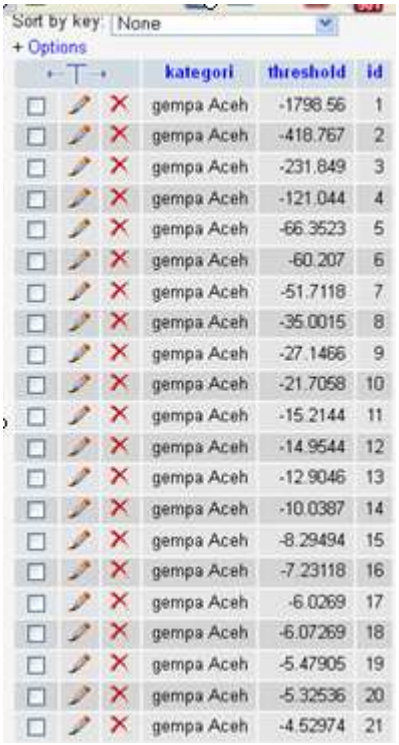
```
$sql2="insert into `likelihood` (
`situsx`,`indokumen`,`kategori` )SELECT  '$situs',count(newsterm.term),'$kategori'
FROM `newsterm`,`keyword` WHERE (situs ='$situs') and
(newsterm.term=keyword.term)";
  $sql2="update `likelihood` set totaldokumen=(select count(*) from totaldokumen)";
  $sql2="update `likelihood` set `likelihood` = -
(indokumen/totaldokumen)*log(indokumen/totaldokumen) ";
  $sql2="insert into `cosimilarity` (
```

```
`situs`,`cosin`,`kategori` ) select '$situs',xny / ( sqrt(x2)*sqrt(y2)) as hasil,'$kategori'
from (select count(j1.term) as xny from
                (SELECT term FROM newsterm WHERE situs='$situs')  j1,
keyword j2 where j1.term=j2.term) atas,
(select count(j3.term) as x2 from newsterm j3 where j3.situs='$situs') x,
(select count(j4.term) as y2 from keyword j4) y";
$sql2="insert into `threshold` (
`kategori`,`threshold` ) select '$kategori',x2+sqrt(sum(pow((y2 - x2),2))/x3) from
(select likelihood as y2 from likelihood)  y,
(SELECT  sum(likelihood)/count(likelihood) as x2,count(likelihood) as x3 FROM
`likelihood`)  x;";
```

Dari algoritma 5.11 akan akan dicari similaritas tertinggi dari masing-masing topic sehingga akan membentuk topic berdasarkan likehood dan hasil perhitungan threshold. Jika nilai similaritasnya terpenuhi maka akan mengklaster menjadi topic yang sama, jika tidak maka akan membentuk topic baru. Tabel hasil perhitungan cosin dan threshold dapat dilihat pada Gambar 5.27 dan Gambar 5.28. Sedangkan tabel hasil perhitungan likehood dapat dilihat pada Gambar 5.29.

	situs	cosin	id	kategori
	http://sains.kompas.com/read/2012/04/12/07025629/G...	0.292529	31	gempa Aceh
	http://regional.kompas.com/read/2012/04/12/0649431...	0.449868	32	gempa Aceh
	http://regional.kompas.com/read/2012/04/12/0131297...	0.162305	33	gempa Aceh
	http://travel.kompas.com/read/2012/04/11/2210470/W...	0.256885	34	gempa Aceh
	http://nasional.kompas.com/read/2012/04/11/2209468...	0.22225	35	gempa Aceh
	http://regional.kompas.com/read/2012/04/11/2207502...	0.235702	36	gempa Aceh
	http://regional.kompas.com/read/2012/04/11/2154318...	0.215772	37	gempa Aceh
	http://nasional.kompas.com/read/2012/04/11/2153307...	0.263768	38	gempa Aceh
	http://nasional.kompas.com/read/2012/04/11/2149304...	0.242536	39	gempa Aceh
	http://regional.kompas.com/read/2012/04/11/2141183...	0.232858	40	gempa Aceh
	http://tekno.kompas.com/read/2012/04/11/21260780/S...	0.216815	41	gempa Aceh
	http://regional.kompas.com/read/2012/04/11/2126034...	0.267261	42	gempa Aceh
	http://regional.kompas.com/read/2012/04/11/2117584...	0.292	43	gempa Aceh
	http://nasional.kompas.com/read/2012/04/11/2110487...	0.186253	44	gempa Aceh
	http://nasional.kompas.com/read/2012/04/11/2107447...	0.199168	45	gempa Aceh

Gambar 5.27 Gambar Tabel Hasil Cosine Coefisien



Sort by key: None

+ Options

	kategori	threshold	id
	gempa Aceh	-1798.56	1
	gempa Aceh	-418.767	2
	gempa Aceh	-231.849	3
	gempa Aceh	-121.044	4
	gempa Aceh	-66.3523	5
	gempa Aceh	-60.207	6
	gempa Aceh	-51.7118	7
	gempa Aceh	-35.0015	8
	gempa Aceh	-27.1466	9
	gempa Aceh	-21.7058	10
	gempa Aceh	-15.2144	11
	gempa Aceh	-14.9544	12
	gempa Aceh	-12.9046	13
	gempa Aceh	-10.0387	14
	gempa Aceh	-8.29494	15
	gempa Aceh	-7.23118	16
	gempa Aceh	-6.0269	17
	gempa Aceh	-6.07269	18
	gempa Aceh	-5.47905	19
	gempa Aceh	-5.32536	20
	gempa Aceh	-4.52974	21

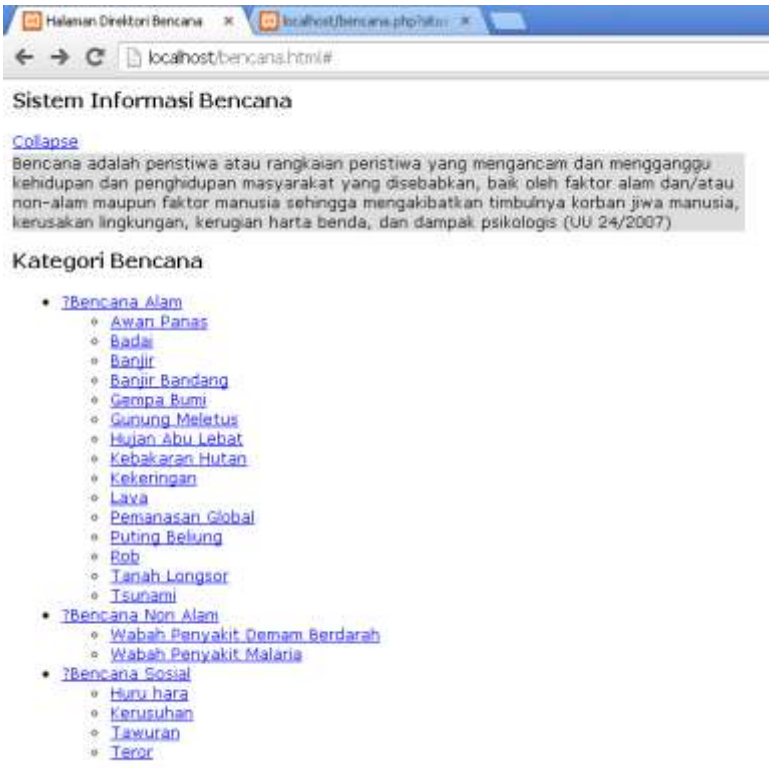
Gambar 5.28 Gambar Tabel Hasil Hitung Threshold

	situss	kategori	likelihood	id	indokumen	totaldokumen
	http://sains.kompas.com/read/2012/04/16/09073470/S...	gempa Aceh	-1.94971	1	313	135
	http://sains.kompas.com/read/2012/04/16/17394852/G...	gempa Aceh	-0.431778	2	185	135
	http://nasional.kompas.com/read/2012/04/16/1523408...	gempa Aceh	-0.531159	3	195	135
	http://sains.kompas.com/read/2012/04/14/06141978/A...	gempa Aceh	0.0434418	4	129	135
	http://regional.kompas.com/read/2012/04/13/2156449...	gempa Aceh	0.232519	5	98	135
	http://sains.kompas.com/read/2012/04/13/20402973/E...	gempa Aceh	-0.817951	6	222	135
	http://sains.kompas.com/read/2012/04/13/17483964/R...	gempa Aceh	-0.471061	7	189	135
	http://regional.kompas.com/read/2012/04/13/1211065...	gempa Aceh	0.274673	8	89	135
	http://regional.kompas.com/read/2012/04/13/1145417...	gempa Aceh	0.130336	9	116	135
	http://sains.kompas.com/read/2012/04/13/07393832/T...	gempa Aceh	0.136588	10	115	135
	http://regional.kompas.com/read/2012/04/13/0009136...	gempa Aceh	0.356665	11	63	135
	http://regional.kompas.com/read/2012/04/13/0006474...	gempa Aceh	-0.571995	12	199	135
	http://nasional.kompas.com/read/2012/04/12/2358394...	gempa Aceh	0.0916034	13	122	135
	http://regional.kompas.com/read/2012/04/12/2347287...	gempa Aceh	0.323446	14	76	135
	http://regional.kompas.com/read/2012/04/12/2343247...	gempa Aceh	0.256732	15	93	135
	http://sains.kompas.com/read/2012/04/12/17473354/G...	gempa Aceh	0.172726	16	109	135
	http://nasional.kompas.com/read/2012/04/12/1713595...	gempa Aceh	0.278952	17	88	135
	http://properti.kompas.com/read/2012/04/12/1654473...	gempa Aceh	-0.510972	18	193	135
	http://sains.kompas.com/read/2012/04/12/1640269/Ge...	gempa Aceh	0.130336	19	116	135
	http://regional.kompas.com/read/2012/04/12/1604515...	gempa Aceh	-0.192719	20	159	135
	http://regional.kompas.com/read/2012/04/12/1532198...	gempa Aceh	0.295218	21	84	135
	http://nasional.kompas.com/read/2012/04/12/1441298...	gempa Aceh	0.0434418	22	129	135
	http://regional.kompas.com/read/2012/04/12/1417059...	gempa Aceh	0.24726	23	96	135
	http://sains.kompas.com/read/2012/04/12/13291742/G...	gempa Aceh	-0.0454178	24	141	135

Gambar 5.29 Gambar Tabel Likehood

Pengujian dan Tampilan Luaran

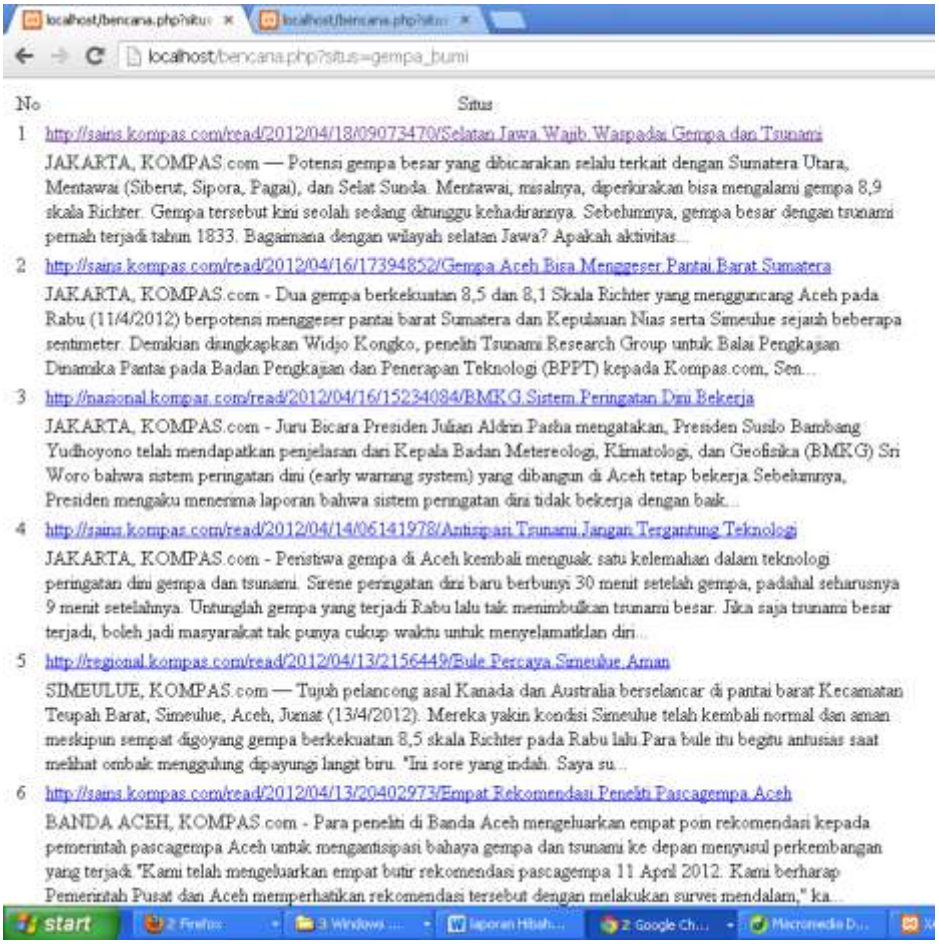
Untuk tiap kategori, setiap klasifikasi diuji dengan 100 artikel. Untuk pengujian 800 artikel diekstrak dari berbagai situs berita online. Kategori yang digunakan situs berita digunakan untuk menentukan kategori yang diberikan ke artikel. Sebagai contoh jika artikel di bawah kategori gempa pada situs maka artikel akan di bawah gempa di kategori. Selanjutnya dilakukan pengujian untuk penemuan topic baru dan klasifikasi. Pengujian dilakukan dengan menggunakan berbagai sumber berita online. Hasil tampilan klasifikasi berita bencana dapat dilihat pada Gambar 5.30



Gambar 5.30 Tampilan Direktori Berita Bencana

Pada gambar akan ditampilkan daftar bencana berdasarkan UU no 24 Tahun 2007. Jika dipilih salah satu jenis bencana, makan akan muncul daftar

berita yang sesuai dengan jenis bencana. Sebagai contoh pada Gambar 5.31 ditampilkan daftar berita bencana gempa bumi.



Gambar 5.31 Tampilan berita dengan kategori bencana alam

Algoritma dalam penelitian ini didasarkan pada ekstraksi kata kunci yang tidak memerlukan koleksi dokumen. Dari hasil pengujian, algoritma yang digunakan mendapatkan hasil yang bagus. Algoritma klasifikasi dapat melatih pengklasifikasinya secara independen untuk tiap kategori dan secara mudah dapat diperbaharui.

BAB VI

KESIMPULAN DAN SARAN

6.1. Kesimpulan

Dari hasil penelitian yang telah dilakukan dapat disimpulkan hal-hal sebagai berikut :

1. Telah dibangun program crawler, yaitu program yang digunakan untuk melakukan penjelajahan dan pengambilan halaman-halaman web yang ada di internet. Hasil pengumpulan situs web selanjutnya akan diindeks oleh mesin pencari sehingga mempermudah pencarian.
2. Implementasi/pembuatan program preprosesing yang terdiri dari program tokenisasi, penghilangan stopword, program stemming.
3. Pembobotan term frekuensi dan cosine similaritas digunakan untuk menunjukkan kemiripan antar dokumen..
4. Sistem dapat menampilkan dokumen yang mempunyai kedekatan similaritas dari query yang diinputkan user.
5. Dokumen yang membahas topik yang sama cenderung untuk mengelompok menjadi satu klaster.
6. *Single Pass clustering* cukup handal digunakan sebagai algoritma untuk klasifikasi *event*
7. Klaster dapat membantu menemukan dokumen yang ada dalam satu klaster dengan query yang diinputkan user.
8. Klaster dapat membantu mendapatkan dokumen yang relevan.

9. Untuk menyusun kategori suatu bencana dilakukan pelatihan dengan cara mengekstrak kata kunci dari situs berita online. Kategori yang dibuat disesuaikan dengan kategori dari situs berita online.
10. Kategori yang berisi kata kunci digunakan untuk mengklasifikasi dokumen artikel situs berita online yang baru. Untuk menetapkan suatu artikel baru termasuk dalam suatu kategori digunakan rumus cosine similaritas.
11. Untuk menetapkan apakah suatu artikel perlu menjadi kategori baru, maka digunakan dua buah threshold. Apabila kedua threshold terpenuhi, maka artikel yang baru akan menjadi dasar kategori yang baru.
12. Dari hasil pengujian,sistem yang dibangun mampu mengklasifikasi artikel baru dari situs berita online dan diapatkan hasil yang bagus.

6.2. Saran

Untuk penelitian selanjutnya ada beberapa ha yang perlu dilakukan :

1. Perlunya system pengenalan entitas (entity recognition) sehingga system mampu membaca data korban dan kerusakan yang terjadi.
2. Manajemen system crawler sehingga mampu membaca sejumlah besar situs berita online dengan tidak membutuhkan penyimpanan yang besar.

DAFTAR PUSTAKA

[1.] Allan et al ,1998, *Topic Detection and Tracking Pilot Study : Final Report.*, Proc. DARPA Broadcast News Transcription & Understanding Workshop, Morgan Kaufman, San Francisco, pp194-218

[2.] Arifin, AZ., Setiono, AN,2002, *Klasifikasi Dokumen Berita Kejadian Berbahasa Indonesia dengan Algoritma Single Pass Clustering*, SITIA, Proceeding of Seminar on Intelligent Technology and Its Applications (SITIA), Teknik Elektro, Institut Teknologi Sepuluh Nopember

[3.] Hartati, S. dan Zuliarso, E. , 2008,*Aplikasi Pengolah Bahasa Alami Untuk Query Basisdata XML*, Dinamik, Jurnal Teknologi Informasi, Universitas Stikubank Semarang, Vol XIV, Juli 2008

[4.] Februariyanti, H, 2010, *Prototipe Mesin Pencari Dokumen Teks*, Penelitian Universitas Stikubank

[5.] Februariyanti, H., Winarko, E., 2010, *Klastering Dokumen Menggunakan Hierarchical Agglomerative Clustering*, Seminar Nasional Teknologi Informasi, STIKOM, Surabaya.

[6.] Nasraoui, O. ,2005 ,*World Wide Web Personalization*, Department of Computer Engineering and Computer Science, University of Louisville, USA.

[7.] Papka., R. ,1999, *On-Line New Event Detection, Culstering, and Tracking*,.Ph. D dissertation on University of Massachusetts.

[8.] Susetyo, W., Hendrantoro, G., Affandi, A.,2008, *Coverage Prediction Of Hf Wireless Network For Disaster Early Warning System In Indonesia*, Seminar Nasional Aplikasi Teknologi Informasi 2008 (SNATI 2008) Yogyakarta.

[9.] Utami, E., Cahyanto, AD., 2008, *Sistem Peringatan Dini Pada Bencana Banjir Berbasis Sms Gateway Di Gnu/Linux Merupakan Alternatif Yang Sederhana Dan Menarik Dalam Meningkatan Pelayanan Badan Meteorologi Dan Geofisika Dengan Alokasi Dana Yang Rendah*, Seminar Nasional Aplikasi Teknologi Informasi 2008 (SNATI 2008) , Yogyakarta.

[10.] Utomo, MS, Winarko, Edi, 2011, *Design And Implementation of Document Similarity Search System For Web-Based Medical Journal Management, Indonesian Journal of Computing and Cybernetic Systems*, Indo CEISS.

[11.] JYH, P. ,2006, *Web Personalization Using Implicit Input*, Thesis, Universiti Sins Malaysia.

[12.] Vert, G., Iyengar, SS, Phoha, 2010**,** *Introduction to Contextual Processing Theory and Applications*, Chapman and Hall/CRC.

[13.] Zuliarso,E., Mustofa,K., 2009a, *Crawling Web Berbasis Konten*, Dinamik, Jurnal Teknologi Informasi, Universitas Stikubank Semarang, Vol XIV, Juli 2009.

[14.] Zuliarso,E.,Mustofa,K., 2009b, *Crawling Web Berdasarkan Ontologi*, Seminar Nasional V, Jurusan Matematika, FMIPA Universitas Negeri Semarang,Ontober 2009

LAMPIRAN 1

MATRIKS JADWAL KERJA PENELITIAN TAHAP I (TAHUN I)

BULAN KEGIATAN	II	III	IV	V	VI	VII	VIII	IX	X	X
PERENCANAAN - Studi Kelayakan - Penaksiran Kebutuhan										
ANALISIS KEBUTUHAN - Survey - Identifikasi masalah - Analisis Kebutuhan Dokumentasi dijital										
SPESIFIKASI SISTEM - Spesifikasi kebutuhan sistem - Spesifikasi Kebutuhan Perangkat Lunak										
PERANCANGAN KONSEPTUAL - Rancangan Global - Rancangan Detail - Prototyping (pemodelan)										
IMPLEMENTASI - Koding - Pengujian										
PENULISAN LAPORAN										

LAMPIRAN 2

PEMAKAIAN ANGGARAN TAHAP I (TAHUN I)

No	Jenis Kegiatan						Dana Yang digunakan	
1							***Persiapan***	
								-
2							***Honor Tim Peneliti***	
	Honor Peneliti							
	a	Peneliti Utama [15 jam per minggu selama 9 Bulan]	1	Org	@	Rp 7,000	Rp	3,360,000
	b	Anggota Peneliti [15 jam per minggu selama 9 Bulan]	2	Org	@	Rp 7,000	Rp	5,760,000
							Rp	***9,120,000***
3							***Bahan dan Peralatan Penelitian***	
	Bahan Habis Pakai							
	a	Kertas HVS 80 Gram	4	Rim	@	Rp 35,000	Rp	140,000
	b	Pulsa Modem	9	Bulan	@	Rp 100,000	Rp	900,000
	c	Toner Printer Laserjet	1	Buah	@	Rp 400,000	Rp	400,000
	d	Sewa Web Server	7	Bulan			Rp	6,010,000
							Rp	***7,380,000***
	Alat							
	a	Modem	1	Buah	@		Rp	600,000
	b	External Hardisk [500 Gbyte]	1	Buah	@		Rp	500,000
							Rp	***1,100,000***
	Rancang Bangun Sistem Temu Kembali							
		Perencanaan Kebutuhan						
	a	Installasi Komputer	1	Org	***@***	Rp 500,000	Rp	500,000
	b	Desain Konseptual [5 pertemuan]	3	Org	***@***	Rp 100,000	Rp	1,500,000
	c	Line of Code					Rp	7,700,000
	d	Testing dan Evaluasi Sistem					Rp	2,240,000
							Rp	***11,940,000***

4	***Seminar Penelitian***							
	a	Transport Monev Desentralisasi					Rp	150,000
	b.	Transport Monev Terpusat					Rp	150,000
	c	Pembayaran Pemakalah	1	Org	@	Rp 200,000	Rp	200,000
	d	Pembayaran Prosiding	4	Buah	@	Rp 200,000	Rp	800,000
	e	Transport Call Papper	1	Orang	@	Rp 300,000	Rp	300,000
							Rp	***1,600,000***
5	***Pembuatan Laporan Penelitian***							
	Penggandaan Laporan							
	a.	Fotocopy dan Penjilidan					Rp	190,000
							Rp	***190,000***
							Rp	***31,400,000***

LAMPIRAN 3

PERSONALIA PENELITI

1. Ketua Peneliti

Nama	:	Herny Februariyanti, ST., M.Cs
Keahlian	:	Information Retrieval
Kebangsaan	:	Indonesia
Agama	:	Islam
Golongan/Pangkat	:	III C / Penata Muda Tk. I
Jabatan Fungsional	:	Lektor
Alamat Rumah	:	Jl. Kendeng V / 12 Sampangan Semarang
Informasi Kontak	:	Telp/Hp : 08156545909 Email : herny@unisbank.ac.id, hernyfeb@gmail.com
Alamat Universitas	:	Jl. Tri Lomba Juang No.1 Mugasari Semarang
Informasi Kontak	:	Telp/Fax : 024-8311668/024-8443240 Email : info@unisbank.ac.id
Kemampuan Berbahasa	:	Bahasa Indonesia, Bahasa Inggris
Pendidikan dan kualifikasi lainnya	:	a. Sarjana Teknik , Manajemen Informatika & Teknik Komputer, Institut Sain & Teknologi ”AKPRIND” Yogyakarta, 1998. b. Magister Komputer, Ilmu Komputer, UGM, 2010
Ringkasan pengalaman : *Beri tanda pengalaman-pengalaman yang serupa dengan ruang lingkup proyek rintisan ini.*		
Pengalaman yang relevan (Diurutkan mulai dari yang terbaru)		

PENELITIAN		
Periode : (Dari – Sampai)	Nama Kegiatan Penelitian	Posisi/jabatan dan kegiatan/tanggung jawab yang dilakukan
2012	Klasifikasi Berita Menggunakan Ontologi	Ketua
2011	Klastering Dokumen Berita dari Web Menggunakan Algoritma Single Pass Clustering	Ketua
2010	Aplikasi Generator Konten untuk Meningkatkan Peringkat Situs pada Halaman Hasil Mesin Pencari.	Anggota
2010	Prototipe Mesin Pencari Dokumen Teks	Ketua
2010	Aplikasi Pengelolaan Peraturan Daerah Provinsi Jawa Tengah Menggunakan Basisdata XML	Anggota
2009	Aplikasi Pengindeks Kata Berbasis Web Pada Dokumen Teks Berbahasa Indonesia Untuk Keperluan Temu Kembali Informasi.	Ketua

2009	Hierarchical Agglomerative Clustering untuk Sistem Temu Kembali Dokumen Bahasa Indonesia	Ketua
2009	Pengindeks Kata Dokumen Teks dengan Menggunakan Aplikasi Berbasis Web	Ketua
Referensi :	Universitas Stikubank Semarang - Kampus Mugas : Jl. Trilomba Juang No 1 Semarang 50241, Telp (62-24) 8451976,8311668, Fax (024) 8443240 - Kampus Kendeng : Jl. Kendeng V Bendan Ngisor Semarang, Telp (62-24) 8414970, fax (024) 8441738 Alamat email : info@unisbank.ac.id	
PENGABDIAN		
Periode : (Dari – Sampai)	Nama Kegiatan Penelitian	Posisi/jabatan dan kegiatan/tanggung jawab yang dilakukan
2011	Pelatihan Pembelajaran Frame By Frame Animation untuk Alat Bantu Ajar Menggunakan Flash Bagi Guru-Guru SMK Ibu Kartini Semarang	Ketua
2010	Pelatihan Pembuatan Blok Sekolah dan Hosting Berbayar Wordpress Bagi Siswa SMK Nusa Bhakti Semarang	Anggota
2009	Pemanfaatan Pengolah Tabel Di Kelurahan Bendan Ngisor Kec. Gajah Mungkur	Anggota
2009	Evaluasi Soal & Pembentukan Tim Olimpiade Matematika Tingkat SMA Bagi SMA Nusaputra Semarang	Anggota
2009	Pelatihan Internet Bagi Guru MTs / MA Taqwiyatul Wathon Mranggen Demak	Ketua
Semarang, 14 November 2012 Herny Februariyanti, ST., M.Cs		

2. Anggota Peneliti I

Nama	:	Eri Zuliarso, Drs. , M. Kom.
Jenis Kelamin	:	Laki-Laki
NIP	:	YU.2.10.11.097
Disiplin Ilmu	:	*Computer Science specialist Digital Image Processing*
Pangkat/golongan	:	Penata Muda Tk II / IIIB
Jabatan Fungsional	:	Asisten Ahli
Alamat Rumah	:	Jl Pucang Permai III/2 , Mranggen, Demak
Telephon/E-mail	:	085876470885 / eri@unisbank.ac.id
Fakultas/Jurusan	:	Teknologi Informasi/Teknik Informatika
Waktu Penelitian	:	15 Jam/minggu

Ringkasan pengalaman :
Pengalaman-pengalaman yang serupa dengan ruang lingkup penelitian ini.

Pengalaman Penelitian yang relevan (Diurutkan mulai dari yang terbaru)

Periode : (Dari – Sampai)	**Nama Kegiatan Penelitian**	**Posisi/jabatan dan kegiatan/tanggung jawab yang dilakukan**
2012	Klasifikasi Berita Menggunakan Ontologi	Anggota
2011	Klastering Dokumen Berita dari Web Menggunakan Algoritma Single Pass Clustering	Anggota
2010	Web Service memanfaatkan layanan Facebook	Ketua
2010	Pembuatan Crawling Web	Ketua
2009	Aplikasi Pengolah Bahasa Alami Untuk Query Basisdata XML Akademik	Ketua
Referensi :	Universitas Stikubank Semarang - Kampus Mugas : Jl. Trilomba Juang No 1 Semarang 50241, Telp (62-24) 8451976,8311668, Fax (024) 8443240 - Kampus Kendeng : Jl. Kendeng V Bendan Ngisor Semarang, Telp (62-24) 8414970, fax (024) 8441738 Alamat email : info@unisbank.ac.id	

Semarang, 14 November 2012

(Drs. Eri Zuliarso, M. Kom)

3. Anggota Peneliti II

Nama	:	Mardi Siswo Utomo, M.Cs
Keahlian	:	Temu kembali Informasi dan Pengembangan Aplikasi Berbasis Web
Kebangsaan	:	Indonesia
Agama	:	Islam
Golongan/Pangkat	:	III B / Penata Muda
Jabatan Fungsional	:	LEKTOR
Alamat Rumah	:	Klipang Pesona Asri III Blok A No 235 Semarang 50272
Informasi Kontak	:	Telp/Hp : 085865349880 Email : mardiutomo@gmail.com
Alamat Universitas	:	Jl. Tri Lomba Juang No.1 Mugasari Semarang
Informasi Kontak	:	Telp/Fax : 024-8311668/024-8443240 Email : info@unisbank.ac.id
Kemampuan Berbahasa	:	Bahasa Indonesia, Bahasa Inggris
Pendidikan dan kualifikasi lainnya	:	a. Sarjana Komputer , Teknik Informatika, STMIK Stikubank Semarang b. Magister Komputer, Ilmu Komputer, UGM, 2010
Ringkasan pengalaman : *Beri tanda pengalaman-pengalaman yang serupa dengan ruang lingkup proyek rintisan ini.*		
Pengalaman yang relevan (Diurutkan mulai dari yang terbaru)		
PENELITIAN		

Periode : (Dari – Sampai)	**Nama Kegiatan Penelitian**	**Posisi/jabatan dan kegiatan/tanggung jawab yang dilakukan**
2010	Aplikasi Generator Konten Untuk Meningkatkan Peringkat Situs Halaman Hasil Mesin Pencari	Anggota
2009	Pengujian Kemiripan Antar Dokumen Teks Pada Aplikasi Berbasis Web.	Ketua
2009	Implementasi Stemmer Bahasa Indonesia Tanpa Kamus Berbasis Web Untuk Keperluan Temu Kembali Informasi	Anggota
2008	Rekayasa Perangkat Lunak Database Jurnal Berbasis Web Pada Jurnal Media Medika Indonesiana Fakultas Kedokteran Universitas Diponegoro Semarang	Anggota
2004	Rekayasa Perangkat Lunak Sistem Informasi Apotek Berbasis Web	Anggota

2003	Pemanfaatan Teknologi Wireless Application Protocol (Wap) Untuk Mengakses Informasi Akademik Melalui Telepon Seluler	Ketua
2001	Analisa Dan Implementasi Lexical Analyzer Pada Struktur Kalimat Bahasa Indonesia Dalam Rangka Information Retrieval	Anggota
Referensi :	Universitas Stikubank Semarang - Kampus Mugas : Jl. Trilomba Juang No 1 Semarang 50241, Telp (62-24) 8451976,8311668, Fax (024) 8443240 - Kampus Kendeng : Jl. Kendeng V Bendan Ngisor Semarang, Telp (62-24) 8414970, fax (024) 8441738 Alamat email : info@unisbank.ac.id	
PENGABDIAN		
Periode : (Dari – Sampai)	Nama Kegiatan Penelitian	Posisi/jabatan dan kegiatan/tanggung jawab yang dilakukan
2004	Pelatihan Internet Bagi Guru Dan Karyawan Sma Perintis Semarang	Ketua
2002	Pelatihan Instalasi Sistem Operasi Freebsd Bagi Teknisi Laboratorium Komputer	Anggota
Semarang, 14 November 2012 Mardi Siswo Utomo, M.Cs		

LAMPIRAN 4
DRAF PUBLIKASI CALL PAPPER