SKRIPSI

KLASTERISASI DAN GEOVISUALISASI *TWEET* PENYEBARAN PENYAKIT MENULAR LANGSUNG (STUDI KASUS COVID-19)

FAHMIRULLAH ABDILLAH

PROGRAM STUDI S1 SISTEM INFORMASI

DEPARTEMEN MATEMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS AIRLANGGA

2023

SKRIPSI

KLASTERISASI DAN GEOVISUALISASI *TWEET* PENYEBARAN PENYAKIT MENULAR LANGSUNG (STUDI KASUS COVID-19)

FAHMIRULLAH ABDILLAH

081811633002

PROGRAM STUDI S1 SISTEM INFORMASI

DEPARTEMEN MATEMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS AIRLANGGA

2023

KLASTERISASI DAN GEOVISUALISASI *TWEET* PENYEBARAN PENYAKIT MENULAR LANGSUNG (STUDI KASUS COVID-19)

SKRIPSI

Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Komputer Bidang Sistem Informasi Pada Fakultas Sains Dan Teknologi Universitas Airlangga

Oleh :

Fahmirullah Abdillah
NIM : 081811633002

Disetujui Oleh :

Pembimbing I,	Pembimbing II,
Ira Puspitasari, S.T., M.T., Ph.D.	Drs. Eto Wuryanto, DEA.
NIP. 198410272010122005	NIP. 196609281991021001

LEMBAR PENGESAHAN NASKAH SKRIPSI

Judul	:	Klasterisasi dan Geovisualisasi *Tweet* Penyebaran Penyakit Menular Langsung (Studi Kasus COVID-19)
Penyusun	:	Fahmirullah Abdillah
NIM	:	081811633002
Pembimbing I	:	Ira Puspitasari, S.T., M.T., Ph.D..
Pembimbing II	:	Drs. Eto Wuryanto, DEA
Tanggal Seminar	:	31 Oktober 2023

Disetujui Oleh,

Pembimbing I,	PembimbingII,
Ira Puspitasari, S.T., M.T., Ph.D.	Drs. Eto Wuryanto, DEA.
NIP. 198410272010122005	NIP. 196609281991021001

Mengetahui,

Ketua Departemen Matematika Fakultas Sains dan Teknologi Universitas Airlangga	Koordinator Program Studi S1 Sistem Informasi Fakultas Sains dan Teknologi Universitas Airlangga
Dr. Herry Suprajitno, S.Si, M.Si.	Dr. Rimuljo Hendradi, S.Si., M.Si.
NIP. 196804041994031020	NIP. 197102111997021001

SURAT PERNYATAAN TENTANG ORISINALITAS

Yang bertanda tangan di bawah ini, saya:

Nama	:	Fahmirullah Abdillah
NIM	:	081811633002
Program Studi	:	Sistem Informasi
Fakultas	:	Sains dan Teknologi
Jenjang	:	Sarjana (S1)

Menyatakan bahwa saya tidak melakukan kegiatan plagiarisme dalam penelitian skripsi saya yang berjudul:

"Klasterisasi dan Geovisualisasi *Tweet* Penyebaran Penyakit Menular Langsung (Studi Kasus COVID-19)"

Apabila suatu saat nanti terbukti melakukan tindakan plagiarisme, maka saya akan menerima sanksi yang telah ditetapkan.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya.

Surabaya, 3 Juli 2023

Fahmirullah Abdillah

NIM. 081811633002

PEDOMAN PENGGUNAAN SKRIPSI

Skripsi ini tidak dipublikasikan, namun tersedia di perpustakaan dalam lingkungan Universitas Airlanga, diperkenankan untuk dipakai sebagai referensi kepustakaan, tetapi pengutipan harus seizin penyusun dan harus menyebutkan sumbernya sesuai kebiasaan ilmiah.

Dokumen skripsi ini merupakan hak milik Universitas Airlangga

KATA PENGANTAR

Puji syukur penulis ucapkan atas kehadirat Tuhan Yang Maha Esa karena telah memberikan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan penyusunan skripsi yang berjudul **Klasterisasi dan Geovisualisasi *Tweet* Penyebaran Penyakit Menular Langsung (Studi Kasus COVID-19)**.

Ucapan terima kasih dari penulis kepada seluruh pihak yang telah membantu dan mendukung dalam pengerjaan skripsi ini hingga dapat terselesaikan dengan baik. Skripsi ini tidak akan terselesaikan tanpa bantuan dan dukungan dari seluruh pihak yang terlibat. Ucapan tersebut ditujukan oleh penyusun kepada:

1. Allah SWT yang senantiasa meberikan segala rahmat, hidayah, dan karunia-Nya serta Rasulullah SAW yang selalu menjadi panutan dan suri tauladan terbaik bagi penulis sehingga penulisan skripsi ini dapat terselesaikan.
2. Ibu Retno Setiasih dan Bapak Abdul Kodir selaku orang tua penulis yang senantiasa memberikan doa dan dukungan dalam bentuk apa pun sehingga penulis dapat menyelesaikan penulisan skripsi ini.
3. Mas Faizal dan Mbak Richa selaku kakak dan kakak ipar penulis yang senantiasa memberikan doa dan dukungan teknis maupun moril sehingga penulis dapat menyelesaikan penulisan skripsi ini.
4. Ira Puspitasari, S.T., M.T., Ph.D., selaku dosen wali sekaligus dosen pembimbing I yang senantiasa sabar membimbing, membantu, menyarankan, dan memberi ilmu selama proses penulisan skripsi ini dimulai hingga selesai.
5. Drs. Eto Wuryanto, DEA, selaku dosen pembimbing II yang senantiasa membimbing, membantu, menyarankan, dan memberi ilmu selama proses penulisan skripsi ini dimulai hingga selesai.
6. Dr. Rimuljo Hendradi, S.Si., M.Si., selaku Koprodi Sistem Informasi dan segenap dosen dan karyawan Prodi yang senantiasa memberikan dukungan moral dan teknis dalam membimbing penulis selama masa perkuliahan hingga skripsi ini selesai.

7. Dara Ninggar selaku *support system* penulis yang selalu bersedia memberikan doa dan dukungan moral selama proses penulisan skripsi ini hingga selesai.
8. Nanda, Hega, Dinda, Fira, Tata, Aldyth, Takbir, Azriel, dan Robby selaku sahabat-sahabat penulis yang bersedia menemani dan memberikan dukungan moral selama proses penulisan skripsi dimulai hingga selesai.
9. Reza, Hilmi, Nady, Arva, Hisyam, Boy, Ico, Basuki, Kinara, Rizki, Farhat, Fadhil, Arya, Arian, Dimas, dan Avril selaku teman-teman "W" yang senantiasa menemani penulis serta berbagi ilmu dan cerita selama masa perkuliahan hingga skripsi ini selesai.
10. Badrus, Fajrul, Niki, Melly, Sasti, Derre, Shadi, Ajeng dan lainnya selaku teman-teman organisasi penulis di BPH BEM FST 2021 yang senantiasa mendukung dan menjadi tempat berbagi cerita selama berkehidupan di kampus.
11. Mas Robit, Mas Affan, Mas Hismoyo, dan Mas Ilham selaku kakak tingkat penulis yang senantiasa memberi solusi dan dukungan saat penyusunan skripsi ini dimulai hingga selesai.
12. Pak Aji, Pak Lutvi, Mas Yahya, Mas Andri, Firman, dan segenap karyawan PTPN 12 yang senantiasa mendukung dan mendoakan ketika masa magang dan penyusunan skripsi.

Penulis menyadari bahwa dalam penyusunan skripsi ini terdapat banyak kekurangan, maka dari itu penulis senantiasa terbuka dalam menerima kritik dan saran atas kekurangan dan kesalahan yang ada dalam penelitian ini. Harapan dari penulis, semoga penelitian ini dapat memberikan manfaat dan wawasan yang berguna dan menjadi sumber ilmu yang bermanfaat kedepannya.

Surabaya, 3 Juli 2023

Penulis

Fahmirullah Abdillah

Fahmirullah Abdillah 2023, **Klasterisasi dan Geovisualisasi *Tweet* Penyebaran Penyakit Menular Langsung (Studi Kasus COVID-19)**, Skripsi ini dibawah bimbingan Ira Puspitasari, S.T., M.T., Ph.D., dan Drs. Eto Wuryanto, DEA., Program Studi S1 Sistem Informasi. Fakultas Sains dan Teknologi, Universitas Airlangga, Surabaya

ABSTRAK

Layanan media sosial *microblog* seperti Twitter menghasilkan aliran besar dalam penyebaran informasi terhadap suatu kejadian. Maka dari itu media sosial dapat dimanfaatkan sebagai bentuk pengawasan penyebaran penyakit menular langsung, seperti kasus pandemi COVID-19. Analisis klaster digunakan dalam penelitian ini untuk mengelompokkan gejala yang dialami oleh penderita COVID-19 secara daring melalui media sosial. Penelitian ini bertujuan untuk menciptakan hasil klaster yang optimal dalam pengawasan penyebaran penyakit menular langsung. Metode yang digunakan dalam membangun luaran dari penelitian ini adalah metode algoritma pengelompokan berbasis kepadatan *Density-Based Spatial Cluster of Application with Noise* (DBSCAN) dan algoritme augmentasi *Ordering Points to Identify the Clustering Structure* (OPTICS), serta *Silhouette Coefficient* untuk uji validasi luaran klaster yang terbentuk, yang diimplementasikan di bahasa pemrograman Python. Luaran dari penelitian ini adalah hasil klaster yang membentuk terhadap *tweet* terkait penyebaran penyakit menular langsung studi kasus COVID-19. Data *tweet* yang sudah diakuisisi akan dilakukan *preprocessing*, kemudian dilakukan pembobotan *term* yang selanjutnya akan melalui proses analisis metode *cluster*. Hasil klaster dari kedua algoritma diimplementasikan dalam peta geovisualisasi wilayah Indonesia yang menggambarkan titik – titik sampel dokumen *tweet*, isi atribut yang digunakan adalah *username* dan *final_tweet*. Perbedaan klaster dan *noise* dilambangkan pada perbedaan warna titik yang mewakili sampel dokumen. Hasil terbaik dari analisis yang dilakukan kedua metode, didapatkan algoritma OPTICS memberikan hasil optimal dengan parameter *xi score* = 0,05 dan minpts = 10, dan uji validasi *silhouette coefficient* sebesar 0,6508317895. Analisis ini menghasilkan 6 klaster dan *term*, yaitu klaster 1 membentuk *term* sakit kepala, klaster 2 membentuk *term* diare, klaster 3 membentuk *term* pilek, klaster 4 membentuk *term* batuk, klaster 5 membentuk *term* covid, dan klaster 6 membentuk *term* demam.

Kata Kunci: Twitter, *Preprocessing, Clustering*, DBSCAN, OPTICS, *Density-based Algorithm, silhouette coefficient*.

Fahmirullah Abdillah 2023, **Klasterisasi dan Geovisualisasi *Tweet* Penyebaran Penyakit Menular Langsung (Studi Kasus COVID-19)**, Skripsi ini dibawah bimbingan Ira Puspitasari, S.T., M.T., Ph.D., dan Drs. Eto Wuryanto, DEA., Program Studi S1 Sistem Informasi. Fakultas Sains dan Teknologi, Universitas Airlangga, Surabaya

---

ABSTRACT

Microblogging social media services such as Twitter generate a large flow of information dissemination towards an event. Therefore, social media can be used as a form of monitoring the spread of direct infectious diseases, such as the case of the COVID-19 pandemic. Cluster analysis was used in this study to group symptoms experienced by COVID-19 sufferers online through social media. This study aims to create optimal cluster results in monitoring the spread of direct infectious diseases. The methods used in building the output of this research are the density-based clustering algorithm method Density-Based Spatial Cluster of Application with Noise (DBSCAN) and the augmentation algorithm Ordering Points to Identify the Clustering Structure (OPTICS), as well as Silhouette Coefficient for validation tests of the output of the cluster formed, which is implemented in the Python programming language. The output of this study is the result of clusters that form tweets related to the spread of infectious diseases directly COVID-19 case studies. Tweet data that has been acquired will be preprocessed, then weighted terms which will then go through the cluster method analysis process. The cluster results of both algorithms are implemented in a geovisualization map of Indonesia that depicts sample points of tweet documents, the content attributes used are *username* and *final_tweet*. The difference in cluster and noise is denoted by the difference in the color of the point representing the sample document. The best results from the analysis carried out by both methods, the OPTICS algorithm obtained optimal results with parameters *xi score* = 0.05 and *minpts* = 10, and silhouette coefficient validation test of 0.6508317895. This analysis produced 6 clusters and terms, namely cluster 1 formed the term headache, cluster 2 formed the term diarrhea, cluster 3 formed the term cold, cluster 4 formed the term cough, cluster 5 formed the term covid, and cluster 6 formed the term fever.

Keywords: Twitter, *Preprocessing*, *Clustering*, DBSCAN, OPTICS, *Density-based Algorithm, silhouette coefficient*.

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I

PENDAHULUAN

1.1 Latar Belakang

Layanan media sosial mikroblog seperti Twitter menghasilkan aliran besar dalam penyebaran informasi terhadap suatu kejadian. Sumber informasi realtime ini sangat berharga untuk banyak area aplikasi, khususnya untuk deteksi bencana dan skenario respons. Terbukti dengan aliran volume maupun kecepatan tweet saat kejadian berlangsung sangat tinggi dan cepat, sehingga masyarakat yang terdampak maupun petugas profesional sedikit mengalami kesulitan saat pemrosesan informasi (Imran et al., 2013).

Sakaki et al. 2013 menuturkan bahwa, melalui pemantauan tweet dapat dideteksi adanya gempa bumi. Probabilitas yang dihasilkan oleh Japan Meteorology Agency cukup tinggi, yaitu 96% untuk gempa bumi dengan skala richter 3 atau lebih. Situs mikroblog ini dapat digunakan sebagai sistem sensor untuk mendeteksi suatu bencana alam atau kejadian lainnya (Crooks et al. 2013).

Beberapa penelitian yang menggunakan data dari media sosial Twitter telah dilakukan sebelumnya. Dwiarni (2019) melakukan penelitian tentang akuisisi dan klasterisasi data teks Twitter untuk memperoleh dasar pengetahuan terhadap profil pengguna Twitter. Penelitian dilakukan dengan ujicoba keyword “K-Pop” dan “K-Drama”. Dari hasil ujicoba akuisisi data didapatkan sebanyak 68.393 tweet. Hasil tersebut disebar menjadi 3 klaster / k=3, yang mana klaster pertama adalah waktu tweet dianggap pada pagi hari, klaster kedua adalah waktu tweet dianggap pada siang hari, dan klaster ketiga adalah waktu tweet dianggap pada malam hari. Kemudian, hasil klasterisasi didapat jam 21.00 - 01.00 merupakan mayoritas orang-orang melakukan tweet. Dari hasil penelitian ini kita dapatkan bahwa penentuan nilai k untuk memperkirakan topik suatu klaster didasarkan pada asumsi kebiasaan pengguna dalam menggunakan media sosial Twitter.

Penelitian lainnya tentang kemungkinan analisis secara realtime pada media sosial dan otomatis dari pesan Twitter selama terjadinya situasi darurat dikemukakan oleh Terpstra et al. (2012). Analisis dilakukan menggunakan tool ekstraksi informasi yang berhasil mendapatkan 97.000 tweet yang dikirim sebelum, saat, dan setelah kejadian alam (badai) terjadi. Lokasi kejadian adalah di Belgia saat berlangsungnya festival Pukkelpop di tahun 2011. Tool ekstraksi dapat menganalisis tweet melalui tampilan geografis, jenis isi pesan (kerusakan, korban), dan jenis tweet (seperti retweet).

Penyakit menular langsung merupakan suatu infeksi yang disebabkan oleh mikroorganisme, seperti virus, parasit, atau jamur. Infeksi ini dapat berpindah dari orang yang sakit ke orang yang sehat. Bentuk penularannya bisa terjadi secara langsung maupun tidak langsung, penularan secara langsung terjadi ketika benda tak kasat mata di atas pada orang yang sakit berpindah melalui kontak fisik, misalnya lewat sentuhan (Alodokter, 2018).

Saat ini penyakit menular langsung telah menjadi wabah yakni virus Covid-19. Wabah yang terjadi secara mendunia ini diberi nama Coronavirus Disease 2019 (Covid-19) yang disebabkan oleh Severe Acute Respiratory Syndrome Coronavirus-2 (SARS-CoV-2). Penyebaran penyakit menular langsung ini hingga ke seluruh penjuru nusantara dan dunia. Menurut Susilo (2020), virus ini dapat ditularkan dari manusia ke manusia dan telah menyebar secara luas di China (sebagai tempat kemunculan pertama) dan lebih dari 190 negara dan teritori lainnya. Pada 12 Maret 2020, WHO mengumumkan COVID-19 sebagai pandemi. Hingga tanggal 29 Maret 2020, terdapat 634.835 kasus dan 33.106 jumlah kematian di seluruh dunia. Sementara di Indonesia sudah ditetapkan 1.528 kasus dengan positif COVID-19 dan 136 kasus kematian. Per tanggal 20 Desember 2020, Satgas Covid-19 menerbitkan laporan yang berisi informasi kasus terkonfirmasi positif, sembuh, ataupun meninggal. Sebanyak 735.124 kasus terkonfirmasi positif dan 19.880 (2,99%) jumlah kematian di Indonesia, serta jumlah kasus sembuh 541.811 (81,48%).

Metode *Density-Based Spatial Cluster of Application with Noise* (DBSCAN) merupakan salah satu metode cluster mengacu pada densitas atau kepadatan. Kepadatan yang dimaksudkan yaitu dalam metode DBSCAN mengelompokkan wilayah dengan jarak yang telah ditentukan menggunakan nilai parameter Epsilon dan MinPts, sehingga dihasilkan suatu kelompok yang padat dengan jarak antar anggota kelompok yang beragam. Parameter Epsilon merupakan jarak maksimal antar titik pusat dengan titik anggota dalam suatu cluster. Sedangkan MinPts merupakan minimal anggota yang harus terpenuhi dalam sebuah klaster. Apabila kedua parameter tersebut telah terpenuhi, maka akan terbentuklah suatu klaster (Daszykowski & Walczak, 2009).

Analisis *cluster* merupakan teknik multivariat dalam analisis statistik yang dapat mengumpulkan objek-objek dengan karakteristik sama pada suatu kelompok yang lebih kecil. Pada penelitian ini metode klasterisasi tweet yang digunakan adalah algoritma DBSCAN dan OPTICS. Metode-metode ini dipilih dan dibandingkan karena keduanya dapat menghasilkan cluster tanpa penentuan centroids dan juga dapat menemukan titik-titik yang menyimpang. Data hasil klasterisasi divisualisasikan untuk menerapkan geovisualisasi tweet untuk kasus penyebaran penyakit menular langsung (studi kasus Covid-19). Proses geovisualisasi digunakan untuk mendapatkan hasil tampilan data tweet hasil klasterisasi dan lokasi penyebaran tweet terkait penyebaran penyakit menular langsung (studi kasus Covid-19). Pengujian dilakukan dengan mengevaluasi hasil analisis klasterisasi menggunakan analisis silhouette yang berdasarkan nilai *silhouette coefficient*.

1.2 Rumusan Masalah

Rumusan masalah pada penelitian ini adalah sebagai berikut:

1. Bagaimana menerapkan algoritma klasterisasi DBSCAN dan OPTICS untuk mengolah data *tweet*?

2. Bagaimana perbandingan algoritma klasterisasi DBSCAN dan OPTICS agar menghasilkan analisis terbaik?
3. Bagaimana menerapkan geovisualisasi hasil klasterisasi data *tweet* untuk kasus penyebaran penyakit menular langsung (studi kasus Covid-19)?

1.3 Tujuan Penelitian

Tujuan penelitian ini adalah:

a. Menerapkan algoritma klasterisasi DBSCAN dan OPTICS untuk mengolah data *tweet.*
b. Mengetahui hasil perbandingan terbaik antara dua algoritma klasterisasi yang digunakan, yaitu DBSCAN dan OPTICS.
c. Menerapkan geovisualisasi hasil klasterisasi data *tweet* untuk kasus penyebaran penyakit menular langsung (studi kasus Covid-19).

1.4 Manfaat

Adapun manfaat penelitian ini adalah:

a. Memberikan wawasan kepada masyarakat awam, peneliti, dan pemerintah terkait data penyebaran penyakit menular langsung (Covid-19).
b. Membantu pemerintah dalam memantau keluhan gejala yang dialami masyarakat saat penyebaran penyakit menular langsung dalam waktu tertentu.
c. Membantu pemerintah dalam mengambil keputusan untuk menindaklanjuti kebijakan berdasarkan data.
d. Membantu pemerintah dalam memutuskan daerah mana saja yang perlu diantisipasi penanggulangan dan pencegahan dini pada penyakit menular langsung.

1.5 Batasan Masalah

Batasan penelitian ini adalah:

a. Penelitian ini menggunakan data teks dari media sosial Twitter dengan kata kunci tentang penyebaran penyakit menular langsung (studi kasus Covid-19).
b. Data teks yang digunakan yaitu *tweet* berbahasa Indonesia.
c. Data yang digunakan dalam klasterisasi adalah data yang dikumpulkan sejak April 2021 – September 2021 dan Januari 2022 – Maret 2022.
d. Output dari algoritma klasterisasi DBSCAN dan OPTICS yang ditampilkan adalah sistem informasi geografis yang menampilkan sebaran data di wilayah Indonesia.
e. Target pengguna dari penelitian ini adalah masyarakat awam, peneliti, atau pemerintah yang ingin mengetahui sebaran data masyarakat terhadap pandemi Covid-19 saat varian delta terjadi.

BAB II

TINJAUAN PUSTAKA

2.1 Twitter API

Application Programming Interface merupakan interaksi online yang melibatkan komponen perangkat lunak. API sudah banyak digunakan, mulai dari command-line tool, aplikasi enterprise, hingga aplikasi web. Twitter API merupakan API JavaScript Object Notation (JSON) berbasis web yang dapat digunakan pengembang untuk berinteraksi dengan data Twitter melalui suatu program. Twitter API harus diakses dengan cara membuat request ke layanan yang disediakan oleh Twitter melalui internet. Dengan API berbasis web, seperti Twitter API, aplikasi akan mengirim request HyperText Transfer Protocol (HTTP), sama seperti web browser, namun response tidak ditampilkan sebagai halaman web, melainkan dengan format yang dapat dipisahkan dengan mudah oleh aplikasi. Response memiliki format yang bermacam-macam. Twitter menggunakan format yang terkenal dan mudah digunakan yaitu JSON. Salah satu bagian dasar dari Twitter adalah tweet. Twitter API dapat digunakan untuk melakukan pencarian tweet, membuat tweet, dan menandai tweet yang disukai. Ketika akan melakukan pencarian tweet, diperlukan untuk memasukkan kriteria, seperti kata kunci atau hashtag, geolokasi, bahasa, dan lain-lain (Freeman, 2018).

Twitter API merupakan contoh dari REST API, yaitu API yang menggunakan gaya arsitektur Representational State Transfer (REST). REST adalah gaya dalam mengembangkan sistem yang dapat melakukan komunikasi yang fleksibel dan menampilkan informasi lintas web dengan menyediakan struktur yang diperlukan untuk mengembangkan komponen yang memiliki tujuan umum secara mudah.

2.2 Data Mining

Data mining adalah proses penemuan pola dan pengetahuan dari kumpulan data dengan jumlah yang besar. Sumber data meliputi basis data, data warehouse, website, penyimpanan informasi lainnya, atau data streaming yang digunakan oleh suatu sistem secara dinamis (Han *et al.* 2012). Sedangkan menurut Baumgartner *et al.* (1996), *Data mining* adalah langkah analisis terhadap proses penemuan pengetahuan di dalam basis data atau *Knowledge Discovery in Databases* (KDD). Pengetahuan dapat berupa pola data yang valid atau hubungan antar data (tidak diketahui sebelumnya). *Data mining* adalah kombinasi dari banyak disiplin ilmu komputer. Disiplin ini didefinisikan sebagai proses menemukan pola baru dari kumpulan data yang sangat besar, termasuk metode seperti kecerdasan buatan, pembelajaran mesin, statistik, dan sistem basis data.

Sedangkan Chakrabarti *et al.* (2006) menjelaskan bahwa, *Data mining* digunakan untuk mengekstrak (mengambil intisari) pengetahuan dari sekumpulan data sehingga didapatkan struktur yang dapat dimengerti manusia serta melibatkan basis data dan manajemen data, prapemrosesan data, pertimbangan model dan inferensi ukuran ketertarikan, pertimbangan kompleksitas, pasca-pemrosesan terhadap struktur yang ditemukan, visualisasi, dan pembaruan secara *online.*

2.3 Praproses Data

Text mining adalah salah satu teknik yang dapat digunakan untuk melakukan klasifikasi di mana, penambangan teks adalah variasi penambangan data yang mencoba menemukan pola menarik dari banyak koleksi data tekstual. Menurut Liao et al. (2012), penambangan teks mirip dengan penambangan data, kecuali untuk teknik penambangan data yang dirancang untuk mengerjakan data terstruktur dalam database, tetapi penambangan teks dapat mengerjakan data yang tidak terstruktur atau semi terstruktur seperti dokumen teks lengkap, halaman web kode/skrip, dan lainnya. Terdapat 5 langkah dalam praproses data yaitu tokenizing, normalisasi kata, penghapusan stopword, stemming, dan pembuatan Term Document Matrix (TDM).

Secara umum, tahapan utama dalam penambangan teks terdiri dari tiga bagian utama yaitu pra-pemrosesan teks, pemilihan fitur, dan analitik teks. Pada tahapan

praprosesi teks secara umum adalah tokenisasi, pemfilteran, stemming, penandaan, dan analisis. Tokenisasi adalah langkah untuk memisahkan setiap kata (token) dalam dokumen input. Pemfilteran adalah proses pemilihan untuk kata-kata yang dihasilkan dari proses tokenisasi, dapat dilakukan dengan daftar berhenti atau algoritma daftar kata. Algoritma stop list akan membuang kata-kata yang tidak penting seperti kata ganti, kata keterangan, konjungsi, preposisi, dan pakaian. Sebaliknya, algoritma daftar kata akan menyimpan kata-kata penting.

A. Case Folding

Case Folding adalah proses mengubah semua karakter huruf pada sebuah kalimat menjadi huruf kecil dan menghilangkan karakter yang dianggap tidak valid seperti angka, tanda baca, dan Uniform Resource Locator (URL) (Jumadi et al., 2021). Contoh teks "Pengumuman", "PENGUMUMAN", "Pengumuman.com" atau "pengumuman" akan tetap dibaca sama, yaitu "pengumuman".

B. Tokenizing

Tokenizing adalah proses pemotongan kumpulan teks dalam dokumen input serta dilakukan pembuangan karakter-karakter tertentu, seperti tanda baca.

Token juga dapat disebut sebagai term atau kata, namun terkadang perlu dibedakan antara type/token. Token adalah kumpulan beberapa karakter pada suatu dokumen, sedangkan type merupakan kelas dari semua token yang memiliki urutan karakter yang sama. Menurut Manning et al. (2009), Term merupakan type yang termasuk ke dalam kamus sistem temu kembali informasi.

C. Normalisasi Kata

Normalisasi kata adalah proses pengolahan susunan kata agar didapatkan kata tersebut menjadi baku meskipun terdapat susunan karakter yang berbeda (Manning et al. 2009). Cara paling sederhana dalam membakukan kata tersebut adalah dengan membuat pemetaan kelas berdasarkan kata yang memiliki persamaan. Contohnya, kata "hrus" atau "hrs" diubah menjadi kata baku yaitu "harus".

Untuk membuat normalisasi kata adalah dengan mengelompokkan atau mengklasterisasi kata tersebut. Klasterisasi kata-kata akan menghasilkan kumpulan kata yang sering digunakan dan memiliki konteks yang sama. Kata-kata tersebut digunakan dalam pemetaan kelas dengan kata baku.

D. Penghapusan Stopword

Lo et al. (2005) menjelaskan bahwa stopword adalah kata yang terdapat pada sebuah dokumen yang sering muncul, namun tidak memiliki nilai informasi kata. Beliau juga menjelaskan bahwa banyak yang beranggapan pada stopword ini tidak memiliki peran terhadap konteks atau informasi dokumen dan stopword harus dihapus sebelum dilakukan proses pada sistem temu kembali informasi, meskipun menggunakan daftar stopword tunggal dari berbagai kumpulan dokumen dapat merugikan efektifitas proses pengambilan informasi.

Penghapusan stopword secara komprehensif dapat mengurangi jumlah kata yang harus disimpan oleh sistem (Manning et al. 2009). Contoh stopword dalam Bahasa Indonesia diantaranya yaitu dahulu, ada, dalam, adanya, dan lain-lain.

E. Stemming Nazief-Adriani

Stemming adalah proses penghapusan imbuhan kata untuk mendapatkan kata dasar. Teknik ini sering digunakan dalam penelitian *text mining.* Contoh kata tulisnya, tulisannyakah, dan dituliskannya memiliki kata dasar yang sama yaitu tulis. Teknik ini mengurangi kompleksitas teks tanpa memengaruhi nilai informasi (Jumadi et al., 2021)

Algoritme *stemming* salah satunya adalah Nazief-Adriani. Algoritme ini dikembangkan menggunakan pendekatan pencocokan *term* dengan pencarian kamus.

Algoritma Nazief-Adriani memiliki langkah-langkah sebagai berikut:

1. Kata yang akan dilakukan *stemming* dicari dalam kamus. Jika ditemukan, maka akan dianggap kata tersebut adalah kata dasar dan algoritme berhenti. Jika tidak ditemukan maka lanjut ke langkah 2.
2. Menghilangkan imbuhan infleksi/*inflectional suffixes* (“-lah”, “-kah”, “-pun”), kemudian *possessive pronoun* (“-ku”, “-mu”, dan “-nya”). Kata dicari dalam kamus, jika ditemukan, algoritma berhenti. Jika tidak ditemukan, maka lanjut ke langkah 3.
3. Menghilangkan imbuhan derivasi/*derivation suffixes* (“-an”, “-i”, dan “-kan”). Jika akhiran “-an” dihapus dan ditemukan akhiran “-k”, maka akhiran “-k” dihapus. Jika ditemukan, algoritma berhenti. Jika tidak ditemukan, maka lanjut ke langkah 4.
4. Langkah 4 memiliki 3 iterasi.
    a. Iterasi berhenti jika:

        1. Ditemukannya kombinasi awalan yang tidak diizinkan berdasarkan awalan.

        2. Awalan yang dideteksi saat ini sama dengan awalan yang dihilangkan sebelumnya.

        3. tiga awalan telah dihilangkan.

Tabel 2.1 Kombinasi Awalan yang Tidak Diizinkan

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

    b. Mengidentifikasi tipe awalan dan hilangkan. Awalan terdiri dari dua tipe:

1. Standar (“di-”, “ke-”, dan “se-”) yang dapat langsung dihilangkan dari kata.

2. Kompleks (“me-”, “be-”, “pe-”, “te-”) adalah tipe-tipe awalan yang dapat bermorfologi sesuai kata dasar yang mengikutinya.

c. Mencari kata yang telah dihilangkan awalannya. Jika tidak ditemukan, maka langkah 4 diulang kembali. Jika ditemukan, algoritma berhenti.

5. Apabila setelah langkah 4 kata dasar masih belum ditemukan, maka proses *recoding* dilakukan dengan mengacu pada Tabel 2.2. *Recoding* dilakukan dengan menambahkan karakter di awal kata yang dipenggal. Pada Tabel 2.2, karakter *recoding* adalah huruf kecil setelah tanda hubung (‘-’) dan kadang berada sebelum tanda kurung.
6. Jika semua gagal, maka masukan kata yang diuji pada algoritma ini dianggap sebagai kata dasar.

F. Term Document Matrix

Term Document Matrix (TDM) adalah matriks 2 dimensi yang memiliki baris yang mewakili dokumen dan kolom yang berisi daftar term serta memiliki nilai frekuensi kemunculan suatu term pada suatu dokumen. Term tersebut didapatkan dari hasil proses stemming, kemudian dilakukan pengindeksan. Contohnya, kata “bukunya” dan “bukukan” memiliki indeks nilai yang sama dengan term buku (Manning et al. 2009).

G. Algoritma TF-IDF

Term frequency (TF) dan *Inverse Document Frequency* (IDF) merupakan perhitungan pembobotan dalam frekuensi kemunculan sebuah dokumen tertentu dan *inverse* frekuensi dokumen yang mengandung kata tersebut. Frekuensi kemunculan kata yang menunjukkan seberapa penting kata tersebut dalam kumpulan dokumen (Wahyuni et al., 2017), untuk perhitungannya dapat dilihat pada persamaan 2.1, 2.2, dan 2.3.

$$TF(d,t) = f(d,t) \quad (2.1)$$

Dimana *f*(*d*,*t*) adalah frekuensi kemunculan kata *t* pada dokumen *d*.

$$IDF(t) = log\ (\frac{N}{df(t)}) \quad (2.2)$$

Dimana *df*(*t*) adalah jumlah dokumen yang memiliki kata *t.*

$$TF\ IDF = TF\ (d,t) \times IDF\ (t) \quad (2.3)$$

2.4 Klasterisasi

Klasterisasi adalah proses pengelompokan kumpulan objek data ke klaster yang memiliki kemiripan dan membedakan yang tidak mirip ke dalam klaster lain. Proses ini tidak dilakukan oleh manusia, sedangkan oleh alat algoritma klasterisasi. Maka dari itu, klasterisasi berguna untuk penentuan kelompok data yang tidak diketahui sebelumnya (Han *et al.* 2012).

Metode pengklasteran secara umum dapat dikelompokkan ke dalam tabel sesuai dengan tabel 2.2.

Tabel 2.2 Metode Klasterisasi *(Han* et al. *2012)*

Metode	Karakteristik Umum	Contoh Algoritma
Metode Partisi	1. Mencari klaster eksklusif yang mirip 2. Berbasis jarak 3. Menggunakan rata-rata atau medoid, untuk menggambarkan pusat klaster 4. Efektif untuk kumpulan data kecil hingga menengah	1. K-Means 2. K-Medoids

Metode Hierarki	1. Klasterisasi dekomposisi hierarki 2. Tidak dapat memperbaiki penggabungan dan pemisahan yang salah 3. Menggabungkan teknik lainnya seperti klasterisasi mikro atau keterhubungan objek	1. Klasterisasi Aglomeratif/Divisif 2. Pengukuran jarak 3. *Balanced Iterative Reducing and Clustering (BIRD)* 4. *Chameleon* 5. Klasterisasi hierarki probabilistik
Metode Berbasis Kepadatan	1. Dapat menentukan klaster yang bentuknya berubah-ubah 2. Klaster merupakan wilayah kepadatan objek pada suatu ruang yang dipisahkan oleh wilayah dengan kepadatan yang lebih rendah 3. Kepadatan klaster; Setiap titik harus memiliki jumlah minimal terdekat	1. *Density-Based Spatial Clustering of Application with Noise (DBSCAN)* 2. *Ordering Points to Identify the Clustering Structure* (OPTICS)
Metode Berbasis Grid	1. Menggunakan struktur data *grid* multiresolusi 2. Waktu proses yang cepat (tergantung pada jumlah objek data dan ukuran *grid*)	1. *Statistical Information Grid* (STING) 2. *Clustering in Quest* (CLIQUE)

A. *NearestNeighbors*

NearestNeighbors memiliki prinsip dengan menemukan sejumlah sampel yang telah ditentukan sebelumnya yang jaraknya paling dekat dengan titik baru. Jumlah sampel dapat berupa konstanta yang ditentukan pengguna (*k-nearest neighbor learning*), atau bervariasi berdasarkan kepadatan titik lokal (*radius-based neighbor learning*). Secara umum, jarak dapat berupa ukuran metrik apa pun: jarak Euclidean standar adalah pilihan yang paling umum (Claude Cariou, 2016).

B. Algoritma DBSCAN

DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) adalah metode *unsupervised-learning* yangs populer digunakan dalam pembuatan model dan algoritma pembelajaran mesin. Mengingat bahwa DBSCAN adalah algoritma pengelompokan berbasis kepadatan, metode ini melakukan pekerjaan yang baik untuk mencari area dalam data yang memiliki kepadatan pengamatan yang tinggi, dibandingkan dengan area data yang tidak terlalu padat dengan pengamatan. Keuntungan lainnya adalah penggunaan metode DBSCAN dapat mengelompokkan data ke dalam kelompok dengan berbagai bentuk dan dapat mengidentifikasi *outliers* yang dianggap sebagai *noise*.

Pada DBSCAN, parameter yang digunakan adalah *minPts* (minimum Points) dan *eps* / ε (epsilon) dan kumpulan *dataset* dalam titik *Density-Based* (Capdevila, 2016). Konsep kepadatan pada DBSCAN melahirkan tiga macam status di setiap data, yaitu inti/*core* (titik pusat dalam klaster yang didasarkan pada kepadatan, dimana ada sejumlah titik yang harus berada dalam *Eps*), batas/*border* (titik yang menjadi batasan dalam kawasan titik pusat), dan *noise* (titik yang tidak dapat dijangkau oleh *core* dan bukan merupakan *border*) (Putri et al., 2021).

Metode klasterisasi DBSCAN menemukan klaster-klaster dengan cara (Silitonga, 2016):

a. DBSCAN menelusuri klaster-klaster dengan memeriksa ε-*neighborhood* dari tiap-tiap *point* dalam *database.* Jika ε-*neighborhood* dari point p mengandung lebih dari MinPts, klaster baru dengan p sebagai *core object* diciptakan.

b. Kemudian DBSCAN secara iteratif mengumpulkan secara langsung objek-objek *density reachable* dari *core object* tersebut, dimana mungkin melibatkan penggabungan dari beberapa *cluster-cluster density reachable*.

Secara umum komputasi dari algoritma DBSCAN adalah sebagai berikut (Devi et al., 2015):

a. Inisialisasi parameter *minPts* dan *ε.*
b. Tentukan titik awal atau *p* secara acak.
c. Hitung ε atau semua jarak titik pada *density reachable* terhadap p dengan menggunakan *cosine similarity*.
d. Jika titik yang memenuhi ε lebih dari *minPts* maka titik p adalah *core object* dan klaster terbentuk.
e. Ulangi langkah secara iteratif hingga dilakukan proses pada semua titik.

C. Algoritma OPTICS

OPTICS (*Ordering Points to Identify the Clustering Structure*) adalah algoritma klastering hierarkis yang bergantung pada kepadatan data. OPTICS mampu mendeteksi klaster yang bermakna dalam data dengan kepadatan yang bervariasi dengan menghasilkan urutan titik-titik yang linier, sehingga titik-titik yang terdekat secara spasial menjadi tetangga dalam urutan tersebut (Patwary, 2013). Berikut langkah-langkah penggunaan algoritma OPTICS (Prabahari, 2014):

a. Temukan jarak inti dari suatu objek. *p* adalah nilai eps/ε terkecil sehingga menjadikannya sebagai *core object*. Jika *p* bukan *core object*, jarak inti *p* tidak terdefinisi.

b. Jarak jangkauan objek *q* dengan objek lain *p* adalah nilai yang lebih besar dari *core object p* dan jarak Euclidean antara *p* dan *q*. Jika *p* bukan *core object*, maka jarak jangkauan antara *p* dan *q* tidak terdefinisi.

D. Uji Validasi

Kualitas klaster Ci dapat diukur dengan menggunakan silhouette coefficient. Teknik ini memberikan representasi grafis singkat dari seberapa baik setiap objek terletak pada kelompok. Analisa metode validitas ini dilakukan dengan melihat besar nilai s. Hasil perhitungan nilai indeks validitas silhouette dapat bervariasi antara -1 hingga 1 (Budiman et al., 2016). Silhouette coefficient ini dirumuskan pada persamaan 2.5, Persamaan 2.6, dan Persamaan 2.7.

$$a(\boldsymbol{o}) \frac{\sum_{\boldsymbol{o}' \in C_i, \boldsymbol{o} \neq \boldsymbol{o}'} dist(\boldsymbol{o}, \boldsymbol{o}')}{|C_i| - 1} =, \boldsymbol{o} \in C_i \ (1 \leq i \leq k) \quad (2.5)$$

Setiap objek $\boldsymbol{o} \in$ dataset, perhitungan dilakukan a(o) untuk mencari jarak rata-rata antara o dan semua objek lain pada klaster yang sama.

$$b(\boldsymbol{o}) = \min C_j{:}1 \leq j < k, j \neq i \left\{ \frac{\sum_{\boldsymbol{o}' \in} dist(\boldsymbol{o}, \boldsymbol{o}')}{|C_j|} \right\} \quad (2.6)$$

b(o) adalah jarak rata-rata terkecil dari o terhadap semua objek lain pada klaster yang lain.

$$s(\boldsymbol{o}) = \frac{b(\boldsymbol{o}) - a\,(\mathbf{o})}{\max\{a(\boldsymbol{o}), b(\boldsymbol{o})\}} \quad (2.7)$$

s(***o***) merupakan koefisien silhouette. Koefisien memiliki nilai -1 hingga 1. Jika nilai silhouette semakin mendekati nilai 1, maka semakin baik pengelompokan data dalam suatu klaster. Jika sebaliknya, semakin mendekati nilai -1 maka semakin buruk pengelompokan data dalam klaster.

2.5 Geovisualisasi

Geovisualisasi adalah proses analisis data geospasial di mana visualisasi dilakukan melalui suatu alat dengan konvergensi informasi, kartografi, dan metode

geografi, menurut Yasobant *et al.* (2015). Fungsi spesifik dari teknik ini adalah digunakan dalam menampilkan data geospasial untuk menjelajahi, menganalisis, dan menyatukan data sehingga dapat menghasilkan hipotesis dan mengembangkan solusi, serta representasi data yang komprehensif. Ada 2 jenis pendekatan geovisualisasi, yaitu pendekatan fenomenologikal dan pendekatan positifistik. Pendekatan fenomenologikal digunakan sebagai interpretasi individu atas ruang dan waktu dalam bentuk abstrak. Berbeda dengan pendekatan positifistik, pendekatan ini menggunakan pemodelan spasial untuk mewakili dunia nyata.

A. Sistem Informasi Geografis

Sistem Informasi Geografis merupakan sistem informasi berbasis komputer yang digunakan untuk mengumpulkan, memeriksa, mengintegrasikan, dan menganalisa informasi data yang memiliki dasar penggunaan secara geografis (Koko Mukti Wibowo, Indra Kanedi, 2021). Pada dasarnya, istilah Sistem Informasi Geografis terbagi menjadi tiga kata, yaitu sistem, informasi, dan geografi. Penggunaan kata "geografi" atau "geografis" ini merujuk pada suatu persoalan mengenai bumi; secara permukaan dua dimensi atau tiga dimensi. Istilah "informasi geografis" mengandung pengertian berupa informasi mengenai tempat, pengetahuan posisi suatu objek, dan keterangan-keterangan (atribut) yang terdapat di permukaan bumi yang posisinya diketahui.

2.6 Penelitian Sebelumnya Tentang Penyebaran Informasi Suatu Kejadian Menggunakan Twitter

Dwiarni (2019) melakukan penelitian tentang akuisisi dan klasterisasi data teks Twitter untuk memperoleh dasar pengetahuan terhadap profil pengguna Twitter. Penelitian dilakukan dengan ujicoba keyword "K-Pop" dan "K-Drama". Dari hasil ujicoba akuisisi data didapatkan sebanyak 68.393 tweet. Hasil tersebut disebar menjadi 3 klaster / k=3, yang mana klaster pertama adalah waktu tweet dianggap pada pagi hari, klaster kedua adalah waktu tweet dianggap pada siang hari, dan klaster ketiga adalah waktu tweet dianggap pada malam hari. Kemudian, hasil klasterisasi didapat jam 21.00 - 01.00 merupakan mayoritas orang-orang melakukan tweet. Dari hasil penelitian ini kita dapatkan bahwa penentuan nilai k

untuk memperkirakan topik suatu klaster didasarkan pada asumsi kebiasaan pengguna dalam menggunakan media sosial Twitter.

Terpstra (2012) melakukan penelitian tentang kemungkinan analisis secara real time dan otomatis dari pesan Twitter selama terjadinya situasi darurat. Analisis dilakukan dengan menggunakan tools ekstraksi informasi yang berhasil mendapatkan 97.000 tweet yang dikirim sebelum, saat, dan sesudah badai terjadi pada Festival Pukkelpop 2011 di Belgia. Tool ekstraksi dapat menganalisis tweet melalui tampilan geografis, jenis isi pesan (kerusakan, korban), dan jenis tweet (seperti retweet).

Denatari (2015) melakukan penelitian mengenai klasterisasi data teks Twitter untuk kasus pertanian di Indonesia. Data teks Twitter terbagi menjadi 2 jenis, yaitu data tweet sejumlah 51 data dan data konten Uniform Resource Locator (URL) sejumlah 51 data. Kedua jenis data tersebut dibandingkan dan dikelompokkan dengan algoritma hierarchial clustering untuk mendapatkan klaster terbaik.

(Crooks et al., 2013) penelitian ini dilakukan dengan analisis performa microblogging sebagai sistem sensor untuk mendeteksi kejadian dengan studi kasus gempa bumi yang ada di daerah East Coast, Amerika Serikat. Peneliti mengambil hasil deteksi yang memiliki karakteristik spasial dan temporal dari penyebaran informasi yang ada di situs microblogging (Twitter). Analisis terhadap situs ini juga dilakukan dengan teknik crowdsourcing, karena setiap media sosial atau situs microblogging juga memiliki informasi geografis ketika seorang pengguna mengomentari suatu kejadian yang dialami terjadi di sekitarnya, atau mengenai pusat lokasi yang menjadi pusat perhatian. Namun, perbedaannya media sosial atau situs microblogging tidak menyediakan informasi geografis pengguna secara terang-terangan, berbeda dengan teknik crowdsourcing yang sudah ada pada aplikasi Wikimapia atau OpenStreetMap.

Penelitian ini bertujuan untuk menilai kualitas informasi yang telah diambil dari masyarakat dengan mempertimbangkan reaksi pengguna Twitter terhadap gempa bumi yang terjadi di Virginia, Amerika Serikat pada tanggal 23 Agustus

2011. Hasilnya, tweet dapat digunakan untuk memberi perkiraan yang cepat dan bagus dari wilayah yang terkena dampak gempa bumi. Perkiraan ini digunakan sebagai informasi yang penting untuk penanganan dan pemulihan dampak bencana. Dengan kemampuannya untuk memperkirakan wilayah yang terkena dampak gempa bumi dengan akurat, hal tersebut mendukung pernyataan bahwa dengan mengambil informasi geospasial di Twitter, peneliti memperoleh informasi yang penting mengenai dampak dari suatu kejadian dengan cepat.

(Rahmanti et al., 2021) melakukan penelitian dengan mengidentifikasi informasi tentang resiko dan respon komunikasi masyarakat Indonesia terhadap pemberlakuan *New Normal* ketika pandemi Covid-19 yang ada pada situs *microblogging* (Twitter) di wilayah Indonesia. Penelitian ini bertujuan untuk menggolongkan *tweet* yang memiliki sentimen positif, negatif, dan netral dengan klasifikator *naïve-bayes* dan memasukkannya ke dalam analisis emosi dasar dari *Plutchik's Wheel of Emotions* (*joy, fear, anticipation, anger, disgust, sadness, surprise,* dan *trust*). Penelitian ini dilakukan pada tanggal 21 Mei 2020 – 18 Juni 2020, dengan hasil data sebanyak 282.216 *tweet* dari 137.057 pengguna. *Tweet* itu semua mengandung 88.677 *mention*, 31.452 *reply*, 164.087 *retweet*. Hasil tersebut disebar ke dalam *Plutchik's Wheel of Emotions* dengan persentase; *joy* (9,01%), *fear* (6,50%), *anticipation* (14,82%), *anger* (4,81%), *disgust* (0,73%), *sadness* (1,74%), *surprise* (8,62%), dan *trust* (53,77%). Kemudian, didapatkan hasil penggunaan tiga *hashtag* terbanyak, yaitu *#NewNormal* (17.051 *tweet*), *#TataKehidupanBaru* (10.980 *tweet*), dan *#DisiplinPolaHidupBaru* (5.200 *tweet*). Dari hasil ini peneliti dapat menggolongkan suatu kejadian yang ada di situs *microblogging* (Twitter) ke dalam pemetaan analisis berdasarkan emosi pengguna.

BAB III

METODE PENELITIAN

Metode penelitian merupakan suatu prosedur yang digunakan untuk melakukan penelitian, sehingga mampu menjawab rumusan masalah dan tujuan penelitian dengan landasan ilmiah tertentu.

3.1 Waktu Penelitian

Penelitian ini dilaksanakan selama 6 bulan mulai dari 1 Mei 2022 – 30 Oktober 2022.

3.2 Objek Penelitian

Objek penelitian ini adalah *tweet* terkait penyebaran penyakit menular langsung di Indonesia dengan periode waktu tertentu, yaitu khusus virus Covid-19 varian delta dimulai dari 1 April 2021 – 30 September 2021 dan khusus virus Covid-19 Varian Omicron dimulai dari 1 Januari 2022 – 30 Juni 2022, dengan menggunakan kata kunci. Kata kunci dapat dilihat pada Tabel 3.1.

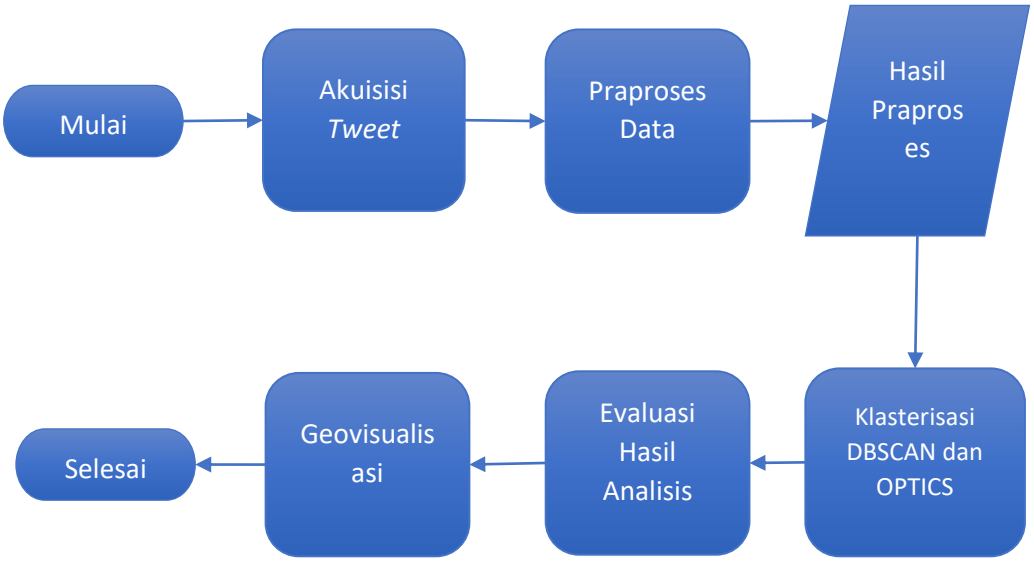
Tabel 3.1 Kata Kunci yang digunakan untuk pencarian *tweet*

No.	Kata Kunci Covid-19	Keterangan
1.	Covid-19	Virus penyakit yang menjadi pandemi dimulai pada tahun 2019
2.	*SARS-CoV-2*	Virus penyebab penyakit Covid-19
3.	Coronavirus	Jenis virus penyakit yang menginfeksi sistem pernapasan
4.	Batuk	Bentuk respons tubuh terhadap infeksi atau iritasi yang terjadi di dalam sistem pernapasan
5.	Batuk Kering	Jenis batuk yang tidak mengandung dahak atau lendir
6.	Pilek	Kondisi saat hidung mengeluarkan lendir yang berlebihan
7.	Kelelahan	Bentuk respons tubuh terhadap virus yang telah menginfeksi sistem imun
8.	Sakit Kepala	Rasa nyeri yang menyerang bagian kepala
9.	Demam	Bentuk respons tubuh terhadap Penyakit

10.	Sesak Napas	Kondisi seseorang mengalami kesusahan dalam bernapas
11.	Varian Omicron Covid-19	Sebuah varian atau jenis penyakit dari virus Covid-19 (*Variants of Concern*) dengan tingkat fatalitas 1,9%
12.	Varian Delta Covid-19	Sebuah varian atau jenis penyakit dari virus Covid-19 (*Variants* of Concern) dengan tingkat fatalitas 3,4%
13.	Sakit Tenggorokan	Rasa nyeri yang menyerang tenggorokan dan susah menelan makanan
14.	Anosmia	Kehilangan kemampuan untuk mendeteksi rasa atau bau
15.	Diare	Buang air besar encer dan berulang

3.3 Tahapan Penelitian

Penelitian ini memiliki beberapa tahapan yang dapat dilihat pada Gambar 3.1 di bawah ini.



Gambar 3.1 Tahapan Penelitian

A. Akuisisi Tweet

Akuisisi *tweet* menggunakan proses *web crawling* dan *web scraping* dengan menggunakan program Python serta Twitter API. Proses *web crawling* dan *web scraping* menggunakan *library snscrape* yang tersedia pada https://github.com/JustAnotherArchivist/snscrape/. *Snscrape* merupakan *tools*

berupa *package library* untuk melakukan interaksi *web browser* secara otomatis yang dikombinasikan dengan program Python. Program Python akan melakukan proses *web crawling* dengan cara membuka halaman pencarian Twitter secara otomatis berdasarkan input tautan yang dimasukkan sebelumnya. Setelah halaman selesai dimuat seluruhnya, dilakukan proses *web scraping* dengan cara mengekstrak data dari halaman hasil pencarian Twitter.

Akuisisi dilakukan dengan memasukkan kata kunci yang telah ditentukan dan rentang waktu selama dua minggu pada tautan pencarian Twitter. Akuisisi dilakukan sebanyak dua kali per bulan mulai bulan April 2021 sampai bulan Juni 2022. Tahap akuisisi *tweet* akan menghasilkan *output* dalam format *Comma Separated Value* (CSV). Metode *web crawling* dan *web scraping* menghasilkan *username, date, time, content,* dan *tweetID.* Twitter API digunakan untuk mendapatkan *placeID,* daerah, lokasi, *longitude,* dan *latitude.*

B. Praproses Data

Praproses data dilakukan untuk pengolahan data yang akan menghasilkan berupa *Term Document Matrix* (TDM) untuk tahap pembobotan *term* dan klasterisasi.

a. *Case Folding*

Case Folding adalah tahapan yang berfungsi mengonversi keseluruhan teks dalam dokumen menjadi huruf kecil, kemudian dilakukan penghapusan mention, URL, dan tanda baca. Contoh data tweet sebelum dan sesudah proses case folding, dapat dilihat pada Tabel 3.2.

Gambar 3.2 Contoh Case Folding

Data awal	Data akhir
dr. Erlina menyarankan agar pasien Covid-19 selalu memantau frekuensi napas agar bisa mengetahui tanda sesak napas - #Sains https://t.co/mxpV2xCBBP	“dr. erlina menyarankan agar pasien covid19 selalu memantau frekuensi napas agar bisa mengetahui tanda sesak napas”

kyk ga sanggup pegang hp lamaÂ².. liat twitter pd nyari oksigen trs kamar RS, di WA juga tiap hr ada aja grup yg ngabarin positif, nyari obat, donar darah plasma, kritis lah sesak napas lah.. gw yg sehat jd berasa ikut sakit.. ðŸ˜µ	“kyk ga sanggup pegang hp lama².. liat twitter pd nyari oksigen trs kamar rs, di wa juga tiap hr ada aja grup yg ngabarin positif, nyari obat, donar darah plasma, kritis lah sesak napas lah.. gw yg sehat jd berasa ikut sakit.. “

b. *Tokenizing*

Tokenizing dilakukan dengan menggunakan *library* Natural Language Toolkit berbasis Python yang tersedia pada https://www.nltk.org/. Data *tweet* diubah menjadi kumpulan data dengan mengubah formatnya menjadi *Comma Separated Values* (CSV). Kumpulan data *tweet* kemudian diubah menjadi *corpus*. *Corpus* merupakan entitas yang secara konseptual mirip dengan basis data dalam penyimpanan dan pengaturan dokumen teks (Feinerer et al., 2008). Semua huruf pada *corpus* telah menjadi huruf kecil. Contoh data *tweet* sebelum dan sesudah proses *tokenizing* dapat dilihat pada Tabel 3.2.

Tabel 3.2 Contoh Tokenizing

Data awal	Data akhir
dr. Erlina menyarankan agar pasien Covid-19 selalu memantau frekuensi napas agar bisa mengetahui tanda sesak napas - #Sains https://t.co/mxpV2xCBBP	“dr”, “erlina”, “menyarankan”, “agar”, “pasien”, “covid19”, “selalu”, “memantau”, “frekuensi”, “napas”, “agar”, “bisa”, “mengetahui”, “tanda”, “sesak”, “napas”
kyk ga sanggup pegang hp lamaÂ².. liat twitter pd nyari oksigen trs kamar RS, di WA juga tiap hr ada aja grup yg ngabarin positif, nyari obat, donar darah plasma, kritis lah sesak napas lah.. gw yg sehat jd berasa ikut sakit.. ðŸ˜µ	“kyk”, “ga”, “sanggup”, “pegang”, “hp”, “lama”, “liat”, “twitter”, “pd”, “nyari”, “oksigen”, “trs”, “kamar”, “RS”, “di”, “WA”, “juga”, “tiap”, “hr”, “ada”, “aja”, “grup”, “yg”, “ngabarin”, “positif”, “nyari”, “obat”, “donar”, “darah”, “plasma”, “kritis”, “lah”, “sesak”, “napas”, “lah”, “gw”, “yg”, “sehat”, “jd”, “berasa”, “ikut”, “sakit”

c. Normalisasi Kata

Normalisasi kata adalah proses pengolahan susunan kata agar didapatkan kata tersebut menjadi baku meskipun terdapat susunan karakter yang berbeda (Manning et al. 2009). Dalam penelusuran kata di media sosial didapatkan banyak kata yang tidak baku. Maka, tahap ini dilakukan normalisasi dengan cara mengumpulkan pada library tertentu dan membuat kode pencocokan kata baku dengan kata yang telah didapatkan pada saat scraping. Kumpulan kata baku ini disimpan pada sebuah file csv. Contoh data tweet sebelum dan sesudah normalisasi kata dapat dilihat pada Tabel 3.3

Tabel 3.3 Contoh Normalisasi Kata

Data awal	Data akhir
"dr", "erlina", "menyarankan", "agar", "pasien", "covid19", "selalu", "memantau", "frekuensi", "napas", "agar", "bisa", "mengetahui", "tanda", "sesak", "napas"	"dr", "erlina", "menyarankan", "pasien", "covid19", "memantau", "frekuensi", "napas", "tanda", "sesak", "napas"
"kyk", "ga", "sanggup", "pegang", "hp", "lama", "liat", "twitter", "pd", "nyari", "oksigen", "trs", "kamar", "RS", "di", "WA", "juga", "tiap", "hr", "ada", "aja", "grup", "yg", "ngabarin", "positif", "nyari", "obat", "donar", "darah", "plasma", "kritis", "lah", "sesak", "napas", "lah", "gw", "yg", "sehat", "jd", "berasa", "ikut", "sakit"	"kayak", "tidak", "sanggup", "pegang", "hp", "lihat", "twitter", "pada", "nyari", "oksigen", "terus", "kamar", "RS", "WA", "hari", "aja", "grup", "yang", "ngabarin", "positif", "nyari", "obat", "donar", "darah", "plasma", "kritis", "sesak", "napas", "gw", "yang", "sehat", "jadi", "berasa", "sakit"

d. Penghapusan *Stopword*

Nilai informasi yang terdapat dalam *stopword* hampir mendekati nol, dengan kata lain entropi yang dimiliki sangat rendah (Feinerer et al. 2008). Sebelum dilakukan analisis lebih lanjut, stopword harus dihilangkan. Tahap penghapusan *stopword* dilakukan untuk membuang kata-kata yang termasuk ke dalam daftar *stopword*. Contoh *stopword* dalam Bahasa Indonesia diantaranya yaitu dahulu,

ada, dalam, adanya, dan, pada, dan lain-lain. Acuan daftar kata-kata yang termasuk ke dalam stopword diperoleh dari library Natural Language Toolkit pada Python dalam Bahasa Indonesia. Contoh data tweet sebelum dan sesudah penghapusan stopword dapat dilihat pada Tabel 3.4.

Tabel 3.4 Contoh Penghapusan *Stopword*

Data awal	Data akhir
“dr”, “erlina”, “menyarankan”, “pasien”, “covid19”, “memantau”, “frekuensi”, “napas”, “tanda”, “sesak”, “napas”	“dr”, “erlina”, “menyarankan”, “pasien”, “covid19”, “memantau”, “frekuensi”, “napas”, “tanda”, “sesak”, “napas”
“kayak”, “tidak”, “sanggup”, “pegang”, “hp”, “lihat”, “twitter”, “pada”, “nyari”, “oksigen”, “terus”, “kamar”, “RS”, “WA”, “hari”, “aja”, “grup”, “yang”, “ngabarin”, “positif”, “nyari”, “obat”, “donar”, “darah”, “plasma”, “kritis”, “sesak”, “napas”, “gw”, “yang”, “sehat”, “jadi”, “berasa”, “sakit”	“kayak”, “tidak”, “sanggup”, “pegang”, “hp”, “lihat”, “twitter”, “pada”, “nyari”, “oksigen”, “terus”, “kamar”, “RS”, “WA”, “hari”, “aja”, “grup”, “yang”, “ngabarin”, “positif”, “nyari”, “obat”, “donar”, “darah”, “plasma”, “kritis”, “sesak”, “napas”, “gw”, “yang”, “sehat”, “jadi”, “berasa”, “sakit”

e. *Stemming*

Proses stemming dilakukan untuk menghapus awalan dan akhiran dari suatu kata. Tujuan dari tahap stemming adalah untuk mendapatkan kata dasar yang sesuai. Proses stemming menggunakan library Sastrawi berbasis Python yang tersedia pada https://github.com/sastrawi/sastrawi. Algoritma yang terdapat pada library Sastrawi adalah Nazief-Adriani yang digunakan untuk menghapus berbagai variasi awalan dan akhiran kata. Contoh data tweet sebelum dan sesudah proses stemming dapat dilihat pada Tabel 3.4.

Tabel 3.5 Contoh *stemming*

Data awal	Data akhir
“dr”, “erlina”, “menyarankan”, “pasien”, “covid19”, “memantau”, “frekuensi”, “napas”, “tanda”, “sesak”, “napas”	“dr”, “erlina”, “saran”, “pasien”, “covid19”, “pantau”, “frekuensi”, “napas”, “tanda”, “sesak”, “napas”
“kayak”, “tidak”, “sanggup”, “pegang”, “hp”, “lihat”, “twitter”, “pada”, “nyari”, “oksigen”, “terus”, “kamar”, “RS”, “WA”, “hari”, “aja”, “grup”, “yang”, “ngabarin”, “positif”, “nyari”, “obat”, “donar”, “darah”, “plasma”, “kritis”, “sesak”, “napas”, “gw”, “yang”, “sehat”, “jadi”, “berasa”, “sakit”	“sanggup”, “pegang”, “oksigen”, “kamar”, “grup”, “positif”, “obat”, “darah”, “plasma”, “kritis”, “sesak”, “napas”, “sehat”, “asa”, “sakit”

C. Pembobotan TF-IDF

Tahap terakhir yang dilakukan pada praproses data dengan TF-IDF. Pada tahap ini dilakukan perhitungan sesuai dengan persamaan 2.1, 2.2, dan 2.3. Hasil dari tahap ini adalah vektor ukuran kemiripan yang dimiliki tiap kata kunci dokumen.

D. Klasterisasi DBSCAN

Pada tahap ini dilakukan klasterisasi dengan algoritme DBSCAN untuk data tweet yang telah dilakukan praproses menjadi Term Document Matrix (TDM) menggunakan library scikit-learn. Klasterisasi digunakan untuk mendapatkan klaster dari setiap dokumen berdasarkan term terkait penyebaran penyakit menular langsung (studi kasus Covid-19). Klaster yang dihasilkan akan digunakan pada proses geovisualisasi. Flowchart yang menunjukkan teknik metode DBSCAN dapat dilihat pada Gambar 3.3.

Penjelasan *flowchart* algoritma DBSCAN sebagai berikut:

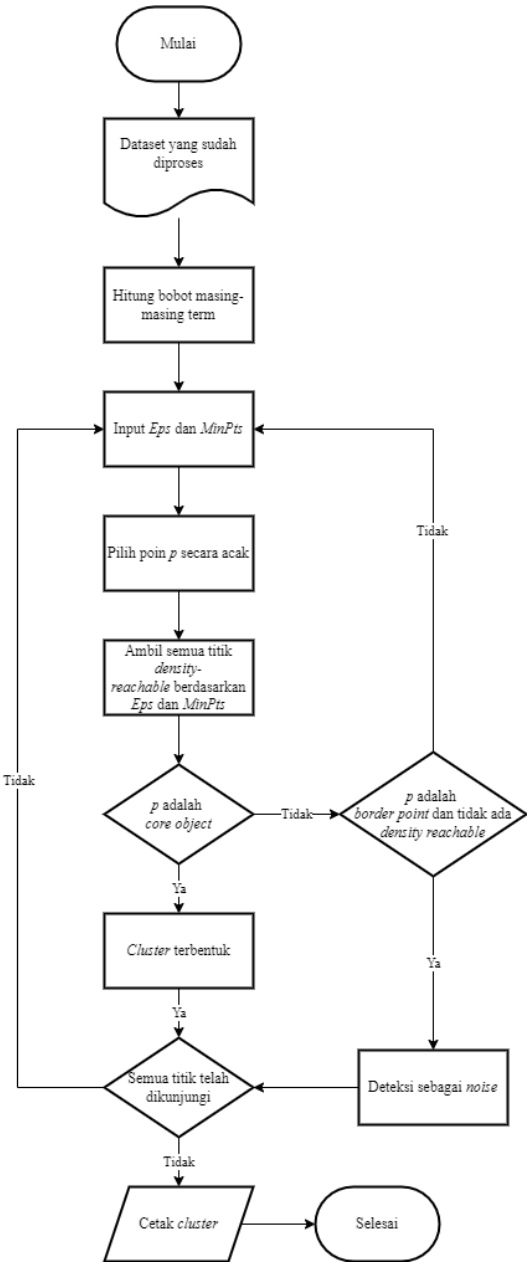
a. Dataset yang berbentuk file .csv dianggap sebagai input
b. Menghitung bobot tiap-tiap term
c. Epsilon dan MinPts dimasukkan dan dianggap sebagai input
d. Menentukan titik awal atau *p* secara acak

e. Mendapatkan titik *p* sebagai *core object* dan titik *p* tidak memiliki *density reachable*

f. Menghitung epsilon atau semua jarak titik pada *density reachable* terhadap *p* dengan menggunakan *cosine similarity* (sesuai dengan persamaan 2.4).

g. Mengulangi langkah perhitungan hingga tercipta titik yang dianggap sebagai *noise*.

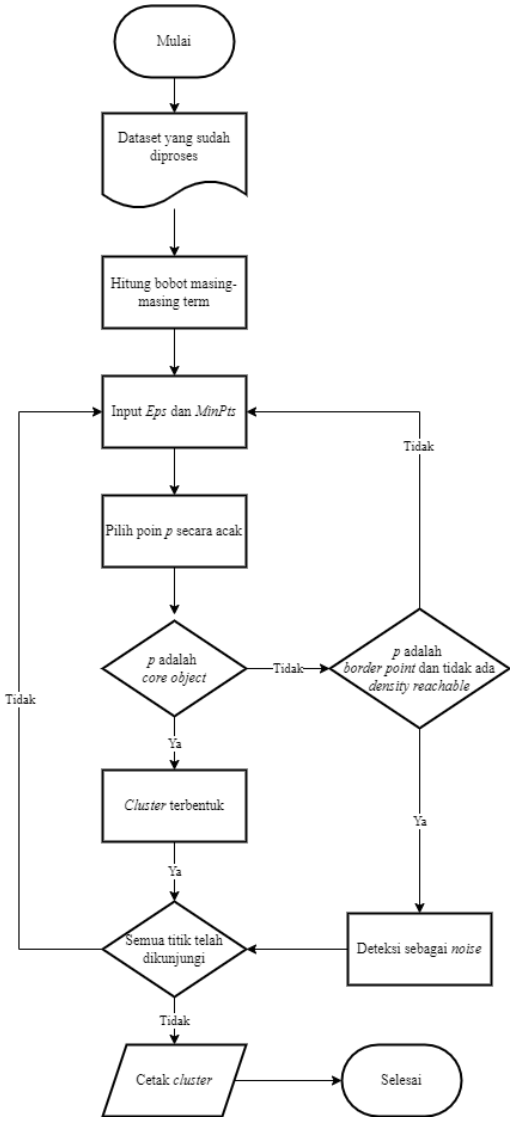
h. Mendapatkan *cluster* terbaik.



Gambar 3.3 *Flowchart* DBSCAN

E. Klasterisasi OPTICS

Pada tahap ini dilakukan klasterisasi dengan algoritme OPTICS untuk data *tweet* yang telah dilakukan praproses menjadi *Term Document Matrix* (TDM) menggunakan *library* scikit-learn. Klasterisasi digunakan untuk mendapatkan klaster dari setiap dokumen berdasarkan *term* terkait penyebaran penyakit menular langsung (studi kasus Covid-19). Klaster yang dihasilkan akan digunakan pada proses geovisualisasi. *Flowchart* yang menunjukkan teknik metode OPTICS dapat dilihat pada Gambar 3.4.



Gambar 3.4 *flowchart* OPTICS

Penjelasan *flowchart* algoritma OPTICS sebagai berikut:

a. Dataset yang berbentuk file .csv dianggap sebagai input

b. Menghitung bobot tiap-tiap term
c. Epsilon dan MinPts dimasukkan dan dianggap sebagai input
d. Menentukan titik awal atau *p* secara acak
e. Menghitung epsilon atau semua jarak titik pada *density reachable* terhadap *p* dengan menggunakan *cosine similarity* (sesuai dengan persamaan 2.4).
f. Mengulangi langkah perhitungan hingga tercipta titik yang dianggap sebagai *noise*.
g. Mendapatkan *cluster* terbaik.

F. Geovisualisasi

Proses geovisualisasi digunakan untuk mendistribusikan hasil klasterisasi yang diambil dari geolokasi setiap data tweet. Penerapan geovisualisasi dengan pemilihan warna, pola, dan ukuran agar dapat menambah informasi yang dibutuhkan untuk evaluasi hasil.

Proses geovisualisasi memanfaatkan *tools* folium yang tersedia pada library bahasa python, merupakan tools analitik dan visualisasi secara online. Berikut contoh gambar peta interaktif dari folium pada Gambar 3.3.

Gambar 3.5 Peta Folium

G. Evaluasi Hasil Analisis

Pada tahap ini klaster dianalisis dari proses klasterisasi dan lokasi penyebaran penyakit menular langsung dari proses geovisualisasi. Analisis hasil klaster dilakukan dengan melihat term yang sering muncul dan sesuai pada setiap klaster

untuk dijadikan dasar dalam penentuan label klaster. Sedangkan analisis lokasi penyebaran penyakit menular langsung dilakukan dengan cara membandingkan pola geolokasi dengan data mengenai daerah penyebaran penyakit menular yang diperoleh dari Kementerian Kesehatan Republik Indonesia (2021) melalui pendekatan statistik deskriptif, yaitu koefisien deskriptif yang dapat mewakili suatu dataset. Hasil analisis diharapkan dapat menjadi dasar keputusan alternatif dalam penanganan penyebaran penyakit menular langsung (studi kasus Covid19).

BAB IV

HASIL DAN PEMBAHASAN

4.1 Akuisisi Tweet

Pengumpulan data *tweet* ini memiliki judul akuisisi *tweet*. Hasil dari akuisisi *tweet* diperoleh dari proses *web crawling* dan *web scraping* dengan bahasa pemrograman Python dan Twitter API. Program Python menggunakan snscrape yang terdapat pada library Python di Github (https://github.com/JustAnotherArchivist/snscrape) untuk mendapatkan data *tweet*. Data *tweet* yang diambil adalah kumpulan kata kunci yang disebutkan pada Tabel 3.1 sejak April – September 2021 dan Januari – Juni 2022. *Source code* untuk scraping data terdapat pada Lampiran. Langkah-langkah untuk melakukan akuisisi *tweet* atau *scraping* data adalah sebagai berikut:

1. Melakukan instalasi *library pandas dan* snscrape pada *environment* python dengan editor Jupyter Notebook.
2. Menginisiasi *dataframe* dengan kolom sesuai objek yang dicari. Objek tersebut yaitu *id* yang memiliki nilai ID tweet, *date* memiliki nilai tanggal dari tweet, *username* memiliki nilai nama akun pengguna twitter, *renderedContent* memiliki nilai konten suatu tweet, *coordinates* memiliki nilai koordinat (latitude dan longitude) suatu tweet, dan *place* memiliki nilai daerah suatu tweet.
3. Melakukan *scraping* berdasarkan *query* pencarian menggunakan *library* snscrape dengan mengirimkan parameter kata kunci yang tertera pada Tabel 4.1.

Tabel 4.1 Tabel Hasil *Scraping*

No.	Kata Kunci	Tweet	Tweet Geolokasi	Tweet Duplikat	Tweet Non Duplikat
1	Covid-19	60.000	1.569	1.478	1.178
2	SARS-CoV-2	3.451	59	59	59
3	Coronavirus	15.640	302	302	302
4	Batuk	60.000	1.085	1.085	985
5	Batuk Kering	3.087	64	64	64
6	Pilek	60.000	860	659	635
7	Kelelahan	22.588	338	337	337
8	Sakit Kepala	60.000	1.248	1.040	940
9	Demam	60.000	1.540	1.240	740
10	Sesak Napas	16.023	245	245	245
11	Sakit Tenggorokan	34.758	403	403	403
12	Diare	34.308	559	559	559
13	Delta	47.381	1.240	1.034	834
14	Omicron	34.755	403	403	403
15	Anosmia	19.193	430	430	430
	Total	531.184	10.345	9.338	8.114

Pada Tabel 4.1 dilakukan proses *data cleaning* sebelum mendapatkan hasil akhir yaitu *tweet* non – duplikat. Tahap *scraping* dimulai dengan mendapatkan semua *tweet* dengan masing-masing kata kunci. Contoh hasil adalah kata kunci “anosmia” diperoleh sebanyak 19.193 *tweet*. Kemudian, dilakukan pengurangan data *tweet* dengan mencari *tweet* yang hanya memiliki *identity* geolokasi, contoh hasil kata kunci “anosmia” adalah diperoleh sebanyak 430 *tweet*. Proses terakhir dilakukan pengurangan *tweet* yang masih terdeteksi ganda / duplikat, dan contoh hasilnya kata kunci “anosmia” adalah 430 *tweet*. Contoh hasil tersebut mengartikan bahwa tidak ada *tweet* duplikat yang dimiliki oleh kata kunci “anosmia”.

4. Hasil dari program disimpan dalam bentuk file *Comma Separated Value* (CSV) dengan atribut *id, date, username, tweet, latitude, longitude.*

	id	date	username	tweet	latitude	longitude
9333	1536525461077262336	2022-06-14 01:46:31+00:00	natasya_puspa4	Pak Jokowi minta para menterinya mewaspadai da...	-6.364100	106.799599
9334	1536525113289752576	2022-06-14 01:45:08+00:00	Irwijaya_q	Pak Jokowi minta para menterinya mewaspadai da...	-6.364100	106.799599
9335	1536515216665899013	2022-06-14 01:05:49+00:00	Agistasyahka	Penyebaran virus Omicron BA.4 dan BA.5 ini leb...	-6.301652	106.974561
9336	1536358141377384448	2022-06-13 14:41:39+00:00	Ardi_Wdyto	Varian baru omicron lagi :)	-7.776581	113.198255
9337	1536251604772745216	2022-06-13 07:38:19+00:00	gemaposID	BREAKING NEWS!!\nOmicron BA.4 dan BA.5 resmi t...	-6.218042	106.857762

Gambar 4.1 Contoh Hasil Akuisisi *Tweet*

4.2 Praproses Data

A. Case Folding

Case Folding adalah tahapan yang berfungsi mengonversi keseluruhan teks dalam dokumen menjadi huruf kecil (*lowercase*). Langkah-langkah melakukan *case folding* yaitu memanggil fungsi *lower()* dalam iterasi setiap data dokumen. Contoh hasil praproses data pada tahap ini dapat dilihat pada Tabel 4.2.

Tabel 4.2 Proses *case folding*

Dokumen	Sebelum *case folding*	Sesudah *case folding*
1	BREAKING NEWS!!\nOmicron BA.4 dan BA.5 resmi terdeteksi di Indonesia. Mari kita sama sama menjegah mengantisipasi penularan varian baru Covid-19. @KemensosRI @dinkesJKT @BPJSKesehatanRI #OmicronVariant #Omicron #COVID https://t.co/NdUWvFg8Ja	breaking news omicron ba dan ba resmi terdeteksi di indonesia. mari kita sama sama menjegah mengantisipasi penularan varian baru covid-19. @kemensosri @dinkesjkt @bpjskesehatanri #omicronoariant #omicron #covid https://t.co/nduwvfg8ja
2	Varian baru omicron lagi :)	varian baru omicron lagi :)
3	Buset aku batuk udah kek orang mau mati aja.	buset aku batuk udah kek orang mau mati aja.

B. Tokenizing

Setelah dilakukan *case folding*, tahap praproses data selanjutnya adalah *Tokenizing* yang berfungsi untuk melakukan penghapusan *hashtag*, angka, *mention*, URL, dan tanda baca. Contoh hasil praproses data tahap *tokenizing* dapat dilihat pada Tabel 4.3. Langkah – langkah *tokenizing* adalah sebagai berikut:

a. Memanggil *library*, *numpy*, *re*, dan *string*.
b. Menggunakan *library* pandas untuk membaca data csv menjadi *dataframe.*
c. Mengambil data di kolom *text* dan menyimpan di dalam *list.*
d. Melakukan iterasi pada *list* dan menghapus *hashtag*, angka, *mention*, URL, dan tanda baca.

Tabel 4. 3 Proses *tokenizing*

Dokumen	Sebelum *Tokenizing*	Sesudah *Tokenizing*
1	breaking news omicron ba dan ba resmi terdeteksi di indonesia. mari kita sama sama menjegah mengantisipasi penularan varian baru covid-19. @kemensosri @dinkesjkt @bpjskesehatanri #omicronoariant #omicron #covid https://t.co/nduwvfg8ja	[breaking, news, omicron, ba, dan, ba, resmi, terdeteksi, di, indonesia., mari, kita, sama, sama, mencegah, mengantisipasi, penularan, varian, baru, covid-19.]
2	varian baru omicron lagi	[varian, baru, omicron, lagi]
3	buset aku batuk udah kek orang mau mati aja.	[buset, aku, batuk, udah, kek, orang, mau, mati, aja.]

C. Penghapusan *Stopword*

Pada tahap ini dilakukan penghapusan stopword. Tahap ini dilakukan untuk membuang kata-kata yang tidak terlalu berpengaruh pada pemrosesan text mining, seperti kata hubung dan termasuk ke dalam daftar stopword. Daftar kata yang akan dilakukan penghapusan adalah daftar kata yang termasuk pada library Natural Language Tool Kit dan daftar kata tambahan. Langkah – langkah penghapusan stopword tertera pada Tabel 4.4.

Tabel 4.4 Proses penghapusan *stopwords*

Dokumen	Sebelum hapus *stopwords*	Sesudah hapus *stopwords*
1	breaking news omicron ba dan ba resmi terdeteksi di indonesia mari kita sama sama mencegah mengantisipasi penularan varian baru covid-19	[breaking, news, omicron, resmi, terdeteksi, indonesia, mari, kita, sama, sama, mencegah, mengantisipasi, penularan, varian, baru, covid-19]
2	varian baru omicron lagi	[varian, baru, omicron]
3	buset aku batuk udah kek orang mau mati aja	[buset, aku, batuk, udah, orang, mati]

D. Stemming Nazief-Adriani

Tahap praproses data berikutnya adalah *stemming*, yaitu mengembalikan kata-kata yang menjadi imbuhan dari kata dasar dan menghapus awalan dan akhiran dari suatu kata. Tujuan dari tahap *stemming* adalah untuk mendapatkan kata dasar yang sesuai. Proses *stemming* menggunakan *library* Sastrawi berbasis Python yang tersedia pada https://github.com/sastrawi/sastrawi. Algoritma yang terdapat pada *library* Sastrawi adalah Nazief-Adriani yang digunakan untuk menghapus berbagai variasi awalan dan akhiran kata. Langkah – langkah tahapan *stemming* adalah sebagai berikut:

a. Memanggil *library* Sastrawi.
b. Melakukan iterasi dokumen, kemudian memanggil fungsi *stem()* untuk mengubah kata dalam dokumen tersebut menjadi kata dasar.

Contoh hasil praproses data tahap *stemming* dapat dilihat pada Tabel 4.5.

Tabel 4.5 Proses *stemming*

Dokumen	Sebelum *Stemming*	Sesudah *stemming*
1	breaking news omicron ba ba resmi terdeteksi indonesia mari kita sama sama mencegah mengantisipasi penularan varian baru covid-19	breaking news omicron ba ba resmi deteksi indonesia mari mencegah antisipasi tular varian covid
2	varian baru omicron	varian baru omicron
3	buset aku batuk udah orang mati	aku batuk udah orang mati

Preprocessing dilakukan agar data yang diolah sesuai tujuan dan lebih baik.

E. Term Document Matrix

Tahap selanjutnya adalah pembuatan Term Document Matrix dilakukan untuk menghasilkan matriks frekuensi kemunculan term pada suatu dokumen. Pada tahap ini menghasilkan 8.114 dokumen dengan minimal 20 sample pada 15 kata kunci yang dipilih. Banyaknya term yang dihasilkan membuat dimensi matriks menjadi terlalu besar sehingga perlu diperkecil dengan cara mereduksi term.

4.3 Pembobotan TF-IDF

Pada tahap pembobotan ini dilakukan dengan metode algoritma TF-IDF, yaitu menghitung tingkat pentingnya kata dalam sebuah dokumen, serta mengubah teks menjadi vektor yang berisikan hasil perhitungan TF-IDF. Perhitungan TF-IDF terdapat 3 langkah, menghitung Term Frequency (TF), menghitung Inverse Document Frequency (IDF), kemudian menghitung hasil perhitungan antara TF-IDF. Ketiga langkah tersebut mengacu pada perhitungan rumus 2.1, 2.2, dan 2.3. Secara rinci, langkah penerapan algoritme TF-IDF adalah sebagai berikut:

1. Menghitung Term Frequency (TF)

   Memanggil modul *CountVectorizer* dan *library scikit-learn*, kemudian menginisialisasi data dengan objek *CountVectorizer*. Kemudian,
2. Menghitung Inverse Document Frequency (IDF)

   Memanggil modul *TfidfTransformer* dari *library scikit-learn*, kemudian menginisialisasi objek *TfidfTransformer*. Kemudian, memasukkan objek *CountVectorizer* yang didapatkan dari langkah sebelumnya menggunakan fungsi *fit_transform().* Langkah terakhir, membuat *DataFrame* untuk menampilkan bobot dari setiap *term*. Langkah opsional, mengurutkan bobot yang dipilih dari frekuensi kemunculan *min_df* sebanyak 300 dokumen.
3. Menghitung hasil perkalian antara *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF)

	badan	banget	batuk	com	covid	dah	delta	demam	diare	kena	...	orang	pilek	sakit	sehat	sesak	tenggorok	tu	vaksin	varian
8110	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	1.000000
8111	0.0	0.0	0.0	0.0	0.576889	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.816823
8112	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.548384	0.0	0.0	0.0	0.0	0.0	0.836226
8113	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	1.000000
8114	0.0	0.0	0.0	0.0	0.576889	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.816823

Gambar 4.2 Hasil Perhitungan TF-IDF

Memanggil modul *TfidfVectorizer* dari *library scikit-learn*. Menginputkan data atau fungsi *fit_transform()*. Lalu, membuat *dataframe* untuk menampilkan matriks hasil perhitungan TF-IDF yang dapat dilihat pada Gambar 4.2. Hasil dari proses pembobotan ini akan disimpan dalam

bentuk *array* dan dilanjutkan ke proses berikutnya. Gambar 4.3 di bawah ini merupakan 10 *term* tertinggi dari hasil seluruh *term* pembobotan TF – IDF. Angka yang dihasilkan menunjukkan seberapa pentingnya suatu *term* terhadap keseluruhan dokumen.

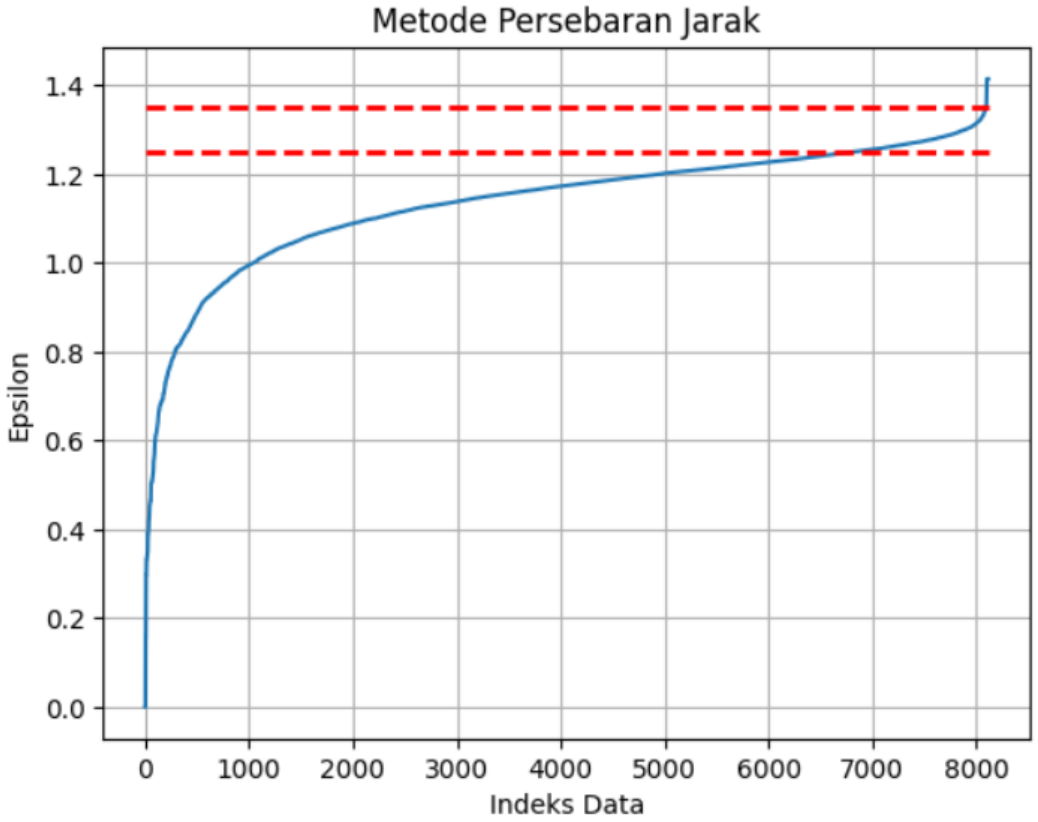
	Top Terms	Overall TF-IDF Scores
0	sakit	319.969905
1	demam	251.550249
2	kepala	225.012232
3	batuk	223.404272
4	covid	198.786286
5	pilek	171.432045
6	ni	131.735599
7	vaksin	126.287038
8	diare	122.664740
9	varian	113.465968

Gambar 4.3 *Top 10 term* Hasil TF – IDF

Top 10 term yang dihasilkan adalah *term* "sakit", "demam", "kepala", "batuk", "covid", "pilek", "ni", "vaksin", "diare", dan "varian". Angka yang dihasilkan adalah semakin tinggi angka terhadap suatu *term*, maka *term* tersebut relatif penting dan secara signifikan dalam sampel korpus yang disajikan.

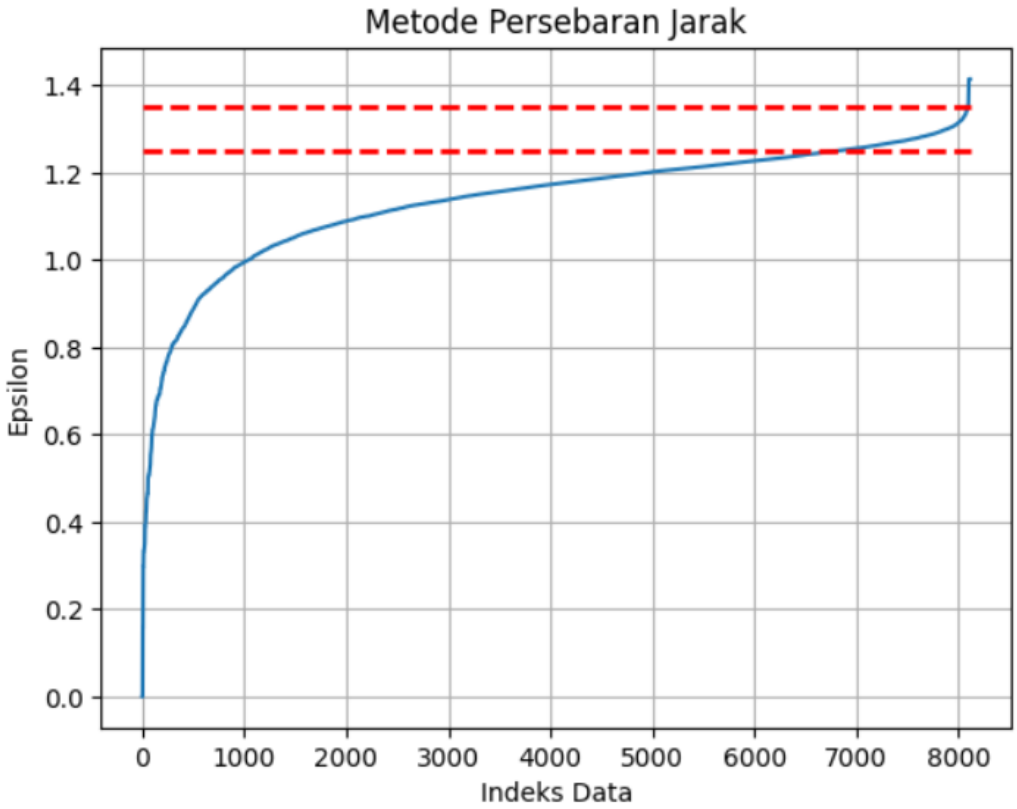
4.4 *NearestNeighbors*

Hasil dari penghitungan *k-distance NearestNeighbors* ini digunakan pada nilai *Eps* dan min_samples / minpts ditunjukkan pada Gambar 4.4.



Gambar 4.4 Grafik Metode *k-distance* pada minpts bernilai 5

Grafik metode di atas menunjukkan bahwa garis putus – putus merah sebagai nilai *range* epsilon yang optimal. Garis tersebut dipilih dalam metode *k-dist* sebagai pembuktian adanya pergeseran lembah dan puncak yang signifikan terhadap garis biru. Sedangkan garis biru yang tercipta adalah persebaran optimal yang mengindekskan banyaknya data pada sumbu x dan besaran epsilon pada sumbu y. Grafik metode ini menunjukkan nilai epsilon sebesar 1,25 dan 1,35 pada minpts bernilai 5.



Gambar 4.5 Grafik Metode *k-distance* pada minpts bernilai 10

Grafik metode di atas menunjukkan bahwa garis putus – putus merah sebagai nilai *range* epsilon yang optimal. Garis tersebut dipilih dalam metode *k-dist* sebagai pembuktian adanya pergeseran lembah dan puncak yang signifikan terhadap garis biru. Sedangkan garis biru yang tercipta adalah persebaran optimal yang mengindekskan banyaknya data pada sumbu x dan besaran epsilon pada sumbu y. Grafik metode ini menunjukkan nilai epsilon sebesar 1,25 dan 1,35 pada minpts bernilai 10.

Pada grafik ini didapatkan nilai *Eps* yang diperoleh sebesar 1,25 dan 1,35, kemudian k sebagai representatif min_samples / *minpts* bernilai 5. Hasil dari penghitungan ini selanjutnya digunakan dalam penghitungan klasterisasi dengan algoritma DBSCAN. Sedangkan algoritma OPTICS tidak membutuhkan hasil penghitungan *k-dist*, karena berlawanan dengan prinsip algoritma OPTICS.

4.5 Klasterisasi

A. DBSCAN

Pada tahap ini, hasil penghitungan jarak NearestNeighbors dengan modul sklearn.neighbors import NearestNeighbors dari package scikit-learn akan diterapkan proses klasterisasi. Modul yang digunakan adalah modul cluster pada package ini. Pada modul ini terdapat cluster DBSCAN yang berfungsi untuk menerapkan klasterisasi DBSCAN pada Term Document Matrix setelah proses reduksi dimensi metrik. Tujuan dari klasterisasi ini adalah untuk mencari kesamaan (*similarity*) feature antar dokumen sehingga dapat ditentukan cluster dari dokumen tersebut. Terdapat beberapa parameter untuk menentukan cluster dalam metode DBSCAN. Dapat dilihat pada Tabel 4.6.

Tabel 4.6 Daftar *input* Parameter DBSCAN yang digunakan

Parameter	Nilai	Deskripsi
min_samples / *minpts*	int	Jumlah sampel (atau bobot total) di suatu lingkungan untuk suatu titik yang dianggap sebagai titik inti. Ini termasuk poin itu sendiri.
epsilon (ε / *Eps*)	float	Jarak maksimum antara dua sampel untuk satu dianggap sebagai di lingkungan yang lain. Ini bukan batas maksimum pada jarak titik dalam sebuah *cluster*.
fit_predict	X / *cluster sample*	Metode hitung *cluster* dari data atau matriks jarak dan prediksi label.
metric	Euclidean (*default*)	Metrik yang akan digunakan saat menghitung jarak antar *instance* dalam larik fitur.

Implementasi validasi klaster menggunakan *silhouette coefficient* dan mendapatkan nilai *k-distance*. Implementasi dilakukan dengan memanfaatkan *library* NearestNeighbors, dengan memasukkan rentang nilai yang sudah dijelaskan sebelum subbab ini. Kode implementasi Python klasterisasi dan validasi jumlah

klaster pada ε = 1,25 dan minpts = 5 terlihat pada Gambar 4, ε = 1,35 dan minpts = 5 terlihat pada Gambar 4, ε = 1,25 dan minpts = 10, dan ε = 1,35 dan minpts = 10.

```python
from sklearn.cluster import DBSCAN
from sklearn.metrics import silhouette_score

# Inisialisasi dan clustering dengan DBSCAN
dbscan = DBSCAN(eps=1.25, min_samples=5)
cluster_labels = dbscan.fit_predict(tfidf_matrix)

# Evaluasi dengan Silhouette Score
silhouette_avg = silhouette_score(tfidf_matrix, cluster_labels)
print(f"Silhouette Score: {silhouette_avg}")
```

Gambar 4.6 Kode Python Implementasi ε = 1,25 dan minpts = 5 dengan *silhouette coefficient*

```python
from sklearn.cluster import DBSCAN
from sklearn.metrics import silhouette_score

# Inisialisasi dan clustering dengan DBSCAN
dbscan = DBSCAN(eps=1.35, min_samples=5)
cluster_labels = dbscan.fit_predict(tfidf_matrix)

# Evaluasi dengan Silhouette Score
silhouette_avg = silhouette_score(tfidf_matrix, cluster_labels)
print(f"Silhouette Score: {silhouette_avg}")
```

Gambar 4.7 Kode Python Implementasi ε = 1,35 dan minpts = 5 dengan *silhouette coefficient*

```python
from sklearn.cluster import DBSCAN
from sklearn.metrics import silhouette_score

# Inisialisasi dan clustering dengan DBSCAN
dbscan = DBSCAN(eps=1.25, min_samples=10)
cluster_labels = dbscan.fit_predict(tfidf_matrix)

# Evaluasi dengan Silhouette Score
silhouette_avg = silhouette_score(tfidf_matrix, cluster_labels)
print(f"Silhouette Score: {silhouette_avg}")
```

Gambar 4.8 Kode Python Implementasi ε = 1,25 dan minpts = 10 dengan *silhouette coefficient*

```python
from sklearn.cluster import DBSCAN
from sklearn.metrics import silhouette_score

# Inisialisasi dan clustering dengan DBSCAN
dbscan = DBSCAN(eps=1.35, min_samples=10)
cluster_labels = dbscan.fit_predict(tfidf_matrix)

# Evaluasi dengan Silhouette Score
silhouette_avg = silhouette_score(tfidf_matrix, cluster_labels)
print(f"Silhouette Score: {silhouette_avg}")
```

Gambar 4. 9 Kode Python Implementasi ε = 1,35 dan minpts = 10 dengan *silhouette coefficient*

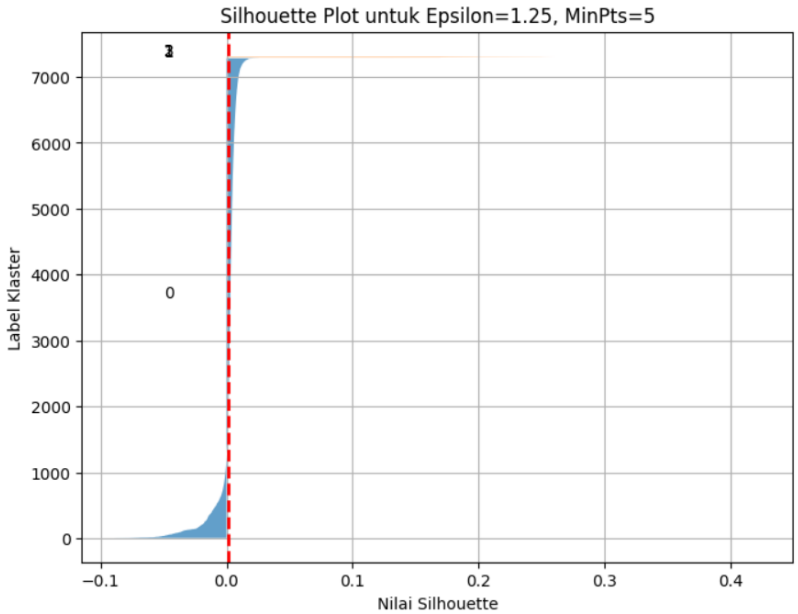
Cuplikan kode implementasi di atas menunjukkan bahwa terdapat *input* dari algoritma DBSCAN. Baris awal dimulai dengan mengimpor *function* DBSCAN dari *library* sklearn.cluster untuk pemrosesan algoritma DBSCAN, serta mengimpor *function* silhouette_score dari *library* sklearn.metrics untuk pemrosesan uji validasi nilai *silhouette coefficient*. Selanjutnya, dilakukan inisialisasi variabel "dbscan" dengan memasukkan *input* epsilon dan minpts sesuai ujicoba sampel hasil pengukuran *nearestneighbors* jarak *k-dist*. Kemudian, variable "cluster_labels" memuat parameter *function* fit_predict dengan variabel hasil pembobotan TF – IDF bernama tfidf_matrix. Fungsi dan pemrosesan DBSCAN sudah bisa berjalan dan selanjutnya dilakukan uji validasi oleh variabel "silhouette_avg" dari parameter hasil "tfidf_matrix" dan "cluster_labels".

Hasil dari implementasi kode dan validasi klaster menggunakan *silhouette coefficient* pada beberapa sampel epsilon dan minpts dapat dilihat pada Tabel. *Silhouette coefficient* memiliki rentang nilai dari -1 sampai 1. Jika koefisien silhouette mendekati 1 maka kualitas klaster semakin baik dan berlaku sebaliknya. Didapatkan *silhouette coefficient* tertinggi dari, pada minpts = 5 adalah 1 klaster, dan minpts = 10 adalah 1 klaster. Algoritma DBSCAN menganggap *noise* sebagai hal yang penting, dan dari 4 sampel percobaan tersebut menghasilkan *noise*. Jumlah klaster, *noise*, dan hasil perhitungan *silhouette coefficient* di setiap klaster bisa dilihat pada Tabel 4.8.

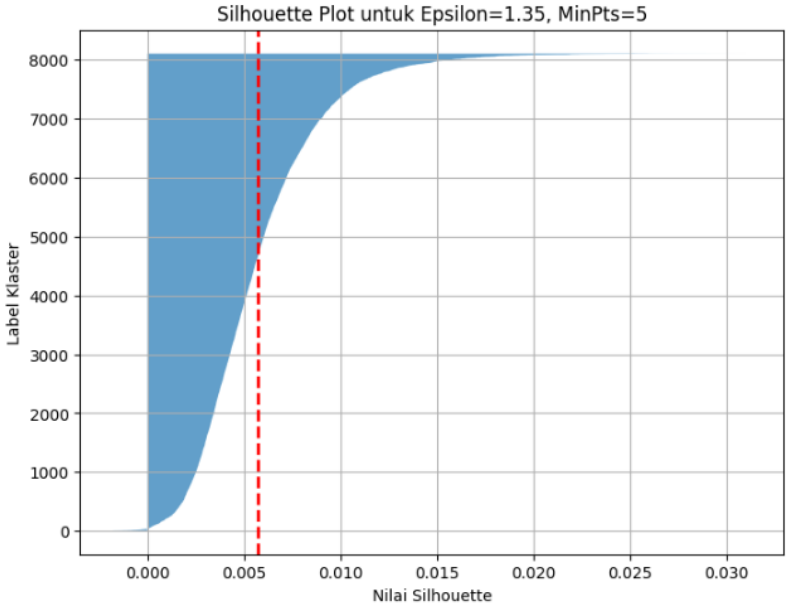
Tabel 4.7 Hasil *Silhouette Coefficient* DBSCAN

minpts	***eps***	**Jumlah Klaster**	***Noise***	***Silhouette Coeffiecient***
5	1,25	3	826	0.0014497934
	1,35	1	18	**0.0056921409**
10	1,25	2	1056	0.0024541545
	1,35	1	19	**0.0056951958**

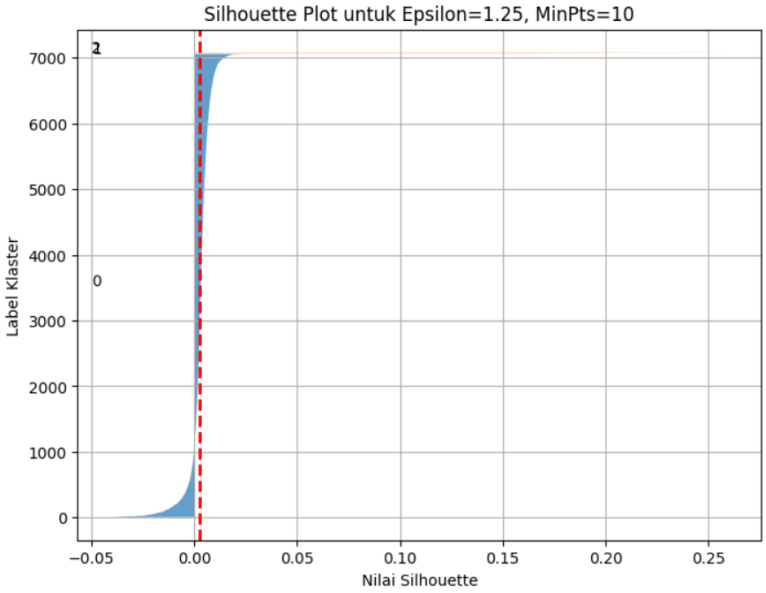
Implementasi hasil plot pada validasi *silhouette coefficient* dapat dilihat pada Gambar 4.9 sampai Gambar 4.12.



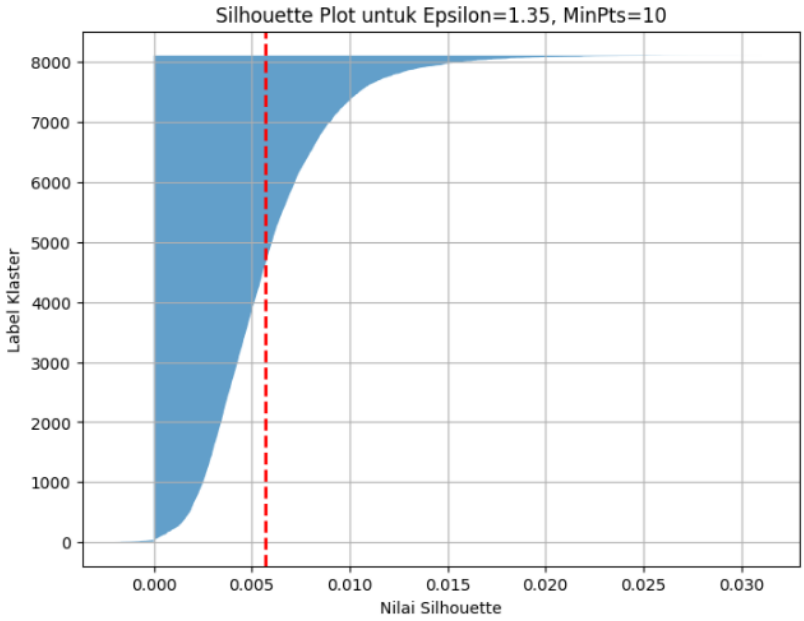
Gambar 4.10 Plot *silhouette coefficient* ε = 1,35 dan minpts = 5



Gambar 4.11 Plot *silhouette coefficient* ε = 1,35 dan minpts = 5



Gambar 4.12 Plot *silhouette coefficient* ε = 1,25 dan minpts = 10



Gambar 4.13 Plot *silhouette coefficient* ε = 1,35 dan minpts = 10

Pembuatan plot *silhouette coefficient* di atas menunjukkan plot biru berisi banyaknya data pelabelan klaster terhadap sumbu y, dan garis merah putus – putus sebagai plot hasil nilai *silhouette coefficient* terhadap sumbu x. Plot *silhouette* biru tersebut menggambarkan *noise* yang lebih banyak, jika semakin mengisi sumbu x

sebelum nilai 0, dan semakin baik hasilnya jika semakin mengisi lebih dari 0 atau mendekati 1.

Cuplikan salah satu implementasi kode yang menyusun plot tersebut dapat dilihat pada Gambar 4.13 dan Gambar 4.14.

```
from sklearn.metrics import silhouette_samples, silhouette_score
import matplotlib.pyplot as plt
from sklearn import metrics

dbscan = DBSCAN(eps=1.35, min_samples=10)
dbscan_labels = dbscan.fit_predict(tfidf_matrix)

silhouette_avg = silhouette_score(tfidf_matrix, dbscan_labels)
silhouette_values = silhouette_samples(tfidf_matrix, dbscan_labels)

plt.figure(figsize=(8, 6))
y_lower = 10  # Nilai awal batas bawah pada plot

for i in range(len(set(dbscan_labels))):
    cluster_silhouette_values = silhouette_values[dbscan_labels == i]
    cluster_silhouette_values.sort()
    size_cluster_i = cluster_silhouette_values.shape[0]
    y_upper = y_lower + size_cluster_i

    plt.fill_betweenx(np.arange(y_lower, y_upper), 0, cluster_silhouette_values, alpha=0.7)
    plt.text(-0.05, y_lower + 0.5 * size_cluster_i, str(i))
    y_lower = y_upper + 10  # Spasi antar klaster
```

Gambar 4.14 Kode Implementasi Plot *silhouette coefficient*

```
# Menghitung noise
# Merupakan salah satu metode untuk menghitung seberapa bagus metode dalam melakukan klasterisasi
noise_dbscan = sum(dbscan_labels == -1) / len(dbscan_labels)
noise = noise_dbscan * 8114
print(f"Estimated number of noise: {noise:.0f}")

sidb = silhouette_score(tfidf_matrix,dbscan.labels_)
print(f"Silhouette Coefficient DBSCAN: {metrics.silhouette_score(tfidf_matrix, dbscan_labels):.10f}")
dbscan_n_clusters = len(set(dbscan_labels)) - (1 if -1 in dbscan_labels else 0)
print(f"Estimated number of clusters: {dbscan_n_clusters}")
print(f"Estimated number of eps, minpts: 1.35, 10")

plt.title(f'Silhouette Plot untuk Epsilon=1.35, MinPts=10')
plt.xlabel('Nilai Silhouette')
plt.ylabel('Label Klaster')
plt.grid(True)
plt.axvline(x=silhouette_avg, color='red', linestyle='--', linewidth=2)
plt.show()
```

Gambar 4.15 Lanjutan Kode Implementasi Plot *silhouette coefficient*

Pada kode implementasi tersebut menghasilkan plot *silhouette coefficient* dan perhitungan *noise*. Kode implementasi menunjukkan sampel percobaan yang sesuai nilai *epsilon* pada pemrosesan *nearestneighbors* untuk mencari *k-dist.* Plot tersebut memanfaatkan library matplotlib.pyplot agar dapat memvisualisasikan plot hasil perhitungan *silhouette coefficient.*

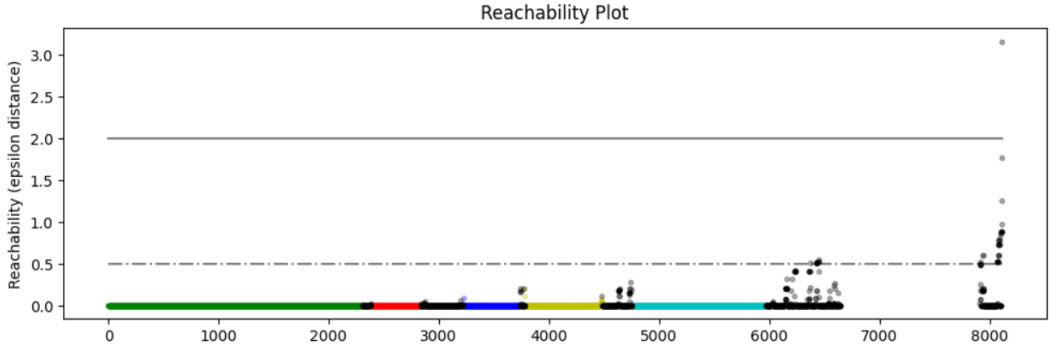
B. OPTICS

Pada tahap ini terdapat package OPTICS yang berfungsi untuk menerapkan klasterisasi OPTICS. Tujuan dari klasterisasi ini adalah untuk mencari kesamaan (*similarity*) feature antar dokumen sehingga dapat ditentukan cluster dari dokumen tersebut. Terdapat beberapa parameter untuk menentukan cluster dalam metode OPTICS. Dapat dilihat pada Tabel 4.

Tabel 4.8 Daftar *input* Parameter OPTICS *Clustering*

Parameter	Nilai	Deskripsi
MinPts	20 (int)	Jumlah sampel (atau bobot total) di suatu lingkungan untuk suatu titik yang dianggap sebagai titik inti. Ini termasuk poin itu sendiri.
xi *score* (xi)	0,05 (float) / *default*	Menentukan kecuraman minimum pada plot keterjangkauan yang membentuk batas cluster
Core distance	X / *cluster sample*	Nilai radius minimum yang diperlukan untuk mengklasifikasikan titik tertentu sebagai titik inti.
Reachability distance	Variabel huruf (p,q, dll)	Hubungan dengan titik data lain q. Jarak Reachability antara titik p dan q adalah maksimum Jarak Inti p dan Jarak Euclidean (atau metrik jarak lainnya) antara p dan q
metric	Minkowski (default)	Metrik yang akan digunakan untuk perhitungan jarak. Metrik apa pun dari scikit-learn atau scipy.spatial.distance dapat digunakan. Jika metrik adalah fungsi yang dapat dipanggil, ia dipanggil pada setiap pasangan instance (baris) dan nilai yang dihasilkan dicatat.

Sebelum menghitung algoritma OPTICS, dibutuhkan prinsip nilai Reachability untuk merepresentasikan setiap term yang digunakan dalam penghitungan plot algoritme.



Gambar 4.16 *Reachability Plot*

Plot di atas menunjukkan bahwa kemampuan jangkauan pada term yang digunakan yaitu sebanyak 8000 data term. Dari setiap data term tersebut akan berpengaruh pada nilai penghitungan algoritma OPTICS. Lemba mewakili klaster potensial yang dipisahkan oleh puncak. Untuk memvisualisasikan urutan set data asli, plot garis dihubungkan oleh titik-titik tersebut. Titik di setiap klaster dikunjungi secara berurutan yang dimulai dengan titik-titik di tengah (wilayah terpadat) dan kemudian titik-titik di daerah sekitarnya. Selanjutnya dilakukan klasterisasi OPTICS. OPTICS memiliki parameter batas klaster yang disebut, *xi score.* Plot *reachability* pada Gambar 4.16 menunjukkan puncak dan lembah yang ditandai dengan adanya *noise*.

```
import matplotlib.gridspec as gridspec
import matplotlib.pyplot as plt
import numpy as np

from sklearn.cluster import OPTICS, cluster_optics_dbscan

clust = OPTICS(min_samples=10, xi=0.05, min_cluster_size=0.05)

X = X_pca

# Run the fit
clust.fit(X)
```

Gambar 4.17 Kode Implementasi OPTICS *clustering*

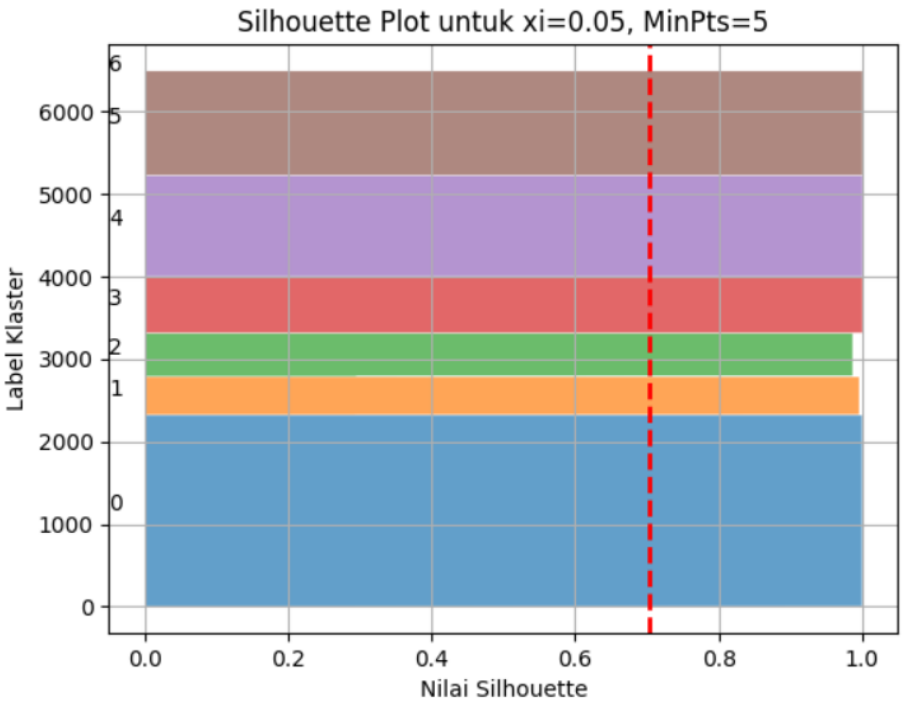
Implementasi kode Python yang digunakan adalah menggunakan *xi score* bernilai *default* (0,05) atau dapat dilihat pada *reachability distance* yaitu menunjukkan titik-titik plot tersebar pada nilai *default*. Output implementasi ini langsung diuji dengan *silhouette coefficient*. Pengujian klaster yang digunakan tertera pada Tabel 4.10.

Tabel 4.9 Hasil *silhouette coefficient* OPTICS

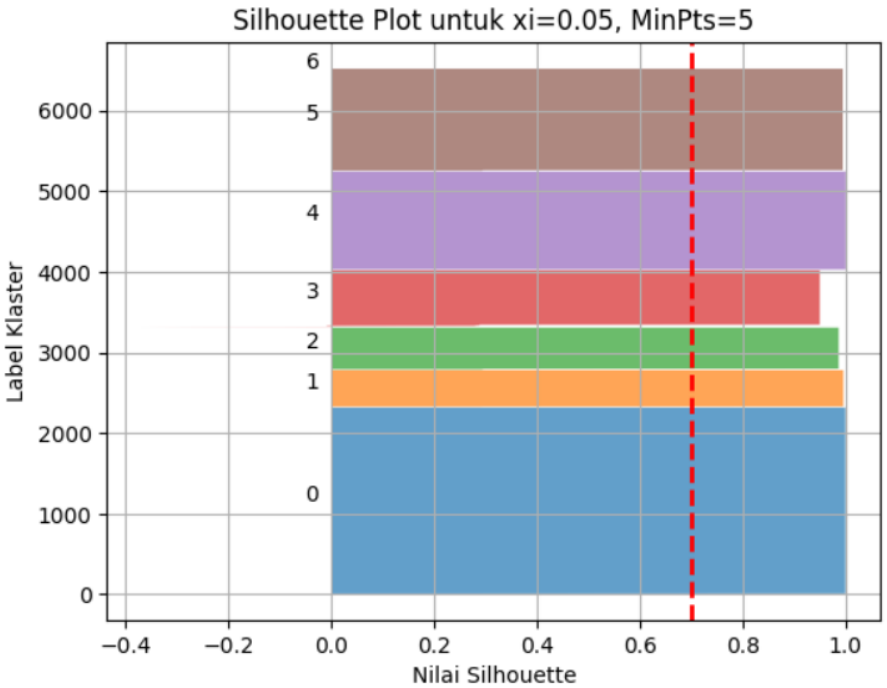
Xi score	***minpts***	**Jumlah Klaster**	***Noise***	***Silhouette Coeffiecient***
0,05	5	6	1689	0.6431164255
	10	6	1655	**0,6508317895**

Tabel 4.10 menunjukkan bahwa nilai *silhouette coefficient* terbaik dari algoritma OPTICS diperoleh dari parameter *xi score* = 0,05 dan minpts = 10. Pengujian ini dilakukan pada dua sampel minpts karena mengacu pada pengujian yang dilakukan pada metode *clustering* DBSCAN. *Noise* dalam algoritma DBSCAN juga berhasil diidentifikasi, ujicoba 1 sebanyak 1689 *noise*, dan ujicoba 2 sebanyak 1655 *noise.* Hasil *noise* terbaik dimiliki oleh xi score = 0,05 dan minpts = 10, karena semakin sedikit jumlah *noise* semakin baik hasil uji validasi. Plot *silhouette coefficient* yang dihasilkan oleh sampel-sampel pengujian ditunjukkan pada Gambar 4.18 dan Gambar 4.19.

Pada plot yang dihasilkan oleh algoritma OPTICS didapatkan bahwa 2 sampel pengujian terhadap xi score bernilai 0,05, minpts bernilai 5 dan 10. Plot berisi garis putus – putus merah sebagai hasil nilai *silhouette coefficient*, dan warna – warna yang menyusun sumbu y / label klaster sebagai klaster yang terbentuk. Hasil nilai *silhouette coefficient* dapat dilihat pada sumbu x. Terlihat plot hasil algoritma OPTICS tersebut terdiri dari 6 warna, yaitu klaster 1 mengisi warna biru, klaster 2 mengisi warna oranye, klaster 3 mengisi warna hijau, klaster 4 mengisi warna merah, klaster 5 mengisi warna ungu, dan klaster 6 mengisi warna coklat.



Gambar 4.18 Plot *silhouette coefficient* xi = 0,05 dan minpts = 5



Gambar 4.19 Plot *silhouette coefficient* xi score = 0,05 dan minpts = 10

```
# clust = OPTICS(min_samples=50, xi=0.05, min_cluster_size=0.05)

# Hitung Silhouette Score untuk masing-masing titik
silhouette_values = silhouette_samples(f_data, optics_labels)

# Hitung Silhouette Score rata-rata
silhouette_avg = silhouette_score(f_data, optics_labels)

y_lower = 10
for i in range(len(set(optics_labels))):
    cluster_silhouette_values = silhouette_values[optics_labels == i]
    cluster_silhouette_values.sort()
    size_cluster_i = cluster_silhouette_values.shape[0]
    y_upper = y_lower + size_cluster_i

    plt.fill_betweenx(np.arange(y_lower, y_upper), 0, cluster_silhouette_values, alpha=0.7)
    plt.text(-0.05, y_lower + 0.5 * size_cluster_i, str(i))
    y_lower = y_upper + 10  # Spasi antar klaster


sidb = silhouette_score(X,clust.labels_)
print(f"Silhouette Coefficient DBSCAN: {metrics.silhouette_score(X, optics_labels):.10f}")
optics_n_clusters = len(set(optics_labels)) - (1 if -1 in optics_labels else 0)
print(f"Estimated number of clusters: {optics_n_clusters}")
print(f"Estimated number of xi, minpts: 0.05, 5")
```

Gambar 4.20 Lanjutan Kode Implementasi Plot *silhouette coefficient*

4.6 Geovisualisasi

Data tweet hasil praproses data kemudian digabung dengan data hasil klasterisasi yang diperoleh dari proses klasterisasi dengan DBSCAN dan OPTICS. Data tweet divisualisasikan berdasar longitude, latitude, dan hasil klaster tiap dokumen. Proses geovisualisasi data hasil klasterisasi meliputi persiapan data dan perancangan peta.

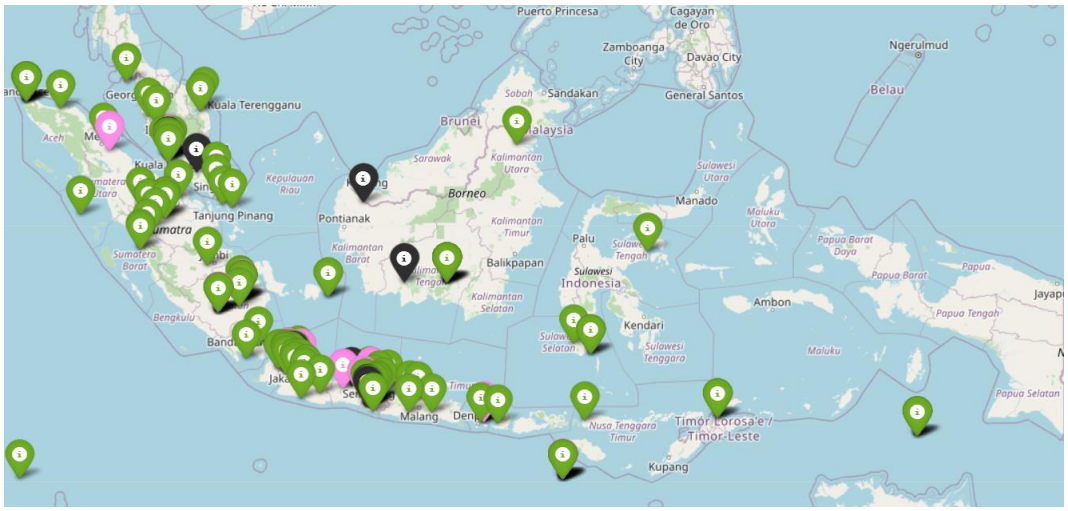
A. Persiapan Data

Variabel data yang digunakan sesuai dengan hasil tahapan uji analisis koefisien silhouette yang menunjukkan bahwa algoritma OPTICS memiliki nilai yang lebih besar atau mendekati 1.

B. Perancangan Peta

Perancangan peta ini menggunakan modul folium yang tersedia dalam library pandas python dengan memanfaatkan peta API dari OpenStreetMap. Data hasil klasterisasi tweet dengan DBSCAN dan OPTICS divisualisasikan menggunakan tanda poin berwarna yang saling menunjukkan klaster masing-masing term. Peta dilengkapi dengan fitur untuk menunjukkan detail isi term serta dari username

tweet. Detail ini dibutuhkan untuk mengidentifikasi term termasuk dalam klaster tertentu. Hasil visualisasi peta dapat dilihat pada Gambar 4.5 dan 4.6.



Gambar 4.21 Peta Hasil Visualisasi DBSCAN

Gambar 4.22 Peta Hasil Visualisasi OPTICS

Hasil visualisasi peta Gambar 4.21 dan Gambar 4.22 menunjukkan bahwa titik – titik sampel persebaran data *tweet* yang mengacu pada klaster – klaster tertentu. Klaster tersebut diinisiasi dengan perbedaan warna yang dimiliki tiap titik sampel. Pada Gambar 4.21 merupakan hasil geovisualisasi algoritma DBSCAN, sedangkan Gambar 4.22 merupakan hasil geovisualisasi algoritma OPTICS.

Warna hitam pada titik – titik *term* menggambarkan *noise* yang dimiliki tiap algoritma klaster. Jika pada Gambar 4.21, warna hijau dan merah muda merupakan titik – titik term yang membentuk klaster DBSCAN, yaitu klaster 0 untuk warna merah muda dan klaster 1 untuk warna hijau. Sedangkan pada Gambar 4.22 yang menunjukkan hasil geovisualisasi algoritma OPTICS, warna merah menunjukkan klaster 1 dan warna oranye menunjukkan klaster 2, dan terdapat beberapa klaster kecil yang menumpuk dan tidak terlihat persebaran dari kondisi *zoom* yang jauh.

Isi dari tiap titik – titik *term* tersebut adalah data *tweet* berupa *username* dan isi *tweet* / final_tweet. Berikut adalah contoh isi dari tiap titik – titik *term* yang membentuk klaster tersebut.

Gambar 4.23 Isi titik klaster geovisualisasi

Dari hasil perancangan peta tersebut masih diperoleh beberapa titik yang berasal dari luar Indonesia, hal itu disebabkan karena pada awal *scraping* data batasan yang digunakan adalah Bahasa Indonesia. Jadi, titik-titik tersebut menunjukkan tetap berbahasa Indonesia, namun berada di luar wilayah Negara Republik Indonesia.

4.7 Evaluasi Hasil Analisis

Tahap evaluasi hasil *clustering* DBSCAN dan OPTICS menggunakan *silhouette coefficient*. Evaluasi dilakukan pada beberapa sampel epsilon – minpts pada DBSCAN, dan *xi score* – minpts pada OPTICS. Evaluasi hasil klasterisasi dilakukan dengan cara memuat klaster yang sudah terbentuk, kemudian dilakukan evaluasi performa pada klaster tersebut menggunakan *library* sklearn *silhouette_score* agar diperoleh skor terbaik klaster sebagai bahan evaluasi.

Mulai

Mendapatkan hasil klaster DBSCAN / OPTICS

Menjalankan library *silhouette_score*

Menampilkan hasil *silhouette coefficient* dengan plot

Selesai

Gambar 4.24 *Pseoudocode* Evaluasi Klasterisasi

Dilakukan uji evaluasi terbaik pada tahap klasterisasi dengan DBSCAN memperoleh klaster bernilai 1 klaster, dengan nilai epsilon sebesar 1,35, dan MinPts sebesar 10 sehingga menghasilkan nilai *silhouette coefficient* sebesar 0,00569 untuk DBSCAN. Hasil klasterisasi dapat dilihat pada Tabel 4.11.

Tabel 4.10 Hasil *silhouette coefficient* DBSCAN

minpts	***eps***	**Jumlah Klaster**	***Noise***	***Silhouette Coeffiecient***
5	1,25	3	826	0,0014497934
	1,35	1	18	0,0056921409
10	1,25	2	1056	0,0024541545
	1,35	**1**	**19**	**0,0056951958**

Kemudian dilakukan uji evaluasi terbaik pada tahap klasterisasi dengan OPTICS memperoleh klaster bernilai 6 klaster, dengan nilai *xi score* sebesar 0,05,

dan MinPts sebesar 10 sehingga menghasilkan nilai *silhouette coefficient* sebesar 0,65083 untuk DBSCAN. Hasil klasterisasi dapat dilihat pada Tabel 4.

Tabel 4.11 Hasil *silhouette coefficient* OPTICS

Xi score	***minpts***	**Jumlah Klaster**	***Noise***	***Silhouette Coeffiecient***
0,05	5	6	1689	0,6431164255
	10	**6**	**1655**	**0,6508317895**

Perbandingan hasil klaster terbaik antara algoritma DBSCAN dan OPTICS, diperoleh klasterisasi OPTICS yaitu bernilai 0,6508317895. Dari hasil tersebut didapatkan 6 klaster yang diantaranya terdiri dari kumpulan anggota sampel. Berikut hitungan sampel setiap hasil *clustering* OPTICS.

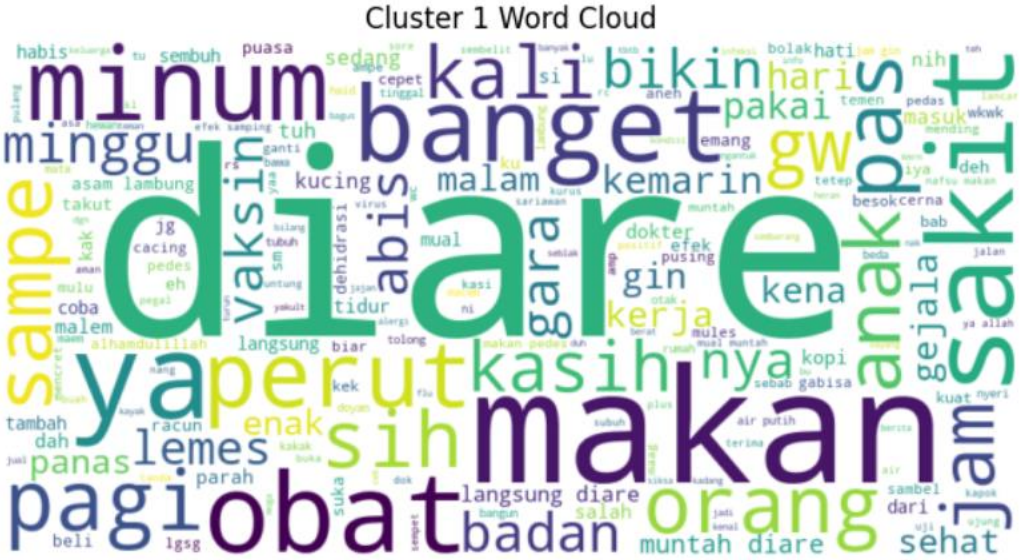
Tabel 4. 12 Hasil Tiap Anggota Klaster

Cluster	**Jumlah Anggota**
0	2307
1	454
2	517
3	700
4	1216
5	1265

Kemudian, dengan bantuan *library* Python bernama WordCloud, dapat divisualisasikan kumpulan term yang membesar dan dipadatkan dalam 1 gambar. Hasil tersebut diketahui bahwa klaster 1 memiliki visual term terbesar adalah “sakit kepala”. Gambar *wordcloud* untuk klaster 1 ada pada Gambar 4.23.

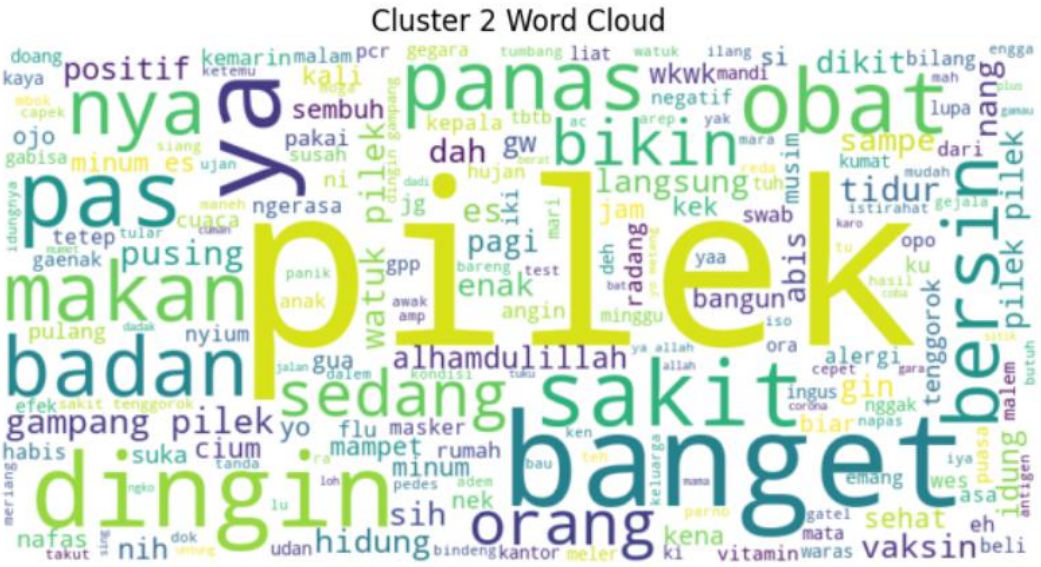
Gambar 4.25 Hasil visualisasi *WordCloud* klaster 1

Hasil klaster 2 didapatkan term visualisasi terbesar yaitu “diare”. Gambar *wordcloud* untuk klaster 2 ada pada Gambar 4.24.



Gambar 4.26 Hasil visualisasi *WordCloud* klaster 2

Hasil klaster 3 didapatkan term visualisasi terbesar yaitu “pilek”. Gambar *wordcloud* untuk klaster 3 ada pada Gambar 4.25.

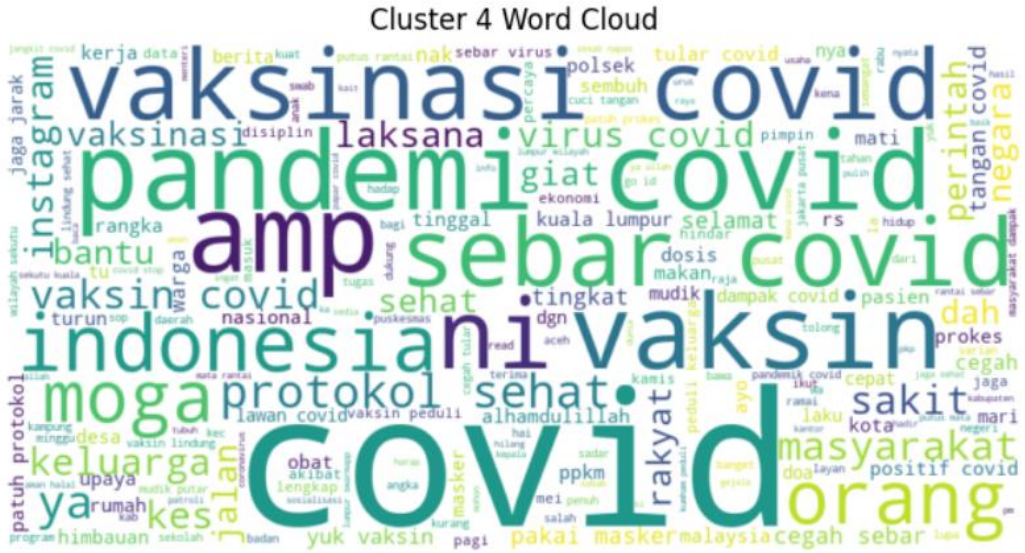


Gambar 4.27 Hasil visualisasi *WordCloud* klaster 3

Hasil klaster 4 didapatkan term visualisasi terbesar yaitu “batuk”. Gambar *wordcloud* untuk klaster 4 ada pada Gambar 4.26.

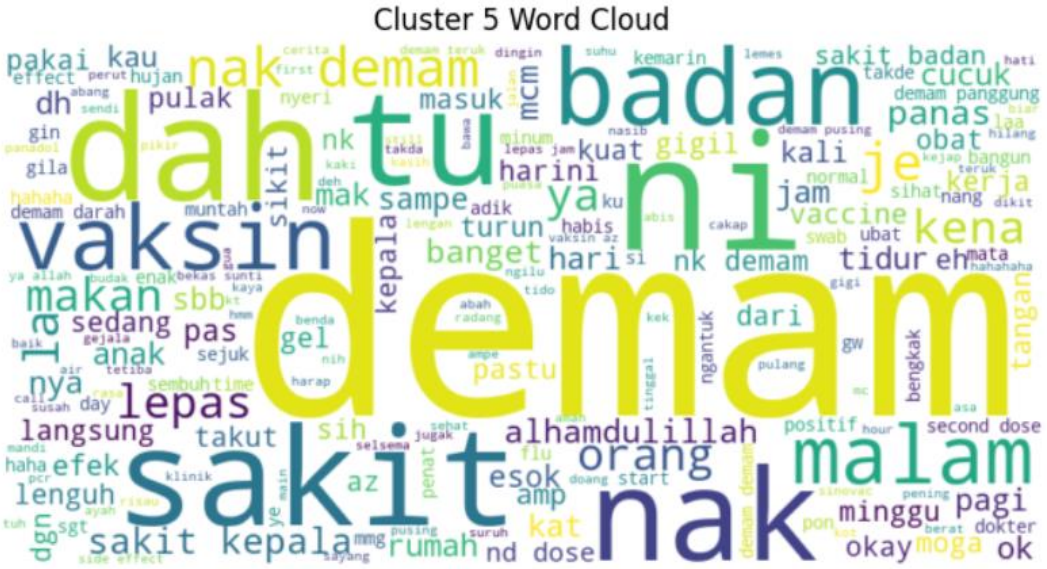
Gambar 4.28 Hasil visualisasi *WordCloud* klaster 4

Hasil klaster 5 didapatkan term visualisasi terbesar yaitu “covid”. Gambar *wordcloud* untuk klaster 5 ada pada Gambar 4.27.



Gambar 4.29 Hasil visualisasi *WordCloud* klaster 5

Hasil klaster 6 didapatkan term visualisasi terbesar yaitu “demam”. Gambar *wordcloud* untuk klaster 6 ada pada Gambar 4.28.



Gambar 4.30 Hasil visualisasi *WordCloud* klaster 6

Dapat disimpulkan dari analisis visualisasi *WordCloud,* yaitu term “sakit kepala” pada klaster 1, term “diare” pada klaster 2, term “pilek” pada klaster 3, term “batuk” pada klaster 4, term “covid” pada klaster 5, dan term “demam” pada klaster 6.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari penelitian yang telah dilakukan dapat diambil beberapa kesimpulan sebagai berikut:

1. Hasil penerapan klasterisasi penyebaran penyakit menular langsung pada studi kasus COVID-19 dengan menggunakan algoritma DBSCAN dan OPTICS pada *tweet* terkait penyebaran penyakit menular langsung (studi kasus Covid-19) pada beberapa sampel yang menghasilkan parameter optimal pada eps = 1,35 dan minpts = 10. Parameter tersebut menghasilkan 1 klaster, 19 *noise*, dan nilai *silhouette coefficient* sebesar 0,0056951958. Sedangkan, algoritma OPTICS menghasilkan parameter optimal pada *xi score* = 0,05 dan minpts = 10. Parameter OPTICS ini menghasilkan klaster terbaik sebesar 6 klaster, 1655 *noise*, dan nilai *silhouette coefficient* sebesar 0,6508317895. Hasil terbaik penerapan algoritma *clustering* ini dimiliki oleh algoritma OPTICS, karena uji validasi dari algoritma ini menunjukkan nilai lebih tinggi daripada algoritma DBSCAN.
2. Geovisualisasi berhasil diterapkan pada hasil klasterisasi kedua data *tweet* dengan algoritma DBSCAN dan OPTICS. Hasil geovisualisasi ini berupa peta dengan titik – titik *term* yang berisi sampel *tweet* dengan atribut *username* dan *final_tweet*.
3. Peta hasil geovisualisasi dengan algoritma OPTICS menghasilkan 6 klaster dengan 6 term tertinggi yang terbentuk pada hasil *clustering* terbaik, yaitu term “sakit kepala” pada klaster 1, term “diare” pada klaster 2, term “pilek” pada klaster 3, term “batuk” pada klaster 4, term “covid” pada klaster 5, dan term “demam” pada klaster 6.

5.2 Saran

Demi mendapatkan model klasterisasi dan hasil evaluasi yang lebih baik, diperlukan pengembangan lebih lanjut. Saran untuk penelitian selanjutnya adalah sebagai berikut:

1. Menggunakan fungsi pencarian lain seperti *hashtag* atau *mention* sehingga data yang didapatkan lebih spesifik.
2. Pengembangan penelitian selanjutnya dapat menggunakan metode lain agar menyesuaikan karakteristik data teks.
3. Analisis dalam penelitian ini dibatasi pada metode *unsupervised learning*. Oleh karena itu, pengembangan penelitian selanjutnya diharapkan mampu menganalisis tingkat gejala penyakit penderita di tingkat kabupaten dan desa dengan menggunakan metode *supervised learning*.

DAFTAR PUSTAKA

Baumgartner, C., & Graber, A. (2007). Data mining and knowledge discovery in metabolomics. *Successes and New Directions in Data Mining*, *39*(11), 141–166. https://doi.org/10.4018/978-1-59904-645-7.ch007

Budiman, S., Safitri, D., & Ispriyanti, D. (2016). Perbandingan Metode K-Means Dan Metode Dbscan Pada Pengelompokan Rumah Kost Mahasiswa Di Kelurahan Tembalang Semarang. *Jurnal Gaussian*, *5*(4), 757–762.

Chakrabarti, S., Ester, M., Fayyad, U., & Gehrke, J. (2006). Data mining curriculum: a proposal. *Acm Sigkdd*, 1–10. http://pdf.aminer.org/000/303/279/decision_tree_construction_from_multidimensional_structured_data.pdf%5Cnhttp://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Data+mining+curriculum:+A+proposal+(Version+1.0)#4%5Cnhttp://scholar.google.com/scholar

Crooks, A., Croitoru, A., Stefanidis, A., & Radzikowski, J. (2013). #Earthquake: Twitter as a Distributed Sensor System. *Transactions in GIS*, *17*(1), 124–147. https://doi.org/10.1111/j.1467-9671.2012.01359.x

Devi, A. S., Putra, I. K. G. D., & Sukarsa, I. M. (2015). Implementasi Metode Clustering DBSCAN pada Proses Pengambilan Keputusan. *Lontar Komputer : Jurnal Ilmiah Teknologi Informasi*, *6*(3), 185. https://doi.org/10.24843/lkjiti.2015.v06.i03.p05

Dwiarni, B. A., & Setiyono, B. (2019). Akuisisi dan Clustering Data Sosial Media Menggunakan Algoritma K-Means sebagai Dasar untuk Mengetahui Profil Pengguna. *Jurnal Sains Dan Seni*, *8*(2), 2337–3520. https://apps.twitter.com/

Fay, D. L. (1967). 済無No Title No Title No Title. *Angewandte Chemie International Edition, 6(11), 951–952.*

Feinerer, I., Hornik, K., & Meyer, D. (2008). Text mining infrastructure in R. *Journal of Statistical Software*, *25*(5), 1–54. https://doi.org/10.18637/jss.v025.i05

Freeman, J. (2019). What is an API? Application programming interfaces explained. In *InfoWorld* (pp. 1–9). https://www.infoworld.com/article/3269878/what-is-an-api-application-programming-interfaces-explained.html

Han, J., Kamber, M., & Pei, J. (Eds.). (2012). About the Authors. In *Data Mining (Third Edition)* (Third Edit, p. xxxv). Morgan Kaufmann. https://doi.org/https://doi.org/10.1016/B978-0-12-381479-1.00027-7

Imran, M., Elbassuoni, S., Castillo, C., Diaz, F., & Meier, P. (2013). Extracting information nuggets from disaster- Related messages in social media. *ISCRAM 2013 Conference Proceedings - 10th International Conference on Information Systems for Crisis Response and Management*, *May*, 791–801.

Koko Mukti Wibowo, Indra Kanedi, J. J. (2021). Sistem Informasi Geografis (Sig) Menentukan Lokasi Pertambangan Batu Bara Di Provinsi Bengkulu Berbasis Website. *Jurnal Media Infotama*, *11*(1), 223–260.

Liao, S. H., Chu, P. H., & Hsiao, P. Y. (2012). Data mining techniques and applications - A decade review from 2000 to 2011. *Expert Systems with Applications*, *39*(12), 11303–11311. https://doi.org/10.1016/j.eswa.2012.02.063

Melcer, E. F., & Isbister, K. (2018). Bots & (main)frames: Exploring the impact of tangible blocks and collaborative play in an educational programming game. *Conference on Human Factors in Computing Systems - Proceedings*, *2018-April*(April). https://doi.org/10.1145/3173574.3173840

Nurdiana, O., Jumadi, J., & Nursantika, D. (2016). Perbandingan Metode Cosine Similarity Dengan Metode Jaccard Similarity Pada Aplikasi Pencarian Terjemah Al-Qur'an Dalam Bahasa Indonesia. *Jurnal Online Informatika*, *1*(1), 59. https://doi.org/10.15575/join.v1i1.12

Prabahari, R. . T. (2014). *A Comparative Analysis of Density Based Clustering Techniques for Outlier Mining*. *3*(11), 132–136.

Putri, M. M., Dewi, C., Permata Siam, E., Asri Wijayanti, G., Aulia, N., & Nooraeni, R. (2021). *Komparasi DBSCAN dan K-Means Clustering pada Pengelompokan Status Desa di Jawa Tengah Tahun 2020*. *17*(3), 394–404. https://doi.org/10.20956/j.v17i3.11704

Rahmanti, A. R., Ningrum, D. N. A., Lazuardi, L., Yang, H. C., & Li, Y. C. (2021). Social Media Data Analytics for Outbreak Risk Communication: Public Attention on the "New Normal" During the COVID-19 Pandemic in Indonesia. *Computer Methods and Programs in Biomedicine*, *205*, 106083. https://doi.org/10.1016/j.cmpb.2021.106083

Sakaki, T., Okazaki, M., & Matsuo, Y. (2013). Tweet analysis for real-time event detection and earthquake reporting system development. *IEEE Transactions on Knowledge and Data Engineering*, *25*(4), 919–931. https://doi.org/10.1109/TKDE.2012.29

Salman, N. (2023). *Density-Based Clustering Analysis*. *8*, 1–8.

Santoso, A. M. . (2022). Covid-19 : Varian Dan Mutasi. *Jurnal Medika Hutama*, *3*(02), 1980–1986. https://jurnalmedikahutama.com/index.php/JMH/article/view/396/271

Silitonga, P. (2016). ANALISIS POLA PENYEBARAN PENYAKIT PASIEN PENGGUNA BADAN PENYELENGGARA JAMINAN SOSIAL (BPJS) KESEHATAN DENGAN MENGGUNAKAN METODE DBSCAN CLUSTERING (Studi Kasus Rumah Sakit Umum Pusat Haji Adam Malik Medan). *Jurnal TIMES*, *Vol. V No*(ISSN : 2337-3601), 11–40. http://etd.lib.metu.edu.tr/upload/12620012/index.pdf

Susanto, H., Sumpeno, S., & Rachmadi, R. F. (2014). Visualisasi Data Teks TwitterBerbasis Bahasa Indonesia Menggunakan Teknik Pengklasteran. *Jurnal Teknik Elektro Institut Teknologi Sepuluh Nopember*, 6. http://digilib.its.ac.id/ITS-paper-22121150006831/35629

Susilo, A., Rumende, C. M., Pitoyo, C. W., Santoso, W. D., Yulianti, M., Herikurniawan, H., Sinto, R., Singh, G., Nainggolan, L., Nelwan, E. J., Chen, L. K., Widhani, A., Wijaya, E., Wicaksana, B., Maksum, M., Annisa, F., Jasirwan, C. O. M., & Yunihastuti, E. (2020). Coronavirus Disease 2019: Tinjauan Literatur Terkini. *Jurnal Penyakit Dalam Indonesia*, *7*(1), 45. https://doi.org/10.7454/jpdi.v7i1.415

Wahyuni, R. T., Prastiyanto, D., & Supraptono, E. (2017). Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen Skripsi. *Jurnal Teknik Elektro Universitas Negeri Semarang*, *9*(1), 18–23. https://journal.unnes.ac.id/nju/index.php/jte/article/download/10955/6659

Terpstra, Teun & Vries, A. & Stronkman, Richard & Paradies, G.L.. (2012). Towards a realtime twitter analysis during crises for operational crisis management. 1-9.

LAMPIRAN

LAMPIRAN 1: Daftar Kata Stopword

‘ada’, ‘adalah’, ’adanya’, ‘adapun’, ‘agak’, ‘agaknya’, ‘agar’, ‘akan’, ‘akankah’, ‘akhir’, ‘akhiri’, ‘akhirnya’, ‘aku’, ’akulah’, ‘amat’, ‘amatlah’, ‘anda’, ‘andalah’, ‘antar’, ‘antara’, ‘antaranya’, ‘apa’, ‘apaan’, ‘apabila’, ‘apakah’, ‘apalagi’, ‘apatah’, ‘artinya’, ‘asal’, ‘asalkan’, ‘atas’, ‘atau’, ‘ataukah’, ‘ataupun’, ‘awal’, ‘awalnya’, ‘bagai’, ‘bagaikan’, ‘bagaimana’, ‘bagaimanakah’, ‘bagaimanapun’, ‘bagi’, ‘bagian’, ‘bahkan’, ‘bahwa’, ‘bahwasanya’, ‘baik’, ‘bakal’, ‘bakalan’, ‘balik’, ‘banyak’, ‘bapak’, ‘baru’, ‘bawah’, ‘beberapa’, ‘begini’, ‘beginian’, ‘beginikah’, ‘beginilah’, ‘begitu’, ‘begitukah’, ‘begitulah’, ‘begitupun’, ‘bekerja’, ‘belakang’, ‘belakangan’, ‘belum’, ‘belumlah’, ‘benar’, ‘benarkah’, ‘benarlah’, ‘berada’, ‘berakhir’, ‘berakhirlah’, ‘berakhirnya’, ‘berapa’, ‘berapakah’, ‘berapalah’, ‘berapapun’, ‘berarti’, ‘berawal’, ‘berbagai’, ‘berdatangan’, ‘beri’, ‘berikan’, ‘berikut’, ‘berikutnya, ‘berjumlah’, ‘berkali-kali’, ‘berkata’, ‘berkehendak’, ‘berkeinginan’, ‘berkenaan’, ‘berlainan’, ‘berlalu’, ‘berlangsung’, ‘berlebihan’, ‘bermacam’, ‘bermacam-macam’, ‘bermaksud’, ‘bermula’, ‘bersama’, ‘bersama-sama’, ‘bersiap’, ‘bersiap-siap’, ‘bertanya’, ‘bertanya-tanya’, ‘berturut’, ‘berturut-turut’, ‘bertutur’, ‘berujar’, ‘berupa’, ‘besar’, ‘betul’, ‘betulkah’, ‘biasa’, ‘biasanya’, ‘bila’, ‘bilakah’, ‘bisa’, ‘bisakah’, ‘boleh’, ‘bolehkah’, ‘bolehlah’, ‘buat’, ‘bukan’, ‘bukankah’, ‘bukanlah’, ‘bukannya’, ‘bulan’, ‘bung’, ‘cara’, ‘caranya’, ‘cukup’, ‘cukupkah’, ‘cukuplah’, ‘cuma’, ‘dahulu’, ‘dalam’, ‘dan’, ‘dapat’, ‘dari’, ‘daripada’, ‘datang’, ‘dekat’, ‘demi’, ‘demikian’, ‘demikianlah’, ‘dengan’, ‘depan’, ‘di’, ‘dia’, ‘diakhiri’, ‘diakhirinya’, ‘dialah’, ‘diantara’, ‘diantaranya’, ‘diberi’, ‘diberikan’, ‘diberikannya’, ‘dibuat’, ‘dibuatnya’, ‘didapat’, ‘didatangkan’, ‘digunakan’, ‘diibaratkan’, ‘diibaratkannya’, ‘diingat’, ‘diingatkan’, ‘diinginkan’, ‘dijawab’, ‘dijelaskan’, ‘dijelaskannya’, ‘dikarenakan’, ‘dikatakan’, ‘dikatakannya’, ‘dikerjakan’, ‘diketahui’, ‘diketahuinya’, ‘dikira’, ‘dilakukan’, ‘dilalui’, ‘dilihat’, ‘dimaksud’, ‘dimaksudkan’, ‘dimaksudkannya’, ‘dimaksudnya’, ‘diminta’, ‘dimintai’, ‘dimisalkan’, ‘dimulai’, ‘dimulailah’, ‘dimulainya’, ‘dimungkinkan’, ‘dini’, ‘dipastikan’, ‘diperbuat’, ‘diperbuatnya’, ‘dipergunakan’, ‘diperkirakan’, ‘diperlihatkan’, ‘diperlukan’, ‘diperlukannya’, ‘dipersoalkan’, ‘dipertanyakan’, ‘dipunyai’, ‘diri’, ‘dirinya’, ‘disampaikan’, ‘disebut’, ‘disebutkan’, ‘disebutkannya’, ‘disini’, ‘disinilah’, ‘ditambahkan’, ‘ditandaskan’, ‘ditanya’, ‘ditanyai’, ‘ditanyakan’, ‘ditegaskan’, ‘ditujukan’, ‘ditunjuk’, ‘ditunjuki’, ‘ditunjukkan’, ‘ditunjukkannya’, ‘ditunjuknya’, ‘dituturkan’, ‘dituturkannya’, ‘diucapkan’, ‘diucapkannya’, ‘diungkapkan’, ‘dong’, ‘dua’, ‘dulu’, ‘empat’, ‘enggak’, ‘enggaknya’, ‘entah’, ‘entahlah’, ‘guna’, ‘gunakan’, ‘hal’, ‘hampir’, ‘hanya’, ‘hanyalah’, ‘hari’, ‘harus’, ‘haruslah’, ‘harusnya’, ‘hendak’, ‘hendaklah’,

‘hendaknya’, ‘hingga’, ‘ia’, ‘ialah’, ‘ibarat’, ‘ibaratkan’, ‘ibaratnya’, ‘ibu’, ‘ikut’, ‘ingat’, ‘ingat-ingat’, ‘ingin’, ‘inginkah’, ‘inginkan’, ‘ini’, ‘inikah’, ‘inilah’, ‘itu’, ‘itukah’, ‘itulah’, ‘jadi’, ‘jadilah’, ‘jadinya’, ‘jangan’, ‘jangankan’, ‘janganlah’, ‘jauh’, ‘jawab’, ‘jawaban’, ‘jawabnya’, ‘jelas’, ‘jelaskan’, ‘jelaslah’, ‘jelasnya’, ‘jika’, ‘jikalau’, ‘juga’, ‘jumlah’, ‘jumlahnya’, ‘justru’, ‘kala’, ‘kalau’, ‘kalaulah’, ‘kalaupun’, ‘kalian’, ‘kami’, ‘kamilah’, ‘kamu’, ‘kamulah’, ‘kan’, ‘kapan’, ‘kapankah’, ‘kapanpun’, ‘karena’, ‘karenanya’, ‘kasus’, ‘kata’, ‘katakan’, ‘katakanlah’, ‘katanya’, ‘ke’, ‘keadaan’, ‘kebetulan’, ‘kecil’, ‘kedua’, ‘keduanya’, ‘keinginan’, ‘kelamaan’, ‘kelihatan’, ‘kelihatannya’, ‘kelima’, ‘keluar’, ‘kembali’, ‘kemudian’, ‘kemungkinan’, ‘kemungkinannya’, ‘kenapa’, ‘kepada’, ‘kepadanya’, ‘kesampaian’, ‘keseluruhan’, ‘keseluruhannya’, ‘keterlaluan’, ‘ketika’, ‘khususnya’, ‘kini’, ‘kinilah’, ‘kira’, ‘kira-kira’, ‘kiranya’, ‘kita’, ‘kitalah’, ‘kok’, ‘kurang’, ‘lagi’, ‘lagian’, ‘lah’, ‘lain’, ‘lainnya’, ‘lalu’, ‘lama’, ‘lamanya’, ‘lanjut’, ‘lanjutnya’, ‘lebih’, ‘lewat’, ‘lima’, ‘luar’, ‘macam’, ‘maka’, ‘makanya’, ‘makin’, ‘malah’, ‘malahan’, ‘mampu’, ‘mampukah’, ‘mana’, ‘manakala’, ‘manalagi’, ‘masa’, ‘masalah’, ‘masalahnya’, ‘masih’, ‘masihkah’, ‘masing’, ‘masing-masing’, ‘mau’, ‘maupun’, ‘melainkan’, ‘melakukan’, ‘melalui’, ‘melihat’, ‘melihatnya’, ‘memang’, ‘memastikan’, ‘memberi’, ‘memberikan’, ‘membuat’, ‘memerlukan’, ‘memihak’, ‘meminta’, ‘memintakan’, ‘memisalkan’, ‘memperbuat’, ‘mempergunakan’, ‘memperkirakan’, ‘memperlihatkan’, ‘mempersiapkan’, ‘mempersoalkan’, ‘mempertanyakan’, ‘mempunyai’, ‘memulai’, ‘memungkinkan’, ‘menaiki’, ‘menambahkan’, ‘menandaskan’, ‘menanti’, ‘menanti-nanti’, ‘menantikan’, ‘menanya’, ‘menanyai’, ‘menanyakan’, ‘mendapat’, ‘mendapatkan’, ‘mendatang’, ‘mendatangi’, ‘mendatangkan’, ‘menegaskan’, ‘mengakhiri’, ‘mengapa’, ‘mengatakan’, ‘mengatakannya’, ‘mengenai’, ‘mengerjakan’, ‘mengetahui’, ‘menggunakan’, ‘menghendaki’, ‘mengibaratkan’, ‘mengibaratkannya’, ‘mengingat’, ‘mengingatkan’, ‘menginginkan’, ‘mengira’, ‘mengucapkan’, ‘mengucapkannya’, ‘mengungkapkan’, ‘menjadi’, ‘menjawab’, ‘menjelaskan’, ‘menuju’, ‘menunjuk’, ‘menunjuki’, ‘menunjukkan’, ‘menunjuknya’, ‘menurut’, ‘menuturkan’, ‘menyampaikan’, ‘menyangkut’, ‘menyatakan’, ‘menyebutkan’, ‘menyeluruh’, ‘menyiapkan’, ‘merasa’, ‘mereka’, ‘merekalah’, ‘merupakan’, ‘meski’, ‘meskipun’, ‘meyakini’, ‘meyakinkan’, ‘mirip’, ‘misal’, ‘misalkan’, ‘misalnya’, ‘mula’, ‘mulai’, ‘mulailah’, ‘mulanya’, ‘mungkin’, ‘mungkinkah’, ‘nah’, ‘naik’, ‘namun’, ‘nanti’, ‘nantinya’, ‘nyaris’, ‘nyatanya’, ‘oleh’, ‘olehnya’, ‘pada’, ‘padahal’, ‘padanya’, ‘pak’, ‘paling’, ‘panjang’, ‘pantas’, ‘para’, ‘pasti’, ‘pastilah’, ‘penting’, ‘pentingnya’, ‘per’, ‘percuma’, ‘perlu’, ‘perlukah’, ‘perlunya’, ‘pernah’, ‘persoalan’, ‘pertama’, ‘pertama-tama’, ‘pertanyaan’, ‘pertanyakan’, ‘pihak’, ‘pihaknya’, ‘pukul’, ‘pula’, ‘pun’, ‘punya’, ‘rasa’, ‘rasanya’, ‘rata’, ‘rupanya’, ‘saat’, ‘saatnya’, ‘saja’, ‘sajalah’, ‘saling’, ‘sama’, ‘sama-sama’, ‘sambil’,

‘sampai’, ‘sampai-sampai’, ‘sampaikan’, ‘sana’, ‘sangat’, ‘sangatlah’, ‘satu’, ‘saya’, ‘sayalah’, ‘se’, ‘sebab’, ‘sebabnya’, ‘sebagai’, ‘sebagaimana’, ‘sebagainya’, ‘sebagian’, ‘sebaik’, ‘sebaik-baiknya’, ‘sebaiknya’, ‘sebaliknya’, ‘sebanyak’, ‘sebegini’, ‘sebegitu’, ‘sebelum’, ‘sebelumnya’, ‘sebenarnya’, ‘seberapa’, ‘sebesar’, ‘sebetulnya’, ‘sebisanya’, ‘sebuah’, ‘sebut’, ‘sebutlah’, ‘sebutnya’, ‘secara’, ‘secukupnya’, ‘sedang’, ‘sedangkan’, ‘sedemikian’, ‘sedikit’, ‘sedikitnya’, ‘seenaknya’, ‘segala’, ‘segalanya’, ‘segera’, ‘seharusnya’, ‘sehingga’, ‘seingat’, ‘sejak’, ‘sejauh’, ‘sejenak’, ‘sejumlah’, ‘sekadar’, ‘sekadarnya’, ‘sekali’, ‘sekali-kali’, ‘sekalian’, ‘sekaligus’, ‘sekalipun’, ‘sekarang’, ‘sekarang’, ‘sekecil’, ‘seketika’, ‘sekiranya’, ‘sekitar’, ‘sekitarnya’, ‘sekurang-kurangnya’, ‘sekurangnya’, ‘sela’, ‘selain’, ‘selaku’, ‘selalu’, ‘selama’, ‘selama-lamanya’, ‘selamanya’, ‘selanjutnya’, ‘seluruh’, ‘seluruhnya’, ‘semacam’, ‘semakin’, ‘semampu’, ‘semampunya’, ‘semasa’, ‘semasih’, ‘semata’, ‘semata-mata’, ‘semaunya’, ‘sementara’, ‘semisal’, ‘semisalnya’, ‘sempat’, ‘semua’, ‘semuanya’, ‘semula’, ‘sendiri’, ‘sendirian’, ‘sendirinya’, ‘seolah’, ‘seolah-olah’, ‘seorang’, ‘sepanjang’, ‘sepantasnya’, ‘sepantasnyalah’, ‘seperlunya’, ‘seperti’, ‘sepertinya’, ‘sepihak’, ‘sering’, ‘seringnya’, ‘serta’, ‘serupa’, ‘sesaat’, ‘sesama’, ‘sesampai’, ‘sesegera’, ‘sesekali’, ‘seseorang’, ‘sesuatu’, ‘sesuatunya’, ‘sesudah’, ‘sesudahnya’, ‘setelah’, ‘setempat’, ‘setengah’, ‘seterusnya’, ‘setiap’, ‘setiba’, ‘setibanya’, ‘setidak-tidaknya’, ‘setidaknya’, ‘setinggi’, ‘seusai’, ‘sewaktu’, ‘siap’, ‘siapa’, ‘siapakah’, ‘siapapun’, ‘sini’, ‘sinilah’, ‘soal’, ‘soalnya’, ‘suatu’, ‘sudah’, ‘sudahkah’, ‘sudahlah’, ‘supaya’, ‘tadi’, ‘tadinya’, ‘tahu’, ‘tahun’, ‘tak’, ‘tambah’, ‘tambahnya’, ‘tampak’, ‘tampaknya’, ‘tandas’, ‘tandasnya’, ‘tanpa’, ‘tanya’, ‘tanyakan’, ‘tanyanya’, ‘tapi’, ‘tegas’, ‘tegasnya’, ‘telah’, ‘tempat’, ‘tengah’, ‘tentang’, ‘tentu’, ‘tentulah’, ‘tentunya’, ‘tepat’, ‘terakhir’, ‘terasa’, ‘terbanyak’, ‘terdahulu’, ‘terdapat’, ‘terdiri’, ‘terhadap’, ‘terhadapnya’, ‘teringat’, ‘teringat-ingat’, ‘terjadi’, ‘terjadilah’, ‘terjadinya’, ‘terkira’, ‘terlalu’, ‘terlebih’, ‘terlihat’, ‘termasuk’, ‘ternyata’, ‘tersampaikan’, ‘tersebut’, ‘tersebutlah’, ‘tertentu’, ‘tertuju’, ‘terus’, ‘terutama’, ‘tetap’, ‘tetapi’, ‘tiap’, ‘tiba’, ‘tiba-tiba’, ‘tidak’, ‘tidakkah’, ‘tidaklah’, ‘tiga’, ‘tinggi’, ‘toh’, ‘tunjuk’, ‘turut’, ‘tutur’, ‘tuturnya’, ‘ucap’, ‘ucapnya’, ‘ujar’, ‘ujarnya’, ‘umum’, ‘umumnya’, ‘ungkap’, ‘ungkapnya’, ‘untuk’, ‘usah’, ‘usai’, ‘waduh’, ‘wah’, ‘wahai’, ‘waktu’, ‘waktunya’, ‘walau’, ‘walaupun’, ‘wong’, ‘yaitu’, ‘yakin’, ‘yakni’, ‘yang’

LAMPIRAN 2: Daftar Kata Normalisasi

Sebelum	Sesudah
aktip	aktif
aktifitas	aktivitas
Apotik	Apotek
apotik	Apotek
analisa	analisis
azas	asas
Azas	asas
atlit	Atlet
Atlit	Atlet
Antri	antre
antri	antre
Atmosfir	Atmosfer
atmosfir	Atmosfer
Erobik	Aerobic
erobik	Aerobic
ahir	akhir
Ahir	akhir
antar-instansi	antarinstansi
Antar-instansi	antarinstansi
Baud	Baut
baud	Baut
Bis	Bus
bis	Bus
Berfikir	berpikir
berfikir	berpikir
Cabe	Cabai
cabe	Cabai
Cinderamata	Cenderamata
cinderamata	Cenderamata
Difinisi	definisi
difinisi	definisi
Disel	Diesel
disel	disel
Dipersilahkan	dipersilakan
dipersilahkan	dipersilakan
Dipindah	dipindahkan
dipindah	dipindahkan
Dollar	Dolar
dollar	Dolar
Daptar	Daftar
daptar	Daftar

Sebelum	Sesudah
Difinisi	Definisi
difinisi	Definisi
Depo	Depot
depo	Depot
Detil	detail
detil	detail
Diagnosa	Diagnosis
diagnosa	diagnosis
Differensial	diferensial
differensial	diferensial
Dipersilahkan	dipersilakan
dipersilahkan	dipersilakan
Disyahkan	disahkan
disyahkan	disahkan
Eksport	Ekspor
eksport	ekspor
Ekstrim	ekstrem
ekstrim	ekstrem
Ekwivalen	ekuivalen
ekwivalen	ekuivalen
Hembus	embus
hembus	embus
esei	esai
Pebruari	Februari
pebruari	Februari
Phiologi	Fiologi
phiologi	Fiologi
Filem	Film
filem	Film
Phisik	fisik
phisik	fisik
Photo	Foto
Photo	Foto
frekwensi	Frekuensi
Frekwensi	Frekuensi
Hapal	hafal
hapal	hafal
Hakekat	Hakikat
hakekat	Hakikat
Hirarki	Hierarki
Hipotesa	Hipotesis

Sebelum	Sesudah	Sebelum	Sesudah
Ijasah	Ijazah	danayanya	dana
Ihlas	Ikhlas	ngantre	mengantri
Himbau	Imbau	gak	tidak
Ilmiawan	Ilmuwan	haii	hai
Import	Impor	dbuka	di buka
Insyaf	Insaf	dy	dia
Hisap	Isap	sndirinya	sendirinya
Isteri	Istri	ndak	tidak
Ijin	Izin	audah	sudah
Jadual	Jadwal	kjlsn	kejelasan
Jenasah	Jenazah	ttg	tentang
Jendral	Jenderal	smp	sampai
Kaedah	Kaidah	nda	tidak
Karir	Karier	beda2	beda beda
Khutbah	Khotbah	dr	dari
Komplek	Kompleks	ga	tidak
Kondite	Konduite	makasih	terima kasih
konperensi	Konferesi	pa	apa
Konkrit	Konkret	bs	bisa
Konsepsionil	Konsepsional	d	di
Koordinir	Koordinasi	thn	tahun
Kwalitas	Kualitas	Trmn ksh	terima kasih
Kwantitas	Kuantitas	thks	terima kasih
Kwitansi	Kuitansi	melakukab	melakukan
Lobang	Lubang	tlg	tolong
Managemen	Manajemen	anak2	anak anak
Manager	Manajer	th	tahun
Memproklamirkan	Memproklamasikan	kmarin	kemarin
Menyolok	Mencolok	diknfirmasike	Dikonfirmasi ke
Mendifinisikan	Mendefinisikan	peswat	pesawat
Menterapkan	Menerapkan	untk	untuk
Menterjemahkan	Menerjemahkan	bokingnya	bookingnya
Melola	Mengelola	mrndapat	mendapat
Mengenyampingkan	Mengesampingkan	skitar	sekitar
Mengeritik	Mengkritik	sblm	sebelum
Mengobah	Mengubah	klo	kalau
merubah	Mengubah	beress	beres
Mensukseskan	Menyukseskan	bgtu	begitu
Musti	Mesti	tlfn	telepon
Metoda	Metode	kta	kata
Motip	Motif	tmn	teman