Text Clustering Pada Opini Warganet Di Twitter Terhadap Kebakaran Hutan Dan Lahan Di Indonesia Menggunakan Pendekatan Semantik, Jarak Dan Densitas

Lomba Penambangan Data

GEMASTIK XII 2019

Nama Tim : Hello-World

ID : 1200200252263

Muhammad Aufa Wibowo **(0511164 000 0184)**

Putri Cinto Buliah M. Eza **(0621164 000 0116)**

Syarifah Widya Ekaputri **(0621164 000 0021)**

INSTITUT TEKNOLOGI SEPULUH NOPEMBER (ITS)

SURABAYA

2019

ABSTRAK

Kebakaran hutan dan lahan terjadi hampir setiap tahun di Indonesia. Ketika memasuki musim kering apalagi musim kering ekstrim (El Nino), kebakaran hutan dan lahan dapat terjadi dibanyak tempat. Tahun 2019 ini Indonesia dilanda bencana kebakaran hutan di Kalimantan Tengah dan Riau yang memiliki indeks pencemaran udara terparah. Berbagai upaya telah dilakukan oleh Pemerintah dalam menanggulangi kebakaran hutan dan lahan. Banyak masyarakat yang resah, turut prihatin, dan berupaya membantu menanggulangi bencana kebakaran hutan dan lahan. Selain itu, banyak dampak buruk yang terjadi akibat kebakaran hutan dan lahan seperti hilangnya habitat hewan, banyaknya hewan yang mati, dan masyarakat terkena infeksi saluran napas akut (ISPA). Opini masyarakat terhadap kebakaran hutan dan lahan ini bermunculan diberbagai media sosial terutama di Twitter. Opini yang disampaikan warganet mengenai bencana yang sedang terjadi ini sangat beragam, seperti keresahan, saran, kritikan, dan informasi lainnya. Sehingga opini warganet tersebut perlu dikelompokkan untuk mengetahui berbagai pokok permasalahan seputar kebakaran hutan dan lahan. Metode statistika yang digunakan untuk analisis data teks dikenal dengan *text mining*. *Text clustering* adalah satu cabang dari ilmu *text mining* yang merupakan sebuah bentuk *unsupervised process* yang mengelompokkan beberapa data kelompok berupa *text* dengan menerapkan algoritma *clustering* yang berbeda-beda. Metode *text clustering* yang digunakan pada penelitian ini yaitu *Latent Dirichlet Allocation* (LDA), *K-Means*, dan *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN). Dari ketiga metode yang digunakan metode *Density-Based Spatial Clustering of Applications with Noise (DBSCAN)* merupakan metode yang terbaik. Dengan nilai *Eps* sebesar 0.75 dan *MinPts* sebesar 300, didapatkan nilai *silhouette coefficient* tertinggi yaitu sebesar 0.999. Sehingga jumlah *cluster* optimal untuk pengelompokan opini warganet di Twitter mengenai kebakaran hutan dan lahan sebanyak 3 *cluster. Cluster* 1, berisi tentang kritikan kepada Pak Jokowi, *cluster* 2 berisi mengenai penanggulangan kebakaran hutan dan lahan. Dan pada *cluster* 3 berisi tentang dampak dari kebakaran hutan dan lahan.

Kata kunci: *Kebakaran Hutan, Opini, Text mining, Text clustering, Twitter, Warganet*

KATA PENGANTAR

Terselesaikannya penelitian dari awal hingga akhir serta pembuatan laporan ini tidak lepas dari bantuan, bimbingan, serta dukungan dari berbagai pihak. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada :

1. Dr. Darmaji, S.Si., M.T selaku Direktur Kemahasiswaan ITS
2. Dr. Suhartono selaku Kepala Departemen Statistika FMKSD ITS.
3. Dr. Eng. Darlis Herumurti S.Kom, M.Kom selaku Kepala Departemen Informatika FTIK ITS.
4. Dr. Santi Wulan Purnami, S.Si., M.Si. selaku Ketua Program Studi Sarjana Departemen Statistika FMKSD ITS.
5. Dr.rer.pol. Dedy Dwi Prastyo, S.Si., M.Si., selaku dosen pembimbing kami yang telah membimbing sampai terselesaikannya karya tulis ini.
6. Orang tua serta keluarga kami atas doa, motivasi, bimbingan, dukungan, serta kesabarannya dalam mendidik kami.
7. Semua pihak yang telah membantu terselesaikannya karya tulis ini, baik secara langsung maupun secara tidak langsung yang tidak dapat disebutkan namanya satu persatu.

Penulis menyadari bahwa laporan ini masih jauh dari kata sempurna. Oleh karena itu, kritik dan saran diharapkan dapat diberikan dari semua pihak untuk tahap pengembangan selanjutnya. Semoga laporan yang penulis susun dapat bermanfaat dan tidak lupa penulis memohon maaf apabila terdapat banyak kekurangan. Atas perhatian dan dukungannya, penulis menyampaikan terima kasih. Semoga Allah SWT selalu menambah berkah dan rahmat- Nya kepada diri kita semua.

Surabaya, 26 September 2019

Penulis

DAFTAR ISI

Halaman

DAFTAR TABEL

Halaman

DAFTAR GAMBAR

Halaman

BAB I
PENDAHULUAN

1.1 Latar Belakang

Kebakaran hutan dan lahan merupakan salah satu bencana yang mendapat perhatian dunia terutama setelah bencana yang disebabkan siklus iklum El Nino ditahun 1998/98 yang menghanguskan 25 juta hektar hutan di seluruh dunia. Kebakaran hutan dan lahan dianggap sebagai ancaman bagi pembangunan berkelanjutan karena efeknya terhadap ekosistem, emisi karbon dan keanekaragaman hayati (Tacconi, 2003).

Tahun 2019 ini Indonesia dilanda bencana kebakaran hutan dan lahan di beberapa daerah Kalimantan dan Riau yang memiliki indeks pencemaran udara terparah. Negara tetangga seperti Malaysia dan Singapura turut menjadi korban dalam bencana kebakaran hutan dan lahan ini. Di Ibu Kota Kalimantan Tengah misalnya, KLHK mencatat angka ISPU Palangkaraya mencapai angka 319 (Berbahaya), pada 20 September 2019 (15.00 WIB). Berarti, konsentrasi PM10 di udara Palangkaraya sudah pada level berbahaya bagi semua populasi yang terpapar. Di Ibu Kota Riau, KLHK mencatat angka ISPU Pekanbaru mencapai angka 419 (Berbahaya), pada 20 September 2019 (15.00 WIB). Artinya, konsentrasi PM10 di udara Pekanbaru berbahaya bagi semua populasi yang terpapar.

Masyarakat sekitar pun terkena dampak dari bencana kebakaran hutan dan lahan ini. Mengutip CNN Indonesia, Minggu (22/9/2019), penderita infeksi saluran napas akut (atau yang selanjutnya disingkat ISPA) di Kota Batam, Kepulauan Riau meningkat 100% pada September 2019. Jumlah ini meningkat dua kali lipat dibandingkan Agustus 2019. Sementara itu, Badan Nasional Penanggulangan Bencana (BNPB) dan Kementrian Kesehatan per 16 Septermber 2019 pukul 16.00 WIB mencatat setidaknya 144.219 warga di Sumatra dan Kalimantan terkenda ISPA. Dengan kondisi bencana seperti itu seringkali masyarakat kehilangan kepercayaan kepada pemerintah dalam penanganan pencegahan kebakaran hutan dan lahan. Ditambah lagi faktor penyebab kebakaran hutan dan lahan tersebut seperti pengalihan fungsi lahan menjadi lahan kelapa sawit oleh oknum yang tidak bertanggung jawab, penegakan hukum yang tidak tegas, dan pencegahan kebakaran hutan dan lahan yang tidak serius.

Seiring dengan perkembangan zaman, penggunaan media sosial terus meningkat. Masyarakat cenderung lebih sering berkomunikasi menggunakan media sosial saat

membutuhkan sesuatu karena lebih mudah diakses kapanpun dan dimanapun. Menurut Selamatta S. selaku Direktur Pelayanan Informasi Internasional Ditjen Informasi dan Komu-nikasi Publik (IKP), situs jejaring sosial yang paling banyak diakses adalah situs jejaring sosial Facebook dan Twitter (Kemenkominfo, 2013). Sehingga media sosial menjadi salah satu wadah dalam penyampaian opini maupun kritik dan saran terhadap suatu permasalahan. Para pengguna internet yang lebih dikenal dengan sebutan warganet bebas memberikan pandangan secara terbuka, saling beropini dalam waktu yang cepat dan tidak terbatas terhadap bencana tersebut di media sosial salah satunya Twitter. Opini dan kritik maupun saran pada *tweet* tersebut dapat digunakan sebagai data. Data yang diperoleh dari Twitter dilakukan dengan cara *crawling* menggunakan API. Kegunaan API

Metode statistika yang digunakan untuk analisis data teks dikenal dengan *text mining*. *Text clustering* adalah satu cabang dari ilmu *text mining* yang merupakan sebuah bentuk *unsupervised process* yang mengelompokkan beberapa data kelompok berupa *text* dengan menerapkan algoritma *clustering* yang berbeda-beda. Tujuan *clustering* adalah mengelompokkan kumpulan data tak berlebel ke dalam *cluster* yang memiliki arti tanpa informasi sebelumnya (Feldman & Sanger, 2007). Metode *text clustering* yang digunakan merupakan metode dengan pendekatan semantik, jarak dan densitas.Metode tersebut adalah metode *Latent Dirichlet Allocation* (LDA), *K-Means,* dan *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN). Dari ketiga metode yang digunakan akan dipilih metode terbaik, Sehingga didapatkan kategori opini yang paling sering diberikan oleh warganet dan diharapkan hasil ini dapat memberikan masukan kepada pemerintah dalam menindak lanjuti masalah bencana kebakaran hutan dan lahan ini.

1.2 Rumusan Masalah

Pandangan masyarakat mengenai bencana kebakaran hutan dan lahan yang sedang marak terjadi di Indonesia khususnya di Provinsi Riau dan beberapa daerah di Pulau Kalimatan merupakan hal yang menarik untuk diteliti. Warganet dapat memberikan pandangan secara terbuka, saling beropini dalam waktu yang cepat dan tidak terbatas terhadap bencana tersebut di media sosial. Pandangan warganet yang berupa opini atau opini tersebut perlu diidentifikasi dan dikelompokkan menjadi beberapa kategori. Kategori opini dapat berupa keluhan, kritik, saran, pemberian informasi, dan lain-lain

sebagainya. Pengidentifikasian informasi dari sebuah *text* memerlukan analisis menggunakan *text mining.* Penentuan kategori pengelompokkan pada *text mning* dapat dilakukan dengan menggunakan metode *text clustering.*

Metode *text clustering* yang digunakan adalah metode *Latent Dirichlet Allocation* (LDA), *K-Means,* dan *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN). Dari ketiga metode yang digunakan akan dipilih metode terbaik, Sehingga didapatkan kategori opini yang paling sering diberikan oleh warganet mengenai masalah tersebut dan diharapkan hasil ini dapat memberikan masukan kepada pemerintah dalam menindak lanjuti masalah bencana kebakaran hutan dan lahan ini.

1.3 Tujuan

Berdasarkan rumusan masalah pada penelitian ini, tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut.

1. Mendeskripsikan karakteristik data *tweet* mengenai bencana kebakaran hutan dan lahan di Indonesia.
2. Mendapatkan kategori opini dari *tweet* mengenai bencana kebakaran hutan dan lahan di Indonesia dengan membandingkan metode *Latent Dirichlet Allocation* (LDA), *K-Means,* dan *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN).
3. Memvisualisasikan *tweet* menggunakan *wordcloud* dari *tweet* mengenai bencana kebakaran hutan dan lahan di Indonesia berdasarkan *cluster* terbaik.

1.4 Manfaat

Berdasarkan permasalahan dan tujuan yang telah dipaparkan, manfaat yang diharapkan dari penelitian ini adalah dapat memberikan kategori dari *tweet* mengenai bencana kebakaran hutan dan lahan di Indonesia sehingga mempermudah memahami opini warganet mengenai bencana alam ini, misalnya opini mengenai keluhan, kritik, saran, pemberian informasi, atau lain-lain yang sering muncul pada *tweet* warganet dengan menggunakan *clustering. S*erta diharapkan hasil ini dapat memberikan masukan kepada pemerintah dalam menindak lanjuti masalah bencana kebakaran hutan dan lahan ini.

1.5 Batasan Masalah

Batasan masalah yang digunakan dalam penelitian ini adalah sebagai berikut.

1. Penelitian menggunakan data *tweet* mengenai bencana kebakaran hutan dan lahan di Indonesia pada periode 15 September 2018 hingga 22 September 2018.
2. Penelitian melakukan analisis terhadap *tweet* berbahasa Indonesia dan bahasa Inggris.
3. Jumlah *clustering* pada *Latent Dirichlet Allocation* (LDA) dan *K-Means* dibatasi 2 hingga 10 *clustering*.
4. Parameter *Eps* pada metode DBSCAN yang digunakan sebesar 0,75 hingga 1.
5. Parameter *MinPts* pada metode DBSCAN yang digunakan sebesar 300.

BAB II
TINJAUAN PUSTAKA

2.1 *Text Mining*

Text mining dapat didefinisikan sebagai proses menggali informasi ketika pengguna berinteraksi dengan beberapa dokumen dari waktu ke waktu dengan menggunakan *tool analysis. Text mining* berguna untuk mengekstrak informasi dari data melalui identifikasi dan eksplorasi dari pola yang menarik. Dalam kasus *text mining*, sumber data yang digunakan adalah kumpulan dokumen dan pola yang ditemukan berada pada data tekstual yang tidak terstruktur (Feldman & Sanger, 2007). *Text mining* berupa proses mengenai struktur penginputan *text*, mencari pola dari data *text* yang telah terstruktur, dan evaluasi final serta interpretasi dari output (Narayana & Kumar, 2015).

Text mining dapat digunakan dalam proses penentuan *rule* baru dengan berbagai alogaritma seperti pengelompokkan, asosiasi, dan *ranking.* Pengelompokan merupakan sebuah fungsi yang sering dilakukan dalam menganalisis *text* dengan menggunakan metode *text clustering* dan *text classification.* Dalam proses penentuan sebuah struktur kelompok yang belum diketahui diketahui (*unsupervised)* dari sekumpulam dokumen dapat menggunakan metode *text clustering*. Sedangkan dalam proses membentuk golongan dari dokumen berdasarkan kelas kelompok yang sudah diketahui sebelumnya (*supervised*) dapat menggunakan metode *text classification* (Durajati & Gumelar, 2012). Langkah-langkah *text mining* yang akan dilakukan dalam penelitian ini adalah *text preprocessing*, *text clustering*, membandingkan metode *clustering* menggunkan nilai *silhouette coefficient*, dan memvisualisasikan menggunakan *word cloud.*

2.2 *Text Preprocessing*

Text preprocessing merupakan tahapan awal yang penting dari NLP (*Natural Language Processing*) karena karakter, kata, dan kalimat yang diidentifikasi pada tahap ini adalah unit dasar yang diteruskan ke semua tahap pemrosesan lebih lanjut. Setiap Teknik *preprocessing* dimulai dengan dokumen yang tidak seluruhnya terstruktur, kemudin diproses untuk melengkapi struktur dengan menyaring fitur-fiur ataupun menambah yang baru. Operasi *text mining* yang efektif didasarkan pada metodologi pemrosesan data yang canggih. Penggunaan *text pre-processing* yang tepat dapat meningkatkan akurasi pada kasus klasifikasi (Feldman & Sanger, 2007).

Text preprocessing dapat dilakukan dengan menggunakan beberapa tahapan. Tahapan-tahapan *text preprocessing* secara umum adalah *cleaning, removing symbols, removing numbers, removing ASCII string, punctuation, tokenization, case folding, stemming,* normalisasi kata, dan *stopword removal*.

1. *Cleaning,* merupakan proses menghilangkan kata yang tidak diperlukan untuk mengurangi *noise*. Kata yang dihilangkan dalam *Twitter* dapat berupa simbol *retweet (*RT), *username (@username), link* URL, dan lain sebagainya (Mujilahwati, 2016).
2. *Removing symbol, number, ASCII strings, and punctuation*, merupakan proses penghapusan simbol, nomor, dan tanda baca lainnya dalam *tweet*. *Tweet* mengandung banyak sekali simbol dan tanda baca. Simbol dan tanda baca yang dihapus adalah seperti "# $ % & \ ' ( ) * + , - . / : ; < = > ? @ [\ \ ] ^ _ ` { | } ~ (Jannah, Fithriasari, Prastyo, & Iriawan, 2018).
3. *Case folding,* merupakan proses untuk mengubah kata ke dalam format yang sama, seperti mengubah semua karakter menjadi huruf kecil atau huruf besar. Proses ini dapat mempercepat proses yang akan dilakukan selanjutnya (Carlantonio & Costa, 2009).
4. *Stemming*, merupakan proses untuk menemukan kata dasar dari sebuah kata (Tala, 2003) Sistem kerja tahap *stemming* ini adalah menghilangkan awalan, akhiran, sisipan, dan *confixes* (kombinasi dari awalan dan akhiran) (Ariadi & Fithriasari, 2015). *Stemming* digunakan agar suatu kata sesuai dengan kaidah Bahasa Indonesia yang benar. Tanpa melakukan *stemming* akan terdapat banyak kata berbeda yang bermakna hampir sama. Sehingga proses ini sangatlah penting untuk dilakukan.
5. Normalisasi kata, merupakan proses mengubah kata-kata yang memiliki arti sama menjadi satu kata seragam (Weiss, Indurkhya, Zhang, & Damerau, 2005) Contohnya adalah kata"trims" dan "makasih" diubah menjadi "terima kasih".
6. *Tokenization*, merupakan proses pemisahan kalimat menjadi kata, frase, symbol, dan elemen lainnya yang memiliki arti atau disebut token dan dapat juga disebut tahap memutuskan kata per kata pada kalimat. Tahapan ini bertujuan untuk memecah yang semula berupa kalimat menjadi potongan-potongan kata, sehingga

urutan *string* akan terputus menjadi potongan-potongan kata penyusunnya (Bing, 2010).

7. *Stopword removal*, merupakan proses menghilangkan kosakata yang bukan termasuk kata unik atau tidk menyampaikan pesan apapun secara signifikan pada *text*. Contohnya adalah kata penghubung dan kata keterangan (Dragut, Fang, Sistla, & Yu, 2009)

2.2.1 *Confix-Stripping Stemmer*

Confix Striping Stemmer atau disebut CS *Stemmer* merupakan pengembangan dari algoritma *Nezief and Adriani's Stemmer.* CS *Stemmer* dikembangkan menyesuaikan pendekatan kaidah Bahasa Indonesia agar meningkatkan hasil *stemming* yang diperoleh.. Berikut merupakan urutan penggunaan afiks, dengan tanda kurung siku berarti bahwa afiks tersebut opsional (Adriani, Asian, Nazief, Tahaghoghi, & Williams, 2007).

[[[DP+]DP+]DP+] kata dasar [[+DS][+PP][+P]]

dengan DP (*Derivational Prefixes*) merupakan awalan, DS (*Deri-vational Suffixes*) merupakan akhiran, PP (*Possessive Pronouns*) merupakan kata ganti kepunyaan, dan P (*Particles*) merupakan partikel.

Langkah-langkah dalam melakukan *stemming* dengan CS *Stemmer* adalah sebagai berikut (Adriani, Asian, Nazief, Tahaghoghi, & Williams, 2007).

1. Pada awal pemrosesan dan pada setiap langkah, dilakukan pemeriksaan kata pada kamus kata dasar. Jika kata tersebut ditemukan, maka dianggap sebagai kata dasar dan seluruh proses dihentikan.
2. Menghilangkan *inflectional suffixes* yang dimulai dari *inflectional particle* ('-kah','-lah','-tah','-pun') dan dilanjutkan menghilangkan *possessive pronoun* ('-ku','-mu','-nya'). Contohnya kata "celanamulah" akan dipotong menjadi "celana-mu" dan kemudian "celana", dimana kata ini sudah merupakan kata dasar sehingga proses berhenti.
3. Menghilangkan *derivational suffixes* ('-i','-kan','-an'). Contohnya kata "membelikan" akan dipotong menjadi "mem-beli", namun karena kata ini bukanlah kata dasar maka proses dilanjutkan ke langkah selanjutnya.
4. Menghilangkan *derivational prefixes* ('be-', 'di-', 'ke-', 'se-', 'me-', 'te-', 'pe-').
   a. Proses berhenti jika:

- Awalan yang teridentifikasi berpasangan dengan akhiran terlarang yang telah dihilangkan pada langkah 3. Berikut merupakn pasangan awalan dan akhiran yang tidak diperbolehkan.

Tabel 2. 1 Kombinasi Awalan dan Akhiran yang Tidak Diperbolehkan

Awalan	Akhiran yang Tidak Diperbolehkan
be-	-i
di-	-an
ke-	-i, -kan
me-	-an
se-	-i, -kan
te-	-an

- Awalan yang dideteksi saat ini sama dengan awalan yang telah dihilangkan sebelumnya.
- Tiga awalan telah dihilangkan.

b. Identifikasi tipe awalan kemudian hilangkan. Awalan terdiri dari dua tipe berikut ini.
   - Standar (’di-’, ’ke-’, ’se-’) dapat dihilangkan langsung dari kata.
   - Kompleks (’be-’, ’te-’, ’me-’, ’pe-’) dapat bermorfologi sesuai kata dasar yang mengikutinya (dapat mengubah bentuk asli kata dasar).

c. Mencari kata yang telah dihilangkan awalannya dalam kamus kata dasar. Apabila pencarian tidak ditemukan maka langkah 4 diulang kembali, sedangkan apabila ditemukan maka keseluruhan proses dihentikan.

5. Apabila hingga langkah 4 kata dasar masih belum ditemukan, maka dilakukan proses *recoding*, yaitu menambah atau mengganti huruf awal dari kata yang terpenggal pada proses *stemming.* Contohnya kata “menyapu” dihilangkan awalan “me” sehingga tersisa “nyapu”. Kata “nyapu” bukanlah kata dasar yang valid sehingga dilakukan *recoding* menjadi kata “sapu”.
6. Apabila semua langkah tidak berhasil, maka *input* kata dianggap sebagai kata dasar dan algoritma akan mengembalikan kata seperti semula.

Terdapat beberapa perbaikan dalam algoritma *Confix-Striping Stemmer* yaitu sebagai berikut (Asian, 2007).

1. Penggunaan kamus kata dasar yang lebih lengkap
2. Melakukan modifikasi dan menambah aturan pemenggalan untuk awalan kompleks.

3. Menambahkan aturan *stemming* untuk kata ulang dan bentuk jamak
4. Melakukan penyesuaian terhadap *rule precedence* dengan mengubah urutan *stemming* pada beberapa kasus tertentu.

2.2.2 *Term Frequency- Inverse Document Frequency* (TF-IDF)

Term Frequency-Inverse Document Frequency (TF-IDF) merupakan pengukuran yang digunakan untuk mengukur seberapa penting suatu kata terhadap kumpulan dokumen. *Term Frequency* (TF) merupakan pendekatan paling sederhana untuk menghitung bobot suatu *term* dimana bobot sama dengan jumlah kemunculan sebuah *term* dalam suatu dokumen. Ketika hanya menggunakan *term frequency* akan terdapat permasalahan yaitu suatu *term* akan memiliki tingkat kepentingan yang sama. *Term Frequency* (TF) berfungsi dalam meringkas kemunculan sebuah kata pada suatu dokumen. Sedangkan *Inverse Document Frequency* (IDF) berfungsi menghitung frekuensi kemunculan sebuah kata pada seluruh kumpulan dokumen (Jannah, Fithriasari, Prastyo, & Iriawan, 2018).

Tahapan dalam melakukan pembobotan dengan metode TF-IDF antara lain sebagai berikut.

1. *Term Weighting*, yaitu suatu proses untuk men*generate* sebuah nilai pada setiap *term* dengan cara menghitung frekuensi kemunculan *term* dalam dokumen (*d*).
2. *Document Frequency*, yaitu suatu proses untuk menghitung banyaknya dokumen yang mengandung *term* ke-*t*.
3. *Inverse Document Frequency*, yaitu suatu proses untuk menghitung nilai *inverse* dari *document frequency*. Rumus yang digunakan adalah sebagai berikut.

$$idf_t = \log_{10} \frac{N}{df_t} \tag{2.1}$$

dimana:

idf_t = *inverse document frequency* pada *term* ke *t*

N = jumlah keseluruhan dokumen

df_t = nilai dari *document frequency* pada *term* ke *t*

4. *Term Frequency-Inverse Document Frequency* (TF-IDF), yaitu proses untuk mendapatkan nilai skor setiap kata terhadap dokumen. Digunakan rumus sebagai berikut.

$$W_{t,d} = Wtf_{t,d} \times idf_t \tag{2.2}$$

dimana:

$W_{t,d}$ = 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*

5. Menghitung nilai skor akhir setiap dokumen dengan menggunakan rumus:

$$Ws_j = \sum_{i=1}^{Nterm} Wtd_{i,j} \quad (2.3)$$

dimana:

Ws_j = skor dari dokumen ke *j*

Nterm = jumlah banyaknya *term*

$Wtd_{,j}$ = nilai TF-IDF pada *term* ke *i*, dokumen ke *j*

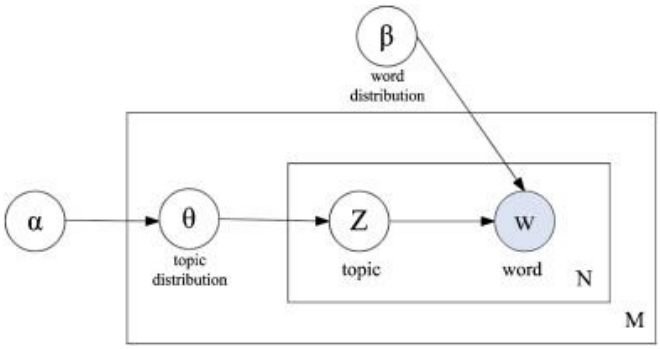
2.3 *Text Clustering*

Text clustering merupakan sebuah bentuk *unsupervised process* yang mengelompokkan beberapa data kelompok berupa *text* dengan menerapkan algoritma *clustering* yang berbeda-beda. Tujuan *clustering* adalah mengelompokkan kumpulan data tak berlebel ke dalam *cluster* yang memiliki arti tanpa informasi sebelumnya (Feldman & Sanger, 2007). Metode *clustering* yang umum digunakan dapat dibagi menjadi dua jenis, yaitu *Hierarchical Clustering* dan *Partitioned Clustering*. Salah satu contoh metode *Hierarchical Clustering* adalah *Single Linkage Clustering* sedangkan metode *Partitioned Clustering* adalah *K-Means* (Alfina, Santosa, & Barakbah, 2012). Metode *text clustering* yang digunakan dalam penelitian ini adalah metode *Latent Dirichlet Allocation* (LDA), *K-Means*, dan *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN).

2.3.1 *Latent Dirichlet Allocation* (LDA)

Latent Dirichlet Allocation (LDA) merupakan metode *topic modeling* yang dipilih dalam melakukan analisis pada dokumen yang berukuran sangat besar. LDA dapat digunakan untuk meringkas, melakukan *clustering*, menghubungkan maupun memproses data yang sangat besar karena LDA menghasilkan daftar topik yang diberi bobot untuk masing-masing dokumen (Campbell, Hindle, & Stroulia, 2014). Distribusi yang digunakan untuk mendapatkan distribusi topik per dokumen disebut distribusi *Dirichlet*, kemudian dalam proses generatif untuk LDA, hasil dari *Dirichlet* digunakan untuk mengalokasikan kata-kata pada dokumen untuk topik yang berbeda. Dalam LDA, dokumen-dokumen merupakan objek yang dapat diamati, sedangkan topik, distribusi

topik per dokumen, penggolongan setiap kata pada topik per dokumen merupakan struktur tersembunyi, maka dari itu, algoritma ini dinamakan *Latent Dirichlet Allocation* (LDA) (Blei, 2012). *Latent Dirichlet Allocation* (LDA) merupakan model probabilistik generatif dari kumpulan tulisan yang disebut *corpus*. Ide dasar yang diusulkan metode LDA adalah setiap dokumen direpresentasikan sebagai campuran acak atas topik yang tersembunyi, yang mana setiap topik memiliki karakter yang ditentukan berdasarkan distribusi kata-kata yang terdapat di dalamnya. LDA sebagai model probabilistik dapat direpresentasikan secara visual seperti berikut ini.



Gambar 2. 1 Visualisasi Metode LDA sebagai Model Probabilistik

Visualisasi model probabilistic LDA di atas menjelaskan bahwa terdapat tiga tingkatan pada LDA Modeling. Parameter α dan β merupakan parameter distribusi topik yang berada pada tingkatan *corpus*, yaitu kumpulan dari *M* dokumen. Parameter α digunakan dalam menentukan distribusi topik dalam dokumen, semakin besar nilai α dalam suatu dokumen, menandakan campuran topik yang dibahas dalam dokumen semakin banyak. Parameter β digunakan untuk menentukan distribusi kata dalam topik. Semakin tinggi nilai β, maka semakin banyak kata-kata yang ada di dalam topik, sedangkan semakin kecil nilai β, maka semakin sedikit kata-kata yang ada di dalam topik sehingga topik tersebut mengandung kata-kata yang lebih spesifik. Variabel θ_M adalah variabel yang berada di tingkat dokumen *(M)*. Variabel θ merepresentasikan distribusi topik untuk dokumen tertentu. Semakin tinggi nilai θ, maka semakin banyak topik yang ada di dalam dokumen, sedangkan semakin kecil nilai θ, maka dapat dikatakan dokumen tersebut se-makin spesifik pada topik tertentu. Variabel Z_N dan W_N adalah variabel tingkat kata *(N)*. Variabel *Z* dan merepresentasikan topik dari kata tertentu pada sebuah dokumen sedangkan variabel *W* merepresentasikan kata yang berkaitan dengan topik tertentu yang terdapat dalam dokumen (Blei, 2012).

Pengolahan kata-kata dengan menggunakan metode LDA dilakukan dalam dua tahapan. Tahapan pertama adalah probabilitas kata dalam topik dan tahapan kedua probabilitas topik dalam dokumen. Probabilitas kata dalam dokumen dapat dihitung dengan menggunakan persamaan sebagai berikut:

$$p(w_i \mid m) = \sum_{j=1}^{K} p(w_i \mid z_j)\, p(z_j \mid m) \tag{2.4}$$

LDA mengasumsikan proses generatif berikut untuk *corpus M*:

1. Untuk setiap topik $k \in \{1, 2, \ldots, K\}$, sampel distribusi kata $w_k \sim Dirichlet\ (\beta)$
2. Untuk setiap dokumen $m \in \{1, 2, \ldots, M\}$, sampel distribusi topik $\theta_d \sim Dirichlet\ (\alpha)$

Secara umum, LDA bekerja dengan masukan dokumen-dokumen individual dan beberapa parameter, untuk menghasilkan luaran berupa model yang terdiri dari bobot yang dapat dinormalisasi sesuai probabilitas. Probabilitas ini mengacu pada dua jenis, yaitu jenis pertama adalah probabilitas bahwa suatu dokumen spesifik tertentu menghasilkan topik yang spesifik pula dan jenis kedua adalah probabilitas bahwa topik spesifik tertentu menghasilkan kata-kata spesifik dari sebuah kumpulan kosakata. Probabilitas jenis pertama dokumen yang sudah diberi label dengan daftar topik seringkali dilanjutkan hingga menghasilkan probabilitas jenis kedua yang menghasilkan kata-kata spesifik tertentu (Campbell, Hindle, & Stroulia, 2014).

2.3.2 *K-Means*

K-Means merupakan salah satu metode analisis *cluster* yang berbasis jarak atau disebut *distance-based cluster analysis* (Han, Kamber, & Pei, 2012). *K-Means* digunakan ketika semua variabel bersifat kuantitatif dan jarak kuadrat euclidean digunakan untuk mengukur ketidaksamaan antar objek (Johnson & Wichern, 2007). *K-Means* termasuk dalam metode analisis *clustering nonhierarki* yang dapat digunakan dalam mempartisi data kedalam beberapa kelompok berdasarkan kedekatan karakteristik. Langkah-langkah atau algoritma dari metode *K-Mean* adalah sebagai berikut (Tan, Steinbach, & Kumar, 2006).

1. Menentukan *K centroid* awal atau titik pusat *cluster* awal secara *random.*
2. Membentuk *K cluster* dengan menempatkan *tweet* pada *centroid* terdekat. Perhitungan jarak *centroid* terdekat menggunakan jarak *Euclidean* sebagai berikut

$$D_{ij} = \sqrt{\sum_{k=1}^{p} (x_{kj} - y_{ki})^2} \tag{2.5}$$

dimana:

D_{ij} = jarak *Euclidean* dari *tweet* ke-*j* ke pusat *cluster ke-i*

x_{kj} = frekuensi kemunculan kata ke-*k* pada *tweet* ke-*j*

y_{ki} = frekuensi kemunculan kata ke-*k* pada pusat *cluster* ke-*i*

p = total banyak kata

3. Menghitung kembali nilai *centroid* seriap *cluster* menggunakan rumus sebagai berikut.

$$v_{ik} = \frac{\sum_{j=1}^{n_i} x_{kj}}{n_i} \quad (2.6)$$

dengan:

v_{ij} = rata-rata *cluster* ke-*i* untuk kata ke-*k*

n_i = jumlah *tweet* yang menjadi anggota *cluster* ke-*i*

i = indeks dari *cluster*

j = indeks dari kata

x_{kj} = frekuensi kemunculan kata ke-*k* pada *tweet* ke-*j* yang berada dalam *cluster* tersebut.

4. Melakukan kembali langah ke-2 dan ke-3 hingga didapatkan *centroid* tidak berubah, yang artinya tidak terdapat anggota *cluster* yang berubah.

2.3.3 *Density-Based Spatial Clustering of Applications with Noise (DBSCAN)*

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) merupakan salah satu metode *density-based clustering. Density-based clustering* adalah salah satu jedis metode *clustering* yang digunakan untuk memperoleh atau menghasilkan pola yang sebelumnya tidak diketahui dari sekumpulan data (Bhuyan & Borah, 2013). DBSCAN merupakan pengelompokkan yang didasarkan pada kepadatan (*density)* data. Konsep kepadatan yang dimaksud adalah jumlah daya yang berada dalam radius *MinPts* (jumlah minimal data dalam radius *Eps*), data tersebut masuk dalam kategori kepadatan yang diinginkan, jumlah data dalam radius tersebut termasuk data inti itu sendiri. Konsep kepadatan seperti ini melahirkan tiga status data yaitu inti *(core),* batas *(border),* dan *noise.* Sebuah data akan dikatakan data inti jika jumlah data tetangga dan dirinya sendiri pada radius *Eps* ≥ *MinPts.* Nilai radius *Eps* dan *MinPts* ditentukan secara mandiri. Data yang jumlah tetangga dan dirinya sendiri dalam radius *Eps* < *MinPts* tetapi tetangganya menjadi inti kerena kehadirannya, data tersebut dikategorikan

sebagai batas. Sedangkan data yang jumlah tetangga dan dirinya sendiri dalam radius *Eps* < *MinPts* tetapi tidak ada tetangga yang menjadi inti maka data tersebut dikategorikan menjadi data *noise* (Tan, Steinbach, & Kumar, 2006).

Algoritma DBSCAN membutuhkan dua parameter penting yaitu parameter radius (*Eps)* dan jumlah minimum poin untuk membentuk kelompok (*MinPts).* Berikut merupakan algoritma dari *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN) (Han, Kamber, & Pei, 2012).

1. Menentukan parameter *MinPts* dan *Eps*. Proses penentuan parameter *MinPts* dan *Eps* berdasarkan hasil kombinasi *Eps* dengan *MinPts* yang memiliki nilai s*ilhouette coefficient* tertinggi.
2. Pilih *tweet p* secara *random* untuk diidentifikasi.
3. Menghitung jumlah *tweet* yang ditentukan oleh parameter radius (*Eps*). Jika jumlahnya mencukupi (lebih dari atau sama dengan *Eps),* data akan ditandai sebagai inti (*core point*).
4. Menghitun jarak *tweet* yang ditandai sebagai *core point* dengan *point* yang lain menggunakan jarak *Euclidean.* Berikut merupakan rumus jarak *Euclidean* yang digunakan.

$$d_{ip} = \sqrt{\sum_{j=1}^{m} (x_{ji} - y_{jp})^2} \qquad (2.7)$$

dimana:

d_{ip} = jarak *Euclidean* dari *tweet* ke-*i* ke pusat *cluster ke-k*

x_{ji} = frekuensi kemunculan kata ke-*j* pada *tweet* ke-*i*

y_{jp} = frekuensi kemunculan kata ke-*j* pada pusat *cluster* ke-*p*

m = total banyak kata

5. Buat *cluster* baru dengan menambahkan *tweet p* ke dalam *cluster.*
6. Melakukan identifikasi pada data yang ditandai sebagai *core point.*
7. Lanjutkan proses sampai semua *point* telah diproses.
8. Jika ada *tweet* yang tidak masuk ke dalam *cluster* maupun akan ditandai sebagai *noise.*

2.4 *Silhouette Coefficient*

Setelah didapatkan hasil *clustering,* perlu adanya evaluasi mengenai kebaikan dari hasil *clustering* tersebut. Salah satu cara untuk mengevaluasi hasil *clustering* yaitu

menggunakan s*ilhouette coefficient. Silhouette coefficient* dilakukan untuk mengukur persamaan yang terjadi pada suatu objek dengan membandingkan antara *cluster* dengan *cluster* lainnya. Hal yang perlu diketahui ketika menentukan nilai *silhouette* adalah hasil dari partisi atau *clustering result* dan pengelompokan semua kedekatan antar objek (Jannah, Fithriasari, Prastyo, & Iriawan, 2018). *Silhouette coefficient* merupakan metode yang digunakan untuk mengevaluasi *clustering* dengan memeriksa seberapa baik *cluster* dikelompokan dan seberapa kompak suatu *cluster*. Langkah-langkah yang dilakukan dalam menghitung nilai *silhouette* adalah sebagai berikut (Han, Kamber, & Pei, 2012).

1. Menghitung rata-rata jarak *a(i)*, yaitu rata-rata jarak dari *tweet i* dengan semua *tweet* yang berada dalam satu *cluster*.

$$a(i) = \frac{1}{|A|-1} \sum_{j=1}^{i} j \in A_{j \neq i} d(i, j) \tag{2.8}$$

dimana:

j = dokumen lain dalam satu *cluster A*

d(i,j) = jarak dari *tweet i* dengan *tweet j*

2. Menghitung rata-rata jarak *b(i)* yaitu rata-rata jarak dari *tweet i* dengan semua *tweet* yang berada di *cluster* lain kemudian diambil nilai terkecil-nya.

$$d(i,C) = \frac{1}{|A|-1} \sum_{j=1}^{i} j \in C, d(i, j) \tag{2.9}$$

dimana:

d(i,C) = jarak rata-rata dokumen *i* dengan semua objek pada *cluster* lain C dimana $A \neq C$

$$b(i) = \min C \neq A, d(i,C) \tag{2.10}$$

dimana:

b(i) = nilai terkecil dari jarak rata-rata dokumen *i* dengan semua objek pada *cluster* lain *C* dimana $A \neq C$

3. Mendapatkan nilai *silhouette coefficient s(i)* sesuai dengan persamaan sebagai berikut.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{2.11}$$

Nilai *silhouette coefficient s(i)* berada diantara nilai -1 dan 1. Nilai negative menandakan bahwa nilai *b(i)* lebih kecil dibandinkan *a(i),* dan pengelompokkan yang

lebih mendekati pada *cluster* yang sama. Sedangkan apabila nilai *s(i)* medekati nilai 1 berarti pengelompokkan jauh dari *cluster* lain (Kogan, 2007).

2.5 *World Cloud*

Wordcloud merupakan salah satu bentuk visualisasi dokumen yang paling sering digunakan. Secara teknis, *wordcloud* memvisualisasikan kata yang memiliki frekuensi kemunculan paling banyak dalam sebuah dokumen. Dengan menggunakan *word cloud*, gambaran frekuensi kata-kata dapat ditampilkan dalam bentuk yang menarik namun tetap informatif. Kata tersebut ditunjukkan melalui ukuran huruf maupun warna yang menarik perhatian pembaca saat pertama kali melihat bentuk *word cloud* yang disajikan. Semakin besar ukuran kata dan ketebalan kata yang ditampilkan, maka semakin sering pula kata tersebut muncul di dalam dokumen (Castella, Quim, Sutton, & Charles, 2014). Orientasi dominan dari kata-kata di *word cloud* dapat diatur dalam berbagai cara, mulai dari sepenuhnya vertikal, sepenuhnya horizontal ataupun ada yang vertikal dan horizontal (Berry & Kogan, 2010). Berikut merupakan salah satu contoh visualisasi dokumen *text* dengan menggunakan *word cloud.*

Gambar 2. 2 Visualisasi Data dengan Wordcloud

2.6 Twitter

Twitter merupakan salah satu media sosial yang sangat sering digunakan warganet. Twitter berbentuk *microblog,* yaitu *blog* yang dapat memberikan fasilitas bagi pengguna untuk menulis pesan, namun Twitter membatasi 280 karakter untuk setiap pesan atau *postingan.* Pada Maret 2006 pertama kali Twitter dibuat dan digunakan. Pengguna Twitter yang terdaftar hingga kini yaitu lebih dari 14 juta pengguna dengan memiliki *username (@username)* yang khusus dan tidak diperbolehkan sama dengan pengguna lainnya. Twitter memiliki beberapa bahasa khas seperti *follow, following,*

tweet, retweet, dan lain sebagainya. Selain itu, ada pula *hastag (#)* yang menandakan konteks dari sebuah pesan Twitter (Zarrella, 2010).

Gambar 2. 3 Logo Twitter

Aplikasi *microblog* memiliki sebuah aplikasi pemograman antarmuka (*Application Programming Interface/API)* agar developer dapat mengembangkan aplikasi tersebut sesuai dengan keinginan dan kebutuhan masing-masing. Data yang diperoleh dari Twitter dilakukan dengan cara *crawling* menggunakan API. Kegunaan API adalah untuk mengetahui informasi tentang pengguna, jaringan pengguna yang terdiri dari koneksi dan *tweet* yang dibicarakan. Twitter API hanya dapat diakses melalui permintaan otentik (Kumar, Morstatter, & Liu, 2014).

2.7 Kebakaran Hutan dan Lahan

Kebakaran hutan dan lahan merupakan kebakaran yang tidak terkendali di area vegetasi yang mudah terbakar. Banyak organisasi menganggap kebakaran hutan sebagai kebakaran yang tidak direncanakan dan tidak diinginkan (Cambridge University, 2008). Penyebaran kebakaran hutan dan lahan bervariasi bergantung pada material mudah terbakar, kemiringan tanah dan kadar airnya, dan kondisi cuaca. Pengaturan dan kepadatan bahan bakar sebagian diatur oleh topografi, karena bentuk lahan menentukan faktor-faktor seperti ketersediaan sinar matahari dan air untuk pertumbuhan tanaman.

Kebakaran hutan dan lahat menghasilkan asap api terutama terdiri dari karbon dioksida dan uap air. Komponen asap umum lainnya hadir dalam konsentrasi yang lebih rendah adalah karbon monoksida, formaldehida, akrolein, hidrokarbon polyaromatik, dan benzena. Partikel kecil tersuspensi di udara yang berbentuk padat atau tetesan cair juga hadir dalam asap. 80 -90% asap api, secara massal, berada dalam kelas ukuran partikel halus berdiameter 2,5 mikrometer atau lebih kecil (National Wildlife Coordination, 2001).

Tingkat paparan asap api kepada seseorang tergantung pada panjang, tingkat keparahan, durasi, dan kedekatan api. Orang-orang terpapar asap langsung melalui saluran pernapasan meskipun menghirup polutan udara. Secara tidak langsung, masyarakat terpapar puing-puing kebakaran yang dapat mencemari persediaan tanah dan air. Badan Perlindungan Lingkungan AS (EPA) mengembangkan indeks kualitas udara (AQI), sumber daya publik yang menyediakan konsentrasi standar kualitas udara nasional untuk polutan udara umum. Masyarakat dapat menggunakan indeks ini sebagai alat untuk menentukan keterpaparan mereka terhadap polutan udara berbahaya berdasarkan rentang jarak pandang. (U.S. Environmental Protection, 2009)

BAB III
METODOLOGI PENELITIAN

3.1 Sumber Data

Data yang digunakan pada penelitian ini merupakan kumpulan *tweet* dengan kata kunci "kebakaran hutan", "Riau", dan "Kalimantan" yang diakses pada tanggal 15 September 2018 hingga 22 September 2018. Data diperoleh dengan *Twitter API* (*Application Programming Interface).*

3.2 Variabel Penelitian dan Struktur Data

Data yang diperoleh berjumlah 10.005 *tweet*. Variabel penelitian yang digunakan merupakan kata-kata yang diperoleh dari keseluruhan *tweet* yaitu frekuensi kata *j* yang muncul pada *tweet* ke-*i* pada Tabel 3.1. Struktur data yang digunakan pada penelitian ini setelah dilakukannya text preprocessing yaitu sebagai berikut.

Tabel 3. 1 Struktur Data

Tweet	**Kata 1**	**Kata 2**	**…**	**Kata ke-*m***
Tweet 1	x_{11}	x_{12}		x_{1m}
Tweet 2	x_{21}	x_{22}		x_{2m}
Tweet 3	x_{31}	x_{32}		x_{3m}
…	…	…		…
Tweet ke n	x_{n1}	x_{n2}		x_{nm}

Keterangan:

x_{ij} = Jumlah kata ke-*j* pada tweet ke-*i*, *i* = 1, 2, 3,…, *n*

n = Banyaknya *tweet*

m = Banyaknya kata

3.3 Langkah Penelitian

Dalam melakukan praktikum ini, berikut adalah langkah-langkah penelitian yang dilakukan:

1. Mengambil data *tweet* dengan *twitter* API

   Pertama memasukkan *keyword* "kebakaran hutan", "Riau" dan "Kalimantan" dengan mengatur lokasi di Indonesia. Kemudian menyimpan hasil pencarian tersebut ke suatu *database.*

2. Melakukan *text preprocessing* dengan langkah-langkah sebagai berikut.

a. Melakukan *cleaning* data, yaitu menghapus link URL, symbol *retweet* (RT), *username* (@u*sername*), baris baru, angka, hastag (#_, emoticon, tanda baca, dan spasi berlebih (*double space*).
b. Melakukan *case folding*, yaitu mengubah seluruh teks menjadi huruf kecil (nonkapital).
c. Melakukan *stemming*, yaitu menghilangkan kata imbuhan sehingga didapatkan kata dasar. *Stemming* dilakukan dengan *package* Sastrawi di *Python*.
d. Melakukan normalisasi kata, yaitu mengubah kata-lata dengan bermakna yang sama menjadi satu kata yang seragam.
e. Melakukan *tokenizing* untuk memecah *tweet* menjadi kata per kata.
f. Menghapus kata pada *tweet* yang terdapat pada daftar *stopwords.* Daftar *stopwords* yang digunakan merupakan *stopwords* Bahasa Indonesia dan *stopwords* Bahasa Inggris.

3. Mengubah data *tweet* ke dalam bentuk frekuensi kemunculan kata pada setiap *tweet* serta pembobotan menggunakan TF-IDF.
4. Melakukan visualisasi data seluruh *tweet* secara *unigram* dan *bigram.*
5. Melakukan visualisasi data penyebaran *tweet.*
6. Melakukan analisis *text clustering* dengan langkah-langkah sebagai berikut.
   a. Melakukan *clustering* dengan metode *Latent Dirichlet Allocation* (LDA). Pada penelitian ini jumlah *K* yang digunakan sebanyak 2 hingga 10.
      - Melakukan *clustering* sesuai dengan langkah-langkah pada subsubbab 2.3.1
      - Menghitung nilai *silhouette coefficient* dari setiap jumlah *K* dan memilih jumlah *K* yang memiliki *silhouette coefficient* tertinggi.
   b. Melakukan *clustering* dengan metode *K-means*. Pada penelitian ini kombinasi *K* yang digunakan 2 hingga 10.
      - Melakukan *cluster*ing sesuai dengan langkah-langkah pada subsubbab 2.3.2
      - Menghitung nilai *silhouette coefficient* dari setiap kombinasi *K* dan memilih parameter *K* yang memiliki *silhouette* tertinggi.
   c. Melakukan *clustering* dengan metode DBSCAN. Pada penelitian ini digunakan parameter *Eps* 0,75 hingga 1 dan *MinPts* sebesar 300.
      - Melakukan *clustering* sesuai langkah-langkah pada subbab 2.3.3

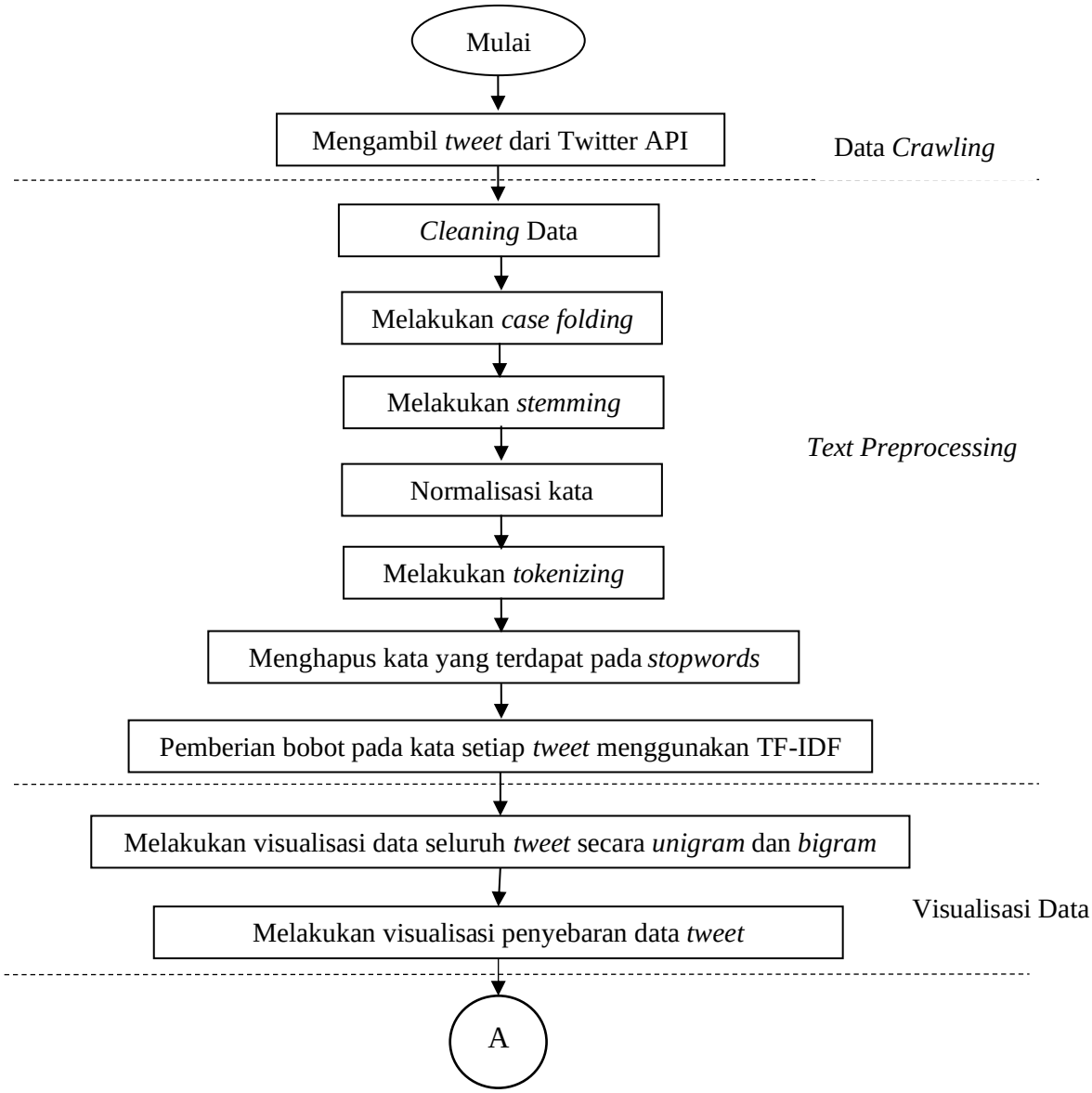
- Menghitung nilai *silhouette coefficient* setiap kombinasi *MinPts* berdasarkan persamaan *MinPts* yang memiliki nilai *silhouette coefficient* tertinggi.

d. Memilih hasil *clustering* terbaik dengan melihat nilai *silhouette coefficient* terbesar berdasarkan persamaan 2.11.

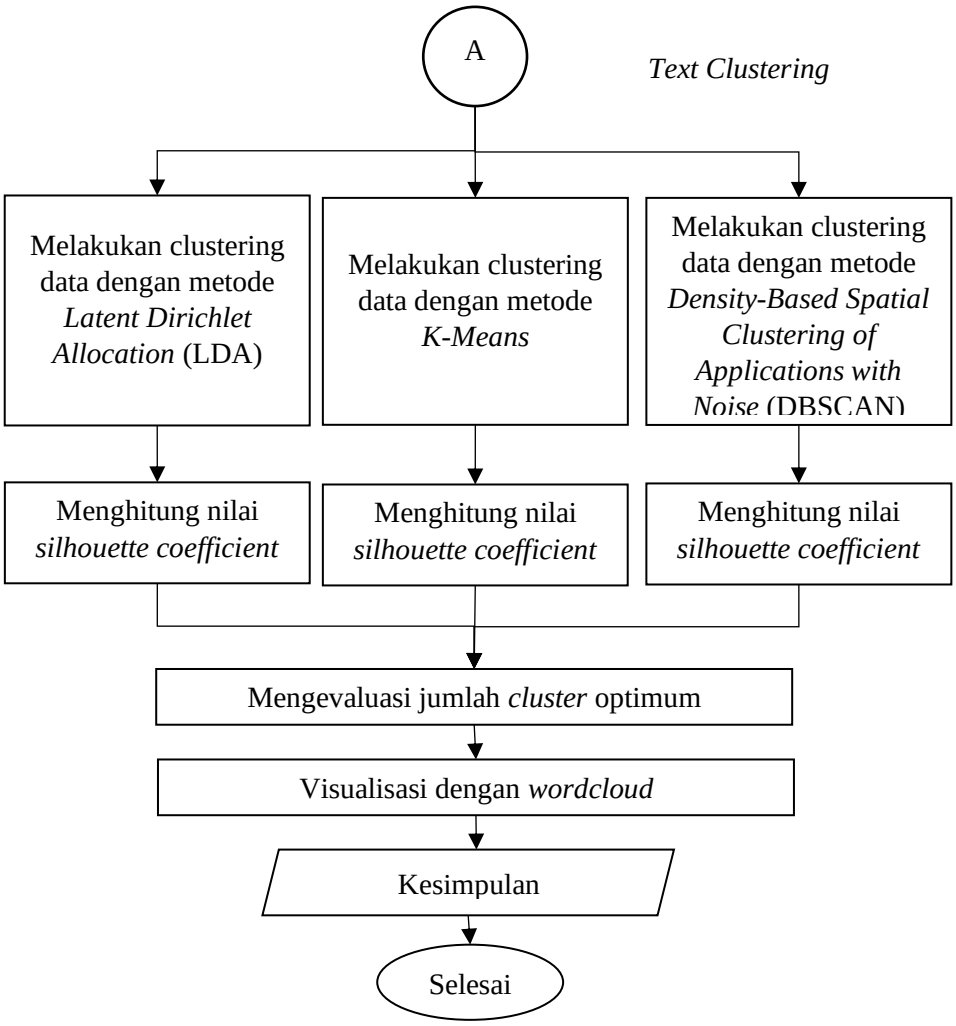
7. Menampilkan visualisasi data dengan *wordcloud* pada *text clustering* terbaik.
8. Melakukan interpretasi dan menarik kesimpulan.

3.4 Diagram Alir

Berikut adalah gambaran dari langkah penelitian dalam bentuk diagram alir.



Gambar 3. 1 Diagram Alir Penelitian



Gambar 3. 2 Diagram Alir Penelitian (Lanjutan)

3.5 Pihak yang Dapat Mengimplementasikan Gagasan

1. Peneliti

Melakukan penelitian-penelitian yang dapat membantu pemerintah dalam mencari solusi dan menanggulangi permasalahan yang ada di Indonesia.

2. Pemerintah Indonesia

Pemerintah berperan sebagai pihak yang menetapkan kebijakan terhadap solusi dan inovasi dalam mengatasi permasalahan yang terjadi di Indonesia. Dengan mengklasterkan opini warganet mengenai permasalahan yang ada di Indonesia, seperti saran, masukkan, kritik, dan opini warganet lainnya. Sehingga gagasan ini dapat digunakan Dewan Perwakilan Rakyat untuk mengetahui keinginan warganet yang kemudian dapat membantu Presiden dalam menyelesaikan permasalahan yang terjadi.

BAB IV
ANALISIS DAN PEMBAHASAN

4.1 *Text Preprocessing*

Langkah pertama sebelum melakukan analisis *clustering* terhadap *tweet* yaitu melakukan *preprocessing* terlebih dahulu. Praproses teks yang dilakukan adalah menghapus simbol, *case folding*, menghapus *stopwords*, *stemming*, dan *tokenizing*.

Tabel 4. 1 Tahap *Text Preprocessing*

Preprocessing	**Kalimat**	**Keterangan**
Kalimat awal	Akhirnya kami yg di aceh ikut merasakan asap😌 seperti saudara2 diriaujambi dan kalimantan segini aja udh perih https://t.co/kwMkSj2dKK	-
Menghilangkan *link* tanda *retweet, usernmae, hashtag, mention*,*emoticon* dan angka	Akhirnya kami yg di aceh ikut merasakan asap seperti saudara d riau jambi dan kalimantan segini aja udh perih	Menghilangkan link https://t.co/kwMkSj2dKK, emoticon "😌", dan angka "2"
Menghilangkan *punctuation*	Akhirnya kami yg di aceh ikut merasakan asap seperti saudara di riau jambi dan kalimantan segini aja udh perih	Menghilangkan tanda baca koma (,)
Lowercase	akhirnya kami yg di aceh ikut merasakan asap seperti saudara diriau jambi dan kalimantan segini aja udh perih	Mengubah kalimat ke dalam huruf kecil
Persamaan kata	akhirnya kami yang di aceh ikut merasakan asap seperti saudara diriau jambi dan kalimantan segini saja sudah perih	Mengganti "yg" menjadi "yang", "udh" menjadi "sudah", dan "aja" menjadi "saja"
Stemming	akhir kami yang di aceh ikut rasa asap seperti saudara di riau jambi dan kalimantan segini saja sudah perih	Mencari kata dasar
Stopword	aceh asap saudara riau jambi kaliman tan perih	Menghilangkan kata "padahal", "di", "ke", "nya", "minggu", "lalu", "tapi", "gimana", "ya"

Pada Tabel 4.1 menunjukkan preprocessing terhadap salah satu *tweet* yang masil memuat URL, emoticon, angka, tanda baca, huruf kapital, kalimat tidak baku, dan

imbuhan. Setelah melakukan seluruh tahap *text preprocessing*, maka didapatkan potongan kata yang dapat digunakan untuk analisis lebih lanjut.

Setelah dilakukan *text preprocessing* yaitu mengubah teks menjadi numerik dengan menghitung *Tem frequency* (TF). Perhitungan TF dilakukan dengan menghitung frekuensi kemunculan kata dalam *tweet*. Selanjutnya dilakukan perhitungan *Document Frequency* (DF). DF merupakan frekuensi kemunculan suatu kata dalam keseluruhan *tweet*. Sebelumnya, perlu dibuat *document term matrix* untuk menentukan jumlah suatu kata yang muncul pada tiap *tweet.*

Tabel 4. 2 *Document Term Matrix*

Tweet	Asap	Hewan	Kabut	…	Semai
1	1	0	0	…	1
2	0	1	0	…	0
3	1	1	0	…	1
4	2	0	0	…	1
⋮	⋮	⋮	⋮	⋱	⋮
10.005	0	0	1	…	1

Tabel 4.2 menunjukkan frekuensi suatu kata muncul pada tiap *tweet.* Contohnya yaitu kata asap muncul pada *tweet* ke-1, *tweet* ke-2, *tweet* ke-4, dan seterusnya. Setelah dibentuk *document term matrix*, maka dapat dilakukan perhitungan DF dan IDF.

Tabel 4. 3 Perhitungan DF dan IDF

Kata	Tweet						DF	IDF	
	T1	T2	T3	T4	…	T10.005			
Asap	1	0	1	2	…	0	5217	$\log\left(\frac{10005}{5217}\right)$=	0,2828
Hewan	0	1	1	0	…	0	3137	$\log\left(\frac{10005}{3137}\right)$=	0,5037
Kabut	0	0	0	0	…	1	2685	$\log\left(\frac{10005}{2685}\right)$=	0,5712
⋮	⋮	⋮	⋮	⋮	⋱	⋮	⋮	⋮	
Semai	1	0	1	1	…	1	3786	$\log\left(\frac{10005}{3786}\right)$=	0,4220

Berdasarkan Tabel 4.3, maka diperoleh nilai DF dan IDF pada tiap kata pada seluruh tweet. Kata mahasiswa memiliki nilai DF sebesar 1 dan nilai IDF sebesar 0.6989, kata tebal memiliki nilai DF sebesar 4 dan nilai IDF sebesar 0.0969. Sehingga didapatkan bahwa semakin besar nilai DF maka semakin kecil nilai IDF. Selanjutnya dilakukan perhitungan TFIDF untuk pembobotan tiap kata.

Tabel 4. 4 Perhitungan TFIDF

Tweet	Kata				
	Asap	**Hewan**	**Kabut**	⋯	**Semai**
1	0,2828	0	0	…	0,4220
2	0	0,5037	0	…	0
3	0,2828	0,5037	0	…	0,4220
4	0,5656	0	0	…	0,4220
⋮	⋮	⋮	⋮	⋱	⋮
10.005	0	0	0,5712	⋯	0,4220

Pada analisis *clustering*, nilai TF-IDF akan digunakan sebagai variabel prediktor. Tabel 4.4 menunjukkan hasil perhitungan TF-IDF pada setiap kata untuk tiap *tweet*. Nilai TF-IDF untuk kata "asap" pada *tweet* keempat yaitu sebesar 0,5656 diperoleh dari nilai TF yaitu 2 dan nilai IDF sebesar 0,2828. Perhitungan nilai TF-IDF untuk kata lainnya juga dilakukan dengan cara demikian.

4.2 Visualisasi Data

Setelah data telah dilakukan *preprocessing*, selanjutnya dilakukan visualisasi data untuk mengetahui karakteristik dari *tweet* mengenai permasalahan kebakaran hutan dan lahan.

Gambar 4. 1 Visualisasi *Word Cloud Unigram*

Pada Gambar 4.1 diperoleh kata yang sering muncul yaitu kata "bunuh", "hewan", "manusia", "marah". Hal ini merujuk pada permasalahan banyak hewan yang mati dan hilang habitatnya akibat kebakaran hutan dan lahan. Sedangkan untuk kata "tolong" merujuk pada permintaan warganet kepada pemerintah untuk segera menanggulangi

permasalahan kebakaran hutan dan lahan ini. Dan pada kata “asap” mengindikasi bahwa asap akibat kebakaran hutan dan lahan telah menyebar dan mengenai beberapa daerah seperti Jambi, Aceh, dan beberapa daerah di Sumatera.

Gambar 4. 2 Visualisasi *Word Cloud Bigram*

Berdasarkan Gambar 4.2 diperoleh *tweet* “hewan manusia”, “bunuh hewan”, dan “marah bunuh” memiliki bobot paling besar. Hal ini karena kata “hewan”, “manusia”, “bunuh”, dan “marah” paling banyak muncul pada permasalahan kebakaran hutan dan lahan. Selain itu kata yang sering muncul yaitu “kabut asap”. Hal ini mengindikasi bahwa warganet sangat mempermasalahkan terkait kabut asap akibat kebakaran hutan dan lahan dan dampak dari kebakaran hutan dan lahan yaitu banyak hewan yang terbunuh.

Gambar 4. 3 Penyebaran Dari *Tweet* Mengenai Kebakaran Hutan Dan Lahan

Pada Gambar 4.3 yang merupakan *v*isualisasi data lokasi tweet menunjukkan bahwa topik *tweet* mengenai permasalahan kebakaran hutan dan lahan, dengan kata kunci 'kebakaran hutan', 'riau', dan 'kalimantan' ditulis oleh warganet negara Indonesia,

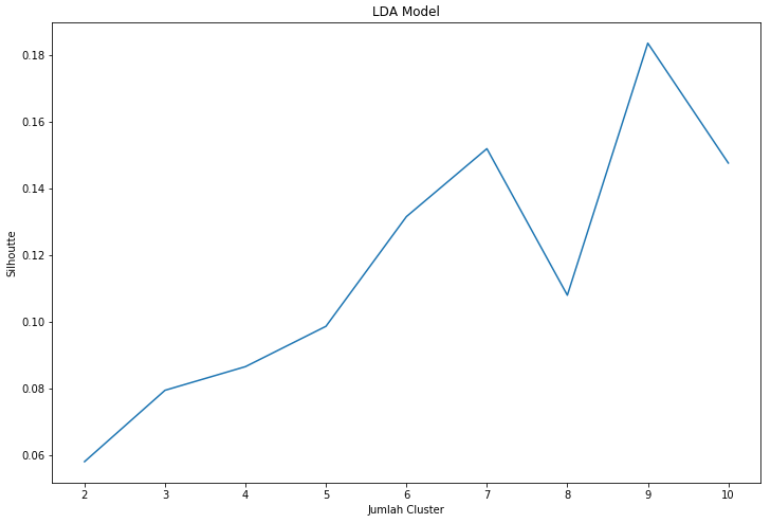
Malaysia, dan Singapura. Dengan variasi data berdasar pulau terbanyak berada di Pulau Kalimantan. Hal ini menunjukkan bahwa kejadian kebakaran hutan di Kalimantan mendapat respon yang cukup besar oleh penduduk yang menulis *tweet* di Pulau Kalimantan. Beberapa kota besar di Pulau Jawa seperti Jakarta, Surabaya, Bandung, Yogyakarta, dan Semarang juga ikut meramaikan topik *tweet* tersebut.

4.3 Analisis *Text Clustering*

Analisis *text clustering* pada penelitian ini bertujuan untuk mengelompokkan tweet mengenai permasalahan kebakaran hutan dan lahan. Analisis *text clustering* yang dilakukan pada penelitian ini menggunakan metode *Latent Dirichlet Allocation* (LDA), *K-Means*, dan DBSCAN. Hasil dari ketiga metode tersebut yaitu mendapatkan *cluster* optimal dari tiap metode.

4.3.1 Analisis *Clustering* Menggunakan Metode *Latent Dirichlet Allocation* (LDA)

LDA dapat digunakan untuk meringkas, melakukan *cluster*ing, menghubungkan maupun memproses data yang sangat besar karena LDA menghasilkan daftar topik yang diberi bobot untuk masing-masing dokumen. Distribusi yang digunakan untuk mendapatkan distribusi topik per dokumen disebut distribusi *Dirichlet.* Pada penelitian ini jumlah *cluster* yang digunakan yaitu pada rentang 2 *cluster* hingga 10 *cluster.* Selanjutnya nilai *silhouette coefficient* dari pembagian *cluster* tersebut dibandingkan untuk menentukan jumlah *cluster* yang optimal.



Gambar 4. 4 Nilai *Silhouette Coefficient* Pada Metode LDA

Gambar 4.4 diperoleh bahwa nilai *silhouette coefficient* tertinggi dengan menggunakan metode LDA yaitu ketika jumlah *cluster* sebanyak 9 *cluster*. Nilai *silhouette coefficient* yang diperoleh yaitu sebesar 0,1814. Namun nilai *silhouette coefficient* tersebut terbilang kecil untuk hasil *cluster* yang optimal, sehingga banyak *tweet* yang tidak tepat untuk di*cluster*kan dengan *tweet* lainnya. Untuk mengetahui lebih jelas kata pada *tweet* tiap *cluster* dapat dilihat pada Tabel 4.5.

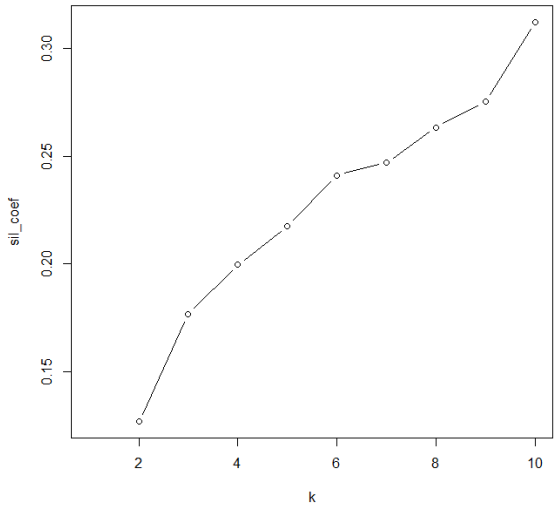
Tabel 4. 5 Jumlah Kata Terbanyak Tiap *Cluster* pada Metode LDA

Cluster	**Word 1**	**Word 2**	**Word 3**	**Word 4**	**Word 5**
***Cluster* 1**	tolong	tolak	malaysia	masker	gubernur
***Cluster* 2**	hujan	turun	alhamdulillah	rakyat	mohon
***Cluster* 3**	jambi	langit	merah	muaro	riau
***Cluster* 4**	negara	sumatera	cucu	daerah	pamer
***Cluster* 5**	paru	masyarakat	mahasiswa	hidup	kotor
***Cluster* 6**	garam	udara	semai	pesawat	kemarin
***Cluster* 7**	asap	dampak	jokowi	warga	kabut
***Cluster* 8**	manusia	hewan	marah	bunuh	usaha
***Cluster* 9**	tolong	indonesia	sumatera	padam	dana

Tabel 4.5 menunjukkan jumlah kata yang sering muncul ditiap *cluster*. Pada penelitian ini hanya diambil 5 kata terbanyak yang muncul tiap *cluster*. Dapat dilihat bahwa pembagian *cluster* tidak terbagi secara baik. Seperti pada *cluster* 4 membahas mengenai 2 topik yang berbeda yaitu mengenai cucu Pak Jokowi dengan kabut asap yang mengenai wilayah Sumatera lainnya. Selain itu, pada *cluster* 5 membahas mengenai aksi maahasiswa, dampak asap kebakaran hutan dan lahan terhadap paru-paru dan sepatu Pak Jokowi yang sempat viral ketika meninjau lokasi kebakaran hutan dan lahan. Sehingga pembagian *cluster* menggunakan metode LDA tidak optimal.

4.3.2 Analisis *Clustering* Menggunakan Metode *K-Means*

K-Means merupakan salah satu metode analisis *cluster* yang berbasis jarak atau disebut *distance-based cluster analysis.* Pada penelitian ini, jarak yang diukur yaitu bobot per kata disetiap *tweet* dengan kata lain nilai TF-IDF tiap kata disetiap *tweet*.



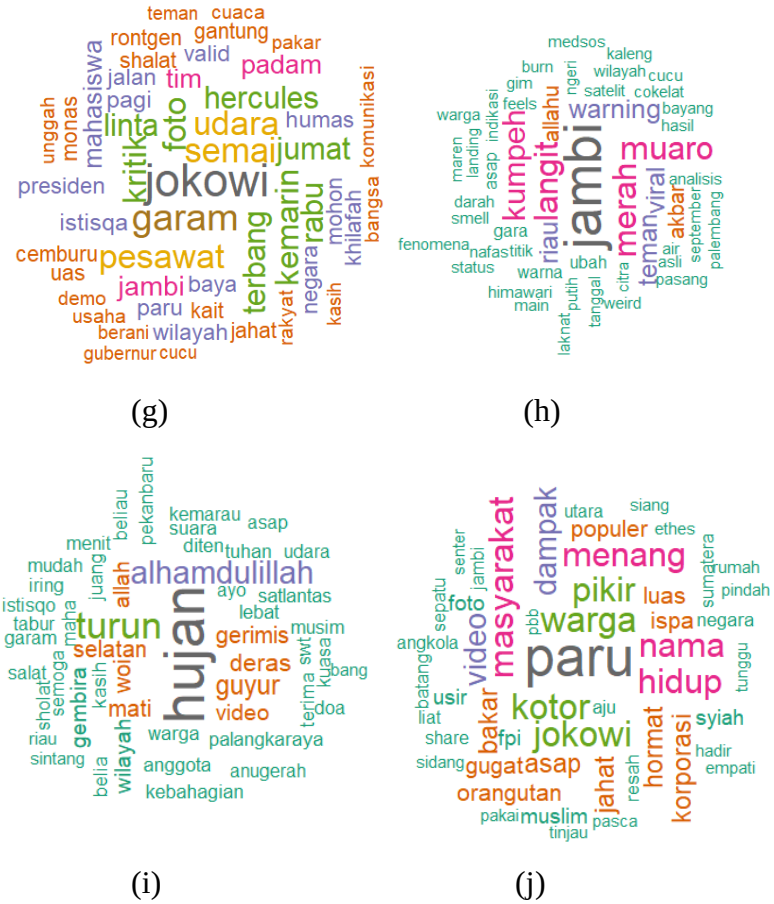
Gambar 4. 5 Nilai *Silhouette Coefficient* Metode K-Means

Berdasarkan Gambar 4.5 diperoleh bahwa nilai silhoeuette untuk metode *K-Means* yaitu ketika jumlah *cluster* sebanyak 10 *cluster*. Nilai *silhouette coefficient* yang diperoleh yaitu sebesar 0,3123. Nilai *silhouette coefficient* yang diperoleh tersebut cukup rendah, artinya sebagian besar *tweet* berada di antara dua *cluster* sehingga kurang jelas untuk menentukkan *cluster* yang tepat untuk *tweet* tersebut.

(a) (b) (c)

(d) (e) (f)

Gambar 4. 6 *Cluster* 1 (a), *Cluster* 2 (b), *Cluster* 3 (c), *Cluster* 4. (d), *Cluster* 5 (e), *Cluster* 6 (f), Metode *K-Means*



Gambar 4. 7 *Cluster* 7 (g), *Cluster* 8 (h), *Cluster* 9 (i), *Cluster* 10 (j) Metode *K-Means*

Pada Gambar 4.6 dan Gambar 4.7 menunjukkan bahwa *word cloud* dari 10 *cluster* yang terbentuk menghasilkan jumlah anggota *cluster* yang berbeda-beda. *Cluster 7* pada Gambar 4.7 (g) merupakan *cluster* yang memiliki anggota terbanyak yaitu sebanyak 3182 kata dan kata-kata yang sering muncul pada *cluster* ini adalah "jokowi", "garam", "pesawat", "semai". Kata-kata ini berhubungan dengan tindak lanjut pemerintah dalam menangani bencana kebakaran hutan dan lahan tersebut contohnya dengan melakukan hujan tiruan dengan cara menaburkan garam. Sedangkan *cluster* yang memiliki jumlah anggota terkecil adalah *cluster* 8 *7* pada Gambar 4.7 (h) dengan jumlah anggota sebanyak 294 kata. " jambi", "langit", "merah", "warning" merupakan kata-kata yang sering muncul pada *cluster* 8 dan berhubungan dengan dampak yang dirasakan oleh daerah-daerah disekitar lokasi bencana kebakaran hutan dan lahan. Berdasarkan *word cloud* yang terbentuk terlihat bahwa perbedaan antar *cluster* tidak terlalu optimal.

4.3.3 Analisis *Clustering* Menggunakan Metode *DBSCAN*

DBSCAN merupakan pengelompokkan yang didasarkan pada kepadatan (*density*) data. Konsep kepadatan yang dimaksud adalah jumlah daya yang berada dalam radius (*eps*) dan minimum points (*MinPts*). Pada penelitian ini, *minimum points* yang digunakan sebesar 300 dengan kombinasi nilai *eps.*

Tabel 4. 6 Nilai *Silhouette Coefficient* pada Metode DBSCAN

Eps	*silhouette coefficient*
0.75	0.999
0.8	0,999
0.85	0,963
0.9	0,961
0.95	0,789
1	0,761

Berdasarkan Tabel 4.6 diperoleh bahwa nilai *silhouette coefficient* tertinggi pada analisis *cluster* menggunakan DBSCAN yaitu ketika nilai radius (*eps*) sebesar 0.75 dan minimum *points* sebesar 300. Sehingga diperoleh *cluster* optimum sebanyak 3 *cluster.*

Tabel 4. 7 Jumlah *Tweet* pada Tiap *Cluster*

Cluster	Jumlah Tweet
Cluster 1	514
Cluster 2	323
Cluster 3	811

Pada Tabel 4.7 dapat dilihat bahwa jumlah *tweet* pada tiap *cluster* tidak berbeda cukup jauh dari total keseluruhan *tweet* sabanyak 10.005 *tweet.* Kata-kata pada *tweet* yang lain tidak masuk ke dalam *cluster* dikarenakan termasuk kategori *noise* dimana bobot dari kata-kata tersebut terletak jauh dari radius (*eps*) yang telah ditentukan.

4.4 Hasil *Clustering* Menggunakan Metode Terbaik

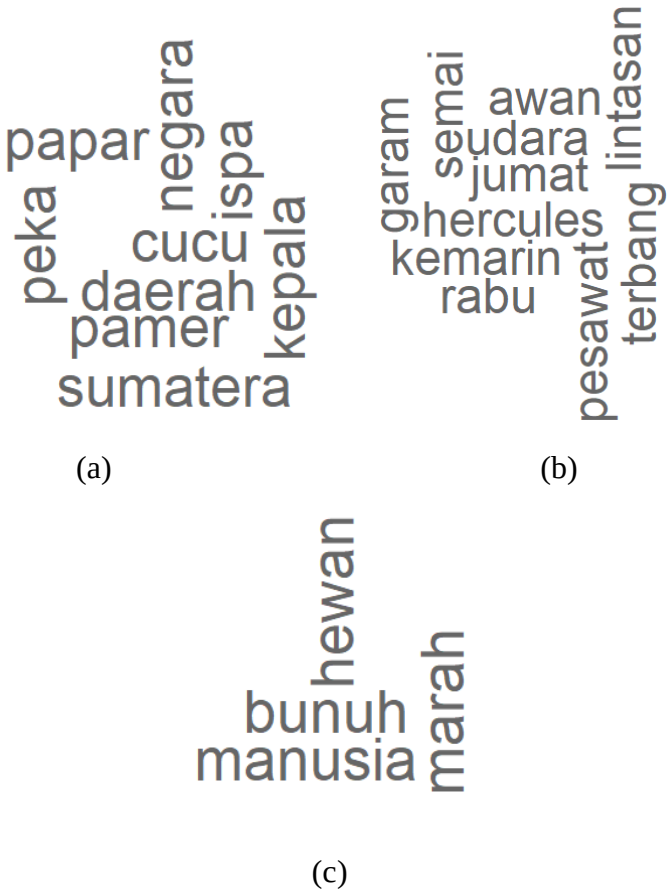
Setelah melakukan analisis *clustering* menggunakan LDA, *K-Means*, dan DBSCAN, didapatkan *cluster* optimum dengan membandingkan nilai *silhouette coefficient.*

Tabel 4. 8 Perbandingan Metode *Clustering* Terbaik

Metode	*silhouette coefficient*
LDA	0,181
K-Means	0,312
DBSCAN	0,999

Setelah melakukan analisis *clustering* dengan menggunakan tiga metode yang telah disebutkan, diperoleh bahwa jumlah *cluster* yang paling optimum yaitu hasil analisis *cluster*ing menggunakan metode DBSCAN. Nilai *silhouette coefficient* yaitu sebesar 0.999. Untuk mengetahui lebih jelas pembagian *cluster* dengan metode DBSCAN dapat dilihat melalui visualisasi menggunakan *wordcloud*.

4.5 Visualisasi *Cluster* Terbaik dengan *Wordcloud*



(a) (b)

(c)

Gambar 4. 8 *Cluster* 1 (a) *Cluster* 2 (b) dan *Cluster* 3 (c) Metode DBSCAN

Pada Gambar 4.8 dapat diperoleh informasi bahwa pada *cluster* 1 berasal dari satu *tweet* yang di*retweet* paling banyak warganet karena ukuran kata pada *wordcloud* yang sama. Begitupun dengan *cluster* 2 dan *cluster* 3 juga berasal dari satu *tweet* yang paling banyak di*retweet* oleh pengguna Twitter. Sehingga ketiga *cluster* tersebut merupakan top *tweet* mengenai permasalahan kebakaran hutan dan lahan. Pada *cluster* 1, berisi tentang kritikan kepada Pak Jokowi untuk lebih memperhatikan warganet didaerah sumatera dan kalimantan yang terkena ISPA. Sedangakan pada *cluster* 2, berisi tentang informasi pesawat Hercules yang terbang diatas Riau dan Kalimantan umtuk

menyemai garam di lintasan awan. Dan pada *cluster* 3, berisi tentang banyak pengguna Twitter yang berduka karena banyak hewan yang mati akibat kebakaran hutan dan lahan yang menurut mereka kebakaran hutan dan lahan tersebut adalah akibat perbuatan manusia.

BAB V
KESIMPULAN DAN REKOMENDASI

5.1 Kesimpulan

Berdasarkan hasil analisis pada penelitian ini, maka bisa diambil kesimpulan pada permasalahan kebakaran hutan dan lahan, pengelompokkan opini warganet di Twitter dengan menggunakan 3 metode diperoleh metode terbaik yaitu metode *Density-Based Spatial Clustering of Applications with Noise (DBSCAN).* Dengan nilai *Eps* sebesar 0.75 dan *MinPts* sebesar 300, didapatkan nilai *silhouette coefficient* tertinggi yaitu sebesar 0.999. Sehingga jumlah *cluster* optimal untuk pengelompokan opini warganet di Twitter mengenai kebakaran hutan dan lahan sebanyak 3 *cluster*. *Cluster* 1, berisi tentang kritikan kepada Pak Jokowi untuk lebih memperhatikan masyarakat didaerah sumatera dan kalimantan yang terkena ISPA. Pada *cluster* 2, berisi mengenai penanggulangan bencana kebakaran hutan dan lahan yaitu pesawat Hercules yang terbang diatas Riau dan Kalimantan untuk menyemai garam di lintasan awan. Dan pada *cluster* 3, berisi tentang dampak dari bencana kebakaran hutan dan lahan yaitu tidak sedikit warganet yang berduka karena banyaknya hewan yang mati akibat kebakaran hutan dan lahan dan menurut mereka kebakaran hutan dan lahan tersebut adalah akibat perbuatan manusia.

5.2 Rekomendasi

Rekomendasi yang dapat diberikan berdasarkan penelitian ini yaitu agar pemerintah memperhatikan masyarakat yang sedang mengalami masalah kritis terutama pada masyarakat yang terkena dampak kebakaran hutan dan lahan serta melakukan pencegahan dalam menanggulangi kebakaran hutan dengan maksimal. Penerapan penelitian ini dapat digunakan untuk mengetahui informasi, opini, aspirasi, serta kritik dari warganet terhadap kondisi yang terjadi di Indonesia. Untuk penelitian serupa selanjutnya, agar lebih teliti lagi dalam melakukan *text preprocessing* agar diperoleh struktur kata yang baik dan tepat. Selain itu, pengambilan data *tweet* sebaiknya dalam rentang waktu yang panjang agar opini warganet yang diperoleh lebih beragam.

DAFTAR PUSTAKA

Adriani, M., Asian, J., Nazief, B., Tahaghoghi, S. M., & Williams, H. E. (2007). Stemming Indonesian: A Confix-Stripping Approach. *ACM Transactions on Asian Language Information Processing (TALIP). 6(4)*, 1-33.

Alfina, T., Santosa, B., & Barakbah, A. R. (2012). Analisa Per-bandingan Metode Hierarchical Clustering, K-Means dan Gabungan Keduanya dalam Cluster Data. *urnal Teknik ITS. 1(1)*, 1-5.

Ariadi, D., & Fithriasari, K. (2015). Klasifikasi Berita Indonesia Menggunakan Bayesian Classification dan Support Vector Machine dengan Confix Stripping Stemmer. *Jurnal Sains dan Seni ITS.*, 4(2). 20.

Asian, J. (2007). *Effective Techniques for Indonesian Text Retrieval*. Melbourne: RMIT University.

Berry, M., & Kogan, J. (2010). *Text Mining: Application and heory* . Chichester: John Wiley & Sons, Ltd.

Bhuyan, R., & Borah, S. (2013). A Survey of Some Density Based Clustering Techniques. *National Conference on Advancementsin Information, Computer, and Communication.*

Bing, L. (2010). *Handbook of Natural Language Processing.* Boca Raton: CRC Press.

Blei, D. (2012). Probabilistic Topic Models. *Communications of the ACM. 55(4)*, 77–84.

Cambridge University, P. (2008). *Cambridge Advanced Learner's Dictionary (Third ed.).* Cambridge University, Press.

Campbell, J., Hindle, A., & Stroulia, E. (2014). Latent Dirichlet Allocation: Extracting Topics from Software Engineering Data. In M. Bird & T. Zimmermann (Eds.). *The Art and Science of Analyzing Software Data (1 ed.).* Burlington, MA: Morgan Kaufmann.

Carlantonio, L., & Costa, R. (2009). Exploring a Genetic Algorithm for Hypertext Document Clustering . In N. Nedjah, L. Mourelle, J. Kacprzyk, F. Franca, & A. Souza, *Intelligent Text Categorization and Clustering* (pp. 95-117). Verlag Berlin Heidelberg: Springer.

Castella, Quim, Sutton, & Charles. (2014). Word Storm: Multiples of Wordclouds for Visual Comparison of Documents.. . *In Proceedings of the 23rd International Confe-rence on World Wide Web*, (pp. 665–676).

Dragut, E., Fang, F., Sistla, P., & Yu, C. (2009). *Stopword and Related Problems in Web Interface Integration.* Chicago: University of illinois.

Durajati, C., & Gumelar, A. B. (2012). Pemanfaatan Teknik Supervised Untuk Klasifikasi Teks Bahasa Indonesia. *Jurnal Link Vol 16/No. 1, ISSN 1858 - 4667* , 1-8.

Feldman, R., & Sanger, J. (2007). *The Text Mining Handbook : Advanced Approaches in Analyzing Unsrtructed Data.* New York: Cambridge University Press.

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining Concepts and Techniques.* USA: Morgan Kaufmann.

Jannah, S. Z., Fithriasari, K., Prastyo, D. D., & Iriawan, N. (2018). Text Mining for Identifying and Visualizing Topics of Citizen Opinion in Media Centre Surabaya. *International Conference on Theretical and Applied Statistics*, (p. 82). Surabaya.

Johnson, R. A., & Wichern, D. W. (2007). *Applied Multivariate Statistical Analysis (6th ed.).* United States of America: Pearson Prentice Hall.

Kogan, J. (2007). *Introduction to Clustering Large and High Dimensional Data.* New York: Cambrige University Press.

Kumar, S., Morstatter, F., & Liu, H. (2014). *Twitter Dat Analytics.* New York: Spinger.

Mujilahwati, S. (2016). Pre-processing Text Mining pada Data Twitter. *Seminar Nasional Teknologi Informasi dan Komunikasi* (pp. 49-56). Yogyakarta: Sentika.

Narayana, B., & Kumar, S. (2015). A New Clustering Tech-nique on Text. *International Journal of Science Engi-neering and Advance Technology (IJSEAT)*, 69-71.

National Wildlife Coordination, G. (2001). *Smoke management guide for prescribed and wildland fire*. Boise: National Interagency Fire Center. Archived (PDF).

Tacconi, L. (2003). Kebakaran Hutan di Indonesia: Penyebab, Biaya, dan Implikasi Kebijakan. *CIFOR Occasional Paper No. 38(i)*, 38.

Tala, F. Z. (2003). *A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia. M.Sc.* Amsterdam: Universiteti van Amsterdam.

Tan, P., Steinbach, M., & Kumar, V. (2006). *Introduction to Data Mining* . Boston: Pearson Addison-Wesley.

U.S. Environmental Protection, A. (2009). *Air quality index: A guide to air quality and health.* U.S. Environmental Protection Agency.

Weiss, S., Indurkhya, N., Zhang, T., & Damerau, F. (2005). *Text Mining Predicitive Methods for Analyzing Unstructures Information.* New York: Spinger Science Business Media. Inc.

Zarrella, D. (2010). *The Social Media Marketing Book.* Canada: O'Reilly Media, Inc.

LAMPIRAN

Lampiran 1. *Crowing* Data

```
from tweepy.streaming import StreamListener
from tweepy import OAuthHandler
from tweepy import Stream
#Variables that contains the user credentials to access Twitter API
#This is a basic listener that just prints received tweets to stdout.
class StdOutListener(StreamListener):
    def on_data(self, data):
        print(data)
        return True

    def on_error(self, status):
        print(status)

if __name__ == '__main__':

    #This handles Twitter authetification and the connection to Twitter Streaming API
    l = StdOutListener()
    auth = OAuthHandler(consumer_key, consumer_secret)
    auth.set_access_token(access_token, access_token_secret)
    stream = Stream(auth = auth, listener = l, tweet_mode= 'extended')

    #This line filter Twitter Streams to capture data by the keywords: 'python',
'javascript', 'ruby'
    stream.filter(track=['dpr'])
```

Lampiran 2. *Preprocessing* Data

```
import pandas as pd
import re
```

```
import nltk
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from nltk.tokenize import word_tokenize
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from collections import OrderedDict
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer

#Menghilangkan link
datalink=[]
for line in text:
    result=re.sub(r"http\S+"," ",line)
    datalink.append(result)
#Menghilangkan tanda retweet
datart=[]
for line in datalink:
    result=re.sub(r"RT"," ",line)
    datart.append(result)
#Menghilangkan username
dataline=[]
for line in datausername:
    result=re.sub("\n"," ",line)
    dataline.append(result)
#Menghilangkan Angka
datanum=[]
for line in dataline:
    result=re.sub("\d"," ",line)
    datanum.append(result)
#Menghilangkan Baris Baru
dataline=[]
```

```
for line in datausername:
    result=re.sub("\n"," ",line)
    dataline.append(result)
#Menghilangkan hastag
datahastag=[]
for line in datanum:
    result=re.sub(r"#\S+","",line)
    datahastag.append(result)
#Menghilangkan emoticon
dataemot=[]
for line in datahastag:
    result=re.sub(r'<*?>',"",line)
    dataemot.append(result)
#Menghilangkan Punction
datapunction=[]
for line in dataemot:
    result=re.sub(r"[^\w\s]"," ",line)
    datapunction.append(result)
#Menghilangkan double space
datadoublespace=[]
for line in datapunction:
    result=re.sub(r'\s+',' ',line)
    datadoublespace.append(result)
#Menghilangkan huruf kapital
datacfold=[]
for line in datadoublespace:
    a=line.lower()
    datacfold.append(a)
#Stemming
factory = StemmerFactory()
stemmer = factory.create_stemmer()
data_stemmed = map(lambda x: stemmer.stem(x), datacfold)
```

```
data_stemmed = list(data_stemmed)
#Sinonim kata
def replace_all(text,dic):
    for i, j in dic.items():
        text = text.replace(i, j)
    return text
dic= OrderedDict(kata)
datanew = []
for line in data_stemmed:
    result = replace_all(line, dic)
    datanew.append(result)
#Stopwords
stopword = open("stopwords-id.txt","r").read()
datafinal=[]
for line in datanew:
    word_token = nltk.tokenize.word_tokenize(line)
    word_token = [word for word in word_token if not word in stopword and not
word[0].isdigit()]
    datafinal.append(" ".join(word_token))
```

Lampiran 3. *TF-IDF*

```
#Document Term Matrix
vectorizer = CountVectorizer(min_df=0.01)
x = vectorizer.fit_transform(datafinal)
DTM = pd.DataFrame(x.toarray(),columns = vectorizer.get_feature_names())
DTM['tweet']=datafinal
DTM['sum_features'] = DTM.sum(axis=1)
DTM = DTM.loc[DTM['sum_features'] != 0].drop(['sum_features'], axis=1)
DTM.to_csv('DTM_stream.csv')
#TFIDF
vectorizer = TfidfVectorizer(min_df=0.01)
```

```
vec = vectorizer.fit_transform(datafinal)
TFIDF = pd.DataFrame(vec.toarray(), columns = vectorizer.get_feature_names())
TFIDF['tweet']=datafinal
TFIDF['sum_features'] = TFIDF.sum(axis=1)
TFIDF = TFIDF.loc[TFIDF['sum_features'] != 0].drop(['sum_features'], axis=1)
TFIDF.to_csv('TFIDF_stream.csv')
```

Lampiran 4. *Latent Dirichlet Allocation (*LDA)

```
from sklearn.decomposition import LatentDirichletAllocation, TruncatedSVD
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.model_selection import GridSearchCV
from pprint import pprint
import re, nltk, gensim

# Plotting tools
import pyLDAvis
import pyLDAvis.sklearn
import matplotlib.pyplot as plt
%matplotlib inline

import warnings
warnings.filterwarnings("ignore",category=DeprecationWarning)

df = TFIDF
data = df['tweet']
vectorizer = CountVectorizer(analyzer='word', min_df=0.01, max_features=50000)
data_vectorized = vectorizer.fit_transform(data)
from sklearn.metrics import silhouette_samples, silhouette_score

sil=[]
topik=[2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
for i in topik:
    lda_model = LatentDirichletAllocation(batch_size=128, doc_topic_prior=None,
            evaluate_every=-1, learning_decay=0.7, learning_method='online',
            learning_offset=10.0, max_doc_update_iter=100, max_iter=15,
            mean_change_tol=0.001,
            n_components=i, n_jobs=1, perp_tol=0.1, random_state=10,
            topic_word_prior=None, total_samples=1000000.0, verbose=0)
    lda_output = lda_model.fit_transform(data_vectorized)
    # column names
    topicnames = ["Topic" + str(j) for j in range(lda_model.n_components)]
    # index names
    docnames = ["Doc" + str(j) for j in range(len(data))]
    # Make the pandas dataframe
    df_document_topic = pd.DataFrame(np.round(lda_output, 2), columns=topicnames,
index=docnames)
    # Get dominant topic for each document
    dominant_topic = np.argmax(df_document_topic.values, axis=1)

    df_document_topic['dominant_topic'] = dominant_topic
    ss = silhouette_score(data_vectorized, dominant_topic)
    sil.append(ss)
    print('topik :', i, 'Silhouette :', ss)

#Grafik Silhouette K=2 hingga K=10
plt.figure(figsize=(12, 8))
plt.plot(topik, sil)
plt.title("LDA Model")
plt.xlabel("Jumlah Cluster")
plt.ylabel("Silhoutte")
plt.show()

# Topic-Keyword Matrix
```

```
df_topic_keywords = pd.DataFrame(lda_model.components_)
# Assign Column and Index
df_topic_keywords.columns = vectorizer.get_feature_names()
df_topic_keywords.index = topicnames
# View
df_topic_keywords.tail()
# Show top n keywords for each topic
def show_topics(vectorizer=vectorizer, lda_model=lda_model, n_words=20):
    keywords = np.array(vectorizer.get_feature_names())
    topic_keywords = []
    for topic_weights in lda_model.components_:
        top_keyword_locs = (-topic_weights).argsort()[:n_words]
        topic_keywords.append(keywords.take(top_keyword_locs))
    return topic_keywords
topic_keywords = show_topics(vectorizer=vectorizer, lda_model=lda_model,
n_words=5)
# Topic - Keywords Dataframe
df_topic_keywords = pd.DataFrame(topic_keywords)
df_topic_keywords.columns = ['Word '+str(i) for i in
range(df_topic_keywords.shape[1])]
df_topic_keywords.index = ['Topic '+str(i) for i in range(df_topic_keywords.shape[0])]
df_topic_keywords

cmp = lda_model.components_ / lda_model.components_.sum(axis=1)[:, np.newaxis]
def print_top_words(model, feature_names, n_top_words):
    for topic_id, topic in enumerate(cmp):
        print('\nTopic %d:' % int(topic_id + 1))
        print(".join([feature_names[i] + ' * ' + str(round(topic[i], 3))
            +' + ' for i in topic.argsort()[:-n_top_words - 1:-1]]))

n_top_words = 5
feature_names = vectorizer.get_feature_names()
```

```
print_top_words(lda_model, feature_names, n_top_words)


lda_output = lda_model.transform(data_vectorized)


topic_prob = pd.DataFrame(lda_output)
topic_prob['tweet'] = df['tweet']
topic_prob
```

Lampiran 5. *Clustering* Menggunakan *K-Means*

```
library(dplyr)
library(tidyr)
library(caret)
library(RColorBrewer)
library(ggplot2)
library(factoextra)
library(NbClust)
library(wordcloud)
library(wordcloud2)
library(tm)
library(cluster)

TFIDF<-read.csv("//Mac/Home/Downloads/TFIDF_stream (1).csv")
df_clust<-TFIDF%>%select(-c(tweet, X))

##K-Means Evaluation
Kmeans <- function(data)
{
  sil_coef <- matrix()
  for (k in 2:10)
  {
    set.seed(12)
```

```
   KMeans <- kmeans(data, centers = k, nstart = 100)
   silcoef <- silhouette(KMeans$cluster, dist(data))
   sil_coef[k] <- summary(silcoef)$avg.width
  }
  win.graph()
  plot(sil_coef, xlab = "k", type = "b")
  win.graph()
  list(Sil_Coef = sil_coef)
}
Kmeans(df_clust)

set.seed(12)
kmeans<-kmeans(df_clust, 10, nstart = 20)
kmeans$size
silcoef <- silhouette(kmeans$cluster, dist(df_clust))
summary(silcoef)$avg.width

TFIDF$clust<-kmeans$cluster
clusters<-TFIDF%>%select(c(tweet, clust))
for (i in 1:10){
  docs<-Corpus(VectorSource(clusters$tweet[clusters$clust==i]))
  dtm <- TermDocumentMatrix(docs)
  m <- as.matrix(dtm)
  v <- sort(rowSums(m),decreasing=TRUE)
  d <- data.frame(word = names(v),freq=v)
  win.graph()[i]
  wordcloud(words = d$word, freq = d$freq, min.freq = 1,
            max.words=50, random.order=FALSE, rot.per=0.35,
            scale=c(2,0.5), colors=brewer.pal(8, "Dark2"))
}
```

Lampiran 6. *Clustering* Menggunakan DBCSAN

```
TFIDF<-read.csv("//Mac/Home/Downloads/TFIDF_stream (1).csv")
df_clust<-TFIDF%>%select(-c(tweet, X))

DBSCAN <-dbscan::dbscan(df_clust, eps=0.1, minPts = 200)
x<-data.frame(df_clust, DBSCAN$cluster)
y<-x[which(x$DBSCAN.cluster!=0),]
silcoef<-silhouette(y$DBSCAN.cluster,dist(y[,-516]))
sil_coef_0.1<-summary(silcoef)$avg.width
sil_coef_0.1

z <-data.frame(df_clust,TFIDF$tweet, DBSCAN$cluster)
y<-z[which(z$DBSCAN.cluster!=0),]

clusters<-y%>%select(c(TFIDF.tweet, DBSCAN.cluster))
for (i in 1:max(y$DBSCAN.cluster)){
  docs<-Corpus(VectorSource(clusters$TFIDF.tweet[clusters$DBSCAN.cluster==i]))
  dtm <- TermDocumentMatrix(docs)
  m <- as.matrix(dtm)
  v <- sort(rowSums(m),decreasing=TRUE)
  d <- data.frame(word = names(v),freq=v)
  win.graph()[i]
  wordcloud(words = d$word, freq = d$freq, min.freq = 1,
            max.words=50, random.order=FALSE, rot.per=0.35,
            scale=c(2,0.5), colors=brewer.pal(8, "Dark2"))
}
```