TEXT MINING PADA *NEWS AGGREGATOR* UNTUK SISTEM REKOMENDASI BERITA

Kaharisman Ramdhani[1], Galih Hermawan[2]

[1,2] Universitas Komputer Indonesia
Jalan Dipatiukur No. 112-116, Coblong, Lebakgede, Bandung, Jawa Barat 40132
E-mail: khrsman@gmail.com[1], galih.hermawan@email.unikom.ac.id[2]

Abstrak - *News aggregator* merupakan suatu perangkat lunak atau aplikasi yang menggabungkan konten dari berbagai web. Dalam news aggregator terdapat banyak data berita yang berisi informasi yang dapat dibaca oleh pembaca. Pada news aggregator bisa terjadi information overload dikarekanan banyak data berita yang sama sekali tidak dibaca oleh pembaca dikarenakan pembaca hanya membaca berita yang menarik. Penelitian text mining pada news aggregator telah dilakukan oleh beberapa peneliti untuk mengelompokan berita berdasarkan topik yang sama dengan menggunakan metode yang berbeda, hanya saja pada penelitian tersebut data cluster selalu di *inisialisasi* di awal. Hal ini kurang efektif diterapkan pada news aggregator yang memiliki jenis data berita yang beragam.

Penggunaan metode *clustering* yang tidak sensitif terhadap inisialiasi dapat dilakukan untuk meningkatkan akurasi setiap *cluster*. Pada penelitian ini menggunakan *metode cosine similarity* dan *single pass clustering*.

Pengujian dilakukan dengan 157 data berita dan menghasilkan akurasi rata-rata sebesar 88.8%. Penentuan threshold sangat berpengaruh terhadap akurasi data pada *cluster* yang dihasilkan, hasil pengujian menujukan bahwa akurasi maksimal yang didapatkan adalah 100% dan akurasi terkecil adalah 63%. Hasil penelitian ini menyimpulkan bahwa metode cosine similarity dan single pass clustering dapat diterapkan pada news aggregator.

Kata kunci*:* penambangan teks, informasi berita, pengumpul berita, *cosine similarity, single pass clustering*

I. PENDAHULUAN

Text mining merupakan proses ekstraksi dari sejumlah besar data yang berbentuk text atau dokumen untuk menemukan informasi yang berguna, tersembunyi, dan tidak diketahui sebelumnya [1]. Tujuan dari *text mining* adalah mendapatkan informasi yang berguna dari sekumpulan dokumen. Sumber data yang digunakan pada *text mining* adalah kumpulan teks yang memiliki format tidak terstruktur atau minimal semi terstruktur. Menurut penelitian dari Lokesh Kumar [2], *text mining* dan *information extraction* telah menjadi area populer penelitian untuk mengekstrak informasi yang menarik dan berguna. Jadi sangat penting mengembangkan teknik dan algoritma yang lebih baik untuk mengekstrak informasi yang berguna. Salah satu area yang dapat dilakukan *text mining* adalah pada *news aggregator.* Dalam sistem *news aggregator*, pengelompokan berita memiliki peran penting karena setiap kelompok berita menyatakan satu topik berita yang anggotanya merupakan artikel-artikel berita dari berbagai portal berita. Kualitas kelompok berita sangat penting karena dapat membantu pembaca untuk memilih topik berita yang diinginkan. Pengelompokan berita berbahasa Indonesia telah dilakukan oleh beberapa peneliti dengan berbagai teknik dan tujuan. Metode mengelompokan berita dengan menggunakan *partitional clustering* dengan cluster yang diinisialisasi merupakan teknik yang paling sederhana dan umum digunakan untuk berita bahasa Indonesia karena metode ini mudah diimplementasikan [3]. Metode ini melakukan *clustering* dengan meng-inisialisasi *cluster* terlebih dahulu, setiap *cluster* diinisialisasi secara *random* sehingga pengelompokkan dokumen yang dihasilkan dapat berbeda-beda. Setiap dokumen akan ditentukan masuk kedalam *cluster* yang mana dengan membandingkan satu persatu dokumen uji dengan *cluster* yang telah diinisialisasi [4]. Jika nilai *random* untuk inisialisasi kurang baik, maka pengelompokkan yang dihasilkan pun menjadi kurang optimal. Penggunaan metode *partitional cluster*ing dengan inisialisasi *cluster* pada *news aggregator* masih menghasilkan *outlier* atau dokumen yang seharusnya tidak dalam satu *cluster* [5]. Berdasarkan penelitan tentang penerapan salah satu metode *partitional clustering* dengan inisialisasi cluster [5], dengan menerapkan metode *partitional*

clustering pada news aggregator terkadang menghasilkan *over cluster* jika jumlah dokumennya semakin banyak. Hasil penelitian tersebut menyarankan untuk menggunakan jenis metode *clustering* lain yang tidak sensitif terhadap inisialiasi atau menggunakan metode lain yang dapat menentukan inisialisasi secara dinamis.

II. LANDASAN TEORI

a. *Web scrapping*

Web scrapping merupakan suatu teknik untuk mengutip data ataupun informasi dari suatu web atau blog menggunakan perangkat lunak dengan metode tertentu. Biasanya perangkat lunak tersebut mensimulasikan aktifitas manusia terhadap suatu web atau blog dengan menggunakan low - level HTTP atau menggunakan web browser [9].*Web scraping* mempunyai banyak kegunaan dan sangat membantu untuk pengambilan dokumen, salah satunya yaitu untuk konten berita dimana isi konten nya langsung diambil dari situs yang dijadikan target. Secara umum dalam mengimplementasikan teknik *web scrapping* dibutuhkan beberapa tahap yaitu

1. Memanggil/*Request* url target.
   Sistem akan memanggil target yang dalam hal ini adalah alamat *url http* dari *web* yang dijadikan target, contohnya adalah www.kompas.com
2. Proses pada server target.
   Server target akan melakukan proses untuk *request* yang telah dilakukan, kemudian akan menyajikan data berdasarkan apa yang di *request*.
3. Ekstrasi data.
   Sistem akan melakukan ekstraksi pada data html yang yang berikan oleh server. Data yang dirasa penting akan diambil, kemudian hasil dari ekstraksi ini kemudian akan disimpan kedalam database

b. *News aggregator*

News aggregator merupakan perangkat lunak atau aplikasi yang menggabungkan konten dari berbagai web [10]. contohnya seperti surat kabar online, blog, atau blog video di satu tempat agar mudah dibaca.

Teknologi ini memudahkan pengguna dengan cara menggabungkan data dari beberapa situs web ke dalam satu halaman dan dapat menunjukan informasi yang telah diperbaharui dari situs tersebut.

c. *TF-IDF*

Term Weighting TF-IDF Merupakan skema yang banyak digunakan dalam pembobotan kata. Beberapa hal yang perlu diperhatikan dalam pencarian informasi dokumen adalah pembobotan *term*. Dapat berupa kata, fase atau hasil index lainnya di dalam suatu dokumen, setiap kata di berikan indikator yang disebut dengan *term weight*.

1. *Term Frequency (TF)*

TF merupakan frekuensi dari munculnya sebuah kata atau *term* di dalam dokumen. Semakin besar jumlah kemunculannya semakin besar juga bobotnya atau nilai kesesuaiannya semakin besar. Berdasarkan dari buku yang ditulis oleh jiawei han [12] rumus untuk mendapatkan nilai *TF* ini adalah

$$TF(d,t) = \begin{cases} 0 & if\, freq(d,t) = 0 \\ 1 + \log(1 + \log(freq(d,t))) & otherwise. \end{cases} \quad \textbf{(1)}$$

Keterangan:
TF(d,t) : Nilai *term d* dalam dokumen *t* .
freq(d,t) : Jumlah *term* d dalam dokumen *t*.

2. *Inverse Document Frequency (IDF)*

Suatu perhitungan dari kata didistribusikan secara luas pada kumpulan dokumen yang bersangkutan. *IDF* menunjukkan hubungan ketersediaan sebuah *term* dalam seluruh dokumen. Semakin sedikit jumlah dokumen yang mengandung *term* yang dimaksud, maka nilai IDF semakin besar. *Inverse Document Frequency* (*IDF*) dapat dihitung dengan menggunakan perhitungan

$$IDF(t) = \log \frac{1 + |d|}{|d_t|}, \quad \textbf{(2)}$$

Keterangan:
IDF(t) : Nilai *IDF*.
d : Jumlah keseluruhan dokumen
d_t : Merupakan jumlah dokumen yang mengandung *term t*

Setelah diketahui nilai dari *TF* dan *IDF* nya selanjutnya adalah menentukan nilai *ITF-IDF* nya dengan menggunakan:

$$TF\text{-}IDF(d,t) = TF(d,t) \times IDF(t). \quad \textbf{(3)}$$

Keterangan:
TF-IDF(d,t) : Nilai tf-idf nya

TF(d,t) : Nilai dari tf
IDF(t) : Nilai dari idf

***d.* Cosine Smilarity**

Cosine similarity merupakan metode yang digunakan untuk menghitung tingkat kesamaan/*similarity* antar dua buah objek. Dalam hal ini yang dibandingkan adalah dua buah dokumen. Nilai *cosine similarity* ini adalah dari 0 menuju 1, semakin besar nilainya maka semakin besar pula kemiripan antara dua dokumen tersebut.

$$sim(v_1, v_2) = \frac{v_1 \cdot v_2}{|v_1||v_2|}, \qquad \textbf{(4)}$$

$$sim(_{A,B}) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} * \sqrt{\sum_{i=1}^{n} B_i^2}} \qquad \textbf{(5)}$$

Keterangan**:**

$\sum_{i=1}^{n} A_i B_i$: Jumlah kata yang ada pada dokumen A dan yang ada pada dokumen B.

$\sqrt{\sum_{i=1}^{n} A_i^2}$: Jumlah kata yang ada pada dokumen A.

$\sqrt{\sum_{i=1}^{n} B_i^2}$: jumlah kata yang ada pada dokumen D2.

e. Single Pass Clustering

single pass clustering merupakan metode yang digunakan untuk menggabungkan beberapa dokumen kedalam satu grup yaitu dengan melakukan pengelompokan data satu persatu dan membentuk kelompok dengan evaluasi dari setiap data yang di masukan ke proses *cluster*. Evaluasi tingkat kesamaan antar data dan juga *cluster* dapat dilakukan dengan berbagai cara termasuk juga menggunakan fungsi jarak, vector similarity dan lain lain.

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 sulit suatu data untuk bergabung ke dalam suatu *cluster*, dan demikian juga sebaliknya.

Algoritma yang sering digunakan dalam *Single Pass Clustering* adalah sebagai berikut: [13]

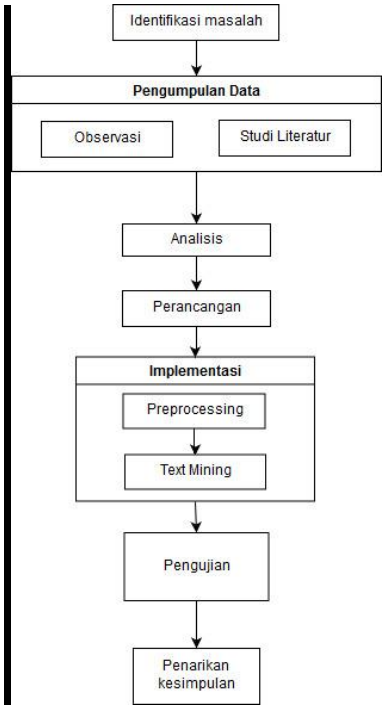
1. untuk setiap perulangan dokumen
   a. Cari *cluster* baru dengan membandingkan dokumen D dan satu dokumen lain untuk dicari nilai *cosine similarity*-nya.
   b. jika nilai *cosine similarity*-nya lebih dari nilai *threshold*, maka dokumen yang dibandingkan tersebut termasuk dalam *cluster* C. Nilai dari *cluster* C yang terbentuk ini adalah nilai tengah antara dokumen D dan dokumen yang dibandingkan.
   c. Jika nilai *cosine similarity*-nya kurang dari nilai *threshold*, maka buat *cluster* baru beranggotakan dokumen D saja
2. Perulangan berhenti

III. METODE PENELITIAN

Metode penelitian yang digunakan dalam membangun sistem ini adalah metode penelitian *deskriptif* [7]. Metode ini digunakan karena pada penelitian ini data sumber berasal dari internet dan tidak dilakukan manipulasi variabel pada data yang akan digunakan. Data yang digunakan merupakan data yang diperoleh apa adanya. Oleh karena itu, metode penelitian deskriptif dirasa cocok untuk digunakan pada penelitian ini. Tahapan penelitian yang akan dilakukan adalah:

a. Identifikasi masalah
b. Pengumpulan Data
c. Analisis
d. Perancangan
e. Implementasi
f. Pengujian
g. Penarikan kesimpulan

Gambaran tahapan yang dilakukan dapat dilihat pada Gambar 1



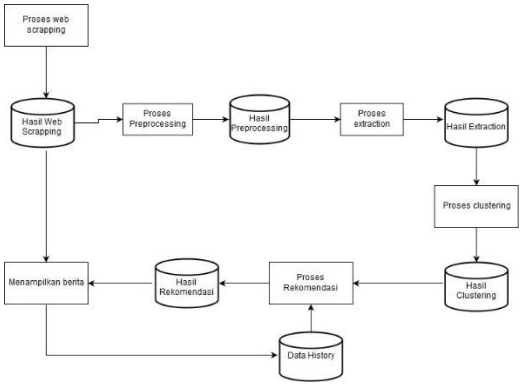
Gambar 1 Metode Penelitian

IV. ANALISIS

Tahapan ini dilakukan untuk mengetahui dan mengevaluasi seluruh komponen pada sistem yang akan dibangun.

a. Gambaran Umum Sistem

Gambaran umum sistem yang akan dibangun dapat dilihat pada gambar 2



Gambar 2 Gambaran Umum Sistem

b. Analisis Data Masukan

Data yang akan diolah untuk sistem ini adalah data berupa teks dimana sumber data didapat dari konten berita. Konten berita yang akan dijadikan data masukan tersebut dikumpulkan dari beberapa portal berita dengan menggunakan *web scrapper*. Data tersebut kemudian disimpan ke dalam database.

Data masukan lain yang digunakan adalah data riwayat berita yang dibaca oleh pembaca. Data ini nantinya akan digunakan sebagai acuan untuk rekomendasi berita yang diberikan oleh *news aggregator*.

c. Preprocessing

Pada tahapan *preprocessing* ini terdapat beberapa tahapan yang dilakukan sebelum data konten berita diproses dengan metode *cosine similarity*. Dalam Analisis proses *preprocessing* dibagi menjadi beberapa tahapan proses yaitu tahap *Cleaning*, *Case Folding*, *Tokenizing*, dan *Stopword Removal.*

1. **Cleaning**

   Pada tahap ini akan dilakukan proses penghapusan *tag markup* dan format khusus dari berita yang akan diolah. Konten berita pada *database* yang berasal dari hasil *scrapping* masih bersifat mentah dalam artian masih banyak *tag html* yang terdapat pada data. Oleh karena itu, pada tahapan ini data berita tersebut akan dibersihkan dari *tag html*. Pada tahapan ini semua tag html seperti <p>, , <a href=”#”> akan dihapus dari data.

2. **Case folding**

   Pada proses ini akan dilakukan pengubahan huruf besar menjadi huruf kecil atau disebut juga *case folding.* Tahap *case folding* adalah mengubah seluruh huruf dari “a” sampai dengan “z“ dalam dokumen menjadi huruf kecil. Tidak semua kata dalam dokumen konsisten menggunakan huruf kapital, maka dari itu *case folding* mengkonversi keseluruhan teks dalam dokumen menjadi huruf kecil.

4. **Tokenizing**

   Tahapan *Tokenizing* ini merupakan proses penguraian deskripsi yang semula berupa kalimat berisi kata-kata dan tanda pemisah antara kata seperti titik(.), koma(,), spasi ( ) dan tanda pemisah lain menjadi *token* atau potongan kata tunggal.

5. **Stopword removal**

   Tahap *stopword removal* merupakan proses mengambil kata -kata penting dari hasil *tokenizing.* Untuk melakukan tahap *stopword removal* ini bisa menggunakan algoritma *stopword removal* (membuang kata yang kurang penting). Metode ini dilakukan dengan cara menghilangkan kata tidak penting (*stopword*) pada dokumen melalui pengecekan kata-kata hasil *tokenizing* apakah termasuk di dalam daftar kata tidak penting atau tidak. Jika termasuk di dalam *stopword* maka kata-kata

tersebut akan di hilangkan dari dokumen sehingga kata-kata yang tersisa di dalam dokumen di anggap sebagai kata-kata penting.

d. Clustering

Proses clustering pada penelitian ini menggunakan metode *cosine similarity* dan *single pass clustering*. Metode ini dilakukan dengan tiga tahap, yaitu pembobotan TF-IDF, pencarian nilai *cosine similarity*, dan *clustering*. Berikut ini adalah tahap yang dilakukan untuk membuat clustering dokumen.

1 Tahap Pembobotan TF-IDF

- Tahap pertama adalah dicari dulu jumlah *TF* dan *DF* nya antara dua buah dokumen yang dibandingkan
- Tahap selanjutnya adalah mencari nilai *TF* dari masing masing dokumen.

$$TF(d,t) = \begin{cases} 0 & if\ freq(d,t) = 0 \\ 1 + \log(1 + \log(freq(d,t))) & otherwise. \end{cases}$$

TF(d,t) merupakan nilai *term d* dalam dokumen *t* . *freq(d,t)* merupakan jumlah *term* d dalam dokumen t.

- Kemudian setelah diketahui nilai *TF* dan *DF* nya maka dihitung nilai *IDF* nya dengan menggunakan persamaaan 2,

$$IDF(t) = \log \frac{1 + |d|}{|d_t|},$$

- Langkah terakhir adalah mencari nilai TF-IDFnya menggunakan persamaan 3,

$$TF\text{-}IDF(d,t) = TF(d,t) \times IDF(t).$$

2. Tahap Pencarian Nilai Cosine Similarity

- Tahap pertama adalah melakukan perhitungan untuk mencari nilai $v_1 v_2$, yaitu dengan menjumlahkan hasil perkalian dari nilai yang ada pada dokumen D1 (v_1) dan dokumen D2 (v_2).

$$sim(_{v1,v2}) = \frac{v_1 v_2}{|v_1|\,|v_2|}$$

$v_1 v_2$merupakan Jumlah kata yang ada pada dokumen D1 dan yang ada pada dokumen D2. $|v_2|$merupakan Jumlah kata yang ada pada dokumen D1. $|v_2|$ merupakan jumlah kata yang ada pada dokumen D2.

- Langkah selanjutnya adalah menetukan nilai $|v_1|$ dengan cara menghitung akar dari hasil perkalian dari nilai setiap kata yang ada pada D1.
- Kemudian langkah selanjutnya adalah menentukan nilai dari $|v_2|$ dengan cara menghitung akar dari hasil perkalian dari nilai setiap kata yang ada pada D2.
- Langkah terakhir adalah menentukan nilai *cosine similarity* dari nilai yang telah didapat,.

3. Tahap Clustering

- Yang dilakukan pada tahap ini adalah membandingkan nilai cosine similarity dengan nilai threshold yang ditentukan. Apabila nilainya melebihi nilai threshold maka menjadi satu cluster, apabila kurang dari nilai threshold maka menjadi cluster yang baru.

Ketiga tahap tersebut dilakukan berulang sampai semua dokumen berhasil dibandingkan dan semua dokumen masuk kedalam cluster.

e. Proses Rekomendasi Berita

News aggregator akan menampilkan semua data berita yang disimpan pada database. Pada saat pengguna memilih salah satu berita yang ditampilkan, maka sistem akan menyimpan pilhan berita yang diplih oleh pengguna. sistem kemudian akan menampilkan berita pada kolom rekomendasi dengan berita yang sesuai dengan *cluster* berita yang dipilih pengguna.

V. PENGUJIAN

Pengujian akurasi *cluster* dilakukan untuk mengetahui tingkat akurasi cluster yang terbentuk. Pengujian ini dilakukan dengan membandingkan hasil dari *cluster* yang terbentuk oleh sistem dengan pengelompokan data secara manual. Pengujian ini dilakukan dengan 157 data berita yang diambil menggunakan metode *scrapping* dari web www.kompas.com dan www.tribunnews.com.

Untuk mengevaluasi secara manual kesamaan diantara dokumen dalam *cluster* yang telah terbentuk digunakan standar yang dapat dilihat pada Tabel 1

Tabel 1 Kategori Hasil Klasifikasi

	In Event	Not In event
In cluster	a	b
Not In Cluster	c	d

Tabel diatas menunjukkan bahwa hasil klasifikasi bisa merupakan data yang benar (a) atau data yang salah (b). Sedangkan dokumen yang tidak

termasuk dalam hasil klasifikasi adakalanya memang data yang salah (d) dan adakalanya data yang benar tetapi tidak dalam cluster (c).

Menguji akurasi atau tingkat ketepatan hasil klasifikasi dari seluruh dokumen hasil klasifikasi dilakukan perhitungan dengan rumus p = a/(a+b) jika a+b > 0. Pengujian ini dilakukan dengan nilai threshold yang berbeda.

1. Pengujian pertama dengan threshold 0.3

   Pengujian pertama ini dilakukan dengan menentukan threshold dari cosine similaritynya adalah 0.3. Pada pengujian ini, jumlah cluster yang terbentuk berjumlah 118 buah *cluster* dengan *cluster* yang paling banyak memiliki 4 anggota.

   Hasil pengujian dengan threshold 0.3 dapat dilihat pada Tabel 2

Tabel 2 Hasil Pengujian Dengan Threshold 0.3

Jumlah Cluster Yang Terbentuk	Jumlah Data Dalam Cluster	Total Data	Data Benar	Data Salah
104	1	104	104	0
18	2	36	36	0
3	3	9	9	0
2	4	8	8	0
118		157	157	0

Hasil pengujian 157 data dengan threshold 0.3 menghasilkan 157 data benar, sehingga akurasinya adalah 100%.

2. Pengujian kedua dengan threshold 0.2

   Hasil pengujian dengan threshold 0.2 dapat dilihat pada Tabel 3

Tabel 3 Hasil Pengujian Dengan Threshold 0.2

Jumlah Cluster Yang Terbentuk	Jumlah Data Dalam Cluster	Total Data	Data Benar	Data Salah
85	1	85	85	0
14	2	28	28	0
8	3	24	25	0
2	4	8	8	0
1	5	5	5	0
1	7	7	7	0
111		157	157	0

Hasil pengujian 157 data dengan threshold 0.2 menghasilkan 157 data benar, sehingga akurasinya adalah 100%.

3. Pengujian ketiga dengan threshold 0.1

   Hasil pengujian dengan threshold 0.1 dapat dilihat pada Tabel 4

Tabel 4 Hasil Pengujian Dengan Threshold 0.1

Jumlah Cluster Yang Terbentuk	Jumlah Data Dalam Cluster	Total Data	Data Benar	Data Salah
68	1 -5	100	100	0
3	6 -10	24	22	5
1	11 - 15	13	13	0
1	16 - 20	20	15	5
73		157	147	10

Hasil pengujian 157 data dengan threshold 0.1 menghasilkan 147 data benar dan 20 data yang salah, sehingga akurasinya adalah 93%.

4. Pengujian keempat dengan threshold 0.07

   Hasil pengujian dengan threshold 0.07 dapat dilihat pada Tabel 5

Tabel 5 Hasil Pengujian Dengan Threshold 0.07

Jumlah Cluster Yang Terbentuk	Jumlah Data Dalam Cluster	Total Data	Data Benar	Data Salah
44	1 - 5	67	67	0
5	6 - 10	41	28	13
1	11 - 15	13	13	0
2	16 - 20	36	29	7
52		157	137	20

Hasil pengujian 157 data dengan threshold 0.07 menghasilkan 137 data benar dan 20 data yang salah, sehingga akurasinya adalah 87%.

5. Pengujian kelima dengan threshold 0.05
   Hasil pengujian dengan threshold 0.05 dapat dilihat pada Tabel 6

Tabel 6 Hasil Pengujian Dengan Threshold 0.05

Jumlah Cluster Yang Terbentuk	Jumlah Data Dalam Cluster	Total Data	Data Benar	Data Salah
21	1 - 5	36	34	2
5	6 - 10	35	29	6
2	11 - 15	23	15	8
1	16 - 20	16	6	10
1	> 20	47	17	30
30		157	101	56

Hasil pengujian 157 data dengan threshold 0.05 menghasilkan 101 data benar dan 56 data yang salah, sehingga akurasinya adalah 64%.

Dari lima kali pengujian pengujian dengan nilai threshold yang berbeda tersebut, hasil pengujiannya dapat disederhanakan seperti dilihat pada Tabel 7

Tabel 7 Hasil Pengujian

Threshold	Jumlah cluster	Akurasi
0.3	118	100%
0.2	111	100%
0.1	73	93%
0.07	52	87%
0.05	30	64%

Dari hasil pengujian di atas dapat dilihat bahwa nilai threshold sangat mempengaruhi jumlah kluster yang terbentuk. Semakin besar nilai threshold yang ditentukan, maka akurasinya sangat besar bisa mencapai 100% tetapi anggota clusternya sangat sedikit. Sebaliknya, dengan semakin kecil nilai threshold yang ditentukan maka anggota kluster yang dibentuk bisa semakin banyak dan cluster yang terbentuk lebih sedikit tetapi mengurangi dan dapat menyebabkan adanya beberapa data yang tidak seharusnya ada dalam suatu cluster. Rata rata akurasi yang didapatkan dari hasil pengujian adalah 88.8%.

VI. KESIMPULAN

Berdasarkan hasil implementasi dan pengujian yang dilakukan terhadap sistem *news aggregator* dengan menerapkan *metode cosine similarity* dan *single pass clustering*, maka dapat disimpulkan bahwa metode yang digunakan berhasil mengelompokkan berita berdasarkan isi dari beritanya dengan akurasi rata-rata sebesar 88.8%. Cluster yang dihasilkan memiliki tingkat akurasi berbeda tergantung dari threshold yang ditentukan. Meskipun begitu, metode yang digunakan masih memiliki kelemahan. Dengan menggunakan metode ini masih terdapat beberapa berita yang seharusnya tidak ada dalam satu kelompok dikarenakan metode ini hanya memeriksa kesamaan teks dan bukan topik nya.

REFERENSI

[1] Daniel Waegel. 2006. The Development Of Text-Mining Tools And Algorithms. Thesis. Ursinus College.

[2] Lokesh Kumar, Parul Kalra Bhatia. 2013. *Text Mining*: Concepts, Process and Applications. *Journal of Global Research in Computer Science*. 4(3): 36-39

[3] Husni, Y.D.P. Negara,M. Syarief. 2015. *Cluster*isasi Dokumen Web (Berita) Bahasa Indonesia Menggunakan Algoritma K-Means. *Jurnal SimanteC*. 4(3): 159:166

[4] NI Widiastuti, E Rainarli, KE Dewi. 2017. Peringkasan dan Support Vector Machine pada Klasifikasi Dokumen. *Jurnal Infotel*. 9(4): 416-421

[5] AT Baidowi, NI Widiastuti. *2015.* Web Content Mining Menggunakan Partitional *Cluster*ing K-Means Pada News Aggregator. *Jurnal Sistem Komputer.* 5(2): 42-46

[6] Ferbruariyanti Henry, Zuliarso Eri. 2012. Algoritma *Single Pass Clustering Untuk Klastering* Halaman Web. *Prosiding Seminar Nasional Komputer dan Elektro (SENOPUTRO)* 1: 1-8

[7] Sugiyono. 2010. *Metode Penelitian Kuantitatif dan Kualitatif dan R & D.* Bandung: CV Alfabeta.

[8] M. Romli. 2012. *Jurnalistik Online: Panduan Mengelola Media Online* Bandung: Nuansa.

[9] Google Inc, "Google," 2010. [Online]. Available: http://www.google.com/webmasters/docs/s

earch-engine-optimization-starter-guide.pdf
[10] Angela M. Lee. 2015. The Rise of Online News Aggregator : Consumption and Competition. *International Journal on Media Management*. 00:1–22
[11] O. Maimon dan L. Rokach. 2010. *Data Mining and Knowledge Discovery Handbook.* New York: Springer-Verlag New York Incorporated
[12] J Han, M Kamber, 2006. *Data Mining Concepts and Techniques (2nd Edition).* Massachusetts: Morgan Kaufmann
[13] Salton, G., 1989. *Automatic Text Processing, The Transformation, Analysis, and Retrieval of Information by Computer*. Boston: Addison – Wesly Publishing Company, Inc. All rights reserved.
[14] Anhar. 2010. *PHP & MySql Secara Otodidak*. Jakarta: PT TransMedia
[15] Alexander F. K. Sibero. 2011. Kitab Suci Web Programing. Yogyakarta: MediaKom.