



***Sentiment Analysis dan Topic Modeling pada Tweet Bahasa
Indonesia Mengenai Pemilihan Umum 2024***

Alfan Ramadhan Sembiring

22917024

Tesis diajukan sebagai syarat untuk meraih gelar Magister Komputer

Konsentrasi Sains Data

Program Studi Teknik Informatika Program Magister

Fakultas Teknologi Industri

Universitas Islam Indonesia

2025

Lembar Pengesahan Pembimbing

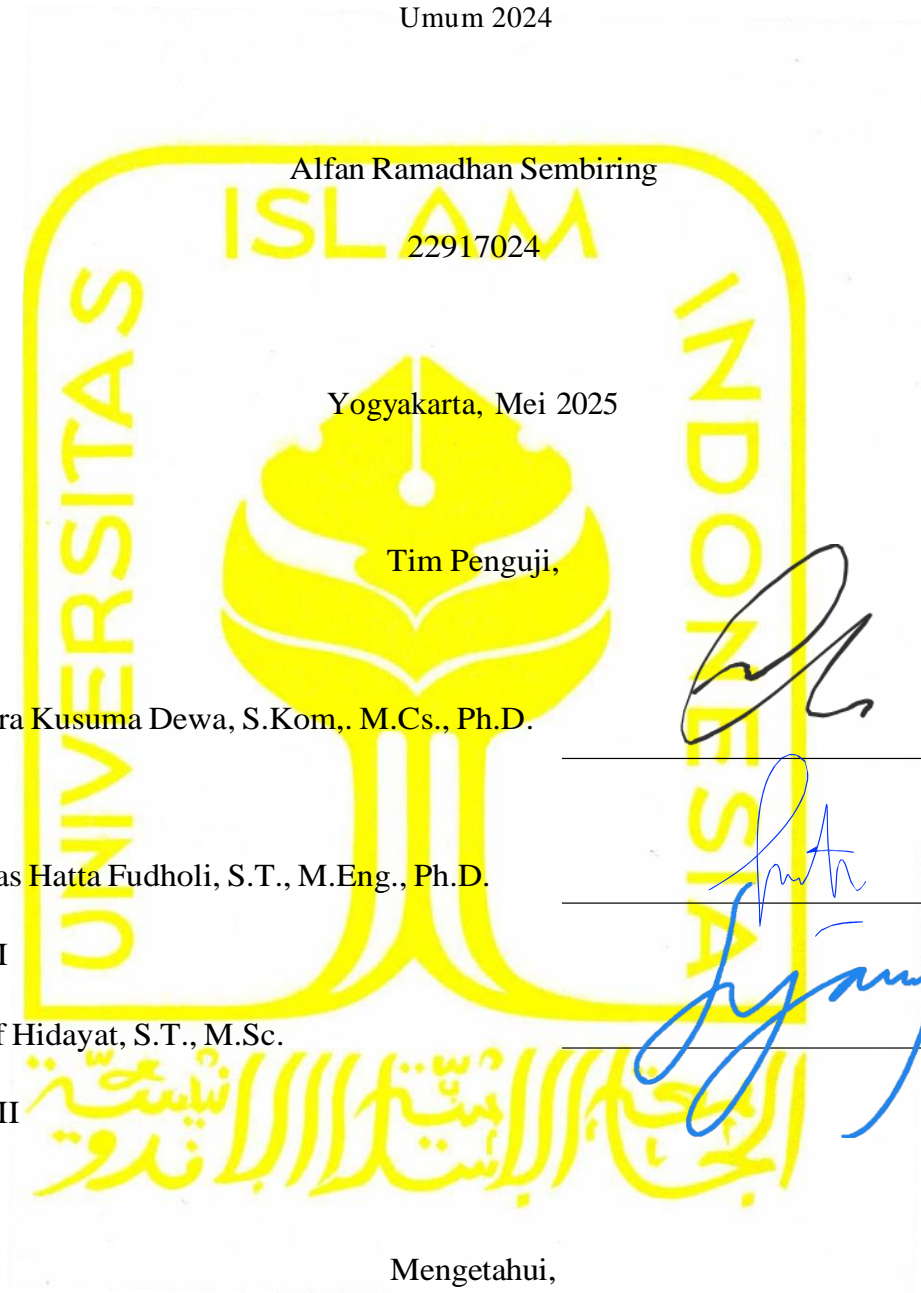
Sentiment Analysis dan Topic Modeling pada Tweet Bahasa Indonesia Mengenai
Pemilihan Umum 2024



Ir. Chandra Kusuma Dewa, S.Kom., M.Sc., Ph.D.

Lembar Pengesahan Penguji

Sentiment Analysis dan Topic Modeling pada Tweet Bahasa Indonesia Mengenai Pemilihan Umum 2024



Mengetahui,

Ketua Program Studi Informatika Program Magister

Universitas Islam Indonesia

Ir. Irving Vitra Papatungan, S.T., M.Sc., Ph.D.

Abstrak

Sentiment Analysis dan Topic Modeling pada Tweet Bahasa Indonesia Mengenai Pemilihan Umum 2024

Media sosial memainkan peran penting dalam menyebarkan informasi dan opini publik, terutama dalam konteks Pemilihan Umum (Pemilu) 2024 di Indonesia. Analisis sentimen dan pemodelan topik menjadi metode yang efektif untuk memahami persepsi masyarakat terhadap pemilu berdasarkan data dari media sosial X (sebelumnya Twitter). Penelitian ini bertujuan untuk mengembangkan model analisis sentimen yang mampu mengklasifikasikan sentimen positif, netral, dan negatif pada tweet berbahasa Indonesia serta mengidentifikasi topik utama yang muncul dalam diskusi publik mengenai pemilu.

Metode yang digunakan dalam analisis sentimen mencakup pendekatan *machine learning* dan *deep learning*, yaitu Naïve Bayes, *Support Vector Machine* (SVM), *Long Short-Term Memory* (LSTM), dan *Gated Recurrent Unit* (GRU). Selain itu, pemodelan topik dilakukan menggunakan *Latent Dirichlet Allocation* (LDA) untuk menemukan pola tematik dalam diskusi di media sosial. Data yang digunakan dikumpulkan dalam rentang waktu Januari hingga April 2024, dengan total 4500 tweet.

Hasil penelitian menunjukkan bahwa metode Naïve Bayes memiliki kinerja terbaik dalam klasifikasi sentimen dengan akurasi 64%, diikuti oleh SVM dengan 63%, sementara metode *deep learning* seperti LSTM dan GRU menunjukkan performa yang lebih rendah. Pemodelan topik mengidentifikasi delapan topik utama yang mencerminkan berbagai aspek perbincangan publik tentang pemilu, termasuk dukungan terhadap calon, keamanan pemilu, peran KPU dan Bawaslu, serta isu-isu terkait kecurangan.

Dengan pendekatan metodologis yang sistematis, penelitian ini memberikan wawasan mengenai opini publik terhadap Pemilu 2024 dan dapat berkontribusi dalam perancangan strategi komunikasi serta kebijakan publik berbasis data.

Kata kunci

sentiment analysis, SVM, GRU, topic modelling, LDA, LSTM, naïve bayes, pemilu.

Pernyataan Keaslian Tulisan

Dengan ini saya menyatakan bahwa tesis ini merupakan tulisan asli dari penulis, dan tidak berisi material yang telah diterbitkan sebelumnya atau dari penulis lain terkecuali referensi atas material tersebut telah disebutkan dalam tesis. Apabila ada kontribusi dari penulis lain dalam tesis ini, maka penulis lain tersebut secara eksplisit telah disebutkan dalam tesis ini.

Dengan ini saya juga menyatakan bahwa segala kontribusi dari pihak lain terhadap tesis ini, termasuk bantuan analisis statistic, desain survey, analisis data, prosedur teknis yang bersifat signifikan, dan segala bentuk aktivitas penelitian yang digunakan atau dilaporkan dalam tesis ini telah secara eksplisit disebutkan dalam tesis ini.

Segala bentuk hak cipta yang terdapat dalam material dokumen tesis ini berada dalam kepemilikan pemilik hak cipta masing-masing. Apabila dibutuhkan, penulis juga telah mendapatkan izi dari pemilik hak cipta untuk menggunakan ulang materialnya dalam tesis ini.

Yogyakarta, 23 Mei 2025



Alfan Ramadhan Sembiring

Daftar Isi

Lembar Pengesahan Pembimbing	i
Lembar Pengesahan Penguji.....	ii
Abstrak	3
Pernyataan Keaslian Tulisan.....	4
Daftar Isi	5
Daftar Gambar	7
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	4
1.6 Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA	6
2.1 Landasan Teori.....	6
2.1.1 <i>Sentiment Analysis</i>	6
2.1.2 <i>Naïve Bayes</i>	6
2.1.3 <i>Support Vector Machine</i>	7
2.1.4 <i>Recurrent Neural Network</i>	8
2.1.5 <i>Long Short Term Memory</i>	8
2.1.6 <i>Gated Recurrent Unit</i>	9
2.1.7 <i>Topic Modeling</i>.....	10
2.1.8 <i>Latent Dirichlet Allocation</i>.....	10
2.1.3 <i>Hyperparameter Tuning</i>	11
2.1.4 <i>Evaluasi Model Sentiment Analysis</i>	12

2.1.5 Evaluasi Model <i>Topic Modeling</i>	14
2.2 Kajian Pustaka	14
BAB III METODOLOGI PENELITIAN	18
3.1 Desain Penelitian	18
3.2 Pengumpulan Data	19
3.3 <i>Preprocessing Data</i>	20
3.4 Pemodelan Naïve Bayes	23
3.5 Pemodelan <i>Support Vector Machine</i>	23
3.6 Hyperparameter Model <i>Deep Learning</i>	23
3.7 Pemodelan LSTM.....	24
3.8 Pemodelan GRU	24
3.9 Pemodelan LDA	25
BAB 4 HASIL DAN PEMBAHASAN	26
4.1 Crawling Data	26
4.2 <i>Preprocessing</i>	30
4.3 Naïve Bayes.....	32
4.4 <i>Support Vector Machine (SVM)</i>	34
4.5 <i>Long Short-Term Memory</i>	35
4.6 <i>Gated Recurrent Unit (GRU)</i>	36
4.7 <i>Topic Modelling</i>	38
4.5 Pembahasan.....	49
BAB 5 KESIMPULAN DAN SARAN	56
5.1 Kesimpulan.....	56
5.2 Saran.....	57

Daftar Gambar

Gambar 3. 1 Desain Penelitian	19
Gambar 3. 2 Alur <i>Preprocessing</i>	21
Gambar 4. 1 <i>Home Page X</i>	26
Gambar 4. 2 <i>Inspect Element</i>	27
Gambar 4. 3 <i>Inspect Element Application</i>	27
Gambar 4. 4 <i>Cookies URL X</i> dengan <i>auth_token</i>	28
Gambar 4. 5 Instalasi <i>library</i> <i>pandas</i> dan <i>Node.js</i>	29
Gambar 4. 6 Parameter yang digunakan dalam <i>tweet harvest</i>	29
Gambar 4. 7 Proses <i>Crawling Tweet</i>	29
Gambar 4. 8 Menampilkan <i>DataFrame</i> file <i>CSV</i> dari hasil <i>crawling</i>	30
Gambar 4. 9 Mengubah label <i>sentiment</i> menjadi numerik	32
Gambar 4. 10 <i>CountVectorizer</i> dan <i>TF-IDF</i>	33
Gambar 4. 11 <i>Hyperparameter SVM</i>	34
Gambar 4. 12 <i>Dense Layer</i> dengan aktivasi <i>softmax</i>	35
Gambar 4. 13 <i>Hyperparameter LSTM</i>	35
Gambar 4. 14 <i>Learning rate Adam</i>	37
Gambar 4. 15 <i>Early Stopping</i>	37
Gambar 4. 16 Pencarian Jumlah Optimal Topik	38
Gambar 4. 17 Visualisasi Umum <i>Topic Modeling</i> dengan <i>pyLDAvis</i>	39
Gambar 4. 18 Visualisasi Topik 1	40
Gambar 4. 19 Visualisasi Topik 2	41
Gambar 4. 20 Visualisasi Topik 3	42
Gambar 4. 21 Visualisasi Topik 4	43
Gambar 4. 22 Visualisasi Topik 5	44
Gambar 4. 23 Visualisasi Topik 6	45
Gambar 4. 24 Visualisasi Topik 7	46
Gambar 4. 25 Visualisasi Topik 8	47
Gambar 4. 26 Distribusi <i>Sentiment</i> per Topik	48
Gambar 4. 27 <i>Pie Chart</i> Distribusi <i>Sentiment</i>	49
Gambar 4. 28 <i>Bar Chart</i> Perbandingan Model <i>Sentiment Analysis</i>	50
Gambar 4. 29 Grafik <i>Loss</i> dan <i>Accuracy</i> untuk model <i>LSTM</i>	51
Gambar 4. 30 Grafik <i>Loss</i> dan <i>Accuracy</i> untuk model <i>GRU</i>	52

Daftar Tabel

Tabel 2. 1 <i>Confusion Matrix</i>	12
Tabel 2. 2 Daftar <i>Literature Review</i>	16
Tabel 3. 1 Contoh Kata Kamus Alay	22
Tabel 3. 2 Konfigurasi Model LSTM dan GRU	23
Tabel 4. 1 Hasil <i>Crawling Data</i>	30
Tabel 4. 2 Hasil <i>Preprocessing</i>	31
Tabel 4. 3 Hasil pengujian Naïve Bayes	33
Tabel 4. 4 Hasil pengujian SVM	34
Tabel 4. 5 Hasil pengujian LSTM	36
Tabel 4. 6 Hasil Pengujian GRU	37
Tabel 4. 7 <i>Coherence Score</i>	38
Tabel 4. 8 Tabel Perbanding Model <i>Sentiment Analysis</i>	49
Tabel 4. 9 <i>Accuracy Rates per Epoch</i>	52
Tabel 4. 10 Hasil <i>Topic Modeling</i>	53

BAB I

PENDAHULUAN

1.1 Latar Belakang

Media sosial saat ini telah menjadi bagian yang tidak terpisahkan dari rutinitas sehari-hari. Tweet sering kali hanya menyertakan beberapa kata yang memiliki makna praktis, dan kata-kata ini sangat penting dalam tahap klasifikasi (Talaat, 2023). Pemilihan Umum di tahun 2024 merupakan bagian yang sangat penting bagi demokrasi Indonesia, terutama ketika memasuki masa kampanye dimana banyak masyarakat yang mengekspresikan diri terhadap Pemilihan Umum di Media Sosial seperti X atau yang dikenal dulu sebagai X. Media sosial telah berkembang secara fenomenal selama satu dekade terakhir (Kar & Debbarma, 2023). Misalnya, X memiliki 321 juta pengguna aktif. Pada aplikasi media sosial ini, orang kecanduan untuk mengekspresikan pendapat, saran, suka, tidak suka, dan pengalaman mereka dengan cara yang sangat kasual (Abid et al., 2020).

Meskipun banyak platform media sosial yang digunakan untuk diskusi online dan pertukaran informasi, X masih memegang posisi penting dalam hal penyebaran data teks (Polyzos, Efstathios & Wang, 2022). Seiring dengan evolusi jaringan media sosial, banyaknya volume teks media sosial yang tersedia untuk analisis *sentiment* telah meningkat beberapa kali lipat, yang mengarah ke korpus yang tangguh (Iddrisu et al., 2023).

Salah satu hal yang sangat penting dalam menganalisis pendapat publik adalah memeriksa perasaan mereka. Di sini, konsep *sentiment analysis* ikut berperan (Pandey & Vishwakarma, 2023). *Sentiment Analysis* adalah cabang dari *Natural Language Processing* (NLP) yang berfokus pada analisis pandangan, perasaan, dan emosi orang (Talaat, 2023). Model *Sentiment Analysis* memainkan peran penting. Banyak sektor, termasuk tantangan politik, pemasaran, kebijakan publik, manajemen bencana, dan kesehatan masyarakat, bergantung pada deteksi emosi (Talaat, 2023). Namun, relevansi dan aplikasinya dalam konteks Pemilihan Umum di Indonesia masih relatif sedikit dieksplorasi.

Salah satu metode *Deep Learning* yang banyak digunakan untuk melakukan *Sentiment Analysis* adalah metode RNN. Metode berbasis CNN semakin populer seiring dengan perkembangan bidang *deep learning*. Namun demikian, sebagian besar teknik CNN

saat ini berfokus pada pekerjaan tertentu (Alruily, 2023). Secara umum, *Recurrent Neural Network* (RNN), *Long Short Term Memory* (LSTM), *Gated Recurrent Unit* (GRU), *Convolutional Neural Network* (CNN) dan jaringan saraf rekursif sering digunakan untuk mengkodekan urutan kata untuk menentukan sentimen polaritas dari aspek, kalimat, atau dokumen tertentu (Liu & Shen, 2020).

Banyaknya antusias rakyat dalam mengekspresikan diri di media sosial, mengakibatkan X menjadi salah satu tujuan rakyat untuk mencari tahu tentang pendapat pengguna lain terhadap Pemilihan Umum 2024. Dari banyaknya tweet yang tersedia secara umum dapat dijadikan sumber data untuk keperluan penelitian dan meng-klasifikasi topik apa yang banyak disampaikan di X. Pemodelan topik dan analisis sentimen di social media analysis digunakan untuk mengidentifikasi masalah yang dijelaskan dalam konten buatan pengguna, ide dan peluang, dan diskusi penting (Mulyani et al., 2024) dengan cepat, efektif, dan murah, serta mengekstrapolasi tema umum di seluruh dokumen (Melton et al., 2021). Pemodelan topik menggunakan *Latent Dirichlet Allocation* (LDA) membutuhkan tiga input penting yaitu, korpus, kamus kata, dan jumlah topik yang ditentukan (Mulyani et al., 2024).

Dalam penelitian sebelumnya, algoritma *machine learning* umumnya digunakan, sementara beberapa penelitian hanya berfokus pada satu atau dua algoritma *deep learning*. Namun, penelitian ini menggabungkan dua algoritma *machine learning*, dua algoritma *deep learning*, dan *topic modeling* menggunakan LDA pada dataset tweet terkait pemilu 2024. Model *machine learning* dapat memberikan performa yang baik dengan dataset yang lebih kecil, sedangkan model *deep learning* biasanya memerlukan data dalam jumlah besar untuk mencapai akurasi yang tinggi. Hal ini menjadikan *machine learning* pilihan yang lebih praktis ketika data terbatas (Varghese et al., 2020). Model *deep learning* tipe RNN dirancang untuk menangani data *sequential* dan efektif dalam memodelkan dependensi jangka panjang, yang sangat bermanfaat untuk tugas-tugas yang melibatkan informasi berurutan (Zulqarnain et al., 2020). Pendekatan ini meningkatkan kedalaman analisis dengan menangkap pola terstruktur dan laten dalam data, sehingga memberikan kontribusi yang lebih komprehensif dibandingkan dengan penelitian sebelumnya.

Penggunaan metode *machine learning*, pada penelitian ini digunakan algoritma Naïve Bayes dan SVM, cocok digunakan pada karakteristik dataset yang berukuran dataset kecil hingga sedang, teks pendek atau memiliki struktur sederhana, seperti ulasan produk, tweet, atau komentar forum. Naïve Bayes sangat efektif untuk data kategorikal, seperti yang terlihat dalam studi tentang autentikasi pengguna di mana algoritma ini

digunakan untuk mengklasifikasikan upaya login ke dalam berbagai kategori (Mavhemwa et al., 2024). Meskipun dataset dalam penelitian ini seimbang, SVM dapat menangani data yang tidak seimbang secara efektif dengan menyesuaikan bobot kelas, yang bermanfaat dalam skenario di mana distribusi kelas tidak merata (Gunawan et al., 2024).

Penggunaan *deep learning* pada penelitian ini dengan metode LSTM dan GRU efektif untuk dataset besar karena kemampuannya dalam menangkap dependensi jangka panjang (Yang et al., 2020). Model GRU, meskipun dalam beberapa kasus sedikit kurang akurat dibandingkan LSTM, menawarkan rasio *performance-cost* yang lebih baik, sehingga cocok untuk skenario di mana sumber daya komputasi terbatas atau ketika diperlukan pelatihan model yang lebih cepat (Gao et al., 2020).

1.2 Rumusan Masalah

Berdasarkan latar belakang yang dipaparkan sebelumnya, rumusan masalah penelitian ini adalah bagaimana membangun dan mengevaluasi model analisis sentimen serta pemodelan topik pada Tweet berbahasa Indonesia terkait Pemilihan Umum 2024 dengan menggunakan pendekatan machine learning dan deep learning? :

1.3 Batasan Masalah

Pembatasan-pembatasan di dalam melakukan penelitian ini yaitu:

1. Dataset yang digunakan hanya berfokus terhadap Tweet berbahasa Indonesia seputaran Pemilihan Umum
2. Data yang dikumpulkan adalah Tweet yang diposting dalam rentang waktu tertentu, misalnya dari Januari 2024 hingga April 2024, untuk fokus pada periode yang terkait erat dengan Pemilihan Umum 2024.
3. Metode klasifikasi pada penelitian ini menggunakan metode Naïve Bayes, SVM dan RNN (*Recurrent Neural Networks*) dan type RNN yang penulis pakai adalah LSTM (*Long Short-Term Memory*) dan GRU (*Gated Recurrent Unit*) dengan hyperparameter yang belum ditentukan
4. Metode *Topic Modelling* yang digunakan adalah *Latent Dirichlet Allocation* (LDA)

1.4 Tujuan Penelitian

Penelitian ini dilakukan dengan tujuan:

1. Mengembangkan dan mengevaluasi model analisis sentimen terhadap Tweet berbahasa Indonesia tentang Pemilihan Umum 2024 menggunakan metode machine learning dan deep learning.
2. Mengidentifikasi topik utama dalam diskusi publik terkait Pemilu 2024 melalui pendekatan topic modeling dengan metode LDA

1.5 Manfaat Penelitian

Manfaat-manfaat yang diperoleh dari hasil penelitian ini dapat dilihat dari beberapa sudut pandang, yaitu bagi penulis, akademis, dan praktis.

- a) Bagi penulis sendiri, dapat menambah wawasan di bidang *Machine Learning*.
- b) Menambah literatur dan referensi ilmiah mengenai penerapan model *Machine Learning* dan *Topic Modeling* dalam analisis sentimen media sosial.
- c) Membantu pengambil keputusan, seperti pemerintah, dalam memahami sentimen publik terhadap isu tertentu, memudahkan dalam strategi komunikasi atau kampanye.

1.6 Sistematika Penulisan

Gambaran umum sistematika penulisan pada tesis ini terdiri dari lima bab yaitu:

BAB I Pendahuluan

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

BAB II Tinjauan Pustaka

Bab ini berisi tinjauan pustaka terkait landasan teori pendukung untuk menjawab tujuan penelitian. Teori-teori terkait *Sentiment Analysis*, *Topic Modeling*, algoritma Naïve Bayes, SVM, LSTM, GRU, dan algoritma LDA yang relevan.

BAB III Metodologi Penelitian

Bab ini menguraikan tahapan-tahapan penelitian serta proses perancangan model yang menjadi fokus utama dalam pemecahan masalah yang dikaji.

BAB IV Hasil dan Pembahasan

Bab ini menjelaskan prosedur pengumpulan data hingga analisis hasil yang dicapai oleh model machine learning dan *deep learning* yang dikembangkan.

BAB V Kesimpulan dan Saran

Bab ini berisi kesimpulan dan saran dari penelitian yang dilakukan.

BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

2.1.1 *Sentiment Analysis*

Akar dari analisis sentimen dapat ditelusuri kembali ke studi opini publik awal abad ke-20 dan linguistik komputasi pada tahun 1990-an. Bidang ini telah berkembang pesat sejak tahun 2004, didorong oleh ketersediaan teks subjektif di web. Fokusnya telah bergeser dari ulasan produk ke aplikasi yang lebih luas, termasuk analisis media sosial dan berbagai sektor industri (Mäntylä et al., 2016). *Sentiment Analysis* adalah proses multi-langkah yang mencakup pengambilan data, ekstraksi, pre-processing, dan ekstraksi fitur (Talaat, 2023). Sentiment Analysis telah diterapkan di berbagai bidang, termasuk prediksi pasar keuangan, kesehatan, analisis pelanggan, pemasaran merek, politik, dan manajemen keadaan darurat. Analisis ini sangat lazim digunakan untuk menganalisis platform media sosial seperti Twitter, di mana beragam opini diungkapkan setiap hari (Birjali et al., 2021). Sentiment analysis menggunakan berbagai metode, mulai dari teknik tradisional seperti kamus dan *Support Vector Machines* hingga jaringan saraf tingkat lanjut dan model pembelajaran mendalam seperti LSTM dan CNN. Metode-metode ini bertujuan untuk mengklasifikasikan teks sebagai pengekspresian sentimen positif atau negatif, dengan kemajuan terbaru yang berfokus pada model yang lebih canggih seperti Transformers (seperti, BERT, GPT-3) (Yue et al., 2018).

2.1.2 Naïve Bayes

Naïve Bayes baru-baru ini telah banyak digunakan dalam teknik klasifikasi, terutama di jaringan media sosial seperti X, dengan menggunakan beberapa metode, antara lain Unigram Naïve Bayes, Multinomial Naïve Bayes dan Maximum Klasifikasi Entropi (Samsir et al., 2021). Naive Bayes secara komputasi ringan, sehingga cocok untuk memproses dataset besar seperti postingan media sosial, ulasan, atau komentar. Ini membutuhkan waktu pelatihan minimal dan mudah diimplementasikan untuk aplikasi dunia nyata (Triyono et al., 2025). Perhitungan kategori probabilitas dalam Naïve Bayes didasarkan pada prinsip dasar Teorema Bayes. Algoritma ini menghitung kemungkinan suatu data termasuk dalam kategori tertentu dengan menggunakan rumus berikut:

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)} \tag{2.1}$$

Penjelasan:

$P(C|X)$: Probabilitas Posterior

$P(X|C)$: *Likelihood*

$P(C)$: Probabilitas Prior

$P(X)$: *Evidence* (Normalisasi)

2.1.3 Support Vector Machine

Klasifikasi dalam *Support Vector Machines* (SVM) didasarkan pada prinsip menemukan hiperbidang optimal yang secara maksimal memisahkan titik data dari kategori yang berbeda. Studi menunjukkan bahwa SVM dapat mencapai tingkat akurasi yang tinggi (misalnya, 91,8% pada klasifikasi sentimen tweet), sehingga dapat diandalkan untuk aplikasi di dunia nyata (Q. Huang, 2023). Aplikasi-aplikasi ini menunjukkan keserbagunaan dan keefektifan SVM dalam menangani set data yang besar dan mengekstraksi wawasan yang bermakna dari konten buatan pengguna (Hasanati et al., 2022). Algoritma ini menghitung batas keputusan menggunakan rumus berikut:

$$f(x) = \text{sgn} \left(- \sum_{n=1}^N \alpha_i - y_i K(X_n X) + b \right) \quad (2.2)$$

Penjelasan:

$F(x)$: Fungsi keputusan SVM (output prediksi: +1 atau -1)

$\text{sgn}()$: Fungsi tanda (signum) yang menentukan kelas

α_i : Lagrange multiplier (bobot) untuk support vector ke-i

y_i : Label kelas (+1 atau -1) dari data training

$K(X_n, X)$: Fungsi kernel (mengukur kesamaan antara support vector X_n dan data X)

b : Bias term (parameter intercept)

N : Jumlah support vectors

2.1.4 Recurrent Neural Network

Natural Language Processing (NLP) telah sangat diuntungkan dari kebangkitan *Deep Neural Networks* (DNN), karena kinerjanya yang tinggi dengan lebih sedikit kebutuhan akan fitur yang direkayasa. Ada dua arsitektur DNN utama: *Convolutional Neural Network* (CNN) dan *Recurrent Neural Network* (RNN) (Yin et al., 2017). *Recurrent neural network* (RNN) telah banyak digunakan dalam masalah *sequence learning* seperti pengenalan tindakan, pelabelan adegan dan pemrosesan bahasa, dan telah mencapai hasil yang mengesankan (Li et al., 2018). Namun, RNN sering mengalami masalah klasik “*Vanishing*” dan “*Exploding*” gradient, yang mengakibatkan kegagalan untuk mempelajari ketergantungan jarak jauh (Rana, 2016). RNN, LSTM, dan GRU sederhana biasanya digunakan untuk analisis sentimen. LSTM dan GRU sangat disukai karena kemampuannya menangani ketergantungan jangka panjang dan mengurangi masalah seperti gradien yang menghilang, yang lazim terjadi pada RNN sederhana (Hindarto, 2023).

2.1.5 Long Short Term Memory

Long Short-Term Memory (LSTM) efektif tidak hanya untuk klasifikasi sentimen secara keseluruhan, tetapi juga untuk tugas-tugas yang lebih bernuansa seperti analisis sentimen berbasis aspek atau dimensi, di mana pemahaman sentimen terhadap target tertentu atau di berbagai dimensi diperlukan (Yukun et al., 2018). *Long Short-Term Memory* (LSTM) dirancang untuk mengatasi masalah *vanishing* gradien dan sejak itu menjadi salah satu arsitektur RNN yang paling banyak digunakan (DiPietro & Hager, 2019). Persamaan untuk Forget Gate dapat dilihat pada persamaan berikut.

$$f_t = \sigma(W_f[h_{t-1}, X_t] + b_f) \quad (2.3)$$

Input Gate digunakan untuk menentukan pentingnya informasi baru. Persamaan untuk *Input Gate* dapat dilihat pada persamaan berikut.

$$i_t = \sigma(X_t * U_i + H_{t-1} * W_i) \quad (2.4)$$

Output Gate mengontrol status sel yang digunakan untuk menghitung status tersembunyi saat ini. Persamaan untuk *Output Gate* cukup mirip dengan dua *gate* sebelumnya.

$$o_t = \sigma(X_t * U_o + H_{t-1} * W_o) \quad (2.5)$$

Cell state dalam LSTM berfungsi sebagai memori jaringan, menjaga informasi jangka panjang sepanjang langkah waktu, sementara status tersembunyi mewakili output dari LSTM pada langkah waktu tertentu, mencakup informasi baik saat ini maupun

sebelumnya untuk ketergantungan jangka pendek. Persamaan untuk status sel dan status tersembunyi dapat dilihat pada persamaan 2.6 dan 2.7.

$$C_t = f_t * C_{t-1} + i_t \cdot C_t \quad (2.6)$$

$$h_t = o_t \cdot \tanh(C_t) \quad (2.7)$$

Penjelasan:

i_t : Input gate

f_t : Forget gate

o_t : Output gate

h_t : Hidden state (output jangka pendek)

x_t : Input pada waktu ke- t

h_{t-1} : Hidden state sebelumnya

W : Matriks bobot

b : Bias

2.1.6 Gated Recurrent Unit

Gated Recurrent Unit (GRU) efektif dalam menangani bahasa yang kompleks dan bernuansa, seperti sarkasme, ironi, dan sentimen berbasis aspek, dengan mempertahankan konteks pada rangkaian yang panjang dan berfokus pada kata-kata penting melalui mekanisme *attention* (Zulqarnain et al., 2022). *Gated Recurrent Unit* (GRU) adalah variasi LSTM yang sedikit lebih disederhanakan. Ini menggabungkan *Forget gate* dan *input gates* menjadi satu "*update gate*" dan memiliki "*reset gate*" tambahan. Model akhir lebih sederhana daripada model LSTM standar dan menjadi semakin populer (Rana, 2016). Persamaan untuk *update gate* (z_t) dapat dilihat pada persamaan 2.8

$$z_t^j = \sigma(W_z x_t + u_z h_t - 1)^j \quad (2.8)$$

Update_t gate (z^j) mengontrol seberapa banyak informasi dari aktivasi unit sebelumnya (h_{t-1}) yang akan dipertahankan atau diperbarui dengan informasi baru dari input saat ini (x_t). Nilai z^j berkisar antara 0 dan 1, di mana 0 berarti "lupakan sepenuhnya" dan 1 berarti "simpan sepenuhnya". Fungsi aktivasi sigmoid (σ) digunakan untuk menghasilkan nilai dalam rentang ini.

Reset Gate menentukan seberapa banyak informasi dari status tersembunyi sebelumnya yang harus dibuang. Mirip dengan *update gate*, gerbang *reset* juga menggunakan fungsi aktivasi sigmoid untuk menghasilkan nilai antara 0 dan 1. Persamaan untuk *reset gate*

(r^j) dapat dilihat pada persamaan 2.9

$$r_t^j = \sigma(W_r x_t + r h_t - 1)^j \quad (2.9)$$

Candidate activation adalah representasi potensial dari *hidden state* baru. Fungsi aktivasi hyperbolic tangent (tanh) digunakan untuk menghasilkan nilai antara -1 dan 1. Persamaan untuk *candidate activation* dapat dilihat pada persamaan 2.10

$$h_t^{-j} = \tanh(W x_t + U(r_t \cdot h_t - 1)^j) \quad (2.10)$$

Hidden State adalah kombinasi tertimbang antara *hidden state* sebelumnya (h_{t-1}^j) dan *candidate activation* (h^{-j}). Pembobotan dikendalikan oleh *update gate* (z_t^j). Persamaan untuk *hidden state* tertera pada persamaan 2.11

$$h_t^j = (1 - z_t^j)h_{t-1}^j + z_t^j h_t^{-j} \quad (2.11)$$

Penjelasan:

r_t : Reset Gate

z_t : Update Gate

h_t :Hidden state output

σ :Fungsi sigmoid

W_r : Bobot untuk reset gate

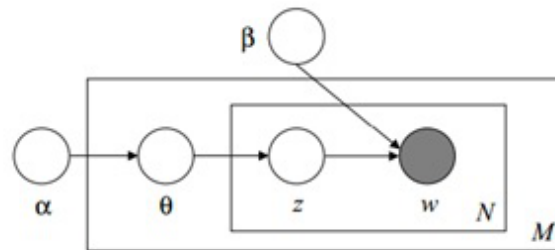
2.1.7 Topic Modeling

Topic Modeling adalah metode menjelajahi koleksi dokumen yang besar. Metode ini dapat membantu secara otomatis menemukan struktur tersembunyi dalam berbagai tugas tinjauan literatur seperti mengidentifikasi topik, mendeteksi tren yang muncul, dan merekomendasikan jurnal yang serupa (D'Ascenzo et al., 2024). Hal ini terutama sangat cocok untuk dengan data teks namun, juga telah digunakan untuk menganalisis data bioinformatika, data sosial, dan data lingkungan (Vayansky & Kumar, 2020). Fase *pre-processing* untuk *topic modelling* analisis dilakukan dalam 2 fase, pertama dilakukan *stemming* dan *lemmatizing* katanya (D'Ascenzo et al., 2024).

2.1.8 Latent Dirichlet Allocation

Dalam penelitian ini, penulis menggunakan model *Latent Dirichlet Allocation* (LDA) yang diperkenalkan oleh David M. Blei pada tahun 2003 untuk menganalisis topik-topik dalam data. LDA merupakan model probabilitas generatif yang digunakan untuk mengidentifikasi topik-topik laten dalam suatu kumpulan teks atau corpus. Model LDA (*Latent Dirichlet*

Allocation) secara efektif menemukan topik laten dalam korpora teks, dengan kriteria evaluasi seperti perplexity, jarak VI, visualisasi hasil klustering, dan kata-kata dengan probabilitas tertinggi (Blei et al., 2002).



Gambar 2. 1 Representasi Grafis LDA (Kandukuri,2020)

Penjelasan:

1. α (alpha) : Hyperparameter yang mengontrol distribusi topik dalam dokumen
2. θ (theta) : Distribusi topik untuk suatu dokumen tertentu (berasal dari α)
3. z : Variabel laten yang mewakili topik yang dipilih untuk kata tertentu
4. β (beta) : Hyperparameter yang mengontrol distribusi kata dalam suatu topik
5. w (word) : Kata yang diamati dalam dokumen
6. M : Jumlah dokumen dalam korpus
7. N : Jumlah kata dalam satu dokumen

Peluang LDA dihitung dengan persamaan berikut:

$$P(w|\alpha, \beta) = \iint P(w, z, \theta, \phi|\alpha, \beta) d\theta d\phi \quad (2.12)$$

2.1.3 Hyperparameter Tuning

Hyperparameter adalah parameter yang dikonfigurasi sebelum proses pelatihan model *machine learning* dimulai. *Tuning* yang tepat dapat menghasilkan peningkatan signifikan dalam akurasi dan efisiensi model, sementara *tuning* yang buruk dapat mengakibatkan kinerja yang tidak optimal atau pemborosan sumber daya komputasi (Victoria & Maragatham, 2020). Proses *tuning* hyperparameter sangat penting untuk mencapai kinerja prediktif yang optimal dan seharusnya menjadi komponen standar dalam evaluasi model

(Arnold et al., 2024). *Grid search* adalah metode yang digunakan dalam optimasi *hyperparameter* untuk secara sistematis menjelajahi subset tertentu dari ruang *hyperparameter* guna menemukan set parameter optimal untuk model *machine learning*, metode ini memastikan bahwa tidak ada konfigurasi potensial yang terlewatkan, yang dapat menjadi hal penting dalam menemukan pengaturan model yang optimal (Belete & Huchaiah, 2021). Secara tradisional, penyetelan *hyperparameter* dilakukan secara manual melalui proses *trial-and-error*, di mana praktisi menyesuaikan parameter berdasarkan intuisi dan pengalaman. Metode ini sering kali merepotkan dan tidak efisien, terutama untuk model yang kompleks dengan banyak *hyperparameter* (Bischi et al., 2021). Meskipun tidak efisien, metode *trial-and-error* tetap menjadi pendekatan yang umum digunakan karena kesederhanaannya dan tidak memerlukan alat atau algoritma khusus.

2.1.4 Evaluasi Model *Sentiment Analysis*

Tahap terakhir dalam penelitian ini adalah mengevaluasi kinerja model *sentiment analysis*. Evaluasi ini dilakukan untuk membandingkan performa dan efektivitas model klasifikasi berbasis *machine learning* dan *deep learning* yang digunakan. Proses evaluasi memegang peran penting dalam pengembangan model, karena bertujuan memastikan bahwa model dapat bekerja dengan baik pada data yang belum pernah digunakan sebelumnya. Salah satu teknik yang digunakan untuk mengevaluasi dan merangkum kinerja pemodelan klasifikasi *machine learning* adalah *confusion matrix*. Metode yang ampuh untuk menganalisis pengklasifikasi multi-kelas adalah *confusion matrix* untuk menunjukkan distribusi prediksi yang salah dalam satu tampilan (Heydarian et al., 2022). *Confusion Matrix* untuk klasifikasi biner menunjukkan bagaimana prediksi model dibandingkan dengan nilai aktual. Dalam kasus ini, label aktual terdiri dari Benar (1) dan Salah (0), sementara model membuat prediksi sebagai Positif (1) dan Negatif (0), kemungkinan model klasifikasi diturunkan dari ekspresi *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), *False Negative* (FN) yang ada di dalam *confusion matrix* (Vujović, 2021).

Confusion Matrix sendiri adalah Tabel dengan baris dan kolom yang berisi label yang diprediksi dan kelas sebenarnya. Contoh *confusion matrix* dapat dilihat di Tabel berikut :

Tabel 2. 1 *Confusion Matrix*

Class designation		Actual class		
		<i>True Negative</i> (0)	<i>True Neutral</i> (1)	<i>True Positive</i> (2)
Predicted class	<i>Negative</i> (0)	TN	FN	FN

	<i>Neutral</i> (1)	FNU	TNU	FNU
	<i>Positive</i> (2)	FP	FP	TP

Berikut adalah rincian dari *confusion matrix*:

1. *True Negative* (TN): Nilai yang sebenarnya *negative* dan diprediksi dengan benar sebagai *negative* oleh model.
2. *True Neutral* (TNU): Nilai yang sebenarnya *neutral* dan diprediksi dengan benar sebagai *neutral* oleh model.
3. *True Positive* (TP): Nilai yang sebenarnya *positive* dan diprediksi dengan benar sebagai *positive* oleh model.
4. *False Negative* (FN): Nilai yang sebenarnya *positive* atau *neutral* tetapi diprediksi dengan salah sebagai *negative* oleh model.
5. *False Neutral* (FNU): Nilai yang sebenarnya *positive* atau *negative* tetapi diprediksi dengan salah sebagai *neutral* oleh model.
6. *False Positive* (FP): Nilai yang sebenarnya *negative* atau *neutral* tetapi diprediksi dengan salah sebagai *positive* oleh model.

Dengan *confusion matrix*, peneliti juga bisa mengetahui *accuracy*, *precision*, *recall* dan *F1-score*. *Accuracy* adalah metrik yang mengukur proporsi yang diprediksi dengan benar dari total (Islam et al., 2022). *Accuracy* dihitung menggunakan persamaan berikut :

$$ACC = \frac{TP+TN}{TP+TN+FN+FP} \quad (2.13)$$

Precision menunjukkan proporsi prediksi positif yang benar di antara semua prediksi positif yang dibuat oleh model (Vats et al., 2024). *Precision* dihitung menggunakan persamaan berikut:

$$PREC = \frac{TP}{(TP+FP)} \quad (2.14)$$

Recall mengukur kemampuan model untuk mengidentifikasi semua instance yang relevan, yaitu proporsi *true positive* yang terdeteksi dari seluruh instance positif aktual (Islam et al., 2022). *Recall* dihitung menggunakan persamaan berikut:

$$REC = \frac{TP}{(TP+FN)} \quad (2.15)$$

F1-Score adalah rata-rata harmonik dari *precision* dan *recall*, yang memberikan keseimbangan antara keduanya (Takahashi et al., 2023). *F1-Score* dihitung menggunakan persamaan berikut:

$$F1 = 2 \cdot \frac{Precision \times Recall}{Precision + Recall} \quad (2.16)$$

Evaluasi tingkat akurasi per epoch menyoroti dampak signifikan dari panjang epoch dan proses identifikasi terhadap performa. Studi tentang modifikasi prosodi menekankan pentingnya akurasi identifikasi epoch, karena hal ini secara langsung memengaruhi kualitas perseptual dari ucapan yang dimodifikasi (Adiga et al., 2014).

2.1.5 Evaluasi Model *Topic Modeling*

Evaluasi model *Latent Dirichlet Allocation* (LDA) melibatkan beberapa langkah kunci untuk memastikan keandalan dan validitas model dalam menangkap topik-topik yang bermakna dari data teks. *Coherence Score* telah diusulkan untuk mengatasi keterbatasan ini dengan mengevaluasi kesamaan semantik antar kata dalam topik, sehingga memberikan evaluasi yang lebih komprehensif terhadap kualitas topik (Blei et al., 2002). LDavis adalah alat populer untuk memvisualisasikan dan menginterpretasikan model LDA. Alat ini membantu memahami hubungan antar topik dan signifikansi kata-kata utama dalam setiap topik, sehingga meningkatkan keterbacaan model (Kim et al., 2023).

2.2 Kajian Pustaka

Berdasarkan Tabel 2.2 dalam kajian pustaka, dapat disimpulkan bahwa pendekatan yang digunakan dalam penelitian ini memiliki kesesuaian yang kuat dengan berbagai studi sebelumnya. Beberapa penelitian terdahulu, seperti yang dilakukan oleh Fitri et al. (2019) dan Hulu & Lhaksana (2019), menunjukkan bahwa algoritma Naïve Bayes dan Support Vector Machine (SVM) sangat efektif dalam mengklasifikasikan sentimen pada tweet dengan topik sosial dan politik, dengan akurasi berkisar antara 62% hingga 83%. Penelitian ini juga menggunakan kedua algoritma tersebut dan menghasilkan akurasi masing-masing sebesar 64% (Naïve Bayes) dan 63% (SVM), yang masih berada dalam rentang wajar sebagaimana ditunjukkan dalam studi sebelumnya.

Selain itu, metode deep learning seperti Long Short-Term Memory (LSTM) dan Gated Recurrent Unit (GRU) yang digunakan dalam penelitian ini juga telah banyak diterapkan dalam berbagai penelitian lain dengan hasil yang menjanjikan. Misalnya, Aakash et al. (2024) dan AL-Smadi et al. (2023) mencatat bahwa LSTM dan GRU mampu mencapai akurasi dan *recall* di atas 90% dalam konteks ulasan produk dan hotel. Meskipun dalam penelitian ini performa LSTM dan GRU belum melampaui model klasik, hal tersebut sesuai dengan temuan literatur bahwa model deep learning cenderung membutuhkan data dalam

jumlah besar untuk dapat menunjukkan keunggulannya secara signifikan. GRU tetap menjadi pilihan yang efisien karena keunggulannya dalam komputasi yang lebih ringan.

Dalam hal pemodelan topik, penggunaan *Latent Dirichlet Allocation (LDA)* dalam penelitian ini juga mendapat dukungan kuat dari literatur. Studi-studi seperti oleh Muhajir et al. (2024), Mulyani et al. (2024), dan Melton et al. (2021) menunjukkan bahwa LDA dapat secara efektif mengidentifikasi topik-topik laten dari data teks media sosial maupun artikel berita. Dalam konteks ini, penelitian ini berhasil mengidentifikasi delapan topik utama dalam diskusi mengenai Pemilu 2024, yang mencakup isu-isu seperti dukungan terhadap calon, keamanan pemilu, hingga peran KPU dan Bawaslu. Hal ini menunjukkan bahwa pendekatan LDA sangat sesuai untuk menggali wacana publik dalam konteks politik.

Secara keseluruhan, pendekatan yang digunakan dalam penelitian ini sejalan dengan tren metodologis dan empiris dalam kajian terdahulu. Kombinasi metode *machine learning*, *deep learning*, dan *topic modeling* tidak hanya relevan secara teknis, tetapi juga kontekstual dengan fokus utama penelitian, yaitu memahami opini publik terhadap Pemilihan Umum 2024 melalui analisis sentimen dan topik dalam tweet berbahasa Indonesia.

Tabel 2. 2 Daftar *Literature Review*

Penulis dan Tahun	Metode	Dataset	Temuan
(Fitri et al., 2019)	Naive Bayes, Decision Tree, Random Forest	Tweet dengan hashtag dan query terkait Anti-LGBT (3744 tweet)	Naive Bayes mencapai akurasi 83,43%, lebih tinggi dibandingkan algoritma lainnya.
(Awotunde et al., 2023)	Naive Bayes	Kaggle: Datafiniti's Business Database	Akurasi uji: 96,08%, F1-score: 96,00%, Precision: 96,00%, Recall: 96,00%
(Hulu & Lhaksana, 2019)	SVM, TF-IDF	Tweet tentang Pemilu Presiden Indonesia 2019	SVM dan TF-IDF mencapai akurasi tertinggi sebesar 62,88%.
(Muhajir et al., 2024)	SVM, LDA	Tweet dengan kata kunci "Bjorkanism" dan "Hacker Bjorka" (15.206 tweet)	Akurasi tinggi konsisten di atas 85%.
(Iddrisu et al., 2023)	GRU, SVM	Tweet tentang dampak pandemi COVID-19 terhadap perjalanan udara	SVM yang dioptimalkan mencapai akurasi 89%.
(Aakash et al., 2024)	SVC, Random Forest, KNN, Logistic Regression, Naive Bayes, LSTM, GRU	Ulasan produk menggunakan URL dari barang tertentu	LSTM: 91% recall, GRU: 92% recall.
(AL-Smadi et al., 2023)	GRU	Ulasan hotel berbahasa Arab dari SemEval	Model mencapai akurasi 91,4% dan F1-score 90,86%.
(Talaat, 2023)	BiLSTM, BiGRU	Dataset tweet dari	Akurasi terbaik: 91,72% (RoBERTa-3G), 88,04%

		HuggingFace	(DistilBERT-GLG).
(Murfi et al., 2024)	CNN, LSTM, GRU, hybrid models, BERT	Ulasan e-commerce dari Shopee, Tokopedia, Lazada	Hasil terbaik menggunakan BERT: Tokopedia (LSTM-CNN, akurasi), Shopee (CNN-LSTM, <i>precision</i>), Lazada (CNN-GRU, <i>recall</i>).
(Damayanti & Lhaksana, 2024)	SVM, Word2Vec	Tweet tentang Pemilu Presiden Indonesia 2024	<i>Accuracy: 90.75%, Precision: 88.94%, Recall: 93.08%, F1-Score: 90.43%.</i>
(Idris et al., 2025)	SVM, Random Forest, Naive Bayes	Ulasan aplikasi Tokopedia dari Google Play Store	SVM dengan FastText mencapai akurasi 89%.
(Akbar et al., 2024)	Naive Bayes	Ulasan pengguna aplikasi InDriver dari Google Play Store	Naive Bayes mencapai akurasi 76%.
(Mulyani et al., 2024)	BERT, LDA	Artikel dari portal online	Diskursus publik tentang adopsi PV (panel surya) campuran; koherensi LDA: 0,63. Media mainstream positif, media sosial skeptis.
(Melton et al., 2021)	Lexicon. LDA	Diskusi terkait vaksin COVID-19 di media sosial	Diskusi Reddit tentang vaksin COVID-19 menunjukkan 56,68% positif; koherensi LDA 0,5641; keragu-raguan masih ada, efek samping mendominasi.

BAB III

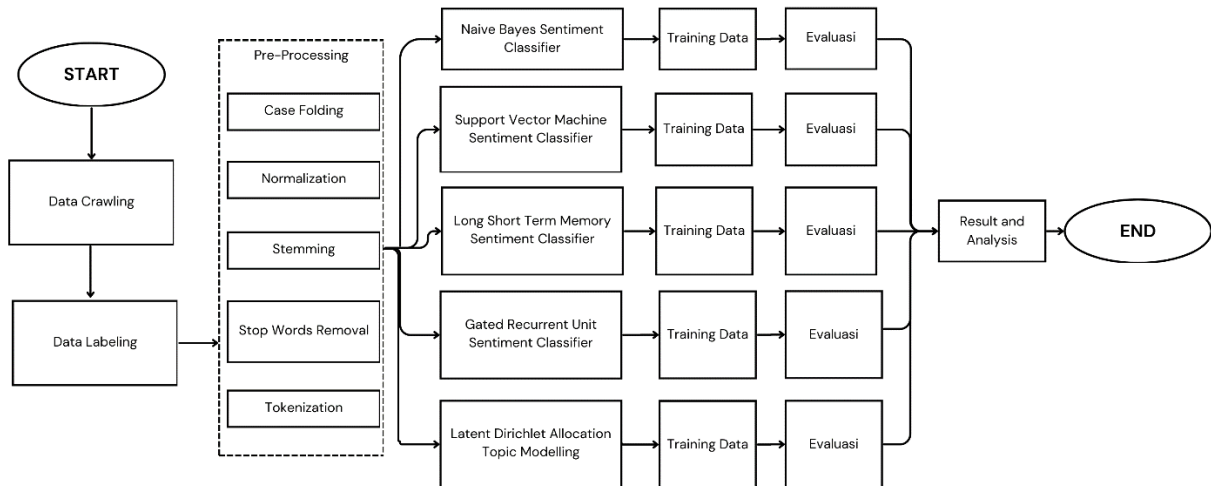
METODOLOGI PENELITIAN

Bab ini menjelaskan metodologi penelitian yang digunakan untuk menganalisis sentimen dan memodelkan topik dari data teks yang diambil dari media sosial. Pendekatan ini memanfaatkan metode *deep learning* dan algoritme statistik untuk menghasilkan analisis yang mendalam dan komprehensif. Bagian ini merinci pendekatan penelitian dari awal hingga akhir, termasuk pengumpulan data, pra-pemrosesan, analisis, dan interpretasi hasil. Penggunaan Naïve Bayes, SVM, LSTM dan GRU dalam analisis sentimen dipilih karena kemampuannya dalam menangani data teks sekuensial, sedangkan LDA digunakan untuk mengidentifikasi distribusi topik dalam kumpulan tweet.

Penelitian ini bertujuan untuk memberikan wawasan yang lebih dalam tentang opini publik melalui analisis sentimen dan pemodelan topik dari data yang tidak terstruktur. Dengan menerapkan pendekatan metodologis yang rigor dalam setiap tahap penelitian, diharapkan hasil yang diperoleh dapat memberikan kontribusi yang signifikan dalam bidang analisis teks dan pemahaman sosial. Bab ini akan menguraikan berbagai teknik dan alat yang digunakan, serta bagaimana masing-masing metode berkontribusi terhadap tujuan keseluruhan penelitian.

3.1 Desain Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan teknik analisis data teks. Desain penelitian ini terdiri dari beberapa tahap, yaitu data crawling, pra-pemrosesan data, penerapan model Naïve Bayes, SVM, LSTM dan GRU untuk analisis sentimen, dan penerapan model LDA untuk pemodelan topik.



Gambar 3. 1 Desain Penelitian

Desain penelitian di atas melakukan analisis sentiment dan *topic modeling* dengan melakukan proses *crawling* tepatnya dari situs X atau dulu dikenal sebagai twitter. Hasil dari *crawling* dari X dikumpulkan dan menjadi sebuah dataset. Dataset kemudian akan diberi label secara manual dengan label *positive*, *neutral*, dan *negative*. Dataset selanjutnya akan melalui tahap *Preprocessing*. *Preprocessing* adalah tahapan untuk membersihkan data yang awalnya tidak terstruktur menjadi lebih terstruktur agar bisa masuk ke tahap pemodelan. Beberapa tahapan pada *Preprocessing*, yaitu *case folding*, *normalization*, *stemming*, *stopwords removal*, dan *tokenization*. Setelah melewati beberapa proses sebelumnya, dataset kemudian akan digunakan untuk melatih model Naïve Bayes, SVM, LSTM, dan GRU untuk klasifikasi *sentiment* menjadi tiga label yaitu *positive*, *negative*, dan *neutral*. Sedangkan LDA digunakan untuk melakukan *topic modeling*. Data yang telah terklasifikasi kemudian dievaluasi melihat keakuratan prediksi.

3.2 Pengumpulan Data

Data diambil dari media social X/Twitter dengan teknik Tweet Harvest. Proses Crawling dilakukan dengan menggunakan library python dan juga X Authorization Token, keyword yang digunakan untuk proses crawling adalah “Pemilu” dengan menentukan tanggal pencarian sesuai dengan batasan masalah. Data yang berhasil dikumpulkan selanjutnya akan disimpan dalam format CSV (Comma-Separated Values). Proses Crawling dilakukan beberapa kali dikarenakan X memiliki limit jumlah tweet yang bisa dilihat. Dataset hasil *crawling* masih belum memiliki label, yang berarti belum memiliki label atau *sentiment positive*, *neutral*, dan *negative*. Proses pelabelan ini akan dilakukan secara manual yang nantinya akan dibagi menjadi tiga label yaitu

positive, neutral, dan negative.

3.3 Preprocessing Data

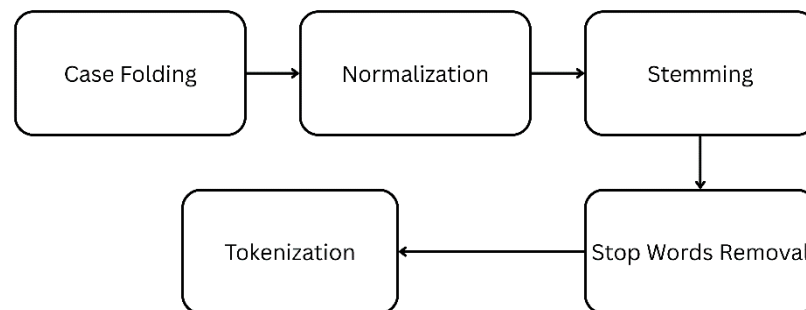
Untuk memastikan transparansi dan reproduktifitas dalam penelitian NLP, penting untuk memberikan dokumentasi yang jelas tentang keputusan prapemrosesan dan dampaknya terhadap analisis. Hal ini termasuk mempertimbangkan jenis *text mining* yang dilakukan dan karakteristik dataset (Hickman et al., 2022). Berikut ini adalah beberapa tahapan dalam *Preprocessing* seperti dibawah ini :

1. **Case Folding:** Dalam analisis teks, case folding digunakan untuk menstandarisasi data teks, yang membantu mengurangi kompleksitas teks dengan memperlakukan kata-kata dengan huruf besar/kecil yang berbeda sebagai kata yang sama. Hal ini sangat berguna dalam tugas pemrosesan bahasa alami seperti analisis sentimen, pencarian informasi, dan klasifikasi teks. Hal ini memungkinkan untuk menghindari perbedaan hasil yang di akibatkan variasi huruf besar dan kecil. Contohnya, teks “PEMILU SERENTAK”, “Pemilu Serentak”, dan “pemilu serentak” akan diubah menjadi “pemilu serentak”.
2. **Normalization:** Normalisasi bertujuan untuk mengubah kata-kata tidak baku menjadi bentuk baku agar kata-kata dengan makna serupa memiliki bentuk yang seragam. Proses ini membantu menyatukan variasi penulisan kata yang berbeda namun memiliki arti yang sama, sehingga menciptakan konsistensi dalam penggunaan bahasa, seperti “bngr” menjadi “banget”.
3. **Stemming:** *Stemming* adalah teknik yang digunakan untuk mengurangi kata menjadi bentuk dasar atau akarnya. *Stemming* melibatkan pemotongan awalan atau akhiran, sementara lemmatization menggunakan pengetahuan linguistik untuk mendapatkan bentuk dasar. Metode-metode ini sangat penting untuk mengurangi dimensi dan meningkatkan kinerja sistem NLP (Jabbar et al., 2023). Seperti kata “kemenangan” dapat diubah menjadi kata dasar “menang”.
4. **Stop Words Removal:** Penghapusan *stopwords* sangat penting untuk mengoptimalkan proses dalam pencarian informasi dan sistem analisis teks. Dengan menyaring kata-kata yang berlebihan, sistem-sistem ini dapat mencapai akurasi dan kinerja yang lebih tinggi (Kaur & Saini, 2015). Penghapusan *stopwords* mengurangi ukuran korpus sebesar 35-45%, yang pada gilirannya meningkatkan efisiensi dan akurasi aplikasi penambangan teks dengan mengurangi kompleksitas ruang dan waktu (Ladani & Desai, 2020). Pada tahap ini, dilakukan

penghapusan kata-kata umum atau *stop-words* seperti "yang", "di", "ke", dan sejenisnya.

5. Tokenization: Memisahkan teks menjadi kata-kata atau token adalah hal yang mendasar untuk sebagian besar tugas NLP (Chai, 2023). Hal ini meningkatkan akurasi penandaan bagian dari ucapan dan analisis lainnya (Camacho-Collados & Pilehvar, 2018). Tahap ini akan memecah teks menjadi “token” yang biasanya berupa kata. Seperti kalimat “Kerja Keras PSI Cegah Kekalahan Terulang Seperti Pemilu 2019” akan dipecah menjadi [‘kerja’, ‘keras’, ‘psi’, ‘cegah’, ‘kalah’, ‘ulang’, ‘pemilu’, ‘indonesiakuatindonesiamaju’, ‘diplomasi progresif’, ‘lanjut’].

Melalui serangkaian tahapan praproses yang dilakukan, data teks menjadi lebih bersih dan terorganisir, sehingga lebih siap digunakan dalam analisis klasifikasi sentimen dan pemodelan topik pada penelitian ini. Langkah ini juga berperan dalam mengurangi gangguan atau noise yang dapat memengaruhi akurasi hasil analisis.



Gambar 3. 2 Alur *Preprocessing*

Dalam penelitian ini, tahapan *Preprocessing* data teks dilakukan secara sistematis untuk menghasilkan data yang bersih, terstruktur, dan siap digunakan dalam proses klasifikasi sentimen berbasis aspek. Proses ini dimulai dengan casefolding, yaitu mengubah seluruh huruf dalam teks menjadi huruf kecil (lowercase) dengan tujuan menyamakan bentuk kata, seperti “Pemilu” dan “pemilu”, agar dikenali sebagai kata yang sama. Setelah itu, dilakukan normalization atau normalisasi, di mana kata-kata tidak baku, singkatan, atau bentuk slang seperti “gk” atau “nggak” diubah menjadi bentuk baku seperti “tidak”. Dengan menggunakan “Kamus Alay” yang merupakan *dictionary* yang berisi kata *slang* dan kata dalam bentuk formalnya. Kamus ini di dapatkan dari Github yang berisi sebanyak 3264 kata *slang* dan formalnya. Untuk contoh kamus dapat dilihat di Tabel 3.1.

Tabel 3. 1 Contoh Kata Kamus Alay

<i>Slang</i>	Formal
woww	wow
aminn	amin
met	selamat
netaas	menetas
keberpa	keberapa

Langkah berikutnya adalah stemming, yaitu proses mengembalikan kata ke bentuk dasarnya. Misalnya, kata “menyuarakan” akan dikembalikan menjadi “suara”, menggunakan *library* dari Sastrawi. Proses ini penting untuk menyederhanakan variasi kata agar lebih mudah dianalisis. Setelah proses stemming dilakukan, tahap selanjutnya adalah *stopwords removal*, yaitu penghapusan kata-kata umum yang tidak memberikan kontribusi berarti terhadap analisis sentimen, seperti “yang”, “dan”, “di”, atau “ke”. Kata-kata tersebut umumnya memiliki frekuensi tinggi dalam teks, tetapi tidak mengandung makna kontekstual yang mendukung pengklasifikasian sentimen. Dalam penelitian ini, proses *stopwords removal* dilakukan dengan memanfaatkan pustaka *Natural Language Toolkit* (NLTK), yang menyediakan daftar stopwords bahasa Indonesia. Dengan menggunakan library ini, sistem dapat secara otomatis menghapus kata-kata tidak penting dari korpus sehingga meningkatkan efisiensi dan akurasi model dalam menangkap makna utama dari teks.

Langkah terakhir dalam tahap *Preprocessing* adalah tokenization, yaitu proses memisahkan kalimat atau paragraf menjadi unit-unit kata yang lebih kecil yang disebut token. Sebagai contoh, kalimat “Pemilu 2024 sangat meriah” akan diubah menjadi daftar token seperti [“pemilu”, “2024”, “sangat”, “meriah”]. Dalam penelitian ini, proses tokenisasi dilakukan menggunakan Tokenizer dari library TensorFlow Keras, yang memungkinkan konversi teks menjadi urutan angka berdasarkan frekuensi kata dalam korpus. Penggunaan tokenizer ini membantu dalam mengubah data teks menjadi format numerik yang dapat digunakan oleh model deep learning. Melalui keseluruhan proses *Preprocessing* — mulai dari case folding, normalization, stemming, stopwords removal, hingga tokenization — data teks menjadi lebih bersih, terstruktur, dan siap digunakan dalam pelatihan model analisis sentimen dan pemodelan topik secara lebih efektif dan akurat.

3.4 Pemodelan Naïve Bayes

Naive Bayes digunakan sebagai salah satu model dasar dalam klasifikasi teks. Metode ini bekerja berdasarkan prinsip probabilistik, di mana setiap kata dalam dokumen dianggap saling bebas (independen). Data teks yang telah diproses diubah menjadi representasi numerik menggunakan CountVectorizer dan TF-IDF untuk menghasilkan bobot dari setiap kata dalam dokumen.

Model Naive Bayes yang digunakan adalah Multinomial Naive Bayes, yang cocok untuk data diskrit seperti frekuensi kata. Dataset dibagi menjadi data latih dan data uji dengan rasio 80:20 menggunakan stratifikasi label. Evaluasi model dilakukan menggunakan metrik akurasi, classification report (*precision*, *recall*, *f1-score*), dan confusion matrix.

3.5 Pemodelan Support Vector Machine

SVM digunakan sebagai metode pembandingan dengan pendekatan klasifikasi berbasis margin. Sama seperti pada Naive Bayes, data teks dikonversi menjadi representasi numerik menggunakan CountVectorizer dan TF-IDF. Model SVM dilatih menggunakan kernel RBF (Radial Basis Function) dengan parameter $C=1$ untuk menangani trade-off antara margin maksimal dan kesalahan klasifikasi. Dataset dibagi secara stratifikasi menjadi data latih dan uji (80:20). Evaluasi performa dilakukan dengan menghitung akurasi, *classification report*, dan *confusion matrix*.

3.6 Hyperparameter Model Deep Learning

Pemilihan hyperparameter merupakan tahapan kritis dalam pengembangan model *deep learning*. Konfigurasi yang optimal ditentukan melalui serangkaian eksperimen untuk menyeimbangkan antara performa model dan kompleksitas komputasi. Berikut adalah konfigurasi akhir yang digunakan pada model LSTM dan GRU:

Tabel 3. 2 Konfigurasi Model LSTM dan GRU

Komponen	LSTM	GRU	Keterangan
Arsitektur	LSTM (196)	GRU (196)	Unit recurrent layer
Dropout	0.2	0.2	Input+recurrent dropout
Relularisasi	-	L2 (0.01)	Hanya pada GRU
Optimizer	Adam	Adam	Learning rate 0.0001
Batch size	32	32	

Epoch	10	10	Early stopping (patience = 3)
-------	----	----	-------------------------------

Tabel 3.1 membandingkan konfigurasi hyperparameter antara model LSTM dan GRU yang digunakan dalam penelitian ini. Kedua model memiliki kesamaan dalam beberapa aspek kunci: (1) jumlah unit pada recurrent layer (196 unit), (2) nilai dropout (0.2) untuk regularisasi, dan (3) penggunaan optimizer Adam dengan learning rate 0.0001. Perbedaan utama terletak pada penerapan regularisasi L2 ($\lambda=0.01$) yang hanya diaplikasikan pada model GRU, sementara model LSTM tidak menggunakan regularisasi tambahan selain dropout.

Untuk proses pelatihan, kedua model menggunakan batch size 32 dengan mekanisme *early stopping* yang menghentikan pelatihan jika tidak terjadi peningkatan pada *validation loss* selama 3 “kesempatan” berturut-turut.

3.7 Pemodelan LSTM

Penelitian ini mengimplementasikan model LSTM dengan arsitektur tiga lapisan untuk analisis sentimen. Lapisan embedding mengubah kata menjadi vektor 128-dimensi dengan vocabulary 20.000 kata. Lapisan LSTM inti terdiri dari 196 unit neuron dengan dropout rate 0.2 untuk mencegah overfitting. Lapisan output menggunakan aktivasi softmax untuk klasifikasi tiga kategori sentimen. Proses pelatihan dioptimasi dengan Adam (learning rate 0.0001) menggunakan batch size 32 dan early stopping. Hasil evaluasi menunjukkan performa yang memadai dengan akurasi. Analisis *classification report* mengungkapkan bahwa model mencapai *precision* dan *recall* untuk kelas dominan. Confusion matrix memperlihatkan bahwa sebagian besar kesalahan klasifikasi terjadi antara kelas netral dan positif.

Visualisasi proses pelatihan menunjukkan pola konvergensi yang stabil antara training dan validation set. Meskipun efektif menangkap konteks kalimat, model ini memerlukan sumber daya komputasi yang signifikan dibandingkan pendekatan klasik.

3.8 Pemodelan GRU

Penelitian ini mengimplementasikan model GRU sebagai alternatif dari model LSTM sebelumnya. Secara arsitektural, GRU lebih sederhana dengan hanya dua gerbang (reset dan update) dibanding LSTM yang memiliki tiga gerbang. Kedua model menggunakan parameter serupa dalam hal dimensi embedding (128), jumlah unit (196), dan mekanisme dropout (0.2). Namun, GRU menambahkan regularisasi L2 ($\lambda=0.01$) yang tidak ada di implementasi LSTM

sebelumnya.

Kedua model dievaluasi menggunakan skema pembagian data yang sama (90% training, 10% testing) dengan *Preprocessing* identik.

3.9 Pemodelan LDA

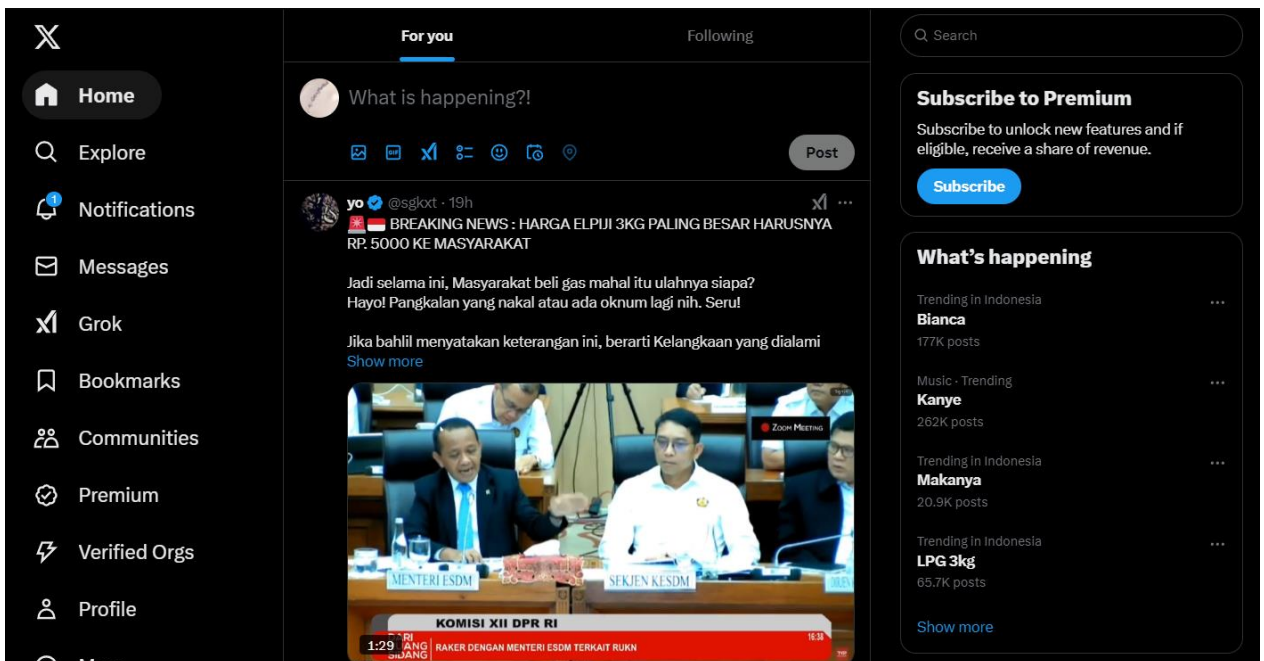
Pemodelan topik menggunakan Latent Dirichlet Allocation (LDA) dilakukan melalui beberapa tahapan utama. Pertama, data teks yang telah diproses diubah menjadi representasi bag-of-words dengan membangun kamus dari seluruh korpus. Model LDA kemudian dibangun dengan menentukan jumlah topik optimal melalui evaluasi koherensi untuk rentang 2-10 topik, menggunakan metrik Coherence Score (CV).

BAB 4

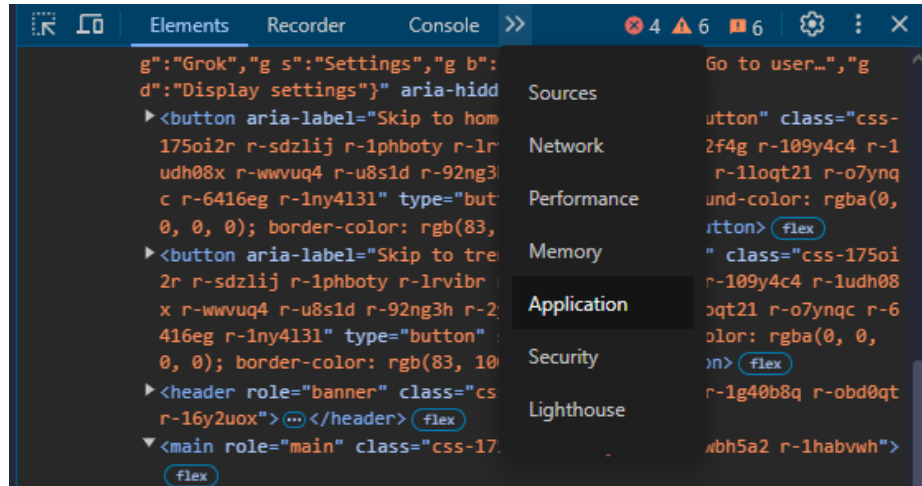
HASIL DAN PEMBAHASAN

4.1 Crawling Data

Proses pengumpulan data dilakukan dengan metode *crawling*, proses ini akan menggunakan *tools* yang bernama *Tweet Harvest*. *Tweet Harvest* adalah alat berbasis command-line yang menggunakan Playwright untuk melakukan *crawling* tweet dari hasil pencarian Twitter/X berdasarkan kata kunci (keywords) dan rentang tanggal yang ditentukan. Tweet yang berhasil diambil kemudian disimpan dalam file CSV. Sebelum melakukan *crawling data*, untuk menggunakan *tools Tweet Harvest* kita harus memiliki akun X dan *Authentication Token* dari akun tersebut. Untuk mendapatkan *auth_token* ada beberapa tahap yang harus dilalui. Pada tahap awal kita harus melakukan *log in* pada sosial media X. Setelah berhasil masuk ke *homepage X* seperti Gambar 4.1, selanjutnya kita harus melakukan *inspect element* seperti Gambar 4.2. *Auth_token* terdapat di *tab application* seperti Gambar 4.3. Tahap terakhir untuk mendapatkan *auth_token* adalah dengan membuka *cookies* dari X, dari *cookies* dapat dilihat *auth_token* dengan *value* seperti pada Gambar 4.4 yang nantinya akan digunakan untuk proses *crawling*.

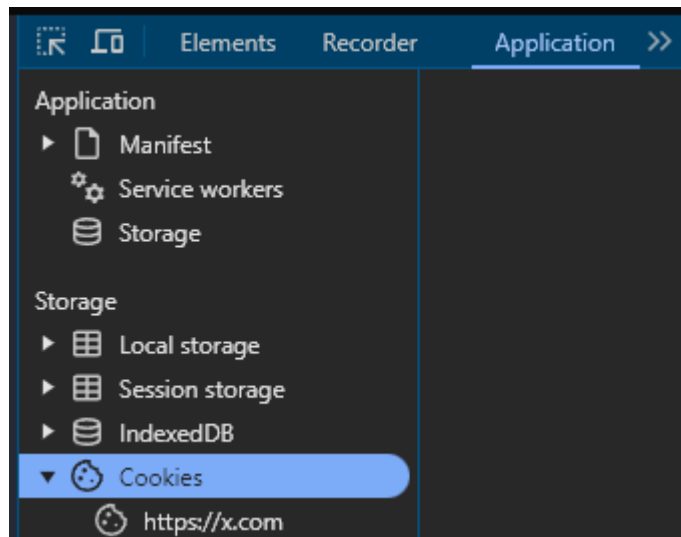


Gambar 4. 1 Home Page X



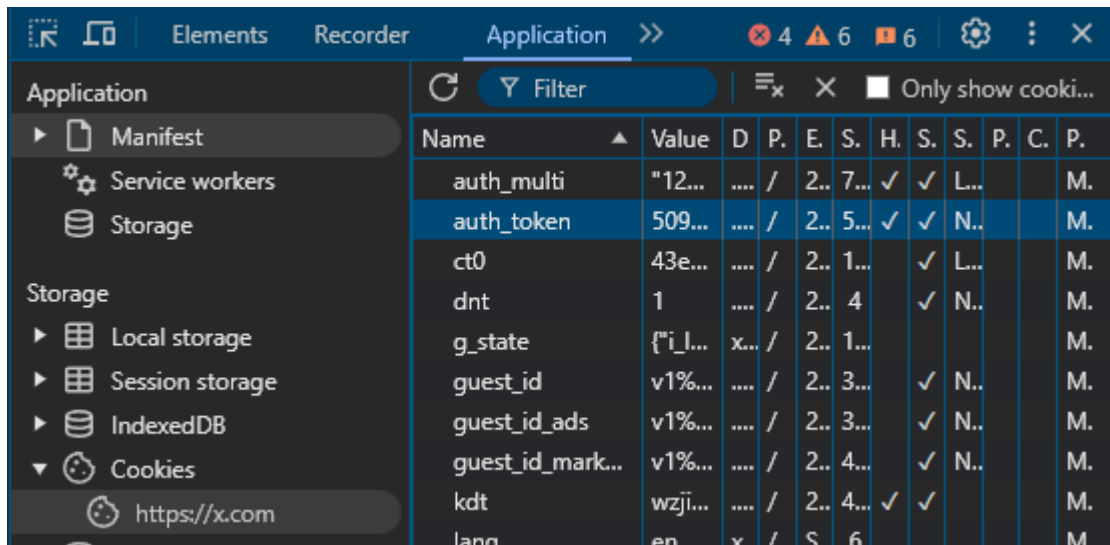
Gambar 4. 2 *Inspect Element*

Inspect element dapat diakses melalui beberapa cara, contohnya dengan membukakan halaman web, lalu klik kanan pada element yang ingin diperiksa dan pilih “*Inspect*”.



Gambar 4. 3 *Inspect Element Application*

Gambar 4.3 menunjukkan tampilan panel Application dalam Developer Tools browser yang berfokus pada fitur penyimpanan dan cache website. Panel ini merupakan bagian penting untuk pengembangan web karena memungkinkan developer memeriksa dan mengelola berbagai jenis penyimpanan sisi klien.



Gambar 4. 4 Cookies URL X dengan *auth_token*

Gambar ini memperlihatkan antarmuka panel Application dalam Developer Tools browser yang mengeksplorasi mekanisme penyimpanan sisi klien untuk situs x.com (Twitter). Pada bagian atas terlihat struktur hierarkis yang mencakup tiga komponen utama: Manifest yang berisi konfigurasi Progressive Web App, Service Workers untuk operasi latar belakang, dan Storage yang mengelola berbagai jenis penyimpanan data. Beberapa cookie kritis yang teridentifikasi termasuk **auth_token** untuk manajemen sesi login, *guest_id* untuk melacak pengunjung anonim, dan *ct0* sebagai token proteksi CSRF.

Proses *crawling* tersebut dimulai dengan menyiapkan token autentikasi X yang diperlukan untuk mengakses data. Selanjutnya, dilakukan instalasi library pandas seperti Gambar 4.5 untuk manipulasi data dan Node.js sebagai lingkungan *runtime* yang dibutuhkan oleh alat *tweet-harvest*. Setelah dependensi terpenuhi, proses *crawling* tweet dilakukan menggunakan *tweet-harvest* dengan menentukan parameter seperti kata kunci pencarian (pemilu dalam bahasa Indonesia), rentang tanggal (dari 17 Februari 2024 hingga 30 Februari 2024, serta batasan jumlah tweet yang diambil (maksimal 2000 tweet) seperti Gambar 4.6. Proses *crawling tweet* dapat dilihat seperti Gambar 4.7. Hasil *crawling* disimpan dalam file CSV. Setelah data berhasil diambil, file CSV tersebut dibaca ke dalam sebuah DataFrame menggunakan pandas untuk memudahkan analisis dan manipulasi data seperti Gambar 4.8.

```
# Import required Python package
!pip install pandas

# Install Node.js (because tweet-harvest built using Node.js)
!sudo apt-get update
!sudo apt-get install -y ca-certificates curl gnupg
!sudo mkdir -p /etc/apt/keyrings
!curl -fsSL https://deb.nodesource.com/gpgkey/nodesource-repo.gpg.key | sudo gpg --dearmor -o /etc/apt/keyrings/nodesource.gpg

NODE_MAJOR=20 && echo "deb [signed-by=/etc/apt/keyrings/nodesource.gpg] https://deb.nodesource.com/node_$NODE_MAJOR.x nodistro main" | sudo tee /etc/apt/sources.list.d/nodesource.list

!sudo apt-get update
!sudo apt-get install nodejs -y

!node -v
```

Gambar 4. 5 Instalasi *library* pandas dan Node.js

```
# Crawl Data

filename = 'pemilu_jan_8_9th_week.csv'
search_keyword = 'pemilu lang:id until:2024-02-31 since:2024-02-17'
limit = 2000

!npx -y tweet-harvest@2.6.1 -o "{filename}" -s "{search_keyword}" --tab "LATEST" -l {limit} --token {twitter_auth_token}
```

Gambar 4. 6 Parameter yang digunakan dalam *tweet harvest*

Proses *scraping* menggunakan beberapa parameter, seperti *keyword* yang dicari, penelitian ini menggunakan *keyword* “Pemilu” dengan *timeline* pencarian digunakan per-minggu dengan tujuan untuk mendapatkan informasi yang lebih luas. Sedangkan limit *tweet* yang ditetapkan sebanyak 2000 *tweet* per *scraping*, ini dikarenakan X memiliki limit *tweet* yang dapat dilihat per hari untuk akun yang belum terverifikasi.

```
This script uses Chromium Browser to crawl data from Twitter with your Twitter auth token.
Please enter your Twitter auth token when prompted.

Note: Keep your access token secret! Don't share it with anyone else.
Note: This script only runs on your local device.

Opening twitter search page...

-- Scrolling... (1) (2) (3) (4)
Filling in keywords: pemilu lang:id until:2024-02-28 since:2024-02-17

(5) (6) (7)Created new directory: /content/tweets-data

Your tweets saved to: /content/tweets-data/pemilu_jan_8_9th_week.csv
Total tweets saved: 18
```

Gambar 4. 7 Proses *Crawling Tweet*

Gambar 4.7 ini menjalankan proses *crawling* data dari Twitter menggunakan browser Chromium, dengan kata kunci "pemilu" dalam bahasa Indonesia, untuk rentang waktu 17 Februari 2024 hingga 28 Februari 2024. Selama proses, script berhasil membuat direktori /content/tweets-data dan menyimpan hasil *crawling* sebanyak **18 tweet** ke dalam file pemilu_jan_8_9th_week.csv.

```

import pandas as pd

# Specify the path to your CSV file
file_path = f"tweets-data/{filename}"

# Read the CSV file into a pandas DataFrame
df = pd.read_csv(file_path, delimiter=",")

# Display the DataFrame
display(df)

```

Gambar 4. 8 Menampilkan DataFrame file CSV dari hasil *crawling*

Library pandas digunakan untuk menyimpan hasil *scraping* tweet DataFrame yang nantinya akan di simpan dalam format Comma Separated Value (CSV). Contoh dari hasil *crawling* data dapat dilihat pada Tabel 4.1.

Tabel 4. 1 Hasil *Crawling Data*

<i>Crawling Data</i>
Kerja Keras PSI Cegah Kekalahan Terulang Seperti Pemilu 2019 - https://t.co/SKg61sB7cH https://t.co/p4PiNJ41aJ #IndonesiaKuatIndonesiaMaju #DiplomasiProgresif02 #02Melanjutkan
Sebenarnya gua udah muak bgt dah mantengin huru-hara pemilu ini gua mencoba buat ga peduli tapi GUA GA SUDI DIPIMPIN SAMA PRESIDEN DAN WAKIL MODELAN GITU belum pendukung-pendukungnya juga rusak semua lagi ga ada yang bener
@RcyberProj0 @Gerindra @prabowo @gibrantweet @budimuni98 @handoko2411 @FALDA81 Survei hanya penggiringan opini lucu nya pemilu negeri KONOHA hasil pilpres bisa ditentukan seblm pencoblosan
Dalam menghadapi Pemilu Damai 2024 Anggota Bhabinkamtibmas Polsek Sukajadi dipimpin Kanit Binmas Iptu Subahagia melaksanakan kegiatan Patroli dialogis dengan menyambangi Ketua PPK Kec. Suakajadi Bpk. Cecep bertempat di Kantor Kec. Sukajadi Kota Bandung. https://t.co/az2TMeZc0R
@nongandah @Paltiwest Ingat pemilu 2019 banyak juga pendukung prabowo ditangkap

4.2 Preprocessing

Tahap *pre-processing* hasil dari *crawling* tweet melewati beberapa tahapan yaitu normalisasi, *stemming*, *stopwords removal*, dan tokenisasi. Pertama, diinisialisasi *library* Sastrawi yang berfungsi sebagai *stemmer* Bahasa Indonesia yang akan digunakan untuk mengubah kata-kata menjadi bentuk dasarnya. Selanjutnya, diunduh dan dimuat daftar *stopwords* bahasa Indonesia,

yaitu kata-kata umum yang sering diabaikan dalam analisis teks karena tidak memberikan makna signifikan seperti (“dan”, “di”, “yang”, dan lainnya).

Kemudian, dimuat kamus normalisasi dari file CSV yang berisi pasangan kata slang atau tidak baku dengan kata formalnya seperti Tabel 4.3. Kamus ini digunakan untuk mengubah kata-kata tidak baku dalam teks menjadi bentuk formalnya. Setelah itu, dibuat fungsi untuk normalisasi teks yang mengganti kata-kata slang dengan kata formal berdasarkan kamus yang telah dimuat.

Proses pra-pemrosesan teks meliputi beberapa langkah: menghapus tag HTML, menghapus URL, menghilangkan karakter non-alfanumerik, dan mengubah teks menjadi huruf kecil. Selanjutnya, teks dinormalisasi menggunakan kamus yang telah disiapkan, dihapus stopwords-nya, dan dilakukan *stemming* untuk mengubah kata-kata menjadi bentuk dasarnya. Namun, kata "pemilu" dikecualikan dari proses *stemming* untuk mempertahankan makna spesifiknya. Hasil akhirnya adalah teks yang telah dibersihkan, dinormalisasi, dan siap digunakan untuk analisis lebih lanjut. Untuk dataset yang telah melewati tahap *pre-processing* dapat dilihat pada Tabel 4.3.

Tabel 4. 2 Hasil *Preprocessing*

<i>Preprocessing Step</i>	Hasi <i>Pre-processing</i> data
<i>Teks Original</i>	Kerja Keras PSI Cegah Kekalahan Terulang Seperti Pemilu 2019 - https://t.co/SKg61sB7cH https://t.co/p4PiNJ41aJ #IndonesiaKuatIndonesiaMaju #DiplomasiProgresif02 #02Melanjutkan
<i>Case Folding</i>	kerja keras psi cegah kekalahan terulang seperti pemilu 2019 - https://t.co/SKg61sB7cH https://t.co/p4PiNJ41aJ #indonesiakuatindonesiamaju #diplomasi progresif02 #02melanjutkan
<i>Removing Symbol, URL, and Numbering</i>	kerja keras psi cegah kekalahan terulang seperti pemilu indonesiakuatindonesiamaju diplomasi progresif lanjut
<i>Stopwords Removal</i>	kerja keras psi cegah kalah ulang pemilu indonesiakuatindonesiamaju diplomasi progresif lanju
<i>Tokenization</i>	['kerja', 'keras', 'psi', 'cegah', 'kalah', 'ulang', 'pemilu', 'indonesiakuatindonesiamaju', 'diplomasi progresif', 'lanjut']

4.3 Naïve Bayes

Setelah dataset melalui tahap *pre-processing*, langkah pertama adalah mengubah label sentimen (seperti positif, negatif, atau netral) menjadi bentuk numerik menggunakan LabelEncoder dan OneHotEncoder. Data kemudian dibagi menjadi data latih dan data uji dengan proporsi 80:20, sambil memastikan distribusi label tetap seimbang melalui parameter stratify seperti pada Gambar 4.9.

```
# Encoding labels as one-hot
labels = data['sentiment'].values
encoder = LabelEncoder()
encoded_labels = encoder.fit_transform(labels)

# One-hot encoding
ohe = OneHotEncoder(sparse_output=False)
encoded_labels = ohe.fit_transform(encoded_labels.reshape(-1, 1))

# Split data
train_sentences, test_sentences, train_labels, test_labels = train_test_split(
    data['full_text'].values, encoded_labels, test_size=0.2, stratify=encoded_labels, random_state=42
)

# Vectorize text using CountVectorizer and TfidfTransformer
cv = CountVectorizer()
X_train_cv = cv.fit_transform(train_sentences)
X_test_cv = cv.transform(test_sentences)
```

Gambar 4. 9 Mengubah label *sentiment* menjadi numerik

Selanjutnya, teks diubah menjadi representasi numerik menggunakan CountVectorizer untuk menghitung frekuensi kata, dan TfidfTransformer untuk memberikan bobot berdasarkan pentingnya kata dalam dokumen (Term Frequency-Inverse Document Frequency). Model Multinomial Naive Bayes dilatih menggunakan data latih yang telah diubah menjadi vektor TF-IDF. Setelah pelatihan, model digunakan untuk memprediksi sentimen pada data uji seperti pada Gambar 4.10.

```

# Vectorize text using CountVectorizer and TfidfTransformer
cv = CountVectorizer()
X_train_cv = cv.fit_transform(train_sentences)
X_test_cv = cv.transform(test_sentences)

tf_transformer = TfidfTransformer(use_idf=True).fit(X_train_cv)
X_train_tf = tf_transformer.transform(X_train_cv)
X_test_tf = tf_transformer.transform(X_test_cv)

```

Gambar 4. 10 CountVectorizer dan TF-IDF

Hasil prediksi dievaluasi menggunakan *classification report* yang mencakup metrik seperti presisi, *recall*, dan *F1-score* untuk setiap kelas *sentiment*. Selain itu, dibuat *confusion matrix* untuk memvisualisasikan performa model dalam memprediksi label yang benar. Hasil evaluasi *classification report* dan *confusion matrix* dapat dilihat pada Tabel 4.4.

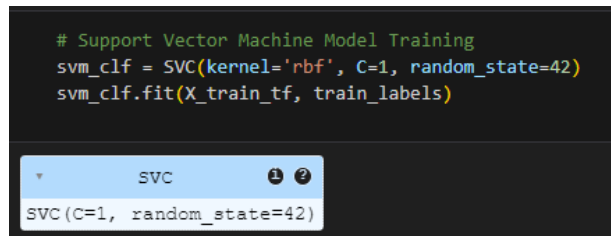
Tabel 4. 3 Hasil pengujian Naïve Bayes

Kategori Klasifikasi	<i>True Negative</i>	<i>True Neutral</i>	<i>True Positive</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
<i>Prediction Negative</i>	194	48	15	66%	75%	71%
<i>Prediction Neutral</i>	65	183	73	58%	57%	58%
<i>Prediction Positive</i>	33	83	186	68%	62%	65%

Berdasarkan hasil pada Tabel 4.4, data dengan total 881 tweet, diperoleh 194 tweet dinyatakan sebagai *true negative*, 183 dinyatakan sebagai *true neutral*, dan 186 dinyatakan sebagai *true neutral*. Rata-rata persentase (%) untuk akurasi adalah 64%, presisi 64%, *recall* 65%, dan *F1-Score* sebesar 64%. Penelitian ini menggunakan Multinomial Naïve Bayes, yang dapat secara efektif mengelola beberapa kelas dan fitur heterogen, yang bermanfaat dalam dataset yang beragam (Pavlov et al., 2004). Hasil ini menunjukkan bahwa Naïve Bayes mampu mengklasifikasikan sentimen dengan performa terbaik pada prediction positive. Namun, pada kelas prediction negative dengan performa lebih rendah yaitu 166, model mengalami kesulitan dalam membedakan sentimen yang mirip atau netral, yang disebabkan oleh kemiripan kata antar kelas.

4.4 Support Vector Machine (SVM)

Model SVM dengan kernel RBF (Radial Basis Function) dilatih menggunakan data latih yang telah diubah menjadi vektor TF-IDF sama seperti tahap Naïve Bayes. Setelah pelatihan, model digunakan untuk memprediksi sentimen pada data uji. Data kemudian dibagi menjadi data latih dan data uji dengan proporsi 80:20, sambil memastikan distribusi label tetap seimbang melalui parameter stratify seperti pada Gambar 4.11.



```
# Support Vector Machine Model Training
svm_clf = SVC(kernel='rbf', C=1, random_state=42)
svm_clf.fit(X_train_tf, train_labels)
```

SVC

SVC(C=1, random_state=42)

Gambar 4. 11 Hyperparameter SVM

Hasil prediksi dievaluasi menggunakan *classification report* yang mencakup metrik seperti presisi, *recall*, dan *F1-score* untuk setiap kelas *sentiment*. Selain itu, dibuat *confusion matrix* untuk memvisualisasikan performa model dalam memprediksi label yang benar. Hasil evaluasi *classification report* dan *confusion matrix* dapat dilihat pada Tabel 4.5.

Tabel 4. 4 Hasil pengujian SVM

Kategori Klasifikasi	<i>True Negative</i>	<i>True Neutral</i>	<i>True Positive</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
<i>Prediction Negative</i>	166	65	25	65%	65%	65%
<i>Prediction Neutral</i>	76	200	90	55%	62%	58%
<i>Prediction Positive</i>	15	90	187	72%	62%	67%

Berdasarkan hasil pada Tabel 4.5, data dengan total 881 tweet, diperoleh 166 tweet dinyatakan sebagai *true negative*, 200 dinyatakan sebagai *true neutral*, dan 187 dinyatakan sebagai *true neutral*. Rata-rata persentase (%) untuk akurasi adalah 63%, presisi 64%, *recall* 63%, dan *F1-Score* sebesar 63%. Dari uraian diatas dapat dilihat hasil yang menunjukkan bahwa SVM mampu mengklasifikasikan sentimen dengan cukup baik, dengan performa terbaik pada prediction positive. Namun, pada kelas prediction negative dengan performa lebih rendah yaitu

166, model mengalami kesulitan dalam membedakan sentimen yang mirip atau netral, yang disebabkan oleh kemiripan kata antar kelas.

4.5 Long Short-Term Memory (LSTM)

Model LSTM dibangun dengan lapisan *Embedding* untuk memetakan kata-kata ke dalam vektor berdimensi tetap, diikuti oleh lapisan LSTM yang menangani ketergantungan jangka panjang dalam data teks, dan lapisan Dense dengan aktivasi softmax untuk menghasilkan probabilitas kelas sentimen seperti Gambar 4.12.

```
# Build LSTM model
model = Sequential()
model.add(Embedding(max_words, 128, input_length=max_len))
model.add(LSTM(196, dropout=0.2, recurrent_dropout=0.2))
model.add(Dense(3, activation='softmax'))
```

Gambar 4. 12 Dense Layer dengan aktivasi softmax

Model dikompilasi menggunakan fungsi loss *categorical_crossentropy* dan optimizer Adam dengan learning rate yang disesuaikan. Proses pelatihan dilakukan dengan batch size 32 dan maksimal 10 epoch, serta menggunakan *EarlyStopping* untuk menghentikan pelatihan jika tidak ada peningkatan pada validation loss seperti Gambar 4.13.

```
# Compile and train the model
from tensorflow.keras.optimizers import Adam
learning_rate = 0.0001
optimizer = Adam(learning_rate=learning_rate)
model.compile(loss='categorical_crossentropy', optimizer=optimizer, metrics=['accuracy'])

early_stopping = EarlyStopping(monitor='val_loss', patience=3, restore_best_weights=True)
history = model.fit(
    X_train, train_labels,
    batch_size=32, epochs=10,
    validation_split=0.1, callbacks=[early_stopping]
)
```

Gambar 4. 13 Hyperparameter LSTM

Setelah pelatihan, model digunakan untuk memprediksi sentimen pada data uji. Hasil prediksi dievaluasi menggunakan *accuracy_score* dan *classification_report*, yang mencakup metrik seperti presisi, *recall*, dan *F1-score* untuk setiap kelas sentimen dapat dilihat pada Tabel 4.7.

Tabel 4. 5 Hasil pengujian LSTM

Kategori Klasifikasi	<i>True Negative</i>	<i>True Neutral</i>	<i>True Positive</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
<i>Prediction Negative</i>	84	41	3	64%	70%	58%
<i>Prediction Neutral</i>	26	99	36	52%	60%	56%
<i>Prediction Positive</i>	18	60	73	69%	52%	60%

Tabel tersebut menunjukkan laporan klasifikasi untuk model LSTM. Untuk kategori Negatif, presisi adalah 64%, *recall* 70%, dan *F1-Score* 67. Untuk kategori Netral, presisi adalah 52%, *recall* 60%, dan *F1-Score* 56%. Untuk kategori Positif, presisi adalah 69%, *recall* 52%, dan *F1-Score* 66%. Rata-rata presisi adalah 62%, *recall* 61%, dan *F1-Score* 61%. Hasil ini menunjukkan bahwa LSTM juga mampu menangkap pola sentimen, model ini bekerja dalam memahami konteks dan urutan kata, sehingga meningkatkan performa dalam mendeteksi sentimen yang lebih kompleks.

4.6 Gated Recurrent Unit (GRU)

Model GRU diimplementasikan menggunakan pustaka Keras dari TensorFlow. Arsitektur model dibangun secara sekuensial, dengan lapisan pertama berupa *Embedding layer* yang mengonversi kata menjadi vektor berdimensi 150. Lapisan berikutnya adalah *GRU layer* dengan 196 unit, yang berfungsi menangkap ketergantungan jangka panjang dalam teks. Lapisan ini juga dilengkapi dengan dropout (0.2) dan recurrent dropout (0.2) untuk mencegah overfitting, serta regularisasi L2 (*kernel_regularizer=l2(0.01)*) untuk meningkatkan kemampuan generalisasi model. Lapisan terakhir adalah *Dense layer* dengan 3 unit dan fungsi aktivasi softmax, yang digunakan untuk mengklasifikasikan teks ke dalam tiga kategori sentimen (positif, netral, atau negatif).

Model ini dikompilasi menggunakan optimizer Adam dengan learning rate 0.0001 dan dilatih menggunakan categorical cross-entropy loss karena tugas yang dilakukan adalah klasifikasi multi-kelas seperti Gambar 4.15.

```
learning_rate = 0.0001
optimizer = Adam(learning_rate=learning_rate)
```

Gambar 4. 14 *Learning rate Adam*

Untuk meningkatkan performa, proses pelatihan menggunakan early stopping, yang menghentikan pelatihan jika nilai validation loss tidak membaik selama tiga epoch berturut-turut. Selain itu, digunakan juga ReduceLROnPlateau callback, yang menurunkan learning rate jika validation loss stagnan, sehingga membantu model berkonvergensi lebih efektif seperti Gambar 4.16.

```
early_stopping = EarlyStopping(monitor='val_loss', patience=3, restore_best_weights=True)
history = model.fit(
    X_train, train_labels,
    batch_size=32, epochs=10,
    validation_split=0.1, callbacks=[early_stopping]
)
```

Gambar 4. 15 *Early Stopping*

Hasil pengujian model GRU menggunakan dataset tweet diperoleh *accuracy*, *precision*, *recall*, F1-Score sebagai berikut:

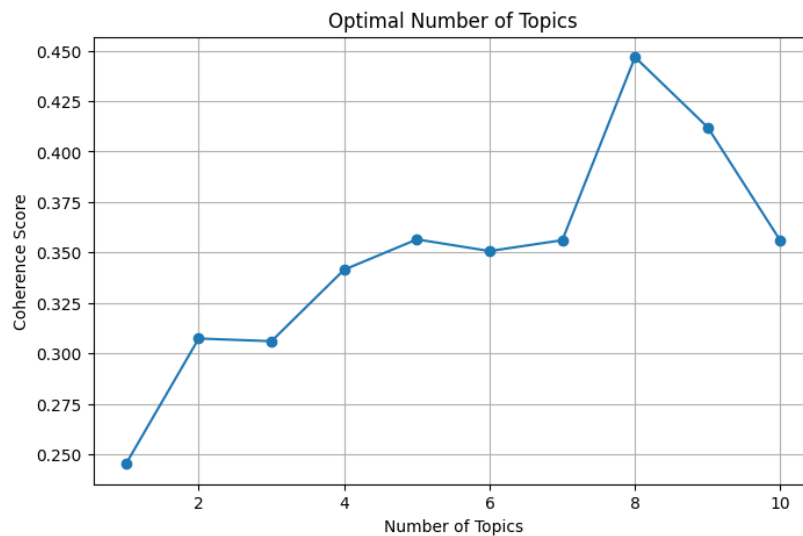
Tabel 4. 6 Hasil Pengujian GRU

Kategori Klasifikasi	<i>True Negative</i>	<i>True Neutral</i>	<i>True Positive</i>	<i>Precision</i>	<i>Recall</i>	F1-Score
<i>Prediction Negative</i>	81	43	4	74%	55%	63%
<i>Prediction Neutral</i>	42	102	17	50%	75%	60%
<i>Prediciton Positive</i>	14	55	82	68%	47%	56%

Tabel tersebut menunjukkan laporan klasifikasi untuk model GRU. Untuk kategori Negatif, presisi adalah 74%, *recall* adalah 55%, dan F1-Score adalah 63%. Untuk kategori Netral, presisi adalah 50%, *recall* adalah 75%, dan F1-Score adalah 60%. Untuk kategori Positif, presisi adalah 68%, *recall* adalah 47%, dan F1-Score adalah 56%. Hasil ini menunjukkan bahwa GRU mampu menangkap pola sentimen dengan baik dan memiliki kinerja yang sebanding dengan LSTM.

4.7 Topic Modelling

Topic Modelling dengan metode LDA mengidentifikasi data berupa teks dan mengatur tema-tema didalam kumpulan dokumen. Penelitian ini melakukan pemodelan topik yang mengidentifikasi pendapat publik tentang Pemilu yang diperoleh melalui proses *crawling tweet* di sosial media X menjadi beberapa topik dengan tujuan mengetahui topik apa saja yang sering menjadi bahasan publik tentang Pemilu. Pencarian jumlah topik yang optimal untuk model LDA dilakukan secara otomatis dengan cara menguji berbagai jumlah topik dalam rentang 1 hingga 10, di mana setiap model dievaluasi berdasarkan *coherence score* untuk menilai kualitas pemisahan topik seperti Gambar 4.18.



Gambar 4. 16 Pencarian Jumlah Optimal Topik

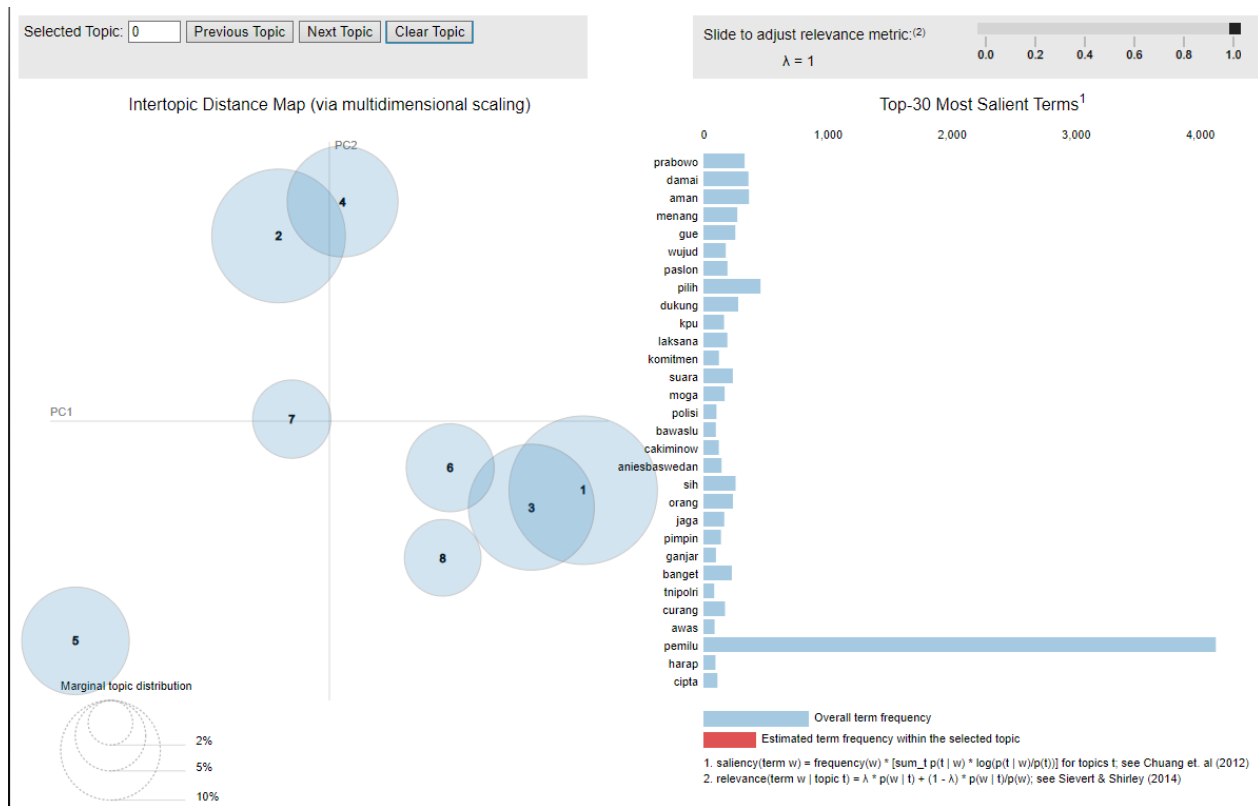
Berdasarkan Gambar 4.18 yang ditampilkan, jumlah topik yang optimal dalam model LDA ditentukan oleh nilai *coherence score* tertinggi, yang terjadi pada 8 topik dengan nilai sekitar 0.45. Seiring bertambahnya jumlah topik, *coherence score* mengalami peningkatan hingga mencapai titik maksimal pada 8 topik, kemudian mulai menurun. Oleh karena itu, dapat disimpulkan bahwa jumlah topik yang paling optimal dalam model ini adalah 8, karena pada titik ini model menghasilkan topik yang lebih koheren dan bermakna dalam analisis teks.

Tabel 4. 7 *Coherence Score*

Topik	<i>Coherence Score</i>
1	0.24525865701428043

2	0.30734171106365327
3	0.30600598239043714
4	0.34144369305529365
5	0.35646970321104077
6	0.3506945989406846
7	0.356070668204013
8	0.44677789397650575
9	0.4120237617631137
10	0.3557905797416736

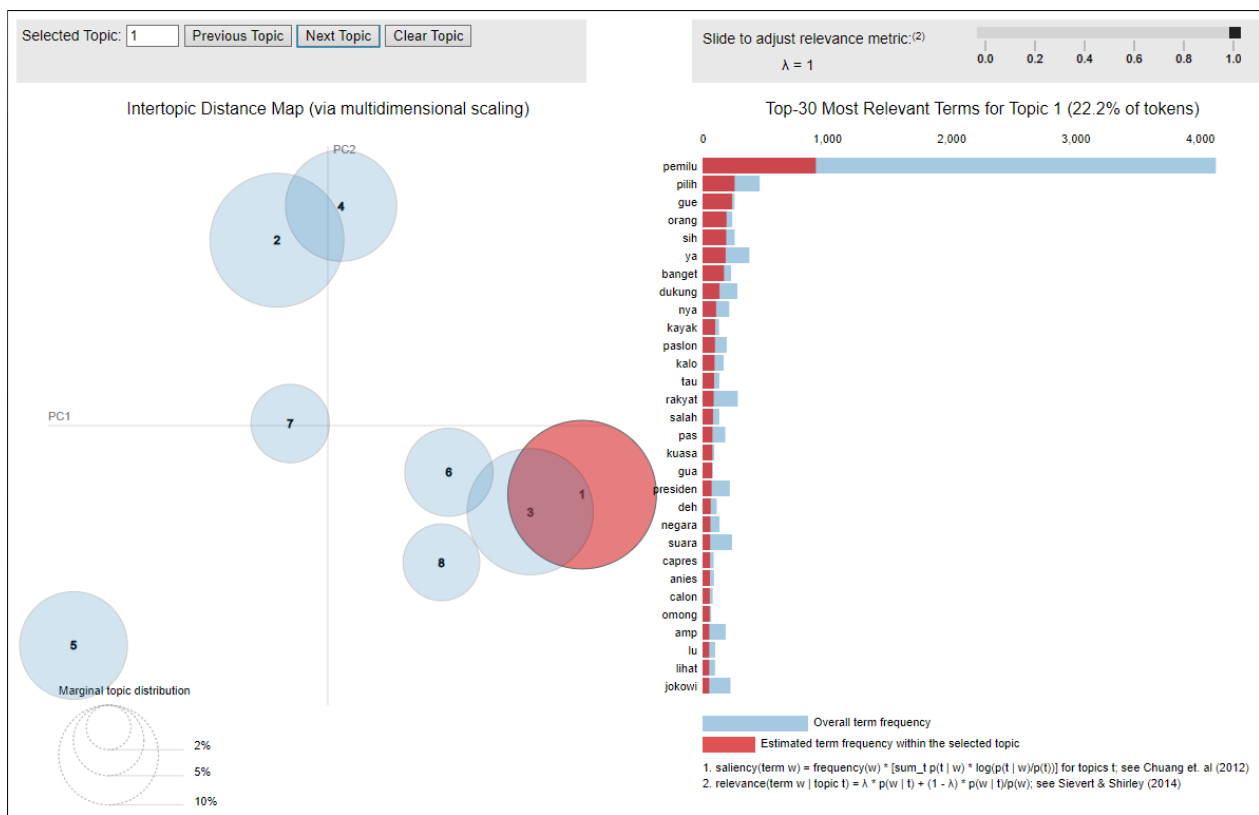
Jumlah topik 8 memiliki *score* terbesar yaitu 0.4467 sehingga akan dijadikan acuan untuk menentukan jumlah *Topic Modeling*. 8 topik tersebut dilakukan visualisasi menggunakan *tools* pyLDAvis. Visualisasi dalam *topic modeling* merupakan salah satu cara paling efektif untuk memahami data. Hasil visualisasi menggunakan pyLDAvis dapat dilihat pada Gambar 4.18.



Gambar 4. 17 Visualisasi Umum *Topic Modeling* dengan pyLDAvis

Gambar 4.19 menampilkan sejumlah *bubble* berisi angka, *bubble* ini mewakili topik tertentu dan menunjukkan jarak antar topik. Visualisasi ini juga menampilkan *Top 30 Most Salient Terms* mencantumkan 30 istilah paling menonjol dari topik yang dipilih, seperti “prabowo”, “damai”, dan “aman”, disertai *bar chart* yang menunjukkan frekuensi secara keseluruhan.

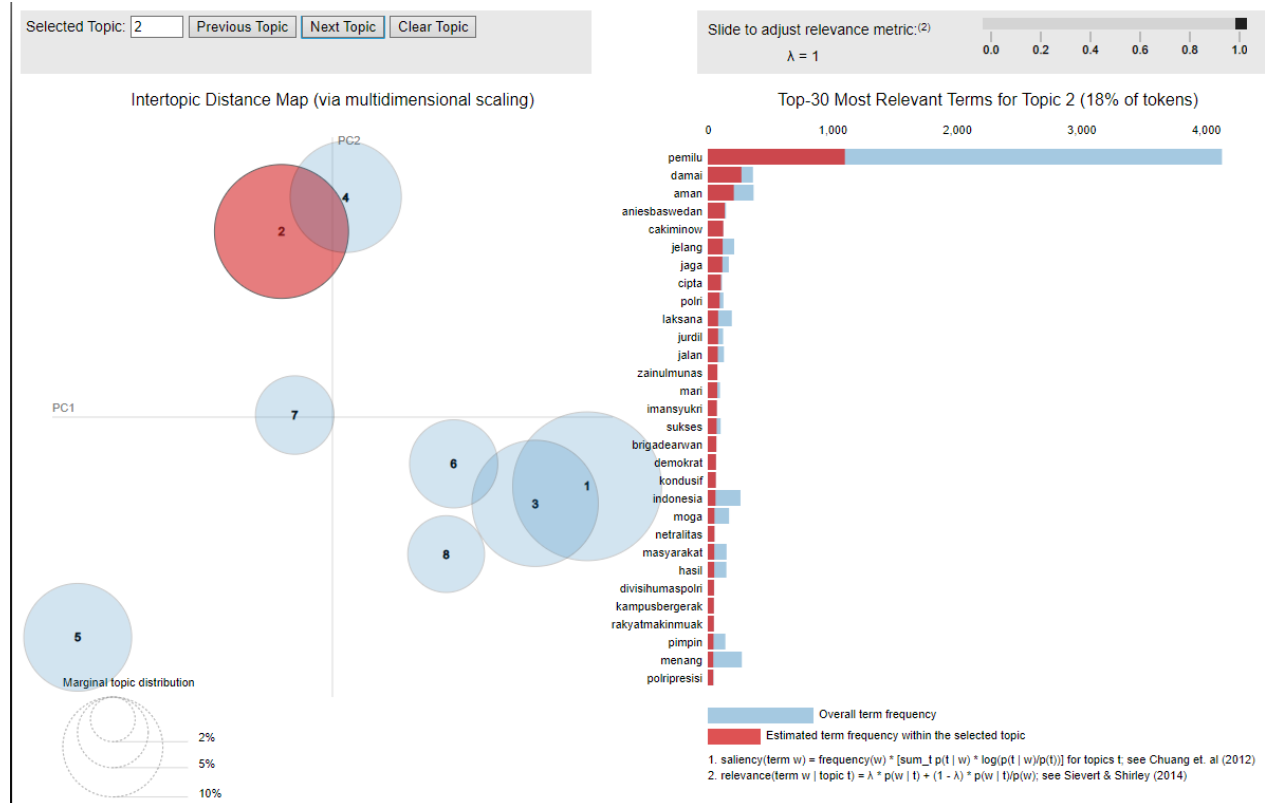
Setelah melakukan visualisasi topik secara keseluruhan dengan pyLDAvis terhadap data *tweet* “Pemilu”. Selanjutnya dilakukan visualisasi topik 1 sampai dengan topik 8 karena setiap *bubble* mewakili sebuah topik.



Gambar 4. 18 Visualisasi Topik 1

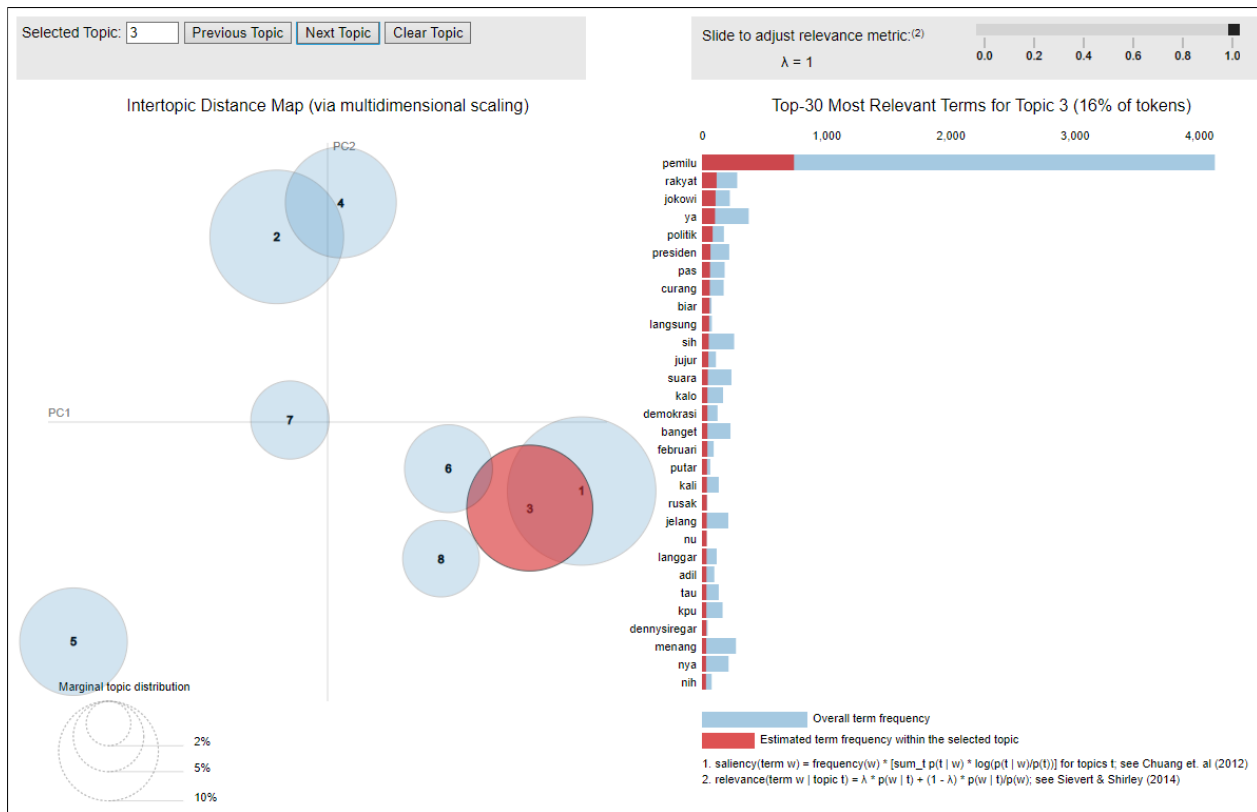
Gambar 4.20 befokus pada topik 1, yang mencakup 22.2% dari total token. Selain itu, bubble yang beririsan dalam visualisasi *Intertopic Distance Map* menunjukkan bahwa topik-topik tersebut memiliki kemiripan atau hubungan tertentu. Irisan ini mengindikasikan bahwa ada istilah atau konsep yang tumpang tindih antara topik-topik tersebut. *Bar chat* merah mewakili frekuensi kata tersebut digunakan dalam topik tersebut, seperti “pemilu” mendekati 1000 frekuensi dalam topik 1. Dalam topik 1 kata dirutkan berdasarkan frekuensi terbanyak adalah “pemilu”, “pilih”,

“gue”, “orang”, “sih”, “ya”, “banget”, “dukung”, “nya”, “kayak”, “paslon”, “kalo”, “tau”, “rakyat”, “salah”, “pas”, “kuasa”, “gua”, “presiden”, “deh”, “negara”, “suara”, “capres”, “anies”, “calon”, “omong”, “lu”, “lihat”, “jokowi.



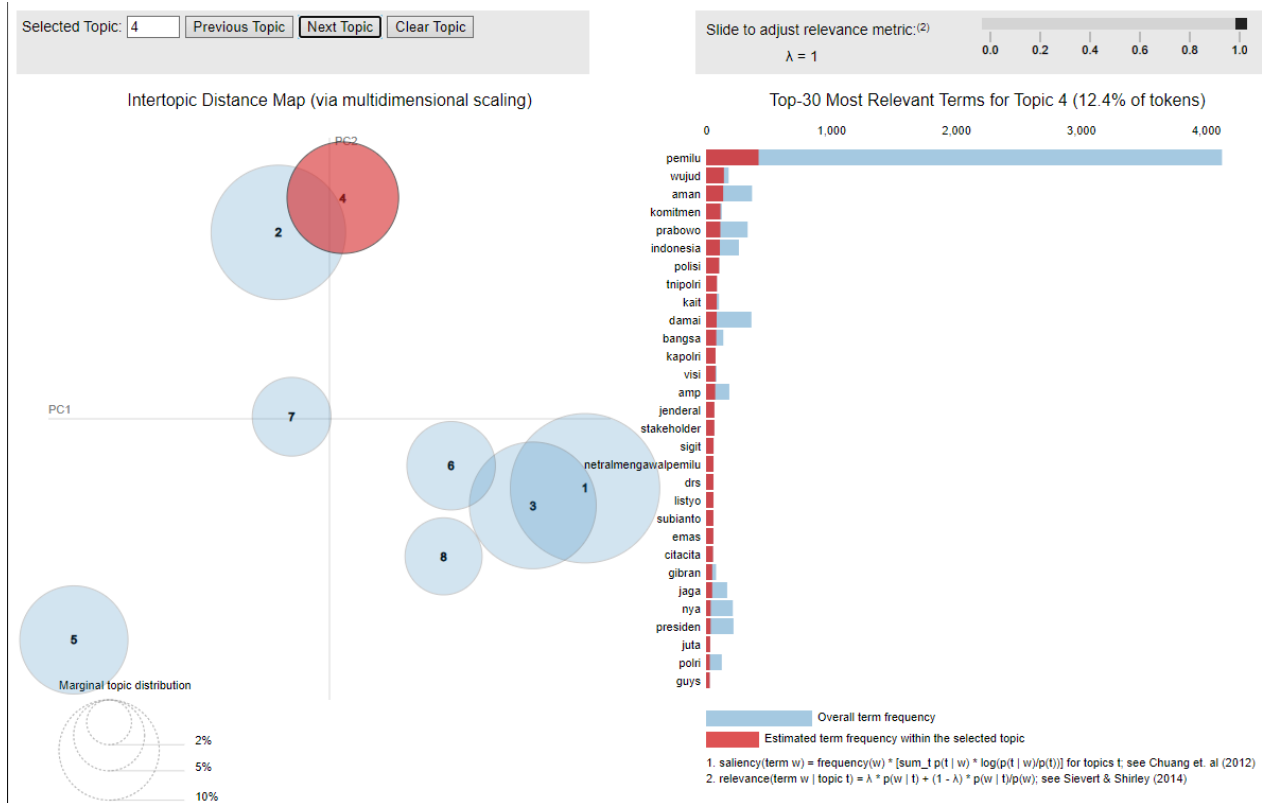
Gambar 4. 19 Visualisasi Topik 2

Gambar 4.21 ini berfokus pada Topik 2, yang mencakup 18% dari total token. Beberapa istilah paling relevan yang muncul dalam topik ini adalah "pemilu", "damai", "aman", "aniesbaswedan", "cakiminow", "jelang", "jaga", "cipta", "polri", "laksana", "jurdil", "jalan", "zainulmunas", "mari", "imansyukuri", "sukses", "brigadearwan", "demokrat", "kondusif", "indonesia", "moga", "netralitas", "masyarakat", "hasil", "divisihumaspolri", "kampusbergerak", "rakyatmakinmuak", "pimpin", "menang", dan "polripresisi". Istilah-istilah ini memberikan Gambaran tentang tema atau fokus dari Topik 2.



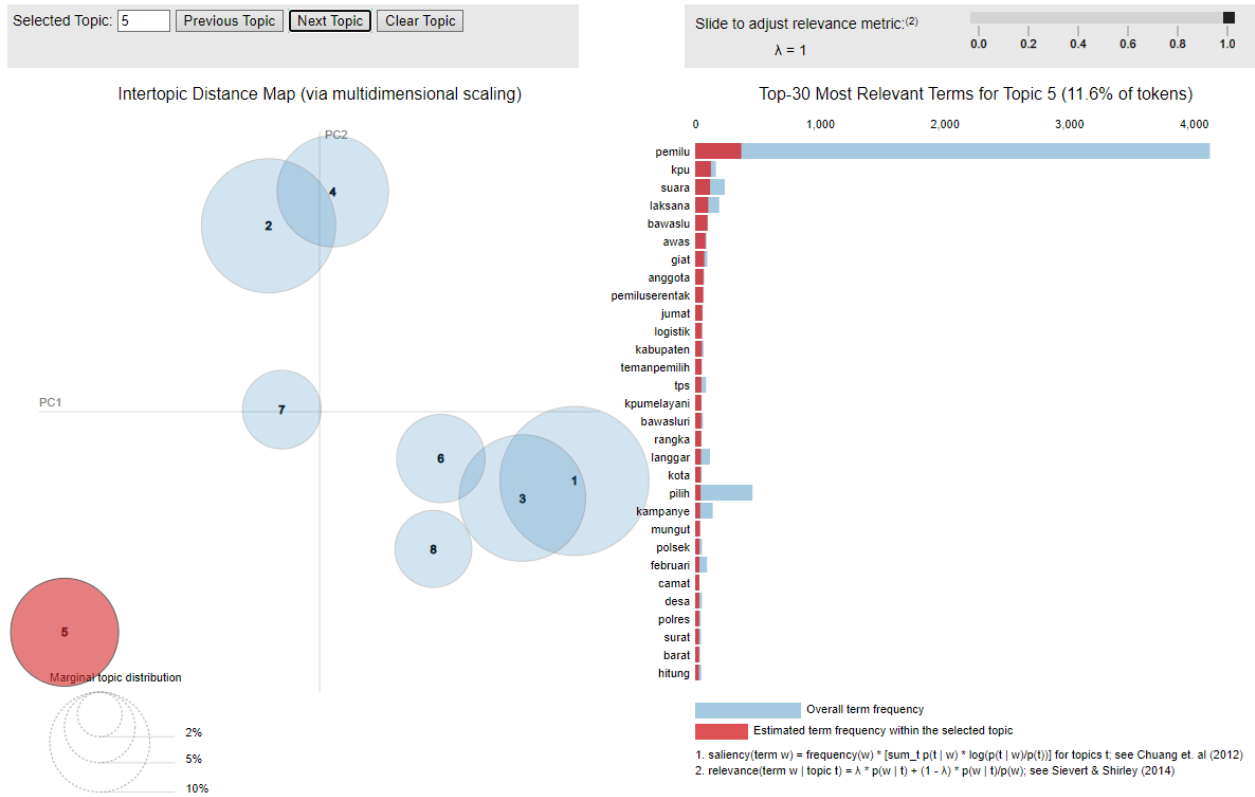
Gambar 4. 20 Visualisasi Topik 3

Gambar 4.22 ini berfokus pada Topik 3, yang mencakup 16% dari total token. Beberapa istilah paling relevan yang muncul dalam topik ini adalah "pemilu", "rakyat", "jokowi", "ya", "politik", "presiden", "pas", "curang", "biar", "langsung", "sih", "jujur", "suara", "kalo", "demokrasi", "banget", "februari", "putar", "kali", "rusak", "jelang", "nu", "langgar", "adil", "tau", "kpu", "dennysiregar", "menang", "nya", dan "nih". Istilah-istilah ini memberikan Gambaran tentang tema atau fokus dari Topik 3.



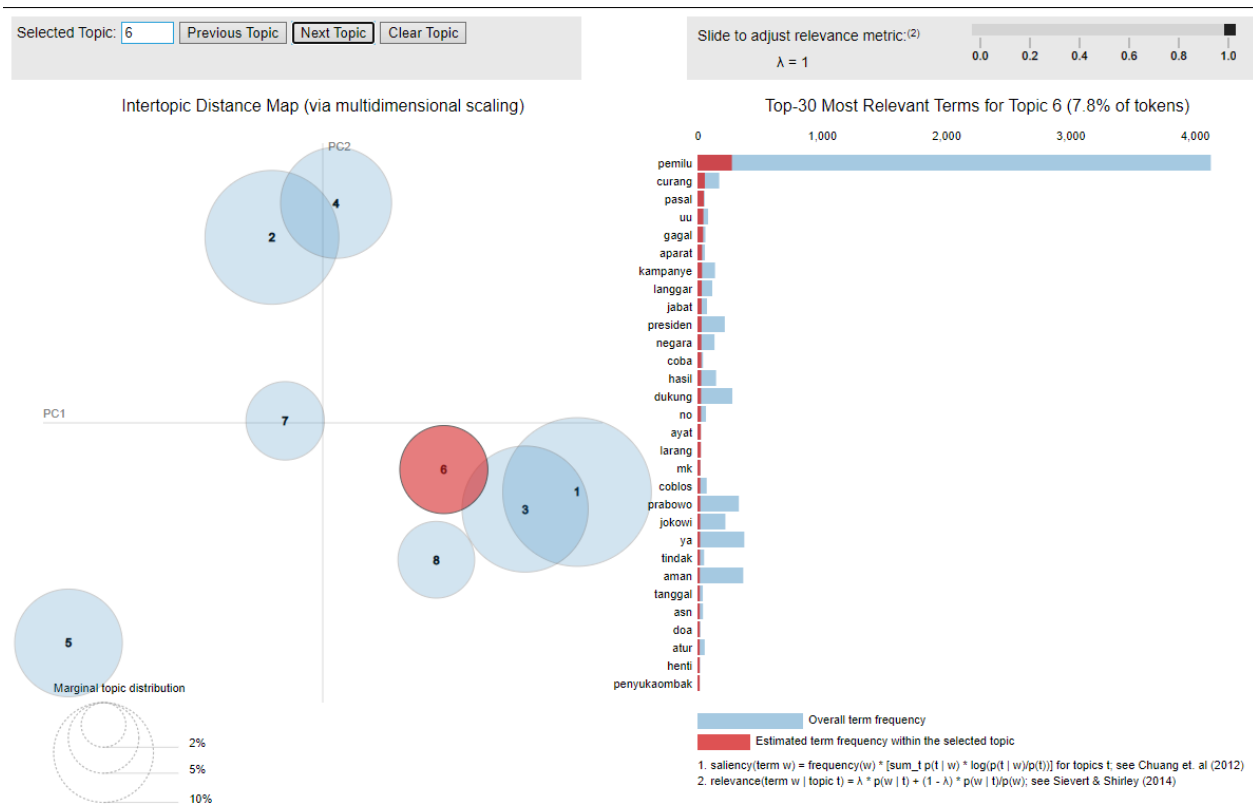
Gambar 4. 21 Visualisasi Topik 4

Gambar 4.23 ini berfokus pada Topik 4, yang mencakup 12.4% dari total token. Beberapa istilah paling relevan yang muncul dalam topik ini adalah "pemilu", "wujud", "aman", "komitmen", "prabowo", "indonesia", "polisi", "tnipolri", "kait", "damai", "bangsa", "kapolri", "visi", "jenderal", "stakeholder", "sigit", "netralmengawalpemilu", "drs", "listyo", "subianto", "emas", "citacita", "gibran", "jaga", "nya", "presiden", "juta", "polri", dan "guys". Istilah-istilah ini memberikan Gambaran tentang tema atau fokus dari Topik 4.



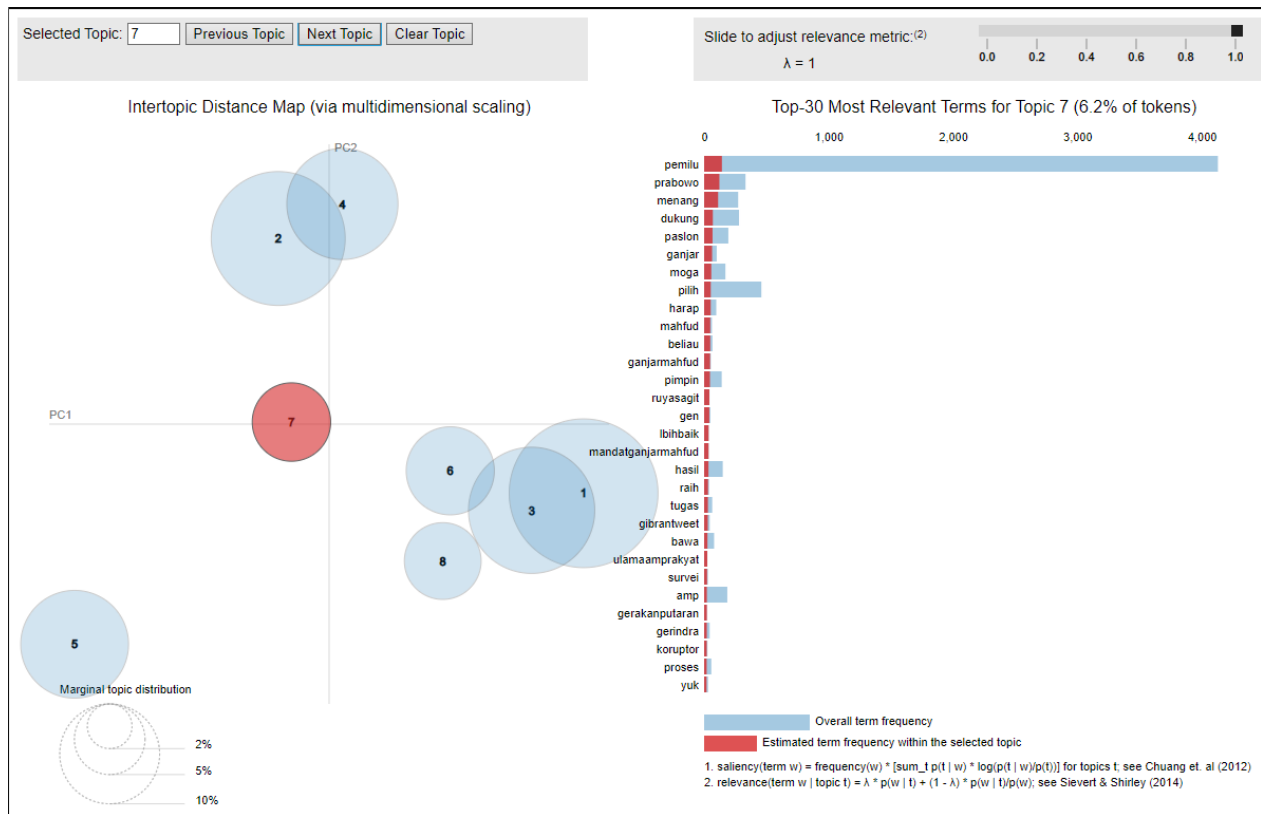
Gambar 4. 22 Visualisasi Topik 5

Gambar 4.24 ini berfokus pada Topik 5, yang mencakup 11.6% dari total token. Beberapa istilah paling relevan yang muncul dalam topik ini adalah "pemilu", "kpu", "suara", "laksana", "bawaslu", "awas", "giat", "anggota", "pemiluserentak", "jumat", "logistik", "kabupaten", "temanpemilih", "tps", "kpumelayani", "bawasluri", "rangka", "kota", "pilih", "kampanye", "mungut", "polsek", "februari", "camat", "desa", "polres", "surat", "barat", dan "hitung". Istilah-istilah ini memberikan Gambaran tentang tema atau fokus dari Topik 5.



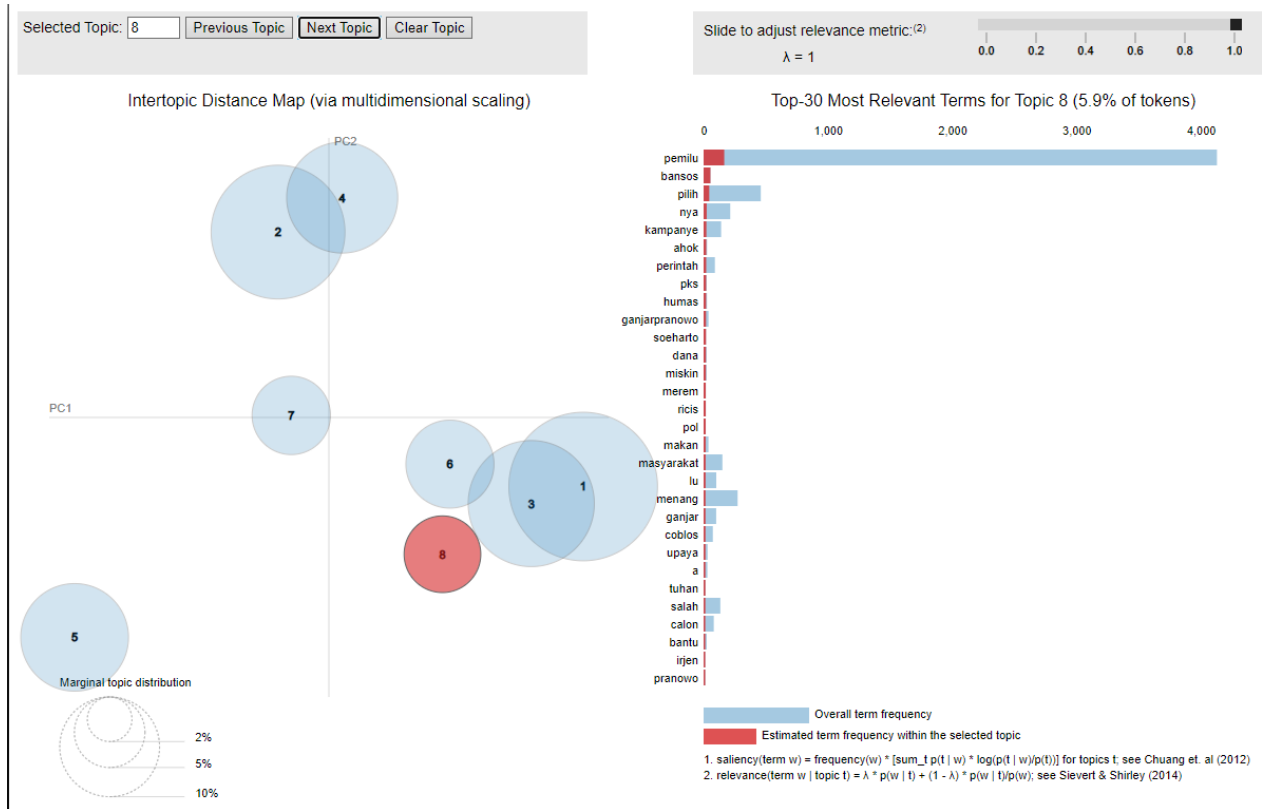
Gambar 4. 23 Visualisasi Topik 6

Gambar 4.25 ini berfokus pada Topik 6, yang mencakup 7.8% dari total token. Beberapa istilah paling relevan yang muncul dalam topik ini adalah "pemilu", "curang", "pasal", "uu", "gagal", "aparat", "kampanye", "langgar", "jabat", "presiden", "negara", "coba", "hasil", "dukung", "no", "ayat", "larang", "mk", "coblos", "prabowo", "jokowi", "ya", "tindak", "aman", "tanggal", "asn", "doa", "atur", "hitung", "henti", dan "penyukaombak". Istilah-istilah ini memberikan Gambaran tentang tema atau fokus dari Topik 6.



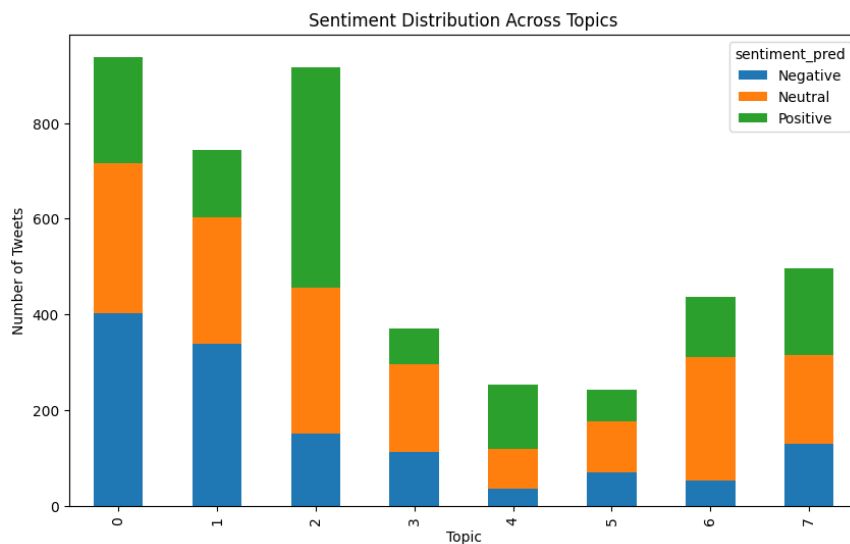
Gambar 4. 24 Visualisasi Topik 7

Gambar 4.26 ini berfokus pada Topik 7, yang mencakup 6.2% dari total token. Beberapa istilah paling relevan yang muncul dalam topik ini adalah "pemilu", "prabowo", "menang", "dukung", "paslon", "ganjar", "moga", "pilih", "harap", "mahfud", "beliau", "ganjarmahfud", "pimpin", "ruyasagit", "gen", "Iblhbaik", "mandatganjarmahfud", "hasil", "raih", "tugas", "gibrantweet", "bawa", "ulamamprakyat", "survei", "gerakanputaran", "gerindra", "koruptor", "proses", dan "yuk". Istilah-istilah ini memberikan Gambaran tentang tema atau fokus dari Topik 7.



Gambar 4. 25 Visualisasi Topik 8

Gambar 4.27 ini berfokus pada Topik 8, yang mencakup 5.9% dari total token. Beberapa istilah paling relevan yang muncul dalam topik ini adalah "pemilu", "bansos", "pilih", "nya", "kampanye", "ahok", "perintah", "pks", "humas", "ganjarpranowo", "soeharto", "dana", "miskin", "merem", "ricis", "pol", "makan", "masyarakat", "lu", "menang", "ganjar", "coblos", "upaya", "tuhan", "salah", "calon", "bantu", "irjen", dan "pranowo". Istilah-istilah ini memberikan Gambaran tentang tema atau fokus dari Topik 8.



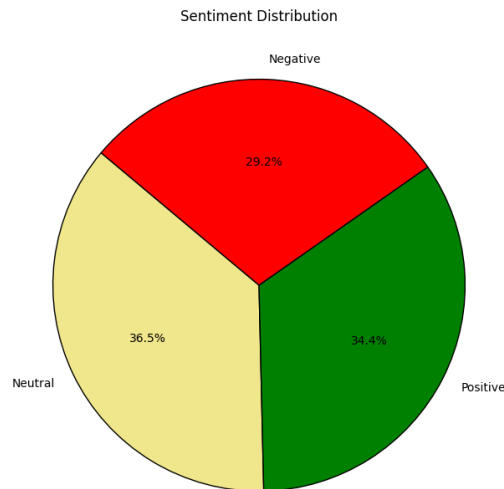
Gambar 4. 26 Distribusi *Sentiment* per Topik

Analisis distribusi sentimen per topik dilakukan melalui tiga tahap terstruktur. Pertama, penentuan topik dominan menggunakan model LDA dengan mengambil probabilitas tertinggi dari distribusi topik setiap dokumen. Kedua, prediksi sentimen dilakukan dengan dataset yang mengklasifikasikan dokumen ke dalam tiga label (negative, neutral, positive). Ketiga, dilakukan agregasi data dengan mengelompokkan dokumen berdasarkan kombinasi topik dominan dan sentimen, kemudian memvisualisasikan hasilnya dalam bentuk stacked bar chart. Hasil analisis ini memberikan Gambaran komprehensif tentang pola sentimen pada berbagai topik pembahasan pemilu.

Grafik 4.26 menunjukkan distribusi sentimen dari tweet-tweet yang dikelompokkan berdasarkan topik tertentu. Pada grafik *stacked bar chart* ini, setiap topik diwakili oleh angka 0 hingga 7 pada sumbu X, sedangkan sumbu Y menunjukkan jumlah tweet dalam setiap topik. Warna biru menunjukkan sentimen negatif, oranye menunjukkan sentimen netral, dan hijau menunjukkan sentimen positif. Topik 0 dan 2 memiliki jumlah tweet tertinggi. Pada topik 2, mayoritas tweet memiliki sentimen positif, sementara topik 0 menunjukkan distribusi sentimen yang lebih seimbang antara negatif dan netral. Topik 1 juga memiliki banyak tweet, didominasi oleh sentimen netral dan positif. Topik 6 dan 7 memperlihatkan distribusi sentimen yang merata, namun tetap dengan dominasi sentimen positif.

Topik 4 dan 5 memiliki jumlah tweet paling sedikit, namun pada topik 4, sentimen positif lebih mendominasi. Sementara itu, topik 3 menunjukkan proporsi yang lebih besar untuk sentimen netral dibandingkan sentimen lainnya.

4.5 Pembahasan

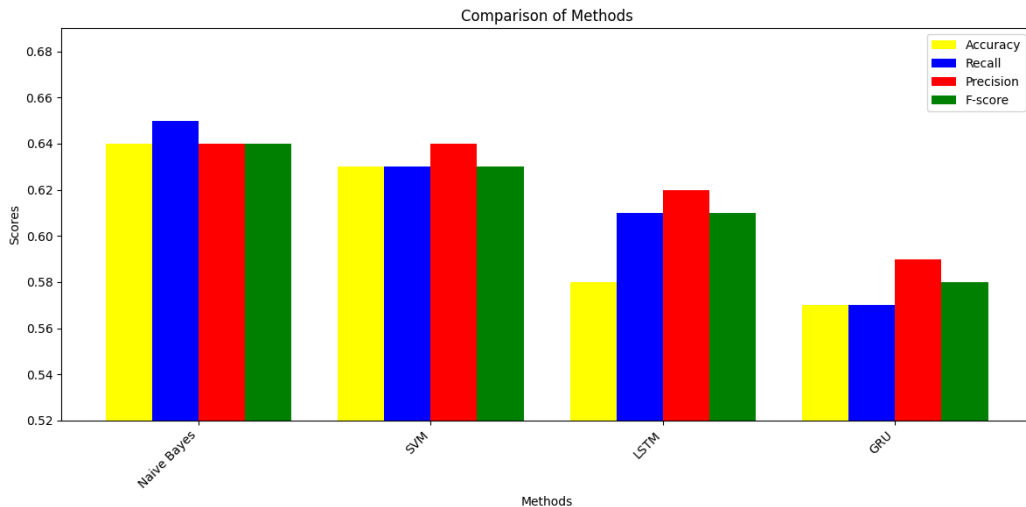


Gambar 4. 27 Pie Chart Distribusi Sentiment

Berdasarkan proses penelitian yang telah dilakukan, dari total tweet yang diperoleh melalui proses *crawling data* diperoleh 4398 tweet dan distribusi *sentiment positive* 34%, *neutral* 36%, dan *negative* 29%. Distribusi ini cukup baik karena *imbalance* data tidak terlalu besar.

Tabel 4. 8 Tabel Perbandingan Model *Sentiment Analysis*

Metode	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Naïve Bayes	64%	64%	65%	64%
SVM	63%	64%	63%	63%
LSTM	58%	62%	61%	61%
GRU	57%	59%	57%	58%



Gambar 4. 28 *Bar Chart* Perbandingan Model *Sentiment Analysis*

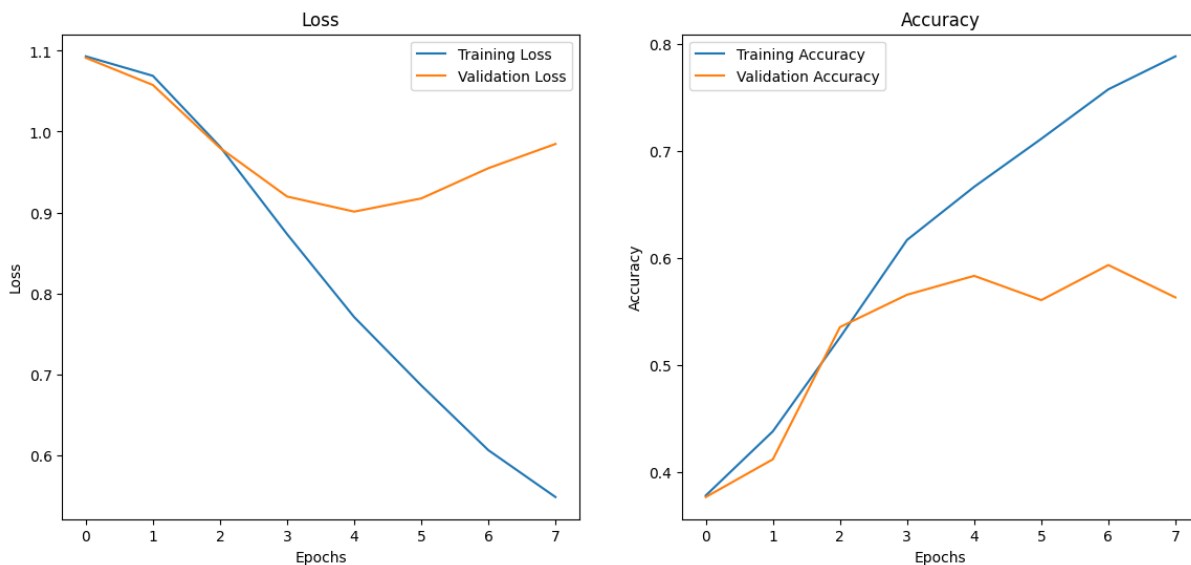
Berdasarkan Tabel komparasi dan *bar chart* performa berbagai metode dalam analisis sentimen, Naïve Bayes menunjukkan kinerja terbaik dengan akurasi sebesar 64%, *precision* 64%, *recall* 65%, dan *F1-score* 64%. SVM berada di posisi kedua dengan akurasi 63%, *precision* 64%, *recall* 63%, dan *F1-score* 63%, yang cukup kompetitif dengan Naïve Bayes.

Sementara itu, metode berbasis *deep learning*, yaitu LSTM dan GRU, memiliki performa yang lebih rendah. LSTM memperoleh akurasi 58%, *precision* 62%, *recall* 61%, dan *F1-score* 61%, sedangkan GRU memiliki akurasi terendah sebesar 57%, dengan *precision* 59%, *recall* 57%, dan *F1-score* 58%. Perbedaan akurasi antar model dipengaruhi oleh karakteristik unik masing-masing pendekatan. Naive Bayes unggul dalam menangani data teks berdimensi tinggi berkat asumsi independensi fiturnya yang cocok untuk distribusi kata tertentu, serta kemampuan generalisasi yang baik pada dataset berukuran sedang. SVM menunjukkan performa stabil karena kemampuannya menemukan hyperplane optimal untuk pemisahan kelas dan ketahanan terhadap overfitting melalui parameter regularisasi.

Di sisi lain, model *deep learning* (LSTM/GRU) memiliki beberapa keterbatasan: (1) memerlukan data dalam jumlah besar untuk mencapai performa optimal, (2) hyperparameter yang kompleks membutuhkan tuning intensif, (3) rentan overfitting pada dataset terbatas, dan (4) biaya komputasi tinggi yang tidak sebanding dengan peningkatan akurasi. Hal ini menjelaskan mengapa model tradisional seperti Naive Bayes dan SVM lebih unggul untuk kasus klasifikasi pada dataset berukuran sedang sesuai dengan penelitian (Zhang & Wallace, 2015) yang

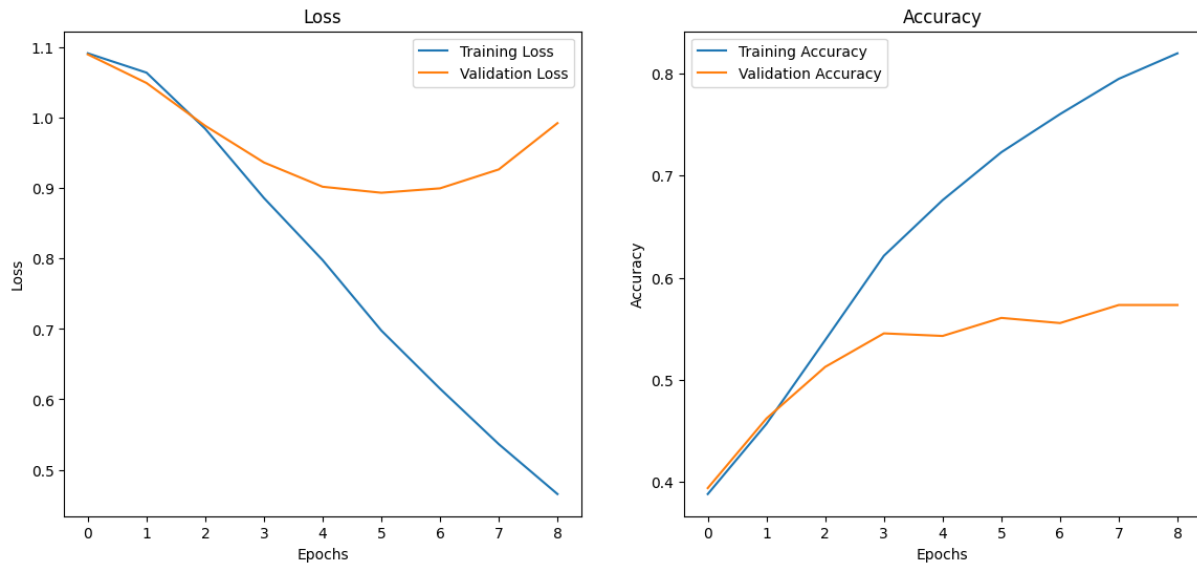
menunjukkan bahwa untuk tugas klasifikasi teks pada dataset berukuran sedang (<100.000 sampel), model tradisional seringkali mengungguli arsitektur *deep learning* dalam hal efisiensi komputasi dan akurasi.

Secara keseluruhan, jika prioritasnya adalah mencapai kinerja model yang konsisten dan seimbang di semua metrik, maka metode Naïve Bayes merupakan pilihan terbaik. Namun, jika terdapat potensi untuk meningkatkan performa dengan data yang lebih banyak atau teknik tuning lanjutan, metode *deep learning* seperti LSTM dan GRU mungkin masih memiliki peluang untuk dioptimalkan lebih lanjut.



Gambar 4. 29 Grafik *Loss* dan *Accuracy* untuk model LSTM

Dari Gambar 4.14, grafik menunjukkan kinerja model selama 7 epoch dalam hal kehilangan (*loss*) dan akurasi untuk data pelatihan dan validasi. Kehilangan pelatihan secara bertahap menurun, menunjukkan pembelajaran yang efektif, sementara kehilangan validasi awalnya menurun tetapi mulai meningkat setelah epoch ke-4, yang menandakan kemungkinan terjadinya *overfitting*. Demikian juga, akurasi pelatihan meningkat secara konsisten, mencapai sekitar 80% pada epoch ke-7, sedangkan akurasi validasi meningkat hingga epoch ke-3 sebelum berfluktuasi dan stabil di sekitar 60%. Ini menunjukkan bahwa meskipun model berkinerja baik pada data pelatihan, kemampuannya untuk menggeneralisasi ke data yang belum terlihat terbatas setelah titik tertentu.



Gambar 4. 30 Grafik *Loss* dan *Accuracy* untuk model GRU

Grafik menunjukkan peningkatan kinerja selama 8 epoch, yang ditandai dengan penurunan nilai kehilangan (*loss*) pada data pelatihan dan validasi, serta peningkatan akurasi pada kedua jenis data tersebut. Namun, terdapat sedikit indikasi *overfitting* yang terlihat dari nilai kehilangan validasi yang sedikit lebih tinggi dibandingkan dengan kehilangan pelatihan.

Tabel 4. 9 *Accuracy Rates per Epoch*

<i>Classification Categories</i>	Epoch (%)									<i>Runtime</i>
	1 st Epoch	2 nd Epoch	3 rd Epoch	4 th Epoch	5 th Epoch	6 th Epoch	7 th Epoch	8 th Epoch	9 th Epoch	
	<i>Accuracy</i>	<i>Accuracy</i>	<i>Accuracy</i>	<i>Accuracy</i>	<i>Accuracy</i>	<i>Accuracy</i>	<i>Accuracy</i>	<i>Accuracy</i>	<i>Accuracy</i>	
LSTM	0.376	0.411	0.535	0.565	0.583	0.56	0.593	0.563	-	3m2s
GRU	0.393	0.462	0.512	0.545	0.542	0.56	0.555	0.573	0.573	2m5s

Berdasarkan Tabel tersebut, terlihat bahwa model LSTM dan GRU diuji dengan hyperparameter yang ditetapkan untuk 10 epoch, namun diterapkan *early stopping* untuk mencegah *overfitting*. *Early stopping* menghentikan pelatihan lebih awal ketika performa pada data validasi tidak lagi meningkat secara signifikan. Pada model LSTM, akurasi awal dimulai di angka 0.376 dan meningkat secara konsisten hingga mencapai puncaknya di epoch ke-7 dengan akurasi 0.593. Namun, pada epoch ke-8 akurasi menurun menjadi 0.563, yang memicu *early stopping*, sehingga pelatihan tidak dilanjutkan ke epoch ke-9. Sementara itu, model GRU memulai dengan akurasi 0.393 dan menunjukkan peningkatan stabil hingga mencapai akurasi

tertinggi 0.573 pada epoch ke-9, di mana *early stopping* kemudian diterapkan. Dari segi *runtime*, GRU memiliki waktu eksekusi yang lebih cepat (2 menit 5 detik) dibandingkan LSTM (3 menit 2 detik), yang menunjukkan bahwa GRU lebih efisien dalam proses pelatihan. Dengan demikian, meskipun LSTM mencapai nilai tertinggi yang sedikit lebih baik dalam beberapa epoch, GRU menawarkan keuntungan dalam kecepatan pelatihan dengan performa yang relatif kompetitif.

Tabel 4. 10 Hasil *Topic Modeling*

Topik	Score	Kesimpulan
1	0.069*"pemilu" + 0.020*"pilih" + 0.018*"gue" + 0.015*"orang" + 0.014*"sih" + 0.014*"ya" + 0.013*"banget" + 0.010*"dukung" + 0.008*"nya" + 0.008*"kayak"	1. Opini Publik tentang Pemilu 2. Diskusi Informal Masyarakat Umum 3. Opini Pribadi Pemilu
2	0.078*"pemilu" + 0.012*"rakyat" + 0.012*"jokowi" + 0.011*"ya" + 0.009*"politik" + 0.007*"presiden" + 0.007*"pas" + 0.007*"curang" + 0.006*"biar" + 0.006*"langsung"	1. Peran Jokowi dalam Pemilu 2. Isu Kecurangan dan Politik 3. Isu Pemilu & Tokoh Nasional
3	0.103*"pemilu" + 0.025*"damai" + 0.020*"aman" + 0.013*"aniesbaswedan" + 0.012*"cakiminow" + 0.011*"jelang" + 0.011*"jaga" + 0.010*"cipta" + 0.009*"polri" + 0.008*"laksana"	1. Perdamaian dan Keamanan dalam Pemilu 2. Harapan Pemilu Damai 3. Pemilu Damai & Kandidat
4	0.060*"pemilu" + 0.013*"curang" + 0.011*"pasal" + 0.010*"uu" + 0.010*"gagal" + 0.008*"aparatus" + 0.008*"kampanye" + 0.007*"langgar" + 0.007*"jabat" + 0.007*"presiden"	1. Isu Kecurangan dan Penyalahgunaan Kekuasaan dalam Pemilu 2. Kritik Prosedural Pemilu 3. Potensi Kecurangan & Pelanggaran Pemilu

5	0.039*"pemilu" + 0.034*"prabowo" + 0.031*"menang" + 0.019*"dukung" + 0.018*"paslon" + 0.018*"ganjar" + 0.016*"moga" + 0.015*"pilih" + 0.014*"harap" + 0.014*"mahfud"	1. Dukungan terhadap Pasangan Calon Presiden 2. Dukugan untuk Paslon 3. Dukungan Kandidat
6	0.047*"pemilu" + 0.016*"bansos" + 0.013*"pilih" + 0.007*"nya" + 0.006*"kampanye" + 0.006*"ahok" + 0.006*"perintah" + 0.006*"pks" + 0.006*"humas" + 0.006*"ganjarpranowo"	1. Kampanye Pemilu dan Strategi Politik 2. Isu Bansos dan Kampanye 3. Bansos & Politik Pemilu
7	0.054*"pemilu" + 0.018*"kpu" + 0.017*"suara" + 0.015*"laksana" + 0.014*"bawaslu" + 0.012*"awas" + 0.011*"giat" + 0.010*"anggota" + 0.009*"pemisuserentak" + 0.008*"jumat"	1. Peran KPU dan Bawaslu dalam Pemilu 2. Peran Institusi Penyelenggara 3. Penyelenggaraan Pemilu
8	0.057*"pemilu" + 0.019*"wujud" + 0.019*"aman" + 0.016*"komitmen" + 0.016*"prabowo" + 0.015*"indonesia" + 0.014*"polisi" + 0.012*"tnipolri" + 0.012*"kait" + 0.012*"damai"	1. Peran TNI-Polri dalam Menjaga Keamanan Pemilu 2. Komitmen Keamanan Pemilu 3. Pemilu Aman & Damai

Berdasarkan hasil *topic modeling*, kesimpulan diambil dengan menganalisis kata-kata kunci yang muncul dalam setiap topik serta melihat hubungan antar topik. Setiap topik memiliki daftar kata dengan skor tertentu, yang menunjukkan sejauh mana kata tersebut berkontribusi dalam pembentukan topik. Misalnya, Topik 1 memiliki kata seperti pilih, dukung, dan gue, yang menunjukkan diskusi tentang opini publik terhadap pemilu, sementara Topik 5 berisi kata prabowo, menang, ganjar, dan paslon, yang menandakan perbincangan terkait dukungan terhadap pasangan calon presiden.

Pada hasil visualisasi dengan pyLDAvis seperti Gambar 4.17 terdapat beberapa topik yang saling beririsan, irisan ini terjadi apabila topik-topik tersebut memiliki *terms* atau kata yang dibahas di topik yang bersamaan terkait Pemilu. Topik 2 dan Topik 4, misalnya, sama-sama mengandung kata pemilu, politik, presiden, dan curang, yang menunjukkan bahwa diskusi politik nasional sering kali bersinggungan dengan isu kecurangan dan hukum pemilu. Sementara itu, Topik 3 dan Topik 8 yang memiliki kata aman, laksana, damai, dan komitmen mengindikasikan adanya pembahasan mengenai keamanan pemilu dan harapan terhadap stabilitas pasca pemilu.

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penetapan rumusan masalah, dan dengan merujuk kepada metodologi penelitian yang menentukan hasil dan diikuti dengan pembahasan, maka penelitian ini merumuskan beberapa kesimpulan sebagai berikut:

- a. Naïve Bayes unggul dengan akurasi 64% dalam klasifikasi sentimen, diikuti SVM (63%), sementara LSTM dan GRU kurang efektif (58% dan 57%) karena *overfitting* pada dataset kecil.
- b. Metode *machine learning* tradisional lebih efektif dibanding *deep learning* pada dataset terbatas. Kesimpulan ini diperkuat hasil studi sebelumnya (Z. Huang, 2024), dimana *machine learning* optimal untuk skenario di mana data terbatas, karena tidak memerlukan data yang luas seperti yang dibutuhkan model *deep learning* untuk memberikan performa yang baik.
- c. Sentimen netral paling sulit diprediksi, menunjukkan ambiguitas yang lebih tinggi dibanding sentimen positif atau negatif. Kesimpulan ini diperkuat hasil studi sebelumnya (Valdivia et al., 2018), (Koppel & Schler, 2006), dimana sentiment netral sering diabaikan dalam analisis sentimen karena sifatnya yang ambigu dan kurangnya indikator yang jelas, sehingga sulit untuk diklasifikasikan dengan akurat.
- d. Sentimen publik terhadap Pemilu 2024 cukup beragam, dengan dominasi sentimen *positive* dan *neutral*.
- e. LDA mengidentifikasi 8-topik utama terkait Pemilu 2024, seperti dukungan calon, keamanan pemilu, dan isu kecurangan, dengan coherence score tertinggi 0.4467.
- f. Distribusi sentimen per topik menunjukkan bahwa sentimen positif mendominasi sebagian besar topik, terutama yang terkait dukungan terhadap calon dan keamanan pemilu, sementara sentimen negatif lebih terlihat pada topik yang membahas isu kecurangan atau masalah teknis pemilu.

Berdasarkan analisis terhadap tweet bahasa Indonesia terkait Pemilu 2024, penelitian ini menemukan bahwa metode *machine learning* tradisional—khususnya Naïve Bayes dengan akurasi 64%—ternyata lebih unggul dibanding pendekatan *deep learning* seperti LSTM (58%) dan GRU (57%) dalam klasifikasi sentimen. Hasil ini menunjukkan bahwa untuk dataset terbatas

(4.500 tweet), algoritma sederhana seperti Naïve Bayes dan SVM (63%) lebih efektif karena efisiensi komputasinya dan kemampuan generalisasi yang baik, sementara model kompleks seperti LSTM/GRU cenderung mengalami overfitting tanpa data yang memadai.

Selain itu, sentimen netral terbukti paling sulit diprediksi akibat sifatnya yang ambigu, berbeda dengan sentimen positif (34%) dan negatif (29%) yang lebih jelas pola bahasanya. Melalui pemodelan topik dengan LDA, teridentifikasi 8 tema dominan dalam diskusi publik, seperti dukungan terhadap calon, isu kecurangan, dan peran KPU/Bawaslu, dengan tingkat koherensi tertinggi (0.4467). Temuan ini memperkuat bahwa dinamika opini masyarakat tentang Pemilu 2024 relatif berimbang, didominasi oleh sentimen positif dan netral (36%), tanpa polarisasi ekstrem.

Secara praktis, penelitian ini merekomendasikan penggunaan Naïve Bayes atau SVM untuk analisis sentimen tweet berbahasa Indonesia dalam skala kecil, sementara deep learning perlu dioptimalkan dengan data lebih besar atau pre-trained model seperti IndoBERT. Implikasinya, pemerintah dan peneliti dapat memanfaatkan temuan ini untuk merancang strategi komunikasi berbasis data yang lebih tepat sasaran.

5.2 Saran

Berdasarkan hasil dan pembahasan penelitian dan merujuk kepada kesimpulan penelitian, dapat ditemukan beberapa penelitian lanjutan yang masih dapat dilakukan penelitian berikutnya, oleh karena ini berikut beberapa saran yang diajukan:

- a. Sentiment *analysis* dengan metode yang diterapkan pada penelitian ini dapat digunakan untuk data tweet yang lagi *trending* saat ini ataupun di waktu mendatang
- b. Metodologi dengan penggunaan *deep learning* ataupun *machine learning* masih dapat dikembangkan selain menggunakan Naive Bayes, SVM, LSTM dan GRU.

DAFTAR PUSTAKA

- Aakash, Gupta, S., & Noliya, A. (2024). URL-Based Sentiment Analysis of Product Reviews Using LSTM and GRU. *Procedia Computer Science*, 235(2023), 1814–1823. <https://doi.org/10.1016/j.procs.2024.04.172>
- Abid, F., Li, C., & Alam, M. (2020). Multi-source social media data sentiment analysis using bidirectional recurrent convolutional neural networks. *Computer Communications*, 157(December 2019), 102–115. <https://doi.org/10.1016/j.comcom.2020.04.002>
- Adiga, N., Govind, D., & Prasanna, S. M. (2014). Significance of epoch identification accuracy for prosody modification. *2014 International Conference on Signal Processing and Communications (SPCOM)*, 1–6. <https://doi.org/10.1109/SPCOM.2014.6984007>
- Akbar, M. Z., Ikhsan, M., & Zufria, I. (2024). Sentiment Analysis of the Indriver Online Ojek Application using the Naïve Bayes Classifier Method. *Sinkron*, 8(3), 2037–2048. <https://doi.org/10.33395/sinkron.v8i3.13934>
- AL-Smadi, M., Hammad, M. M., Al-Zboon, S. A., AL-Tawalbeh, S., & Cambria, E. (2023). Gated recurrent unit with multilingual universal sentence encoder for Arabic aspect-based sentiment analysis. *Knowledge-Based Systems*, 261. <https://doi.org/10.1016/j.knosys.2021.107540>
- Alruily, M. (2023). Sentiment analysis for predicting stress among workers and classification utilizing CNN: Unveiling the mechanism. *Alexandria Engineering Journal*, 81(July), 360–370. <https://doi.org/10.1016/j.aej.2023.09.040>
- Arnold, C., Biedebach, L., Küpfer, A., & Neunhoeffler, M. (2024). The role of hyperparameters in machine learning models and how to tune them. *Political Science Research and Methods*. <https://doi.org/10.1017/psrm.2023.61>
- Awotunde, J. B., Misra, S., Katta, V., & Adebayo, O. C. (2023). An Ensemble-Based Hotel Reviews System Using Naive Bayes Classifier. *CMES - Computer Modeling in Engineering and Sciences*, 137(1), 131–154. <https://doi.org/10.32604/cmcs.2023.026812>
- Belete, D., & Huchaiah, M. (2021). Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results. *International Journal of Computers and Applications*, 44, 875–886. <https://doi.org/10.1080/1206212X.2021.1974663>
- Birjali, M., Kasri, M., & Hssane, A. (2021). A comprehensive survey on sentiment analysis:

- Approaches, challenges and trends. *Knowl. Based Syst.*, 226, 107134. <https://doi.org/10.1016/j.knosys.2021.107134>
- Bischl, B., Binder, M., Lang, M., Pielok, T., Richter, J., Coors, S., Thomas, J., Ullmann, T., Becker, M., Boulesteix, A., Deng, D., & Lindauer, M. (2021). Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 13. <https://doi.org/10.1002/widm.1484>
- Blei, D. M., Ng, A. Y., & Jordan, M. T. (2002). Latent dirichlet allocation. *Advances in Neural Information Processing Systems, January 2001*, 992–1022.
- Camacho-Collados, J., & Pilehvar, M. T. (2018). On the Role of Text *Preprocessing* in Neural Network Architectures: An Evaluation Study on Text Categorization and Sentiment Analysis. *EMNLP 2018 - 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, Proceedings of the 1st Workshop*, 40–46. <https://doi.org/10.18653/v1/w18-5406>
- Chai, C. P. (2023). Comparison of text *Preprocessing* methods. *Natural Language Engineering*, 29(3), 509–553. <https://doi.org/DOI: 10.1017/S1351324922000213>
- D’Ascenzo, F., Rocchi, A., Iandolo, F., & Vito, P. (2024). Evolutionary impacts of artificial intelligence in healthcare managerial literature. A ten-year bibliometric and topic modeling review. *Sustainable Futures*, 7(March), 100198. <https://doi.org/10.1016/j.sfr.2024.100198>
- Damayanti, L., & Lhaksana, K. M. (2024). Sentiment Analysis of the 2024 Indonesia Presidential Election on Twitter. *Sinkron*, 8(2), 938–946. <https://doi.org/10.33395/sinkron.v8i2.13379>
- DiPietro, R., & Hager, G. D. (2019). Deep learning: RNNs and LSTM. In *Handbook of Medical Image Computing and Computer Assisted Intervention*. Elsevier Inc. <https://doi.org/10.1016/B978-0-12-816176-0.00026-0>
- Fitri, V. A., Andreswari, R., & Hasibuan, M. A. (2019). Sentiment analysis of social media Twitter with case of Anti-LGBT campaign in Indonesia using Naïve Bayes, decision tree, and random forest algorithm. *Procedia Computer Science*, 161, 765–772. <https://doi.org/10.1016/j.procs.2019.11.181>
- Gao, S., Huang, Y., Zhang, S., Han, J., Wang, G., Zhang, M., & Lin, Q. (2020). Short-term runoff prediction with GRU and LSTM networks without requiring time step optimization

- during sample generation. *Journal of Hydrology*, 589, 125188.
<https://doi.org/10.1016/j.jhydrol.2020.125188>
- Gunawan, W., Devianto, Y., & Sari, A. P. (2024). Imbalanced Data NearMiss for Comparison of SVM and Naive Bayes Algorithms. *Computer Engineering and Applications Journal*.
<https://doi.org/10.18495/comengapp.v13i03.485>
- Hasanati, N., Aini, Q., & Nuri, A. (2022). Implementation of Support Vector Machine with Lexicon Based for Sentiment Analysis on Twitter. *2022 10th International Conference on Cyber and IT Service Management (CITSM)*, 1–4.
<https://doi.org/10.1109/CITSM56380.2022.9935887>
- Heydarian, M., Doyle, T., & Samavi, R. (2022). MLCM: Multi-Label Confusion Matrix. *IEEE Access*, 10, 19083–19095. <https://doi.org/10.1109/access.2022.3151048>
- Hickman, L., Thapa, S., Tay, L., Cao, M., & Srinivasan, P. (2022). Text Preprocessing for Text Mining in Organizational Research: Review and Recommendations. *Organizational Research Methods*, 25(1), 114–146. <https://doi.org/10.1177/1094428120971683>
- Hindarto, D. (2023). Comparison of RNN Architectures and Non-RNN Architectures in Sentiment Analysis. *Sinkron*. <https://doi.org/10.33395/sinkron.v8i4.13048>
- Huang, Q. (2023). Sentiment analysis for social media using SVM classifier of machine learning. *Applied and Computational Engineering*. <https://doi.org/10.54254/2755-2721/4/20230354>
- Huang, Z. (2024). A Research on Image Recognition and Classification Based on Traditional Machine Learning and Deep Learning. *Transactions on Computer Science and Intelligent Systems Research*. <https://doi.org/10.62051/0dbqaa10>
- Hulu, A. M. F., & Lhaksana, K. M. (2019). Analisis Sentimen Politik pada Twitter Menggunakan Metode Support Vector Machine (Studi Kasus : Pilpres 2019). *E-Proceeding of Engineering*, 6(2), 2017–2020.
- Iddrisu, A. M., Mensah, S., Boafo, F., Yeluripati, G. R., & Kudjo, P. (2023). A sentiment analysis framework to classify instances of sarcastic sentiments within the aviation sector. *International Journal of Information Management Data Insights*, 3(2).
<https://doi.org/10.1016/j.jjime.2023.100180>
- Idris, M., Rifai, A., & Tania, K. D. (2025). *Sentiment Analysis of Tokopedia App Reviews using Machine Learning and Word Embeddings*. 9(1), 210–219.

- Islam, S. S., Haque, M. S., Miah, M., Sarwar, T. Bin, & Nugraha, R. (2022). Application of machine learning algorithms to predict the thyroid disease risk: an experimental comparative study. *PeerJ Computer Science*, 8. <https://doi.org/10.7717/peerj-cs.898>
- Jabbar, A., Iqbal, S., Tamimy, M. I., Rehman, A., Bahaj, S. A., & Saba, T. (2023). An Analytical Analysis of Text Stemming Methodologies in Information Retrieval and Natural Language Processing Systems. *IEEE Access*, 11, 133681–133702. <https://doi.org/10.1109/ACCESS.2023.3332710>
- Kar, P., & Debbarma, S. (2023). Sentimental analysis & Hate speech detection on English and German text collected from social media platforms using optimal feature extraction and hybrid diagonal gated recurrent neural network. *Engineering Applications of Artificial Intelligence*, 126(January). <https://doi.org/10.1016/j.engappai.2023.107143>
- Kaur, J., & Saini, J. R. (2015). A Natural Language Processing Approach for Identification of Stop Words in Punjabi Language. *International Journal of Data Mining And Emerging Technologies*, 5(2), 114. <https://doi.org/10.5958/2249-3220.2015.00015.4>
- Kim, H., Lee, W., Lee, E.-H., & Kim, S. (2023). Review of evaluation and interpretation method for LDA model. *The Korean Data Analysis Society*. <https://doi.org/10.37727/jkdas.2023.25.4.1299>
- Koppel, M., & Schler, J. (2006). THE IMPORTANCE OF NEUTRAL EXAMPLES FOR LEARNING SENTIMENT. *Computational Intelligence*, 22. <https://doi.org/10.1111/j.1467-8640.2006.00276.x>
- Ladani, D. J., & Desai, N. P. (2020). Stopword Identification and Removal Techniques on TC and IR applications: A Survey. *2020 6th International Conference on Advanced Computing and Communication Systems, ICACCS 2020*, 466–472. <https://doi.org/10.1109/ICACCS48705.2020.9074166>
- Li, S., Li, W., Cook, C., Zhu, C., & Gao, Y. (2018). Independently Recurrent Neural Network (IndRNN): Building A Longer and Deeper RNN. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 5457–5466. <https://doi.org/10.1109/CVPR.2018.00572>
- Liu, N., & Shen, B. (2020). Aspect-based sentiment analysis with gated alternate neural network. *Knowledge-Based Systems*, 188. <https://doi.org/10.1016/j.knosys.2019.105010>

- Mäntylä, M., Graziotin, D., & Kuutila, M. (2016). The evolution of sentiment analysis - A review of research topics, venues, and top cited papers. *ArXiv*, *abs/1612.0*. <https://doi.org/10.1016/j.cosrev.2017.10.002>
- Mavhemwa, P., Zennaro, M., Nsengiyumva, P., & Nzanywayingoma, F. (2024). Weighted Naïve Bayes Multi-User Classification for Adaptive Authentication. *Journal of Physics Communications*. <https://doi.org/10.1088/2399-6528/ad8a16>
- Melton, C. A., Olusanya, O. A., Ammar, N., & Shaban-Nejad, A. (2021). Public sentiment analysis and topic modeling regarding COVID-19 vaccines on the Reddit social media platform: A call to action for strengthening vaccine confidence. *Journal of Infection and Public Health*, *14*(10), 1505–1512. <https://doi.org/10.1016/j.jiph.2021.08.010>
- Muhajir, M., Rosadi, D., Mada, U. G., & Regency, S. (2024). *Sentiment Analysis and Topic Modelling of Bjorka Using Support Vector Machine and Latent Dirichlet Allocation*. *5*(1), 57–66.
- Mulyani, Y. P., Saifurrahman, A., Arini, H. M., Rizqiawan, A., Hartono, B., Utomo, D. S., Spanellis, A., Beltran, M., Banjar Nahor, K. M., Paramita, D., & Harefa, W. D. (2024). Analyzing public discourse on photovoltaic (PV) adoption in Indonesia: A topic-based sentiment analysis of news articles and social media. *Journal of Cleaner Production*, *434*. <https://doi.org/10.1016/j.jclepro.2023.140233>
- Murfi, H., Theresia Gowandi, S., Ardaneswari, G., & Nurrohmah, S. (2024). BERT-based combination of convolutional and recurrent neural network for indonesian sentiment analysis. *Applied Soft Computing*, *151*(June 2023). <https://doi.org/10.1016/j.asoc.2023.111112>
- Pandey, A., & Vishwakarma, D. K. (2023). Progress, Achievements, and Challenges in Multimodal Sentiment Analysis Using Deep Learning: A Survey. *Applied Soft Computing*, *152*(December 2023), 111206. <https://doi.org/10.1016/j.asoc.2023.111206>
- Pavlov, D., Balasubramanyan, R., Dom, B., Kapur, S., & Parikh, J. (2004). Document Preprocessing for Naive Bayes classification and clustering with mixture of multinomials. *KDD-2004 - Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 829–834. <https://doi.org/10.1145/1014052.1016922>
- Polyzos, Efstathios & Wang, F. (2022). Polyzos, Efstathios & Wang, Fang. *Energy Economics*,

- Rana, R. (2016). *Gated Recurrent Unit (GRU) for Emotion Classification from Noisy Speech*. 1–9. <http://arxiv.org/abs/1612.07778>
- Samsir, Irmayani, D., Edi, F., Harahap, J. M., Jupriaman, Rangkuti, R. K., Ulya, B., & Watrianthos, R. (2021). Naives Bayes Algorithm for Twitter Sentiment Analysis. *Journal of Physics: Conference Series*, 1933(1). <https://doi.org/10.1088/1742-6596/1933/1/012019>
- Takahashi, K., Yamamoto, K., Kuchiba, A., Shintani, A., & Koyama, T. (2023). Hypothesis testing procedure for binary and multi-class F1-scores in the paired design. *Statistics in Medicine*, 42, 4177–4192. <https://doi.org/10.1002/sim.9853>
- Talaat, A. S. (2023). Sentiment analysis classification system using hybrid BERT models. *Journal of Big Data*, 10(1). <https://doi.org/10.1186/s40537-023-00781-w>
- Triyono, A., Faqih, A., & Fathurrohman. (2025). Implementation of the Naive Bayes Method in Sentiment Analysis of Spotify Application Reviews. *Journal of Artificial Intelligence and Engineering Applications (JAIEA)*. <https://doi.org/10.59934/jaiea.v4i2.824>
- Valdivia, A., Luzón, M., Cambria, E., & Herrera, F. (2018). Consensus vote models for detecting and filtering neutrality in sentiment analysis. *Inf. Fusion*, 44, 126–135. <https://doi.org/10.1016/J.INFFUS.2018.03.007>
- Varghese, A., Agyeman-Badu, G., & Cawley, M. (2020). Deep learning in automated text classification: a case study using toxicological abstracts. *Environment Systems and Decisions*, 1–15. <https://doi.org/10.1007/s10669-020-09763-2>
- Vats, S., Khanna, M. S., Kukreja, V., & Mehta, S. (2024). Federated Learning CNN Empowers Precision In Agriculture: A Case Study in Soybean Leaf Disease. *2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)*, 2, 1–6. <https://doi.org/10.1109/IATMSI60426.2024.10502984>
- Vayansky, I., & Kumar, S. A. P. (2020). A review of topic modeling methods. *Information Systems*, 94, 101582. <https://doi.org/10.1016/j.is.2020.101582>
- Victoria, H., & Maragatham, G. (2020). Automatic tuning of hyperparameters using Bayesian optimization. *Evolving Systems*, 12, 217–223. <https://doi.org/10.1007/s12530-020-09345-2>
- Vujović, Ž. (2021). Classification Model Evaluation Metrics. *International Journal of Advanced Computer Science and Applications*, 12(6), 599–606.

<https://doi.org/10.14569/IJACSA.2021.0120670>

- Yang, S., Yu, X., & Zhou, Y. (2020). LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset as an Example. *2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI)*, 98–101. <https://doi.org/10.1109/IWECAI50956.2020.00027>
- Yin, W., Kann, K., Yu, M., & Schütze, H. (2017). *Comparative Study of CNN and RNN for Natural Language Processing*. <http://arxiv.org/abs/1702.01923>
- Yue, L., Chen, W., Li, X., Zuo, W., & Yin, M. (2018). A survey of sentiment analysis in social media. *Knowledge and Information Systems*, 60, 617–663. <https://doi.org/10.1007/s10115-018-1236-4>
- Yukun, Peng, H., Khan, T., Cambria, E., & Hussain, A. (2018). Sentic LSTM: a Hybrid Network for Targeted Aspect-Based Sentiment Analysis. *Cognitive Computation*, 10, 639–650. <https://doi.org/10.1007/s12559-018-9549-x>
- Zhang, Y., & Wallace, B. (2015). *A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification*. <http://arxiv.org/abs/1510.03820>
- Zulqarnain, M., Ghazali, R., Aamir, M., & Hassim, Y. M. M. (2022). An efficient two-state GRU based on feature attention mechanism for sentiment analysis. *Multimedia Tools and Applications*, 83, 3085–3110. <https://doi.org/10.1007/s11042-022-13339-4>
- Zulqarnain, M., Ghazali, R., Hassim, Y. M. M., & Rehan, M. (2020). A comparative review on deep learning models for text classification. *Indonesian Journal of Electrical Engineering and Computer Science*, 19, 325–335. <https://doi.org/10.11591/IJEECS.V19.I1.PP325-335>