BAB II

TINJAUAN PUSTAKA

2.1 Organisasi Pemerintah

Organisasi pemerintah adalah sebuah organisasi yang bertujuan untuk melayani masyarakat dan sebagai pelaksana kekuasaan eksekutif (Abdulah, 2016; Kue, 2014). Pembagian wewenang urusan pemerintahan dibagi tiga yaitu absolut(pemerintah pusat), konkuren(pemerintah pusat dan daerah), dan umum(presiden) (Yudhoyono, 2014). Kekuasaan pemerintah dibagi seusai dengan tingkatannya. Tingkat paling tinggi adalah organisasi pemerintah pusat kemudian pemerintah provinsi daerah dan diikuti oleh pemerintah kota dan kabupaten. Kekuasaan organisasi pemerintah pusat dipegang oleh presiden dan dibantu oleh wakil presiden (Indonesia P. , Undang-Undang Dasar Negara Republik Indonesia, 1945). Suatu organisasi pemerintah merancangkan organisasinya lima tahun sekali. Presiden dan wakil presiden memiliki perbedaan misi pada setiap periode. Perubahan misi akan merubah tujuan dan strategi organisasi pemerintah.

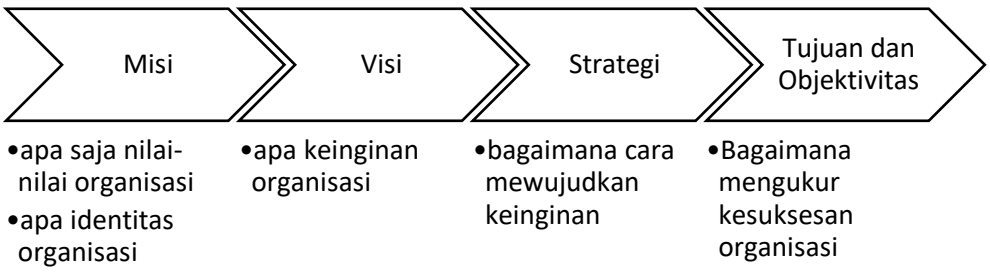
Tabel 2.1 Misi Calon Presiden Terpilih

Tahun	Misi Presiden dan Wakil Presiden
2014-2019	Mewujudkan keamanan nasional yang mampu menjaga kedaulatan wilayah, menopang kemandirian ekonomi dengan mengamankan sumber daya maritim, dan mencerminkan kepribadian Indonesia sebagai negara kepulauan. Mewujudkan masyarakat maju, berkeseimbangan dan demokratis berlandaskan negara hukum. Mewujudkan politik luar negeri bebas-aktif dan memperkuat jati diri sebagai negara maritim.(*) Mewujudkan kualitas hidup manusia Indonesia yang tinggi, maju dan sejahtera. Mewujudkan bangsa yang berdaya-saing. Mewujudkan Indonesia menjadi negara maritim yang mandiri, maju, kuat dan berbasiskan kepentingan nasional.(*) Mewujudkan masyarakat yang berkepribadian dalam kebudayaan.

Tahun	Misi Presiden dan Wakil Presiden
2019-2024	Peningkatan kualitas manusia Indonesia. Struktur ekonomi yang produktif, mandiri, dan berdaya saing. Pembangunan yang merata dan berkeadilan.(*) Mencapai lingkungan hidup yang berkelanjutan.(*) Kemajuan budaya yang mencerminkan kepribadian bangsa. Penegakan sistem hukum yang bebas korupsi, bermartabat, dan terpercaya. Perlindungan bagi segenap bangsa dan memberikan rasa aman pada seluruh warga. Pengelolaan pemerintahan yang bersih, efektif, dan terpercaya. Sinergi pemerintah daerah dalam kerangka Negara Kesatuan.

(*) perbedaan misi

Perubahan strategi dan tujuan organisasi dikarenakan misi yang berubah pada saat perencanaan organisasi pemerintah. Perencanaan organisasi memiliki kaitan dengan arah organisasi, tujuan organisasi, dan strategi organisasi. Komponen perencanaan organisasi terdiri dari visi misi organisasi, penyusunan strategi organisasi, dan tujuan organisasi serta objektivitas organisasi. Urutan peranan dirangkum dalam Gambar 2.1 (Carpenter, Erdogan, & Bauer, 2010)



Gambar 2.1 Peran Kunci Visi dan Misi
(Sumber: Carpenter, Erdogan, & Bauer, 2010, telah diolah kembali)

Misi-Misi yang sudah dirumuskan selanjutnya di sebarluaskan melalui berbagai media salah satunya media sosial twitter. Perilaku manusia semakin hari semakin tidak terpisahkan dari dunai maya sehingga patut menjadi perhatian serius mengenai pemanfaatan media sosial menurut kajian sikologis sosial. Terdapat perilaku unik yang dihasilakan oleh pelaku media sosial

seperti ajang penunjukkan jati diri melalui foto, gaya hidup berbelanja melalui belanja daring, pembentukan kolektif mind. Salah satu cyberwar yang terjadi di indonesia pada tahun 2014 adalah atmosfer kegaduhan yang disebabkan oleh opini maupun pemberitaan mengenai figur jokowi maupun prabowo. Semakin besar ketegangan yang dihasilkan, maka semakin terbesar juga terjadi *kolektif mind* pada pendukung pasangan calon. *Kolektif mind* yang dibentuk oleh masing-masing pasangan calon berisi tentang kepentingan penyaluran misi yang telah rumuskan (Mulawarman & Nurfitri, 2017).

2.2 Data Mining

Data Mining adalah sebuah analisis untuk mencari informasi dari kumpulan data dengan teknik analisis tertentu**.** *Data Mining* dapat menyelesaikan permasalahan analisa data berskala besar. Terdapat beberapa sumber data berskala besar seperti *word wide web*, *financial interaction*, *user interaction*, dan *internet of thing.* Data berskala besar dapat muncul tidak terstruktur dan tidak cocok untuk pemrosesan otomatis (Aggarwal, Data Mining, 2015). *Data Mining* dapat mengatasi masalah data tersebut dengan menggunakan 4 operasi utama dalam teknik *data mining.* dan dapat dilihat di tabel berikut ini

Tabel 2.2 *Data Mining Operations* dan *Association Techiques*

Operations	**Techniques**
Association Pattern	Brute Force Algoritms Apriori Algoritms Enumeration-Tree Algoritm Recursive Suffix-Base Pattern Growth Method
Cluster Analysis	Representative-Base Algoritms Hierarical Clustering Algoritms Probabilistic Model-Base Algoritms Gird-Base and Density-Base Algoritms Graph-Based Algoritms
Outlier Analysis	Local Outlier Factor Histogram and gird-base model Kernel Density Estimation
Classification	Decission Tree

Operations	Techniques
	Naïve Bayes Logistic Regresion Support Vector Machine Neural Network

(Sumber: Aggarwal, Data Mining, 2015, telah diolah kembali)

1. *Association Pattern*

   Teknik ini bertujuan untuk mencari keterkaitan antar *record* atau kumpulan *record* yang ada dalam database. Teknik ini mencari sebab-akibat kemunculan peristiwa yang dipengaruhi peristiwa lain (Aggarwal, Data Mining, 2015).

2. *Cluster Analysis*

   Tujuan dari cluster analysis adalah untuk mempartisi kumpulan data ke dalam beberapa kelompok yang memiliki sifat yang sama. Teknik ini menggunakan pendekatan pembelajaran tidak terkontrol(*unsupervised learning)* untuk mengelompokkan data yang bersifat homogen (Aggarwal, Data Mining, 2015).

3. *Outlaier Analysis*

   Teknik ini digunakan untuk mencari data yang janggal dalam kelompok data(Outliers). Outlier dapat dilihat dengan konsep cluster yang terdapat data yang sendirian di dalam suatu cluster. Terdapat beberapa manfaat *outlier analysis* seperti *data cleaning*, *credit card fraud*, dan *network intrusion detection* (Aggarwal, Data Mining, 2015).

4. *Classification*

   Classification adalah teknik untuk mempelajari struktur dataset contoh yang akan menghasilkan model. Model ini digunakan untuk memprediksi dataset lain yang belum berlabel. Classification merupakan *supervised learning* karena menggunakan dataset berlabel untuk mengetahui struktur suatu kelompok data (Aggarwal, Data Mining, 2015).

Dalam penelitian ini, teknik *Cluster Analysis* digunakan untuk mengetahui kelompok misi dari tweet karena penulis menganggap teknik ini paling relevan untuk kategorisasi misi. *Cluster analysis* termasuk pembelajaran tanpa pengawasan dikarenakan analisis ini tidak memerlukan pelabelan. *Cluster Analysis* dapat mengetahui kelompok tweet yang termasuk data tidak terstruktur.

2.3 *Tool Data Mining*

Tool data mining adalah alat yang digunakan untuk menjalankan proses data mining sehingga lebih cepat dari pada proses manual. Percepatan ini diperlukan karena data yang digunakan berjenis *big data.* Beberapa *tool data mining* pada penelitian ini, seperti:

a) Rapid Miner v.9.3
b) Python v.3.7
c) Library Genzim v.3.8.1
d) Library Sklearn v.0.22.1

Rapid Miner merupakan tool data mining yang dapat digunakan untuk mencari data, menganalsis data, dan memvisualisasi data. Rapid miner memiliki sifat *human friendly* dan mensupport API sehingga user lebih mudah menggunakan tool tanpa harus mengkonfigurasi terlebih dahulu. Namun, rapid miner tidak memberi izin pengguna untuk mengubah operasi logika yang sudah ada. Untuk mengatasi masalah tersebut, Python v.3.7 dapat digunakan sebagai alternatif tool lainnya.

Python v.3.7 merupakan generasi terbaru dalam evoluasi bahasa pemrograman python. Pada geneasi ini, python sudah memiliki beberapa library yang dapat digunakan untuk proses data mining. bahasa pemrograman python juga tidak melakukan pengemasan coding sehingga kode-kode *library* masih bisa dikombinasikan dengan metode yang lebih beragam. Contoh library python yang digunakan pada penelitian ini adalah Genzim v.3.8.1 dan Sklearn v.0.22.1.

Genim v.3.8.1 merupakan library python yang digunakan untuk mencari kata terdekat sesuai dengan makna dan penulisannya. Genzim memiliki sifat memory-independent sehingga dapat memproses lebih besar dari memory yang disediakan. Genzim memerlukan model yang sudah disediakan fasttext untuk menjalankan perhitungan kata terdekat. Genzim tidak dapat melakukan analsisis cluster dikarenakan genzim hanya terbatas untuk perhitungan kedekatan kata. Untuk mengatasi kekurangan genzim, Sklearn v.0.22.1 digunakan untuk melakukan perhitungan lebih lanjut tentang cluster.

Sklearn v.0.22.1 merupakan library python yang digunakan untuk mengalalisis data bersekala besar. Sklearn memuat beberapa fungsi seperti cluster, klasifikasi, outliyer detection, dan vektorisasi teks. Tool-tool tersebut kemudian digunakan untuk proses Twitter crawling, preprocessing, dan analsis cluster.

2.4 Twitter Crawling

Crawling adalah tahap yang memiliki dampak signifikan terhadap proses sistem pembelajaran. *Crawling* bertujuan untuk mengambil halaman website dengan menggunakan query tertentu dalam server. Mekanisme *Crawling* memiliki mekanisme yang sama dengan *Web page base* yaitu dengan menggunakan *Hypertext Transfer Protocol*(HTTP). Mekanisme c*rawling* pada Aplikasi Twitter dapat menggunakan *Application Programming Interface Streaming(*APIs) (Aggarwal, Machine Learning for Text, 2018).

2.5 Preprocessing

Preprocessing diperlukan untuk mengubah format dari *unstructured format* menjadi *structured format* dan multidimensional. Tweet mengandung karakteristik yang berbeda dengan text biasa. Tweet memiliki *Special symbol* seperti *username*, *Retweet*, *hashtag*(#). Selain itu, tweet memiliki kata-kata tidak normal seperti *slang* dan singkatan (Hidayatullah, 2017). Terdapat 8

(delapan) langkah dalam *pre-processing* seperti *tokenization*, *Special symbol on Twitter Removal.* s*temming*, *stopword removal*, normalisasi kata, dan pembuatan *Term Frequency-Inverts Document frequency(TF-IDF).*

2.5.1 Removing Uniform Resource Locator(URL)

Tweet memuat berbagai jenis kata tidak terkecuali URL. URL merupakan alamat website tertentu yang tidak mencirikan topik dari tweet tersebut (Hidayatullah, 2017). URL dihapus karena penelitian ini berfokus pada kata yang dikandung tweet.

2.5.2 Removing Special Symbol on Twtter

Twitter memiliki spesial simbol seperti *hashtag*(#), *username*(@username), dan *retweet*(RT). Penghapusan karakter username (@) dan retweet (RT) dilakukan secara menyeluruh karena tidak memuat topik. Sedangkan, penghapusan hashtag (#) hanya dilakukan pada lambang pagarnya saja karena masih memuat topik tweet yang sedang dibicarakan (Hidayatullah, 2017).

2.5.3 Removing Symbol ASCII, Number, and Punchtuation

Tweet biasanya memuat simbol, angka, dan tanda baca. Penelitian hidayatullah menggunakan tahap ini agar *output preprocessing* dapat ditampilkan dengan jelas (Hidayatullah, 2017). Semua ASCII, angka, dan tanda baca dapat dihapus dengan menggunakan *reguler expressions* python.

2.5.4 Tokenization

Tokenization adalah proses pemotongan kalimat atau tweet yang berbentuk urutan kata menjadi potongan-potongan karakter. Tujuan *tokenization* adalah untuk memudahkan pemrosesan kalimat karena kalimat memiliki sifat sequensial dan merupakan *dependency-oriented data type.* Berikut ini adalah contoh *tokenization* (Manning, Raghavan, & Schütze, 2009):

Input :	Saya telah menggunakan hak pilih saya sebagai warga negara
Output :	Saya | Telah | Menggunakan | Hak | Pilih | Saya | Sebagai | Warga | Negara

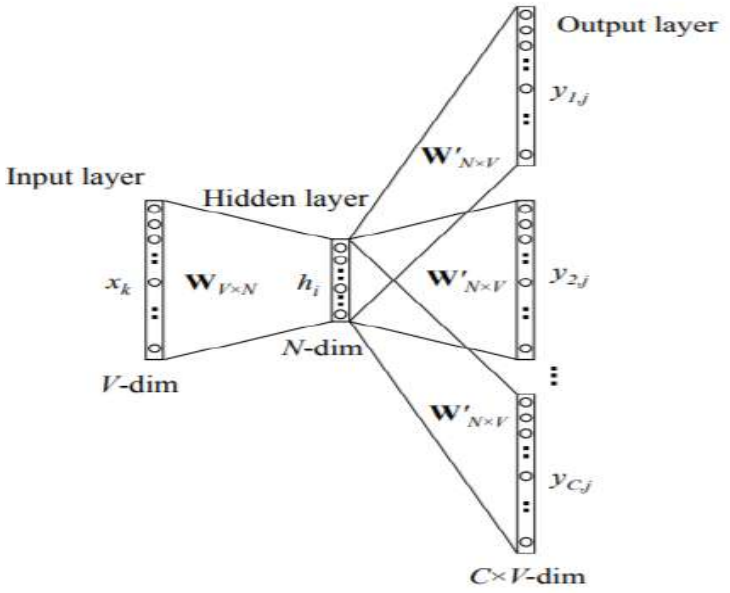
Gambar 2.2 *Tokenizing*
(Sumber : Manning, Raghavan, & Schütze, 2009, telah diolah kembali)

Tokenization dapat mengubah kalimat menjadi potongan kata-kata sehingga dapat diimplementasi oleh representasi ruang vektor. Representasi ruang vektor menggunakan frekuensi kata sebagai dasar analisis. Ruang vektor ini berbentuk multidimensional berukuran $n\ x\ d$ dengan n adalah banyak dokumen dan d adalah terms/kata. Ruang vector ini dapat disebut documents-term matrix (Manning, Raghavan, & Schütze, 2009).

2.5.5 Normalisasi kata

Banyak *tweet* mengandung kata yang tidak baku seperti kombinasi kata angka, pemanjangan kata, dan kata gaul. Hidayatulah meneliti tentang *pre-processing* Twitter Bahasa Indonesia. Hasil penelitiannya memiliki masalah yang disebabkan karena adanya kata singkatan. Hal ini dikarenakan penelitian ini menggunakan normalisasi kata secara manual (Hidayatullah, 2017).

Thomas miklov *et al* (Tomas Mikolov, 2013) menemukan metode untuk memprediksi konteks(kata sekitar) berdasarkan kata target. Metode tersebut dinamakan Skip-Gram. Skip-Gram mampu merepresentasi vektor dari data *training* kata. *Vector input* pada metode skip-gram berisikan *subwords* dari kata target dan *Vector output* berisikan *subwords* dari kata konteks. Dari penelitiannya, tidak hanya menemukan kesamaan sintak(kata) tapi juga semantik(makna). Arsitektur Skip-gram dapat dilihat pada gambar 2.3.

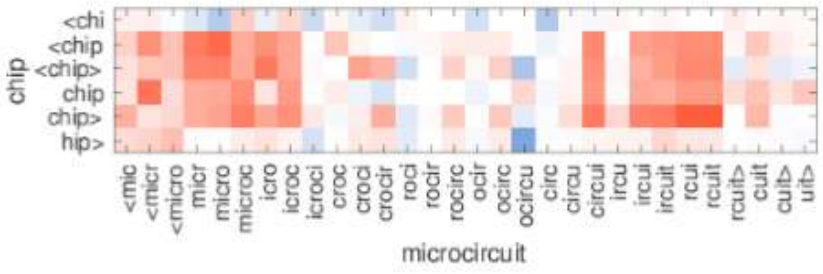


Gambar 2.3 Arsitektur Skip-Gram
(Sumber : Tomas Mikolov, 2013, telah diolah kembali)

Vektor *V-dim* berisikan 2.000.000 kamus kata dengan dimensi *N-dim* berisikan 300 dimensi. *V-dim* merupakan dimensi *subwords* dari kata target. Model Skip-gram memprediksi kata yang terdekat berdasarkan *output Skip-Gram* dan *cosine similarity*. Model *Skip-gram* dibuat dari iterasi data sebanyak satu kali dengan *learning-rate* sebanyak 0.5. Dimensi model *skipgram* juga menyesuaikan dengan jumlah *subwords* dari data *trainning*. *Subwords* kata diambil dari tiga(3) karakter sampai dengan enam(6) karakter. *subword-subword* ini akan dijadikan *dimensi/feature input*. Setelah *input* dari masing-masing data *training* diketahui, setiap *input subwords* akan dikalikan dengan bobot dan dimasukkan ke dalam fungsi aktifasi *sigmoid*. Setelah itu, setiap bobot hasil aktifasi *sigmoid* akan dikalikan dengan bobot output dan dimasukkan ke dalam fungsi aktifasi *hierarchical softmax*. Setelah itu, setiap output dirangkai menjadi suatu matrik berdimensi sesuai dengan dimensi/*feature input* dan diberi label kata input yang sebenarnya (Bojanowski, 2017). Output-output ini akan digunakan

untuk mencari kesamaan kata dari kata *typo* atau kata yang tidak ada di KBBI. Berikut ini adalah langkah-langkah mencari kata terdekat:

1. Memisahkan kata *typo* menjadi *subwords* antara 3 (tiga) sampai 6 (enam) karakter.
2. Membuat Output dari *skipgram* dengan cara yang sama dengan cara *trainning* model.
3. Menghitung *nearest neighbor menggunakan* teknik *cosine similarity* dari output kata *typo* tersebut dengan kata lain yang terdapat di *list* matrik *output*.
4. Menampilkan sepuluh kata terdekat berdasarkan *nearest neighbor*.



Gambar 2.4 Contoh Hasil Perhitungan *Similarity*

2.5.6 Modifikasi Enhanced Confix Stripping Stemmer

Dalam kasus tata bahasa, kata dasar yang memiliki arti sama menjadi berbeda bentuk kata seperti kata dasar “makan” menjadi “memakan” saat menjadi kata kerja, “makan” menjadi “makanan” saat menjadi kata benda, dan “makan” menjadi “dimakan” saat menjadi kata kerja pasif. Untuk mencari kata dasar, *Stemming* dilakukan dengan menggunakan berbagai metode (Manning, Raghavan, & Schütze, 2009). Terdapat beberapa metode *stemming* untuk Bahasa Indonesia seperti *Nazief* & *Andriani* dan *porter*. *Nazief* & *Andriani* memiliki akurasi lebih tinggi dari Porter (Wahyudi, 2017). Namun,

algoritma *Nazief & Andriani* memiliki masalah stemming seperti (Tahitoe, 2011):

1. kesalahan pemenggalan akhiran “an”, “-kan”, dan “-ku”.
2. Kurang relevannya aturan berformat “meng+kata dasar” dan “peng+kata dasar”.
3. Kurangnya pemeggalan untuk kata-kata berformat “mem+p...” dan “peng+k....”
4. Overstemming dan Understemming.

Menurut penelitian andita et al (Tahitoe, 2011), permasalahan Algoritma *Nazief & Andriani* dapat diperbaiki dengan merubah tabel aturan dan menggunakan *algoritma connected component*. Algoritma stemming baru ini dinamakan Modifikasi Enhanced Confix Stripping. Berikut ini adalah tahapan algoritma modifikasi *ECS:*

1. Lakukan Proses stemming pada term-term menggunakan algoritma *ECS* dan dimpan kemungkinan-kemungkinan hasil stem yang ada.
2. Lakukan pencarian nilai *em* dengan melakukan *pairing* atau pemasangan term yang bermasalah dengan tiap anggota term dari tiap kelas stem. Namun, proses pairing ini ada satu catatan, yakni proses *pairing* dilakukan hanya dengan term-term yang benar-benar hanya berada pada kelas stem tersebut (hanya menghasilkan 1 buah hasil stem). Dapatkan nilai *em* tertinggi dari tiap kelas stem. Setelah didapatkan nilai *em* tertinggi, lakukan perbandingan nilai *em* dari term yang bermasalah tersebut pada tiap kelas stem terhadap kelas stem yang lain. Kelas *stem* yang memiliki nilai *em* tertinggi akan ditetapkan sebagai kelas *stem* untuk term yang bermasalah.

Algoritma *ECS* merupakan algoritma pengembangan *Nazief & Andriani*. Algoritma *ECS* memiliki beberapa pembaruan dalam tabel aturan dan tahap pengembalian. Berikut ini adalah enam tahap algoritma *ECS:*

1. Kata yang belum di-*stemming* dicari pada kamus, jika ditemukan, kata tersebut dianggap sebagai kata dasar yang benar dan algoritma dihentikan.
2. Hilangkan *Inflectional* suffixes, yaitu dengan menghilangkan *particle*(“-lah”, ”-kah”, “tah” atau “-pun”), kemudian hilangkan *inflectional possessive pronoun suffixes* (“-ku”, “-mu” atau ”-nya”). Cek kata di dalam kamus kata dasar, jika ditemukan, algortima dihentikan, jika tidak lanjut ke langkah 3.
3. Hapus Derivational Suffix (“-i” atau ”-an”,”). Jika kata ditemukan dalam kamus kata dasar, maka algoritma berhenti. Jika tidak, maka lanjut ke langkah 3a:
    a. Jika akhiran “-an” telah dihapus dan huruf terakhir dari kata tersebut adalah “-k”, maka “-k” juga dihapus. Jika kata tersebut ditemukan dalam kamus maka algoritma berhenti. Jika tidak ditemukan maka lakukan langkah 3b.
    b. Akhiran yang dihapus (“-i”, “- an” atau “-kan”) dikembalikan, lanjut ke langkah 4.
4. Hapus Derivational Prefix (“be-”,”di-”,”ke-”,”me-”,”pe-“,”se-” dan “te-“). Jika kata yang didapat ditemukan di dalam database kata dasar, maka proses dihentikan, jika tidak, maka lakukan *recoding*. Tahapan ini dihentikan jika memenuhi beberapa kondisi berikut:
    a. Terdapat kombinasi awalan dan akhiran yang tidak diizinkan
    b. Awalan yang dideteksi sama dengan awalan yang dihilangkan sebelumnya.
    c. Tiga awalan telah dihilangkan
5. Jika setelah penghapusan awalan rekursif kata tersebut masih belum ditemukan, dilakukan pemeriksaan apakah pengodean ulang dimungkinkan dengan memeriksa kolom terakhir di Tabel 2.4. Kolom ini menunjukkan varian awalan dan karakter pengodean ulang untuk digunakan ketika kata *root* dimulai dengan huruf tertentu atau ketika label pertama dari kata *root* diakhiri dengan

huruf atau fragmen tertentu. Tidak semua awalan memiliki karakter pengodean berulang contohnya kata "menangkap". Kata ini harus mencari root memenuhi Aturan 15 untuk awalan "men-" (awalan awal "men-" diikuti oleh vokal "a-"). Setelah menghapus "men-" seperti pada Langkah 4 kita memperoleh "angkap," yang bukan kata kunci yang valid. Untuk Aturan 15, ada dua karakter pengodean ulang yang mungkin, "n" (seperti dalam "men-nV ") dan " t "(seperti dalam" men-tV ... "). Algoritma ini mengawali “n” ke “angkap” untuk mendapatkan “nangkap” dan kembali ke Langkah 4. Karena ini bukan kata dasar yang valid, “t” digantikan terlebih dahulu untuk mendapatkan “tangkap” dan algoritma kembali ke Langkah 4. Karena “tangkap ” adalah kata dasar yang valid, pemrosesan berhenti.

Tabel 2.3 Kombinasi Awalan yang Tidak Dibolehkan

Awalan	**Akhiran yang tidak diizinkan**
be-	-i
di-	-an
ke-	-i, -kan
me-	-an
se-	-i, -kan

Tabel 2.4 Aturan Pemenggalan Kompleks

Aturan	**Format kata**	**Pemenggalan**
1	berV…	ber-V … | be-rV
2	berCAP…	ber-CAP… dimana C != ‘r’ & P != ‘er’
3	berCAerV	ber-CaerV… dimana C != ‘r’
4	Belajar	bel-ajar
5	berC_1erC_2…	be-C1erC2… dimana C1 != ‘r’ | ‘l’
6	terV…	ter-V… | te-rV…
7	terCerV…	ter-CerV... dimana C!= ‘r’
8	terCP...	ter-CP... dimana C!=‟r‟ dan P! ‘e’
8	terCer…	ter-Cer dimana C != ‘r’
9	teC_1erC_2…	Te-C_1erC_2… dimana C_1 != ‘r’

Aturan	Format kata	Pemenggalan
10	me{l|r|w|y}V…	me – {l|r|w|y} V…
11	mem{b|f|v}…	mem-{b|f|v}…
12	Mempe	mem-pe…
13	mem{rV|V}…	me-m{rV|V}… | me-p{rV|V}
14	men{c|d|j|s|z}…	men-{c|d|j|s|z}…
15	menV…	me-nV… | me-tV
16	meng{g|h|q|k}…	meng-{g|h|q|k}…
17	mengV…	meng-V… | meng-kV… | (mengV-... jika V= ‘e’)
18	menyV…	meny-sV….
19	mempA...	Mem-pA... dimana A!=’e’...
20	pe{w|y}V…	pe-{w|y}V…
21	perV…	per-V… | pe-rV…
22	perCAP…	per-CAP… dimana C != ‘r’ dan P != ‘er’
23	perCAerV…	per-CAerV… dimana C != ‘r’
24	pem{b|f|V}…	pem-{b|f|V}…
25	pem{rV|V}…	pe-m{rV|V}… | pe-p{rV|V}…
26	pen{c|d|j|z}…	pen-{c|d|j|z}…
27	penV…	pe-nV… | pe-tV…
28	Peng{g|h|q}	Peng-{g|h|q}
29	pengC…	peng-C… | peng-kV… | pengV-... jika V=’e’
30	pengV…	peng-V… | peng-kV… | pengV-... jika V=’e’
31	penyV...	Peny-sV...
32	pelV…	pe-lV… kecuali “pelajar” yang menghasilkan “ajar”
33	peCerV…	Per-erV … dimana C!= {r|w|y|l|m|n}
34	peCP	Pe-CP… dimana C!={r|w|y|l|m|n}dan P!= ‘er’
35	terC_1erC_2...	ter-C_1erC_2... dimana C_1!= ‘r’
36	peC_1erC_2...	Pe-C_1erC_2... dimana C_1!={r|w|y|l|m|n}

(Sumber: Tahitoe, 2011)

6. Jika semua langkah telah dilakukan tetapi kata dasar tersebut tidak ditemukan pada kamus, maka lakukan algoritma *loopPengembalianAkhiran*.

Algoritma *loopPengembalianAkhiran* bertujuan untuk mengatasi masalah *suffix removal*. Berikut ini adalah empat tahapan algoritma *loopPengembalianAkhiran* (Arifin, 2009):

1. Kembalikan seluruh awalan yang telah dihilangkan sebelumnya, sehingga menghasilkan model kata seperti berikut: [DP+[DP+[DP]]] + Kata Dasar. Pemenggalan awalan dilanjutkan dengan proses pencarian di kamus kemudian dilakukan pada kata yang telah dikembalikan menjadi model tersebut.
2. Kembalikan akhiran sesuai dengan urutan model pada bahasa Indonesia. Ini berarti bahwa pengembalian dimulai dari DS (“-i”, “-kan”, “-an”), lalu PP(“-ku”, “-mu”, “-nya”), dan terakhir adalah P (“-lah”, “-kah”, “-tah”, “-pun”). Untuk setiap pengembalian, lakukan langkah 3) hingga 5) berikut. Khusus untuk akhiran “-kan”, pengembalian pertama dimulai dengan “k”, baru kemudian dilanjutkan dengan “an”.
3. Lakukan pengecekan di kamus kata dasar. Apabila ditemukan, proses dihentikan. Apabila gagal, maka lakukan proses pemenggalan awalan berdasarkan aturan pada Tabel 2.3.
4. Lakukan Recording

2.5.7 *Removing Stopwords*

Penghapusan *stopwords* bertujuan untuk menghilangkan kata-kata umum dan sering yang tidak memiliki pengaruh signifikan dalam kalimat. Contoh kata *stopword* adalah “dan” dan “yaitu” (Hidayatullah, 2017).

2.5.8 *Term Frequency-Inverse Document Frequency Matrix*

Tweet tidak bisa digunakan untuk proses pembelajaran secara langsung. Pemrosesan tweet dapat dibantu dengan menggunakan model *bag-of-words*. Model *bag-of-word* dapat digunakan untuk pemodelan pencarian misi sesuai topik tweet yang diangkat. Model *bag-of-words* merubah rangkaian kata tweet menjadi *sparse*

multidimensional representation. *sparse multidimensional representation* adalah multidimensional matriks yang menjadikan kata/istilah sebagai dimensi/fitur. (Aggarwal, Machine Learning for Text, 2018).

Ari Aulia Hakim telah melakukan penelitian tentang matriks kata dengan menggunakan TF-IDF. Penelitiannya menghasilkan akurasi sistem hingga 99.8 % dengan minimal 96.13%. TF-IDF adalah pembobotan yang menggabungkan antara jumlah kemunculan *term/*kata dengan log dari kebalikan kemungkinan kata yang ditemukan dalam dokumen. TF-IDF(*term frequency-inverse document frequency*) merupa dapat didefinisikan dengan Rumus 2.1.

$$tfidf(t,d,D) = tf(t,d)\ .\ idf(t,D) \quad \text{.....(2.1)}$$

$tf(t,d)$ merupakan frekuensi *term* dalam satu dokumen. $idf(t,D)$ merupakan *inverse document frequency* dan dapat didefinisikan dengan rumus 2.2.

$$idf(t,D) = \log\left(\frac{N}{|\{d \in D : t \in d\}|}\right) \quad \text{.....(2.2)}$$

N pada Rumus 2.2 merupakan banyak dokumen dalam corpus D dan $|\{d \in D : t \in d\}|$ adalah banyak dokumen yang mengandung term t (Hakim, 2014).

2.6 Cluster Analysis

Cluster analysis adalah teknik yang digunakan untuk mempartisi data menjadi suatu grup sesuai dengan kemiripan data. Cluster analysis dapat digunakan untuk mengelompokkan data yang belum berlabel. Cluster analysis dapat digunakan untuk beberapa aplikasi contohnya merangkum data, *customer segmentasion, social network analysis,* dan mencari hubungan antar data. (Aggarwal, Data Mining, 2015). Terdapat beberapa jenis metode dan algoritma cluster analysis seperti yang terdapat pada Tabel 2.5.

Tabel 2.5 Metode Cluster Analysis

Metode	**Algoritma**
Representative-Base Algoritms	K-Means, Kernel K-Means, K-Medians, K-Medoids
Heirarchal Clustering Algoritms	Group-Based Statistic, Bisecting K-Means
Probabilistic Model-Based Algoritms	Relationship of EM to k-means and Other Representative Methods
Grid-Based and Density-Based Algoritms	Grid-Based Method, DBSCAN, DENCLUDE
Graph-Based Algoritms	Graph-Based Algorithms

(Sumber: Aggarwal, Data Mining, 2015, telah diolah)

1. *Representative-Base Algoritms*

   Representative-Base Algoritms adalah metode *clustering* dengan menggunakan jarak untuk mengelompokkan cluster. Algoritma-algoritma ini menggunakan *centroid* untuk mewakili kelompok cluster dan tidak terbentuk hubungan hierarki antara cluster. Tujuan dari metode cluster ini adalah mencari pusat cluster yang tepat sehingga minimal jarak antara *centroid* dan masing-masing data dalam suatu cluster bernilai minimal. Rumus minimal dapat dihitung menggunakan rumus di bawah ini:

   $$O = \sum_{i=1}^{n}[min_j Dist\ (\overline{X_i}, \overline{Y_j})] \quad (2.3)$$

   $\overline{Y_j}$ adalah centroid terdekat dari suatu dataset $\overline{X_i}$. Dengan kata lain, representative-Base mencari jarak vector minimal $\overline{Y_j}$ dan $\overline{X_i}$. Terdapat beberapa jenis Representative-Base seperti K-Medoids, kernel K-Means, K-Medians, dan K-Means. K-Medoids menggunakan dataset sebagai pusat cluster dan K-Means mencari centroid bebas sesuai kebutuhan cluster. *K-Medoid* tidak dapat mencapai kondisi optimal karena secara umum *Representative-Base algoritms* mengasumsikan bahwa perwakilan database tidak diambil secara otomatis (Aggarwal, Data Mining, 2015).

2. *Heirarichal Clustering Algoritms*

   Heirarchal Clustering Algoritms menggunakan hierarki untuk memisahkan dataset. Algoritma ini akan membuat taksonomi cluster. Contoh taksonomi

klusternya adalah text web disusun menggunakan berbagai topik kemudian cluster berikutnya berdasarkan subtopik. *Heirarical Clustering* memerlukan biaya perhitungan yang lebih tinggi dari *Representative-Base Algoritm*s.

3. *Probabilistic Model-Base Algoritms*

   Sebagian besar algoritma *Clustering* adalah algoritma *hard clustering* yang secara deterministik dikelompokkan ke dalam suatu kluster. Algoritma berbasis model probabilistik adalah algoritma *soft clustering* yang setiap titik data dapat memiliki probabilitas penetapan tidak nol untuk sebagian besar dataset. Algoritma *soft clustering* dapat dikonversi menjadi ke Algoritma *hard clustering* dengan menetapkan titik data ke sebuah kluster yang memiliki kemungkinan terbesar.

4. *Grid-Based and Density-Based Algoritms*

   Salah satu masalah utama dengan metode berbasis jarak dan probabilistik adalah bahwa bentuk cluster yang mendasari sudah didefinisikan secara implisit oleh fungsi jarak yang mendasari atau distribusi probabilitas. Sebagai contoh, algoritma k-means secara implisit mengasumsikan bentuk bola untuk cluster. Begitu pula algoritma EM dengan Gaussian umum mengasumsikan cluster elips. Dalam praktiknya, kluster mungkin sulit untuk dimodelkan dengan bentuk prototipikal yang tersirat oleh fungsi pendistribusian atau distribusi probabilitas. Namun, Grid-Based and Density-Based Algoritms memerlukan *density threshold* yang dapat mempengaruhi jumlah cluster (Aggarwal, Data Mining, 2015).

5. *Graph-Based Algoritms*

   Graph-Base Algoritms adalah algoritma clustering dengan cara membangun graph antar dataset terlebih dahulu kemudian mengelompokkan sesuai dengan bobot edge. Membangun graph antar database dapat menggunakan K-Nearest Neighbour. Edge akan dibangun di antara dua noda jika nilai kedekatannya lebih kecil dari batas ambang dan akan berbentuk edge tidak terarah. Pengelompokan dilakukan dengan memotong edge sehingga suatu cluster akan membentuk *cyclic*. Metode ini memerlukan biaya tinggi untuk diaplikasikan kepada matriks $n * n$ (Aggarwal, Data Mining, 2015).

2.6.1 Algoritma Improved K-Means

Algoritma K-Means adalah algoritma yang digunakan untuk membentuk partisi data sehingga tidak terbentuk hierarki dan memiliki tingkat tinggi yang sama. K-Means merupakan algoritma partisi yang berdasarkan jarak kedekatan antar dataset. Jarak antar dataset didefinisikan dengan rumus 2.4, perhitungan kesamaan didefinisikan dengan rumus *Cosine Similarity* yang sesuai dengan tipe data mayoritas nol. Rumus *Cosine Similarity* dapat dilihat pada 2.5 dan jarak antara dataset ke koleksi didefinisikan dengan rumus 2.6.

$$d(x_i, x_j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \ldots + (x_{in} - x_{jm})^2} \quad (2.4)$$

$$s(x_i, x_j) = \frac{x_i \cdot x_j}{||x_i|| . ||x_j||} \quad (2.5)$$

$$d(x_i, C) = \min(d(x_i, x_j), x_j \in C). \quad (2.6)$$

Setelah rumus jarak diketahui, maka rumus rata-rata distance menjadi Rumus 2.7.

$$MeanDist = \frac{1}{C_n^2} * \sum d(x_i, x_j) \quad (2.7)$$

Rata-Rata jarak menjadi patokan kepadatan yang dirumuskan pada Rumus 2.8.

$$Dens(x_i) = \sum_{j=1}^{n} u(MeanDist - d(x_i, x_j)) \quad (2.8)$$

$u(z)$ pada Rumus 2.8 adalah fungsi bernilai 1 (satu) jika z bernilai lebih besar sama dengan 0 (nol) dan bernilai 0 (nol) jika z bernilai lebih kecil dari 0 (nol). Setelah *density* diketahui, maka rumus rata-rata kepadatan / *density* dapat dirumuskan dengan Rumus 2.9.

$$MeanDens(D) = \frac{1}{n} \sum_{i=1}^{n} Dens(x_i) \quad (2.9)$$

K-Means memiliki kelemahan yaitu kualitas partisi K-Means tergantung dengan inisiasi *centroid* awal. K-Means tradisional menggunakan inisiasi *centroid* acak. Algoritma Improved K-Means menggunakan tambahan metode inisiasi *centroid* sehingga memiliki presisi dan *recall* yang lebih tinggi daripada tradisional K-Means

(Xiong, 2016). Perbedaan tahap-tahap inisiasi *centroid* Improved K-Means dan K-Means pada Tabel 2.6.

Tabel 2.6 Tabel Tahapan Inisiasi *Centroid*

Improved K-Means	**Tradisional K-Means**
1. Hitung jarak dan rata-rata jarak antara dua data dalam data set *D* 2. Hitung kepadatan dan rata-rata kepadatan dalam data set *D* 3. Pindahkan dataset yang memiliki kepadatan yang tinggi ke dalam koleksi *A* dan hapus dataset yang memiliki kepadatan rendah 4. Pindah koleksi *A* yang memiliki kepadatan tertinggi ke dalam koleksi *B.* 5. Pindah koleksi *A* yang memiliki jarak terjauh dengan koleksi *B.* 6. Ulangi tahap 5 sampai terdapat *k* data dalam koleksi *B.*	Memilih *centroid* acak dari dataset sesuai jumlah *k*

(Sumber: Xiong, 2016, telah diolah)

Improved K-Means memiliki optimalisasi yang sama dengan tradisional K-Means. Optimalisasi Improved K-Means bertujuan untuk mengurangi jarak antara pusat cluster dengan cara merubah *centroid* lama menjadi *centroid* yang sesuai kebutuhan. Algoritma optimalisasi Improved K-Means secara umum dapat dilakukan dengan cara berikut (Xiong, 2016):

1. Menghitung jarak antara setiap dataset dan setiap centroid menggunakan *Cosine Similarity*. Kemudian memasukkan dataset ke dalam suatu cluster berdasarkan jarak terdekat dataset.
2. Menghitung rata-rata nilai dari objek pada suatu cluster.
3. Merubah centroid lama menjadi rata-rata nilai dari objek suatu cluster.
4. Ulangi dari langkah pertama sampai tidak ada perubahan centroid.

2.6.2 Validation

Validasi cluster menggunakan evaluasi internal karena cluster K-Means didefinisikan sebagai *unsupervised*. Terdapat beberapa teknik validasi internal seperti *Sum of Square distances to centroid*, *intracluster to interclaster distance ratio*, *Silhouette coefficient*, dan *probabilistic measure*. karakteristik masing-masing teknik validasi ditampilkan dalam Tabel 2.7.

Tabel 2.7 Teknik Validasi

Teknik	**Karakteristik**
Sum of Square distance to centroid	Berbasis jarak namun tidak memberikan informasi tentang kualitas cluster
Intracluster to intercluster distance ratio	Memperhitungkan jarak dalam satu cluster dan jarak dengan cluster lain
Silhouette coefficient	Memperhitungkan percampuran dataset antar cluster dan memiliki nilai absolut -1 sampai 1
Probabilistic measure	Memperhitungkan parameter yang berhubungan dengan *EM algorithms* dan bersifat probabilistik

(Sumber: Aggarwal, Data Mining, 2015, telah diolah)

Perhitungan *Sum of Square distance to centroid* memiliki tingkat komputasi rendah dan menggunakan Rumus 2.9.

$SSE = \sum_{i=1}^{k} \sum d(x, c_i)^2$ .....(2.10)

Sedangkan, *Silhouette coefficient* menggunakan $a(i)$ dan $b(i)$. $a(i)$ adalah rata-rata jarak data i ke setiap dataset dalam cluster tersebut dan $b(i)$ adalah rata-rata jarak data i ke cluster lain terdekat. Dengan mengguakan Rumus 2.9, 2.10, 2.11, dan 2.12.

$a(i) = \frac{1}{|C_i|-1} \sum_{j \in C_i, i \neq j} d(i,j)$ .....(2.11)

dan

$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i,j)$.....(2.12)

$$s(i) = \frac{b(i)-a(i)}{\max\{a(i),b(i)\}}, \text{jika } |C_i| \neq 1 \quad \text{.....(2.13)}$$

$$s(i) = 0, \text{jika } |C_i| = 1 \text{.....(2.14)}$$

Dan nilai *Silhouette coefficient* dalam suatu cluster dapat ditentukan dengan menghitung rata-rata *Silhouette coefficient* (Aggarwal, Data Mining, 2015).

2.7 Penelitian Terdahulu

Penelitian terdahulu berisikan penelitian-penelitian yang digunakan sebelum penelitian ini dilakukan. Twitter sudah digunakan objek penelitian oleh beberapa peneliti karena struktur katanya yang unik dan cocok dikaji. Pada tahun 2014, Hanf melakukan penelitian politik tentang kepresidenan untuk mengetahui tingkat kepemimpinan calon presiden saat itu. Data yang digunakan pada penelititan ini menggunakan data tweet. serta pendekatan analsis pada penelitian ini menggunakan analsis kualitatif (Hasf, 2017). Kemudian pada tahun tahun 2017, Hidayatullah mengidentifikasi cara *preprocessing* tweet Bahasa Indonesia (Hidayatullah, 2017). Penelitian ini menghasilkan hasil *preprocessing* yang bagus dengan satu kekurangan yaitu kata tidak normal masih dapat lolos *preprocessing* karena normalisasi pada penelitiannya masih menggunakan normalisasi manual. Untuk mengatasi masalah tersebut, penulis mengganti normalisasi manual penelitian Hidayatullah dengan mencari kata terdekat menggunakan teknik *word embedding* (Tomas Mikolov, 2013). Setelah *preprocessing* dilakukan, peneliti menggunakan pendekatan TF-IDF(*Term Frequency-Invert Document Frequency*) untuk menganalisis tweet. TF-IDF memiliki akurasi yang tinggi yaitu 96.13% dalam studi kasus Bahasa Indonesia (Hakim, 2014). Setelah itu, *cluster analysis* dilakukan untuk menganalisis cluster. Cluster yang digunakan adalah Improved K-Means karena memiliki presisi dan *recall* yang lebih tinggi dari tradisional K-Means (Xiong, 2016).