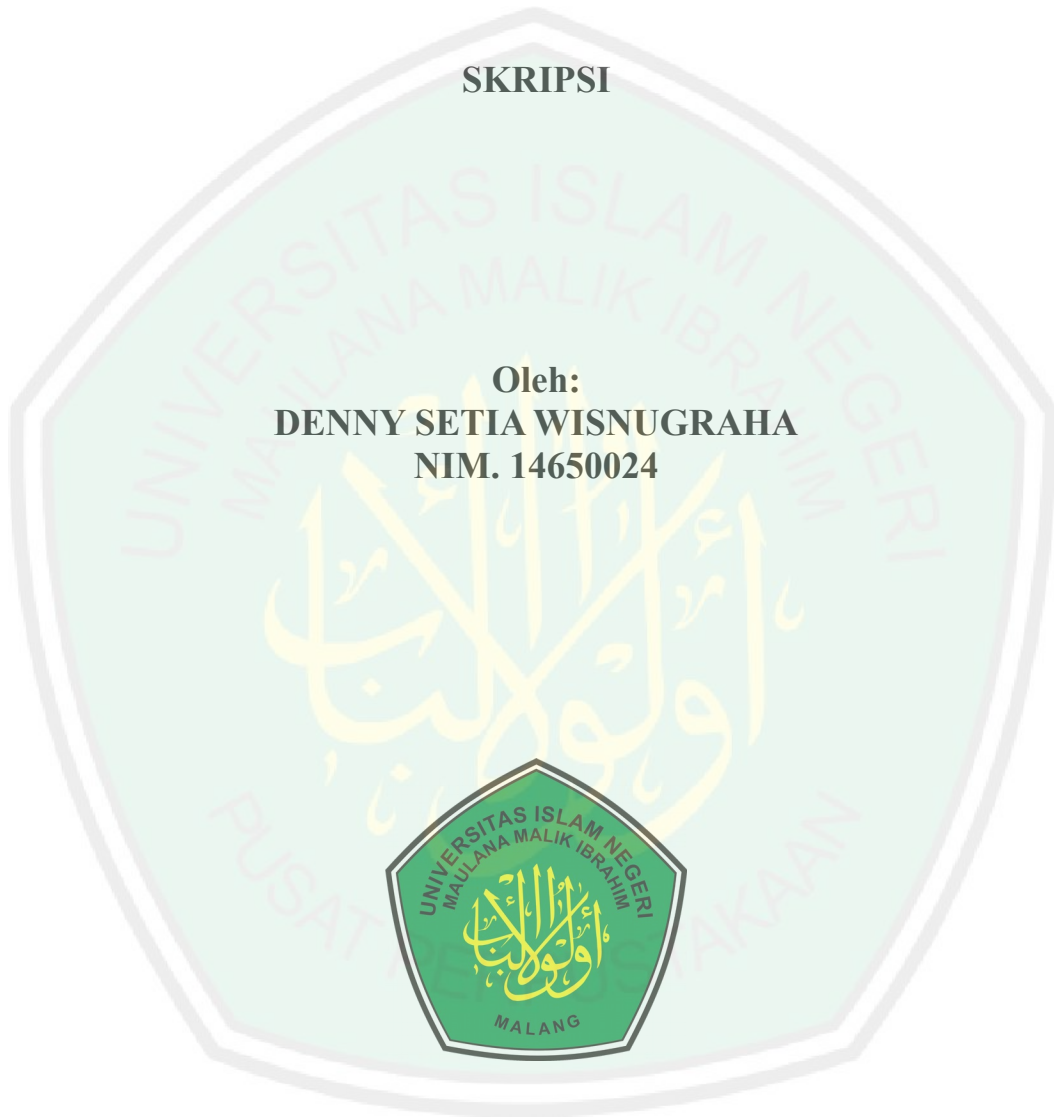


**INFORMATION RETRIEVAL SYSTEM (IRS) BERITA JAWA TIMUR
DENGAN METODE *SELF-ORGANIZING MAP* (SOM)**

SKRIPSI

Oleh:
DENNY SETIA WISNUGRAHA
NIM. 14650024



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2019**

**INFORMATION RETRIEVAL SYSTEM (IRS) BERITA JAWA TIMUR
DENGAN METODE *SELF-ORGANIZING MAP* (SOM)**

SKRIPSI

**Diajukan Kepada:
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh:
DENNY SETIA WISNUGRAHA
NIP. 14650024**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2019**

LEMBAR PERSETUJUAN

**INFORMATION RETRIEVAL SYSTEM (IRS) BERITA JAWA TIMUR
DENGAN METODE SELF-ORGANIZING MAP (SOM)**

SKRIPSI


Oleh:
DENNY SETIA WISNUGRAHA
NIM. 14650024


Telah Diperiksa dan Disetujui untuk Diuji:

Tanggal: *26 November 2018*

Pembimbing I

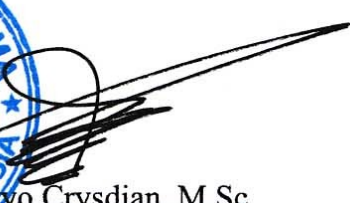
Pembimbing II


Dr. Cahyo Crysdian, M.Sc
NIP. 19740424 200901 1 008


Mochamad Imamudin, Lc., MA
NIP. 19740602 200901 1 010

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang




Dr. Cahyo Crysdian, M.Sc
NIP. 19740424 200901 1 008

HALAMAN PENGESAHAN

INFORMATION RETRIEVAL SYSTEM (IRS) BERITA JAWA TIMUR DENGAN METODE *SELF-ORGANIZING MAP* (SOM)

SKRIPSI

Oleh:
DENNY SETIA WISNUGRAHA
NIM. 14650024

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)

Tanggal: 12 Desember 2018

Susunan Dewan Penguji

Penguji Utama : Irwan Budi Santoso, M.Kom
NIP. 19770103 201101 1 004

Ketua Penguji : Dr. M. Amin Hariyadi, M.T
NIP. 19670118 200501 1 001

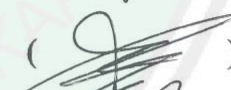
Sekretaris Penguji : Dr. Cahyo Crysdiyan, M.Sc
NIP. 19740424 200901 1 008

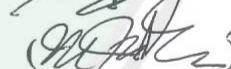
Anggota Penguji : Mochamad Imamudin, Lc., MA
NIP. 19740602 200901 1 010

Tanda Tangan

()

()

()

()

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang




Dr. Cahyo Crysdiyan, M.Sc
NIP. 19740424 200901 1 008

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini:

Nama : Denny Setia Wisnugraha

NIM : 14650024

Fakultas/Jurusan : Sains dan Teknologi / Teknik Informatika

Judul Penelitian : *Information Retrieval System (IRS) Berita Jawa Timur*
dengan Metode *Self-Organizing Map (SOM)*

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila dikemudian hari terbukti atau dapat dibuktikan Skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 26 November 2018

Yang Membuat Pernyataan,



Denny Setia Wisnugraha
NIM. 14650024

MOTTO

Dalam setiap melakukan suatu hal, maka lakukan 4 hal berikut:

“Husnuzan, Ikhtiar-kan, Tawakal, Berdoa”



PERSEMBAHAN



Assalamu 'alaykum Warahmatullahi Wabarakatuh

Alhamdulillahirabbil 'alamin

Puji syukur kepada Allah Yang Maha Esa atas limpahan rahmat, taufik, dan hidayah-Nya kepada kita semua, Dzat yang menguasai seluruh alam dan seisinya. Sholawat dan salam kita haturkan kepada Nabi Muhammad SAW, keluarganya, dan para sahabatnya. Semoga kita termasuk umat beliau yang taat sehingga kita termasuk orang-orang yang *husnul khatimah* dan ditetapkan nikmat iman dan nikmat Islam didalam hati kita, *aamiin*.

Kupersembahkan karya sederhana ini kepada orang yang kusayangi dan kuharapkan ridhonya yaitu bapakku Ali Wiratno dan ibuku Warti untuk sebagian baktiku kepada beliau. Semoga kebaikan, rahmat, dan perlindungan Allah SWT senantiasa tercurahkan kepada beliau.

Kupersembahkan juga karya sederhana ini kepada kakakku Nur Hadi Setianto, adikku Bayu Setia Rama, dan teman-temanku semua. Terima kasih banyak atas do'a, dukungan dan motivasi yang telah diberikan sehingga dapat terselesaikannya penelitian ini.

Wassalamu 'alaykum Warahmatullahi Wabarakatuh

Denny Setia Wisnugraha

KATA PENGANTAR

الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ وَالصَّلَاةُ وَالسَّلَامُ عَلَى أَشْرَفِ الْأَنْبِيَاءِ وَالْمُرْسَلِينَ وَعَلَى آلِهِ وَصَحْبِهِ أَجْمَعِينَ أَمَا بَعْدُ

Segala puji penulis haturkan kepada Allah Yang Maha Esa yang telah memberikan limpahan rahmat, taufiq dan hidayah sehingga penulis dapat menyelesaikan skripsi dengan judul “*Information Retrieval System (IRS) Berita Jawa Timur dengan Metode Self-Organizing Map (SOM)*” ini.

Dalam penulisan skripsi ini penulis banyak memperoleh bantuan dari banyak pihak. Baik berupa motivasi, bimbingan, kritik dan saran yang membangun. Oleh karena itu penulis mengucapkan terima kasih juga kepada:

1. Bapak Dr. Cahyo Crysdiان selaku dosen pembimbing I dan Ketua Jurusan Teknik Informatika Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim Malang.
2. Bapak Mochamad Imamudin, Lc., MA selaku dosen pembimbing II.
3. Seluruh dosen dan staf Jurusan Teknik Informatika Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim Malang yang telah memberikan bekal ilmu yang dapat penulis gunakan untuk menyelesaikan skripsi ini.
4. Bapak dan ibu serta keluarga tercinta yang selalu memberikan doa, motivasi, dan pendidikan yang sangat berharga bagi penulis.
5. Teman-teman semua yang juga telah memberikan doa, dukungan, dan informasi sehingga penulis dapat menyelesaikan skripsi ini.

Semoga semua yang telah diberikan mereka kepada penulis, menjadi amalan yang diterima oleh Allah SWT dan Allah membalasnya dengan sesuatu yang lebih baik. Penulis menyadari bahwa dalam skripsi ini juga masih banyak kekurangannya. Oleh karena itu penulis dengan senang hati menerima kritik dan saran yang membangun dari para pembaca. Akhir kata, semoga apa yang kita kerjakan mendapat ridho dari Allah SWT.

Malang, 20 November 2018

Penulis



DAFTAR ISI

HALAMAN JUDUL.....	ii
LEMBAR PERSETUJUAN.....	iii
HALAMAN PENGESAHAN.....	iv
PERNYATAAN KEASLIAN TULISAN.....	v
MOTTO.....	vi
PERSEMBAHAN.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiii
ABSTRAK.....	xiv
ABSTRACT.....	xv
المخلص.....	xvi
BAB I.....	1
1.1. Latar Belakang.....	1
1.2. Pernyataan Masalah.....	3
1.3. Tujuan Penelitian.....	3
1.4. Batasan Penelitian.....	3
BAB II.....	4
2.1. <i>Information Retrieval System (IRS)</i>	4
2.2. <i>Self-Organizing Map (SOM)</i>	6
BAB III.....	9
3.1. Desain Sistem.....	9
3.1.1 Pengumpulan Data.....	11
3.1.2 <i>Case Folding</i>	13
3.1.3 Tokenizing.....	13
3.1.4 Stemming.....	14
3.1.5 <i>Stopword Removal</i>	17

3.1.6 Pembobotan Kata atau <i>Term</i>	18
3.1.7 Normalisasi.....	21
3.1.8 <i>Cosine Similarity</i>	22
3.1.9 <i>Self-Organizing Map (SOM)</i>	25
3.1.10 <i>Desain Interface</i>	28
3.2. Implementasi.....	30
3.2.1 Pengumpulan Data.....	31
3.2.2 <i>Case Folding</i>	31
3.2.3 Tokenizing.....	31
3.3.4 Stemming.....	32
3.3.5 Stopword Removal.....	33
3.3.6 Pembobotan Kata atau <i>Term</i>	34
3.3.7 Normalisasi.....	37
3.3.8 <i>Cosine Similarity</i>	38
3.3.9 <i>Self-Organizing Map</i>	41
3.3.10 Implementasi GUI.....	50
BAB IV.....	52
4.1 Langkah-Langkah Uji Coba.....	52
4.2 Hasil Uji Coba.....	60
4.3 Pembahasan.....	73
BAB V.....	81
5.1. Kesimpulan.....	81
5.2. Saran.....	81
DAFTAR PUSTAKA.....	82

DAFTAR TABEL

Tabel 3.1. Hasil DF dan IDF.....	36
Tabel 3.2. Hasil TF-IDF dan Pencarian Nilai Minimal Maksimal.....	37
Tabel 3.3. Hasil Normalisasi Min-Max.....	38
Tabel 3.4. Hasil TF, DF dan IDF.....	39
Tabel 3.5. Hasil TF-IDF dan Pencarian Maksimal Minimal.....	40
Tabel 3.6. Hasil Normalisasi.....	40
Tabel 3.7. Hasil Cosine Similarity.....	41
Tabel 3.8. Inisialisasi Bobot Cluster.....	42
Tabel 3.9. Bobot Dokumen.....	42
Tabel 3.10. Bobot Query Pada Iterasi 1.....	43
Tabel 3.11. Bobot Dokumen 1 Pada Iterasi 1.....	44
Tabel 3.12. Bobot Dokumen 2 Pada Iterasi 1.....	45
Tabel 3.13. Bobot Dokumen 3 Pada Iterasi 1.....	46
Tabel 3.14. Bobot Dokumen 4 Pada Iterasi 1.....	47
Tabel 3.15. Bobot Dokumen Pada Iterasi 2.....	48
Tabel 3.16. Hasil Cluster dan Manual.....	49
Tabel 4.1. List Judul Dokumen Berita.....	53
Tabel 4.2. Hasil Cluster dengan Range Nilai Min-Max 0 s/d 1.....	60
Tabel 4.3. Hasil Cluster dengan Range Nilai Min-Max 0 s/d 9.....	64
Tabel 4.4. Confusion Matrix.....	68
Tabel 4.5: Evaluasi Precision, Recall dan F-Measure.....	69

DAFTAR GAMBAR

Gambar 3.1. Desain Proses.....	11
Gambar 3.2. Flowchart Pengumpulan Data.....	12
Gambar 3.3. Flowchart Case Folding.....	13
Gambar 3.4. Flowchart Tokenizing.....	14
Gambar 3.5. Flowchart Stopword Removal.....	17
Gambar 3.6. Flowchart TF.....	19
Gambar 3.7. Flowchart DF.....	20
Gambar 3.8. Flowchart Cosine Similarity.....	24
Gambar 3.9. Arsitektur SOM.....	25
Gambar 3.10. Flowchart SOM.....	27
Gambar 3.11. Interface Hasil Pencarian.....	28
Gambar 3.12. Interface Menu Get Data.....	29
Gambar 3.13. Interface Menu Pre-processing.....	30
Gambar 3.14. Interface Menu Count Term.....	30
Gambar 3.15. Tokenizing.....	32
Gambar 3.16. Stemming.....	33
Gambar 3.17. Stopword Removal.....	34
Gambar 3.18. Pembobotan TF.....	35
Gambar 3.19. Proses Query Sampai Pembobotan TF.....	38
Gambar 3.20. Implementasi GUI.....	50
Gambar 4.1. Ilustrasi Keterkaitan Precision dan Recall.....	59
Gambar 4.2. Hubungan Precision, Recall dan Akurasi.....	75

ABSTRAK

Wisnugraha, Denny Setia. 2018. *Information Retrieval System (IRS) Berita Jawa Timur dengan Metode Self-Organizing Map (SOM)*. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. Cahyo Crys dian, M.Sc, (II) Muhammad Imamuddin, Lc., MA

Kata Kunci: Temu Kembali Informasi, Self Organizing Map, Cosine Similarity, Berita Jawa Timur

Information Retrieval System (IRS) merupakan sistem yang membantu pengguna untuk menemukan suatu informasi yang relevan sesuai kebutuhan pengguna. Dalam menemukan informasi berita Jawa Timur yang relevan, digunakan metode pembobotan dokumen berita menggunakan TF-IDF. Kemudian hasil pembobotan dinormalisasikan menggunakan metode *Min-Max*, lalu metode *Self-Organizing Map* untuk meng-*cluster* kategori berita. Untuk memaksimalkan sistem temu kembali informasi pada penelitian ini digunakan perhitungan *cosine similarity* untuk menghitung kemiripan *query* dengan dokumen berita dan untuk memberi ranking pada dokumen berita. Hasil pengujian sistem temu kembali informasi dengan menggunakan 120 dokumen berita yang terdiri dari 40 berita tentang wisata, 40 berita tentang pendidikan dan 40 berita tentang bencana diperoleh nilai rata-rata *precision* 83,31%, *recall* 90,56%, *f-measure* 86,40% dan akurasi sebesar 90,19%.

ABSTRACT

Wisnugraha, Denny Setia. 2018. **Information Retrieval System (IRS) East Java News with Self-Organizing Map (SOM) Method**. Thesis. Department of Informatics Engineering. Faculty of Sains and Technology. Maulana Malik Ibrahim State Islamic University of Malang. Advisors: (I) Dr. Cahyo Crysdian, M.Sc, (II) Muhammad Imamuddin, Lc., MA

Keywords: Information Retrieval System, Self Organizing Map, Cosine Similarity, East Java News

The Information Retrieval System (IRS) is a system that helps users to find relevant information according to user needs. In finding relevant East Java news information, the method used to weight news documents using TF-IDF. Then the weighing results are normalized using the Min-Max method, then the Self-Organizing Map method to cluster news categories. To maximize the information retrieval system in this study, cosine similarity calculations were used to calculate query similarities with news documents and to rank news documents. The results of the information retrieval system test using 120 news documents consisting of 40 news about travel, 40 news about education and 40 news about disasters obtained an average precision value of 83,31%, recall 90,56%, f-measure 86,40% and accuracy of 90,19%.

الملخص

ويسنوغراها، داني ستيا. 2018. نظام بحث المعلومات (*Information Retrieval System (IRS)* عن أخبار جاوى الشرقية بمنهج نظام الخريطة النفسى (*Self-Organizing Map (SOM)*). البحث الجامعي. قسم التقنية المعلوماتية كلية العلوم والتكنولوجيا جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف: (1) الدكتور جاحيو جريسديان الماجستير (2) محمد إمام الدين الماجستير

الكلمات الرئيسية: نظام بحث المعلومات، نظام الخريطة النفسى، تسوية جيب التمام/Cosine Similarity، أخبار جاوى الشرقية

نظام بحث المعلومات هو نظام يساعد مستخدمه في تحصيل المعلومات الوثيقة على حسب حاجة المستخدم. في تحصيل المعلومات عن أخبار جاوى الشرقية الوثيقة، أستخدم منهج تزويد ملفات الأخبار بمنهج TF-IDF. ثم يجري التطبيع بمنهج Min-Max ومنهج *Self-Organizing Map* لتعيين صنف الأخبار. لإفادة نظام بحث المعلومات أكثر في هذا البحث أستخدم حساب تسوية جيب التمام لحساب موافقة تقاريرات بملفات الأخبار وإعطاء الترتيب في ملفات الأخبار. نتيجة اختبار نظام بحث المعلومات باستخدام 120 ملف أخبار يتكون من 40 خبرا عن السياحة و40 خبرا عن التربية و40 خبرا عن المصيبة. فتحصل نتيجة $precision\ 83,31\%$ ، $recall\ 90,56\%$ ، $f-measure\ 86,40\%$ ، ودقة $90,19\%$.

BAB I

PENDAHULUAN

1.1. Latar Belakang

Dalam suatu daerah setiap harinya pasti mengalami suatu kejadian yang bermacam-macam seperti fenomena alam dan kegiatan masyarakat. Beberapa kejadian tersebut ada yang diliput oleh wartawan yang kemudian diberitakan kepada masyarakat dalam bentuk media cetak, radio, televisi maupun media *online*. Contoh dari situs berita atau media *online* yang terkenal dan terpercaya di Indonesia adalah news.detik.com yang setiap harinya mem-*posting* lebih dari 70 berita, baik berita luar negeri maupun berita dalam negeri.

Dengan membaca berita-berita, kita dapat menambah wawasan atau pengetahuan, meningkatkan kosa kata, meningkatkan keterampilan komunikasi atau mengasah kecerdasan. Dari sebuah berita juga dapat diketahui keadaan suatu daerah, sehingga dapat digunakan sebagai acuan untuk pembangunan atau pengembangan suatu daerah yang lebih baik.

Tentang menyampaikan suatu berita yang benar juga dijelaskan dalam ayat Al-Qur'an pada surat An-Nissa: 83, sebagai berikut:

وَإِذَا جَاءَهُمْ أَمْرٌ مِنَ الْأَمْنِ أَوْ الْخَوْفِ أَذَاعُوا بِهِ ۖ وَلَوْ رَدُّوهُ إِلَى الرَّسُولِ وَإِلَى أُولِي الْأَمْرِ مِنْهُمْ لَعَلِمَهُ الَّذِينَ يَسْتَنْبِطُونَهُ مِنْهُمْ ۗ وَلَوْلَا فَضْلُ اللَّهِ عَلَيْكُمْ وَرَحْمَتُهُ لَاتَّبَعْتُمُ الشَّيْطَانَ إِلَّا قَلِيلًا (٨٣)

Artinya: “Dan apabila sampai kepada mereka suatu berita tentang keamanan ataupun ketakutan, mereka (langsung) menyiarkannya. (Padahal) apabila mereka menyerahkannya kepada Rasul dan *ulil amri* di antara mereka, tentulah orang-orang yang ingin mengetahui kebenarannya (akan dapat) mengetahuinya (secara resmi) dari mereka (Rasul dan *ulil amri*). Sekiranya bukan karena karunia dan rahmat Allah kepadamu, tentulah kamu mengikuti setan kecuali sebagian kecil saja (di antara kamu).”

Jawa Timur merupakan salah satu provinsi yang ada di Indonesia. Dalam situs id.wikipedia.org Jawa Timur terdiri atas 29 kabupaten dan 9 kota dengan ibu kotanya adalah Surabaya. Jawa Timur memiliki total luas 47.922 km² dan jumlah penduduk pada tahun 2015 adalah 42.030.633 jiwa dengan kepadatan 884 jiwa/km². Jawa Timur juga memiliki bermacam-macam pariwisata terkenal seperti Gunung Bromo, Pantai Prigi, wahana permainan Jatim Park, Situs Trowulan di Mojokerto yang dulunya merupakan pusat kerajaan Majapahit, Air Terjun Madakaripura di Probolinggo, Telaga Sarangan di gunung Lawu Magetan, terdapat juga sejumlah makam para Wali yang menjadi wisata religi para peziarah bagi umat Islam dan masih banyak lagi wisata Jawa Timur yang lainnya.

Dari uraian diatas dan berdasarkan visi kominfo provinsi Jawa Timur yaitu terwujudnya penyebarluasan informasi dan pelayan publik melalui TIK di Jawa Timur serta misi kominfo provinsi Jawa Timur yaitu meningkatkan kapasitas layanan penyebaran informasi, memberdayakan potensi masyarakat serta kerjasama lembaga komunikasi dan informatika serta mengembangkan aplikasi, muatan layanan publik, standarisasi penyelenggaraan pos dan telekomunikasi

serta pemanfaatan jaringan TIK dalam rangka peningkatan pelayanan publik, peneliti ingin membuat suatu *Information Retrieval System* (IRS) seputar berita di Jawa Timur dengan metode *Self-Organizing Map* (SOM). SOM merupakan metode berbasis *unsupervised learning* yang mampu memetakan data berdimensi tinggi ke dalam bentuk peta berdimensi rendah dengan mempertahankan topologi data tersebut.

1.2. Pernyataan Masalah

Identifikasi masalah dari penelitian ini adalah seberapa tinggi *precision*, *recall*, dan *F-measure* untuk menemukan kembali informasi yang relevan dengan *query user* jika digunakan metode SOM?

1.3. Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mengukur tingkat *precision*, *recall*, dan *F-measure* dalam menemukan kembali informasi yang relevan dengan *query user* yang menggunakan metode *Self-Organizing Map* (SOM).

1.4. Batasan Penelitian

1. Query atau inputan dari user menggunakan Bahasa Indonesia.
2. Dokumen berita diambil dari 4 situs berita *online* yang sudah terverifikasi administrasi dan faktual oleh Dewan Pers, yaitu: detik.com, sindonews.com, okezone.com dan jawapos.com
3. Berita yang digunakan adalah berita seputar daerah Jawa Timur yang terbit pada tahun 2018.

BAB II

STUDI PUSTAKA

2.1. *Information Retrieval System (IRS)*

Lestari (2016) dalam penelitiannya untuk mencoba mengetahui efektivitas OPAC Perpustakaan ITS Surabaya menggunakan pendekatan *recall* dan *precision*, menyebutkan pembuatan sistem temu kembali informasi (*Information Retrieval System*) sudah ada sejak tahun 1908 oleh *United Kingdom* dan *United States* dimulai dengan menghasilkan *Anglo-America Catalog Rule (AACR)*. Salah satu penerapan prinsip relevansi yang sejak dulu digunakan dalam pengembangan sistem IR adalah penggunaan ukuran *recall* dan *precision*.

Cholifah, Purwananto dan Bramantoro (2006) menjelaskan dalam penelitiannya bahwa dalam sistem IR, pengguna menterjemahkan kebutuhan informasinya kedalam bentuk kata kunci (*keyword*), lalu *keyword* ini oleh sistem IR akan diterjemahkan menjadi suatu *query*, dengan *query* ini maka sistem akan melakukan pencarian kedalam kumpulan dokumen. Kualitas informasi yang dihasilkan dari sistem IR dipengaruhi oleh penyajian dokumen secara logikal oleh sistem dan perilaku pengguna yang menentukan kata kunci yang sesuai dengan informasi yang diinginkan.

Mas'udia, Atmaja dan Mustafa (2017) dalam penelitiannya menjelaskan bahwa untuk dapat melakukan pencarian berdasarkan substansi yang paling mirip, dapat menggunakan teknologi yang disebut *information Text Retrieval*.

Penelitiannya mengembangkan sistem temu kembali informasi judul tugas akhir dan perhitungan kemiripan dokumen menggunakan *vector space model*. Sistem secara *offline* otomatis melakukan *indexing* dan temu kembali (*retrieval*) secara *real time*. Proses *retrieval* dimulai dengan mengambil *query* dari pengguna, menerapkan *stop word removal* sehingga dihasilkan *keyword* yang *compact* tetapi dapat mewakili *query* tersebut, kemudian sistem menghitung kemiripan antara *keyword* dengan daftar dokumen yang diwakili oleh term-term di dalam *index*. Dokumen ditampilkan dan diurutkan berdasarkan dokumen yang paling mirip.

Sukma, Zaman dan Purwanti (2015) meneliti untuk mengetahui bagaimana penerapan sistem pencarian informasi dalam klasifikasi jurnal dengan menggunakan kesamaan kosinus dan K-Nearest Neighbor (KNN) untuk mengklasifikasikan dokumen. Dengan menggunakan *cosine similarity*, klasifikasi dokumen akan lebih mudah dilakukan, *cosine similarity* akan mengolah dengan merubah dokumen tersebut menjadi vektor dan melakukan perbandingan antara vektor dokumen tersebut sehingga menghasilkan suatu nilai similaritas antar dokumen. Kemudian metode *k-nearest neighbor* dapat melakukan klasifikasi terhadap dokumen-dokumen yang telah menghasilkan nilai similaritas. Sebelum dilakukan klasifikasi dengan *k-nearest neighbor* dilakukan sorting dari nilai similaritas berdasarkan nilai similaritas terbesar hingga terkecil. Berdasarkan nilai “k” yang telah diinisialisasi pengguna, dapat dilihat mayoritas kategori dokumen training yang muncul dalam lingkup nilai “k” sebagai dasar untuk melakukan klasifikasi terhadap dokumen.

2.2. *Self-Organizing Map (SOM)*

Tan (2006) menjelaskan *clustering* adalah sebuah proses untuk mengelompokkan data ke dalam beberapa *cluster* atau kelompok sehingga data dalam satu *cluster* memiliki tingkat kemiripan yang maksimum dan data antar *cluster* memiliki kemiripan yang minimum.

Menurut Correa & Pinheiro (2010) dalam penelitiannya yang mengusulkan penerapan *Self-Organizing Map (SOM)* dalam pembangunan *Information Retrieval Systems (IRS)* untuk Digital Library of Theses and Dissertations at Federal University of Pernambuco (BDTDUFPE), *Information Retrieval Systems (IRS)* yang pada dasarnya tentang pengindeks-an, pencarian dan klasifikasi dokumen text untuk memenuhi kebutuhan informasi pengguna, biasanya diekspresikan melalui *query* yang terdiri dari kata kunci atau ekspresi sederhana yang berkaitan dengan query tersebut.

Metode *Self-Organizing Map (SOM)* yang juga disebut Jaringan Kohonen merupakan suatu metode Jaringan Syaraf Tiruan yang diperkenalkan oleh Professor Teuvo Kohonen pada tahun 1980-an, yang mempunyai bentuk *Unsupervised Artificial Neural Network (Unsupervised ANN)* atau *unsupervised learning* yang pelatihannya tidak memerlukan pengawasan atau target keluaran, sehingga dari data yang ada bisa mengelompokkan data tersebut menjadi 2 bagian atau 3 bagian dan seterusnya, beda halnya dengan *supervised learning* yang merupakan sebuah pendekatan dimana sudah terdapat variabel yang ditargetkan sehingga tujuan dari pendekatan ini adalah mengelompokkan suatu data ke data yang sudah ada (Chandra, 2017, *Perbedaan Supervised And Unsupervised*

Learning, https://www.datascience.or.id/detail_artikel/52/supervised-and-unsupervised-learning, diakses pada tanggal 10 Mei 2018).

Azmi (2014) dalam penelitiannya “Komparasi Metode Jaringan Syaraf Tiruan SOM dan LVQ untuk Mengidentifikasi Data Bunga Iris” Menghasilkan, SOM mempunyai performa lebih baik (89.33%) dari pada LVQ (86.67%), tetapi LVQ mempunyai proses waktu yang lebih cepat (0.11 s) dari pada SOM (0.20 s). SOM terdiri dari 2 lapis, yaitu input dan output. SOM menganalisis data dimensi tinggi ke dimensi 2 dengan mempertahankan topologi data tersebut. SOM merupakan pengembangan dari Jaringan Syaraf Tiruan (JST). Bobot (W) digunakan untuk menentukan jarak terhadap output layer. Jarak terdekat akan menjadi referensi pengklasteran data. Apabila ada jarak ke output 2 adalah yang paling dekat maka vektor yang dimaksud akan ditempatkan di kelas output ke 2, dan kemudian bobot yang ada akan diperbarui dengan mempertimbangkan masuknya vektor masukan tersebut ke *output* layer kelas 2. Bobot baru ini kemudian akan menjadi referensi berikutnya untuk masukan selanjutnya.

Anis & Isnanto (2014) dalam penelitiannya menggunakan metode SOM untuk melakukan klasterisasi Data Pemilih Tetap (DPT). Hasil klasterisasi kemudian digunakan sebagai dasar untuk menentukan kriteria warna dalam pengembangan aplikasi Web-GIS berdasarkan interval jumlah pemilih. SOM menyediakan teknik visualisasi data yang membantu memahami data yang memiliki dimensi yang kompleks dengan mengurangi dimensi data kedalam *map* atau peta.

Pada arsitektur SOM, unit yang bersebelahan akan memiliki lebih sedikit perbedaan daripada unit yang letaknya lebih jauh. Bobot (W) digunakan sebagai salah satu komponen untuk menentukan jarak terhadap *output layer* (Azmi, 2014).

SOM menggunakan basis *winner takes all*, dimana hanya *neuron* pemenang yang akan diperbarui bobotnya. Pada SOM, suatu lapisan yang berisi *neuron-neuron* akan menyusun dirinya sendiri berdasarkan *input* nilai tertentu dalam suatu cluster. *Cluster* yang memiliki jarak paling dekat (nilai keluaran terkecil) akan terpilih menjadi pemenang. *Neuron* yang menjadi pemenang akan memperbaiki nilai bobotnya beserta *neruon-neuron* tetangganya.

BAB III

DESAIN DAN IMPLEMENTASI

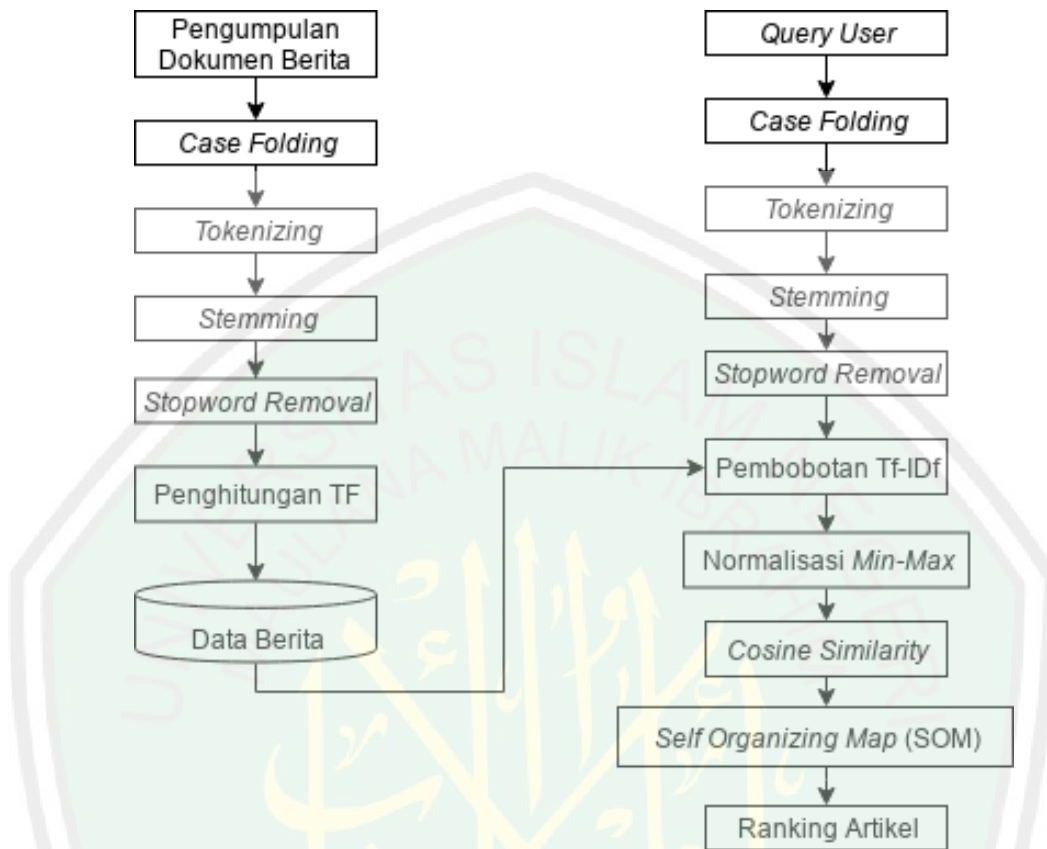
Pada bab ini akan dijelaskan analisa dan perancangan *information retrieval system* berita Jawa Timur menggunakan metode *Self Organizing Map* (SOM).

3.1. Desain Sistem

Desain sistem pada penelitian ini dapat diuraikan sebagai berikut:

- 1) Pengumpulan dokumen berita yang merupakan proses mengambil data berita seputar Jawa Timur dari situs berita online ternama dan terpercaya di Indonesia. Proses pengumpulan data dilakukan dengan meng-crawl konten yang diperlukan saja dari laman tersebut seperti waktu posting, tanggal posting berita dan isi dari berita, kemudian data tersebut disimpan dalam *database*.
- 2) Selanjutnya aplikasi akan menyimpan dokumen berita hasil *crawling* kedalam *database* dan melakukan penyaringan terhadap dokumen, jika data tersebut belum ada dalam *database* maka akan dilakukan penyimpanan dokumen kedalam *database*.
- 3) Pada tahap *preprocessing* terdapat proses merubah huruf ke non-kapital semua (*case folding*), pemenggalan suku kata (*tokenizing*), mengubah suku kata ke bentuk kata dasar (*stemming*), menghilangkan kata yang ada dalam *stoplist* (*stopword removal*).

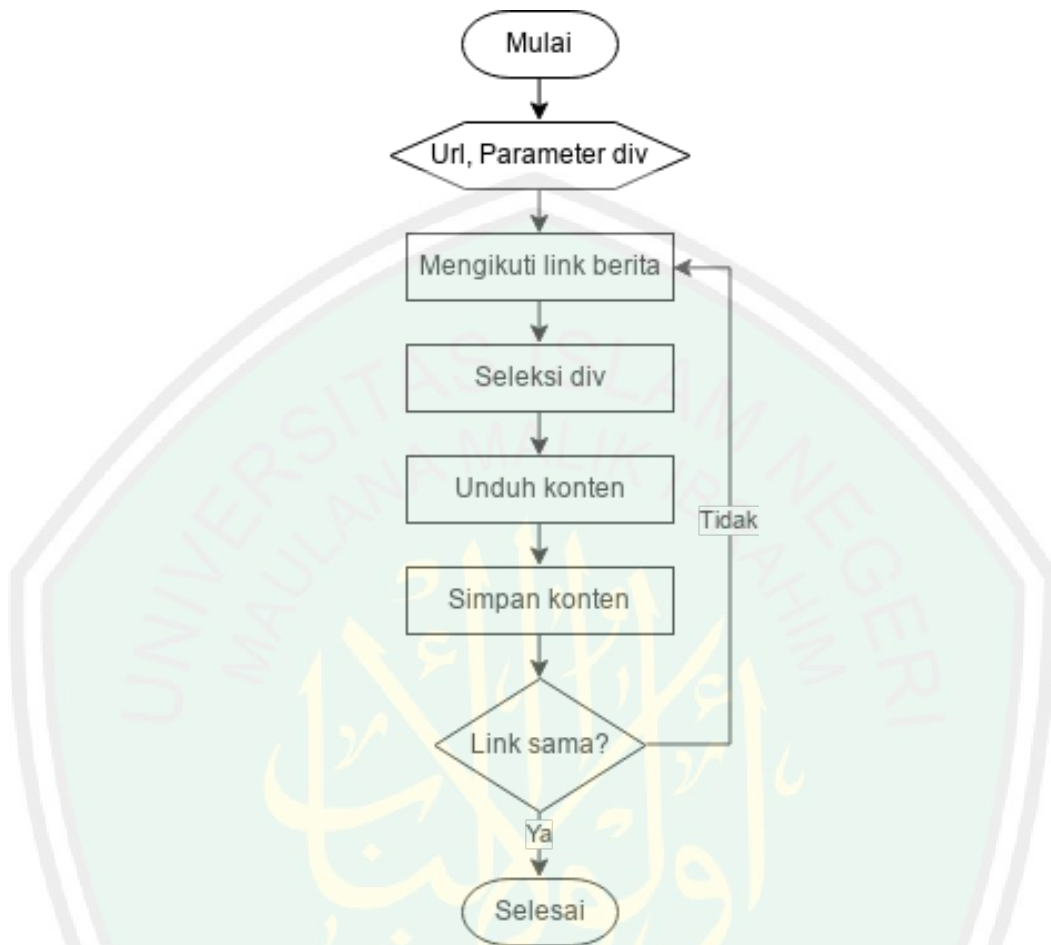
- 4) Tahap selanjutnya adalah pembobotan untuk mendapatkan nilai dari kata (*term*) yang telah diekstrak. Metode pembobotan yang digunakan adalah pembobotan *tf-idf*.
- 5) Setelah proses pembobotan maka dilakukan proses normalisasi dengan menggunakan metode Min-Max untuk meningkatkan akurasi dan efisiensi algoritma data mining yang melibatkan pengukuran jarak.
- 6) Hasil pembobotan yang sudah dinormalisasi disimpan kedalam sebuah variabel lalu dilakukan penghitungan nilai *cosine similitiy* untuk menghitung kemiripan antara *query* dan dokumen.
- 7) Tahap selanjutnya adalah *clustering* dokumen menggunakan metode *Self Organizing Map* (SOM). Untuk jelasnya bisa dilihat pada Gambar 3.1. Desain Proses berikut.



Gambar 3.1. Desain Proses

3.1.1 Pengumpulan Data

Pengumpulan data dilakukan dengan cara *crawling* data berita pada situs berita *online* terpercaya dan ternama di Indonesia. *Crawling* dilakukan dengan mengikuti link pada list berita area Jawa Timur. Proses *crawling* akan mengambil informasi yang diperlukan saja, kemudian hasil *crawl* akan disimpan kedalam *database*. Untuk penjelasan lebih lanjut dapat dilihat pada Gambar 3.2. Flowchart Pengumpulan Data.



Gambar 3.2. Flowchart Pengumpulan Data

Preprocessing merupakan tahapan awal dalam mengolah data setelah kolektif dokumen, yang nantinya dapat dilakukan pembobotan yang kemudian dilakukan proses *clustering* menggunakan metode *Self Organizing Map* (SOM).

Somantri (2017) menjelaskan bahwa dalam *preprocessing* terdapat beberapa proses yang diantaranya adalah *tokenizing* (memisahkan kata dari kalimat menjadi suku kata), *transform cases* (merubah huruf ke bentuk huruf non-kapital), *filter tokens* (pengambilan kata-kata penting), *filter stopword* (proses menghilangkan kata-kata yang sering muncul namun tidak memiliki

pengaruh apapun dalam ekstraksi klasifikasi teks) dan stemming (mengubah kata kebentuk kata dasar).

3.1.2 Case Folding

Case folding merupakan proses mengubah semua text pada dokumen dari kapital menjadi non-kapital atau sebaliknya. Dalam penelitian ini, *case folding* dilakukan untuk mengubah semua text dalam dokumen menjadi non-kapital. Lebih jelasnya dapat dilihat pada Gambar 3.3. Flowchart Case Folding.



Gambar 3.3. Flowchart Case Folding

3.1.3 Tokenizing

Tokenizing merupakan proses pemisahan atau penguraian kalimat menjadi *term* atau suku kata dan dalam proses ini juga dilakukan pembuangan tanda baca. Lebih jelasnya bisa dilihat pada Gambar 3.4. Flowchart Tokenizing.



Gambar 3.4. Flowchart Tokenizing

3.1.4 Stemming

Stemming atau juga bisa disebut dengan *lemmatization* merupakan proses mengubah kata berimbuhan menjadi kata dasar. Dalam proses ini akan menghilangkan kata-kata berulang dan kata-kata yang berimbuhan seperti imbuhan me-, di- dan -an sehingga suku kata hasil *tokenizing* akan berubah menjadi kata dasar, contohnya:

- membetulkan → betul
- berpegangan → pegang

Persoalan yang dihadapi pada proses *stemming* Bahasa Indonesia diantaranya yaitu:

- Imbuhan pada Bahasa Indonesia cukup kompleks, terdiri dari:
 - Prefiks, imbuhan di depan kata: **ber-tiga**
 - Suffiks, imbuhan di akhir kata: makan-**an**
 - Konfiks, imbuhan di depan dan di akhir kata: **per-ubah-an**
 - Infiks, imbuhan di tengah kata: k-**em**-ilau
 - Imbuhan dari bahasa asing: final-**isasi**, sosial-**isasi**
 - Aturan perubahan prefiks, seperti (me-) menjadi (meng-, mem-, men-, meny-)
- *Word-Sense Ambiguity* (Ambiguitas Rasa Kata), yaitu satu kata dapat memiliki dua makna (seperti misalnya homonim), dan berasal dari kata dasar yang berbeda. Contohnya:
 - Berikan → Ber-ikan
 - Berikan → Beri-kan
- *Overstemming*

Kata berikan berdasarkan aturan pemenggalan, dapat dipenggal menjadi Ber-i-kan. Menjadi kata dasar i. Untuk mencegah *overstemming*, algoritma membutuhkan daftar kata dasar. Jika kata yang dipenggal ada di kata dasar, hentikan proses pemenggalan.

- *Understemming*

Mengecek → menjadi meng-ecek, seharusnya menge-cek. Hal ini dapat disebabkan karena pada kamus kata dasar, ecek juga merupakan kata dasar.

- Ketergantungan terhadap kamus / daftar kata dasar

Untuk mencegah *overstemming*, algoritma menjadi tergantung pada kata dasar. Adanya kekurangan atau kelebihan pada kata dasar dapat menyebabkan *overstemming* atau *understemming*.

- Pengguna Bahasa Indonesia tidak konsisten dalam menentukan *stem* (secara manual)

Kita, manusia, juga kadang berbeda pendapat dalam menentukan *stem* sebuah kata. Contohnya, apakah "adalah" merupakan kata berimbuhan dari kata dasar "ada"? Apakah "bagian" adalah kata berimbuhan dari kata dasar "bagi"?

- Kata bentuk jamak = buku-buku (kata dasar nya buku)

- Kata serapan dari bahasa asing = mengakomodir → meng-akomodir

- Kesalahan penulisan = penambahanan, harusnya penambahan, sehingga tidak dapat di-*stem*

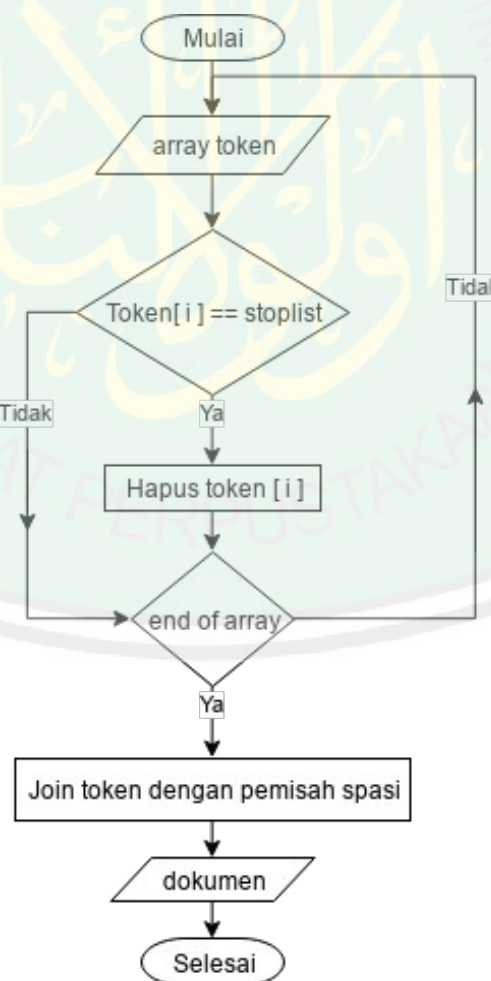
- Akronim = pemilu distem menjadi pe-milu

- *Proper Noun* (Nama Benda), misal nama orang, nama kota: Abdullah di-*stem* menjadi abdul, seharusnya tidak di-*stem*

Pada penelitian ini penulis menggunakan *Stemming* Bahasa Indonesia yaitu Sastrawi *stemmer* yang menerapkan algoritma Nazief dan Adriani yang ditingkatkan oleh algoritma *Modified ECS*.

3.1.5 Stopword Removal

Stopword removal merupakan proses menghilangkan kata-kata yang tidak penting yang terdapat di *stoplist*. Hal ini dilakukan untuk mengambil data text yang dianggap penting saja dan untuk mengoptimalkan proses pengolahan text selanjutnya. Lebih jelasnya bisa dilihat pada.



Gambar 3.5. Flowchart Stopword Removal

3.1.6 Pembobotan Kata atau *Term*

Pembobotan kata atau *term* yang digunakan dalam penelitian ini adalah metode *tf-idf*. Pada tahap pembobotan, sistem akan mempresentasikan dokumen berita sebagai vektor dengan nilai bobot masing-masing *term* dan setiap kalimat dari kumpulan data teks dianggap sebagai dokumen dalam perhitungan.

Pembobotan *tf-idf* dilakukan dengan 7 tahap sebagai berikut:

1. *Document Indexing*: merupakan proses untuk menentukan *term* indeks (*t*) mana yang akan digunakan sebagai representasi dari dokumen. Dalam penelitian ini setiap kata yang tersisa hasil *preprocessing* akan digunakan sebagai *term* indeks.
2. *Term Weighting*: merupakan proses untuk men-generate sebuah nilai pada setiap term dengan cara menghitung frekuensi kemunculan *term* dengan cara menghitung frekuensi kemunculan *term* dalam dokumen (*d*).
3. *Log-Frequency Weighting*: merupakan proses untuk menghitung nilai bobot hasil dari nilai kemunculan *term weighting*, dimana digunakan rumus yang dapat dilihat dalam persamaan berikut:

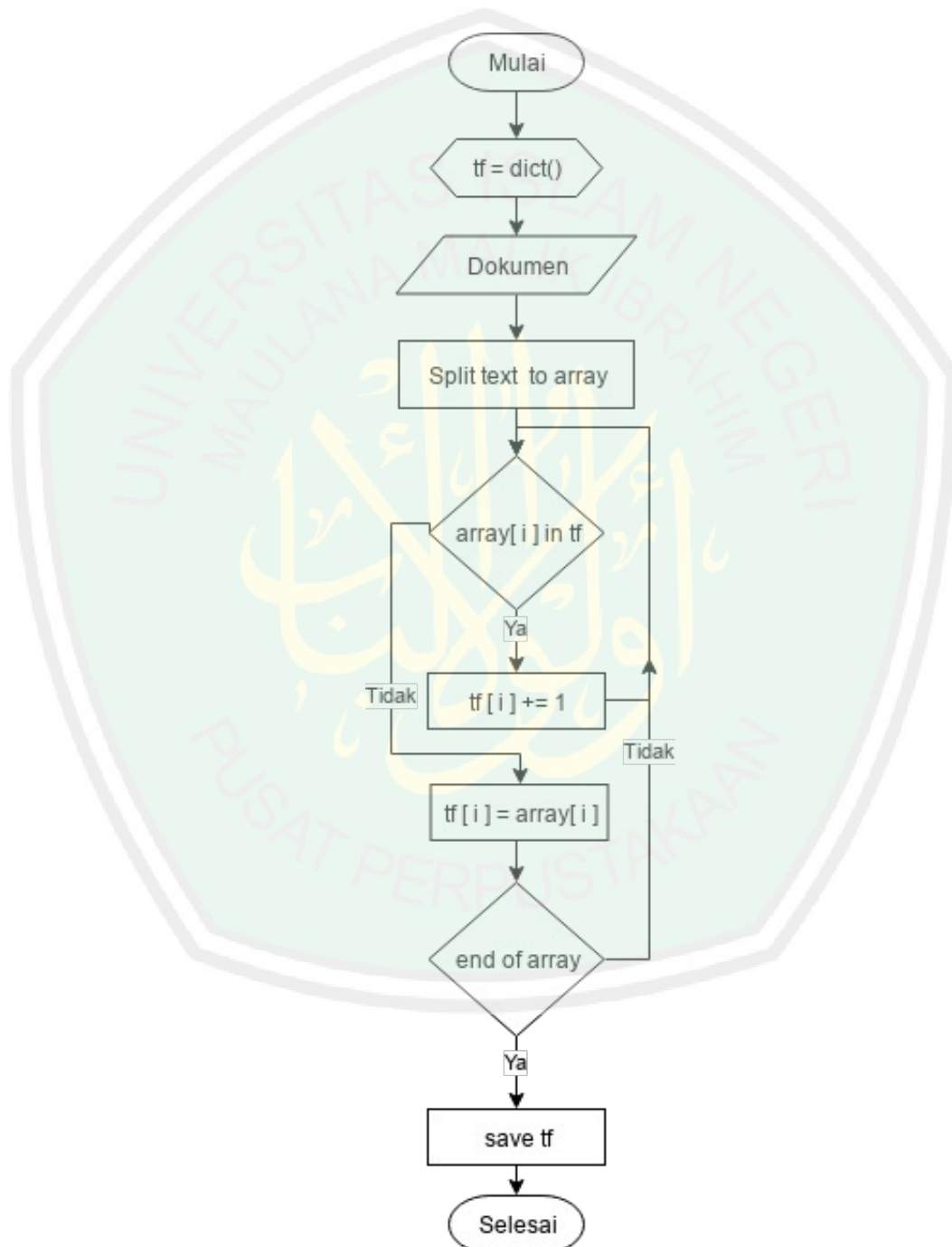
$$W_{tf_{t,d}} = \begin{cases} 1 + \log_{10} tf_{t,d} & \text{jika } tf_{t,d} > 0 \\ 0, & \text{lainnya} \end{cases}$$

dengan keterangan:

- $W_{tf_{t,d}}$ = *log-frequency weighting* pada *term* ke *t*, dokumen ke *d*.

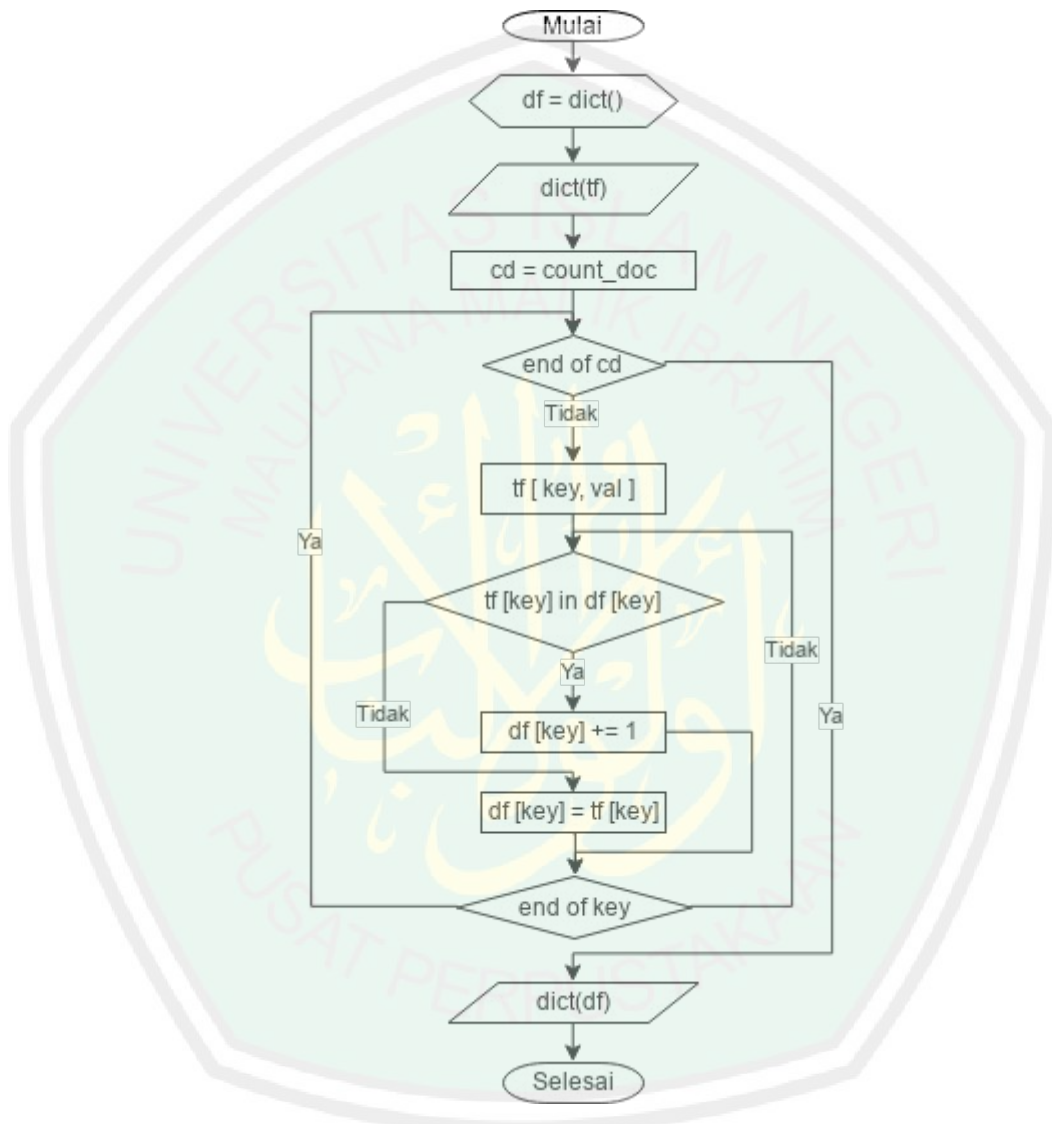
- $tf_{t,d}$ = nilai dari *term weighting* pada *term* ke t , dokumen ke d .

Berikut merupakan Gambar 3.6. Flowchart TF:



Gambar 3.6. Flowchart TF

4. *Document Frequency*: merupakan proses untuk menghitung banyaknya dokumen yang mengandung *term* ke *t*.



Gambar 3.7. Flowchart DF

5. *Inverse Document Frequency*: merupakan proses untuk menghitung nilai *inverse* dari *document frequency*, dimana digunakan rumus sebagai berikut:

$$idf_t = \log_{10} \frac{N}{df_t} \text{ atau } \log_{10} \left(1 + \frac{N}{df_t} \right)$$

dengan keterangan:

- idf_t = *inverse document frequency* pada *term* ke t .
- N = jumlah keseluruhan dokumen yang ada.
- df_t = nilai dari *document frequency* pada *term* ke t .

6. TF-IDF: merupakan proses untuk mendapatkan nilai skor setiap terhadap dokumen, dimana digunakan rumus sebagai berikut:

$$W_{t,d} = Wtf_{t,d} \times idf_t$$

dengan keterangan:

- $W_{t,d}$ = TF-IDF pada *term* ke t , dokumen ke d .
- $Wtf_{t,d}$ = *log-frequency weighting* pada *term* ke t , dokumen ke d .
- idf_t = *inverse document frequency* pada *term* ke t .

7. Menghitung nilai skor akhir setiap dokumen, dimana digunakan rumus sebagai berikut:

$$Ws_j = \sum_{i=1}^{Nterm} Wtd_{i,j}$$

dengan keterangan:

- Ws_j = skor dari dokumen ke j .
- $Nterm$ = jumlah banyaknya *term*.
- $Wtd_{i,j}$ = nilai dari TF-IDF pada *term* ke i , dokumen ke j .

3.1.7 Normalisasi

Pada tahap normalisasi data akan menggunakan metode *min-max normaliation*. *Min-max normalization* melakukan transformasi linear pada

data asli. Misalkan \min_A dan \max_A adalah nilai minimum dan maksimum dari sebuah atribut. *Min-max normalization* memetakan sebuah nilai v dari A menjadi v' dalam range nilai minimal dan maksimal yang baru, new_min_A dan new_max_A . Pada penelitian ini normalisasi dilakukan dengan *scale* data antara 1 dan 9.

Normalisasi *min-max* dirumuskan sebagai berikut:

$$v' = \frac{v - \min_A}{\max_A - \min_A} ((\text{new_max}_A - \text{new_min}_A) + \text{new_min}_A)$$

v adalah representasi *value* dari sebuah atribut, \min_A adalah nilai minimum dalam data set, \max_A adalah nilai maksimum dalam data set, new_max_A adalah nilai maksimum yang baru, new_min_A adalah nilai minimum yang baru.

3.1.8 Cosine Similarity

Sukma, Zaman dan Purwanti (2015) untuk mengetahui tingkat similaritas antar dokumen dapat menggunakan metode *cosine similarity*. Dalam penelitian ini metode *cosinus similarity* digunakan untuk menghitung tingkat similaritas *query user* dengan dokumen.

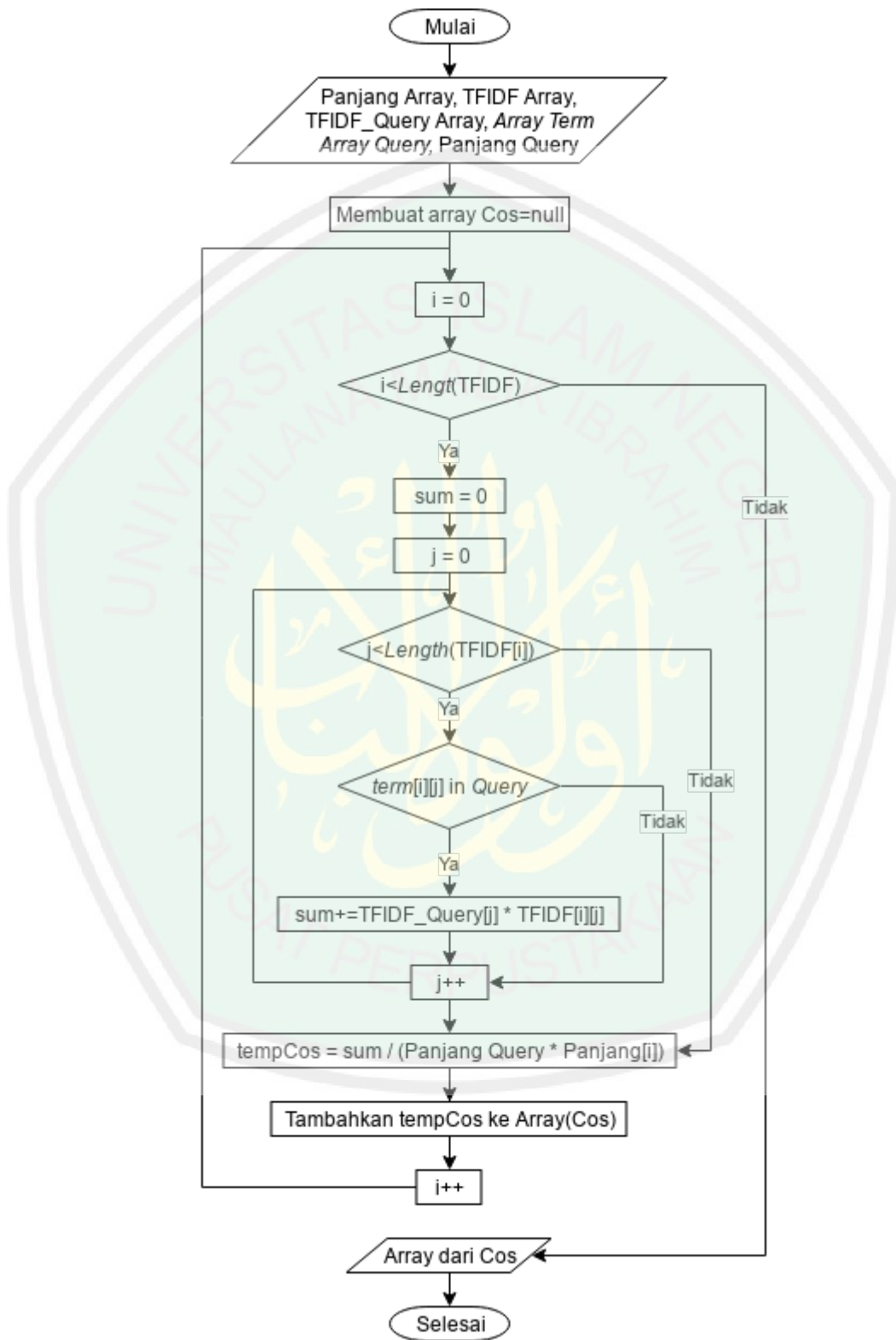
Untuk persamaan metode *cosine similarity* adalah sebagai berikut:

$$\cos(q, d_j) = \frac{\sum_i [w(t_i, q)] * [w(t_i, d_j)]}{\sqrt{\sum |w(q)|^2 * \sum |w(d_j)|^2}}$$

Dari persamaan tersebut akan menghasilkan nilai similaritas antara 0 sampai 1. Nilai yang paling mendekati 1 menunjukkan tingkat similaritas yang

tinggi. Dalam persamaan $\cos(q, d_j)$ adalah cosinus antara *query* dan dokumen j , $w(t_i, q)$ adalah berat dari *term* t_i TF.IDF pada *query* dan $w(t_i, d)$ adalah berat untuk setiap *term* TF.IDF t_i berdasarkan dokumen pada distribusi *term* dalam ruang kelas. Sementara $|w(q)|$ dan $|w(d_j)|$ masing-masing adalah panjang dari vektor q dan panjang dari vektor dokumen j . Berikut ini adalah Gambar 3.8. Flowchart Cosine Similarity:



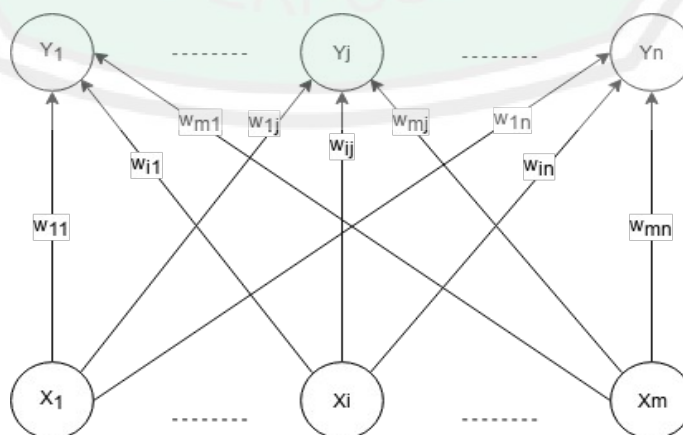


Gambar 3.8. Flowchart Cosine Similarity

3.1.9 Self-Organizing Map (SOM)

Metode *Self Organizing-Map* (SOM) pada penelitian ini akan digunakan untuk mengelompokkan dokumen berita. SOM merupakan algoritma jaringan dengan kategori pembelajaran secara kompetitif dan bersifat unsupervised. Sistem secara otomatis dapat melakukan pengelompokan tanpa menggunakan pembelajaran dengan pasangan data terlebih dahulu.

SOM merupakan jaringan yang terdiri dari dua lapisan (layer), yaitu lapisan *input* dan lapisan *output*. Setiap *neuron* dalam lapisan *output* merepresentasikan kelas (*cluster*) dari *input* yang diberikan. *Input* layer (X) akan berbentuk *linear* dan demikian pula *output*-nya (Y). Karena SOM berbasis *winner takes all*, maka *neuron* terdekat akan di anggap sebagai pemenang. *Neuron* yang menjadi pemenang akan memperbaiki nilai bobotnya (w) beserta *neuron-neuron* tetangganya. Lebih jelasnya dapat dilihat pada Gambar 3.9. Arsitektur SOM:



Gambar 3.9. Arsitektur SOM

Untuk algoritma *Self Organizing Map* (SOM) adalah sebagai berikut:

1. Menentukan jumlah *cluster* yang dalam penelitian ini sebanyak 3 cluster.
2. Menentukan laju pembelajaran awal (α) dengan $\alpha = 0,5$.
3. Menentukan bobot (w) awal secara acak antara 0 sampai dengan 1.
4. Mencari jarak terdekat menggunakan rumus *euclidian distance* dari masing-masing sebagai *neuron output* ke data *input* (x_i) yang merupakan hasil dari normalisasi *min-max* dengan nilai *range* 0 sampai dengan 9.

Berikut merupakan persamaan *euclidian distance*:

$$D_i = \sum_{i=1}^n (w_{ij} - x_i)^2$$

Dari seluruh bobot (D_i) dicari yang paling kecil jaraknya, indeks dari bobot (D_i) ini disebut *winning neuron*.

5. Untuk setiap bobot w_{ij} diperbarui bobot tetangga menggunakan rumus sebagai berikut:

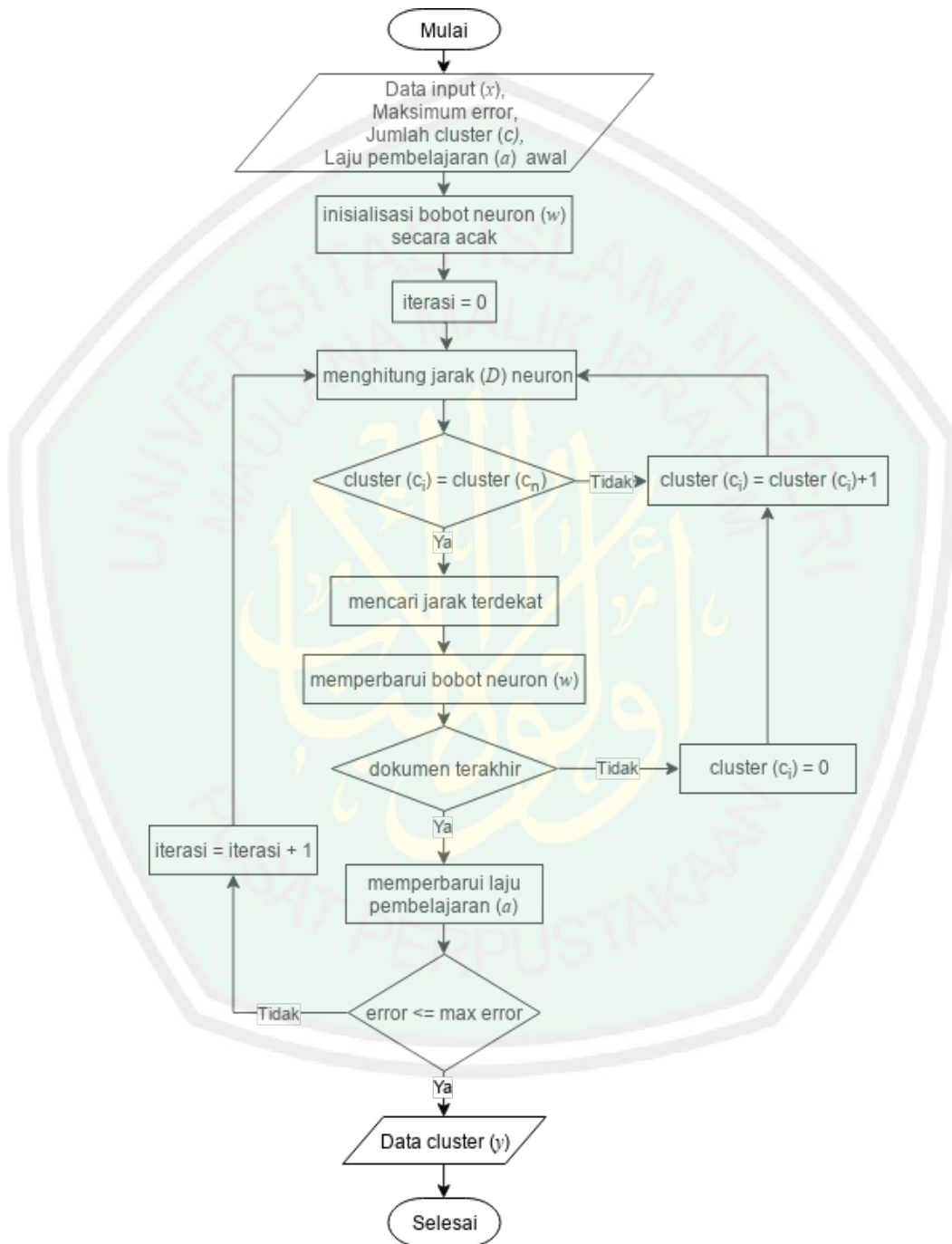
$$w_{ij}(t+1) = w_{ij}(t) + \alpha(t)[x_i - w_{ij}(t)]$$

6. Memperbarui laju pembelajaran (α) dengan 0,6 dengan persamaan berikut:

$$\alpha(\text{baru}) = 0.6 \times \alpha(\text{lama})$$

7. Iterasi dihentikan ketika sudah mencapai kondisi *konvergen*. Jika belum mencapai kondisi *konvergen* maka ulangi langkah 3 sampai 5 hingga tidak ada perubahan *weight* (bobot) atau telah mencapai kondisi *konvergen* atau *stop* (*error* terkecil).

Lebih jelasnya dapat dilihat pada Gambar 3.10. Flowchart SOM berikut:



Gambar 3.10. Flowchart SOM

Setelah proses *clustering*, tahap selanjutnya adalah mengurutkan dokumen sesuai *cluster* dan nilai *cosine similarity*.

3.1.10 Desain *Interface*

Desain *interface* dari program yang berbasis web pada penelitian ini adalah sebagai berikut:

a. Tampilan Hasil Pencarian dan Menu Reset

Pada halaman hasil pencarian akan menampilkan *form* input pertanyaan dan menampilkan jawaban dari pertanyaan yang di-inputkan oleh *user*. Jawaban yang ditampilkan berupa judul dokumen yang terdapat link menuju sumber dokumen tersebut. Di bawah judul terdapat *headline* atau ringkasan singkat dari isi dokumen. Di bawah *headline* terdapat label keterangan waktu saat dokumen di-*publish*. Sedang menu *reset* digunakan untuk me-*reset* hasil pencarian dan menuju ke halaman awal. Halaman pada menu ini akan terlihat seperti Gambar 3.11. Interface Hasil Pencarian berikut:

The screenshot shows a web interface for search results. At the top, there is a search bar with the placeholder text 'Ketikkan pertanyaan seputar berita jawa timur' and a blue 'Jawab' button. Below the search bar, there are navigation links: 'Reset || Get Data || Pre-processing || Count Term'. The main content area displays two identical result cards. Each card has a blue header 'Judul berita', followed by 'Headline dari berita', and 'Waktu publish berita'.

Gambar 3.11. Interface Hasil Pencarian

b. Menu *Get Data*

Ketika menu *Get Data* diklik maka program akan melakukan proses penyimpanan dokumen hasil *crawling* ke dalam *database* MySQL. Dalam proses penyimpanan ke *database* MySQL, program juga akan melakukan seleksi dokumen yang apabila dokumen sudah terdapat dalam *database* maka dokumen tersebut tidak akan disimpan kedua kalinya, hal ini dilakukan untuk mengoptimalkan penyimpanan dan pemrosesan selanjutnya. Halaman pada menu ini akan terlihat seperti Gambar 3.12. Interface Menu Get Data berikut:

The screenshot shows a web application interface for the 'Get Data' menu. At the top, there is a header 'Nama Aplikasi'. Below it is a search bar with the placeholder text 'Ketikkan pertanyaan seputar berita jawa timur' and a blue 'Jawab' button. Under the search bar, there are two buttons: 'Reset' and 'Get Data'. Below these buttons, there are two sections, each titled 'Nomor urut'. Each section contains a list of fields: 'Headline :', 'Date :', 'Main Headline :', 'Url :', and 'Content :'.

Gambar 3.12. Interface Menu Get Data

c. Menu *Pre-processing*

Ketika menu *Pre-processing* diklik maka program akan melakukan proses *case folding*, *tokenizing*, *stopword removal* hingga proses *stemming*. Halaman pada menu ini akan terlihat seperti Gambar 3.13. Interface Menu *Pre-processing* berikut:

Gambar 3.13. Interface Menu Pre-processing

d. Menu *Count Term*

Ketika menu ini diklik, program akan melakukan proses penghitungan kata atau *term* pada setiap dokumen, yang nantinya akan digunakan untuk pembobotan pada metode TF-IDF. Halaman pada menu ini akan terlihat seperti Gambar 3.14. Interface Menu Count Term berikut:

Gambar 3.14. Interface Menu Count Term

3.2. Implementasi

Pengimplementasian dilakukan menggunakan *web framework* Django versi 2.0.3 yang menggunakan bahasa pemrograman Python 3. Sistem operasi yang digunakan adalah Debian GNU/Linux 9 (*stretch*) x86-64. Perangkat lunak lainnya yang digunakan adalah XAMPP 7.1.21-0 dengan database server MySQL dan browser Google Chrome Version 70.0.3538.67. Perangkat keras menggunakan

laptop Asus X452CP dengan prosesor Intel® Core™ i3 3217U, RAM 6GB, 500GB HDD, AMD Radeon® HD 8530M dengan 1GB DDR3 VRAM, display 14.0" 16:9 HD/GL/LED (1366x768). Implementasi dilakukan dengan menerapkan tahapan-tahapan dari desain sistem yang diajukan.

3.2.1 Pengumpulan Data

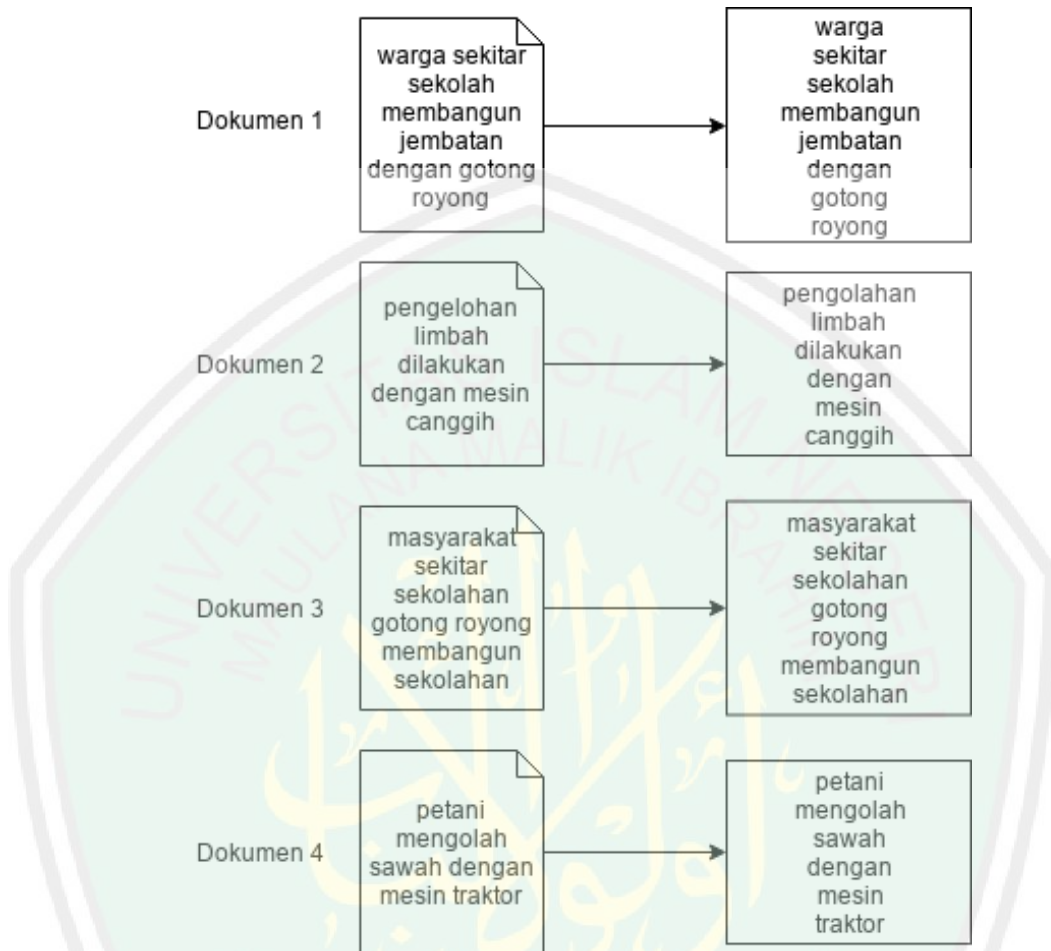
Pengumpulan data dilakukan dengan meng-*crawl* dari situs media berita online. *Library* python yang digunakan untuk melakukan *crawl* adalah Beautiful Soup 4 untuk mengambil data dari HTML. Langkah awal pengambilan data adalah dengan menginisialisasi url situs media berita *online* Jawa Timur. Langkah selanjutnya yaitu mengambil data berita yang diperlukan seperti judul, url, tanggal, dan konten dengan mengabaikan bagian-bagian laman berita *online* yang tidak diperlukan.

3.2.2 Case Folding

Pada proses ini semua huruf akan dirubah menjadi non-kapital. Misal ada kata “Baik” dengan awalan huruf kapital, maka huruf awal pada kata tersebut akan diubah menjadi non-kapital menjadi “baik” sehingga semua huruf pada dokumen menjadi non-kapital.

3.2.3 Tokenizing

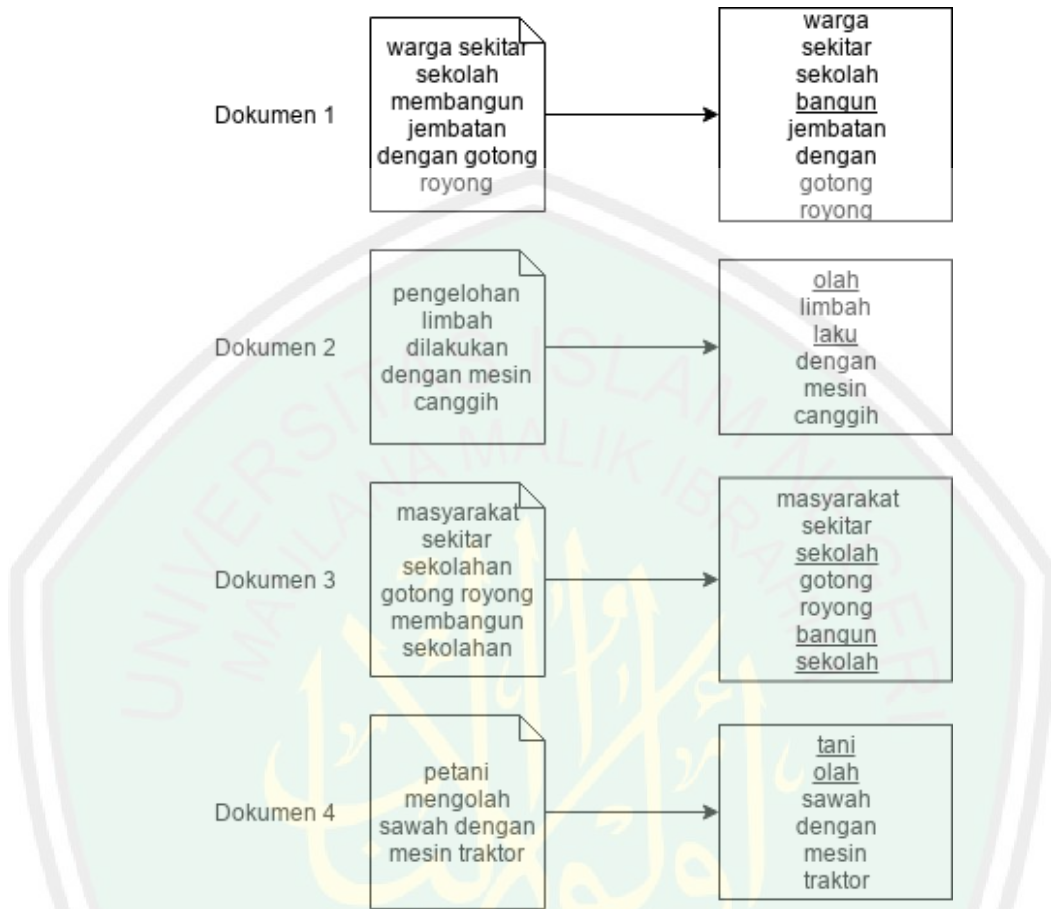
Tahap *tokenizing* akan merubah paragraf atau kalimat kedalam bentuk suku kata. Lebih jelasnya dapat dilihat contoh penerapan tokenizing pada 4 dokumen berikut Gambar 3.15. Tokenizing:



Gambar 3.15. Tokenizing

3.3.4 Stemming

Proses stemming akan merubah kata-kata yang berimbuhan menjadi kata dasar. Misal terdapat kata “berjalan” maka dalam proses stemming akan dirubah menjadi kata “jalan”. Lebih jelasnya dapat dilihat pada contoh berikut Gambar 3.16. Stemming:



Gambar 3.16. Stemming

3.3.5 Stopword Removal

Langkah selanjutnya adalah *stopword* removal yang menghilangkan kata-kata dari dokumen yang tidak penting sesuai yang terdapat pada *stoplist*. Misal dalam dokumen terdapat kata “dengan” dan di *stoplist* juga terdapat kata “dengan”, maka kata tersebut akan dihapus dari dokumen. Contoh penerapannya adalah sebagai berikut Gambar 3.17. Stopword Removal:



Gambar 3.17. Stopword Removal

3.3.6 Pembobotan Kata atau *Term*

Langkah awal dalam pembobotan yaitu menghitung frekuensi kata (*term frequency*) pada suatu dokumen dan hasilnya disimpan ke dalam *database*. Untuk lebih jelasnya dapat dilihat Gambar 3.18. Pembobotan TF:



Gambar 3.18. Pembobotan TF

Setelah menghitung *term frequency*, maka langkah selanjutnya adalah menghitung DF (*Document Frequency*) yang diikuti dengan menghitung nilai IDF (*Inverse Document Frequency*). Contoh perhitungan IDF pada *term* “warga” pada dokumen berita yang pertama dapat dilihat sebagai berikut:

$$IDF_1 = \log_{10} \left(1 + \left(\frac{N}{df_1} \right) \right)$$

$$IDF_1 = \log_{10} \left(1 + \left(\frac{4}{1} \right) \right) = 0.698970004336019$$

Berikut Tabel 3.1. Hasil DF dan IDF:

Tabel 3.1. Hasil DF dan IDF

Kata	D1	D2	D3	D4	DF	IDF
	TF	TF	TF	TF		
warga	1	0	0	0	1	0.69897
traktor	0	0	0	1	1	0.69897
bangun	1	0	1	0	2	0.47712
jembatan	1	0	0	0	1	0.69897
gotong	1	0	1	0	2	0.47712
royong	1	0	1	0	2	0.47712
olah	0	1	0	1	2	0.47712
limbah	0	1	0	0	1	0.69897
laku	0	1	0	0	1	0.69897
mesin	0	1	0	1	2	0.47712
canggih	0	1	0	0	1	0.69897
masyarakat	0	0	1	0	1	0.69897
sekolah	1	0	2	0	3	0.36798
tani	0	0	0	1	1	0.69897
sawah	0	0	0	1	1	0.69897

Langkah selanjutnya adalah menghitung bobot dokumen dengan menghitung nilai TF-IDF dari dokumen (wd). Lalu selanjutnya mencari nilai minimal dan maksimal dari bobot TF-IDF pada tiap-tiap *term* yang akan digunakan nantinya pada proses berikutnya. Contoh perhitungan bobot (wd) pada *term* “warga” pada dokumen pertama dapat dilihat sebagai berikut:

$$wd_{1,1} = tf_{1,1} \times idf_1$$

$$wd_{1,1} = 1 \times 0.698970004336019 = 0.69897$$

Berikut Tabel 3.2. Hasil TF-IDF dan Pencarian Nilai Minimal Maksimal :

Tabel 3.2. Hasil TF-IDF dan Pencarian Nilai Minimal Maksimal

Kata	D1	D2	D3	D4	Max	Min
	wd1	wd2	wd3	wd4		
warga	0.70	0.00	0.00	0.00	0.70	0.00
traktor	0.00	0.00	0.00	0.70	0.70	0.00
bangun	0.48	0.00	0.48	0.00	0.48	0.00
jembatan	0.70	0.00	0.00	0.00	0.70	0.00
gotong	0.48	0.00	0.48	0.00	0.48	0.00
royong	0.48	0.00	0.48	0.00	0.48	0.00
olah	0.00	0.48	0.00	0.48	0.48	0.00
limbah	0.00	0.70	0.00	0.00	0.70	0.00
laku	0.00	0.70	0.00	0.00	0.70	0.00
mesin	0.00	0.48	0.00	0.48	0.48	0.00
canggih	0.00	0.70	0.00	0.00	0.70	0.00
masyarakat	0.00	0.00	0.70	0.00	0.70	0.00
sekolah	0.37	0.00	0.74	0.00	0.74	0.00
tani	0.00	0.00	0.00	0.70	0.70	0.00
sawah	0.00	0.00	0.00	0.70	0.70	0.00

3.3.7 Normalisasi

Setelah mencari nilai minimal dan maksimal barulah dilakukan proses normalisasi data menggunakan metode *Min-Max Normalization* yang akan menghasilkan nilai baru dengan *range* antara 1 sampai dengan 9. Berikut contoh normalisasi dokument 1 (wd1 pada kata “warga”):

$$v' = \frac{v - \min_A}{\max_A - \min_A} ((\text{new_max}_A - \text{new_min}_A) + \text{new_min}_A)$$

$$v' = \frac{0.698970004336019 - 0}{0.698970004336019 - 0} ((9 - 1) + 1) = 9$$

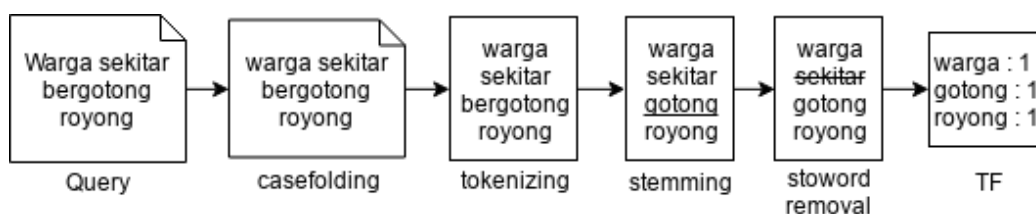
v' adalah hasil dari normalisasi, new_max adalah nilai maksimal yang baru dan new_min adalah nilai minimal yang baru. Hasil normalisasi dapat dilihat pada Tabel 3.3. Hasil Normalisasi Min-Max berikut:

Tabel 3.3. Hasil Normalisasi Min-Max

Kata	D1	D2	D3	D4
	wd1	wd2	wd3	wd4
warga	9	0	0	0
traktor	0	0	0	9
bangun	9	0	9	0
jembatan	9	0	0	0
gotong	9	0	9	0
royong	9	0	9	0
olah	0	9	0	9
limbah	0	9	0	0
laku	0	9	0	0
mesin	0	9	0	9
canggih	0	9	0	0
masyarakat	0	0	9	0
sekolah	4.5	0	9	0
tani	0	0	0	9
sawah	0	0	0	9

3.3.8 Cosine Similarity

Proses selanjutnya adalah menghitung tingkat kemiripan antara *query* atau inputan dari *user* dengan dokumen berita. Setiap vektor dokumen berita akan dihitung nilai similaritas dengan *query*. Perhitungan ini juga digunakan untuk pemberian ranking pada dokumen berita terhadap *query*. Semakin tinggi nilai similaritas, maka semakin mirip dokumen tersebut dengan *query*. Misal *query* yang dimasukkan oleh *user* adalah “Warga sekitar gotong royong”, maka ilustrasi proses dimulai dari *casefolding* hingga *term frequency* pada *query* adalah sebagai berikut:



Gambar 3.19. Proses Query Sampai Pembobotan TF

Kemudian dilakukan penghitungan nilai TF, DF dan IDF seperti pada Tabel 3.4. Hasil TF, DF dan IDF berikut:

Tabel 3.4. Hasil TF, DF dan IDF

Kata	Q	D1	D2	D3	D4	DF	IDF
	TF	TF	TF	TF	TF		
warga	1	1	0	0	0	2	0.47712
traktor	0	0	0	0	1	1	0.69897
bangun	0	1	0	1	0	2	0.47712
jembatan	0	1	0	0	0	1	0.69897
gotong	1	1	0	1	0	3	0.36798
royong	1	1	0	1	0	3	0.36798
olah	0	0	1	0	1	2	0.47712
limbah	0	0	1	0	0	1	0.69897
laku	0	0	1	0	0	1	0.69897
mesin	0	0	1	0	1	2	0.47712
canggih	0	0	1	0	0	1	0.69897
masyarakat	0	0	0	1	0	1	0.69897
sekolah	0	1	0	2	0	3	0.36798
tani	0	0	0	0	1	1	0.69897
sawah	0	0	0	0	1	1	0.69897

Langkah selanjutnya yaitu menghitung bobot dokumen (w) dengan metode TF-IDF yang dilanjutkan dengan pencarian nilai maksimal dan minimal seperti pada Tabel 3.5. Hasil TF-IDF dan Pencarian Maksimal Minimal berikut:

Tabel 3.5. Hasil TF-IDF dan Pencarian Maksimal Minimal

Kata	Q	D1	D2	D3	D4	Max	Min
	wq	wd1	wd2	wd3	wd4		
warga	0.48	0.48	0.00	0.00	0.00	0.48	0.00
traktor	0.00	0.00	0.00	0.00	0.70	0.70	0.00
bangun	0.00	0.48	0.00	0.48	0.00	0.48	0.00
jembatan	0.00	0.70	0.00	0.00	0.00	0.70	0.00
gotong	0.37	0.37	0.00	0.37	0.00	0.37	0.00
royong	0.37	0.37	0.00	0.37	0.00	0.37	0.00
olah	0.00	0.00	0.48	0.00	0.48	0.48	0.00
limbah	0.00	0.00	0.70	0.00	0.00	0.70	0.00
laku	0.00	0.00	0.70	0.00	0.00	0.70	0.00
mesin	0.00	0.00	0.48	0.00	0.48	0.48	0.00
canggih	0.00	0.00	0.70	0.00	0.00	0.70	0.00
masyarakat	0.00	0.00	0.00	0.70	0.00	0.70	0.00
sekolah	0.00	0.37	0.00	0.74	0.00	0.74	0.00
tani	0.00	0.00	0.00	0.00	0.70	0.70	0.00
sawah	0.00	0.00	0.00	0.00	0.70	0.70	0.00

Proses selanjutnya adalah menormalisasi data menjadi *range* antara 0 sampai dengan 9 yang hasilnya seperti Tabel 3.6. Hasil Normalisasi berikut:

Tabel 3.6. Hasil Normalisasi

Kata	Q	D1	D2	D3	D4
	wq	wd1	wd2	wd3	wd4
warga	9	9	0	0	0
traktor	0	0	0	0	9
bangun	0	9	0	9	0
jembatan	0	9	0	0	0
gotong	9	9	0	9	0
royong	9	9	0	9	0
olah	0	0	9	0	9
limbah	0	0	9	0	0
laku	0	0	9	0	0
mesin	0	0	9	0	9
canggih	0	0	9	0	0
masyarakat	0	0	0	9	0
sekolah	0	4.5	0	9	0
tani	0	0	0	0	9
sawah	0	0	0	0	9

Hasil dari normalisasi tersebut akan digunakan untuk perhitungan tingkat kemiripan teks dengan metode *cosine similarity*. Hasil dari proses perhitungan *cosine similarity* adalah seperti Tabel 3.7. Hasil Cosine Similarity berikut:

Tabel 3.7. Hasil Cosine Similarity

	Q * wdi				w ^ 2				
	wd1	wd2	wd3	wd4	wq	wd1	wd2	wd3	wd4
	81	0	0	0	81	81	0	0	0
	0	0	0	0	0	0	0	0	81
	0	0	0	0	0	81	0	81	0
	0	0	0	0	0	81	0	0	0
	81	0	81	0	81	81	0	81	0
	81	0	81	0	81	81	0	81	0
	0	0	0	0	0	0	81	0	81
	0	0	0	0	0	0	81	0	0
	0	0	0	0	0	0	81	0	81
	0	0	0	0	0	0	81	0	0
	0	0	0	0	0	0	0	81	0
	0	0	0	0	0	20.25	0	81	0
	0	0	0	0	0	0	0	0	81
	0	0	0	0	0	0	0	0	81
sum	243	0	162	0	243	425.25	405	405	405
sqrt(sum)					15.588	20.622	20.125	20.125	20.125
akar_Q * akar_d_i						321.46	313.71	313.71	313.71
cosine_similarity						0.7559	0	0.5164	0

Berdasarkan perhitungan yang dilakukan maka diketahui dokumen berita 1 dan 3 masing-masing mempunyai nilai kemiripan dengan *query* sebesar 0.7559 dan 0.5164, sedangkan dokumen berita 2 dan 4 masing-masing nilai kemiripannya adalah 0. Jadi ranking dari tiap-tiap indek dokumen berita secara berurutan adalah dokumen 1, 3, 2 dan 4.

3.3.9 Self-Organizing Map

Tahap selanjutnya adalah proses *clustering* dokumen menggunakan metode *Self-Organizing Map* (SOM) menjadi 2 *cluster*. Langkah awal untuk *clustering*

adalah dengan menginisialisasi bobot *cluster* (W) secara *random* seperti pada Tabel 3.8. Inisialisasi Bobot Cluster :

Tabel 3.8. Inisialisasi Bobot Cluster

W1	0.7	0.8	0	0.5	0.2	1	0	0.4	0.6	0.3	0.4	0.6	0	0.2	0.3
W2	0.1	1	0.1	0.4	0.6	0.2	0.7	0.4	0.4	1	0	0	0.7	0	0.7

Nilai 0.7 merupakan inisialisasi bobot (w_{11}) pertama secara *random* dari kata “warga” pada *cluster* 1. Nilai 0.8 merupakan inisialisasi bobot (w_{12}) pertama secara *random* dari kata “traktor” pada *cluster* 1. Nilai 0.1 merupakan inisialisasi bobot (w_{21}) pertama secara *random* dari kata “warga” pada *cluster* 2 dan seterusnya. Lalu untuk bobot dokumen yang disimbolkan $D(i)$ yang telah dinormalisasi adalah sebagai berikut seperti pada Tabel 3.9. Bobot Dokumen :

Tabel 3.9. Bobot Dokumen

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Q	9	0	0	0	9	9	0	0	0	0	0	0	0	0	0
D(1)	9	0	9	9	9	9	0	0	0	0	0	0	4.5	0	0
D(2)	0	0	0	0	0	0	9	9	9	9	9	0	0	0	0
D(3)	0	0	9	0	9	9	0	0	0	0	0	9	9	0	0
D(4)	0	9	0	0	0	0	9	0	0	9	0	0	0	9	9

Langkah selanjutnya adalah melakukan iterasi untuk memperbarui bobot *cluster* hingga hasil perulangan menjadi *konvergen* atau tidak berubah-ubah. Bobot akan diperbarui jika nilai jarak (D) merupakan nilai terkecil. Hasil dari iterasi dapat dilihat sebagai berikut:

Iterasi 1

Pada iterasi pertama, laju pembelajaran (α) menggunakan 0,5 yang kemudian untuk laju pembelajaran pada iterasi selanjutnya akan diperbarui dengan perhitungan nilai 0,6. Berikut merupakan penghitungan jarak yang

disimbolkan dengan huruf “ D_i ”, pada *query* (Q) yang nantinya digunakan untuk acuan memperbarui bobot *cluster* (w).

$$D_i = \sum_{i=1}^n (w_{ij} - x_i)^2$$

$$\begin{aligned} D1 &= (0.7-9)^2 + (0.8-0)^2 + (0-0)^2 + (0.5-0)^2 + (0.2-9)^2 + (1-9)^2 + (0-0)^2 + (0.4-0)^2 + \\ &\quad (0.6-0)^2 + (0.3-0)^2 + (0.4-0)^2 + (0.6-0)^2 + (0-0)^2 + (0.2-0)^2 + (0.3-0)^2 \\ &= \mathbf{14.57669} \end{aligned}$$

$$\begin{aligned} D2 &= (0.1-9)^2 + (1-0)^2 + (0.1-0)^2 + (0.4-0)^2 + (0.6-9)^2 + (0.2-9)^2 + (0.7-0)^2 + (0.4-0)^2 \\ &\quad + (0.4-0)^2 + (1-0)^2 + (0-0)^2 + (0-0)^2 + (0.7-0)^2 + (0-0)^2 + (0.7-0)^2 \\ &= \mathbf{14.57669} \end{aligned}$$

Pada perhitungan jarak *query* diatas $D1$ memiliki jarak yang lebih kecil, maka bobot *cluster* ke-1 (w_1) yang akan diperbarui bobotnya. Lakukan perhitungan dibawah hingga pada $w_{1,15}$ untuk memperbarui bobot *cluster*.

$$w_{1,1} = 0.7 + 0.5(9 - 0.7) = 4.85$$

Sehingga diperoleh hasil dari pembaruan bobot cluster w pada *query* adalah seperti pada Tabel 3.10. Bobot Query Pada Iterasi 1:

Tabel 3.10. Bobot Query Pada Iterasi 1

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
w_1	4.85	0.4	0	0.25	4.6	5	0	0.2	0.3	0.15	0.2	0.3	0	0.1	0.15
w_2	0.1	1	0.1	0.4	0.6	0.2	0.7	0.4	0.4	1	0	0	0.7	0	0.7

Lakukan hal yang sama seperti perhitungan *query* pada dokumen 1 dengan langkah pertama hitung jarak dokumen dan menggunakan bobot *cluster* yang telah diperbarui:

$$D1 = (4.85-9)^2 + (0.4-0)^2 + (0-9)^2 + (0.25-9)^2 + (4.6-9)^2 + (5-9)^2 + (0-0)^2 + (0.2-0)^2 + (0.3-0)^2 + (0.15-0)^2 + (0.2-0)^2 + (0.3-0)^2 + (0-4.5)^2 + (0.1-0)^2 + (0.15-0)^2$$

$$= 230.87$$

$$D2 = (0.1-9)^2 + (1-0)^2 + (0.1-9)^2 + (0.4-9)^2 + (0.6-9)^2 + (0.2-9)^2 + (0.7-0)^2 + (0.4-0)^2 + (0.4-0)^2 + (1-0)^2 + (0-0)^2 + (0-0)^2 + (0.7-4.5)^2 + (0-0)^2 + (0.7-0)^2$$

$$= 398.12$$

Pada perhitungan jarak dokumen 1 diatas $D1$ memiliki jarak yang lebih kecil, maka bobot *cluster* ke-1 (w_1) yang akan diperbarui bobotnya. Lakukan perhitungan dibawah hingga pada $w_{1,15}$ untuk memperbarui bobot *cluster*.

$$w_{1,1} = 4.85 + 0.5(9 - 4.85) = 6.93$$

Sehingga diperoleh hasil dari pembaruan bobot cluster w pada dokumen 1 adalah seperti pada Tabel 3.11. Bobot Dokumen 1 Pada Iterasi 1:

Tabel 3.11. Bobot Dokumen 1 Pada Iterasi 1

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
w_1	6.93	0.2	4.5	4.63	6.8	7	0	0.1	0.15	0.08	0.1	0.15	2.25	0.05	0.08
w_2	0.1	1	0.1	0.4	0.6	0.2	0.7	0.4	0.4	1	0	0	0.7	0	0.7

Lakukan perhitungan yang sama pada dokumen 2 dengan langkah pertama hitung jarak dokumen dan menggunakan bobot *cluster* yang telah diperbarui:

$$\begin{aligned}
 D1 &= (6.93-0)^2 + (0.2-0)^2 + (4.5-0)^2 + (5.63-0)^2 + (6.8-0)^2 + (7-0)^2 + (0-9)^2 + (0.1-9)^2 \\
 &\quad + (0.15-9)^2 + (0.08-9)^2 + (0.1-9)^2 + (0.15-0)^2 + (2.25-0)^2 + (0.05-0)^2 + (0.08-0)^2 \\
 &= 587.3675
 \end{aligned}$$

$$\begin{aligned}
 D2 &= (0.1-0)^2 + (1-0)^2 + (0.1-0)^2 + (0.4-0)^2 + (0.6-0)^2 + (0.2-0)^2 + (0.7-9)^2 + (0.4-9)^2 \\
 &\quad + (0.4-9)^2 + (1-9)^2 + (0-9)^2 + (0-0)^2 + (0.7-0)^2 + (0-0)^2 + (0.7-0)^2 \\
 &= 364.37
 \end{aligned}$$

Pada perhitungan jarak dokumen 2 diatas $D2$ memiliki jarak yang lebih kecil, maka bobot *cluster* ke-2 (w_2) yang akan diperbarui bobotnya. Lakukan perhitungan dibawah hingga pada $w_{2,15}$ untuk memperbarui bobot *cluster*.

$$w_{2,1} = 0.1 + 0.5(0 - 0.1) = 0.05$$

Sehingga diperoleh hasil dari pembaruan bobot cluster w pada dokumen 2 adalah seperti pada Tabel 3.12. Bobot Dokumen 2 Pada Iterasi 1:

Tabel 3.12. Bobot Dokumen 2 Pada Iterasi 1

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
w_1	6.93	0.2	4.5	4.63	6.8	7	0	0.1	0.15	0.08	0.1	0.15	2.25	0.05	0.08
w_2	0.05	0.5	0.05	0.2	0.3	0.1	4.85	4.7	4.7	5	4.5	0	0.35	0	0.35

Lakukan perhitungan yang sama pada dokumen 3 dengan langkah pertama hitung jarak dokumen dan menggunakan bobot *cluster* yang telah diperbarui:

$$\begin{aligned}
 D1 &= (6.93-0)^2 + (0.2-0)^2 + (4.5-9)^2 + (5.63-0)^2 + (6.8-9)^2 + (7-9)^2 + (0-0)^2 + (0.1-0)^2 \\
 &\quad + (0.15-0)^2 + (0.08-0)^2 + (0.1-0)^2 + (0.15-9)^2 + (2.25-9)^2 + (0.05-0)^2 + (0.08-0)^2 \\
 &= 222.4175
 \end{aligned}$$

$$\begin{aligned}
 D2 &= (0.05-0)^2 + (0.5-0)^2 + (0.05-9)^2 + (0.2-0)^2 + (0.3-9)^2 + (0.1-9)^2 + (4.85-0)^2 + \\
 &\quad (4.7-0)^2 + (4.7-0)^2 + (5-0)^2 + (4.5-0)^2 + (0-9)^2 + (0.35-9)^2 + (0-0)^2 + (0.35-0)^2 \\
 &= \mathbf{504.1925}
 \end{aligned}$$

Pada perhitungan jarak dokumen 3 diatas $D1$ memiliki jarak yang lebih kecil, maka bobot *cluster* ke-1 (w_1) yang akan diperbarui bobotnya. Lakukan perhitungan dibawah hingga pada $w_{1,15}$ untuk memperbarui bobot *cluster*.

$$w_{1,1} = 6.93 + 0.5(0 - 6.93) = 3.46$$

Sehingga diperoleh hasil dari pembaruan bobot *cluster* w pada dokumen 3 adalah seperti pada :

Tabel 3.13. Bobot Dokumen 3 Pada Iterasi 1

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
w_1	3.46	0.1	6.75	2.31	7.9	8	0	0.05	0.08	0.04	0.05	4.58	5.63	0.03	0.04
w_2	0.05	0.5	0.05	0.2	0.3	0.1	4.85	4.7	4.7	5	4.5	0	0.35	0	0.35

Lakukan perhitungan yang sama pada dokumen 4 dengan langkah pertama hitung jarak dokumen dan menggunakan bobot *cluster* yang telah diperbarui:

$$\begin{aligned}
 D1 &= (3.46-0)^2 + (0.1-9)^2 + (6.75-0)^2 + (2.31-0)^2 + (7.9-0)^2 + (8-0)^2 + (0-9)^2 + \\
 &\quad (0.05-0)^2 + (0.08-0)^2 + (0.04-9)^2 + (0.05-1-0)^2 + (4.58-0)^2 + (5.63-0)^2 + (0.03)^2 + \\
 &\quad (0.04-9)^2 = \mathbf{643.3044}
 \end{aligned}$$

$$\begin{aligned}
 D2 &= (0.05-0)^2 + (0.5-9)^2 + (0.05-0)^2 + (0.2-0)^2 + (0.3-0)^2 + (0.1-0)^2 + (4.85-9)^2 + \\
 &\quad (4.7-0)^2 + (4.7-0)^2 + (5-9)^2 + (4.5-0)^2 + (0-0)^2 + (0.35-0)^2 + (0-9)^2 + (0.35-9)^2 \\
 &= \mathbf{325.9925}
 \end{aligned}$$

Pada perhitungan jarak dokumen 3 diatas D_2 memiliki jarak yang lebih kecil, maka bobot *cluster* ke-2 (w_2) yang akan diperbarui bobotnya. Lakukan perhitungan dibawah hingga pada $w_{2,15}$ untuk memperbarui bobot *cluster*.

$$w_{2,1} = 0.05 + 0.5(0 - 0.05) = 0.03$$

Sehingga diperoleh hasil dari pembaruan bobot cluster w pada dokumen 4 adalah seperti pada :

Tabel 3.14. Bobot Dokumen 4 Pada Iterasi 1

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
w1	3.46	0.1	6.75	2.31	7.9	8	0	0.05	0.08	0.04	0.05	4.58	5.63	0.03	0.04
w2	0.03	4.75	0.03	0.1	0.15	0.05	6.93	2.35	2.35	7	2.25	0	0.18	4.5	4.68

Pada iterasi pertama menunjukkan bahwa *query*, dokumen 1 dan dokumen 3 termasuk kedalam *cluster* 1, kemudian dokumen 2 dan dokumen 4 termasuk kedalam *cluster* 2.

Iterasi 2

Pada iterasi ke-2 dan pada setiap iterasi selanjutnya dilakukan perbaikan laju pembelajaran (α) 0,6 seperti berikut:

$$\begin{aligned}\alpha(\text{baru}) &= 0.6 \times \alpha(\text{lama}) \\ \alpha(\text{baru}) &= 0.6 \times 0.5 = 0.3\end{aligned}$$

Jadi pada iterasi ke-2 ini akan menggunakan laju pembelajaran 0,3 pada penghitungan bobot *cluster* (w). Berikut merupakan hasil perhitungan bobot *cluster* iterasi ke-2 seperti pada Tabel 3.15. Bobot Dokumen Pada Iterasi 2:

Tabel 3.15. Bobot Dokumen Pada Iterasi 2

<i>i</i>	Q		Doc1		Doc2		Doc3		Doc4	
	w1	w2	w1	w2	w1	w2	w1	w2	w1	w2
1	5.12	0.03	6.29	0.03	6.29	0.02	4.40	0.02	4.40	0.01
2	0.07	4.75	0.05	4.75	0.05	3.33	0.03	3.33	0.03	5.03
3	4.73	0.03	6.01	0.03	6.01	0.02	6.91	0.02	6.91	0.01
4	1.62	0.10	3.83	0.10	3.83	0.07	2.68	0.07	2.68	0.05
5	8.23	0.15	8.46	0.15	8.46	0.11	8.62	0.11	8.62	0.07
6	8.30	0.05	8.51	0.05	8.51	0.04	8.66	0.04	8.66	0.02
7	0.00	6.93	0.00	6.93	0.00	7.55	0.00	7.55	0.00	7.98
8	0.04	2.35	0.02	2.35	0.02	4.35	0.02	4.35	0.02	3.04
9	0.05	2.35	0.04	2.35	0.04	4.35	0.03	4.35	0.03	3.04
10	0.03	7.00	0.02	7.00	0.02	7.60	0.01	7.60	0.01	8.02
11	0.04	2.25	0.02	2.25	0.02	4.28	0.02	4.28	0.02	2.99
12	3.20	0.00	2.24	0.00	2.24	0.00	4.27	0.00	4.27	0.00
13	3.94	0.18	4.11	0.18	4.11	0.12	5.57	0.12	5.57	0.09
14	0.02	4.50	0.01	4.50	0.01	3.15	0.01	3.15	0.01	4.91
15	0.03	4.68	0.02	4.68	0.02	3.27	0.01	3.27	0.01	4.99

Angka yang bercetak tebal merupakan bobot *cluster* yang diperbarui sehingga iterasi dapat dihentikan pada iterasi 2 karena hasil *cluster* iterasi 2 menunjukkan *konvergen* terhadap iterasi 1, yaitu secara berturut-turut mulai dari *query* sampai dokumen 4 menghasilkan *cluster* [1,1,2,1,2]. Jadi anggota cluster 1 adalah dokumen 1, 3, dan *query*, sedangkan anggota cluster 2 adalah dokumen 2 dan 4.

Hasil *clustering* menggunakan *Self-Organizing Map* (SOM) dan pengelompokan secara manual dengan kelompok 1 merupakan kategori sosial dan

kelompok 2 merupakan kategori teknologi dapat dilihat pada Tabel 3.16. Hasil Cluster dan Manual berikut:

Tabel 3.16. Hasil Cluster dan Manual

Dokumen		Pengelompokan Manual		Self Organizing Map (SOM)	
		1	2	1	2
D(1)	Warga desa membangun jembatan dengan gotong royong	*		*	
D(2)	Pengolahan limbah dilakukan dengan mesin cangkih		*		*
D(3)	Masyarakat gotong royong membangun sekolahan	*		*	
D(4)	Petani mengolah sawah dengan mesin traktor		*		*
Q	Warga sekitar gotong royong	*		*	

Proses terakhir adalah menampilkan hasil dari perhitungan kepada *user* sesuai dengan *query* yang dimasukkan. Dokumen yang ditampilkan ke *user* merupakan dokumen yang mempunyai nilai kemiripan lebih dari 0 secara berurutan sesuai tingkat kemiripan dan yang merupakan anggota *cluster* dari *query* yang dimasukkan *user*. Berdasarkan aturan tersebut maka dokumen yang ditampilkan kepada *user* adalah dokumen 1 dan 3 dengan nilai tingkat kemiripan berturut-turut adalah 0.7559 dan 0.5164.

3.3.10 Implementasi GUI

Berikut merupakan Gambar 3.20. Implementasi GUI dari aplikasi Information Retrieval System (IRS) Berita Jawa Timur sesuai dengan desain interface yang telah dibuat.



Gambar 3.20. Implementasi GUI

Keterangan dari setiap poin yang terdapat dalam tampilan adalah sebagai berikut:

1. Teks *Input Query*

Form input-an yang digunakan *user* untuk memasukkan kata kunci (*query*) dari dokumen berita yang ingin dicari.

2. Tombol *Link Home*

Tombol *link* yang digunakan untuk me-*reset* halaman.

3. Tombol *Link Save Data to MySQL*

Tombol untuk mengambil data berita dari media *online* yang kemudian disimpan dalam *database*.

4. Tombol *Link Pre-Processing*

Tombol untuk menjalankan proses preprocessing mulai dari case folding, tokenizing, stemming, hingga stopword removal yang kemudian hasilnya akan disimpan dalam *database*.

5. Tombol *Link Count Term*

Tombol untuk menghitung jumlah *term* pada dokumen yang kemudian hasilnya akan disimpan dalam *database*.

6. Tombol Cari

Tombol untuk melakukan pencarian dokumen berita sesuai *query* atau masukkan dari *user*.

7. *Link* Berita

Tautan untuk menuju ke halaman berita.

BAB IV

UJI COBA DAN PEMBAHASAN

4.1 Langkah-Langkah Uji Coba

Langkah-langkah yang dilakukan untuk uji coba sistem diantaranya adalah sebagai berikut:

1) Pengumpulan data dokumen berita *online*

Pengumpulan dokumen berita dibatasi 3 kategori yaitu wisata, pendidikan dan bencana. Dokumen berita diambil dari 4 media *online* yaitu detik.com, sindonews.com, okezone.com dan jawapos.com yang masing-masing media diambil 10 berita tentang wisata, 10 berita tentang pendidikan dan 10 berita tentang bencana. Jadi total berita yang terkumpul adalah 120 dokumen yang kemudian disimpan dalam *database*. Jumlah dokumen akan menjadi 121 karena ditambah dengan *query* dari *user*. Pengelompokan terdiri dari 3 *cluster*, setiap *cluster* minimal terdapat 1 anggota yang sehingga dapat dikatakan anggota *cluster* tersebar dengan maksimal. Nilai *range* yang digunakan dalam normalisasi *min-max* adalah antara 0 sampai dengan 9, nilai *range* ini dipilih karena telah dilakukan serangkaian percobaan berdasarkan hasil *cluster* yang menggunakan normalisasi *min-max*. Inisialisasi nilai pembelajaran (α) awal menggunakan 0,5 yang kemudian akan diperbarui menggunakan 0,6. Pada inisialisasi awal bobot cluster (w) dilakukan secara *random* atau acak. Hasil pengumpulan data dapat dilihat pada Tabel 4.1. List Judul Dokumen Berita berikut:

Keterangan warna:

- Kategori tentang wisata (ID dokumen 1 s/d 40)
- Kategori tentang pendidikan (ID dokumen 41 s/d 80)
- Kategori tentang bencana (ID dokumen 81 s/d 120)

Tabel 4.1. List Judul Dokumen Berita

ID	Judul Berita
1	Kota Tua di Kawasan Surabaya Utara akan Disulap Jadi Wisata Urban
2	Kemenpar Serius Kembangkan Wisata Alam di Banyuwangi
3	Risma akan Sulap Surabaya Utara Jadi Wisata Kota Tua
4	Menpar Kepincut Hotel di Banyuwangi, Seperti Apa?
5	Kecamatan di Pasuruan Didorong Bikin Kampung Wisata Kopi
6	Khofifah Serius Garap Destinasi Wisata di Selatan Jatim
7	Kemenpar Gali Potensi Wisata Sejarah di Jawa Timur
8	Gus Ipul Ingin Wisata Religi di Jawa Timur Dilengkapi Museum
9	Puti Ingin Film Nasional Promosikan Wisata Jatim
10	Yuk ke Banyuwangi, Ada Destinasi Wisata Baru di Tengah Hutan Jati
11	Wajah Baru Wisata Pacet, Bagaimana Penampakannya?
12	Setiap Tahun Jutaan Wisatawan Kunjungi Kabupaten Malang
13	Geopark dan Cagar Biosfer Dunia, Pendorong Baru Wisata Banyuwangi
14	Di Tuban, Puti Guntur Angkat Wisata Religi di Jawa Timur
15	Banyuwangi Genjot Paket Peleisir Pelaku Wisata India hingga Malaysia
16	Kecantikan Alam Mojokerto Hadirkan Pesona Wisata Baru
17	Menunggangi Taxi Manusia di Kawah Ijen Banyuwangi
18	Wisata Alam Mojokerto Makin Diminati
19	4 Tempat Liburan Hit di Malang
20	Nikmati Sensasi Kuliner Dipadu dengan Alam di Taman Sungai Sumber Biru Jombang

21	Investor AS Tanamkan Modal di Sektor Pariwisata Jawa Timur
22	Wisata Religi di Jawa Timur Potensial Dikembangkan
23	Berwisata ke Blitar Menikmati Hamparan Kebun Kopi hingga Menengok Peninggalan Bung Karno di Hotel Antik
24	Diserbu Wisatawan di Bangsring Underwater, Gus Ipul Beberkan Program Seribu Dewi
25	Mau Liburan Seru? Yuk Mampir ke Wisata Rumah Negeri Atas Air Bojonegoro yang Instagramable
26	Destinasi Wisata Surabaya Mencuri Perhatian di Brunei Travel Fair 2018
27	Pesona Pantai Telok, Malang yang Tersembunyi di Balik Semak-Semak
28	Objek Wisata Gunung Kelud Dibuka untuk Umum pada 2019
29	Bukit Daun, Destinasi Wisata Bekas Tambang Kapur
30	Destinasi Banyuwangi Naik Daun, Pelaku Wisata Sayangkan Hal Ini
31	Kampung Warna Warni Masuk Nominasi Anugerah Wisata Jatim 2018
32	Pukau Ribuan Wisatawan, Festival Gandrung Sewu Banjir Pujian
33	Puti Soekarno Dorong Film Nasional Promosikan Wisata Jatim
34	Long Weekend, Ribuan Wisatawan Padati Pantai di Malang
35	Guwoterus, Wisata Berbasis Sedekah
36	Disbudpar Kota Malang Genjot Realisasi Wisata Halal
37	Yuk Berwisata ke Pantai Paling Bersih se-Asia Tenggara di Banyuwangi
38	Pemkab Malang dan Lumajang Saling Klaim Air Terjun Coban Sewu
39	Pencapaian Asli Daerah Sektor Pariwisata Harus Dimaksimalkan
40	Menikmati Desa Pujon Kidul, Agrowisata Ngehits di Kabupaten Malang
41	Jenjang Pendidikan di Jatim Masih Rendah, Ini Kata Khofifah
42	Tahun Depan Pemprov Jatim Pastikan SPP dan Seragam Gratis
43	Ditanya Tentang Pendidikan, Emil Sodorkan Tis Tas
44	Gus Ipul Ajak Muhammadiyah Majukan Pendidikan di Jawa Timur
45	Emil Dardak: Kami Pro Pendidikan Gratis
46	Guru Honorer Mogok, Pendidikan di Kabupaten Malang Bakal Lumpuh
47	Bukan Semen, Beton Bikin ITS Ini Berbahan Limbah Kelapa Sawit
48	16 Perguruan Tinggi Swasta Sepakat Ubah Standar Pendidikan Pancasila
49	Keren! Pemkot Surabaya Beri Beasiswa Khusus Penghafal Al Quran

50	Dinas Pendidikan Kota Blitar Larang Guru Beri PR ke Siswa
51	Pemprov Jatim Terapkan Pendidikan Double Track SMA
52	Anak-Anak Pemulung Ini Bisa Mendapatkan Pendidikan Layak
53	Tolak SPP SMA/SMK Negeri Naik, Gus Ipul-Puti Usung Pendidikan Gratis
54	Selamat! Institut Teknologi Telkom Surabaya Resmi Dibuka
55	Sukseskan Pendidikan Gratis, Pemkot Blitar Alokasikan Rp60 Miliar
56	Soekarwo: Pengembangan Pendidikan Harus Didukung Kesehatan
57	Pemprov Jatim Terapkan Pendidikan Double Track SMA
58	Sistim Online PPDB Tutup Peluang Praktik Titipan Siswa
59	Jawa Timur Juara Gala Siswa Indonesia (GSI) 2018
60	Dinas Pendidikan Kota Blitar Larang Sekolah Beri PR Terhadap Siswa
61	10 Tahun Murid SD di Tulungagung Belajar di Samping Kandang Sapi
62	Menag Akan Sanksi Sekolah yang Tolak Pancasila dan Upacara Bendera
63	Gus Ipul-Puti Tolak Wacana Kenaikan SPP SMA/SMK
64	Bupati Malang Jadi Tersangka Korupsi, Total Nilainya Rp7 Miliar
65	Ingin Pendidikan Gratis di Jatim, Alasan Risma Ajak Masyarakat Pilih Gus Ipul-Puti
66	Keadilan Pendidikan dan Zonasi Sekolah
67	Bangun Sekolah untuk Anak Tak Mampu, Yuni Shara Pungut Biaya Rp5 Ribu
68	Universitas Airlangga Kini Miliki RS Hewan Pendidikan
69	Peran Pesantren Dalam Pembangunan Ekonomi Masyarakat
70	21 SMA/SMK di Bondowoso Belum Bisa Gelar UNBK
71	Ajarkan Anak Berbisnis Dengan Nol Rupiah
72	Persiapan Unas, Dinas Pendidikan Surabaya Tambah 10 Rumah Matematika
73	BMPS: Apa Pemerintah Ingin Sekolah Swasta Habis Perlahan?
74	Hari Ini, 91 Persen Siswa SMA dan MA Mulai UNBK
75	Kepala SMKN 1 Surabaya Tidak Diberi Sanksi Setelah Tampar Tiga Siswa
76	SMA Swasta di Surabaya Dominasi Peringkat 10 Besar
77	Nilai UN SMA Sederajat Jatim Jeblok, Guru dan Kasek Siap-siap Dimutasi

78	Gelar UNBK, 6 SMK di Bojonegoro Nebeng
79	Risma Ajak Puti Promosikan Pendidikan
80	10 Tahun Berjuang Tumbuhkan Literasi di Kalangan Anak Muda
81	Tambang Emas Banyuwangi Beri Bantuan Korban Bencana Sigi Sulteng
82	2 Patahan Aktif di Surabaya, Pakdhe: Kita Siap Kalau Ada Bencana
83	Masuki Musim Hujan, Pemprov Jatim Petakan Daerah Rawan Bencana
84	Hujan Semalaman, Wilayah Trenggalek Diterjang Banjir dan Longsor
85	Banjir dan Longsor Intai Jatim di Musim Hujan, Ini Langkah Pemprov
86	Musim Hujan, 9 Kecamatan di Ponorogo Rawan Longsor
87	Waspada Banjir Bandang, BPBD Minta 16 Desa di Banyuwangi Siaga
88	Waspada, 18 Titik Jalur KA Pasuruan-Banyuwangi Ini Rawan Bencana
89	Rumah-rumah Warga di Probolinggo Ikut Terendam Banjir
90	Korban Banjir Bandang di Banyuwangi Butuh Bahan Material
91	Selama 2017, Jatim Dilanda 382 Bencana Alam
92	Bencana Alam Masih Mengintai Sejumlah Wilayah Blitar
93	Muhammadiyah Berikan Respon Cepat Bencana di Palu dan Donggala
94	Banjir Bandang di Musim Kemarau Terjang Banyuwangi, 325 Rumah Rusak
95	Diguyur Hujan Deras, Longsor Timbun 5 Rumah Warga Tulungagung
96	Perkuat Mitigasi Bencana, Pemkot Surabaya dan SAR Perkuat Water Rescue
97	Di Lumajang Rumah Warga Rusak Diterjang Longsor
98	BNPB : Korban Gempa Situbondo Akan Mendapat Bantuan
99	Tagana Jawa Timur, Siap Berangkat ke Palu dan Donggala
100	PWNU Instruksikan Warganya Galang Donasi untuk Korban Gempa Lombok
101	Ini Penyebab Gempa 4,5 SR Rusak Puluhan Bangunan di Sumenep
102	Puluhan Rumah Rusak Akibat Gempa di Sumenep Tersebar dalam 3 Kecamatan Ini
103	Puluhan Bangunan Rusak Akibat Gempa di Sumenep, 3 Orang Luka-Luka
104	Info Terkini Gempa Situbondo, 26 Korban Terluka dan 36 Rumah Rusak
105	Rumah Rusak di Sumenep Akibat Gempa Ditanggung Pemprov Jatim

106	Banjir Landa 2 Kecamatan di Pamekasan, Aktivitas Warga Masih Normal
107	Gempa Situbondo Tidak Ada Kaitan dengan Bencana di Sulteng
108	Rumah Warga di Lumajang Rusak Diterjang Longsor
109	Gempa Situbondo: 246 Rumah Rusak, 1 di Antaranya Rata dengan Tanah
110	Gempa Bumi 5,0 SR Guncang Pacitan Jatim
111	Gempa 6,4 SR Guncang Jatim dan Bali, 3 Warga Madura Meninggal Dunia
112	Bencana di Kota Malang Timbulkan Kerugian Rp 5,3 Miliar
113	650 Personel Perkuat Satgas Bhayangkara Siaga Bencana
114	105 Bencana Landa Kota Malang
115	Kota Malang Surga Bencana, 3 Bulan Tercatat 71 Kasus
116	Hal Ini Membuat Potensi Bencana di Malang Menurun
117	Banjir Bandang Terjang Enam Kecamatan di Bojonegoro
118	Gempa Bumi Di Jatim Tak Berpotensi Tsunami, Ini Penjelasan BMKG
119	Bencana Longsor Ancam Jatim
120	Jatim Gempa 6,3 SR, Pakar Geofisika: Belum yang Terbesar

2) *Preprocessing* dokumen berita

Preprocessing dokumen berita dilakukan dengan meng-klik *link* menu *Pre-processing* pada halaman aplikasi. Lama dari proses ini ditentukan dari banyaknya dokumen berita yang terdapat dalam database. Semakin banyak dokumen berita yang terdapat dalam *database*, maka pada proses ini akan semakin lama. Proses berakhir ditandai dengan tampilnya hasil *preprocessing* dari dokumen berita terakhir yang disimpan.

3) Penghitungan jumlah kata pada dokumen berita

Setelah *preprocessing* selesai maka proses selanjutnya adalah menghitung jumlah kata pada dokumen dengan meng-klik *link* menu *Count Term* pada halaman aplikasi. Selesaiya proses penghitungan jumlah kata dokumen berita

ditandai dengan tampilnya hasil penghitungan jumlah kata pada dokumen terakhir berita yang tersimpan dalam *database*.

4) Masukkan kata kunci atau *query*

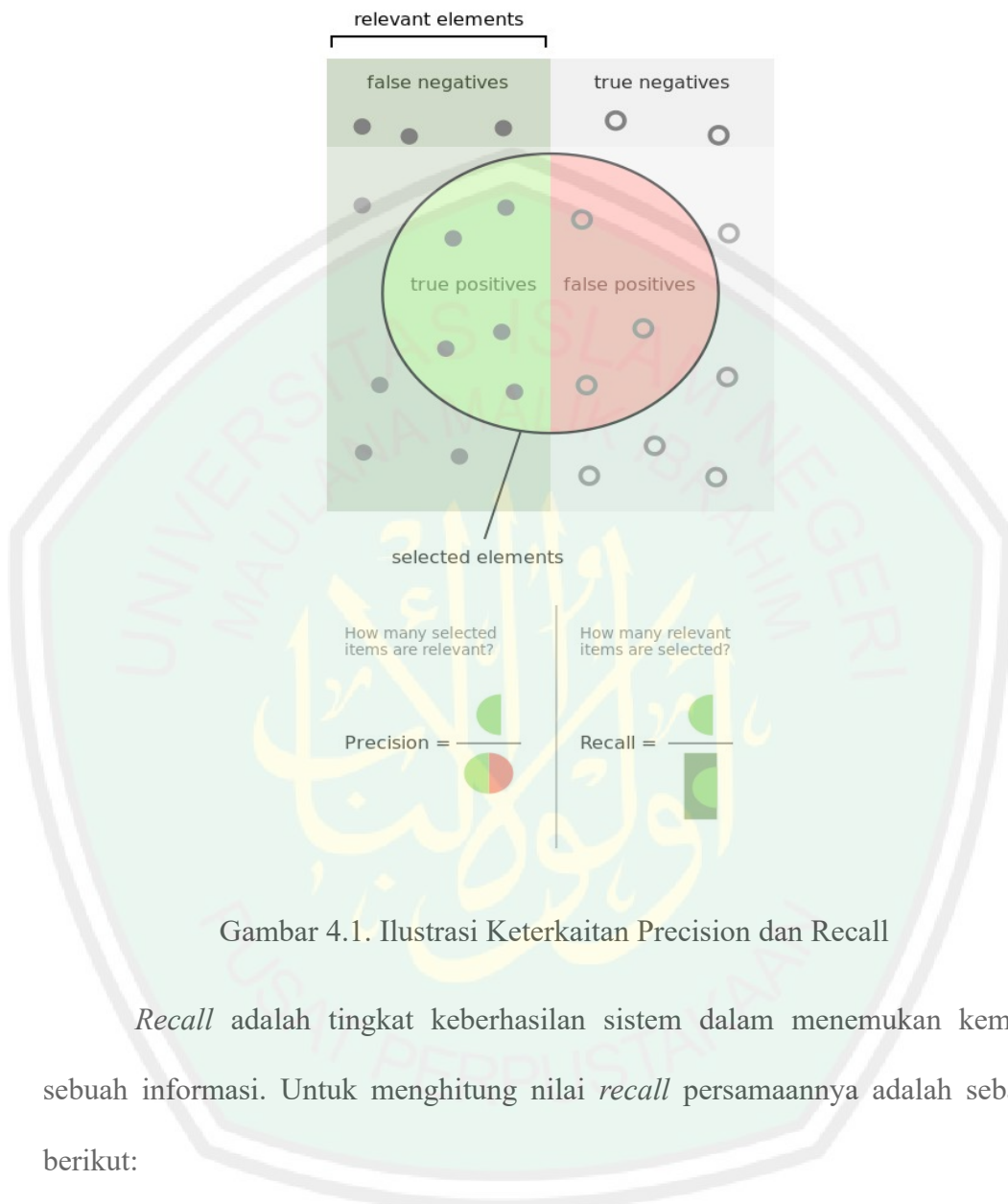
Dalam langkah ini akan dimasukkan *query* atau kata kunci untuk mencari dokumen yang diinginkan, sehingga dokumen yang akan diolah menjadi 121 dokumen termasuk *query*. Setelah memasukkan *query* dan menekan tombol cari maka sistem akan melakukan proses pencarian dokumen berdasarkan *query* yang dimasukkan.

5) Hasil Pencarian

Proses terakhir dari sistem adalah menampilkan judul dokumen, ulasan singkat berita (*headline news*) dan tanggal berita berdasarkan *query* yang dimasukkan oleh pengguna. Di dalam judul berita tersematkan *link* untuk mengarahkan ke halaman *online*, jadi apabila judul berita diklik maka sistem akan mengarahkan ke sumber halaman berita *online*.

6) Evaluasi hasil

Langkah terakhir dari uji coba adalah melakukan evaluasi dari hasil pengolahan sistem. Evaluasi sistem dengan tujuan melihat tingkat keberhasilan *information retrieval system* adalah dengan melihat hasil dari perhitungan *recall*, *precision*, *f-measures* dan akurasi. Untuk ilustrasi keterkaitan antara *precision* dan *recall* dapat dilihat pada Gambar 4.1. Ilustrasi Keterkaitan Precision dan Recall.



Gambar 4.1. Ilustrasi Keterkaitan Precision dan Recall

Recall adalah tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. Untuk menghitung nilai *recall* persamaannya adalah sebagai berikut:

$$\text{Recall} = \frac{\text{Jumlah dokumen relevan yang didapatkan sistem (TP)}}{\text{Jumlah dokumen dalam database (TP + FN)}}$$

Precision adalah tingkat ketepatan antar informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. Untuk menghitung nilai *precision* persamaannya adalah sebagai berikut:

$$\text{Precision} = \frac{\text{Jumlah dokumen relevan yang didapatkan sistem (TP)}}{\text{Jumlah dokumen yang didapatkan sistem (TP + FP)}}$$

F-Measures merupakan salah satu perhitungan evaluasi dalam informasi temu kembali yang mengkombinasikan recall dan precision. Nilai recall dan precision pada suatu keadaan dapat memiliki bobot yang berbeda. Ukuran yang menampilkan timbal balik antara recall dan precision adalah *F-Measures* yang merupakan bobot rata-rata hitung harmonik dari recall dan precision. Untuk menghitung nilai *F-Measures* persamaannya adalah sebagai berikut:

$$F\text{-Measures} = \frac{2 \times (\text{Recall} \times \text{Precision})}{\text{Recall} + \text{Precision}}$$

Akurasi merupakan kesamaan atau kedekatan suatu hasil pengukuran dengan angka atau data yang sebenarnya. Berikut merupakan persamaan untuk menghitung nilai akurasi:

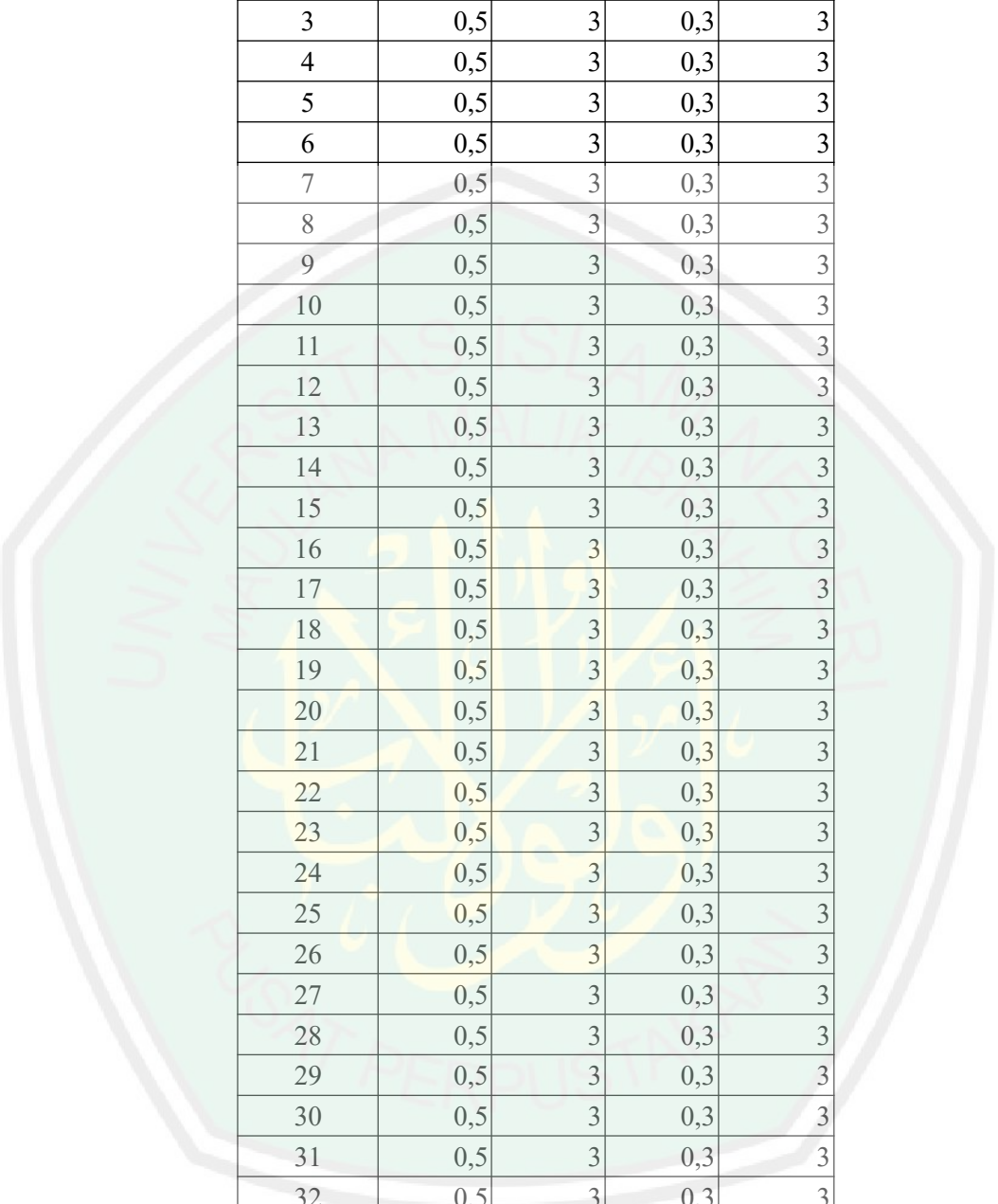
$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN}$$

4.2 Hasil Uji Coba

Uji coba sistem dilakukan dengan menggunakan 9 query yang berbeda. Diperoleh hasil cluster pada query pertama yaitu “Bencana gempa bumi” seperti berikut:

Tabel 4.2. Hasil Cluster dengan Range Nilai Min-Max 0 s/d 1

Document	Iterasi 1		Iterasi 2	
	<i>a</i>	<i>cluster</i>	<i>a</i>	<i>cluster</i>
Q	0,5	3	0,3	3
1	0,5	3	0,3	3
2	0,5	3	0,3	3



3	0,5	3	0,3	3
4	0,5	3	0,3	3
5	0,5	3	0,3	3
6	0,5	3	0,3	3
7	0,5	3	0,3	3
8	0,5	3	0,3	3
9	0,5	3	0,3	3
10	0,5	3	0,3	3
11	0,5	3	0,3	3
12	0,5	3	0,3	3
13	0,5	3	0,3	3
14	0,5	3	0,3	3
15	0,5	3	0,3	3
16	0,5	3	0,3	3
17	0,5	3	0,3	3
18	0,5	3	0,3	3
19	0,5	3	0,3	3
20	0,5	3	0,3	3
21	0,5	3	0,3	3
22	0,5	3	0,3	3
23	0,5	3	0,3	3
24	0,5	3	0,3	3
25	0,5	3	0,3	3
26	0,5	3	0,3	3
27	0,5	3	0,3	3
28	0,5	3	0,3	3
29	0,5	3	0,3	3
30	0,5	3	0,3	3
31	0,5	3	0,3	3
32	0,5	3	0,3	3
33	0,5	3	0,3	3
34	0,5	3	0,3	3
35	0,5	3	0,3	3
36	0,5	3	0,3	3
37	0,5	3	0,3	3
38	0,5	3	0,3	3
39	0,5	3	0,3	3
40	0,5	3	0,3	3
41	0,5	3	0,3	3
42	0,5	3	0,3	3

43	0,5	3	0,3	3
44	0,5	3	0,3	3
45	0,5	3	0,3	3
46	0,5	3	0,3	3
47	0,5	3	0,3	3
48	0,5	3	0,3	3
49	0,5	3	0,3	3
50	0,5	3	0,3	3
51	0,5	3	0,3	3
52	0,5	3	0,3	3
53	0,5	3	0,3	3
54	0,5	3	0,3	3
55	0,5	3	0,3	3
56	0,5	3	0,3	3
57	0,5	3	0,3	3
58	0,5	3	0,3	3
59	0,5	3	0,3	3
60	0,5	3	0,3	3
61	0,5	3	0,3	3
62	0,5	3	0,3	3
63	0,5	3	0,3	3
64	0,5	3	0,3	3
65	0,5	3	0,3	3
66	0,5	3	0,3	3
67	0,5	3	0,3	3
68	0,5	3	0,3	3
69	0,5	3	0,3	3
70	0,5	3	0,3	3
71	0,5	3	0,3	3
72	0,5	3	0,3	3
73	0,5	3	0,3	3
74	0,5	3	0,3	3
75	0,5	3	0,3	3
76	0,5	3	0,3	3
77	0,5	3	0,3	3
78	0,5	3	0,3	3
79	0,5	3	0,3	3
80	0,5	3	0,3	3
81	0,5	3	0,3	3
82	0,5	3	0,3	3

83	0,5	3	0,3	3
84	0,5	3	0,3	3
85	0,5	3	0,3	3
86	0,5	3	0,3	3
87	0,5	3	0,3	3
88	0,5	3	0,3	3
89	0,5	3	0,3	3
90	0,5	3	0,3	3
91	0,5	3	0,3	3
92	0,5	3	0,3	3
93	0,5	3	0,3	3
94	0,5	3	0,3	3
95	0,5	3	0,3	3
96	0,5	3	0,3	3
97	0,5	3	0,3	3
98	0,5	3	0,3	3
99	0,5	3	0,3	3
100	0,5	3	0,3	3
101	0,5	3	0,3	3
102	0,5	3	0,3	3
103	0,5	3	0,3	3
104	0,5	3	0,3	3
105	0,5	3	0,3	3
106	0,5	3	0,3	3
107	0,5	3	0,3	3
108	0,5	3	0,3	3
109	0,5	3	0,3	3
110	0,5	3	0,3	3
111	0,5	3	0,3	3
112	0,5	3	0,3	3
113	0,5	3	0,3	3
114	0,5	3	0,3	3
115	0,5	3	0,3	3
116	0,5	3	0,3	3
117	0,5	3	0,3	3
118	0,5	3	0,3	3
119	0,5	3	0,3	3
120	0,5	3	0,3	3

Tabel 4.3. Hasil Cluster dengan Range Nilai Min-Max 0 s/d 9

Document	Iterasi 1		Iterasi 2		Iterasi 3	
	α	cluster	α	cluster	α	cluster
Q	0,5	1	0,3	3	0,18	3
1	0,5	1	0,3	3	0,18	3
2	0,5	2	0,3	3	0,18	3
3	0,5	1	0,3	3	0,18	3
4	0,5	3	0,3	3	0,18	3
5	0,5	3	0,3	3	0,18	3
6	0,5	1	0,3	3	0,18	3
7	0,5	2	0,3	3	0,18	3
8	0,5	1	0,3	3	0,18	3
9	0,5	1	0,3	3	0,18	3
10	0,5	3	0,3	3	0,18	3
11	0,5	1	0,3	3	0,18	3
12	0,5	1	0,3	3	0,18	3
13	0,5	1	0,3	3	0,18	3
14	0,5	1	0,3	3	0,18	3
15	0,5	1	0,3	3	0,18	3
16	0,5	1	0,3	3	0,18	3
17	0,5	1	0,3	3	0,18	3
18	0,5	1	0,3	3	0,18	3
19	0,5	1	0,3	3	0,18	3
20	0,5	1	0,3	3	0,18	3
21	0,5	1	0,3	3	0,18	3
22	0,5	1	0,3	3	0,18	3
23	0,5	1	0,3	3	0,18	3
24	0,5	2	0,3	3	0,18	3
25	0,5	2	0,3	2	0,18	2
26	0,5	2	0,3	2	0,18	2
27	0,5	1	0,3	3	0,18	3
28	0,5	1	0,3	3	0,18	3
29	0,5	1	0,3	3	0,18	3
30	0,5	1	0,3	3	0,18	3
31	0,5	1	0,3	3	0,18	3
32	0,5	1	0,3	3	0,18	3
33	0,5	1	0,3	3	0,18	3
34	0,5	1	0,3	3	0,18	3
35	0,5	1	0,3	3	0,18	3

36	0,5	1	0,3	3	0,18	3
37	0,5	1	0,3	3	0,18	3
38	0,5	1	0,3	3	0,18	3
39	0,5	1	0,3	3	0,18	3
40	0,5	1	0,3	3	0,18	3
41	0,5	1	0,3	3	0,18	3
42	0,5	1	0,3	3	0,18	3
43	0,5	1	0,3	3	0,18	3
44	0,5	1	0,3	3	0,18	3
45	0,5	1	0,3	3	0,18	3
46	0,5	1	0,3	3	0,18	3
47	0,5	1	0,3	3	0,18	3
48	0,5	1	0,3	3	0,18	3
49	0,5	1	0,3	3	0,18	3
50	0,5	1	0,3	3	0,18	3
51	0,5	1	0,3	3	0,18	3
52	0,5	1	0,3	3	0,18	3
53	0,5	3	0,3	3	0,18	3
54	0,5	3	0,3	3	0,18	3
55	0,5	3	0,3	3	0,18	3
56	0,5	3	0,3	3	0,18	3
57	0,5	1	0,3	3	0,18	3
58	0,5	3	0,3	3	0,18	3
59	0,5	3	0,3	3	0,18	3
60	0,5	1	0,3	3	0,18	3
61	0,5	1	0,3	3	0,18	3
62	0,5	1	0,3	3	0,18	3
63	0,5	1	0,3	3	0,18	3
64	0,5	1	0,3	3	0,18	3
65	0,5	1	0,3	1	0,18	1
66	0,5	1	0,3	1	0,18	1
67	0,5	3	0,3	3	0,18	3
68	0,5	3	0,3	3	0,18	3
69	0,5	3	0,3	3	0,18	3
70	0,5	3	0,3	3	0,18	3
71	0,5	3	0,3	3	0,18	3
72	0,5	3	0,3	3	0,18	3
73	0,5	3	0,3	3	0,18	3
74	0,5	3	0,3	3	0,18	3
75	0,5	3	0,3	3	0,18	3

76	0,5	3	0,3	3	0,18	3
77	0,5	3	0,3	3	0,18	3
78	0,5	3	0,3	3	0,18	3
79	0,5	3	0,3	3	0,18	3
80	0,5	3	0,3	3	0,18	3
81	0,5	3	0,3	3	0,18	3
82	0,5	3	0,3	3	0,18	3
83	0,5	3	0,3	3	0,18	3
84	0,5	3	0,3	3	0,18	3
85	0,5	3	0,3	3	0,18	3
86	0,5	3	0,3	3	0,18	3
87	0,5	3	0,3	3	0,18	3
88	0,5	3	0,3	3	0,18	3
89	0,5	3	0,3	3	0,18	3
90	0,5	3	0,3	3	0,18	3
91	0,5	3	0,3	3	0,18	3
92	0,5	3	0,3	3	0,18	3
93	0,5	3	0,3	3	0,18	3
94	0,5	3	0,3	3	0,18	3
95	0,5	3	0,3	3	0,18	3
96	0,5	3	0,3	3	0,18	3
97	0,5	3	0,3	3	0,18	3
98	0,5	3	0,3	3	0,18	3
99	0,5	3	0,3	3	0,18	3
100	0,5	3	0,3	3	0,18	3
101	0,5	3	0,3	3	0,18	3
102	0,5	3	0,3	3	0,18	3
103	0,5	3	0,3	3	0,18	3
104	0,5	3	0,3	3	0,18	3
105	0,5	3	0,3	3	0,18	3
106	0,5	3	0,3	3	0,18	3
107	0,5	3	0,3	3	0,18	3
108	0,5	3	0,3	3	0,18	3
109	0,5	3	0,3	3	0,18	3
110	0,5	3	0,3	3	0,18	3
111	0,5	3	0,3	3	0,18	3
112	0,5	3	0,3	3	0,18	3
113	0,5	3	0,3	3	0,18	3
114	0,5	3	0,3	3	0,18	3
115	0,5	3	0,3	3	0,18	3

116	0,5	3	0,3	3	0,18	3
117	0,5	3	0,3	3	0,18	3
118	0,5	3	0,3	3	0,18	3
119	0,5	3	0,3	3	0,18	3
120	0,5	3	0,3	3	0,18	3

Dari hasil uji coba menggunakan nilai *range* 0 sampai dengan 1 pada normalisasi *min-max* dapat diketahui bahwa iterasi *clustering* dapat diberhentikan pada iterasi ke-2 karena data sudah *konvergen* dengan *query* termasuk kedalam *cluster 2*, namun *cluster* belum tersebar secara maksimal. Sedangkan dari hasil uji coba menggunakan nilai *range* 0 sampai dengan 9 pada normalisasi *min-max* dapat diketahui bahwa iterasi *clustering* dapat diberhentikan pada iterasi ke-3 karena data sudah *konvergen* dengan *query* masuk kedalam *cluster 3*. Dokumen berita dengan ID 25 dan 26 termasuk kedalam *cluster 2*, sedang dokumen berita dengan ID 65 dan 66 termasuk kedalam anggota *cluster 1* dan dokumen berita selebihnya termasuk kedalam anggota *cluster 3*.

Evaluasi *Information Retrieval System* (IRS) dalam penelitian ini adalah dengan menghitung nilai *recall*, *precision* dan *F-measure* dari output yang ditampilkan oleh sistem kepada user yang merupakan hasil dari *clustering* menggunakan *Self Organizing-Map* dan pencocokan similaritas menggunakan *Cosine Similarity*.

Contoh perhitungan evaluasi pada *query* “Bencana gempa bumi” yang hasil dari sistem IR menampilkan sebanyak 42 dokumen dari 120 dokumen, dengan dokumen terpilah dengan benar (TP) sebanyak 40 dokumen dan 2 dokumen termasuk anggota yang lain (FP). Setiap anggota terdapat 40 dokumen

dan 80 dokumen termasuk anggota yang lain. Jadi untuk $FN = (40-TP)$ dan untuk $TN = (80-FP)$. Untuk lebih jelasnya dapat dilihat pada perhitungan berikut:

Tabel 4.4. Confusion Matrix

-		Nilai sebenarnya	
		TRUE	FALSE
Nilai prediksi	TRUE	40 TP (True Positive) <i>Correct result</i>	2 FP (False Positive) <i>Unexpected result</i>
	FALSE	0 FN (False Negative) <i>Missing result</i>	78 TN (True Negative) <i>Correct absence of result</i>

$$Recall = \frac{TP}{(TP+FN)} = \frac{40}{(40+0)} = 1 \times 100 = 100\%$$

$$Precision = \frac{TP}{(TP+FP)} = \frac{40}{(40+2)} = 0,9524 \times 100 = 95,24\%$$

$$F\text{-Measures} = \frac{2 \times (Recall \times Precision)}{Recall + Precision} = \frac{2 \times (1 \times 0,95)}{1 + 0,95} = 0,9756 \times 100 = 97,56\%$$

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} = \frac{40+78}{40+78+2+0} = 0,9833 \times 100 = 98,33\%$$

Dari uji coba menggunakan 9 *query* yang berbeda, diperoleh hasil perhitungan seperti pada Tabel 4.5: Evaluasi Precision, Recall dan F-Measure:

Tabel 4.5: Evaluasi Precision, Recall dan F-Measure

No.	Query	ID Document	n (ID)	T P	F P	T N	F N	Precision		Recall		F-Measures		Akurasi	
								p	%	r	%	F-M	%	a	%
1	Bencana gempa bumi	13, 16, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120	42	40	2	78	0	0.95	95.24	1.00	100	0.98	97.56	0.98	98.33
2	Pendidikan siswa sekolah	8, 9, 31, 32, 41, 42, 43, 44, 45, 46, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 91, 95, 96, 100, 103, 109, 115, 116	49	37	12	68	3	0.76	75.51	0.93	92.50	0.83	83.15	0.88	87.50

3	Wisatawan berpariwisata	1, 2, 4, 7, 8, 9, 10, 11, 12, 13, 15, 16, 17, 18, 20, 21, 22, 23, 24, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 51, 57	35	33	2	78	7	0.94	94.29	0.83	82.50	0.88	88.00	0.93	92.50
4	Pertolongan pada korban bencana	27, 28, 75, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 119	41	38	3	77	2	0.93	92.68	0.95	95.00	0.94	93.83	0.96	95.83
5	Tempat wisata yang indah	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 52, 59, 61, 62, 65, 66, 67, 69, 72, 74, 81, 86, 89, 91, 94, 95, 97, 102, 104, 108, 109, 117	58	36	22	58	4	0.62	62.07	0.90	90.00	0.73	73.47	0.78	78.33
6	Kunjungan	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,	40	38	2	78	2	0.95	95.00	0.95	95.00	0.95	95.00	0.97	96.67

	wisatawan ke wisata yang indah	14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 43, 61														
7	Korban bencana banjir	27, 28, 32, 75, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 119	42	38	4	76	2	0.90	90.48	0.95	95.00	0.93	92.68	0.95	95.00	
8	Didik siswa di sekolah	8, 9, 31, 32, 41, 42, 43, 44, 45, 46, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 91, 95, 96, 100, 103, 109, 115, 116	49	37	12	68	3	0.76	75.51	0.93	92.50	0.83	83.15	0.88	87.50	
9	Kondisi sekolah	14, 20, 26, 41, 42, 43, 44, 45, 46, 49, 50, 52, 54, 55, 56, 58, 59, 60, 61, 62, 63, 65, 66, 67, 70, 71, 72, 73, 74, 75,	42	29	13	67	11	0.69	69.05	0.73	72.50	0.71	70.73	0.80	80.00	

	76, 77, 78, 79, 80, 84, 91, 95, 98, 99, 100, 103, 109, 115, 119													
Rata-rata							0.83	83.31	0.91	90.56	0.86	86.40	0.90	90.19



Persentase *precision*, *recall*, dan *f-measure* diperoleh dari hasil perhitungan dari masing-masing evaluasi dikalikan dengan 100. Dari Tabel 4.5: Evaluasi Precision, Recall dan F-Measure dengan percobaan 9 *query* dapat diketahui tingkat *precision* terendah terdapat pada *query* “Tempat wisata yang indah” dengan nilai 62,07%, kemudian *precision* dan *f-measure* tertinggi terdapat pada *query* “Bencana gempa bumi” dengan masing-masing nilai 95,26% dan 97,56%. Untuk *f-measure* terendah pada *query* “Kondisi sekolah” dengan nilai 70,73%.

Pada evaluasi *recall* dan akurasi tertinggi terdapat pada *query* “Bencana gempa bumi” dengan nilai masing-masing 100% dan 98,33%, sedang *recall* terendah terdapat pada *query* ke 9 dengan nilai 72,50%. Akurasi terendah terdapat pada *query* “Tempat wisata yang indah” dengan tingkat akurasi 78,33%. Diperoleh persentase rata-rata dari evaluasi *precision* adalah 83,31%, *recall* adalah 90,56%, *f-measure* adalah 86,40% dan akurasi adalah 90,19%.

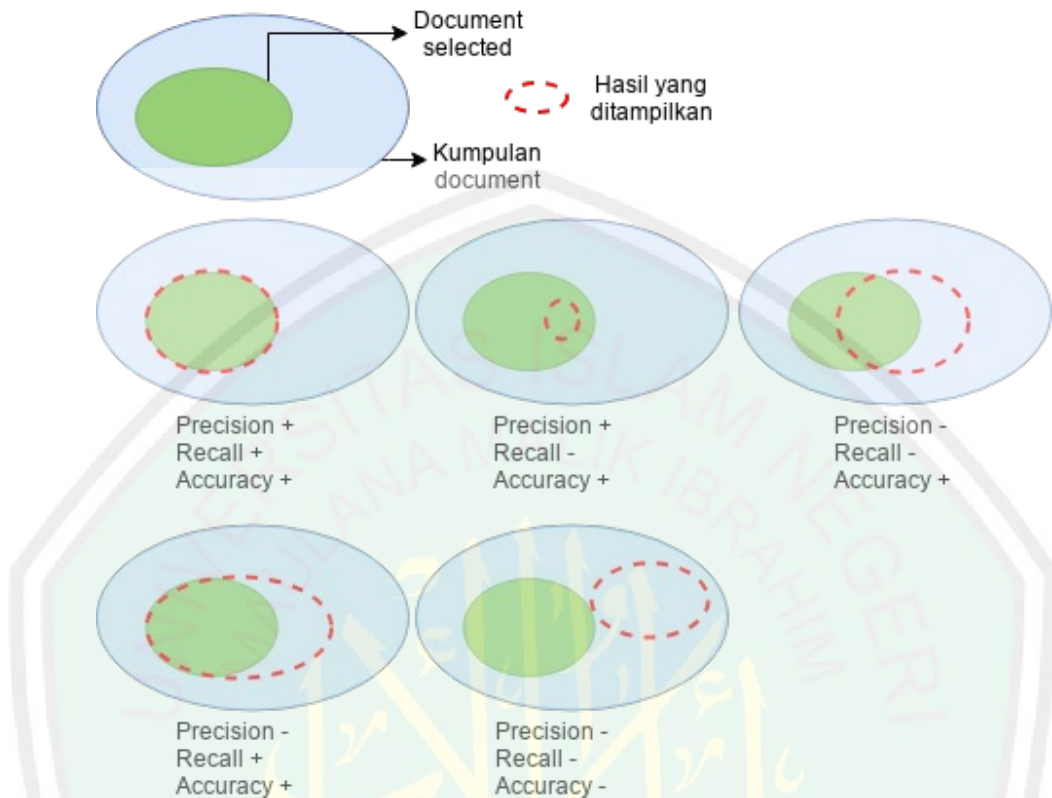
4.3 Pembahasan

Nilai *recall* merupakan nilai yang menunjukkan tingkat perolehan hasil yang dikembalikan oleh sebuah sistem. Nilai ini diperoleh dengan membandingkan jumlah item relevan yang dikembalikan oleh sistem dengan total jumlah item relevan yang ada di dalam koleksi sistem. Nilai *recall* yang semakin besar tidak dapat menunjukkan suatu sistem baik atau tidak. Nilai *recall* tertinggi adalah 1 atau 100%, yang berarti bahwa seluruh dokumen dalam koleksi berhasil ditemukan. Pada percobaan diperoleh nilai rata-rata *recall* dari sistem IR yang dibuat adalah 90,56%.

Nilai *precision* menunjukkan tingkat ketepatan sebuah sistem untuk mengembalikan informasi relevan kepada pengguna. Nilai ini diperoleh dengan membandingkan jumlah item relevan yang dikembalikan dengan total jumlah item yang dikembalikan. Semakin besar nilai *precision* suatu sistem, maka sistem dapat dikatakan baik. Nilai *precision* tertinggi adalah 1 atau 100%, yang berarti seluruh dokumen yang ditemukan adalah relevan. Pada percobaan diperoleh nilai rata-rata *precision* dari sistem IR yang dibuat adalah 83,31%.

Akurasi menunjukkan kesamaan atau kedekatan suatu hasil pengukuran dengan data yang sebenarnya. Pada percobaan diperoleh rata-rata dari akurasi sebesar 90,19%. Semakin tinggi akurasi yang dihasilkan sistem IR maka semakin baik. *F-Measure* merupakan ukuran yang menampilkan timbal balik antara *Recall* dan *Precision*. Nilai *F-Measure* tertinggi adalah 1 atau 100%. Pada percobaan diperoleh hasil rata-rata *F-Measure* sebesar 86,40%.

Untuk mengetahui hubungan antara *recall*, *precision*, dan akurasi dapat dilihat pada Gambar 4.2. Hubungan Precision, Recall dan Akurasi berikut:



Gambar 4.2. Hubungan Precision, Recall dan Akurasi

Berdasarkan hasil percobaan dapat diketahui bahwa hasil *clustering* dapat dipengaruhi oleh penetapan nilai range minimal dan maksimal pada proses normalisasi data menggunakan metode *min-max*. Penetapan nilai *range* yang tepat atau tidak terlalu kecil dapat membuat hasil *clustering* tersebar secara maksimal. Apabila *range* baru yang digunakan terlalu kecil, maka *clustering* akan kesulitan dalam melakukan *cluster* dokumen, akibatnya dokumen tidak tersebar kedalam anggota *cluster* dengan maksimal.

Hasil *preprocessing* seperti proses *stopword removal* dan proses *stemming* pada dokumen berita juga dapat mempengaruhi hasil dari pembobotan yang menggunakan metode *tf-idf*. *Stopword removal* merupakan proses penghapusan kata-kata yang dianggap tidak diperlukan dalam proses pembobotan *tf-idf*.

Semakin cermat dalam pemilihan kata yang dikumpulkan pada *stoplist* maka hasil pembobotan menggunakan *tf-idf* akan semakin bagus, sehingga akan semakin bagus juga hasil *clustering* dokumen berita. Dalam pemilihan *stoplist* juga harus memperhatikan kata-kata yang dimungkinkan di-inputkan oleh *user* atau yang disebut *query*, karena juga akan berpengaruh pada *Information Retrieval System* (IRS). Menurut penelitian (Agastya, 2018), proses stemming itu dapat meningkatkan *recall* tetapi juga dapat menurunkan *precision* karena berkurangnya informasi dari kata yang telah di-*stem*.

Menurut Kamus Besar Bahasa Indonesia (KBBI) berita adalah cerita atau keterangan mengenai kejadian atau peristiwa yang hangat. Berita dalam bahasa Arab menggunakan kata **خَبْرٌ** yang berarti kabar atau keterangan atau berita. Syarat dari berita diantaranya mengandung 5W dan 1H (*what, where, when, who, why* dan *how*), selain itu berita juga harus merupakan informasi yang terbukti kebenarannya atau faktual. Jadi berita merupakan laporan atau informasi atau kabar dari suatu kejadian atau peristiwa yang terbaru atau terkini, yang bermanfaat dan bersifat faktual.

Menurut Mochtar Lubis yang merupakan wartawan asal Indonesia menyatakan bahwa berita adalah informasi yang baru, menarik perhatian, mempengaruhi orang banyak, dan mampu membangkitkan selera masyarakat untuk mengikutinya. Sedangkan menurut pembantu utama redaktur malam New York Times Neil McNeil menjelaskan bahwa bertia adalah gabungan fakta dan peristiwa-peristiwa yang menimbulkan perhatian atau kepentingan bagi para pembaca surat kabar yang memuatnya.

Dalam kitab Tafsir Jalalain pada QS. An-Nissa ayat 83 menerangkan: (Dan apabila datang kepada mereka suatu berita) mengenai hasil-hasil yang dicapai oleh ekspedisi tentara Nabi saw. (berupa keamanan) maksudnya kemenangan (atau ketakutan) maksudnya kekalahan (mereka lalu menyiarkannya). Ayat ini turun mengenai segolongan kaum munafik atau segolongan orang-orang mukmin yang lemah iman mereka, dan dengan perbuatan mereka itu lemahlah semangat orang-orang mukmin dan kecewalah Nabi saw. (Padahal kalau mereka menyerahkannya) maksudnya berita itu (kepada Rasul dan *Uliil amri* di antara mereka) maksudnya para pembesar sahabat, jika mereka diam mengenai berita itu menunggu keputusannya (tentulah akan dapat diketahui) apakah hal itu boleh disiarkan atau tidak (oleh orang-orang yang ingin mengetahui kebenarannya) artinya yang mengikuti perkembangannya dan dituntut untuk mengetahuinya, mereka adalah orang-orang yang berhak menyiarkan berita itu (dari mereka) yakni Rasul dan *Uliil amri* (Dan kalau bukanlah karena karunia Allah kepadamu) yakni dengan agama Islam (serta rahmat-Nya) kepadamu dengan Al-Quran (tentulah kamu sekalian akan mengikuti setan) untuk mengerjakan kekejian-kekejian yang diperintahkannya (kecuali sebagian kecil saja di antaramu) yang tidak.

Dalam Al Quran surat Al-Hujurat (49) ayat 6:

يَا أَيُّهَا الَّذِينَ آمَنُوا إِن جَاءَكُمْ فَاسِقٌ بِنَبَأٍ فَتَبَيَّنُوا أَن تُصِيبُوا قَوْمًا بِجَهَالَةٍ فَتُصْحَبُوا عَلَىٰ مَا فَعَلْتُمْ
نَادِمِينَ

Artinya: “Wahai orang-orang yang beriman! Jika seseorang yang fasik datang kepadamu membawa suatu berita, maka telitilah kebenarannya, agar kamu

tidak mencelakakan suatu kaum karena kebodohan (kecerobohan), yang akhirnya kamu menyesali perbuatanmu itu.”

Sedang dalam kitab Tafsir Al-Jalalain karangan dari 2 imam besar yaitu Syekh Jalaluddin al-Mahalli dan Syekh Jalaluddin al-Suyuthi menafsirkan QS. Al-Hujarat ayat 6 sebagai berikut: (Hai orang-orang yang beriman! Jika datang kepada kalian orang fasik membawa suatu berita) (maka periksalah oleh kalian) kebenaran beritanya itu, apakah ia benar atau berdusta. Menurut suatu *qiraat* dibaca *Fatatsabbatuu* berasal dari lafal *Ats-Tsabaat*, artinya telitilah terlebih dahulu kebenarannya (agar kalian tidak menimpakan musibah kepada suatu kaum) menjadi *Maf’ul* dari lafal *Fatabayyanuu*, yakni dikhawatirkan hal tersebut akan menimpa musibah kepada suatu kaum (tanpa mengetahui keadaannya) menjadi Hal atau kata keterangan keadaan dari *Fa’il*, yakni tanpa sepengetahuannya (yang menyebabkan kalian) membuat kalian (atas perbuatan kalian itu) yakni berbuat kekeliruan terhadap kaum tersebut (menyesal) selanjutnya Rasulullah saw. Mengutus Khalid kepada mereka sesudah mereka kembali ke negerinya. Ternyata Khalid tidak menjumpai mereka melainkan hanya ketaatan dan kebaikan belaka, lalu ia menceritakan hal tersebut kepada Nabi saw.

QS. Al-Hujurat ayat 6 menjelaskan bahwa dalam menyampaikan berita haruslah suatu informasi yang penting dan bermanfaat. Sebelum menyampaikan suatu berita juga harus ber-*tabayyun* atau mencari kejelasan dari berita tersebut. Adapun isu-isu ringan, berita bohong atau omong kosong dan berita yang tidak bermanfaat tidak perlu diselidiki dan diberitakan karena hanya akan menyita waktu, energi dan bahkan dapat menimbulkan fitnah atau musibah. Ketika

mengetahui bahwa berita yang disampaikan itu merupakan berita bohong maka jangan diam saja seperti yang dijelaskan dalam Al-Quran surat An-Nur ayat 12:

لَوْلَا إِذْ سَمِعْتُمُوهُ ظَنَّ الْمُؤْمِنُونَ وَالْمُؤْمِنَاتُ بِأَنفُسِهِمْ خَيْرًا وَقَالُوا هَذَا إِفْكٌ مُّبِينٌ.

Artinya: “Mengapa orang-orang mukmin dan mukminat tidak berbaik sangka terhadap diri mereka sendiri, ketika kamu mendengar berita bohong itu dan berkata, ‘Ini adalah (suatu berita) bohong yang nyata.’”

Dijelaskan juga dalam Al-Quran surat An-Nur ayat 15:

إِذْ تَلَقَّوْنَهُ بِأَلْسِنَتِكُمْ وَتَقُولُونَ بِأَفْوَاهِكُمْ مَا لَيْسَ لَكُم بِهِ عِلْمٌ وَتَحْسَبُونَهُ هَيِّئًا وَهُوَ عِنْدَ اللَّهِ عَظِيمٌ.

Artinya: “(Ingatlah) ketika kamu menerima (berita bohong) itu dari mulut ke mulut dan kamu katakan dengan mulutmu apa yang tidak kamu ketahui sedikitpun, dan kamu menganggapnya remeh, padahal dalam pandangan Allah itu soal besar.”

QS. An-Nur ayat menjelaskan ketika mendapat suatu berita bohong, maka jangan malah ikut menyebarkan dan jangan meremehkan berita bohong itu karena dalam pandangan Allah itu merupakan persoalan yang besar dan bisa berakibat fatal atau bahaya. Hal demikian juga dijelaskan dalam Al-Quran surat Al-Isra’ ayat 36 sebagai berikut:

وَلَا تَقْفُ مَا لَيْسَ لَكَ بِهِ عِلْمٌ إِنَّ السَّمْعَ وَالْبَصَرَ وَالْفُؤَادَ كُلُّ أُولَئِكَ كَانَ عَنْهُ مَسْئُولًا.

Artinya: “Dan janganlah kamu mengikuti sesuatu yang tidak kamu ketahui. Karena pendengaran, penglihatan, dan hati nurani, semua itu akan diminta pertanggung jawabannya.”

Dari Abu Hurairah ra, Rasulullah SAW pernah bersabda:

وَمَنْ كَانَ يُؤْمِنُ بِاللَّهِ وَالْيَوْمِ الْآخِرِ فَلْيَقُلْ خَيْرًا أَوْ لِيَصْمُتْ

Artinya: “Barang siapa yang beriman kepada Allah dan hari akhir maka hendaklah mengucapkan yang baik atau diam.” [HR. Bukhari - Muslim]

Hadits ini menjelaskan bahwa hendaklah orang-orang yang beriman kepada Allah dan hari akhir supaya menyampaikan hal-hal yang baik, apabila tidak bisa melakukan demikian maka hendaklah diam supaya tidak timbul fitnah. Kesimpulannya dalam menyampaikan berita itu harus ber-*tabayyun* terlebih dahulu untuk mengetahui kejelasan dan kebenaran dari berita. Sebab setiap perlakuan akan dipertanggung jawabkan kepada Allah SWT. Janganlah berdiam diri dan jangan ikut menyebarkan suatu berita jika mengetahui bahwa berita itu merupakan berita bohong.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil implementasi dan uji coba sistem maka dapat diambil kesimpulan bahwa berdasarkan 9 kali percobaan *query user* dengan nilai *range min-max* 0 sampai dengan 9 pada *Information Retrieval System* (IRS) dengan 120 dokumen berita diperoleh hasil rata-rata dari *precision* sebesar 83,31%, *recall* 90,56%, *f-measure* 86,40% dan *akurasi* 90,19%.

5.2. Saran

Beberapa saran dari penulis sebagai pertimbangan bagi penelitian selanjutnya adalah:

- a) Perlu diperhatikan dalam penetapan nilai *range* pada normalisasi *min-max* dapat berpengaruh terhadap jumlah iterasi dan hasil *clustering* yang menggunakan metode Self Organizing-Map.
- b) Perlu diperhatikan dalam pembuatan daftar *stoplist* pada proses *stopword removal* dan pemilihan nilai *range* baru pada normalisasi data dengan metode *min-max*, karena dapat berpengaruh terhadap hasil *clustering*.

DAFTAR PUSTAKA

- Agastya, M.A. (2018). *Pengaruh Stemmer Bahasa Indonesia Terhadap Peforma Analisis Sentimen Terjemahan Ulasan Film*. Jurnal TEKNOKOMPAK, Vol. 12, No. 1, 2018, 18-23. ISSN 1412-9663
- Anis, Y., Isnanto, R.R. (2014). *Penerapan Metod Self-Organizing Map (SOM) Untuk Visualisasi Data Geospasial Pada Informasi Sebaran Data Pemilih Tetap (DPT)*. JSINBIS (Jurnal Sistem Informasi Bisnis): Vol 4 No 1, ISSN 2502- 2377.
<https://doi.org/10.21456/vol4iss1pp48-57>
- Azmi, M. (2014). *Komparasi Metode Jaringan Syaraf Tiruan SOM Dan LVQ untuk Mengidentifikasi Data Bunga Iris*. Jurnal TEKNOIF: Vol 2 No 1 (2014), ISSN 2338-2724
- Cahyani, D.A., Khotimah, B.K. & Rizkillah, R.T. (2014). *Perbandingan Metode SOM (Self Organizing Map) dengan Pembobotan Berbasis RBF (Radial Basis Function)*. Jurnal Teknologi Technoscintia: Vol. 7 No. 1, ISSN: 1979– 8415
- Chandra, A. (2017). *Perbedaan Supervised And Unsupervised Learning*. https://www.datascience.or.id/detail_artikel/52/supervised-and-unsupervised-learning, (diakses pada 10 Mei 2018)
- Cholifah, Purwananto, Y. dan Bramantoro, A. (2006). *Aplikasi Information Retrieval untuk Pembentukan Tesaurus Berbahasa Indonesia Secara Otomatis*. Scan Vol 2 No1, ISSN 1978-0087
- Correa, R.F., & Pinheiro, B.F. (2010). *Self-Organizing Maps Applied to Information Retrieval of Dissertations and Theses from BDTD-UFPE*. 2010 Eleventh Brazilian Symposium on Neural Networks (SBRN): <https://doi.org/10.1109/SBRN.2010.14>
- Jalaluddin Al-Mahalli dan Jalaluddin As-Suyuthi. *Tafsir Jalalain*. Surabaya: Nurul Huda
- Lestari, N.P. (2016). *Stemming Bahasa Indonesia*. <https://github.com/sastrawi/sastrawi/wiki/Stemming-Bahasa-Indonesia> diakses pada 21 November 2018
- Librian, A. (2016). *Uji Recall and Precision Sistem Temu Kembali Informasi OPAC Perpustakaan ITS Surabaya*. Vol 5 No 3
- Mas'udia, P.E., Atmadja, M.D. dan Mustafa, L.D. (2017). *Information Retrieval Tugas akhir dan Perhitungan Kemiripan Dokumen Mengacu pada Abstrak*

Menggunakan Vector Space Model. Jurnal SIMETRIS: Vol 8 No 1, ISSN 2252 – 4983

- Muhammad bin Shalih Al'Utsaimin. (2005). *Hadis Arba'in Nawawiyah*. Yogyakarta: Absolut
- Munawar, Ghifari. (2015). *Implementasi Algoritma Self Organizing Map (SOM) Untuk Clustering Mahasiswa Pada Mata Kuliah Proyek (Studi Kasus: JTK POLBAN)*. ISBN 978-979-3541-50-1
- Mustaffa, Z. & Yusof, Y. (2011). *A Comparison of Normalization Techniques in Predicting Dengue Outbreak*. 2010 International Conference on Business and Economics Research. 1. hal. 345-349. Kuala Lumpur: IACSIT Press.
- Nugroho, G.A.P. (2016). *Analisis Sentimen Data Twitter Menggunakan K-Means Clustering*. <https://repository.usd.ac.id/6811/>, (diakses pada 19 Juli 2018)
- Prabowo, D.A, Fhadli, M., Najib, M.A., Fauzi, H.A. dan Cholissodin, I. (2016). *TF-IDF-Enhanced Genetic Algorithm Untuk Extractive Automatic Text Summarization*. Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK): Vol 3 No 3, hlm. 208-215
- Prasetyo, E. (2012). *Data Mining – Konsep dan Aplikasi Menggunakan Matlab*. Yogyakarta: C.V Andi
- Somantri, O. (2017). *Text Mining Untuk Klasifikasi Kategori Cerita Pendek Menggunakan Naive Bayes (NB)*. Jurnal Telematika: Vol 12 No 1, p-ISSN: 1858-2516 e-ISSN: 2579-3772
- Sukma, A., Zaman, B. dan Purwanti, E. (2015). *Klasifikasi Dokumen Temu Kembali Informasi dengan K-Nearest Neighbour*. E-ISSN 2442-5168 Vol 1 No 2
- Suryaningsih, V., Sihwi, S.W, & Sulistyono, M.E. *Clustering Dokumen Menggunakan Algoritma Self Organizing Map (SOM)*. <https://digilib.uns.ac.id/dokumen/download/181498/MTgxNDk4>
- Tan, P. et al. (2006). *Introduction to Data Mining*. Boston: Pearson Education