TOPIC MODELING PADA KOMUNITAS MARAH MARAH DI X

Disusun Oleh:

N a m a : Rizal Muhammad Ramli

NIM : 21523138

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM INDONESIA

2026

HALAMAN PENGESAHAN DOSEN PEMBIMBING

TOPIC MODELING PADA KOMUNITAS MARAH MARAH DI X

TUGAS AKHIR

Disusun Oleh:

N a m a : Rizal Muhammad Ramli

NIM : 21523138

Yogyakarta, 9 Februari 2026

Pembimbing,

(Chanifah Indah Ratnasari, S.Kom., M.Kom.)

HALAMAN PENGESAHAN DOSEN PENGUJI

TOPIC MODELING PADA KOMUNITAS MARAH MARAH DI X

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 9 Februari 2026

Tim Penguji

Chanifah Indah Ratnasari, S.Kom., M.Kom. _____

Anggota 1

Dr. Raden Teduh Dirgahayu, S.T., M.Sc. _____

Anggota 2

Moh. Idris, S.Kom., M.Kom. _____

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia

(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Rizal Muhammad Ramli

NIM : 21523138

Tugas akhir dengan judul:

TOPIC MODELING PADA KOMUNITAS MARAH MARAH DI X

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 14 Januari 2026

(Rizal Muhammad Ramli)

HALAMAN PERSEMBAHAN

Dengan syukur dan bangga, karya ini saya persembahkan kepada:

Kedua pahlawan saya,
Bapak Arief Ismet Ramli dan Mama Suprihatin
Atas segala yang sudah dikorbankan
Ini sedikit kontribusiku

Simbah Roimah
Yang selalu senantiasa membesarkan saya selayaknya orang tua sendiri

Eyza
Semoga karya ini dapat memotivasi ke depannya

HALAMAN MOTO

“Kalau hidup kurang terus sesekali lihat ke bawah”

- mama

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatruh,

Segala puji bagi Allah Subhanahu wa Ta'ala, dengan rahmat dan karunia-Nya penelitian dan penulisan tugas akhir yang berjudul "*Topic Modeling* pada Komunitas Marah Marah di X" ini dapat terselesaikan. Proses ini merupakan perjalanan yang sangat panjang dan juga menantang.

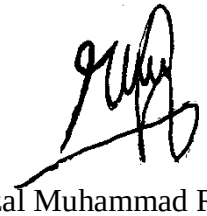
Penulis menyadari bahwa selesainya skripsi ini tidak terlepas dari bantuan, bimbingan, dan dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Bapak Prof. Fathul Wahid, S.T., M.Sc., Ph.D selaku Rektor Universitas Islam Indonesia.
2. Bapak Prof. Dr. Ir. Hari Purnomo, M.T., IPU, ASEAN.Eng selaku Dekan Fakultas Teknologi Industri.
3. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc. selaku Ketua Jurusan Informatika.
4. Bapak Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D. selaku Kepala Prodi Informatika.
5. Ibu Chanifah Indah Ratnasari, S.Kom., M.Kom. selaku dosen pembimbing yang selalu sabar dan bijak selama proses penyusunan skripsi ini.
6. Bapak Arief Ismet Ramli dan Mama Suprihatin selaku kedua orang tua hebat penulis yang selalu mendukung dalam bentuk materi, non materi, serta doa yang tidak pernah putus.
7. Muhammad Rheyza Ramli selaku adik saya yang selalu memberikan waktu dan energi.
8. Terima kasih kepada Hanin Nadhifa, yang kehadirannya memberikan kekuatan dan ketenangan emosional yang sangat berarti bagi penulis selama ini.
9. Ravel, Daffa, dan Pradika atas persahabatan tulus yang melampaui sekadar kata-kata. Terima kasih telah menemani dalam setiap situasi, memberikan dukungan moral, dan menjadi pengingat untuk tetap bahagia di tengah segala tekanan.
10. Indra, Pandu, Haikal, dan Javier teman-teman seperjuangan di bawah satu atap kos yang telah menjadi keluarga kedua. Terima kasih atas keceriaan yang kalian hadirkan setiap harinya, serta bantuan-bantuan yang sangat berarti di masa-masa sulit penyelesaian tugas akhir ini.

Penyelesaian skripsi ini merupakan awal dari proses pembelajaran yang menanti di depan. Harapan terbesar penulis adalah semoga karya ini dapat memicu diskusi dan berkontribusi pada analisis teks. Semoga penelitian ini memberi manfaat kepada pembaca dan menjadi pengembangan penelitian selanjutnya.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Yogyakarta, 14 Januari 2026



(Rizal Muhammad Ramli)

SARI

Media sosial X kini menjadi ruang publik utama bagi penggunanya untuk mengekspresikan emosi, salah satunya melalui "Komunitas MARAH MARAH". Komunitas ini menjadi wadah bagi pengguna untuk menumpahkan kekesalan mereka secara terbuka tanpa batasan yang kaku. Penelitian ini bertujuan untuk mengidentifikasi dan mengelompokkan topik-topik dominan yang menjadi sumber kemarahan dalam komunitas tersebut menggunakan metode *Latent Dirichlet Allocation* (LDA). Data penelitian dikumpulkan melalui teknik *web scraping* dan diproses melalui serangkaian tahapan *preprocessing* data, termasuk penanganan duplikasi data, *tokenizing*, hingga *stemming*. Berdasarkan pengujian model, terbentuk 9 topik optimal dengan nilai *coherence score* sebesar 0,5086. Hasil analisis menunjukkan bahwa kemarahan pengguna terbagi ke dalam berbagai dimensi, mulai dari masalah hubungan personal dan keluarga, kewaspadaan terhadap modus penipuan, kendala teknis dan layanan publik, hingga kelelahan akibat tekanan kerja, dan gangguan lingkungan tempat tinggal.

Kata kunci: Kemarahan; *Topic Modeling*; X; LDA

GLOSARIUM

Coherence Score	Metrik evaluasi yang digunakan untuk mengukur kualitas topik dengan cara menghitung tingkat kemiripan makna (semantik) antar kata yang menyusun topik tersebut.
Gensim	*library* berbasis Python yang dikembangkan secara khusus untuk memfasilitasi tugas-tugas *Natural Language Processing*, termasuk pemodelan topik dan analisis kesamaan teks.
Preprocessing	Tahapan awal pengolahan data yang bertujuan untuk mentransformasi data mentah yang kotor (bising/tidak konsisten) menjadi format yang bersih dan terstruktur agar siap untuk dianalisis lebih lanjut.
Scraping Data	Teknik ekstraksi informasi secara otomatis dari situs web untuk mengumpulkan data spesifik dan menyimpannya dalam format yang dapat diolah.
Topic Modeling	Metode statistik atau *machine learning* yang digunakan untuk mengidentifikasi pola tema abstrak yang tersembunyi di dalam sekumpulan dokumen teks.

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi dan komunikasi yang pesat telah mengubah secara fundamental cara manusia berinteraksi, berbagi informasi, dan mengekspresikan diri. Media sosial telah bertransformasi dari sekadar alat jejaring sosial menjadi ruang publik digital yang kompleks, di mana dinamika sosial, politik, dan psikologis masyarakat terekam secara *real-time*. Platform X, atau yang sebelumnya dikenal sebagai Twitter, memegang peranan vital dalam ekosistem ini sebagai medium berbasis teks pendek yang memfasilitasi penyebaran informasi secara cepat dan masif (Hastuti et al., 2025). Signifikansi platform ini di Indonesia terlihat dari data Dataloka.id (2025) yang menempatkan Indonesia di urutan keempat dunia dengan total 23,76 juta pengguna, menyusul Amerika Serikat di posisi pertama (100,03 juta), Jepang di posisi kedua (71,03 juta), dan India di posisi ketiga (24,52 juta). Karakteristik platform ini yang memungkinkan anonimitas, kecepatan respons, dan jangkauan audiens yang luas menjadikannya wadah yang ideal untuk meluapkan ekspresi emosi yang spontan dan sering kali intens (Azzahrani et al., 2024).

Dalam konteks psikologi siber dan sosiologi digital, fenomena ekspresi emosi di media sosial, khususnya emosi negatif seperti kemarahan, kekecewaan, dan frustrasi, menjadi subjek kajian yang semakin mendesak. Emosi negatif memiliki karakteristik viralitas yang unik. Penelitian menunjukkan bahwa konten yang memicu kemarahan cenderung menyebar lebih cepat dan memicu keterlibatan yang lebih tinggi dibandingkan konten bernada netral atau positif (Han et al., 2023). Algoritma media sosial, yang dirancang untuk memaksimalkan waktu pengguna di platform, sering kali secara tidak sengaja memperkuat penyebaran emosi ini, menciptakan apa yang disebut sebagai *echo chambers* atau ruang gema di mana kemarahan divalidasi dan diamplifikasi secara kolektif (Al Fatih et al., 2024).

Fenomena ini termanifestasi secara nyata dalam pembentukan komunitas-komunitas daring yang spesifik. Salah satu entitas yang menarik perhatian dalam lanskap media sosial Indonesia adalah komunitas ekspresi emosi negatif di platform X, atau yang biasa dikenal oleh penggunanya sebagai "Komunitas MARAH MARAH". Komunitas ini bukan sekadar kumpulan individu yang meluapkan emosi, melainkan sebuah konstruksi sosial digital yang berfungsi sebagai kanal pelepasan ekspresi emosional. Di dalam komunitas ini, pengguna merasa memiliki ruang aman untuk menumpahkan keluhan, sumpah serapah, dan narasi

kemarahan yang mungkin dianggap tabu atau tidak pantas jika diungkapkan dalam interaksi sosial tatap muka atau di platform yang lebih berorientasi visual dan pencitraan diri (Sarifudin & Zuhri, 2025).

Meskipun fenomena ini sangat mencolok, pemahaman akademik mengenai struktur internal dari diskursus kemarahan ini masih terbatas. Sebagian besar penelitian terdahulu dalam ranah analisis teks media sosial Indonesia cenderung berfokus pada analisis sentimen (*sentiment analysis*) yang bertujuan untuk mengklasifikasikan opini ke dalam polaritas positif, negatif, atau netral. Sementara analisis sentimen mampu memberikan gambaran makro mengenai "suhu" emosional publik, metode ini gagal dalam menjelaskan mengapa pengguna marah, apa yang menjadi objek kemarahan mereka, dan bagaimana narasi kemarahan tersebut dikonstruksi secara tematik. Tanpa pemahaman mengenai konteks dan topik, analisis terhadap komunitas semacam ini hanya akan menghasilkan data kuantitatif yang dangkal tanpa wawasan sosiologis yang mendalam.

Kesenjangan pengetahuan inilah yang mendorong perlunya penerapan metode *text mining* yang lebih canggih, yaitu *topic modeling*. *Topic modeling* memungkinkan peneliti untuk mengekstraksi tema-tema tersembunyi dari sekumpulan dokumen besar tanpa perlu melakukan pelabelan manual sebelumnya (*unsupervised learning*) (Dalimunthe & Putri, 2024). Di antara berbagai algoritma *topic modeling*, *Latent Dirichlet Allocation* (LDA) telah terbukti sebagai metode yang *robust* (tangguh) dan efektif, terutama dalam menangani korpus data yang besar dan beragam. LDA bekerja dengan asumsi probabilistik bahwa setiap dokumen adalah campuran dari berbagai topik, dan setiap topik adalah distribusi dari kata-kata tertentu (Blei et al., 2003). Pendekatan ini sangat relevan untuk membedah percakapan di "Komunitas MARAH MARAH", di mana satu *tweet* keluhan saja dapat memuat campuran isu pekerjaan, hubungan personal, hingga kritik sosial.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, permasalahan utama yang menjadi fokus penelitian ini dapat dirumuskan ke dalam beberapa pertanyaan penelitian sebagai berikut:

a. Bagaimana strategi *preprocessing* yang dapat mendukung pemodelan topik untuk mengatasi karakteristik data yang sangat tidak terstruktur pada “Komunitas MARAH MARAH” di X?

b. Topik apa saja yang berhasil diidentifikasi dalam “Komunitas MARAH MARAH” di X dengan metode *Latent Dirichlet Allocation*?

1.3 Batasan Masalah

Batasan masalah pada penelitian ini adalah:

a. Data yang digunakan pada penelitian ini merupakan data tekstual yang diperoleh dari “Komunitas MARAH MARAH” di X.
b. Pengambilan data dilakukan dalam rentang waktu 21 Februari 2025 hingga 31 Juli 2025.
c. Data yang digunakan adalah *tweet* yang berbahasa Indonesia.

1.4 Tujuan Penelitian

Tujuan penelitian ini adalah:

a. Mengidentifikasi dan memodelkan topik-topik utama yang muncul dalam “Komunitas MARAH MARAH” di X.
b. Menganalisis distribusi tema dominan yang memicu emosi negatif dan menginterpretasikan konteks narasi pengguna dalam Komunitas MARAH MARAH.

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah:

a. Memberikan pengetahuan kepada pembaca dan pemangku kebijakan, dalam hal ini pemerintahan divisi atau bidang terkait, mengenai topik-topik yang sering dibicarakan dalam “Komunitas MARAH MARAH” di X
b. Menjadi berkontribusi dalam memperdalam pemahaman mengenai perilaku digital masyarakat, memberikan rujukan nyata bagi perancangan kebijakan moderasi platform yang adaptif, serta menyediakan landasan ilmiah bagi studi lanjutan terkait dampak media sosial terhadap kesehatan mental pengguna.

1.6 Sistematika Penulisan

Laporan skripsi ini disusun secara sistematis dalam 5 bab utama untuk memudahkan pembacaan dan pemahaman alur penelitian:

BAB I PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, serta sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini berisi landasan teori yang memuat uraian mengenai konsep-konsep dasar yang dijadikan acuan dalam merumuskan solusi atas permasalahan penelitian, serta telaah terhadap berbagai penelitian sebelumnya yang relevan atau memiliki kesamaan topik dengan studi ini.

BAB III METODOLOGI PENELITIAN

Bab ini berisi tahapan penelitian yang dilakukan oleh peneliti seperti pengambilan data, *preprocessing*, TF-IDF, *topic modeling* menggunakan LDA, analisis hasil, dan interpretasi topik.

BAB IV HASIL DAN PEMBAHASAN

Bab ini membahas penerapan metode yang digunakan beserta hasil yang diperoleh.

BAB V KESIMPULAN DAN SARAN

Bab ini menjadi bagian akhir yang menyajikan kesimpulan dan saran berdasarkan penelitian tugas akhir yang telah dilakukan.

BAB II
LANDASAN TEORI

2.1 Platform Media Sosial X

Platform X, yang sebelumnya dikenal secara global dengan nama Twitter, adalah platform media sosial berbasis mikroblog yang memungkinkan penggunanya untuk mengirim dan membaca pesan teks pendek yang disebut "*tweet*" atau "cuitan" (Azhar & Bakry, 2025). Platform ini telah berkembang menjadi salah satu media sosial paling berpengaruh di dunia, termasuk di Indonesia, di mana ia berfungsi sebagai ruang publik digital untuk berbagi pemikiran, berita, dan opini secara *real-time*.

Salah satu karakteristik utama X adalah keterbukaan dan kecepatan penyebaran informasinya. Pengguna dapat berinteraksi melalui berbagai mekanisme seperti *retweet*, *like*, dan *reply*, yang memungkinkan sebuah topik menjadi viral dalam waktu singkat (Azhar & Bakry, 2025). Selain itu, X memiliki keunggulan unik di mana pengguna dapat memposting konten yang hanya berupa teks tanpa kewajiban melampirkan multimedia, menjadikannya medium yang sangat efisien untuk ekspresi verbal dan diskusi yang padat.

Relevansi X dalam penelitian ini semakin kuat dengan adanya fitur komunitas (*communities*). Fitur ini diluncurkan untuk memfasilitasi pengguna dalam membentuk grup diskusi yang lebih terfokus berdasarkan kesamaan minat, hobi, atau bahkan emosi tertentu, terpisah dari lini masa utama (*timeline*) publik. Ketersediaan data tekstual yang melimpah dan terpusat dalam fitur komunitas ini menjadikan X sebagai sumber data yang sangat kaya (*rich data source*) untuk analisis *text mining* dan pemodelan topik.

2.2 *Text Mining*

Text mining, yang sering juga disebut sebagai *Text Data Mining* (TDM) atau *Knowledge Discovery in Text* (KDT), adalah proses penggalian informasi dan pola yang berharga, baru, dan tidak diketahui sebelumnya dari sekumpulan data teks yang tidak terstruktur atau semi-terstruktur (Shamshiri et al., 2024). Berbeda dengan *Information Retrieval* (IR) yang bertujuan untuk menemukan dokumen yang relevan berdasarkan kata kunci, *text mining* bertujuan untuk menganalisis isi dari dokumen-dokumen tersebut untuk menemukan tren, hubungan, atau struktur yang lebih dalam (Khan et al., 2020).

Proses *text mining* mengintegrasikan teknik-teknik dari berbagai disiplin ilmu, termasuk *Information Retrieval*, *Natural Language Processing* (NLP), *Data Mining*, *Machine Learning*,

dan Statistik (Hassani et al., 2020). Tahapan standar dalam *text mining* meliputi pengumpulan data, *text preprocessing*, transformasi teks atau ekstraksi fitur (*text transformation/feature extraction*), pemilihan fitur (*feature selection*), penambangan data/pola (*pattern discovery*), dan evaluasi/interpretasi hasil (*interpretation/evaluation*) (Kovalchuk et al., 2022).

2.3 *Topic Modeling*

Topic modeling atau pemodelan topik adalah salah satu teknik pembelajaran mesin tanpa pengawasan (*unsupervised machine learning*) yang bertujuan untuk menemukan struktur tematik tersembunyi (*latent thematic structure*) dalam sekumpulan dokumen teks yang besar (Blei et al., 2003). Secara sederhana, algoritma ini bekerja dengan memindai dokumen, mendeteksi pola kata dan frasa yang sering muncul bersamaan (*co-occurrence*), dan mengelompokkannya ke dalam "topik" yang merepresentasikan inti pembahasan dari dokumen-dokumen tersebut (Yijia, 2024). Metode ini sangat berguna untuk mengorganisir, memahami, meringkas, dan memprediksi isi dari koleksi dokumen yang terlalu besar untuk dibaca secara manual. Beberapa metode *topic modeling* yang paling populer dan banyak digunakan dalam penelitian adalah:

a. *Latent Semantic Analysis* (LSA)

   LSA adalah salah satu metode pemodelan topik paling awal yang menggunakan pendekatan aljabar linier, khususnya dekomposisi nilai singular (*Singular Value Decomposition* - SVD). LSA bekerja dengan mereduksi dimensi matriks istilah-dokumen (*term-document matrix*) untuk menangkap hubungan semantik antar kata. Meskipun efisien, LSA memiliki kelemahan dalam menangkap polisemi (kata yang memiliki makna ganda) dan hasil topiknya terkadang sulit diinterpretasikan secara langsung karena representasi matematisnya dapat bernilai negatif (Shen & Ho, 2020).

b. *Probabilistic Latent Semantic Analysis* (pLSA)

   Dikembangkan sebagai perbaikan dari LSA, pLSA menggunakan pendekatan probabilistik daripada aljabar. pLSA memodelkan setiap kata dalam dokumen sebagai sampel dari model campuran (*mixture model*) (Pradhan et al., 2022). Namun, pLSA rentan terhadap *overfitting* karena jumlah parameter model bertambah seiring dengan bertambahnya jumlah dokumen.

c. *Latent Dirichlet Allocation* (LDA)

   Latent Dirichlet Allocation (LDA) adalah *unsupervised* model yang mengambil unit data utama (Gupta & Katarya, 2021). Secara konseptual, LDA memandang dokumen

sebagai kumpulan campuran topik (*mixture of topics*), di mana setiap topik didefinisikan sebagai distribusi probabilitas dari sekumpulan kata (Wang et al., 2020). Tujuan utama dari algoritma ini adalah untuk memodelkan struktur topik tersembunyi (*latent*) dari korpus dokumen yang besar (Culasso et al., 2023). LDA menerapkan pendekatan Bayesian dengan menambahkan distribusi Dirichlet sebagai *prior* pada distribusi topik dalam dokumen. Penambahan *Dirichlet prior* ini memungkinkan model untuk melakukan generalisasi yang lebih baik dan menjaga konsistensi distribusi topik, bahkan pada data yang belum pernah dilatih sebelumnya. Oleh karena itu, LDA dianggap secara efektif mampu mereduksi dimensi data teks sekaligus mempertahankan makna semantiknya (Gupta & Katarya, 2021).

d. *Non Negative Matrix Factorization* (NMF)

Non Negative Matrix Factorization (NMF) adalah metode analisis multivariat dan teknik reduksi dimensi yang berbasis pada aljabar linear. Dalam konteks penambangan teks (*text mining*), NMF bekerja dengan memfaktorkan matriks input non-negatif menjadi dua matriks aproksimasi dengan dimensi yang lebih rendah (Wang & Zhang, 2023).

Karakteristik fundamental yang membedakan NMF dari metode faktorisasi lain adalah penerapan kendala non-negatif (*non-negativity constraint*). Kendala ini mengharuskan setiap elemen dalam matriks hasil faktorisasi bernilai lebih besar atau sama dengan nol. Secara teoritis, batasan ini memaksa algoritma untuk mempelajari representasi berbasis bagian (*parts-based representation*), di mana data direkonstruksi hanya melalui penambahan komponen (aditif) tanpa adanya pengurangan (Nugumanova et al., 2022).

Implikasi dari sifat aditif tersebut menjadikan hasil dekomposisi NMF memiliki tingkat interpretabilitas semantik yang tinggi (Yang et al., 2023). Karena tidak memuat nilai negatif, setiap topik dapat dipandang sebagai himpunan kata-kata yang secara positif membangun makna dokumen tersebut. Hal ini menjadikan NMF metode yang sangat relevan untuk menangani *dataset* dengan struktur yang jarang (*sparse*), karena mampu mengisolasi pola fitur yang terdistribusi secara lokal dalam dokumen (Lu et al., 2023).

2.4 *Preprocessing*

Kualitas hasil dari algoritma *machine learning* sangat bergantung pada kualitas data yang dimasukkan. Data teks dari media sosial sangat "kotor" karena mengandung banyak elemen yang tidak relevan secara semantik (Çetin & Yıldız, 2022). *Text preprocessing* adalah

serangkaian teknik untuk membersihkan dan menstandarisasi teks agar siap diolah. Tahapan-tahapan krusial dalam *preprocessing* meliputi:

a. *Remove Punctuation*

Tahap ini bertujuan untuk menghapus *noise* seperti tag HTML, URL, karakter spesial, *username* (@), dan tagar (#) yang dapat mengganggu analisis frekuensi kata (Çetin & Yıldız, 2022).

b. *Case Folding*

Tahap ini bertujuan untuk mengonversi seluruh karakter menjadi huruf kecil (*lowercase*). Ini penting agar komputer memperlakukan kata "Marah", "MARAH", dan "marah" sebagai satu entitas yang sama (Dalimunthe & Putri, 2024).

c. *Tokenizing*

Tahap ini bertujuan untuk memecah aliran teks menjadi unit-unit terkecil yang disebut token (biasanya berupa kata). Ini adalah dasar dari analisis leksikal (Gurusamy & Kannan, 2014).

d. *Normalization*

Proses mengubah kata-kata yang tidak baku menjadi bentuk bakunya (Singh & Singh, 2020). Dalam konteks bahasa Indonesia di media sosial, ini sangat vital karena tingginya penggunaan singkatan seperti: yg → yang, bgt → banget, dan variasi ejaan. Tanpa normalisasi, frekuensi kata akan terpecah ke dalam banyak variasi ejaan yang berbeda.

e. *Stopwords Removal*

Stopwords removal merupakan proses menghilangkan kata-kata umum yang memiliki frekuensi kemunculan sangat tinggi namun membawa sedikit makna semantik (Negara, 2025). Menghapus *stopwords* dapat mengurangi dimensi data secara signifikan dan membiarkan algoritma fokus pada kata-kata konten.

f. *Stemming*

Proses untuk mengubah kata berimbuhan menjadi kata dasarnya (Siswandi et al., 2021). Dalam bahasa Indonesia, algoritma yang umum digunakan adalah algoritma Nazief & Adriani yang diimplementasikan dalam pustaka Sastrawi.

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

Dalam pengolahan data teks atau *text mining*, representasi data ke dalam format numerik atau vektor merupakan prasyarat fundamental agar algoritma komputasi dapat melakukan

analisis matematis. Salah satu pendekatan pembobotan yang menjadi standar dalam *Information Retrieval* dan pengolahan bahasa alami adalah *Term Frequency-Inverse Document Frequency* (TF-IDF). TF-IDF didefinisikan sebagai ukuran statistik yang digunakan untuk mengevaluasi seberapa penting suatu kata (*term*) terhadap sebuah dokumen dalam kumpulan dokumen atau korpus. Metode ini bertujuan untuk mengatasi kelemahan pada representasi sederhana yang memperlakukan semua kata dengan bobot yang setara. TF-IDF bekerja dengan memberikan bobot tinggi pada kata-kata yang relevan namun mengurangi bobot pada kata-kata umum yang sering muncul di banyak dokumen (Sundaram et al., 2021). Rumus TF-IDF ditunjukkan pada persamaan (1)

$$W_{t,d} = TF_{t,d} \times IDF_t \tag{1}$$

Keterangan

$W_{t,d}$ = Bobot kata *t* terhadap dokumen *d*.

$TF_{t,d}$ = Jumlah kemunculan kata *t* dalam dokumen *j*.

IDF_t = *Inverse Document Frequency*

2.5.1 *Term Frequency* (TF)

Term frequency digunakan untuk mengukur seberapa sering sebuah kata muncul dalam sebuah dokumen. Semakin sering muncul, semakin penting kata tersebut dalam merepresentasikan dokumen itu (Sundaram et al., 2021). Rumus TF ditunjukkan pada persamaan (2).

$$TF_{t,d} = \begin{cases} 1 + \log_{10}(f_{t,d}), f_{t,d} > 0 \\ 0, f_{t,d} = 0 \end{cases} \tag{2}$$

Keterangan

t = Jumlah kata

d = Dokumen

$TF_{t,d}$ = Jumlah kemunculan kata *t* dalam dokumen *j*

2.5.2 *Inverse Document Frequency*

Inverse Document Frequency (IDF) digunakan untuk mengukur seberapa informatif sebuah kata. Kata-kata yang muncul di hampir semua dokumen dianggap kurang informatif dibandingkan kata yang hanya muncul di sedikit dokumen. IDF memberikan bobot lebih tinggi pada kata-kata yang jarang muncul di korpus (Sundaram et al., 2021). Rumus IDF ditunjukkan pada persamaan (3).

$$IDF_t = \log\left(\frac{N}{DF_t}\right) \tag{3}$$

Keterangan

IDF_t = *Inverse Document Frequency*

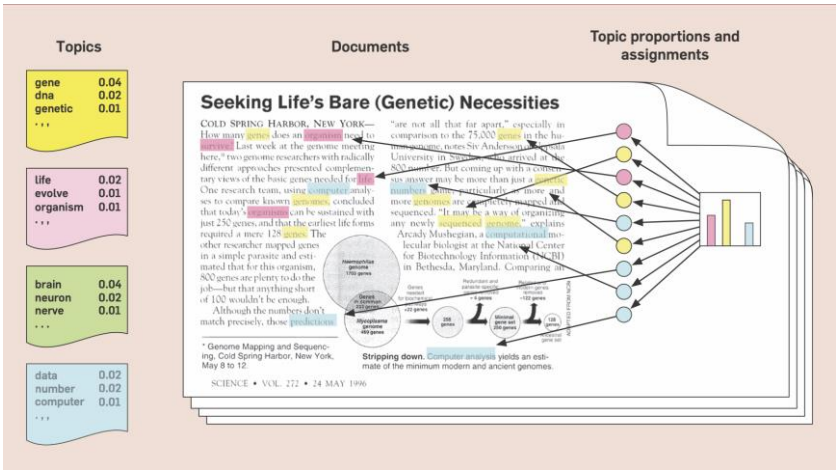
N = Jumlah dokumen

DF_t = jumlah dokumen yang mengandung kata *t*

2.6 *Latent Dirichlet Allocation*

Dalam ranah analisis teks dan *Natural Language Processing* (NLP), metode yang dianggap sebagai *gold standard* untuk menemukan struktur tematik tersembunyi adalah *Latent Dirichlet Allocation* (LDA). Teknik ini diklasifikasikan sebagai model generatif probabilistik yang beroperasi secara *unsupervised*. Santoso et al. (2022) menjelaskan bahwa LDA bekerja dengan cara memetakan dokumen ke dalam daftar topik, dan memetakan topik ke dalam daftar kata, sehingga memungkinkan komputer untuk mengelompokkan dokumen berdasarkan kemiripan pola katanya.

Secara konseptual, LDA memiliki intuisi bahwa dokumen bukanlah entitas tunggal, melainkan sebuah campuran acak dari berbagai topik laten (*latent topics*). Blei et al. (2003) dalam penelitian fundamentalnya menekankan bahwa meskipun topik-topik ini tidak diketahui sebelumnya, keberadaannya dapat ditelusuri kembali melalui observasi statistik terhadap frekuensi kemunculan kata dalam korpus data.

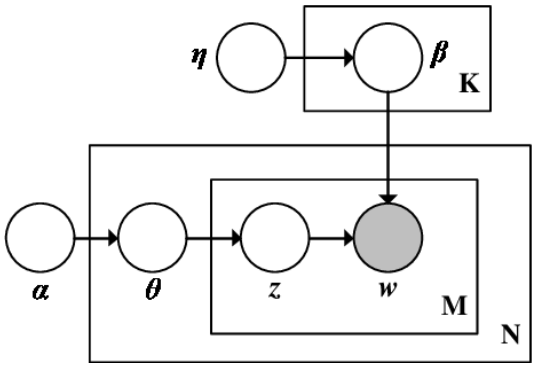


Gambar 2.1 Konsep LDA

Sumber: Blei (2012)

Seperti yang diilustrasikan pada Gambar 2.1, model LDA berangkat dari asumsi bahwa setiap dokumen tersusun atas campuran berbagai topik dengan proporsi tertentu. Setiap topik itu sendiri merupakan distribusi probabilitas dari kata-kata. Dalam proses pembentukan setiap dokumen dalam koleksi, terdapat dua tahapan utama:

1. Memilih distribusi topik secara acak.
2. Untuk setiap kata yang ada di dalam dokumen tersebut:
   a. Memilih satu topik secara acak dari distribusi topik yang telah ditentukan pada langkah pertama.
   b. Memilih kata secara acak dari distribusi kosakata yang sesuai dengan topik tersebut.



Gambar 2.2 *Plate Notation* Model LDA

Sumber : Anastasiu & Tagarelli (2017)

Elemen-elemen dalam notasi yang ditunjukkan pada Gambar 2.2 didefinisikan sebagai berikut:

η = Parameter *Dirichlet* yang mengontrol distribusi kata per topik.

β = distribusi kata terhadap topik dalam corpus

K = Kumpulan topik

$\mathcal{W}$ = Kata

M = Kumpulan kata

N = Kumpulan dokumen

$\mathcal{Z}$ = Topik *index assignment*

θ = dokumen

α = Parameter *Dirichlet* yang mengontrol distribusi topik per dokumen.

Secara matematis, proses generatif pada LDA dirumuskan melalui probabilitas gabungan (*joint probability distribution*) dari seluruh variabel acak dalam model. Mengacu pada formulasi Blei et al. (2003), probabilitas gabungan ini ditunjukkan pada persamaan (4).

$$p(w, z, \theta, \varphi|\alpha, \beta) = p(\theta|\alpha) \cdot p(z|\theta) \cdot p(\varphi|\beta) \cdot p(w|z, \varphi) \tag{4}$$

Persamaan probabilitas gabungan di atas terbentuk dari empat komponen distribusi utama yang saling berkaitan. Komponen pertama adalah distribusi topik pada dokumen (θ) yang dibangkitkan dari parameter α, di mana probabilitasnya dihitung menggunakan rumus sebagaimana terlihat pada Persamaan (5).

$$p(\theta|\alpha) = \frac{\Gamma(\sum_i \alpha_i)}{\coprod_i \Gamma(\alpha_i)} \prod_{k=1}^{\mathrm{K}} \theta_k^{\alpha_k - 1} \tag{5}$$

Selanjutnya, komponen kedua menjelaskan penentuan topik (z) untuk setiap kata dalam dokumen. Penugasan topik ini didasarkan pada distribusi θ yang telah terbentuk sebelumnya dan mengikuti distribusi multinomial seperti yang dirumuskan pada persamaan (6):

$$p(z|\theta) = \prod_{m=1}^{M} \prod_{n=1}^{N_m} \theta_{m,k}^{n_{m,k}} \tag{6}$$

Distribusi Di sisi lain, komponen ketiga memodelkan distribusi kata terhadap topik (φ). Distribusi ini diambil dari distribusi Dirichlet yang dikontrol oleh parameter β, sebagaimana ditunjukkan pada Persamaan (7):

$$p(\varphi|\beta) = \prod_{k=1}^{K} \frac{\Gamma(\sum_v \beta_v)}{\prod_v \Gamma(\beta_v)} \prod_{v=1}^{V} \varphi_{k,v}^{\beta_{k,v}-1} \tag{7}$$

Terakhir, komponen keempat merepresentasikan probabilitas kemunculan kata aktual (w). Variabel ini merupakan satu-satunya data yang diobservasi dan nilainya bergantung pada topik z yang terpilih serta distribusi kata-topik φ. Hubungan ini diformulasikan dalam Persamaan (8):

$$p(w|z, \varphi) = \prod_{k=1}^{K} \prod_{v=1}^{V} \varphi_{k,v}^{n_{k,v}} \tag{8}$$

Melalui persamaan-persamaan di atas, LDA melakukan proses inferensi statistik (seperti *Gibbs Sampling*) untuk mengestimasi nilai variabel-variabel tersembunyi tersebut berdasarkan data teks yang tersedia.

2.7 *Coherence Score*

Tantangan utama dalam LDA adalah menentukan jumlah topik (K) yang optimal, karena ini adalah parameter yang harus ditentukan di awal. Untuk mengevaluasi kualitas topik yang dihasilkan, digunakan metrik *Topic Coherence*. *Topic Coherence* mengukur tingkat kemiripan semantik antara kata-kata dengan probabilitas tertinggi dalam satu topik (Blair et al., 2020). Asumsinya adalah, jika sebuah topik berkualitas baik, maka kata-kata kunci di dalamnya haruslah kata-kata yang sering muncul bersamaan dalam konteks yang sama, sehingga membentuk konsep yang koheren. Skor koherensi yang sering digunakan adalah `C_v`. Nilai `C_v` berkisar antara 0 hingga 1, semakin tinggi nilainya, semakin baik kualitas topik tersebut untuk diinterpretasikan manusia.

2.8 Python

Python adalah bahasa pemrograman tingkat tinggi berbasis interpreter yang berorientasi objek dan memiliki semantik dinamis. Diciptakan oleh Guido van Rossum dan pertama kali dirilis pada tahun 1991, Python telah menjadi salah satu bahasa pemrograman paling populer di dunia, khususnya dalam domain *Data Science* dan *Artificial Intelligence*. Keunggulan utama Python terletak pada sintaksisnya yang sederhana, bersih, dan mudah dibaca (*readable*), yang menyerupai bahasa Inggris manusia. Hal ini memungkinkan peneliti untuk lebih fokus pada pemecahan masalah algoritma daripada berkutat dengan kompleksitas penulisan kode. Selain itu, Python didukung oleh ekosistem pustaka (*libraries*) pihak ketiga yang sangat luas dan matang, yang menyediakan fungsi-fungsi siap pakai untuk berbagai keperluan analisis data. Dalam penelitian ini, pustaka-pustaka kunci yang digunakan meliputi Pandas untuk manipulasi dan analisis data tabular, NumPy untuk komputasi numerik dan operasi *array*, NLTK dan Sastrawi untuk pemrosesan bahasa alami (NLP), serta Gensim untuk pemodelan topik. Fleksibilitas dan kekayaan fitur inilah yang menjadikan Python sebagai alat utama yang sangat efektif untuk mengolah data teks tidak terstruktur dalam jumlah besar (Muis & Muhammad, 2023).

2.9 *Jupyter Notebook*

Jupyter Notebook adalah sebuah aplikasi web sumber terbuka (*open-source*) yang memungkinkan pengguna untuk membuat dan membagikan dokumen yang berisi kode langsung (*live code*), persamaan matematika, visualisasi, dan teks naratif penjelas. Nama "Jupyter" sendiri merupakan akronim dari tiga bahasa pemrograman inti yang didukungnya: Julia, Python, dan R. Dalam konteks penelitian *data science* dan *text mining*, Jupyter Notebook berfungsi sebagai lingkungan pengembangan interaktif yang sangat vital. Arsitekturnya yang berbasis sel (*cell-based*) memungkinkan peneliti untuk menulis dan mengeksekusi kode secara bertahap, serta langsung melihat hasilnya (seperti grafik atau tabel) tepat di bawah kode tersebut. Hal ini mendukung paradigma *literate programming*, di mana kode, hasil, dan dokumentasi narasinya terintegrasi dalam satu dokumen tunggal. Fitur ini sangat memudahkan proses eksplorasi data (*Exploratory Data Analysis*), *debugging*, dan visualisasi hasil pemodelan topik secara iteratif. Lebih lanjut, Jupyter Notebook menjamin transparansi dan reproduksibilitas (*reproducibility*) penelitian, karena seluruh alur kerja analisis terekam dengan

jelas dan dapat dijalankan ulang oleh peneliti lain untuk memverifikasi hasil temuan (Silaparasetty, 2020).

2.10 Penelitian Terdahulu

Dalam bab ini, penulis membahas penelitian-penelitian terdahulu yang relevan sebagai landasan pertimbangan dalam pemilihan metode. Pembahasan ini mencakup perbandingan kinerja, penerapan algoritma, serta kelebihan dan kekurangan dari masing-masing metode *topic modeling*. Untuk memastikan sumber referensi yang digunakan bersifat mutakhir dan relevan dengan perkembangan teknologi saat ini, literatur yang dipilih dibatasi pada rentang publikasi tahun 2020–2025. Proses pencarian literatur dilakukan secara sistematis melalui basis data akademik dan pengindeks jurnal bereputasi, antara lain Google Scholar, IEEE Xplore, ResearchGate, dan ScienceDirect. Adapun penelusuran referensi dilakukan menggunakan kombinasi kata kunci yang relevan, meliputi *topic modeling*, twitter, emosi, serta *topic modeling on short text*. Ringkasan penelitian terdahulu tersebut disajikan dalam Tabel 2.1.

Tabel 2.1 Perbandingan Metode *Topic Modeling*

Sumber	Metode	Hasil
Mohammed & Al-Augby (2020)	LDA, LSA	Penelitian ini bertujuan secara spesifik untuk membandingkan kinerja metode *topic modeling* LDA dan LSA dalam mengelompokkan kata menjadi topik, guna menjadikan kata-kata teratas (*keywords*) sebagai fitur dalam klasifikasi *e-book* berbasis *full-text*. Menggunakan *dataset* yang terdiri dari 300 dokumen *full-text* dari 10 bidang ilmu yang bersumber dari *e-library* Bookboon, evaluasi dilakukan menggunakan metrik *Topic Coherence* (UMass dan UCI/PMI). Berdasarkan hasil pengujian pada *dataset* tersebut menggunakan metrik UCI, LDA menunjukkan kinerja terbaik pada jumlah 20 topik dengan skor 0,592179, sedikit lebih unggul dibandingkan LSA yang mencapai performa terbaiknya pada 10 topik dengan skor 0,577303.
Mifrah (2020)	LDA, NMF	Studi ini bertujuan membandingkan kinerja metode LDA dan NMF berdasarkan metrik *topic coherence*

		menggunakan dataset 13.000 sitasi COVID-19 (2019–2020) dari CORD-19 yang dikumpulkan via *Sketch Engine* dalam format CSV. Setelah melalui tahapan *preprocessing* hasil evaluasi menunjukkan bahwa LDA secara eksplisit memiliki nilai *coherence* yang lebih tinggi dibandingkan NMF. Keunggulan LDA terlihat dari pembentukan topik yang secara semantik lebih "rapi" dan relevan, berbeda dengan NMF yang cenderung memunculkan banyak artefak sitasi atau metadata sehingga menghasilkan topik yang kurang fokus secara ilmiah.
Kamdan et al. (2022)	LDA	Penelitian ini bertujuan untuk mengevaluasi kesesuaian tema penelitian mahasiswa dengan lima profil lulusan Program Studi Teknik Informatika Universitas Nusa Putra secara efisien, mengingat evaluasi manual terhadap ribuan judul membutuhkan sumber daya yang besar. Dengan menggunakan metode *Latent Dirichlet Allocation* (LDA) pada 159 judul skripsi periode 2020-2022 yang telah melalui tahap *preprocessing* studi ini berhasil mengidentifikasi lima klaster topik dominan, yaitu *Software Development*, *Security & Network*, *Information Systems*, *Monitoring & Server*, serta *IoT & Mobile*. Visualisasi menggunakan *Word Cloud* dan *Scatter Plot* (pyLDAvis) memperjelas distribusi topik tersebut, yang menyimpulkan bahwa tema penelitian mahasiswa telah selaras dengan konsentrasi program studi.
Zoya et al. (2021)	NMF, LDA, LSA, dan pLSA	Penelitian ini berfokus pada perbandingan kinerja algoritma *topic modeling* menggunakan *dataset* berskala besar yang terdiri dari 0,89 juta *tweet* berbahasa Urdu. Melalui metode ekstraksi fitur *Bag of Words* dan TF-IDF (*n-gram*) serta strategi inovatif penggabungan teks berbasis korelasi tagar (*hashtag pooling*), studi ini menemukan perbedaan signifikan dalam efektivitas model

		tergantung pada panjang teks. Hasil eksperimen menunjukkan bahwa NMF khususnya dengan fitur TF-IDF *bi-gram* secara konsisten mengungguli model probabilistik lain pada level *tweet* individu (teks pendek) karena kemampuannya menangani *data sparsity* dan *noise*, yang juga dikonfirmasi melalui validasi manusia dengan skor *Cohen's Kappa* 0,8. Sebaliknya, LDA menunjukkan kinerja yang kurang optimal pada *tweet* individu namun mengalami peningkatan performa yang signifikan ketika diterapkan pada dokumen semu panjang (*aggregated tweets*).
Ali et al. (2021)	LDA	Penelitian ini mengembangkan pendekatan hibrida yang mengintegrasikan *topic modeling* menggunakan LDA dan analisis sentimen berbasis leksikon untuk mengekstraksi wawasan pariwisata kota Marrakech (Maroko) dari 39.200 ulasan TripAdvisor. Algoritma LDA, yang jumlah topiknya dioptimalkan berdasarkan *Coherence Score*, berhasil memetakan empat dimensi utama pengalaman wisatawan. Analisis sentimen mengonfirmasi dominasi kepuasan pengunjung dengan 74,4% ulasan positif, di mana pengujian model menunjukkan bahwa algoritma TextBlob mencapai akurasi tertinggi sebesar 77,3%, mengungguli VADER (72,6%) dan JST (69,6%).
Rkia et al. (2024)	LDA, NMF, dan LSA	Penelitian ini bertujuan mengotomatisasi pemantauan kesejahteraan psikososial masyarakat dengan menganalisis lebih dari 10.000 entri data tekstual dari departemen kesehatan menggunakan teknik *Natural Language Processing* (NLP). Dalam studi komparatif antara algoritma LDA, NMF, dan LSA, metode *Latent Dirichlet Allocation* (LDA) terbukti memegang peran paling efektif untuk mengekstrak struktur topik dari data tidak terstruktur. Hasil evaluasi kuantitatif menunjukkan bahwa LDA dengan fitur *unigram* mencapai kinerja

		terbaik dengan skor koherensi 0,59, mengungguli NMF dan LSA.

Berdasarkan Berdasarkan tinjauan terhadap literatur yang disajikan dalam Tabel 2.1, pemetaan analisis (*framing analysis*) dilakukan untuk mengidentifikasi tren penggunaan metode, perbandingan performa, serta pengaruh karakteristik data terhadap hasil penelitian. Dari keseluruhan studi yang ditinjau, terlihat pola dominasi penggunaan metode *Latent Dirichlet Allocation* (LDA), baik sebagai metode utama maupun sebagai standar pembanding (*baseline*). LDA diterapkan secara luas di berbagai domain, mulai dari klasifikasi *e-book* akademik, analisis sitasi medis, hingga evaluasi tema skripsi. Konsistensi penggunaan ini mengindikasikan bahwa LDA memiliki fleksibilitas tinggi dan dianggap sebagai standar dalam pemodelan topik probabilistik, sementara metode lain seperti LSA, pLSA, dan NMF lebih sering diposisikan sebagai variabel pembanding untuk menguji efektivitas LDA dalam berbagai konteks.

Dalam bingkai perbandingan kinerja secara kuantitatif dan kualitatif, mayoritas penelitian menunjukkan keunggulan LDA dalam menghasilkan topik yang koheren dan bermakna. Sebagai bukti, penelitian Mohammed & Al-Augby (2020) mendemonstrasikan bahwa pada data *full-text*, LDA mencapai skor. *Topic Coherence* (UCI) sebesar 0.592, sedikit mengungguli LSA yang mencapai skor 0.577. Keunggulan ini diperkuat oleh studi pada data kesehatan mental yang menunjukkan LDA dengan fitur *unigram* mencapai skor koherensi terbaik 0.59, melampaui NMF dan LSA. Selain unggul dalam metrik koherensi, LDA juga terbukti menghasilkan topik yang secara semantik lebih "rapi" dan relevan, berbeda dengan NMF yang pada kasus tertentu, seperti dalam studi data sitasi COVID-19, cenderung memunculkan artefak atau metadata yang kurang fokus secara ilmiah.

Namun demikian, analisis terhadap karakteristik data mengungkapkan adanya dikotomi performa berdasarkan panjang teks yang menjadi tantangan tersendiri. Pada karakteristik data teks pendek (*short text*) seperti media sosial, studi Zoya et al. (2021) menemukan bahwa metode NMF, khususnya dengan fitur TF-IDF *bi-gram*, mampu mengungguli model probabilistik dalam menangani *data sparsity* dan *noise* dengan capaian skor *Cohen's Kappa* 0.8. Temuan ini menggarisbawahi bahwa teks pendek memerlukan penanganan khusus dibandingkan dokumen panjang.

Meskipun terdapat tantangan pada teks pendek, literatur yang sama juga mencatat bahwa performa LDA mengalami peningkatan yang signifikan ketika diterapkan pada *tweet* yang telah

diagregasi menjadi dokumen semu panjang. Hal ini memberikan landasan teoretis yang kuat bagi penelitian ini untuk tetap menggunakan LDA pada data Twitter. Dengan demikian, dapat disimpulkan bahwa meskipun NMF kompetitif untuk data *sparse*, LDA tetap menjadi pilihan metode yang lebih superior dalam aspek interpretabilitas semantik dan koherensi topik, asalkan didukung dengan strategi *preprocessing* dan agregasi data yang tepat untuk mengatasi keterbatasan panjang teks. Untuk mempertegas landasan pemilihan metode, penulis merangkum perbandingan teknis kelebihan dan kekurangan masing-masing algoritma dalam Tabel 2.2.

Tabel 2.2 Kekurangan dan Kelebihan Metode

Metode	Kelebihan	Kekurangan
LDA	Menghasilkan topik dengan nilai koherensi tertinggi dan interpretasi semantik yang paling jelas pada dokumen.	Membutuhkan proses penentuan jumlah topik yang optimal.
NMF	Efektif dan efisien dalam menangani *noise* serta *data sparsity* pada *dataset* berupa teks pendek.	Topik yang dihasilkan sering kali kurang fokus karena cenderung memunculkan metadata atau artefak yang tidak relevan secara semantik.
LSA	Memiliki kompleksitas komputasi yang rendah dan mampu menangkap hubungan sinonim antar kata dengan cukup baik.	Akurasi pemodelan topiknya relatif terbatas karena kesulitan menangkap makna ganda (polisemi) dengan skor koherensi yang umumnya paling rendah.
pLSA	Mampu menangani masalah polisemi (kata bermakna ganda) lebih baik daripada LSA karena menggunakan landasan statistik yang kuat.	Sangat rentan terhadap masalah *overfitting* dan tidak memiliki mekanisme untuk menangani dokumen baru (*unseen data*) tanpa melatih ulang model.

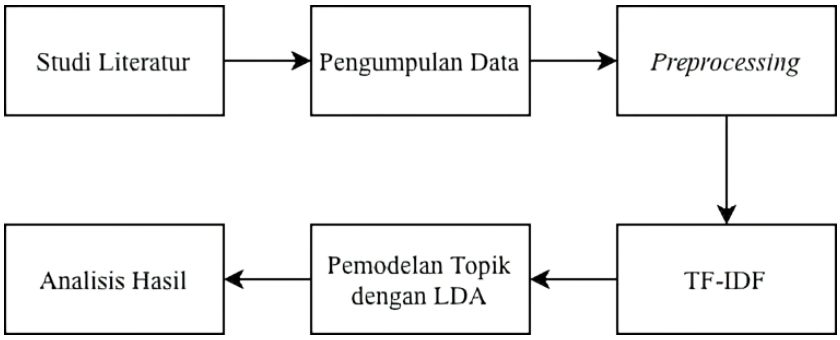
Berdasarkan analisis perbandingan kelebihan dan kekurangan pada Tabel 2.2 serta tinjauan terhadap penelitian terdahulu, penelitian ini menetapkan untuk menggunakan metode *Latent Dirichlet Allocation* (LDA). Keputusan ini didasarkan pada kemampuan LDA yang

terbukti lebih unggul dalam menghasilkan topik dengan struktur semantik yang jelas dan nilai koherensi (*coherence score*) yang tinggi dibandingkan metode lain seperti LSA, pLSA, maupun NMF. Selain itu, LDA dipilih karena kemampuannya mengatasi masalah *overfitting* yang sering terjadi pada pLSA, sehingga lebih handal untuk mengungkap pola topik yang tersembunyi secara akurat dari *dataset* yang digunakan dalam penelitian ini.

BAB III
METODE PENELITIAN

3.1 Tahapan Penelitian

Tahapan Penelitian ini dilaksanakan melalui serangkaian tahapan sistematis yang terdiri dari enam langkah utama, sebagaimana diilustrasikan pada Gambar 3.1.



Gambar 3.1 Tahapan Penelitian

Tahapan alur penelitian dimulai dengan studi literatur, di mana dilakukan penelusuran dan kajian terhadap teori serta metode yang relevan untuk membangun landasan penelitian yang kokoh. Setelah landasan teori terbentuk, tahap selanjutnya adalah Pengumpulan Data. Pada tahap ini, data berupa *tweet* diambil dari komunitas di X (sebelumnya Twitter) dan disimpan dalam format csv. Data mentah yang telah terkumpul kemudian memasuki tahap *preprocessing*. Tahap ini bertujuan untuk membersihkan dan mempersiapkan data agar siap diolah. Proses ini meliputi beberapa langkah krusial, yaitu *duplicate removal*, *remove punctuation*, *case folding*, *tokenizing*, *normalization*, *stopwords removal*, serta *stemming*. Setelah data teks menjadi bersih, dilakukan proses TF-IDF. Proses ini berfungsi untuk mengubah representasi teks menjadi vektor numerik serta memberikan pembobotan untuk mengukur tingkat kepentingan suatu kata (*term*) dalam dokumen. Data yang telah memiliki bobot kemudian diolah pada tahap pemodelan topik dengan LDA untuk mengekstraksi topik-topik tersembunyi. Tahap terakhir adalah analisis hasil, di mana topik yang terbentuk diinterpretasikan dan dianalisis sehingga makna dalam dapat dipahami dengan lebih baik.

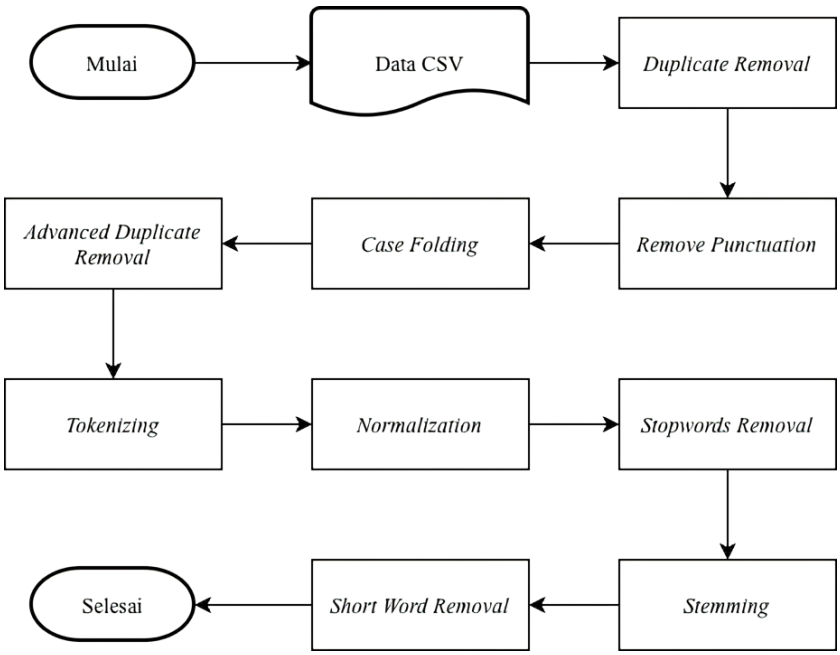
3.2 Pengumpulan Data

Sumber data dalam penelitian ini berasal dari X, secara spesifik difokuskan pada unggahan dalam “Komunitas MARAH MARAH”. Proses pengumpulan data dilakukan

menggunakan teknik *web scraping*, sebuah metode ekstraksi informasi secara otomatis dari laman web. Dalam pelaksanaannya, instrumen yang digunakan adalah bahasa pemrograman Python dengan memanfaatkan *library* Selenium. Pemilihan Selenium didasarkan pada efektivitasnya dalam mensimulasikan interaksi pengguna secara nyata pada peramban web, seperti melakukan navigasi dan pengguliran halaman secara otomatis untuk memuat konten yang dinamis. Atribut data yang diekstraksi dari proses ini meliputi tanggal unggahan, nama pengguna (*username*), dan isi *tweet*. Seluruh data yang berhasil dikumpulkan kemudian disimpan secara terstruktur dalam format CSV.

3.3 *Preprocessing*

Preprocessing merupakan tahap fundamental dalam analasis teks yang bertujuan untuk membersihkan data. Tahapan ditampilkan pada Gambar 3.2.



Gambar 3.2 Tahapan *Preprocessing*

Preprocessing merupakan langkah yang krusial untuk menghilangkan kelebihan data dan menjaga konsistensi data. Pada penelitian ini berikut beberapa tahap *preprocessing*.

a. *Duplicate Removal*

Proses penghapusan duplikat (*duplicate removal*) dilakukan untuk menangani redundansi data yang muncul akibat metode pengumpulan data yang dilakukan secara bertahap. Dalam proses *scraping* yang dilakukan dalam beberapa sesi atau periode waktu,

sering kali terjadi irisan data (*overlapping*) di mana data yang sudah terunduh pada sesi sebelumnya dapat terambil kembali pada sesi berikutnya. Oleh karena itu, tahap ini bertujuan untuk mengeliminasi data ganda tersebut guna memastikan setiap entitas data dalam *dataset* bersifat unik dan valid.

b. *Remove Punctuation*

Tahap ini berfokus pada penghapusan elemen-elemen yang dianggap sebagai *noise* dan tidak memberikan kontribusi signifikan terhadap makna teks. Proses ini meliputi penghapusan elemen non-alfabet seperti tautan (URL), *username mentions*, karakter tanda baca, *hashtag*, angka, serta spasi berlebih. Tujuannya adalah menyisakan teks bersih yang hanya berisi informasi relevan.

c. *Case Folding*

Case folding bertujuan untuk menyeragamkan bentuk huruf dalam *dataset* menjadi huruf kecil (*lowercase*). Langkah ini menjaga konsistensi data, misalnya agar kata "Halo", "HALO", dan "halo" dianggap sebagai entitas kata yang sama.

d. *Advanced Duplicate Removal*

Berbeda Tahap *Advanced Duplicate Removal* merupakan prosedur penyaringan data ulang yang dilakukan setelah dataset melewati proses pembersihan awal, khususnya *remove punctuation* dan *case folding*. Langkah ini secara metodologis sangat diperlukan karena adanya keterbatasan sistem komputasi dalam mendeteksi redundansi pada data mentah yang masih mengandung variasi karakter unik. Secara visual, banyak entri data yang tampak identik bagi peneliti, namun sistem (Microsoft Excel) mengategorikannya sebagai data unik akibat perbedaan kecil pada penggunaan huruf kapital, spasi ganda, maupun keberadaan simbol tertentu.

Oleh karena itu, proses ini dilakukan kembali untuk memastikan bahwa seluruh residu duplikasi yang tersamarkan oleh format teks dapat tereliminasi sepenuhnya setelah data diseragamkan ke dalam bentuk huruf kecil dan bersih dari simbol. Implementasi tahap ini bertujuan untuk menjamin integritas data dan mencegah terjadinya bias frekuensi pada tahap pemodelan topik menggunakan LDA. Dengan memastikan setiap dokumen dalam korpus benar-benar unik, algoritma dapat berfokus pada ekstraksi pola tema yang objektif tanpa terdistorsi oleh kemunculan data repetitif yang tidak informatif.

e. *Tokenization*

Tokenization adalah proses pemecahan kalimat atau *string* teks menjadi potongan-potongan kata yang berdiri sendiri, yang disebut sebagai token. Tahap ini merupakan

jembatan utama untuk mengubah struktur kalimat menjadi kumpulan kata yang dapat dihitung frekuensinya.

f. *Normalization*

Normalisasi Tahap normalisasi berfungsi untuk menyeragamkan variasi leksikal dengan mengubah kata-kata tidak baku, singkatan, *slang*, maupun kesalahan penulisan (*typo*) menjadi bentuk baku. Proses ini dilakukan dengan menggunakan kamus normalisasi kustom yang disusun melalui pendekatan *data-driven*. Kata-kata tidak baku yang muncul dalam korpus data diidentifikasi terlebih dahulu, kemudian bentuk bakunya ditetapkan dengan menjadikan Kamus Besar Bahasa Indonesia (KBBI) dan pedoman Ejaan Bahasa Indonesia yang Disempurnakan (EYD) sebagai rujukan utama validasi. Mekanisme ini bertujuan untuk mereduksi *noise* akibat variasi penulisan (misalnya: "gk", "gak", "ngga" menjadi "tidak") sehingga frekuensi kata dapat dihitung secara akurat pada tahap pemodelan.

g. *Stopwords Removal*

Tahap ini bertujuan untuk mengeliminasi kata-kata umum yang sering muncul namun tidak memiliki makna yang kuat (*common words*), seperti kata hubung ("dan", "yang", "dari") atau kata ganti. Daftar *stopwords* yang digunakan disesuaikan dengan *library* NLTK untuk Bahasa Indonesia.

h. *Stemming*

Stemming merupakan proses mentransformasi kata berimbuhan menjadi kata dasarnya dengan menghilangkan awalan, akhiran, dan sisipan. Misalnya, kata "memberikan" dan "pemberian" akan diubah menjadi satu kata dasar yang sama yaitu "beri". Hal ini efektif untuk mengurangi variasi kata tanpa mengubah konteks utama.

i. *Short Words Removal*

Tahap ini bertujuan untuk mengeliminasi token kata yang memiliki jumlah karakter kurang dari 3 huruf (len < 3). Proses ini difokuskan untuk membersihkan residu singkatan tidak baku yang terdiri dari satu atau dua karakter (seperti 'yg', 'ga', 'ok') yang dianggap sebagai *noise*. Penetapan ambang batas ini dilakukan dengan mempertimbangkan preservasi informasi, di mana kata-kata dengan panjang tiga karakter tetap dipertahankan. Hal ini bertujuan untuk mencegah hilangnya kata-kata substantif berstruktur pendek yang memiliki kontribusi signifikan terhadap pembentukan topik.

3.4 *Term Frequency-Inverse Document Frequency*

Setelah pembersihan data melalui *preprocessing*, data teks dikonversi menjadi bentuk numerik melalui proses pembobotan kata menggunakan TF-IDF (*Term Frequency-Inverse Document Frequency*). Metode ini dipilih karena kemampuannya dalam mengevaluasi relevansi kata, di mana kata kunci yang krusial akan diberikan bobot lebih besar dibandingkan kata-kata umum.

Secara matematis, metrik ini menggabungkan nilai *Term Frequency* (TF) yaitu intensitas kemunculan kata dalam sebuah dokumen dengan *Inverse Document Frequency* (IDF), yang merupakan kalkulasi dari kelangkaan kata tersebut dalam keseluruhan *dataset*. Hasil perkalian kedua faktor ini akan menghasilkan nilai bobot yang menunjukkan seberapa penting peran kata tersebut dalam mendefinisikan topik dokumen yang bersangkutan.

3.5 *Topic Modeling* dengan *Latent Dirichlet Allocation* (LDA)

Pemodelan topik dilakukan menggunakan metode *Latent Dirichlet Allocation* (LDA). LDA merupakan model statistik generatif yang memungkinkan sekumpulan pengamatan (dalam hal ini, *tweet*) dijelaskan oleh kelompok-kelompok yang tidak teramati (topik) yang menjelaskan mengapa beberapa bagian data memiliki kemiripan. Pada tahap ini, matriks bobot dari proses TF-IDF digunakan sebagai input untuk melatih model LDA. Proses ini melibatkan estimasi dua distribusi probabilitas utama:

1. Distribusi topik dalam dokumen: Menunjukkan seberapa besar proporsi suatu topik dibahas dalam sebuah *tweet*.
2. Distribusi kata dalam topik: Menunjukkan probabilitas kata-kata tertentu muncul dalam suatu topik.

Dalam implementasinya, penelitian ini menggunakan *library* Gensim pada bahasa pemrograman Python. Tantangan utama dalam metode LDA adalah penentuan parameter jumlah topik yang tepat. Oleh karena itu, penelitian ini akan melakukan *hyperparameter tuning* dengan menguji variasi nilai dan parameter (alpha) serta (beta) untuk mendapatkan model dengan performa terbaik yang diukur menggunakan metrik *Coherence Score*.

3.6 Analisis Hasil

Pada tahap analisis hasil, fokus utama diarahkan pada evaluasi mendalam terhadap topik-topik yang telah terbentuk dari pemodelan guna memastikan kesesuaiannya dengan tujuan penelitian. Proses interpretasi dilakukan secara kualitatif dengan menelaah deretan kata yang

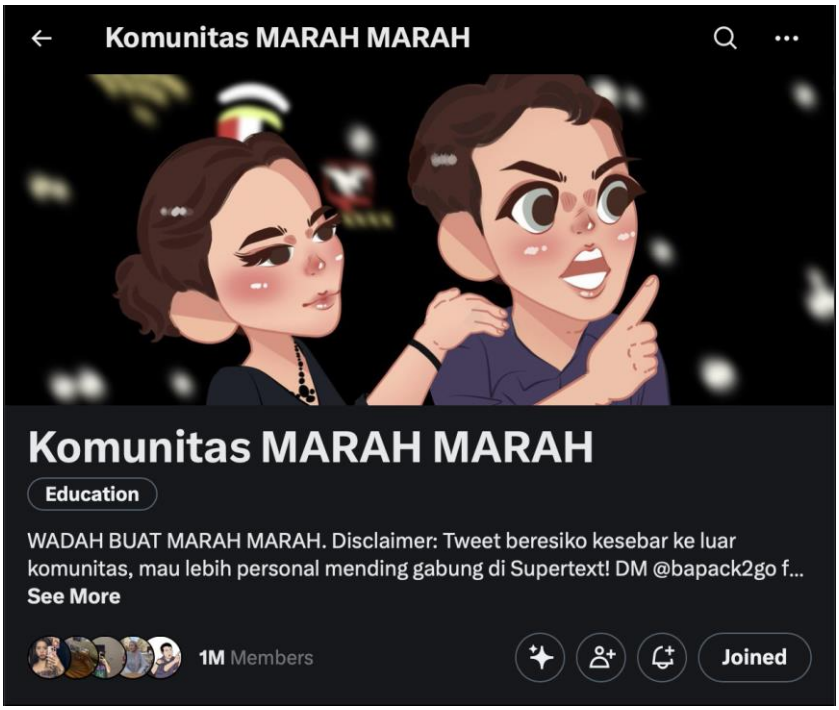
memiliki bobot tertinggi pada setiap topik untuk memahami makna tematik yang terkandung di dalamnya. Analisis ini tidak hanya mencakup pengelompokan makna dari kata kunci utama, tetapi juga pengamatan terhadap keterkaitan antar topik serta validasi terhadap tingkat koherensi yang dihasilkan.

BAB IV
HASIL DAN PEMBAHASAN

4.1 Pengambilan Data

Tahap pengambilan data merupakan langkah fundamental dalam penelitian ini untuk memastikan kualitas *dataset* yang akan dianalisis. Proses pengambilan data dilakukan menggunakan metode *web scraping*, yaitu teknik ekstraksi data otomatis dari laman web. Penelitian ini memanfaatkan ekosistem pemrograman Python yang dijalankan melalui *Integrated Development Environment* (IDE) Visual Studio Code.

Objek pengambilan data difokuskan pada platform media sosial X (sebelumnya dikenal sebagai Twitter). Secara spesifik, penelitian ini menargetkan data percakapan dari sebuah komunitas daring bernama "Komunitas MARAH MARAH". Komunitas ini dipilih sebagai subjek penelitian karena memiliki basis keanggotaan yang masif, yakni mencapai sekitar satu juta anggota, yang mengindikasikan tingginya volume interaksi dan variasi data tekstual yang tersedia. Tangkapan layar halaman Komunitas MARAH MARAH pada X ditunjukkan pada Gambar 4.1.



Gambar 4.1 Komunitas MARAH MARAH

Pengambilan Pengambilan data pada penelitian ini menggunakan *library* Selenium dengan *webdriver* berbasis Google Chrome. Selenium berfungsi sebagai alat otomatisasi peramban yang memungkinkan program untuk meniru interaksi manusia, seperti menavigasi halaman, menggulirkan layar, hingga mengidentifikasi elemen HTML secara dinamis.

Tahap pertama adalah inisialisasi lingkungan kerja dan otentikasi. Sebagaimana ditampilkan pada Gambar 4.2, kode program diawali dengan konfigurasi *webdriver* berbasis Chrome. Pada blok kode awal, fungsi `driver.get()` digunakan untuk menavigasi sistem ke halaman *login* X. Mengingat kompleksitas keamanan platform yang sering memunculkan tantangan CAPTCHA, strategi otentikasi dilakukan dengan pendekatan hibrida, di mana program memberikan jeda waktu (*time.sleep*) selama 30 detik agar peneliti dapat memasukkan kredensial secara manual dengan aman sebelum program mengambil alih kendali navigasi menuju URL target komunitas.

```
from selenium.webdriver.chrome.service import Service
import time
import csv
from selenium import webdriver
from selenium.webdriver.common.by import By
from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC

def remove_newlines(text: str) -> str:
    return text.replace('\n', ' ').replace('\r', ' ')


def scrape_community():
    driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()))
    driver.maximize_window()

    driver.get("https://twitter.com/login")
    print("Silakan login secara manual di jendela Chrome. Tunggu 30 detik...")
    time.sleep(30)

    community_url = "https://x.com/i/communities/1562271278744354816"
    driver.get(community_url)
    time.sleep(10)

    WebDriverWait(driver, 10).until(
```

```
        EC.presence_of_element_located((By.XPATH,                "//article[@data-
testid='tweet']"))
    )
```

Gambar 4.2 Inisialisasi WebDriver dan Otentikasi Manual

Setelah akses ke halaman komunitas berhasil didapatkan, tahapan selanjutnya adalah mekanisme ekstraksi data yang bersifat dinamis. Karena konten pada X dimuat menggunakan sistem *infinite scroll*, data tidak tersedia sekaligus melainkan harus dimuat secara bertahap. Pada Gambar 4.3, diimplementasikan fungsi bantu `collect_tweets()` yang bertugas memindai elemen HTML menggunakan Xpath (`//article[@data-testid='tweet']`).

Untuk menjamin kualitas data, diterapkan logika deduplikasi awal menggunakan struktur data himpunan (*set*) bernama `seen_ids`. Setiap *tweet* memiliki ID unik pada URL-nya, sebelum data diekstrak, program memeriksa apakah ID tersebut sudah ada dalam `seen_ids`. Jika sudah ada, data akan diabaikan. Mekanisme ini berfungsi mencegah terjadinya duplikasi data yang terjadi dalam satu sesi eksekusi program. Atribut yang diambil meliputi tanggal (*datetime*), nama pengguna (*username*), dan *tweet* (*content*), yang kemudian disimpan sementara dalam senarai (*list*) items.

```
    SCROLL_PAUSE = 4
    MAX_SCROLL = 500
    seen_ids = set()
    all_tweets = []

    def collect_tweets():
        items = []
        articles      =      driver.find_elements(By.XPATH,      "//article[@data-
testid='tweet']")
        for art in articles:
            try:
                url                  =                  art.find_element(By.XPATH,
".//time/ancestor::a").get_attribute("href")
                tweet_id = url.rstrip("/").split("/")[-1]
                if tweet_id in seen_ids:
                    continue  # # Lewati jika sudah diproses
                seen_ids.add(tweet_id)

                tweet_date = art.find_element(By.XPATH, ".//time")\
```

```
                                .get_attribute("datetime")

                username = art.find_element(
                    By.XPATH,
                    ".//div[@data-testid='User-Name']//span[contains(text(),
'@')]"
                ).text.strip()

                content_elem    =    art.find_element(By.XPATH,    ".//div[@data-
testid='tweetText']")
                tweet_content = content_elem.text.strip() if content_elem else ""

                items.append((tweet_date, username, tweet_content))
            except Exception as e:
                print(f"Error processing tweet: {e}")
        return items

    all_tweets.extend(collect_tweets())
```

Gambar 4.3 Logika Ekstraksi Elemen

Tahap terakhir adalah manajemen navigasi halaman dan penyimpanan data. Guna mengoptimalkan proses ini pada sistem *infinite scroll*, skrip dikonfigurasi dengan dua parameter kunci, yaitu `SCROLL_PAUSE = 4` sebagai jeda waktu tunggu pemuatan konten baru setelah setiap pengguliran, serta `MAX_SCROLL = 500` sebagai batas maksimal iterasi untuk menjaga stabilitas memori operasional.

Sebagaimana diperlihatkan pada Gambar 4.4, program menjalankan perulangan (*loop*) sebanyak `MAX_SCROLL` untuk menyimulasikan perilaku pengguna menggulir layar ke bawah menggunakan perintah `window.scrollTo`. Setiap kali pengguliran terjadi, program akan memanggil kembali fungsi ekstraksi untuk mengambil data baru yang muncul. Setelah seluruh proses iterasi selesai, data yang terkumpul disimpan ke dalam format CSV. Pada blok kode penyimpanan, diterapkan fungsi *preprocessing* sederhana `remove_newlines` untuk membersihkan karakter baris baru yang dapat merusak struktur tabel. Penulisan *file* menggunakan modul `csv.writer` dengan parameter `QUOTE_ALL`, yang berfungsi mengapit seluruh elemen data dengan tanda kutip. Teknik ini sangat krusial untuk memastikan bahwa karakter koma atau simbol khusus di dalam isi *tweet* tidak dianggap sebagai pemisah kolom, sehingga format data tetap terjaga integritasnya saat dibuka di aplikasi pengolah angka.

```
    all_tweets.extend(collect_tweets())

    last_height = driver.execute_script("return document.body.scrollHeight")
    for _ in range(MAX_SCROLL):
        driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")
        time.sleep(SCROLL_PAUSE)

        new_height = driver.execute_script("return document.body.scrollHeight")
        if new_height == last_height:
            print("Tidak ada konten baru. Stop.")
            break
        last_height = new_height

        all_tweets.extend(collect_tweets())

    driver.quit()

    output_file = "12-08-2025.csv"
    with open(output_file, "w", newline="", encoding="utf-8") as f:
        writer = csv.writer(f, quoting=csv.QUOTE_ALL)
        writer.writerow(["date", "username", "content"])

        for date, user, text in all_tweets:
            clean_row = (
                remove_newlines(date),
                remove_newlines(user),
                remove_newlines(text),
            )
            writer.writerow(clean_row)

    print(f"Selesai! Total tweet unik: {len(all_tweets)}")

if __name__ == "__main__":
    scrape_community()
```

Gambar 4.4 Mekanisme *Infinite Scroll* dan Penyimpanan Data ke .csv

Karena batasan teknis Selenium dalam melakukan *scroll* yang terlalu jauh dalam satu sesi eksekusi, pengambilan data tidak dapat dilakukan secara sekaligus untuk seluruh rentang waktu penelitian. Oleh sebab itu, proses pengambilan data dilakukan secara bertahap dalam beberapa periode waktu yang berbeda. Total keseluruhan data yang berhasil dikumpulkan dan siap untuk langkah *preprocessing* berjumlah 75.032 data. Dokumen *dataset* hasil akuisisi ini dapat dilihat secara lebih rinci pada bagian Lampiran A.

Tabel 4.1 Periode *Scraping* Data

Periode	Tanggal	Jumlah
1	21 Februari – 28 Februari 2025	4.287
2	12 Maret 2025	719
3	30 April – 6 Mei 2025	26.895
4	9 Mei – 14 Mei 2025	17.380
5	24 Mei 2025	7.886
6	4 Juni 2025	7.883
7	9 Juni 2025	7.622
8	31 Juli	2.360
	Jumlah	75.032

4.2 *Preprocessing*

Tahapan *Preprocessing* merupakan fase krusial untuk mentransformasi data mentah menjadi format yang terstruktur dan berkualitas. *Dataset* mentah hasil proses *scraping* umumnya masih memiliki tingkat kegaduhan (*noise*) yang tinggi, tidak konsisten, dan belum baku. *Preprocessing* yang dilakukan dalam penelitian ini adalah sebagai berikut.

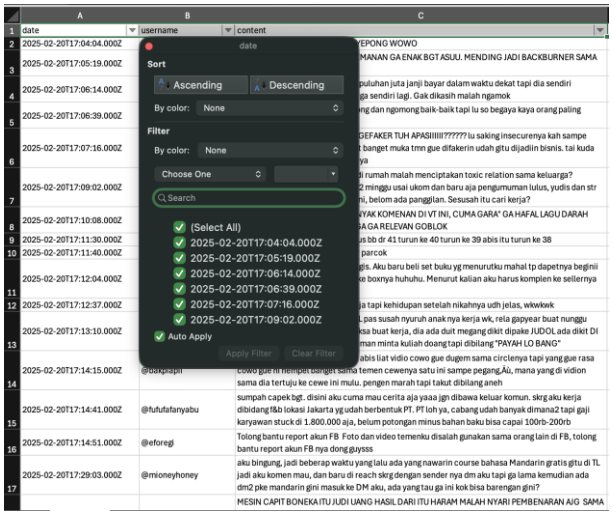
4.2.1 *Duplicate Removal*

Proses pembersihan data duplikat menjadi tahap yang sangat krusial dalam penelitian ini dikarenakan karakteristik operasional alat *scraping* yang digunakan. Duplikasi data terjadi sebagai konsekuensi logis dari mekanisme pengambilan data yang dilakukan secara bertahap menggunakan Selenium dengan batasan *scroll* tertentu. Fenomena redundansi ini muncul bukan karena kesalahan sistem, melainkan akibat pengambilan data yang dilakukan dalam beberapa sesi terpisah pada *platform* linimasa yang bersifat sangat dinamis.

Kondisi tersebut memungkinkan data *tweet* yang sebenarnya sudah tersimpan pada sesi pengambilan sebelumnya, muncul kembali di layar dan terambil ulang oleh sistem pada sesi berikutnya sebelum posisi halaman benar-benar berpindah ke konten yang baru. Adanya irisan (*overlapping*) konten antar-sesi inilah yang menyebabkan redundansi data tidak dapat dihindari selama proses pengumpulan berlangsung. Oleh karena itu, prosedur penghapusan duplikasi data secara sistematis mutlak diperlukan guna memastikan setiap entri data bersifat unik dan mencegah terjadinya bias frekuensi yang dapat mendistorsi hasil model LDA.

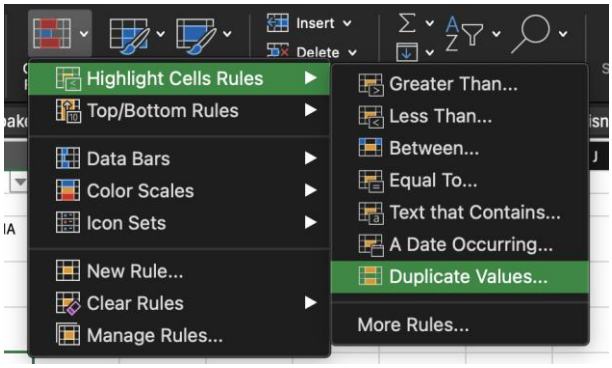
Prosedur pembersihan ini diawali dengan pengorganisasian struktur data menggunakan perangkat lunak Microsoft Excel. *Dataset* terlebih dahulu diurutkan (*sorting*) secara *ascending*

berdasarkan atribut *timestamp* atau waktu unggahan *tweet*. Pengurutan secara kronologis ini bertujuan untuk mengelompokkan data secara sekuensial sehingga memudahkan deteksi pola duplikasi pada entri yang berdekatan. Fitur yang digunakan untuk pengurutan data ditunjukkan pada Gambar 4.5



Gambar 4.5 Pengurutan Data

Setelah data terstruktur secara kronologis, proses identifikasi duplikasi dilakukan dengan memanfaatkan fitur *Conditional Formatting* pada Microsoft Excel, secara spesifik menggunakan aturan *Highlight Cells Rules.* Mekanisme ini bekerja dengan memindai keseluruhan baris data dan mencari kesamaan nilai identik pada kolom "*content*" (isi *tweet*). Konfigurasi ini ditampilkan pada Gambar 4.6.



Gambar 4.6 *Conditional Formatting*

Sebagaimana hasil konfigurasi tersebut, sistem akan secara otomatis menandai data yang teridentifikasi sebagai duplikat dengan indikator warna merah (*red field*). Visualisasi hasil penandaan ini dapat dilihat pada Gambar 4.7. Penandaan visual ini berfungsi sebagai instrumen verifikasi untuk membedakan secara tegas antara data unik yang harus dipertahankan dan data redudan yang perlu dieliminasi.

6061	2025-05-03T10:13:22.000Z	@911yeyun	aku juga mau kuliah mah.. cape bgt mau kuliah aja harus pake duit sendiri, kapan kuliahnya wkwkw, gaji di Indonesia sekecil itu harus cukup buat sehari sehari, buat kuliah sampe lulus mana harus ngasih ortu juga
6062	2025-05-03T10:14:23.000Z	@matchabreadss	APA APAAN BANGET ANJIING, BBILANGNYA MAH DISURUH KUMPULIN HP DOANG. TAPI MALAH DIBUKA GALERI + WA, SEKOLAH ANJIING, GURU ANJIING EMANG. jujur, gue kaga peduli klo ada yang aneh anehh nya, tapi ITU PRIVASI ANJIING. KONTOL AMAT
6063	2025-05-03T10:14:27.000Z	@macchiatio	¿Üihmm oong ¿Ü budu amat anjir tolol lu alay asu
6064	2025-05-03T10:15:00.000Z	@callmejujuuuuu	cape jadi manusia
6065	2025-05-03T10:17:00.000Z	@MassageUtkCewek	
6066	2025-05-03T10:17:06.000Z	@nonamanis21544	ANEH BGTT MERTUA GUE ANAKNY UDH NIKAH ISTRINYA LAGI HAMIL MASA IYA G BOLEH KETEMU SMA ISTRINYA PLISS SPILL HADIS YG MENYANGKUT MSLH INI..gua strss bgt hamil udh tm 3
6067	2025-05-03T10:20:12.000Z	@haltetendean	Mas-mu tinggal di rumah orang tuamu itu kan PASTI juga masih kasih jatah bulanan ke orang tuamu, nder. Ya suruhlah ibumu pake uang itu untuk beli siomay Kasian mas-mu udah berumah tangga tapi masih dirongrong orang tuanya.
6068	2025-05-03T10:20:58.000Z	@jsuhonny	bersyukur dua abang gw selalu inget mak dan gw klo beli makanan krn ya itu gak tega klo makan sendiri
6069	2025-05-03T10:22:32.000Z	@alterchev	MUAK BANGET YA ALLAH, MAMAK KU KENA VIRUS PICA PICA DEWI PERSIK SEMENIT SEKALI "CEüCEüCEüCEé AMBON PICAPICAPICA CEüCEüCEüCEé CEüCEéNTI LINCALINCALINCA MACA PAPUA GARAS TANAH" (DENGAN NADA DEWI PERSIK)
6070	2025-05-03T10:22:32.000Z	@alterchev	MUAK BANGET YA ALLAH, MAMAK KU KENA VIRUS PICA PICA DEWI PERSIK SEMENIT SEKALI "CEüCEüCEüCEé AMBON PICAPICAPICA CEüCEüCEüCEé CEüCEéNTI LINCALINCALINCA MACA PAPUA GARAS TANAH" (DENGAN NADA DEWI PERSIK)
6071	2025-05-03T10:23:52.000Z	@moonlightshasaa	Capek bgt nyari kostan di Jaksel harga ya pada gak ngotak dengan fasilitas yg jauh dari kata standar
6072	2025-05-03T10:23:52.000Z	@moonlightshasaa	Capek bgt nyari kostan di Jaksel harga ya pada gak ngotak dengan fasilitas yg jauh dari kata standar
6073	2025-05-03T10:24:38.000Z	@K_aji_99	Negara isinya cuma omong kosong
6074	2025-05-03T10:24:38.000Z	@K_aji_99	Negara isinya cuma omong kosong
6075	2025-05-03T10:25:23.000Z	@renn89815641	Sedih bgt anjayyy wkwkwkwk Mau juga nongkrong bareng temen, tapi paham jg keadaan rmh gimana. ortu sibuk nyuruh bergaul keluar, main sm teman tapi yh sdhlh
6076	2025-05-03T10:25:23.000Z	@renn89815641	Sedih bgt anjayyy wkwkwkwk Mau juga nongkrong bareng temen, tapi paham jg keadaan rmh gimana. ortu sibuk nyuruh bergaul keluar, main sm teman tapi yh sdhlh
6077	2025-05-03T10:28:12.000Z	@boyyssdontcryy	kasih saran dong, yg paling lucu sama menarik dapet hadiah
6078	2025-05-03T10:28:12.000Z	@boyyssdontcryy	kasih saran dong, yg paling lucu sama menarik dapet hadiah
6079	2025-05-03T10:30:01.000Z	@ealamas	begini doang pdhl, sibuk banget dah orang2 wkwk
6080	2025-05-03T10:30:01.000Z	@ealamas	begini doang pdhl, sibuk banget dah orang2 wkwk

Gambar 4.7 Duplikasi Data dalam *Dataset*

Langkah selanjutnya adalah eksekusi penghapusan baris data yang telah ditandai tersebut. Setelah proses eliminasi selesai, dilakukan pengecekan ulang untuk memastikan integritas data. Gambar 4.8 memperlihatkan kondisi *dataset* setelah pembersihan, di mana tidak lagi ditemukan indikator warna merah pada sel data, yang memvalidasi bahwa seluruh entri data kini bersifat unik.

6027	2025-05-03T10:13:22.000Z	@911yeyun	aku juga mau kuliah mah.. cape bgt mau kuliah aja harus pake duit sendiri, kapan kuliahnya wkwkw, gaji di Indonesia sekecil itu harus cukup buat sehari sehari, buat kuliah sampe lulus mana harus ngasih ortu juga
6028	2025-05-03T10:14:23.000Z	@matchabreadss	APA APAAN BANGET ANJIING, BBILANGNYA MAH DISURUH KUMPULIN HP DOANG. TAPI MALAH DIBUKA GALERI + WA, SEKOLAH ANJIING, GURU ANJIING EMANG. jujur, gue kaga peduli klo ada yang aneh anehh nya, tapi ITU PRIVASI ANJIING. KONTOL AMAT
6029	2025-05-03T10:14:27.000Z	@macchiatio	¿Üihmm oong ¿Ü budu amat anjir tolol lu alay asu
6030	2025-05-03T10:15:00.000Z	@callmejujuuuuu	cape jadi manusia
6031	2025-05-03T10:17:06.000Z	@nonamanis21544	ANEH BGTT MERTUA GUE ANAKNY UDH NIKAH ISTRINYA LAGI HAMIL MASA IYA G BOLEH KETEMU SMA ISTRINYA PLISS SPILL HADIS YG MENYANGKUT MSLH INI..gua strss bgt hamil udh tm 3
6032	2025-05-03T10:20:12.000Z	@haltetendean	Mas-mu tinggal di rumah orang tuamu itu kan PASTI juga masih kasih jatah bulanan ke orang tuamu, nder. Ya suruhlah ibumu pake uang itu untuk beli siomay Kasian mas-mu udah berumah tangga tapi masih dirongrong orang tuanya.
6033	2025-05-03T10:20:58.000Z	@jsuhonny	bersyukur dua abang gw selalu inget mak dan gw klo beli makanan krn ya itu gak tega klo makan sendiri
6034	2025-05-03T10:22:32.000Z	@alterchev	MUAK BANGET YA ALLAH, MAMAK KU KENA VIRUS PICA PICA DEWI PERSIK SEMENIT SEKALI "CEüCEüCEüCEé AMBON PICAPICAPICA CEüCEüCEüCEé CEüCEéNTI LINCALINCALINCA MACA PAPUA GARAS TANAH" (DENGAN NADA DEWI PERSIK)
6035	2025-05-03T10:23:52.000Z	@moonlightshasaa	Capek bgt nyari kostan di Jaksel harga ya pada gak ngotak dengan fasilitas yg jauh dari kata standar
6036	2025-05-03T10:24:38.000Z	@K_aji_99	Negara isinya cuma omong kosong
6037	2025-05-03T10:25:23.000Z	@renn89815641	Sedih bgt anjayyy wkwkwkwk Mau juga nongkrong bareng temen, tapi paham jg keadaan rmh gimana. ortu sibuk nyuruh bergaul keluar, main sm teman tapi yh sdhlh
6038	2025-05-03T10:28:12.000Z	@boyyssdontcryy	kasih saran dong, yg paling lucu sama menarik dapet hadiah
6039	2025-05-03T10:30:01.000Z	@ealamas	begini doang pdhl, sibuk banget dah orang2 wkwk
6040	2025-05-03T10:30:21.000Z	@sahicimol	guys wta dong spay ortuku beberapa kali kebobolan topup kaya gini padahal gapernah transaksi pulsa/tagihan di spay. ini kirakira kenapa ya? apa kena hack?
6041	2025-05-03T10:31:09.000Z	@197jae_	huft,Ãemg gocar sekarang kalo 20k ga mau ia?
6042	2025-05-03T10:34:27.000Z	@asthrophille_	guys ini nipu?
6043	2025-05-03T10:35:04.000Z	@leopardd33	KENAPA YA KALO UDAH DITAHAP MUTUALAN IG DARI LEO TUH JADI ASINGG??!! #leomatch #tale
6044	2025-05-03T10:35:15.000Z	@kaleyars	MECIMA GOBLOK ANJIING TOLOL LU ANJING BANGSAAATTTT DASAR DZOLIM ANJING ORG ORG DI DALEMNYA KONTOL, MECIMA KONTOLLLLLLLLLLLL @mecimapro
6045	2025-05-03T10:35:27.000Z	@kimchan9829	Pertama kali gw kerja megang facial treatment ke pasien gw langsung dikomplen lewat wa klinik, wajar ga sih gw belum lancar bgt, gw banyak kurangnya terus klo dipikir2:)))
			Kalian pernah marah sm ibu berapa lama? Gw kesel banget udah 3 hari ini diem2an. Capek bgt

Gambar 4.8 Hasil Penghapusan Data Duplikat

Proses eliminasi ini berdampak pada penyusutan volume data yang cukup signifikan dari total data awal. Berdasarkan hasil pembersihan, *dataset* akhir yang tersisa berjumlah 38.956 data. Rincian komparasi kuantitas data sebelum dan sesudah proses penghapusan duplikat disajikan dalam Tabel 4.2. *Dataset* yang telah bebas dari duplikasi ini kemudian disimpan kembali dalam format *Comma Separated Values* (CSV) sebagai data baku yang siap untuk menjalani tahapan *preprocessing* teks lanjutan yang lebih kompleks.

Tabel 4.2 Data Penghapusan Duplikasi

Sebelum Penghapusan Duplikasi	**Setelah Penghapusan Duplikasi**
75.032	38.956
Jumlah data yang dihapus (karena duplikat): 36.076	

4.2.2 *Remove Punctuation*

Tahap kedua dalam rangkaian *preprocessing* teks adalah penghapusan tanda baca dan karakter khusus. Tujuan utama dari tahapan ini adalah untuk mengeliminasi elemen-elemen non-alfabetis, simbol, serta metadata platform yang dianggap sebagai *noise*. Dalam analisis teks, karakter-karakter non-alfabet sering kali tidak memberikan kontribusi semantik yang signifikan dan justru berpotensi mengaburkan fokus analisis. Oleh karena itu, proses ini memastikan bahwa data yang diproses hanyalah data tekstual murni yang memuat informasi substantif.

Implementasi teknis tahap ini dilakukan dengan memanfaatkan modul *regular expression* (`re`) pada Phyton. Algoritma pembersihan dijalankan secara bertahap melalui fungsi `remove_punct` yang dirancang untuk mendeteksi dan menghapus pola karakter tertentu. Sebagaimana ditampilkan pada kode program pada Gambar 4.9, proses filtrasi dimulai dengan menghapus entitas spesifik platform X, meliputi nama pengguna (*mentions*) yang diawali simbol @, tagar (*hashtags*), penanda *retweet* (RT), serta tautan URL (*hyperlinks*). Keberadaan entitas-entitas ini dihhapus karena cenderung bersifat administratif dan tidak mencerminkan sentimen atau topik utama dari percakapan.

```
import nltk
nltk.download('punkt')
import re


def remove_punct(text):
    text = re.sub(r'@[A-Za-z0-9_]+', '', text)
```

```
    text = re.sub(r'#\w+', '', text)
    text = re.sub(r'RT[\s]+', '', text)
    text = re.sub(r'https?://\S+', '', text)
    text = re.sub(r'[^A-Za-z ]', ' ', text)
    text = re.sub(r'\s+', ' ', text).strip()

    return text

df.loc[:, 'remove_punctuation'] = df['content'].apply(remove_punct)

df.to_csv('/kaggle/working/preprocessed_dataset.csv', index=False)

print(df[['content', 'remove_punctuation']].head(10))
```

Gambar 4.9 Kode Program *Remove Punctuation*

Setelah atribut platform dibersihkan, program melanjutkan proses dengan menghapus seluruh karakter non-alfabet, termasuk angka, tanda baca, simbol matematika, dan emoji, sehingga hanya menyisakan karakter huruf (A-Z). Langkah terakhir dalam fungsi ini adalah normalisasi spasi (*whitespace removal*). Program menghapus spasi berlebih yang muncul akibat penghapusan kata sebelumnya serta membersihkan spasi di awal dan akhir kalimat (*trimming*) untuk menjaga kerapian struktur data.

Hasil dari proses transformasi ini kemudian disimpan ke dalam kolom baru bernama `remove_punctuation` pada *dataframe*. Sebagai validasi terhadap keberhasilan proses eksekusi kode tersebut, *output* data ditampilkan pada Tabel 4.3. Tabel ini memperlihatkan bagaimana simbol dan elemen non-alfabet telah tereliminasi secara efektif tanpa mengubah konteks kalimat utama.

Tabel 4.3 Hasil Data Setelah *Remove Punctuation*

Raw Text	*Remove Punctuation*
MAU MARAH BGT tp ujungnya nangis. Aku baru beli set buku yg menurutku mahal tp dapetnya beginii T____T bolongnya nembus sampe ke boxnya huhuhu. Menurut kalian aku harus komplen ke sellernya atau ke ekspedisinya dulu ya?:((	MAU MARAH BGT tp ujungnya nangis Aku baru beli set buku yg menurutku mahal tp dapetnya beginii T T bolongnya nembus sampe ke boxnya huhuhu Menurut kalian aku harus komplen ke sellernya atau ke ekspedisinya dulu ya
gw kadang gak abis pikir sama orang orang tiktok, komennya diluar nalar semua anjing. #IndonesiaGelap	gw kadang gak abis pikir sama orang orang tiktok komennya diluar nalar semua anjing
Udh pake kartu tri dari SMA mski sering gnti nmr. Dan udah terlanjur dipake buat no	Udh pake kartu tri dari SMA mski sering gnti nmr Dan udah terlanjur dipake buat no

WA. Biasanya internet pake kartu lain tp kmrn iseng isi yg Tri karna takut nomor hangus gbs dipake. Eh sisa kuota 18 GB malah gbs dipake. Chat Admin gada respon apa¬¨,â§. @3CareIndonesia jelek bgt	WA Biasanya internet pake kartu lain tp kmrn iseng isi yg Tri karna takut nomor hangus gbs dipake Eh sisa kuota GB malah gbs dipake Chat Admin gada respon apa jelek bgt

4.2.3 *Case Folding*

Setelah *dataset* bersih dari simbol dan karakter non-alfabet, tahapan selanjutnya dalam rangkaian *preprocessing* data adalah penyeragaman format huruf atau *case folding*. Proses ini bertujuan untuk mengonversi seluruh karakter huruf dalam dokumen teks menjadi bentuk huruf kecil (*lowercase*). Implementasi teknis dari proses ini dilakukan dengan menerapkan fungsi manipulasi *string* sederhana namun berdampak signifikan pada *dataframe*. Sebagaimana diilustrasikan pada kode program di Gambar 4.10, sistem diperintahkan untuk memindai seluruh kolom teks yang telah melalui tahap pembersihan tanda baca, lalu mengubah setiap karakter kapital yang ditemukan menjadi bentuk non-kapital. Hasil eksekusi dari tahapan *case folding* dapat dilihat pada Tabel 4.4.

```
df.loc[:, 'case_folding'] = df['remove_punctuation'].str.lower()

df.to_csv('/kaggle/working/preprocessed_dataset.csv', index=False)

print(df[['remove_punctuation', 'case_folding']].head(10))
```

Gambar 4.10 Kode Program *Case Folding*

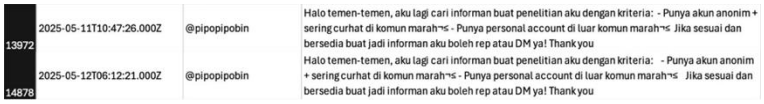
Tabel 4.4 Hasil Data Setelah *Case Folding*

Remove Punctuation	***Case Folding***
MAU MARAH BGT tp ujungnya nangis Aku baru beli set buku yg menurutku mahal tp dapetnya beginii T T bolongnya nembus sampe ke boxnya huhuhu Menurut kalian aku harus komplen ke sellernya atau ke ekspedisinya dulu ya	mau marah bgt tp ujungnya nangis aku baru beli set buku yg menurutku mahal tp dapetnya beginii t t bolongnya nembus sampe ke boxnya huhuhu menurut kalian aku harus komplen ke sellernya atau ke ekspedisinya dulu ya

4.2.4 *Advanced Duplicate Removal*

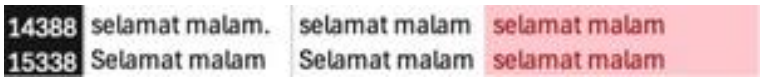
Setelah tahapan *remove punctuation* dan *case folding*, dilakukan verifikasi mendalam terhadap integritas *dataset*. Berdasarkan hasil inspeksi visual terhadap komparasi data mentah dan data hasil pemrosesan, ditemukan adanya anomali berupa redundansi data sekunder. Anomali ini merujuk pada data yang lolos dari pembersihan awal, namun sebenarnya merupakan duplikasi yang tersamarkan.

Fenomena ini disebabkan oleh dua faktor teknis utama. Pertama, adanya inkonsistensi karakter tersembunyi atau residu *encoding* pada data mentah. Sebagaimana ditampilkan pada Gambar 4.11, terdapat baris data yang secara visual tampak identik bagi pengamat manusia, namun diterjemahkan sebagai urutan *byte* yang berbeda oleh sistem komputasi akibat adanya karakter non-standar atau perbedaan spasi.



Gambar 4.11 Contoh dari Inkonsistensi Karakter Data

Kedua, proses penyeragaman format teks itu sendiri turut berkontribusi memunculkan duplikasi baru. Data yang sebelumnya dianggap unik karena perbedaan format teks, setelah diseragamkan menjadi huruf kecil dan dihilangkan tanda bacanya, kini memiliki struktur teks yang sama persis. Ilustrasi perubahan ini dapat dilihat pada Gambar 4.12.



Gambar 4.12 Contoh dari Penyeragaman Teks

Untuk memvalidasi temuan tersebut secara statistik, dilakukan pengecekan kuantitatif menggunakan fungsi `duplicated()` yang dikombinasikan dengan fungsi agregasi `.sum()` pada kolom yang telah melewati proses *remove punctuation* dan *case folding*. Proses tersebut ditampilkan pada Gambar 4.13

```
df.duplicated(subset=['case_folding']).sum()
```

Gambar 4.13 Kode Program Verifikasi Duplikasi

Hasil eksekusi program mengonfirmasi bahwa tahapan normalisasi teks berhasil menyingkap 67 data duplikat yang sebelumnya tidak terdeteksi. Oleh karena itu, eliminasi lanjutan mutlak diperlukan untuk menjamin validitas statistik model. Implementasi pembersihan dilakukan menggunakan fungsi `drop_duplicates` dengan menargetkan kolom *case_folding*. Hasil pembersihan disimpan secara terpisah ke dalam variabel baru bernama `df_unique` menggunakan metode `.copy()`. Langkah ini bertujuan untuk mengisolasi data bersih dan mencegah konflik memori pada proses pengolahan selanjutnya.

```
print(f"Total data awal: {len(df)}")
df_unique = df.drop_duplicates(subset=['case_folding'], keep='first').copy()
print(f"Total data bersih: {len(df_unique)}")
```

Gambar 4.14 Kode Program Penghapusan Duplikasi Sekunder

Berdasarkan *output* program pada Gambar 4.14, proses ini berhasil menyaring seluruh data redundan dan menyisakan total 38.887 data unik (dari sebelumnya pembersihan data duplikat tahap pertama sejumlah 38.956 data). *Dataset* yang tersimpan dalam variabel `df_unique` inilah yang dipastikan bebas dari *noise* duplikasi dan akan dijadikan acuan tunggal untuk tahapan selanjutnya.

4.2.5 *Tokenizing*

Setelah data teks dipastikan bersih dan unik melalui tahapan eliminasi duplikasi lanjutan, langkah selanjutnya dalam rangkaian *preprocessing* adalah tokenisasi (*tokenizing*). Tokenisasi merupakan proses penguraian kalimat atau dokumen teks panjang menjadi potongan-potongan unit kata yang berdiri sendiri, yang dalam istilah komputasi disebut sebagai token. Tujuan utama dari tahapan ini adalah untuk mentransformasi struktur data teks yang semula berupa kalimat utuh (*string*) menjadi sekumpulan kata terpisah (*list of words*). Transformasi ini sangat fundamental karena memungkinkan model analisis untuk mengenali, menghitung, dan memproses setiap kata sebagai entitas fitur yang independen.

Implementasi teknis tokenisasi dalam penelitian ini memanfaatkan pustaka *Natural Language Toolkit* (NLTK), sebuah modul standar dalam bahasa pemrograman Python yang dirancang khusus untuk Pemrosesan Bahasa Alami (*Natural Language Processing*). Secara spesifik, fungsi yang digunakan adalah `word_tokenize`. Sebagaimana diilustrasikan pada kode program di Gambar 4.15, fungsi ini diaplikasikan pada kolom *case_folding* yang terdapat dalam *dataframe* `df_unique`.

```
from nltk.tokenize import word_tokenize
df_unique.loc[:, 'tokens'] = df_unique['case_folding'].apply(word_tokenize)
df_unique.to_csv('/kaggle/working/preprocessed_dataset.csv', index=False)
print(df_unique[['case_folding', 'tokens']].head(10))
```

Gambar 4.15 Kode Program *Tokenizing*

Algoritma `word_tokenize` bekerja dengan memindai setiap baris kalimat dan memisahkannya berdasarkan karakter spasi (*whitespace*), *tab*, karakter baris baru (*newline*),

serta tanda baca (seperti titik, koma, tanda tanya, dan tanda seru). Hasil dari pemisahan ini kemudian disimpan ke dalam kolom baru bernama *tokens*. Perbedaan mendasar struktur data sebelum dan sesudah proses ini dapat diamati pada Tabel 4.5. Pada kolom *case_folding*, data masih berbentuk kalimat tunggal, sedangkan pada kolom *tokens*, data telah berubah format menjadi daftar kata yang dipisahkan oleh tanda koma.

Tabel 4.5 Hasil Data Setelah *Tokenizing*

Case Folding	***Tokenizing***
tadi sempat pusing sedikit sesak dan muntah akhirnya ke dokter umum buat memeriksakan diri tapi setelah ditanya keluhan dan diperiksa sama dokternya dokternya nyaranin ke dokter jiwa ada yg sama gk sih	['tadi', 'sempat', 'pusing', 'sedikit', 'sesak', 'dan', 'muntah', 'akhirnya', 'ke', 'dokter', 'umum', 'buat', 'memeriksakan', 'diri', 'tapi', 'setelah', 'ditanya', 'keluhan', 'dan', 'diperiksa', 'sama', 'dokternya', 'dokternya', 'nyaranin', 'ke', 'dokter', 'jiwa', 'ada', 'yg', 'sama', 'gk', 'sih']
sama bgt beberapa kali gini untung promo tapi kan tetep aja dih najis dokter dokter yang begini tuh maunya apa mending dari awal bilang aja gausah php	['sama', 'bgt', 'beberapa', 'kali', 'gini', 'untung', 'promo', 'tapi', 'kan', 'tetep', 'aja', 'dih', 'najis', 'dokter', 'dokter', 'yang', 'begini', 'tuh', 'maunya', 'apa', 'mending', 'dari', 'awal', 'bilang', 'aja', 'gausah', 'php']

4.2.6 *Normalization*

Proses normalisasi bertujuan untuk mentransformasi kata-kata yang tidak sesuai dengan standar ke dalam bentuk yang baku. Mengingat data komentar media sosial mengandung variasi bentuk kata tidak standar yang sangat dinamis seperti singkatan, *slang*, dan kesalahan pengetikan (*typo*) tahapan ini menjadi krusial untuk menyamakan persepsi makna kata.

Untuk mengakomodasi kebutuhan tersebut, penelitian ini menggunakan kamus normalisasi (normalizeWord.csv) yang disusun secara kuratif berdasarkan karakteristik linguistik data "Komunitas MARAH MARAH". Penyusunan kamus ini dilakukan dengan memetakan entri kata tidak baku yang dominan muncul dalam *dataset* ke bentuk bakunya dengan merujuk secara ketat pada Kamus Besar Bahasa Indonesia (KBBI) Daring. Pendekatan kurasi mandiri ini dipilih untuk memastikan akurasi yang lebih tinggi dibandingkan menggunakan kamus umum, karena mampu menangkap istilah-istilah spesifik komunitas yang mungkin belum terdaftar di repositori publik.

Sebagaimana ditampilkan pada Gambar 4.16, implementasi teknis diawali dengan memuat data kamus tersebut menggunakan fungsi `pd.read_csv`. Data kemudian dikonversi menjadi struktur data *dictionary* Python `norm_dict` melalui mekanisme penggabungan

kolom `kata_tidak_baku` sebagai kunci (*key*) dan `kata_normal` sebagai nilai (*value*) menggunakan fungsi `zip`. Metode pemetaan ini dipilih agar proses pencarian dan penggantian kata (*lookup and replace*) dapat dilakukan dengan kompleksitas waktu yang efisien.

```
import pandas as pd

mapping_df = pd.read_csv('/kaggle/input/normalisasi/normDict2.csv')

norm_dict = dict(
    zip(
        mapping_df['kata_tidak_baku'].astype(str).str.lower(),
        mapping_df['kata_normal'].astype(str).str.lower()
    )
)

def normalisasi(token_list):
    if not isinstance(token_list, list):
        return ""

    normalized_tokens = []

    for token in token_list:
        token = str(token).lower()
        normalized_token = norm_dict.get(token, token)
        normalized_tokens.append(normalized_token)

    return ' '.join(normalized_tokens)

df_unique.loc[:, 'normalize'] = df_unique['tokens'].apply(normalisasi)

df_unique.to_csv('/kaggle/working/preprocessed_dataset.csv', index=False)

print(df_unique[['tokens', 'normalize']].head(10))
```

Gambar 4.16 Kode Program Normalisasi

Sebagai ilustrasi mengenai pemetaan yang dilakukan, Tabel 4.6 menampilkan sampel perubahan dari kata tidak baku menjadi kata yang dinormalisasi sesuai kaidah bahasa Indonesia yang baik dan benar. Adapun daftar lengkap dari kamus *normalization* tersebut dapat dilihat secara rinci pada bagian Lampiran B.

Tabel 4.6 Contoh Kamus Normalisasi

Kata_tidak_baku	**Kata_normal**
gangguuuu	ganggu
dlu	dulu
dedek	adik

Setelah Setelah kamus referensi terbentuk, fungsi `normalisasi` diterapkan pada seluruh data. Mekanisme substitusi kata dilakukan menggunakan metode `.get()`, di mana token yang ditemukan dalam kamus akan diganti secara otomatis, sedangkan token lainnya

dipertahankan. Pada akhir proses, token-token tersebut digabungkan kembali (*reconstruct*) menjadi kalimat utuh menggunakan metode `' '.join()` dan disimpan dalam kolom normalize. Hasil implementasi dari tahap normalisasi ini dapat dilihat pada Tabel 4.7, yang memperlihatkan perbandingan antara data token awal dengan data yang telah dinormalisasi menjadi kalimat baku.

Tabel 4.7 Hasil Data Setelah *Normalization*

Tokenizing	***Normalization***
['sederhana', 'tapi', 'bikin', 'kesel', 'udh', 'nunggu', 'proses', 'pengiriman', 'lama', 'trs', 'pesen', 'yg', 'coklat', 'malah', 'yg', 'datang', 'strawberry', 'bodoamat', 'gue', 'rate', 'ga', 'mood', 'bgt', 'sialan']	sederhana tapi bikin **kesal sudah** nunggu proses pengiriman lama **terus** pesen **yang** coklat malah **yang** datang strawberry bodoamat **saya** rate **tidak** mood **banget** sialan
['hari', 'ini', 'izin', 'ga', 'masuk', 'kerja', 'karena', 'semalem', 'abis', 'nangis', 'terus', 'hari', 'ini', 'badan', 'gue', 'ga', 'enak', 'banget', 'jujur', 'gue', 'capek', 'bgt', 'anjing']	hari ini izin **tidak** masuk kerja karena semalem **habis** nangis terus hari ini badan **saya tidak** enak banget jujur **saya** capek **banget** anjing

4.2.7 *Stopwords Removal*

Setelah struktur kalimat distandarisasi melalui tahap normalisasi, langkah selanjutnya dalam rangkaian *preprocessing* adalah penghapusan kata umum atau *stopwords removal*. Tahapan ini bertujuan untuk mengeliminasi kata-kata fungsional yang memiliki frekuensi kemunculan tinggi namun minim kontribusi semantik terhadap topik utama. Kata-kata seperti "yang", "dan", "di", atau "dari" sering kali mendominasi korpus teks tetapi tidak membawa informasi substantif mengenai sentimen atau isi percakapan. Dengan menghapus kata-kata tersebut, dimensi fitur data dapat direduksi secara signifikan sehingga model dapat berfokus pada kata-kata kunci yang lebih berbobot.

Implementasi teknis tahap ini memanfaatkan korpus *stopwords* bahasa Indonesia yang disediakan oleh pustaka *Natural Language Toolkit* (NLTK). Sebagaimana ditampilkan pada kode program di Gambar 4.17, proses dimulai dengan memuat daftar kata umum dari modul `nltk.corpus`. Untuk meningkatkan efisiensi komputasi, daftar kata tersebut dikonversi menjadi struktur data himpunan (*set*). Penggunaan tipe data *set* ini sangat krusial dalam pemrograman karena menawarkan kecepatan pencarian yang jauh lebih tinggi dibandingkan tipe data daftar (*list*) biasa, terutama saat melakukan penyaringan terhadap ribuan data teks. Selain menggunakan daftar standar dari NLTK, penelitian ini juga menerapkan teknik *customized stopwords* dengan menambahkan daftar kata spesifik melalui variabel `addition`.

Penambahan ini mencakup kata-kata umpatan kasar, interjeksi, dan partikel bahasa lisan (seperti "sih", "tuh", "nya") yang memiliki frekuensi kemunculan sangat tinggi dalam "Komunitas MARAH MARAH" namun tidak memberikan kontribusi semantik terhadap topik utama.

Penghapusan kata umpatan ini dilakukan dengan pertimbangan bahwa kata-kata tersebut murni berfungsi sebagai ekspresi emosi sesaat dan bukan merupakan representasi dari isu atau substansi permasalahan yang dibicarakan. Dengan mengeliminasi kata-kata tersebut, model LDA dapat bekerja lebih efektif dalam mengekstraksi tema informasi yang lebih substantif tanpa terdistorsi oleh residu bahasa lisan yang tidak informatif.

```
import pandas as pd
import nltk
from nltk.corpus import stopwords

try:
    nltk.data.find('corpora/stopwords')
except LookupError:
    nltk.download('stopwords')

stop_words = set(stopwords.words('indonesian'))

addition = {
    'anjing', 'kontol', 'bangsat', 'tahi', 'goblok', 'tolol', 'ngentot', 'bgst',
'kntl', 'sih',
    'tuh', 'nya', 'kalo', 'pas', 'udh', 'bgt', 'kayak', 'gini', 'itu', 'aja',
'shibal', 'kamu',
    'monyet', 'babi', 'fak', 'fuck', 'fakkkk', 'anjir', 'cok', 'nang', 'kak',
'this', 'pantek',
    'neng', 'bangke', 'astaga', 'buset', 'the', 'wni', 'all', 'what', 'day',
'just', 'anjjjj',
    'that', 'pick', 'after', 'people', 'not', 'bat', 'wkwkwkwk', 'duh', 'bangsatt',
'anjirrrr',
    'real', 'bitch', 'pen', 'bpk', 'and', 'like', 'kentot', 'anjirr', 'dih',
'nyari', 'memek',
    'ngentod', 'bnr', 'ngeliat', 'bajingan', 'brengsek', 'you', 'tot', 'puki',
'wkwk', 'wkwkwk',
    'kau', 'your', 'titik', 'tau', 'sok', 'nyokap', 'ngerokok', 'cuih', 'asuu',
'asu', 'yang',
    'berengsek', 'aing', 'ngechat', 'banget', 'ntar', 'mba', 'jancok', 'bet',
'mentang', 'nge',
    'bajing', 'smp', 'hahaha', 'ajaa', 'orng', 'huhuhu', 'anjjj', 'gpp', 'saja',
'dan'
}

stop_words.update(addition)

def remove_stopwords(text):
    if not isinstance(text, str):
        return ""

    words = text.split()
    filtered_words = [
        word for word in words
        if word.lower() not in stop_words
    ]
```

```
    return ' '.join(filtered_words)

df_unique.loc[:, 'stopwords'] = df_unique['normalize'].apply(remove_stopwords)

df_unique.to_csv('/kaggle/working/preprocessed_dataset.csv', index=False)

print(df_unique[['normalize', 'stopwords']].head(10))
```

Gambar 4.17 Kode Program *Stopwords Removal*

Eksekusi penyaringan dilakukan dengan menerapkan fungsi customized `remove_stopwords` pada kolom *normalize*. Algoritma fungsi ini bekerja dengan memecah kalimat menjadi unit kata menggunakan metode `.split()`, kemudian melakukan iterasi pengecekan pada setiap kata. Jika sebuah kata terdaftar dalam himpunan *stopwords*, maka kata tersebut akan dibuang. Sebaliknya, kata-kata yang dianggap penting (tidak ada dalam daftar *stopwords*) akan dipertahankan dan digabungkan kembali menjadi kalimat utuh menggunakan metode `' '.join()`.

Hasil dari proses penyaringan ini disimpan ke dalam kolom baru bernama *stopwords*. Untuk memvalidasi efektivitas proses ini, Tabel 4.8 menyajikan komparasi antara teks sebelum dan sesudah penghapusan *stopwords*. Terlihat bahwa kalimat yang dihasilkan menjadi lebih ringkas dan padat karena hanya menyisakan kata-kata yang memiliki muatan makna utama.

Tabel 4.8 Hasil Data Setelah *Stopwords Removal*

Normalization	***Stopwords Removal***
hari ini izin tidak masuk kerja karena semalem habis nangis terus hari ini badan saya tidak enak banget jujur saya capek banget anjing	izin masuk tidak kerja semalem habis nangis badan tidak enak banget jujur capek banget anjing
teman maaf tanya di sini adakah yang tau info psikolog yang bisa dipanggil ke rumah area soloraya	teman maaf adakah tau info psikolog dipanggil rumah area soloraya

4.2.8 *Stemming*

Implementasi *stemming* dilakukan sebagai langkah final dalam *preprocessing* data untuk mentransformasi kata-kata berimbuhan menjadi kata dasar. Pada tahap ini, penelitian menggunakan *library* Sastrawi yang diterapkan melalui fungsi *customized* bernama `stem_text`. Sebagaimana ditampilkan pada kode program di Gambar 4.18, proses eksekusi dimulai dengan menginisialisasi objek `stemmer` dari kelas `StemmerFactory`. Fungsi tersebut kemudian diaplikasikan pada kolom *stopwords*, di mana setiap kalimat dipecah

menjadi unit kata dan diproses satu per satu untuk menghilangkan imbuhan awalan, akhiran, maupun sisipan.

```
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

factory = StemmerFactory()
stemmer = factory.create_stemmer()

def stem_text(text):
    if not isinstance(text, str):
        return text
    words = text.split()
    stemmed_words = [stemmer.stem (word) for word in words]

    return ' '.join(stemmed_words)

df_unique.loc[:, 'stemming'] = df_unique['stopwords'].apply(stem_text)

df_unique.to_csv('/kaggle/working/preprocessed_dataset.csv', index=False)

print(df_unique[['stopwords', 'stemming']].head(10))
```

Gambar 4.18 Kode Program *Stemming*

Hasil eksekusi program tersebut disimpan ke dalam kolom baru bernama *stemming*. Mekanisme ini berhasil menyederhanakan variasi kata kerja aktif maupun pasif menjadi bentuk dasarnya, yang secara signifikan mengurangi dimensi fitur pada *dataset*. Tabel 4.9 memperlihatkan hasil komparasi data secara langsung antara teks sebelum dan sesudah proses *stemming*, di mana kata-kata seperti "mengerjakan" atau "dikirim" telah berhasil diubah menjadi "kerja" dan "kirim". *Dataset* yang telah melalui seluruh tahapan *preprocessing* ini kemudian diekspor kembali ke dalam format CSV (preprocessed_dataset.csv).

Tabel 4.9 Hasil Data Setelah *Stemming*

Stopwords Removal	***Stemming***
teman maaf adakah tau info psikolog dipanggil rumah area soloraya	teman maaf **ada** tau info psikolog **panggil** rumah area soloraya
serius nanya emang nilai diubah ya diakhir perkuliahan anjir ambil ijazah cocokin nilai transkip akademik kartu hasil studi semester matkul nilainya diturunin sialan	serius nanya emang nilai **ubah** ya akhir **kuliah** anjir ambil ijazah cocokin nilai transkip akademik kartu hasil studi semester matkul nilai diturunin **sial**

4.2.9 *Short Words Removal*

Tahapan final dalam rangkaian *preprocessing* data adalah *short words removal* (penghapusan kata pendek). Langkah ini bertujuan untuk mengeliminasi token yang memiliki panjang karakter kurang dari tiga huruf. Implementasi teknis dilakukan dengan mendefinisikan fungsi `remove_short_words` sebagaimana ditampilkan pada Gambar 4.19. Algoritma

fungsi ini bekerja dengan memecah teks menjadi token dan melakukan penyaringan kondisional, di mana hanya token dengan panjang karakter minimal tiga huruf (`len >= 3`) yang akan dipertahankan, sementara token dengan panjang satu atau dua karakter akan dihapus.

```
import pandas as pd

def remove_short_words(text, min_char_length=3):

    if not isinstance(text, str):
        return ""
    tokens = text.split()
    filtered_tokens = [token for token in tokens if len(token) >= min_char_length]

    return ' '.join(filtered_tokens)

df_unique.loc[:, 'cleaned_text'] = df_unique['stemming'].apply(remove_short_words)

df_unique.to_csv('/kaggle/working/preprocessed_dataset.csv', index=False)

print(df_unique[['stemming', 'cleaned_text']].head(10))
```

Gambar 4.19 Kode Program *Short Words Removal*

Penetapan ambang batas ini didasarkan pada pertimbangan preservasi konteks semantik. Keputusan untuk tidak menaikkan batas eliminasi menjadi tiga huruf bertujuan mencegah hilangnya kata-kata kunci substantif dalam korpus "Komunitas MARAH MARAH" yang memiliki struktur pendek, seperti "ibu", "bos", "kos", "jam", "tes", maupun "cek". Jika batasan eliminasi dinaikkan, kata-kata bermakna tersebut berisiko turut terhapus, yang dapat mendistorsi hasil interpretasi topik. Oleh karena itu, meskipun terdapat risiko lolosnya beberapa partikel bahasa lisan tak bermakna, batasan ini tetap dipertahankan sebagai konsekuensi teknis demi memastikan data-data vital penyebab kemarahan tetap terjaga.

Hasil penyaringan kemudian disimpan dalam kolom *cleaned_text* yang merepresentasikan data bersih final. Tabel 4.10 menampilkan cuplikan data akhir setelah proses ini, di mana kata-kata pendek non-esensial telah dihilangkan. Seluruh data yang telah melalui rangkaian panjang *preprocessing* ini selanjutnya diekspor dan disimpan secara permanen ke dalam berkas preprocessed_dataset.csv untuk digunakan sebagai input utama pada tahapan pemodelan topik LDA.

Tabel 4.10 Hasil Data Setelah *Short Words Removal*

Stemming	***Short Words Removal***
masjid baju ny sopan banget	masjid baju sopan banget
eh janji tuh batal gitu ya monyet excited h gajadi gimana balikin moodnya	janji tuh batal gitu monyet excited gajadi gimana balikin moodnya

4.3 Pembobotan Kata dengan TF-IDF

Setelah seluruh rangkaian pembersihan data selesai, tahap selanjutnya adalah mengubah data tekstual menjadi representasi numerik agar dapat diproses oleh algoritma pemodelan topik. Metode yang digunakan dalam penelitian ini adalah *Term Frequency-Inverse Document Frequency* (TF-IDF). TF-IDF berfungsi untuk memberikan bobot pada setiap kata berdasarkan frekuensi kemunculannya dalam dokumen dan keunikannya di seluruh korpus. Kata yang sering muncul dalam satu dokumen namun jarang muncul di dokumen lain akan mendapatkan bobot tinggi, sehingga dianggap sebagai kata kunci yang mencirikan topik dokumen tersebut.

Sebelum proses pembobotan dilakukan, data teks pada kolom `cleaned_text` yang tersimpan dalam format kalimat (*string*) perlu diubah kembali menjadi format daftar token (*list of tokens*). Proses ini dilakukan menggunakan fungsi `.split()` diikuti dengan pengecekan akhir untuk memastikan tidak ada dokumen kosong yang lolos akibat proses pembersihan sebelumnya. Dokumen yang tidak memiliki token akan dihapus dari *list* untuk mencegah *error* komputasi. Implementasi kode persiapan data, penyaringan fitur, dan pembentukan model TF-IDF ditampilkan pada Gambar 4.20.

```
from gensim import corpora, models

texts = [str(t).split() for t in df_unique['cleaned_text'].fillna('').tolist()]
texts = [t for t in texts if len(t) > 0]

def build_corpus(texts):
    dictionary = corpora.Dictionary(texts)
    dictionary.filter_extremes(no_below=20, no_above=0.5)

    corpus_bow = [dictionary.doc2bow(doc) for doc in texts]

    tfidf_model = models.TfidfModel(corpus_bow)
    corpus_tfidf = tfidf_model[corpus_bow]

    return dictionary, corpus_bow, corpus_tfidf

dictionary, corpus_bow, corpus_tfidf = build_corpus(texts)
corpus_for_lda = corpus_tfidf
```

Gambar 4.20 Kode Program TF-IDF

Langkah Langkah inti dari proses ini dimulai dengan membangun kamus (*dictionary*) menggunakan *library* Gensim yang memetakan setiap kata unik ke dalam ID integer. Guna mengoptimalkan kualitas topik yang dihasilkan, dilakukan penyaringan fitur ekstrem (*filter extremes*) pada kamus tersebut dengan parameter yang ketat. Sebagaimana terlihat pada kode program, parameter `no_below=20` diterapkan untuk mengabaikan kata-kata yang muncul

pada kurang dari 20 dokumen. Nilai ambang batas ini dipilih untuk membersihkan *noise* level rendah berupa kesalahan pengetikan (*typo*) unik atau kata *slang* yang sangat jarang digunakan, agar model dapat berfokus pada pola global. Penerapan nilai ini dilakukan dengan hati-hati, jika ambang batas dinaikkan terlalu tinggi, terdapat risiko hilangnya informasi pada topik-topik *niche* yang memiliki frekuensi kemunculan rendah.

Selanjutnya, parameter `no_above=0.5` digunakan untuk mengeliminasi kata-kata yang muncul di lebih dari 50% total dokumen. Kata-kata dengan frekuensi kemunculan masif ini dianggap sebagai *corpus-specific stopwords* yang memiliki daya pembeda rendah. Jika nilai ini dinaikkan, kata-kata umum akan tetap masuk ke dalam model, yang berisiko menyebabkan topik-topik yang dihasilkan menjadi seragam (*overlapping*) dan sulit dibedakan satu sama lain.

Setelah kamus optimal terbentuk, data teks dikonversi menjadi representasi *Bag-of-Words* (BoW) menggunakan fungsi `doc2bow`. Representasi BoW ini mencatat frekuensi kemunculan kata dalam setiap dokumen. Terakhir, model `TfidfModel` diterapkan pada korpus BoW tersebut untuk menghitung nilai bobot TF-IDF setiap kata. Hasil akhirnya berupa variabel `corpus_for_lda` yang berisi representasi vektor dokumen dengan bobot yang telah disesuaikan, serta `dictionary` yang merupakan kamus pemetaan kata. Kedua luaran inilah yang akan menjadi input utama dalam pembangunan model *Latent Dirichlet Allocation* (LDA).

4.4 *Topik Modelling* dengan LDA

Setelah Setelah representasi numerik dokumen terbentuk melalui pembobotan TF-IDF, tahapan inti penelitian ini adalah pemodelan topik menggunakan algoritma *Latent Dirichlet Allocation* (LDA). Salah satu tantangan fundamental dalam penerapan LDA sebagai metode *unsupervised learning* adalah penentuan jumlah topik yang optimal, karena algoritma ini tidak mengetahui secara apriori berapa banyak kategori topik yang tersembunyi dalam korpus data. Penentuan jumlah topik yang tepat sangat krusial; jumlah topik yang terlalu sedikit dapat menggabungkan topik-topik yang berbeda (*underfitting*), sedangkan jumlah topik yang terlalu banyak dapat memecah satu topik menjadi bagian-bagian yang terlalu spesifik dan sulit diinterpretasikan (*overfitting*). Oleh karena itu, diperlukan evaluasi kinerja model berbasis *Coherence Score* untuk menemukan titik keseimbangan yang paling representatif.

Guna menjamin validitas penentuan jumlah topik dan memastikan hasil yang diperoleh bukan merupakan kebetulan semata, penelitian ini menerapkan dua skenario eksperimen terpisah. Eksperimen pertama dilakukan dengan melatih model menggunakan sintaks

`range(1, 11)` untuk melihat pembentukan topik pada skala 1 hingga 10. Selanjutnya, eksperimen kedua dilakukan dengan cakupan yang lebih lebar menggunakan sintaks `range(1, 16)`, yaitu dari 1 hingga 15 topik. Pelaksanaan dua percobaan ini bertujuan untuk menguji konsistensi performa model dan memverifikasi apakah puncak nilai koherensi tetap bertahan pada angka yang sama meskipun jumlah topik diuji lebih lanjut. Pada setiap iterasi, parameter pelatihan dikontrol secara ketat. Parameter `random_state` diatur pada nilai 42 untuk mengeliminasi faktor acak pada algoritma, sehingga menjamin konsistensi hasil eksperimen di mana model akan menghasilkan luaran yang sama persis setiap kali dijalankan. Selanjutnya, parameter `passes` ditetapkan sebesar 20, yang berarti algoritma melakukan pembacaan penuh terhadap seluruh korpus data sebanyak 20 putaran untuk memastikan model mencapai kestabilan. Implementasi kode program untuk pengujian ini ditampilkan pada Gambar 4.21.

```
import matplotlib.pyplot as plt
from gensim.models import CoherenceModel, LdaModel

coherence_scores = []
topic_range = range(1, 16)

for num_topics in topic_range:
    lda_model = LdaModel(
        corpus=corpus_for_lda,
        id2word=dictionary,
        num_topics=num_topics,
        random_state=42,
        passes=20,
        alpha='auto'
    )

    coherence_model = CoherenceModel(
        model=lda_model,
        texts=texts,
        dictionary=dictionary,
        coherence='c_v'
    )

    score = coherence_model.get_coherence()
    coherence_scores.append(score)
    print(f"Num Topics = {num_topics}, Coherence Score = {score:.4f}")

plt.figure(figsize=(10, 6))
plt.plot(topic_range, coherence_scores, marker='o', linestyle='-', color='b')
plt.title('Coherence Score Berdasarkan Jumlah Topik')
plt.xlabel('Jumlah Topik')
plt.ylabel('Coherence Score (c_v)')
plt.xticks(topic_range)
plt.grid(True)
plt.savefig('grafik_coherence_score.png', bbox_inches='tight')
plt.show()
```

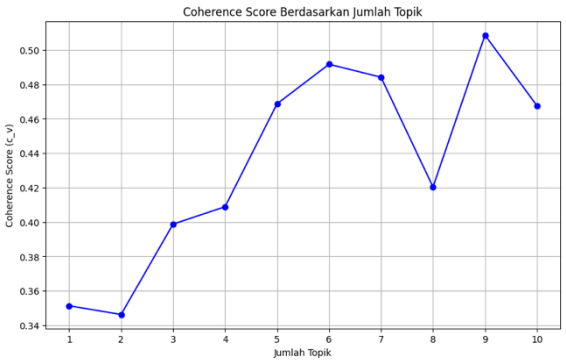
Gambar 4.21 Kode Program Pengujian Jumlah Topik

Untuk mengukur kualitas topik yang dihasilkan pada setiap skenario, penelitian ini menggunakan metrik *Coherence Score*. Metrik ini mengukur tingkat keterkaitan semantik antar kata dalam satu topik, di mana nilai yang semakin tinggi mengindikasikan bahwa topik tersebut semakin mudah dipahami dan diinterpretasikan oleh manusia. Hasil perhitungan *Coherence Score* dari kedua skenario eksperimen tersebut kemudian dikomparasi sebagaimana ditampilkan pada Tabel 4.11.

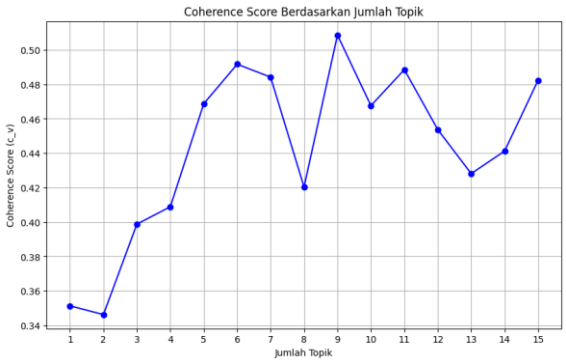
Tabel 4.11 *Conherence Score*

Limit	**10**		**15**	
	Num Topics	**Coherence Score**	**Num Topics**	**Coherence Score**
	1	0.3513	1	0.3513
	2	0.3462	2	0.3462
	3	0.3988	3	0.3988
	4	0.4089	4	0.4089
	5	0.4687	5	0.4687
	6	0.4917	6	0.4917
	7	0.4842	7	0.4842
	8	0.4203	8	0.4203
	9	**0.5086**	**9**	**0.5086**
	10	0.4676	10	0.4676
			11	0.4885
			12	0.4537
			13	0.4281
			14	0.4412
			15	0.4823

Berdasarkan data pada Tabel 4.11, terlihat adanya keselarasan hasil antara percobaan pertama dan kedua. Nilai koherensi mengalami fluktuasi pada jumlah topik rendah, namun mencapai titik kulminasi atau puncak tertinggi secara konsisten pada percobaan 9 topik dengan skor 0,5086. Temuan krusial dari eksperimen kedua (rentang 1-15) memperlihatkan bahwa setelah melewati angka 9 khususnya pada topik ke-11 hingga ke-15 skor koherensi justru mengalami tren penurunan dan tidak pernah melampaui skor yang dicapai oleh 9 topik. Penurunan ini mengindikasikan bahwa memecah data menjadi lebih dari 9 kategori tidak memberikan struktur semantik yang lebih baik, melainkan justru melemahkan kepadatan makna dalam topik. Tren perubahan nilai koherensi ini divisualisasikan lebih lanjut dalam bentuk grafik garis pada Gambar 4.22 dan Gambar 4.23.



Gambar 4.22 Visualisasi Limit 10



Gambar 4.23 Visualisasi Limit 15

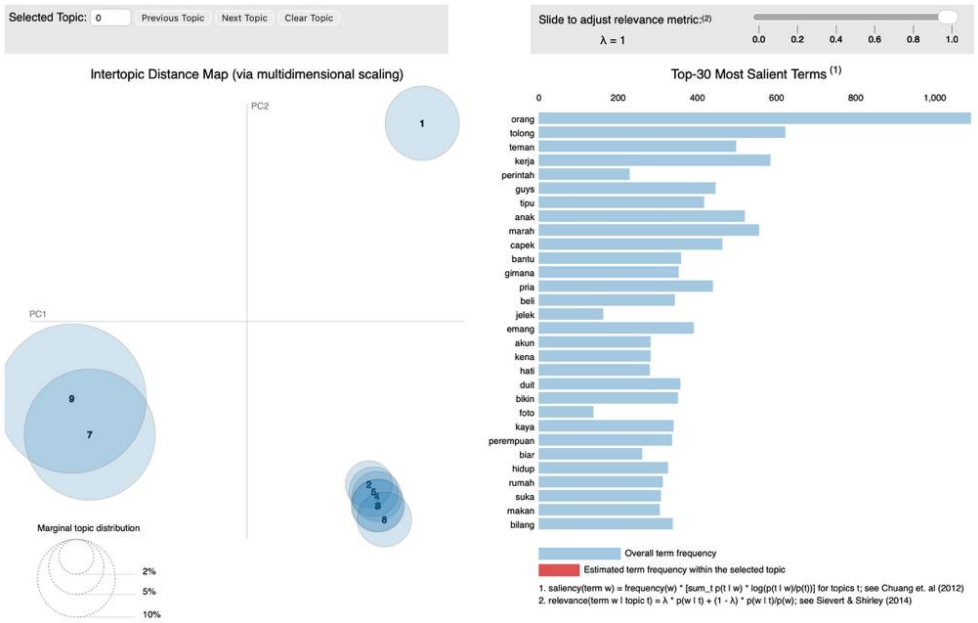
Merujuk pada konsistensi visualisasi grafik dan data tabel di atas, titik tertinggi pada sumbu Y secara jelas berada pada angka 9. Oleh karena itu, jumlah topik terbaik (*best number of topics*) ditetapkan sebanyak 9 topik. Nilai inilah yang kemudian digunakan sebagai parameter kunci untuk melatih ulang model LDA final (`lda_model_final`), sebagaimana ditunjukkan pada kode program di Gambar 4.24. Model final ini dibangun menggunakan korpus TF-IDF dengan parameter `alpha='auto'`, yang memungkinkan algoritma mempelajari distribusi prior topik dokumen secara otomatis dari data, guna menghasilkan pembagian topik yang paling akurat untuk dianalisis lebih lanjut.

```
best_num_topics = topic_range[np.argmax(coherence_scores)]
print("Jumlah topik terbaik:", best_num_topics)

lda_model_final = LdaModel(
    corpus=corpus_for_lda,
    id2word=dictionary,
    num_topics=best_num_topics,
    random_state=42,
    passes=20,
    alpha='auto'
)
```

Gambar 4.24 Kode Program Pelatihan Model Final

Untuk memahami karakteristik semantik dan hubungan antar topik yang dihasilkan oleh model final, dilakukan visualisasi interaktif menggunakan pustaka pyLDAvis. Metode ini memetakan topik ke dalam ruang dua dimensi berdasarkan prinsip *Multidimensional Scaling* (MDS), yang memungkinkan representasi jarak semantik antar topik diamati secara visual melalui *Intertopic Distance Map*. Sebagaimana ditampilkan pada Gambar 4.25, peta jarak antar topik memperlihatkan distribusi sembilan topik yang terbentuk.



Gambar 4.25 Visualisasi LDA

Berdasarkan visualisasi di atas, teridentifikasi tiga pola distribusi utama yang mencerminkan struktur semantik komunitas ini. Pertama, Topik 1 terlihat terisolasi jauh di Kuadran 1 (Kanan Atas) dengan ukuran lingkaran yang dominan. Posisi yang menyendiri ini menandakan bahwa Topik 1 membahas tema utama yang sangat distinktif, yakni isu layanan publik dan kebijakan pemerintah, yang secara kosakata tidak memiliki irisan makna dengan topik lain. Kedua, terdapat pasangan Topik 7 dan Topik 9 yang berukuran besar dan saling berhimpitan (*overlapping*) di perbatasan Kuadran 2 dan 3, yang mengindikasikan adanya kedekatan konteks pembahasan yang sangat erat. Ketiga, Topik 2, 3, 4, 5, 6, dan 8 membentuk sebuah klaster padat di Kuadran 4 (Kanan Bawah). Jarak yang sangat dekat antar topik dalam klaster ini menunjukkan bahwa mereka merupakan variasi nuansa dari keresahan harian yang saling berkaitan satu sama lain.

Secara lebih detail, pola irisan pada peta visualisasi menunjukkan bahwa Topik 1 tetap berdiri sendiri tanpa irisan. Sebaliknya, Topik 2, 3, 4, 5, 6, dan 8 saling beririsan satu sama lain, sementara Topik 7 memiliki irisan yang signifikan dengan Topik 9. Fenomena tumpang tindih (*overlapping*) yang cukup masif pada beberapa topik ini diinterpretasikan bukan sebagai kegagalan model dalam memisahkan topik, melainkan sebagai refleksi dari keterkaitan isu yang nyata dalam realitas sosial pengguna. Terdapat beberapa alasan mendasar yang mendukung justifikasi ini.

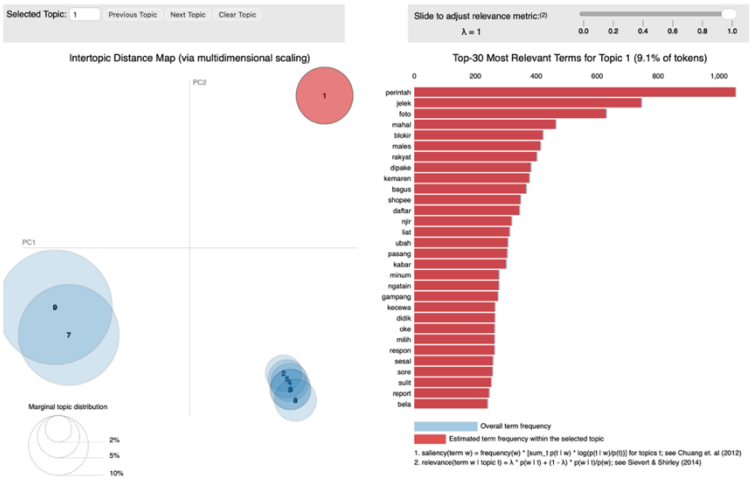
Pertama, irisan tersebut dipicu oleh keseragaman linguistik dan muatan emosional pada media sosial X. Sebagai komunitas "Marah-Marah", terdapat penggunaan kata makian dan slang yang sangat seragam di hampir seluruh tweet, yang secara otomatis menarik topik-topik tersebut ke dalam ruang semantik yang berdekatan. Kedua, banyak isu dalam komunitas ini bersifat multidimensi, sebagai contoh keterkaitan antara Topik 7 (penipuan) dan Topik 9 (keluarga) terjadi karena narasi pengguna sering kali menghubungkan masalah anggota keluarga yang terjerat utang atau judi online dengan tindakan penipuan.

Ketiga, validitas pemisahan topik ini didukung oleh nilai *Coherence Score* sebesar 0,5086, yang mengindikasikan bahwa meskipun secara visual berhimpitan, setiap topik tetap memiliki stabilitas makna dan kumpulan kata kunci yang solid secara statistik. Dengan demikian, visualisasi pyLDAvis ini justru berhasil menangkap ambiguitas alami dari percakapan manusia di media sosial, di mana keresahan personal, sosial, dan ekonomi sering kali muncul secara bersamaan dalam satu konteks pembicaraan yang utuh.

Berdasarkan kedekatan posisi pada peta visualisasi, teridentifikasi pola irisan topik sebagai berikut:

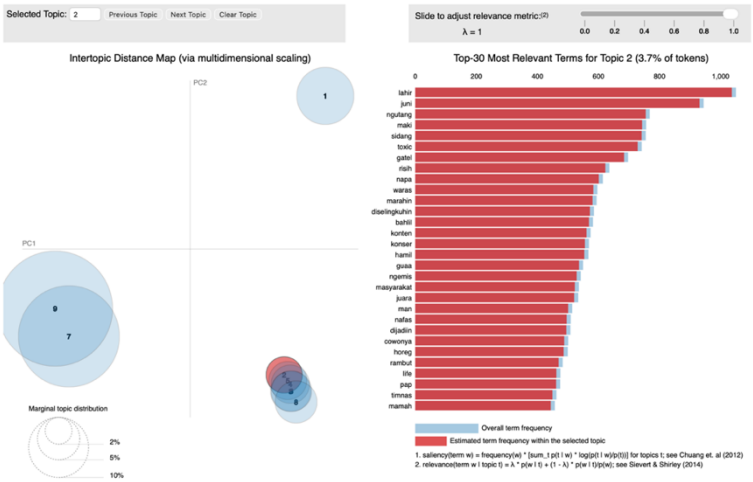
- Topik 1 tidak beririsan dengan topik lain.
- Topik 2 beririsan dengan Topik 3,4,5,6, dan 8.
- Topik 3 beririsan dengan Topik 2,4,5,6, dan 8.
- Topik 4 beririsan dengan Topik 2,3,5,6, dan 8.
- Topik 5 beririsan dengan Topik 2,3,4,6, dan 8.
- Topik 6 beririsan dengan Topik 2,3,4,5, dan 8.
- Topik 7 beririsan dengan Topik 9.
- Topik 8 beririsan dengan Topik 2,3,4,5, dan 6.
- Topik 9 beririsan dengan Topik 7.

Analisis lebih mendalam dilakukan dengan membedah kata kunci dominan (*top salient terms*) yang menyusun setiap topik untuk mengungkap makna substantif di balik pengelompokan tersebut.



Gambar 4.26 Visualisasi Topik-1

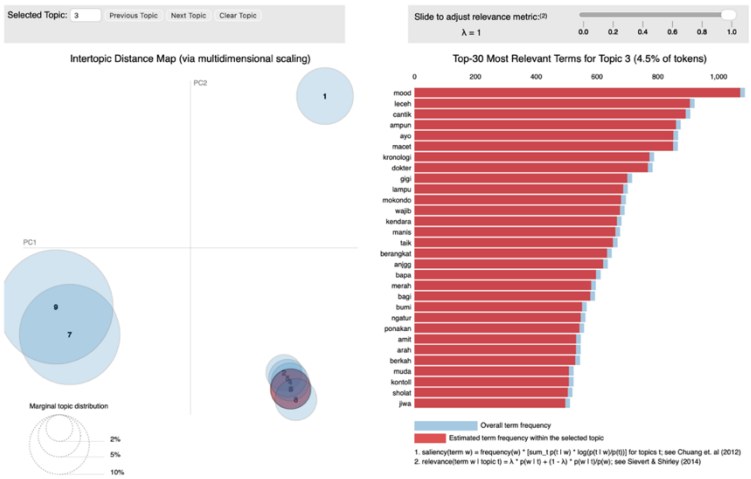
Distribusi kata yang muncul pada Topik 1 dapat dilihat pada Gambar 4.26 Komposisi kata pada topik ini meliputi: ‘0.025*"perintah" + 0.018*"jelek" + 0.015*"foto" + 0.011*"mahal" + 0.010*"blokir" + 0.010*"males" + 0.010*"rakyat" + 0.009*"dipake" + 0.009*"kemaren" + 0.009*"bagus"’. Nilai tertinggi yang muncul pada kata "perintah" dan "jelek".



Gambar 4.27 Visualisasi Topik-2

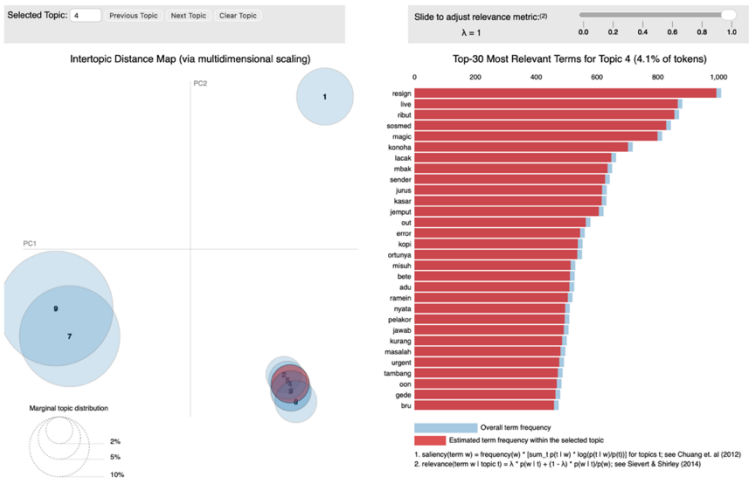
Selanjutnya, representasi kata untuk Topik 2 ditampilkan pada Gambar 4.27. Kata-kata yang memiliki probabilitas tertinggi dan mewakilkan topik ini yaitu: ‘0.018*"lahir" + 0.016*"juni" + 0.013*"ngutang" + 0.013*"maki" + 0.013*"sidang" + 0.013*"toxic" +

0.012*"gatel" + 0.011*"risih" + 0.010*"napa" + 0.010*"waras"'. Nilai tertinggi terdapat pada keyword "lahir" dan "juni".



Gambar 4.28 Visualisasi Topik-3

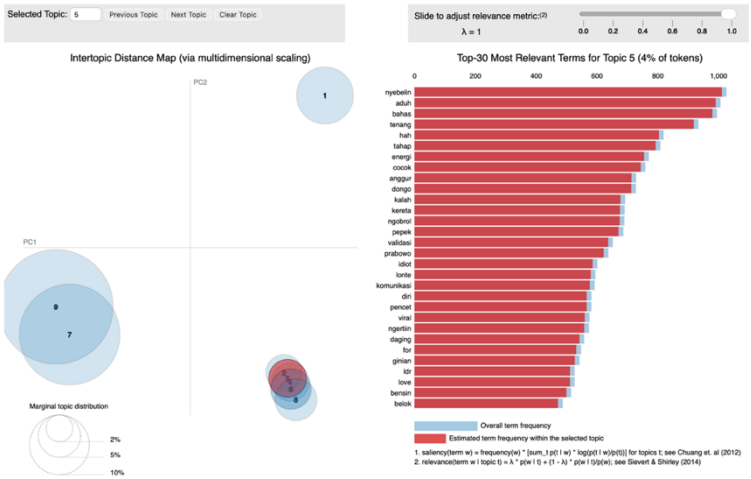
Pada Topik 3, kata-kata yang muncul diperlihatkan pada Gambar 4.28. Daftar kata yang mewakili topik ini yaitu: '0.013*"mood" + 0.011*"leceh" + 0.011*"cantik" + 0.010*"ampun" + 0.010*"ayo" + 0.010*"macet" + 0.009*"kronologi" + 0.009*"dokter" + 0.008*"gigi" + 0.008*"lampu"'. Dengan bobot yang sangat menonjol pada kata "mood", "leceh", dan "cantik.



Gambar 4.29 Visualisasi Topik-4

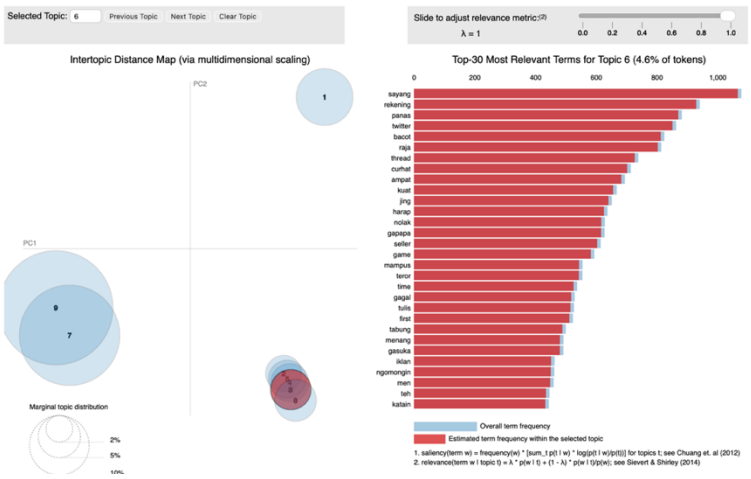
Distribusi kata kunci untuk Topik 4 terlihat pada Gambar 4.29. Kata-kata penyusun utamanya adalah: '0.013*"resign" + 0.012*"live" + 0.011*"ribut" + 0.011*"sosmed" +

0.011*"magic" + 0.009*"konoha" + 0.009*"lacak" + 0.009*"mbak" + 0.008*"sender" + 0.008*"jurus"'. Dominasi terdapat pada kata "resign" dan "live".



Gambar 4.30 Visualisasi Topik-5

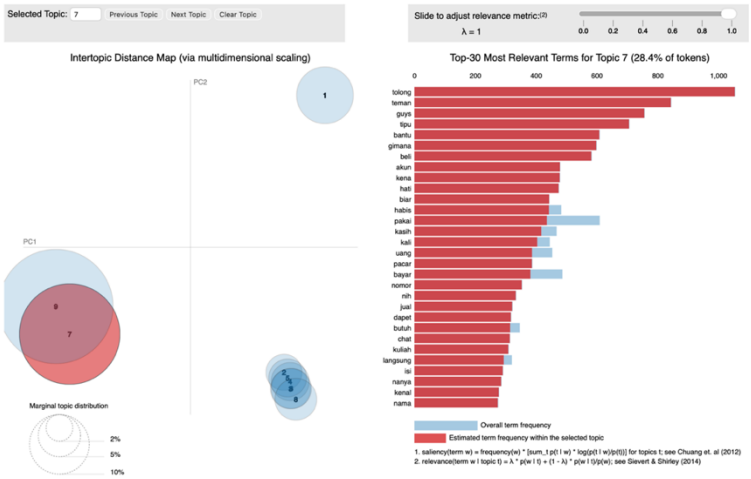
Beralih ke Topik 5, struktur kata penyusunnya disajikan pada Gambar 4.30. Kata-kata yang memiliki probabilitas tertinggi dan mewakili topik ini adalah: '0.014*"nyebelin" + 0.013*"aduh" + 0.013*"bahas" + 0.012*"tenang" + 0.011*"hah" + 0.011*"tahap" + 0.010*"energi" + 0.010*"cocok" + 0.010*"anggur" + 0.010*"dongo"'. Nilai tertinggi terdapat pada keyword "nyebelin", "aduh", dan "bahas".



Gambar 4.31 Visualisasi Topik-6

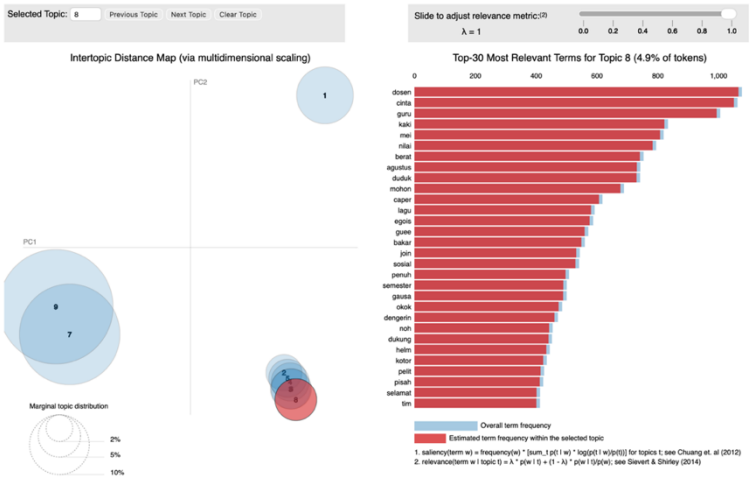
Untuk Topik 6, visualisasi kata kuncinya ditampilkan pada Gambar 4.31. Kata-kata yang menjadi penciri utama topik ini adalah: '0.017*"sayang" + 0.015*"rekening" + 0.014*"panas"

+ 0.014*"twitter" + 0.013*"bacot" + 0.013*"raja" + 0.012*"thread" + 0.011*"curhat" + 0.011*"ampat" + 0.010*"kuat"'. Dominasi terdapat pada kata "sayang" dan "rekening".



Gambar 4.32 Visualisasi Topik-7

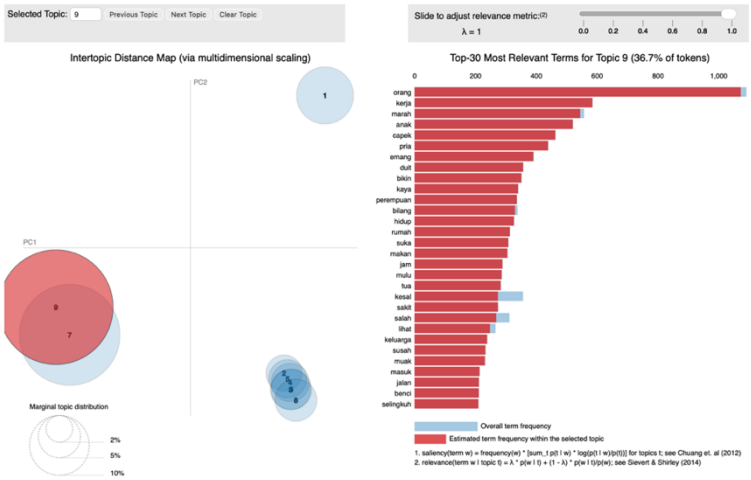
Kata yang muncul pada Topik 7 terlihat pada Gambar 4.32. Komposisi kata kunci pada topik ini meliputi: '0.022*"tolong" + 0.018*"teman" + 0.016*"guys" + 0.015*"tipu" + 0.013*"bantu" + 0.012*"gimana" + 0.012*"beli" + 0.010*"akun" + 0.010*"kena" + 0.010*"hati"'. Kata dengan bobot terbesar adalah "tolong" dan "teman".



Gambar 4.33 Visualisasi Topik-8

Selanjutnya, Topik 8 memiliki distribusi kata yang dapat dilihat pada Gambar 4.33. Deretan kata dengan probabilitas tertinggi yaitu: '0.016*"dosen" + 0.016*"cinta" +

0.015*"guru" + 0.012*"kaki" + 0.012*"mei" + 0.012*"nilai" + 0.011*"berat" + 0.011*"agustus" + 0.011*"duduk" + 0.010*"mohon"'. Berdasarkan bobot tertingginya pada kata "dosen" dan "cinta".



Gambar 4.34 Visualisasi Topik-9

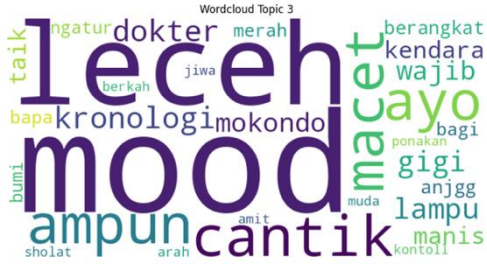
Terakhir, kata kunci yang membentuk Topik 9 ditampilkan pada Gambar 4.34. Kata-kata dengan bobot signifikansi tertinggi meliputi: '0.029*"orang" + 0.016*"kerja" + 0.015*"marah" + 0.014*"anak" + 0.013*"capek" + 0.012*"pria" + 0.011*"emang" + 0.010*"duit" + 0.010*"bikin" + 0.009*"kaya"'. Dua nilai tertinggi pada *keyword* "orang" dan "kerja".

Selanjutnya selain representasi diagram batang, hasil pemodelan topik juga divisualisasikan dalam bentuk *wordcloud* untuk memberikan gambaran intuitif mengenai distribusi kata dominan. Visualisasi ini mempermudah interpretasi cepat terhadap esensi setiap topik yang dihasilkan oleh model LDA. Dalam representasi ini, ukuran *font* setiap kata berbanding lurus dengan nilai probabilitas atau bobotnya dalam topik tersebut; semakin besar ukuran kata yang ditampilkan, semakin tinggi kontribusi kata tersebut dalam mendefinisikan topik.

Mengingat model optimal menghasilkan 9 topik, maka disajikan sembilan visualisasi *Word Cloud* yang merepresentasikan karakteristik unik masing-masing topik. Rangkaian visualisasi ini ditunjukkan secara berurutan pada Gambar 4.35 hingga Gambar 4.43.

Gambar 4.35 *Wordcloud* 1

Gambar 4.36 *Wordcloud* 2



Gambar 4.37 *Wordcloud* 3

Gambar 4.38 *Wordcloud* 4

Gambar 4.39 *Wordcloud* 5

Gambar 4.40 *Wordcloud* 6

Gambar 4.41 *Wordcloud* 7

Gambar 4.42 *Wordcloud* 8

Gambar 4.43 *Wordcloud* 9

Implementasi pembangkitan *Word Cloud* ini dilakukan secara otomatis melalui kode program yang ditampilkan pada Gambar 4.44. Tahapan teknis dimulai dengan mengekstraksi 30 kata teratas beserta bobotnya dari `lda_model_final` menggunakan fungsi `show_topics(formatted=False)`. Program kemudian melakukan iterasi (*looping*) terhadap setiap topik. Pada setiap iterasi, pasangan kata dan bobot dikonversi menjadi tipe data *dictionary* (`dict`), yang selanjutnya menjadi input untuk fungsi `generate_from_frequencies`. Fungsi inilah yang memetakan bobot kata menjadi ukuran visual pada kanvas. Terakhir, pustaka `matplotlib` digunakan untuk me-*render* gambar, menghilangkan sumbu koordinat (`axis('off')`) agar tampilan lebih bersih, serta menyimpan hasilnya secara otomatis ke dalam format gambar png sesuai nomor urut topik.

```
import matplotlib.pyplot as plt
from wordcloud import WordCloud

topics = lda_model_final.show_topics(
    num_topics=best_num_topics,
    formatted=False,
    num_words=30
)

topics = sorted(topics, key=lambda x: x[0])

for topic in topics:
    topic_id = topic[0]
    topic_words_list = topic[1]

    word_dict = dict(topic_words_list)

    wc = WordCloud(
        width=800,
        height=400,
        background_color='white',
        colormap='viridis',
        random_state=42
    ).generate_from_frequencies(word_dict)

    plt.figure(figsize=(10, 5))
    plt.imshow(wc, interpolation='bilinear')
    plt.axis('off')

    plt.title(f"Wordcloud Topic {topic_id + 1}")

    plt.savefig(f'wordcloud_topic_{topic_id + 1}.png', bbox_inches='tight')
    plt.show()
```

Gambar 4.44 Kode Program *Wordcloud*

4.5 Analisis Hasil

Berdasarkan hasil *topic modeling*, dilakukan analisis secara kualitatif untuk memahami konteks percakapan yang terbentuk dalam komunitas. Kata-kata yang muncul dalam setiap

topik diinterpretasikan agar lebih mudah dipahami maknanya oleh pembaca. Tabel 4.12 sebelumnya telah menunjukkan sepuluh kata dominan pada setiap topik.

Tabel 4.12 Dominasi Kata Pada Topik

Topik	Model
1	‘0.025*"perintah" + 0.018*"jelek" + 0.015*"foto" + 0.011*"mahal" + 0.010*"blokir" + 0.010*"males" + 0.010*"rakyat" + 0.009*"dipake" + 0.009*"kemaren" + 0.009*"bagus"’
2	‘0.018*"lahir" + 0.016*"juni" + 0.013*"ngutang" + 0.013*"maki" + 0.013*"sidang" + 0.013*"toxic" + 0.012*"gatel" + 0.011*"risih" + 0.010*"napa" + 0.010*"waras"’
3	‘0.013*"mood" + 0.011*"leceh" + 0.011*"cantik" + 0.010*"ampun" + 0.010*"ayo" + 0.010*"macet" + 0.009*"kronologi" + 0.009*"dokter" + 0.008*"gigi" + 0.008*"lampu"’
4	‘0.013*"resign" + 0.012*"live" + 0.011*"ribut" + 0.011*"sosmed" + 0.011*"magic" + 0.009*"konoha" + 0.009*"lacak" + 0.009*"mbak" + 0.008*"sender" + 0.008*"jurus"’
5	‘0.014*"nyebelin" + 0.013*"aduh" + 0.013*"bahas" + 0.012*"tenang" + 0.011*"hah" + 0.011*"tahap" + 0.010*"energi" + 0.010*"cocok" + 0.010*"anggur" + 0.010*"dongo"’
6	‘0.017*"sayang" + 0.015*"rekening" + 0.014*"panas" + 0.014*"twitter" + 0.013*"bacot" + 0.013*"raja" + 0.012*"thread" + 0.011*"curhat" + 0.011*"ampat" + 0.010*"kuat"’
7	‘0.022*"tolong" + 0.018*"teman" + 0.016*"guys" + 0.015*"tipu" + 0.013*"bantu" + 0.012*"gimana" + 0.012*"beli" + 0.010*"akun" + 0.010*"kena" + 0.010*"hati"’
8	‘0.016*"dosen" + 0.016*"cinta" + 0.015*"guru" + 0.012*"kaki" + 0.012*"mei" + 0.012*"nilai" + 0.011*"berat" + 0.011*"agustus" + 0.011*"duduk" + 0.010*"mohon"’
9	‘0.029*"orang" + 0.016*"kerja" + 0.015*"marah" + 0.014*"anak" + 0.013*"capek" + 0.012*"pria" + 0.011*"emang" + 0.010*"duit" + 0.010*"bikin" + 0.009*"kaya"’

Topik 1 menampilkan kata kunci perintah, jelek, mahal, foto, rakyat. Topik ini menggarisbawahi ketidakpuasan masyarakat terhadap kinerja pemerintah dan kualitas layanan publik. Konteks yang muncul mencakup keluhan terhadap biaya pendidikan (UKT) yang mahal, birokrasi layanan kesehatan (BPJS), hingga kritik terhadap fasilitas publik yang dianggap tidak memadai atau "jelek". Pengguna merasa sebagai "rakyat" yang terbebani oleh kebijakan yang menyulitkan.

Topik 2 didominasi oleh kata *lahir, ngutang, maki, toxic, waras*. Topik ini merepresentasikan konflik interpersonal dan tekanan sosial dalam lingkup kedekatan. Percakapan di sini berfokus pada keresahan terhadap perilaku "toxic", baik dari orang tua

(terkait biaya hidup/uang) maupun dalam hubungan pertemanan. Penggunaan kata "waras" menunjukkan adanya tekanan mental akibat interaksi sosial yang tidak sehat tersebut.

Topik 3 memiliki kata kunci *mood, leceh, macet, kronologi, dokter*. Topik ini berkaitan erat dengan isu keamanan publik dan ketidaknyamanan sehari-hari. Konteks yang kuat dalam topik ini adalah pelaporan tindak kriminalitas seperti begal payudara dan pencurian kendaraan bermotor, di mana pengguna sering membagikan "kronologi" kejadian. Selain itu, keluhan mengenai kemacetan jalanan juga menjadi bagian dari penurunan *mood* pengguna.

Topik 4 memuat kata *resign, live, ribut, lacak, konoha*. Topik ini mencerminkan dinamika dunia kerja dan keamanan digital. Muncul narasi mengenai keinginan untuk berhenti kerja (*resign*) hingga upaya "melacak" perilaku pasangan atau penipu. Istilah "Konoha" sering digunakan sebagai metafora untuk menyebut situasi di Indonesia dalam konteks sindiran sosial.

Topik 5 menampilkan kata *nyebelin, bahas, energi, cocok, dongo*. Topik ini mengarah pada rasa frustrasi terhadap interaksi sosial dan standar fisik di masyarakat. Keresahan yang muncul berkaitan dengan kriteria fisik dalam mencari pasangan (tinggi badan/berat badan) yang dianggap melelahkan atau "nyebelin". Topik ini menunjukkan adanya kelelahan sosial (*social fatigue*) dalam menghadapi ekspektasi lingkungan.

Topik 6 didominasi oleh kata sayang, rekening, panas, thread, curhat. Topik ini merupakan ruang bagi pengguna untuk melakukan *storytelling* atau berbagi cerita panjang melalui "thread". Kontennya sangat beragam, mulai dari masalah hak asuh anak, curhatan personal tentang keuangan (rekening), hingga keluhan tentang cuaca ekstrem (panas) di berbagai wilayah di Indonesia.

Topik 7 memiliki kata kunci tolong, teman, tipu, bantu, akun. Topik ini secara spesifik berfungsi sebagai "alarm" atau sistem peringatan dini bagi komunitas. Konteks utamanya adalah laporan penipuan belanja online (olshop), jastip fiktif, dan modus penipuan di media sosial. Kata "tolong" dan "bantu" menunjukkan adanya solidaritas digital untuk menyebarkan informasi agar orang lain tidak menjadi korban.

Topik 8 menampilkan kata dosen, guru, cinta, nilai, duduk. Topik ini mengangkat keresahan di institusi pendidikan dan etika di ruang publik. Pengguna mengeluhkan beban akademik (tugas dari guru/dosen) serta jadwal ujian yang padat.

Topik 9 didominasi oleh kata orang, kerja, marah, anak, capek, duit. Topik ini menggambarkan beban domestik dan tanggung jawab keluarga yang berat. Konteks yang sangat kental adalah keluhan terhadap anggota keluarga (kakak atau adik) yang menjadi beban

finansial karena pengangguran atau jeratan judi online. Kata "capek" merefleksikan kelelahan psikis pengguna yang harus menanggung beban ekonomi anggota keluarga lainnya.

Setelah melakukan interpretasi terhadap setiap topik melalui kata kunci dominan, langkah selanjutnya adalah memvalidasi konteks tersebut menggunakan dokumen asli yang memiliki kaitan paling kuat dengan topik yang terbentuk. Untuk memperkuat representasi, dilakukan pencarian contoh *tweet* yang merepresentasikan topik-topik tersebut. Gambar 4.45 menampilkan kode program pencarian *tweet representative* berdasarkan probabilitas.

```
all_topics_data = []

for topic_id in range(lda_model_final.num_topics):
    docs = representative_data_topn(
        lda_model_final,
        corpus_for_lda,
        texts,
        topic_id=topic_id,
        top_n=20
    )

    for doc_id, text, prob in docs:
        all_topics_data.append({
            'topic_id': topic_id + 1,
            'document_id': doc_id,
            'probability': prob,
            'text': ' '.join(text) if isinstance(text, list) else text
        })

df_representative = pd.DataFrame(all_topics_data)
df_representative.head(20)
```

Gambar 4.45 Kode Program *Tweet Representative*

Proses identifikasi *tweet representative* ini dilakukan dengan mengekstraksi dokumen yang memiliki nilai kontribusi probabilitas tertinggi pada setiap topik yang dihasilkan oleh model LDA. Secara teknis, sistem melakukan iterasi pada setiap topik melalui perintah `for topic_id in range (lda_model_final.num_topics)` dan memanggil fungsi `representative_data_topn`. Fungsi tersebut bekerja dengan memindai seluruh korpus (`corpus_for_lda`) untuk mencocokkan distribusi topik pada dokumen asli (texts).

Dalam kode tersebut, digunakan parameter `top_n=20`, yang berarti peneliti mengambil 20 dokumen terbaik dengan skor probabilitas tertinggi untuk setiap topik. Data yang diekstraksi kemudian disimpan ke dalam *list* `all_topics_data` yang mencakup kolom `topic_id` (nomor topik), `document_id` (indeks dokumen), `probability` (nilai kedekatan dokumen dengan topik), serta `text` (konten tweet).

Hasil akhir dari proses ini kemudian dikonversi menjadi sebuah *DataFrame* (`df_representative`) untuk mempermudah verifikasi kualitatif. Tahapan ini sangat krusial karena memungkinkan peneliti untuk melihat bagaimana kata kunci seperti 'perintah', 'tipu', atau 'capek' bekerja dalam sebuah kalimat utuh. Dengan demikian, validasi antara hasil komputasi mesin (LDA) dan makna kontekstual yang dirasakan pengguna di komunitas 'marah-marah' dapat dilakukan secara lebih akurat dan objektif. *Tweet representative* dapat dilihat dalam Lampiran C.

4.6 Kelebihan dan Kekurangan

Berdasarkan seluruh rangkaian proses eksperimen dan analisis yang telah dilakukan, penelitian ini memiliki sejumlah kekuatan signifikan serta beberapa keterbatasan yang perlu dievaluasi. Kelebihan Penelitian Kekuatan fundamental dari penelitian ini terletak pada ketatnya strategi *preprocessing* data yang dirancang untuk menangani karakteristik *noise* tinggi pada media sosial X. Penelitian ini tidak hanya menerapkan pembersihan standar, melainkan mengimplementasikan alur kerja berlapis yang meliputi Advanced Duplicate Removal dan Normalization. Strategi eliminasi duplikasi yang dilakukan secara bertahap terbukti sangat efektif dalam menyaring data, di mana dari total akuisisi awal sebanyak 75.032 *tweet*, berhasil disaring menjadi 38.887 data unik yang benar-benar berkualitas. Proses ini menjamin bahwa model LDA tidak bias terhadap *tweet* spam atau bot yang berulang, sehingga topik yang dihasilkan murni merepresentasikan variasi opini pengguna asli.

Selain dari aspek kualitas data, kekuatan penelitian ini juga terlihat pada optimalisasi konfigurasi model. Penerapan algoritma LDA dengan pembobotan TF-IDF yang diuji melalui rentang topik iteratif berhasil menemukan titik keseimbangan terbaik pada 9 topik dengan nilai *Coherence Score* sebesar 0,5086. Nilai ini mengindikasikan bahwa topik-topik yang terbentuk memiliki stabilitas semantik yang tinggi.

Lebih lanjut, penelitian ini berhasil menyajikan interpretasi substantif yang komprehensif melalui pemetaan masalah yang sangat spesifik. Model mampu membedakan secara jelas antara kemarahan yang bersifat sistemik (seperti isu layanan publik dan biaya pendidikan pada Topik 1), kemarahan sosial yang bersifat kewaspadaan kolektif (isu penipuan online pada Topik 7 dan kriminalitas jalanan pada Topik 3), hingga peluapan emosi pada ranah privat (Topik 9). Kemampuan model untuk mengisolasi topik-topik spesifik seperti memisahkan keluhan "dinamika *toxic* keluarga" (Topik 9) dari "konflik relasi interpersonal" (Topik 2) serta menangkap keresahan institusional pada sektor pendidikan (Topik 8), menunjukkan bahwa

metode yang digunakan sangat sensitif dalam menangkap nuansa konteks yang berbeda dalam satu komunitas yang sama.

Meskipun menghasilkan topik yang baik, penelitian ini masih memiliki beberapa keterbatasan teknis dan metodologis yang menjadi catatan untuk perbaikan di masa depan:

a. Pada hasil visualisasi *Wordcloud*, masih ditemukan kemunculan kata-kata slang yang lolos dari proses penyaringan. Hal ini terjadi karena penelitian ini masih bergantung pada kamus *stopwords* standar pustaka NLTK. Untuk memperbaiki hal ini, penggunaan kamus *custom stopwords* yang disusun manual berdasarkan frekuensi kata di korpus sangat disarankan agar topik menjadi lebih bersih.
b. Penelitian ini berfokus murni pada pemetaan topik (apa yang dibicarakan), namun belum menyentuh dimensi emosi (bagaimana perasaannya). Mengingat objek penelitian adalah "Komunitas MARAH MARAH", sangat potensial jika topik yang terbentuk dikaitkan dengan analisis sentimen atau emosi. Hal ini akan memperkaya wawasan mengenai intensitas kemarahan pada setiap topik.
c. Penelitian ini hanya menggunakan algoritma LDA tanpa membandingkannya dengan metode lain. Akibatnya, tidak ada tolok ukur komparatif untuk menilai apakah LDA adalah metode terbaik untuk data ini.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Strategi *preprocessing* yang dirancang dalam penelitian ini berhasil menangani karakteristik data teks media sosial yang sangat tidak terstruktur melalui serangkaian tahapan transformasi data yang komprehensif. Proses pengubahan data dimulai dengan tahap pembersihan intensif (*remove punctuation* dan *case folding*) untuk menghilangkan *noise* seperti simbol, angka, dan elemen non-alfabet, serta menyeragamkan format teks. Strategi ini kemudian diperdalam dengan teknik normalisasi (*normalization*) yang secara efektif mengubah variasi kata tidak baku, singkatan, dan *slang* menjadi bentuk standar, serta *tokenizing* yang memecah kalimat menjadi unit kata untuk keperluan analisis. Guna mempertajam kualitas data agar model hanya berfokus pada informasi substantif, diterapkan tahapan *stopwords removal* dan *stemming* yang mentransformasi kata berimbuhan menjadi kata dasar. Seluruh rangkaian proses ini disempurnakan dengan teknik *duplicate removal* yang ketat untuk mengeliminasi redundansi, yang terbukti berhasil menyaring volume data dari 75.032 *tweet* mentah menjadi 38.887 data unik yang bersih, terstruktur, dan siap untuk tahap pemodelan.

Berdasarkan hasil pengujian model *Latent Dirichlet Allocation* (LDA) dengan pembobotan TF-IDF terhadap data yang telah diproses, penelitian menyimpulkan bahwa pembagian data ke dalam 9 (sembilan) topik adalah konfigurasi yang paling optimal. Hal ini dibuktikan dengan pencapaian nilai *Coherence Score* tertinggi sebesar 0,5086, yang menunjukkan tingkat kestabilan yang lebih baik dibandingkan variasi jumlah topik lainnya. Nilai koherensi ini mengindikasikan bahwa model berhasil membentuk kelompok-kelompok topik dengan struktur semantik yang jelas, solid, dan memiliki keterkaitan makna yang kuat sehingga mudah diinterpretasikan.

Secara substantif, sembilan topik yang berhasil diidentifikasi dalam komunitas 'Marah Marah' di X memetakan dimensi utama sumber keresahan pengguna yang sangat beragam. Topik-topik tersebut mencakup ranah domestik yang mendalam seperti dinamika keluarga *toxic*, masalah utang, dan judi online (Topik 9) serta konflik relasi interpersonal dan tekanan sosial (Topik 2). Di ranah publik, penelitian ini mengungkap peran komunitas sebagai sistem peringatan dini melalui identifikasi laporan penipuan online (Topik 7) serta pelaporan kriminalitas jalanan dan gangguan keamanan (Topik 3). Ketidakpuasan terhadap sistem juga

terlihat kuat pada topik layanan publik, kebijakan pemerintah, dan biaya pendidikan (Topik 1), serta beban akademik dan etika di ruang publik (Topik 8). Selain itu, penelitian mendeteksi pemicu frustrasi harian yang meliputi gangguan teknis finansial dan isu terkini (Topik 4), kritik terhadap standar sosial dan kelelahan interaksi (Topik 5), hingga kanal peluapan emosi personal terkait kondisi lingkungan dan cerita panjang melalui *thread* (Topik 6). Temuan ini menegaskan bahwa komunitas tersebut berfungsi sebagai ruang katarsis sekaligus sarana solidaritas digital dalam menghadapi tekanan sistemik maupun personal..

5.2 Saran

Berdasarkan Merujuk pada pembahasan mengenai evaluasi dan keterbatasan penelitian yang telah dipaparkan pada akhir Bab 4, berikut adalah rangkuman saran yang direkomendasikan untuk pengembangan penelitian selanjutnya:

a. Disarankan untuk menyusun dan menerapkan daftar *stopwords* khusus yang dikurasi secara manual guna mengeliminasi kata-kata partikel bahasa gaul (*slang*) yang tidak tertangkap oleh kamus standar.
b. Disarankan untuk mengombinasikan pemodelan topik dengan analisis sentimen atau emosi (*Emotion Analysis*) agar dapat mengukur intensitas kemarahan yang terkandung dalam setiap topik.
c. Disarankan untuk membandingkan kinerja LDA dengan algoritma pemodelan topik lain, seperti *Non-Negative Matrix Factorization* (NMF) atau BERTopic, untuk menguji konsistensi dan akurasi topik pada teks pendek.
d. Disarankan untuk memperluas rentang waktu pengambilan data guna menangkap tren topik yang bersifat musiman dan mendapatkan variasi isu yang lebih beragam.

DAFTAR PUSTAKA

Ali, T., Marc, B., Omar, B., Soulaimane, K., & Larbi, S. (2021). Exploring destination's negative e-reputation using aspect based sentiment analysis approach: Case of Marrakech destination on TripAdvisor. *Tourism Management Perspectives*, *40*. https://doi.org/10.1016/j.tmp.2021.100892

Anastasiu, D. C., & Tagarelli, A. (2017). Document Clustering. In *Wiley StatsRef: Statistics Reference Online* (pp. 1–11). Wiley. https://doi.org/10.1002/9781118445112.stat07973

Azhar, M. F., & Bakry, G. N. (2025). Pengaruh Penggunaan Twitter (X) terhadap Orientasi Pemilih Pemula pada Pemilihan Presiden 2024. *JSSH (Jurnal Sains Sosial Dan Humaniora)*, 101–112. https://doi.org/10.30595/jssh.v9i1.20385

Azzahrani, S., Lukmantoro, T., & Manalu, S. R. (2024). *Ekspresi Emosi Negatif Dalam Media Sosial (Studi Pada Komunitas "Marah-marah" di Twitter)*. https://ejournal3.undip.ac.id/index.php/interaksi-online/article/view/47483

Blair, S. J., Bi, Y., & Mulvenna, M. D. (2020). Aggregated topic models for increasing social media topic coherence. *Applied Intelligence*, *50*(1), 138–156. https://doi.org/10.1007/s10489-019-01438-z

Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, *55*(4), 77–84. https://doi.org/10.1145/2133806.2133826

Blei, D. M., Ng, A. Y., & Edu, J. B. (2003). Latent Dirichlet Allocation Michael I. Jordan. In *Journal of Machine Learning Research* (Vol. 3).

Çetin, V., & Yıldız, O. (2022). A comprehensive review on data preprocessing techniques in data analysis. *Pamukkale University Journal of Engineering Sciences*, *28*(2), 299–312. https://doi.org/10.5505/pajes.2021.62687

Culasso, F., Gavurova, B., Crocco, E., & Giacosa, E. (2023). Empirical identification of the chief digital officer role: A latent Dirichlet allocation approach. *Journal of Business Research*, *154*, 113301. https://doi.org/10.1016/j.jbusres.2022.113301

Dalimunthe, R. G., & Putri, R. A. (2024). Pemetaan Topik Tugas Akhir Program Studi Ilmu Komputer Menggunakan Algoritma Latent Dirichlet Allocation. *Journal of Computer System and Informatics (JoSYC)*, *5*(4), 921–932. https://doi.org/10.47065/josyc.v5i4.5759

Fatih, I. Z. A., Putera, R. A., & Umar, Z. H. (2024). Peran Algoritma Media Sosial dalam Penyebaran Propaganda Politik Digital Menjelang Pemilu. *Jurnal Kajian Stratejik Ketahanan Nasional*, *7*(1). https://doi.org/10.7454/jkskn.v7i1.10090

Gupta, A., & Katarya, R. (2021). PAN-LDA: A latent Dirichlet allocation based novel feature extraction model for COVID-19 data using machine learning. *Computers in Biology and Medicine*, *138*, 104920. https://doi.org/10.1016/j.compbiomed.2021.104920

Gurusamy, V., & Kannan, S. (2014). *Preprocessing Techniques for Text Mining*.

Han, J., Lee, S. E., & Cha, M. (2023). The secret to successful evocative messages: Anger takes the lead in information sharing over anxiety. *Communication Monographs*, *90*(4), 545–565. https://doi.org/10.1080/03637751.2023.2236183

Hassani, H., Beneki, C., Unger, S., Mazinani, M. T., & Yeganegi, M. R. (2020). Text Mining in Big Data Analytics. *Big Data and Cognitive Computing*, *4*(1), 1. https://doi.org/10.3390/bdcc4010001

Hastuti, Maulana, H. F., & Satria, E. (2025). Sentiment Analysis and Topic Modeling of Twitter Conversations in Indonesia's 2024 Presidential Election. *Jurnal Pekommas*, *10*(1). https://doi.org/10.56873/jpkm.v9i1.5545

Kamdan, Ivana Lucia Kharisma, Gina Purnama Insany, & Paikun. (2022). Research topic modeling in informatics engineering study program at Nusa Putra University using LDA method. *INTERNATIONAL JOURNAL ENGINEERING AND APPLIED TECHNOLOGY (IJEAT)*, *5*(2), 24–35. https://doi.org/10.52005/ijeat.v5i2.76

Khan, M., Sarwar, S., Khan, D., & Alharbi, Y. (2020). Text Mining Challenges and Applications, A Comprehensive Review. *IJCSNS International Journal of Computer Science and Network Security*, *20*(12). https://doi.org/10.22937/IJCSNS.2020.20.12.15

Kovalchuk, O., Banakh, S., Masonkova, M., Berezka, K., Mokhun, S., & Fedchyshyn, O. (2022). Text Mining for the Analysis of Legal Texts. *Proceedings - International Conference on Advanced Computer Information Technologies, ACIT*, 502–505. https://doi.org/10.1109/ACIT54803.2022.9913169

Lu, G., Leng, C., Li, B., Jiao, L., & Basu, A. (2023). Robust dual-graph discriminative NMF for data classification. *Knowledge-Based Systems*, *268*, 110465. https://doi.org/10.1016/j.knosys.2023.110465

Mifrah, S. (2020). Topic Modeling Coherence: A Comparative Study between LDA and NMF Models using COVID'19 Corpus. *International Journal of Advanced Trends in Computer Science and Engineering*, *9*(4), 5756–5761. https://doi.org/10.30534/ijatcse/2020/231942020

Mohammed, S. H., & Al-Augby, S. (2020). LSA & LDA Topic Modeling Classification: Comparison study on E-books. *Indonesian Journal of Electrical Engineering and Computer Science*, *19*(1). https://doi.org/10.11591/ijeecs.v19.i1.pp%25p

Muis, A., & Muhammad, F. (2023). *ABDIMAS LANGKANAE JURNAL PENGABDIAN KEPADA MASYARAKAT*. https://pusdig.web.id/index.php/abdimas/index

Negara, A. (2025). The Influence Of Applying Stopword Removal And Smote On Indonesian Sentiment Classification. *Lontar Komputer : Jurnal Ilmiah Teknologi Informasi*, *14*(03), 172–185. https://doi.org/10.24843/LKJITI.2023.v14.i03.p05

Nouvan. (2025, September 5). Negara dengan Pengguna X (Twitter) Terbanyak Juli 2025, Indonesia Urutan Keempat. *Dataloka.id*. https://dataloka.id/humaniora/4627/negara-dengan-pengguna-x-twitter-terbanyak-juli-2025-indonesia-urutan-keempat/#google_vignette

Nugumanova, A., Akhmed-Zaki, D., Mansurova, M., Baiburin, Y., & Maulit, A. (2022). NMF-based approach to automatic term extraction. *Expert Systems with Applications*, *199*, 117179. https://doi.org/10.1016/j.eswa.2022.117179

Pradhan, A., Senapati, M. R., & Sahu, P. K. (2022). Improving sentiment analysis with learning concepts from concept, patterns lexicons and negations. *Ain Shams Engineering Journal*, *13*(2), 101559. https://doi.org/10.1016/j.asej.2021.08.004

Rkia, A., Fatima-Azzahrae, A., Mehdi, A., & Lily, L. (2024). NLP and Topic Modeling with LDA, LSA, and NMF for Monitoring Psychosocial Well-being in Monthly Surveys. *Procedia Computer Science*, *251*, 398–405. https://doi.org/10.1016/j.procs.2024.11.126

Santoso, K. R. A. P., Husna, A., Putri, N. W., & Rakhmawati, N. A. (2022). Analisis Topik Tagar Covidindonesia pada Instagram Menggunakan Latent Dirichlet Allocation. *JISKA (Jurnal Informatika Sunan Kalijaga)*, *7*(1), 1–9. https://doi.org/10.14421/jiska.2022.7.1.1-9

Sarifudin, B., & Zuhri, S. (2025). TRASH TALKING DALAM KOMUNITAS VIRTUAL: STUDI KASUS PADA KOMUNITAS MARAH-MARAH PLATFORM X. *Jurnal Ilmu Komunikasi UHO : Jurnal Penelitian Kajian Ilmu Komunikasi Dan Informasi*, *10*(1), 188–206. https://doi.org/10.52423/jikuho.v10i1.1496

Shamshiri, A., Ryu, K. R., & Park, J. Y. (2024). Text mining and natural language processing in construction. *Automation in Construction*, *158*, 105200. https://doi.org/10.1016/j.autcon.2023.105200

Shen, C., & Ho, J. (2020). Technology-enhanced learning in higher education: A bibliometric analysis with latent semantic approach. *Computers in Human Behavior*, *104*, 106177. https://doi.org/10.1016/j.chb.2019.106177

Silaparasetty, N. (2020). *Machine Learning Concepts with Python and the Jupyter Notebook Environment*. Apress. https://doi.org/10.1007/978-1-4842-5967-2

Singh, D., & Singh, B. (2020). Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, *97*, 105524. https://doi.org/10.1016/j.asoc.2019.105524

Siswandi, A., Permana, Y., & Emarilis, A. (2021). Stemming Analysis Indonesian Language News Text with Porter Algorithm. *Journal of Physics: Conference Series*, *1845*(1), 012019. https://doi.org/10.1088/1742-6596/1845/1/012019

Sundaram, V., Ahmed, S., Muqtadeer, S. A., & Ravinder Reddy, R. (2021). Emotion Analysis in Text using TF-IDF. *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 292–297. https://doi.org/10.1109/Confluence51648.2021.9377159

Wang, J., & Zhang, X. (2023). Deep NMF topic modeling. *Neurocomputing*, *515*, 157–173. https://doi.org/10.1016/j.neucom.2022.10.002

Wang, Y., Tong, Y., & Shi, D. (2020). Federated Latent Dirichlet Allocation: A Local Differential Privacy Based Framework. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(04), 6283–6290. https://doi.org/10.1609/aaai.v34i04.6096

Yang, L., Yan, L., Wei, X., & Yang, X. (2023). Label consistency-based deep semisupervised NMF for tumor recognition. *Engineering Applications of Artificial Intelligence*, *117*, 105511. https://doi.org/10.1016/j.engappai.2022.105511

Yijia, L. (2024). *Comparison of LDA and BERTopic in News Topic Modeling: A Case Study of The New York Times' Reports on China*. *7*.

Zoya, Latif, S., Shafait, F., & Latif, R. (2021). Analyzing LDA and NMF Topic Models for Urdu Tweets via Automatic Labeling. *IEEE Access*, *9*, 127531–127547. https://doi.org/10.1109/ACCESS.2021.3112620

LAMPIRAN

Lampiran A – Dataset Penelitian
https://drive.google.com/file/d/100LzRmWaOO1KBfFl3LOSJKZ9L1p_bQXA/view?usp=sharing

Lampiran B – Kamus *Normalization*
https://drive.google.com/file/d/1vF3rKR-Y6oUrL7WzoCl37a0Wi72mbZVW/view?usp=sharing

Lampiran C – *Tweet Representative*
https://docs.google.com/spreadsheets/d/1LJIhLjKfH76CT6HmHrkps5aZeG2AYxL9/edit?usp=sharing&ouid=102063973741579788062&rtpof=true&sd=true