ABSTRACT

Stunting is a national issue in Indonesia. To address this problem, the Indonesian government launched the Free Nutritious Meal Program as a strategic policy solution. However, this nationally prioritized policy has triggered various public opinions on social media, reflecting society's perception of the program. To systematically understand public sentiment, a sentiment analysis method is needed that can capture the structure and meaning of short texts, such as tweets. This study offers a solution by developing a sentiment classification model based on Graph Convolutional Network (GCN) that utilizes Term Frequency-Inverse Document Frequency (TF-IDF) as text representation. The study also compares the impact of undersampling techniques on GCN model performance. Data were collected from the social media platform X (Twitter) and manually labeled by the researchers into three sentiment categories: positive, negative, and neutral. The labeled dataset, consisting of 5,979 tweets, was balanced using a class distribution-based undersampling technique to mitigate data imbalance. A document graph was constructed based on cosine similarity between TF-IDF vectors and used as input for the GCN model. Model evaluation was conducted using accuracy, precision, recall, and F1-score as performance metrics. The evaluation results show that the TF-IDF-based GCN model is capable of classifying public sentiment with an accuracy of 82.93% and an F1-score of 82.93%. The model also outperformed several classical machine learning algorithms such as Logistic Regression, SVM, XGBoost, Random Forest, and Naïve Bayes. These findings demonstrate that the integration of undersampling, TF-IDF representation, and GCN provides an effective and efficient approach to understanding public opinion on national policies through short text data on social media.

Keywords : Free Nutritious Meal Program, public sentiment, social media X (Twitter), undersampling, TF-IDF representation, Graph Convolutional Network (GCN)

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1 Latar Belakang

Stunting merupakan masalah penting yang harus diatasi di Indonesia. Stunting disebabkan oleh malnutrisi sejak dalam kandungan serta asupan gizi yang tidak memadai atau infeksi di masa kanak-kanak dapat menghambat pertumbuhan fisik dan perkembangan kognitif anak, melemahkan sistem kekebalan tubuh, serta meningkatkan risiko infeksi, keterlambatan perkembangan jangka panjang, dan kematian (UNICEF, 2022). Menurut statistik UNICEF (2022) Di antara negara-negara ASEAN, Indonesia mencatat prevalensi bayi lahir dengan berat badan rendah (*low birthweight*) sebesar 6,2%, menjadikannya negara dengan angka terendah dalam daftar tersebut. Bayi dengan berat lahir kurang dari 2.500 gram dikategorikan sebagai *low birthweight*, yang merupakan indikator penting dari kondisi kesehatan ibu dan janin selama kehamilan. Meskipun angka ini lebih baik dibandingkan negara ASEAN lain seperti Filipina (14,5%) dan Laos (17,3%), Indonesia tetap menghadapi tantangan serius karena bayi dengan berat badan rendah memiliki risiko lebih tinggi untuk kematian neonatal (dalam 28 hari pertama), pertumbuhan terhambat, IQ yang lebih rendah, serta kerentanan terhadap penyakit kronis seperti diabetes dan obesitas di usia dewasa. Sebagai respons terhadap permasalahan tersebut, pemerintah Indonesia meluncurkan program makan bergizi gratis yang ditargetkan kepada anak sekolah dan ibu hamil.

Program Makan Bergizi Gratis merupakan salah satu program prioritas nasional yang diinisiasi oleh Presiden Prabowo Subianto dengan tujuan utama meningkatkan kualitas gizi masyarakat, khususnya anak-anak sekolah dan ibu hamil, guna menurunkan angka stunting serta memperkuat sumber daya manusia Indonesia di masa mendatang (Sarjito, 2024). Program ini Bertujuan untuk meningkatkan asupan harian dengan anggaran yang terjangkau, sekitar Rp10.000 per individu per target (Kemenkes, 2023). Program ini dirancang untuk memberikan makan siang dan susu secara gratis kepada jutaan siswa dari jenjang SD hingga SMA/SMK, serta kelompok rentan lainnya, dengan harapan dapat mengubah kemampuan dan masa depan anak-anak Indonesia (Esa dkk., 2024). Namun hal ini menimbulkan berbagai pro dan kontra dari masyarakat melalui media sosial.

Media sosial, khususnya platform seperti *Twitter* yang sekarang dikenal sebagai X, telah menjadi wadah utama bagi masyarakat untuk menyampaikan opini, berbagi informasi,

dan membentuk opini publik secara luas. X mencapai rekor tertinggi sepanjang masa dengan 541,56 juta pengguna aktif bulanan pada tahun 2023. Jumlah ini meningkat 47% dari tahun 2022, yang hanya memiliki 368 juta pengguna aktif bulanan. Dikutip dalam famewall.io, 2024 oleh Amelia & Yusuf (2025), Indonesia menduduki peringkat keempat di dunia dengan pengguna aktif di X sebanyak 24,85 juta dari populasi masyarakat Indonesia yang berjumlah 278.696.200 jiwa. Tingginya jumlah pengguna tersebut menjadikan X sebagai sumber data yang sangat potensial untuk analisis sentimen, karena volume unggahan yang dihasilkan setiap harinya sangat besar, beragam, dan mencerminkan opini publik secara *real-time*. Selain itu, sifat X yang berbasis teks singkat membuat informasi yang dibagikan lebih padat dan langsung pada inti opini, sehingga memudahkan proses pengolahan data untuk tujuan penelitian. Karakteristik ini menjadikan platform X sebagai pilihan yang relevan dan strategis untuk mengkaji kecenderungan sentimen masyarakat terhadap topik tertentu.

Analisis sentimen merupakan metode yang efektif untuk mengkategorikan opini publik ke dalam sentimen positif, negatif, atau netral. Proses ini memungkinkan data teks yang tidak terstruktur, seperti unggahan di platform X, diubah menjadi informasi terukur yang dapat dianalisis secara sistematis. Dengan demikian, analisis sentimen tidak hanya memberikan gambaran yang jelas mengenai kecenderungan opini masyarakat terhadap platform X, tetapi juga membantu dalam mengidentifikasi tren, isu populer, serta perubahan persepsi publik dari waktu ke waktu. Banyak penelitian sebelumnya telah memanfaatkan analisis sentimen untuk memahami bagaimana opini publik terbentuk dan berubah terhadap suatu topik, peristiwa, atau entitas tertentu.

Penelitian yang dilakukan oleh Hermansyah & Hasibuan (2025) menganalisis sentimen masyarakat terhadap Program Makan Bergizi Gratis melalui media sosial X dengan memanfaatkan algoritma *machine learning* klasik yaitu *Linear Regression* dan *Random Forest*. Dalam studi ini, peneliti mengumpulkan data dari media sosial, melakukan *preprocessing* teks seperti normalisasi dan *stemming*, serta memberikan label sentimen untuk kemudian diklasifikasikan menggunakan kedua algoritma tersebut. Hasil penelitian menunjukkan bahwa algoritma *Random Forest* memiliki performa lebih unggul dibandingkan *Linear Regression*, dengan akurasi mencapai 85% dan F1-Score sebesar 90%, terutama dalam mendeteksi sentimen positif. Sementara itu, *Linear Regression* menunjukkan keunggulan dalam aspek presisi, meskipun secara keseluruhan *Random Forest* memberikan hasil klasifikasi yang lebih seimbang. Mayoritas sentimen publik terhadap

program ini bersifat positif, yakni sebesar 69,5%, meskipun tetap terdapat kritik terkait aspek implementasi dan pendanaan. Penelitian Hermansyah & Hasibuan (2025) menyoroti pentingnya pemanfaatan media sosial sebagai sumber informasi untuk memahami persepsi publik secara langsung dan sebagai dasar perumusan kebijakan publik yang lebih responsif dan efektif.

Penelitian lain dilakukan oleh Khemani dkk. (2024) berbeda dari penelitian Hermansyah & Hasibuan (2025) yang menggunakan algoritma *machine learning* klasik, Penelitian Khemani dkk. (2024) menggunakan metode *deep learning* untuk menguji efektivitas model dalam klasifikasi sentimen dibandingkan model *machine learning* klasik seperti *Linear Regression* dan *Random Forest*. Hasil penelitian membuktikan bahwa model deep learning yang dipakai yaitu *Graph Convolutional Network* (GCN) yang merupakan salah satu model *Graph Neural Network* (GNN) mendapatkan akurasi lebih baik daripada model - model *machine learning* pada umumnya. GCN terbukti memiliki akurasi tertinggi yaitu 93,86% dan lebih baik dibandingkan model lain seperti *passive aggressive classifier, random forest, decision tree, logistic regression, light GBM.*

Dalam bidang pemrosesan bahasa alami, pemilihan metode ekstraksi fitur yang tepat sangat penting untuk menghasilkan representasi teks yang informatif dan akurat. Salah satu metode yang banyak digunakan adalah *Term Frequency-Inverse Document Frequency* (TF-IDF). TF-IDF merupakan teknik statistik yang menonjolkan kata-kata penting dalam dokumen dengan mempertimbangkan frekuensi kemunculan kata di dokumen tertentu dan keseluruhan korpus, sehingga efektif untuk menyoroti kata kunci unik yang relevan dengan konteks. Studi terbaru oleh Zhou, dkk. (2024) yang membandingkan antara TF-IDF dan word2vec menunjukkan bahwa TF-IDF efektif dalam mengekstraksi perilaku respons utama dari data proses, sedangkan Word2Vec lebih unggul dalam menangkap fitur urutan perilaku yang kompleks. Penelitian Zhou dkk. (2024) juga menegaskan bahwa TF-IDF lebih cocok digunakan pada dataset kecil atau kasus sederhana karena sifatnya yang cepat, mudah, dan tetap memberikan hasil yang kompetitif, sementara Word2Vec lebih direkomendasikan untuk dataset besar dan kebutuhan generalisasi yang tinggi karena kemampuannya menangkap makna dan konteks kata serta ketahanannya terhadap *overfitting*, untuk hasil yang lebih optimal, kombinasi antara TF-IDF dan Word2Vec dapat dipertimbangkan guna memaksimalkan pemahaman model terhadap kata kunci sekaligus konteks. Temuan ini

menegaskan pentingnya pemilihan metode ekstraksi fitur yang sesuai dengan karakteristik data dan tujuan analisis dalam proyek-proyek analisis teks dan sentimen.

Berdasarkan temuan-temuan tersebut, penelitian skripsi ini akan berfokus pada pemanfaatan model *Graph Neural Network* (GNN), yaitu *Graph Convolutional Network* (GCN) untuk tugas klasifikasi sentimen terhadap opini masyarakat mengenai Program Makan Bergizi Gratis melalui media sosial X (*Twitter*). Untuk merepresentasikan teks ke dalam bentuk numerik, digunakan pendekatan umum dalam pemrosesan bahasa alami, yaitu *Term Frequency-Inverse Document Frequency* (TF-IDF). Pemilihan kombinasi antara representasi teks dan model GNN didasarkan pada kebutuhan untuk menangkap baik informasi semantik lokal maupun struktur relasional antar kata dalam dokumen dan jumlah dataset yang terbatas. Dengan pendekatan ini, penelitian ini bertujuan mengeksplorasi efektivitas metode berbasis graf dalam memahami dan mengklasifikasikan sentimen publik yang tersebar melalui platform digital secara lebih kontekstual dan terstruktur.

1.2 Rumusan Masalah

Penelitian ini berangkat dari kebutuhan untuk memahami bagaimana performa model *Graph Convolutional Network* (GCN) dengan representasi teks berbasis TF-IDF dalam mengklasifikasikan sentimen opini publik terhadap Program Makan Bergizi Gratis di media sosial X dengan cara membandingkan dengan model-model *machine learning* klasik seperti *Logistic Regression, Support Vector Machine* (SVM), *Random Forest, XGboost, Naïve bayes*. Selain itu, penelitian ini juga mengevaluasi sejauh mana performa menggunakan strategi *undersampling* berbasis pemilihan kalimat terpanjang dari masing-masing kelas mayoritas yang akan di *undersampling*. Pendekatan ini didasarkan pada asumsi bahwa kalimat yang lebih panjang cenderung mengandung informasi yang lebih kaya dan kontekstual, sehingga berpotensi meningkatkan kualitas representasi teks dalam proses pelatihan model klasifikasi.

1.3 Batasan Masalah

Penelitian ini dibatasi pada klasifikasi sentimen publik terhadap Program Makan Bergizi Gratis yang disampaikan melalui platform media sosial X (*Twitter*). Pengambilan data dari media sosial X karena media sosial ini karena beberapa faktor antara lain, banyaknya pengguna indonesia di X, relevansi dan kecepatan dalam menangkap opini

publik, X juga menyediakan akses data lebih mudah dibanding media lainnya. Data dikumpulkan secara otomatis menggunakan sistem pencarian berbasis kata kunci, yaitu *"*makan bergizi gratis*"* dan *"*makan siang gratis*"*, kata kunci "MBG" tidak digunakan karena waktu itu masih belum jelas informasi tentang nama program yang digunakan pemerintah, antara "makan bergizi gratis" atau "makan siang gratis". Rentang waktu pengambilan data dibatasi antara tanggal 1 November 2024 hingga 31 Maret 2025, dengan asumsi bahwa periode ini mencerminkan masa awal perbincangan publik terkait program kebijakan tersebut. Total data yang dikumpulkan sebanyak 9249 baris. Model hanya fokus pada GCN dan perbandingan hanya dilakukan dengan model *machine learning classic*, hal ini karena keterbatasan waktu. Label sentimen dilakukan secara manual dan evaluasi kinerja model dibatasi pada empat metrik utama, yaitu *accuracy*, *precision*, *recall*, dan *F1-score.*

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk mengklasifikasi sentimen publik terhadap program makan bergizi gratis dan membandingkan performa model *Graph Convolutional Network* (GCN) yang menggunakan representasi teks berbasis TF-IDF dalam mengklasifikasikan sentimen publik terhadap Program Makan Bergizi Gratis di media sosial X, dengan beberapa model *machine learning* klasik seperti *Logistic Regression*, SVM, *Random Forest, XGBoost*, dan Naïve Bayes. Selain itu, penelitian ini juga bertujuan untuk mengevaluasi efektivitas strategi *undersampling* berbasis pemilihan kalimat terpanjang dari masing-masing kelas.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1.5.1 Manfaat Teoritis

Penelitian ini diharapkan dapat memberikan kontribusi terhadap pengembangan ilmu pengetahuan, khususnya dalam bidang pemrosesan bahasa alami (*Natural Language Processing*/NLP) dan analisis sentimen. Dengan memanfaatkan model *Graph Convolutional Network* (GCN) berbasis representasi TF-IDF, penelitian ini berkontribusi dalam memperluas kajian penerapan model berbasis graf dalam konteks analisis teks berbahasa Indonesia. Selain itu, penelitian ini juga memberikan pemahaman lebih lanjut mengenai

efektivitas teknik penyeimbangan data melalui strategi *undersampling* dalam pengembangan model klasifikasi di masa mendatang.

1.5.2 Manfaat Praktis

Secara praktis, hasil penelitian ini diharapkan dapat dimanfaatkan oleh:

1. Pemerintah dan pembuat kebijakan, sebagai acuan untuk memahami persepsi publik terhadap kebijakan Program Makan Bergizi Gratis, sehingga dapat dijadikan sebagai dasar evaluasi dan penyempurnaan program secara lebih adaptif dan berbasis data.
2. Peneliti dan praktisi di bidang data dan teknologi informasi, sebagai rujukan dalam memilih model klasifikasi sentimen yang tepat berdasarkan karakteristik data, serta eksplorasi metode baru untuk penanganan ketidakseimbangan kelas pada data sosial media.
3. Mahasiswa dan akademisi, sebagai literatur tambahan dalam pengembangan kajian ilmiah di bidang NLP, *deep learning*, serta penerapan GNN dalam klasifikasi data teks berbahasa Indonesia.

BAB II
TINJAUAN PUSTAKA

Bab ini menyajikan landasan teori yang menjadi dasar dalam penelitian berjudul "Klasifikasi Sentimen Publik terhadap Program Makan Bergizi Gratis di Media Sosial X Menggunakan *Graph Convolutional Network* dan Representasi TF-IDF". Teori-teori yang dibahas mencakup konsep dasar analisis sentimen, representasi teks menggunakan TF-IDF, teori graf dalam konteks *Graph Neural Network* (GNN), serta algoritma *Graph Convolutional Network* (GCN) sebagai model utama yang digunakan dalam model. Pemahaman terhadap teori-teori ini diperlukan agar pembaca dapat memahami latar belakang teknis dan metodologis yang mendasari pendekatan penelitian ini secara menyeluruh.

2.1 *State of the Arts*

Beberapa penelitian terkait *topic trend analysis* maupun *topic modeling* dapat dilihat pada tabel 2.1.

Tabel 2.1 *State of the Arts*

No	Penelitian	Domain Masalah	Metode Penelitian	Hasil Terbaik (Akurasi)
1	Hermansyah & Hasibuan (2025)	Analisis Sentimen terhadap Program Makan Bergizi Gratis pada X (2 kelas)	*Linear Regression* dan *Random Forest* (TF-IDF)	83% dan 85%
2	Triningsih dkk. (2025)	Analisis Sentimen Terhadap Program Makan Bergizi Gratis pada X (3 kelas)	SVM dan *Random Forest* (TF-IDF)	85,74% dan 81,53%
3	Sitanggang dkk. (2024)	Analisis Sentimen Terhadap Program Makan Bergizi Gratis pada X	*Naïve Bayes* (TF-IDF)	72,2%
4	Rahmatullah dkk. (2025)	Analisis Sentimen Terhadap Program Makan Bergizi Gratis pada komentar *Youtube* (2 kelas)	*Naïve Bayes*	84,69%

No	Penelitian	Domain Masalah	Metode Penelitian	Hasil Terbaik (Akurasi)
5	Mazya dkk. (2024)	Analisis Perbandingan Metode Klasifikasi Sentimen Berita Saham	*Logistic Regression*, LSTM, LSTM+Word2Vec, Bi-LSTM+Word2Vec, **GCN**,	68,5%, 70%, 69%, 71%, **73,1%,**
6.	Shang dkk. (2024)	Analisis sentimen tingkat aspek pada ulasan produk atau layanan (misal: restoran, hotel, elektronik)	*Aspect-Sentence* GCN	90,15%
7	Žunić dkk. (2021)	Analisis sentimen tingkat aspek dalam domain kesehatan dan kesejahteraan	**GCN,** RNN, LSTM, BiLSTM	**81,78%,** 67,61%, 67,61%, 75,71%
8	Khemani dkk. (2024)	Deteksi misinformasi kesehatan terkait isu-isu kesehatan seperti COVID-19, kesehatan umum di media sosial	*Passive Aggressive Classifier*, *Random Forest*, *Decision Tree*, *Logistic Regression*, *Light GBM*, GCN *base*, GCN+BERT, **GCN+TF IDF,** GCN+Word2Vec	85,75%, 86%, 81,30%, 83,29%, 84,53%, 85%, 88,86%, **93,86%,** 81%

Penelitian dengan topik makan bergizi gratis telah dilakukan banyak dilakukan sebelumnya, salah satunya oleh Hermansyah & Hasibuan (2025) yang melakukan analisis sentimen terhadap data *Twitter* terhadap program makan bergizi gratis. Data yang digunakan adalah 2074 data cuitan dari *Twitter* berbahasa Indonesia kemudian diklasifikasikan menjadi 2 kelas yaitu positif dan negatif. Penelitian ini menggunakan metode *linear regression* dan *random forest* menggunakan TF-IDF sebagai pendekatannya. Hasil yang didapatkan dari penelitian ini adalah *random forest* mendapat hasil terbaik dengan akurasi 85% diikuti dengan *random forest* dengan akurasi 83%.

Penelitian lain oleh Triningsih dkk. (2025) juga melakukan analisis sentimen dengan topik program makan bergizi gratis. Data yang digunakan dalam penelitian ini adalah 2.400 data cuitan pada bulan Januari hingga Desember di tahun 2024 dari twitter atau X yang diklasifikasikan menjadi 3 kelas yaitu positif, netral dan negatif. Metode yang digunakan adalah SVM dan *Random Forest* menggunakan pendekatan TF-IDF. Hasil akhir penelitian

ini menunjukan bahwa SVM memiliki akurasi lebih tinggi yaitu 85,74% dibandingkan *Random Forest* yang memiliki akurasi 81,53%.

Penlitian lain yang dilakukan oleh Sitanggang dkk. (2024) yang juga memiliki topik yang sama yaitu program makan bergizi gratis. Penelitian ini menggunakan data sebanyak 2.216 cuitan bulan maret 2024 dari X atau *twitter* dan memiliki 2 kelas yaitu positif dan negatif. Model yang dipakai dalam penelitian ini adalah *naïve bayes* dengan pendekatan TF-IDF. Hasilnya menunjukan bahwa *naïve bayes* mendapatkan akurasi 72,2%.

Penelitian lain dengan topik yang dilakukan oleh Rahmatullah dkk. (2025) yaitu program makan bergizi gratis. Berbeda dari sebelumnya, penelitian ini mengambil data dari komentar *youtube*, dari video yang berkaitan dengan program makan bergizi gratis. Data berjumlah 1.470 dan memiliki 2 kelas yaitu positif dan negatif. Hasil penelitian ini adalah *naïve bayes* mendapatkan akurasi 84,69%.

Penelitian terkait model GCN juga pernah dilakukan oleh Mazya dkk. (2024) dengan topik klasifikasi sentimen berita saham. Data yang digunakan berjumlah 1.000 yang didapat melalui *web scrapping* dan memiliki 2 label yaitu positif dan negatif. Mazya dkk. (2024) membandingkan beberapa model yaitu *logistic regression* dengan TF-IDF, LSTM dengan TF-IDF, LSTM dengan Word2Vec, Bi-LSTM dengan Word2Vec, BERT, RoBERTa dan GCN Text. Hasilnya menunjukkan bahwa BERT memiliki akurasi paling tinggi yaitu 81%. Uniknya dalam penelitian ini, GCN mendapatkan akurasi lebih tinggi dibanding yang lain selain model BERT. Temuan ini menunjukkan bahwa model GCN lebih baik daripada model model *machine learning* pada umumnya.

Penelitian oleh Shang dkk. (2024) memperkenalkan model *Aspect-Sentence Graph Convolutional Network* (ASGCN) untuk *aspect-level sentiment analysis* yang secara khusus mengunggulkan kekuatan *Graph Convolutional Network* (GCN) dalam menangkap relasi sintaksis dan semantik dalam kalimat. Model ini memanfaatkan *adjacency matrix* berbasis *syntactic dependency tree* dengan mekanisme *position encoding*, serta membangun graf relasi antar *aspect words* dengan bobot berdasarkan posisi. Dengan pendekatan ini, ASGCN mampu menangkap hubungan semantik yang lebih kaya, baik antar kata dalam satu kalimat maupun antar aspek yang berbeda. Penelitian ini menggunakan *dataset benchmark* dari SemEval serta dataset ulasan produk, dan hasil eksperimen menunjukkan bahwa ASGCN secara konsisten melampaui model *baseline* dengan akurasi mencapai 86,34% dan *F1- score*

sebesar 79,96%. Keunggulan utama GCN dalam penelitian ini terletak pada kemampuannya mengintegrasikan struktur sintaksis dan relasi semantik, sehingga dapat mempersepsi informasi sentimen secara lebih komprehensif dibandingkan metode konvensional lainnya.

Penelitian lain terkait model yang sama dilakukan oleh Žunić dkk. (2021) memperkenalkan model *Aspect-Based Sentiment Analysis* (ABSA) berbasis *Graph Convolutional Network* (GCN) yang secara khusus memanfaatkan *dependency parse tree* untuk menangkap relasi sintaksis antar kata dalam kalimat. Model ini dievaluasi menggunakan dataset ulasan obat dari Drugs.com, di mana setiap ulasan secara otomatis dianotasi dengan konsep dari *Unified Medical Language System* (UMLS) sebagai aspek yang sentimennya akan diklasifikasikan. Hasil eksperimen menunjukkan bahwa pendekatan GCN yang mengoperasikan konvolusi pada *dependency parse tree* secara signifikan mengungguli model *deep learning* standar seperti LSTM, GRU, dan CNN, dengan perolehan *F1-score* sebesar 81,79%. Temuan ini menegaskan bahwa integrasi struktur sintaksis melalui GCN mampu meningkatkan performa ABSA, khususnya pada domain kesehatan dan kesejahteraan yang cenderung memiliki bias sentimen negatif.

Penelitian yang dilakukan oleh Khemani dkk. (2024) secara khusus menonjolkan keunggulan *Graph Convolutional Network* (GCN) dalam mendeteksi misinformasi kesehatan di media sosial, terutama pada data *Twitter* terkait COVID-19. Studi ini membandingkan performa GCN dengan berbagai model *machine learning* tradisional seperti *Passive Aggressive Classifier*, *Random Forest*, *Decision Tree*, *Logistic Regression*, serta model *hybrid* seperti GCN dengan BERT, TF-IDF, dan Word2Vec. Dataset yang digunakan terdiri dari 15.635 tweet yang telah dianotasi secara manual menjadi label "*reliable*" dan "*unreliable*" berdasarkan referensi resmi *Google*. Hasil eksperimen menunjukkan bahwa GCN yang dikombinasikan dengan embedding TF-IDF memberikan akurasi tertinggi, yaitu 93,86%, mengungguli model-model lain yang diuji, termasuk GCN murni maupun GCN dengan *embedding* lain. Keunggulan utama GCN dalam penelitian ini terletak pada kemampuannya menangkap relasi dan dependensi struktural antar dokumen dalam bentuk graf, sehingga menghasilkan pemahaman kontekstual yang lebih baik dalam klasifikasi misinformasi kesehatan dibandingkan pendekatan konvensional.

2.2 Media Sosial X

Media sosial X (sebelumnya *Twitter*) merupakan platform komunikasi daring yang memberikan akses bagi jutaan individu untuk berbagi informasi, berinteraksi, dan menerima berbagai jenis konten, termasuk promosi dan diskusi topik tertentu, secara *real-time* dan sangat dipengaruhi oleh aktivitas pengguna di dalamnya (Lerma dkk., 2024). Menurut Amelia & Yusuf (2025) dikutip dalam famewall Indonesia menduduki peringkat keempat di dunia dengan pengguna aktif di X sebanyak 24,85 juta dari populasi masyarakat Indonesia yang berjumlah 278.696.200 jiwa. Hal ini membuat banyak peneliti manfaatkan platform ini sebagai sumber data penting dalam analisis persepsi dan respons masyarakat Indonesia terhadap berbagai isu.

Interaksi pengguna secara aktif mengekspresikan opini, bertukar informasi, dan membentuk persepsi kolektif terhadap isu-isu sains, teknologi, maupun topik sosial lainnya, sehingga opini publik yang berkembang di X dapat memengaruhi adopsi, legitimasi, dan penyebaran inovasi di masyarakat (Suk dkk., 2025). Dengan banyaknya opini publik yang tersebar, ini dapat menjadi sumber data yang dapat dijadikan penelitian untuk melihat bagaimana masyarakat bereaksi terhadap isu-isu saat itu.

Menurut Hermansyah & Hasibuan (2025) analisis sentimen terhadap opini publik di X mengenai program makan bergizi gratis sangat penting karena memberikan wawasan berharga bagi pemerintah dan pembuat kebijakan. Hal ini penting untuk melihat respons dan persepsi masyarakat serta merancang intervensi yang lebih efektif guna meningkatkan keberhasilan program. Tantangan utama dalam mengolah data dari media sosial X meliputi volume data yang sangat besar, keberagaman format data, serta kebutuhan untuk menangani data yang tidak terstruktur dan berbahasa alami secara efektif (Singla dkk., 2025).

2.3 *Natural Language Processing*

Menurut Wilks (2005) yang merupakan ahli di bidang *Natural Language Processing* (NLP), NLP merupakan bidang *interdisipliner* yang berfokus pada pemrosesan dan pemahaman bahasa alami oleh komputer, mencakup berbagai tugas mulai dari *machine translation, question answering*, *information extraction*, hingga *document summarization*, yang perkembangannya sangat dipengaruhi oleh kemajuan linguistik, kecerdasan buatan, dan ketersediaan korpus digital. NLP memiliki peranan yang sangat penting di berbagai

sektor, seperti analisis sentimen, pemodelan topik, penerjemahan otomatis, asisten digital, dan sistem pencarian informasi. Secara keseluruhan, tujuan utama NLP adalah untuk memfasilitasi komunikasi antara manusia dan mesin menggunakan bahasa alami, sehingga komputer dapat menangani tugas-tugas yang melibatkan teks dan suara dengan cara yang lebih mirip dengan interaksi manusia.

2.4 Analisis Sentimen

Analisis sentimen pertama kali dipopulerkan secara sistematis oleh Pang dkk. (2002), yang menerapkan teknik *machine learning* untuk mengklasifikasikan opini publik berdasarkan polaritas sentimen dalam teks. Anlisis Sentimen adalah studi komputasional terhadap opini, sikap, dan emosi yang diekspresikan dalam teks, dengan tujuan mengidentifikasi sentimen yang terkandung dan mengklasifikasikan polaritasnya, baik pada tingkat dokumen, kalimat, maupun aspek tertentu dari suatu entitas (Medhat dkk., 2014). Dalam literatur analisis sentimen oleh Medhat dkk. (2014), pendekatan-pendekatan umum diklasifikasikan menjadi tiga kategori, yaitu pendekatan berbasis leksikon, pembelajaran mesin klasik, dan pendekatan *deep learning*.

Metode berbasis leksikon merupakan salah satu pendekatan awal yang digunakan dalam analisis sentimen. Pendekatan ini memanfaatkan kamus kata berpolaritas untuk menentukan orientasi sentimen suatu teks, sebagaimana dijelaskan oleh Taboada dkk. (2011) dalam studi komprehensif mengenai teknik leksikal untuk klasifikasi opini publik. Metode pembelajaran mesin klasik dalam analisis sentimen menggunakan algoritma seperti SVM, *Naïve Bayes*, dan *K-Nearest Neighbor* untuk mengekstraksi pola opini dari teks terstruktur melalui fitur numerik seperti TF-IDF atau *Bag-of-Words* (Medhat dkk., 2014). Teknik *deep learning* dalam analisis sentimen menggunakan jaringan saraf dalam seperti LSTM dan *Transformer* (seperti BERT) untuk menangkap struktur semantik dan konteks emosional dalam teks dengan performa tinggi yang melampaui metode klasik (Medhat dkk., 2014). Sebagian besar pendekatan *deep learning* tersebut dikembangkan di atas fondasi jaringan syaraf tiruan.

2.5 *Text Preprocessing*

Text Preprocessing adalah serangkaian tahapan awal yang bertujuan untuk membersihkan, menyederhanakan, dan menstandarkan teks mentah agar dapat diproses

lebih lanjut oleh algoritma pemrosesan bahasa alami atau model pembelajaran mesin. Proses ini mencakup langkah-langkah seperti penghapusan tanda baca, konversi huruf ke bentuk kecil (*lowercasing*), penghapusan kata-kata umum (*stopwords*), *stemming*, dan normalisasi kata. Menurut Selim & Assiri (2025) dalam penelitiannya yang berfokus pada pengembangan sistem *Text-To-Speech* (TTS) berbahasa Arab mengatakan bahwa *preprocessing* merupakan tahap krusial dalam sistem NLP karena memungkinkan transformasi teks mentah menjadi format terstruktur yang siap digunakan dalam proses pembelajaran mesin. Dalam penelitian Abdullah dkk. (2025) tentang mendeteksi cuitan *spam* pada *Twitter* juga mengatakan bahwa proses *preprocessing* seperti penghapusan kata-kata tidak penting dan normalisasi teks berkontribusi besar dalam meningkatkan akurasi klasifikasi.

2.5.1 Pembersihan Data

Terdapat 16 teknik yang bisa dipakai agar teks mentah dapat digunakan menjadi teks siap pakai dalam NLP khususnya *twitter* atau X (Symeonidis dkk., 2018). 16 teknik itu antara lain:

1. Menghapus *Unicode Strings* dan *Noise*

   Tidak semua dataset bersih, terlebih di X, banyak sisa sisa dari *scraping* data yaitu *unicode strings* seperti “\u002c” dan \x06” yang merupakan sisa saat pengumpulan data yang harus dihapus. Contoh pembersihan *Unicode Strings dan Noise* ada dalam tabel 2.2.

Tabel 2.2 Sampel Menghapus *Unicode Strings* dan *Noise*

No	Sebelum menghapus Unicode Strings dan Noise	Menghapus *unicode strings* dan *noise*
1	Aku seneng bet sama makan bergizi gratisss ini!!! :D, lanjot 02 pak prabowo \xfs9 #MBGKEREN https://twitter.com/g918e425	Aku seneng bet sama makan bergizi gratisss ini!!! :D, lanjot 02 pak prabowo #MBGKEREN https://twitter.com/g918e425
2	Asbun nih @prabowo Mending PIKIRIN pendidikan sama kesehatan dulu ga si, daripada mbg gajelas gini	Asbun nih Mending PIKIRIN pendidikan sama kesehatan dulu ga si, daripada mbg gajelas gini

2. Menghapus URL dan Penyebutan Pengguna Lain

   Di *Twitter*, sebagian besar kalimat mengandung URL, *mention* pengguna, atau *hashtag*. Karena elemen ini tidak membawa sentimen, mereka diganti dengan *tag* khusus,

misalnya ‘URL’ untuk link dan ‘AT_USER’ untuk *mention*. Simbol *hashtag* dihilangkan. Pendekatan ini hanya berlaku untuk teks *Twitter* dan harus dilakukan sebelum teknik *preprocessing* lain. Contoh menghapus URL dan Penyebutan Pengguna Lain dapat dilihat pada tabel 2.3.

Tabel 2.3 Sampel Menghapus URL dan Penyebutan Pengguna Lain

No	Sebelum menghapus URL dan Penyebutan Pengguna Lain	Menghapus URL dan Penyebutan Pengguna Lain
1	Aku seneng bet sama makan bergizi gratisss ini!!! :D, lanjot 02 pak prabowo #MBGKEREN https://twitter.com/g918e425	Aku seneng bet sama makan bergizi gratisss ini!!! :D, lanjot 02 pak prabowo #MBGKEREN
2	Asbun nih @prabowo Mending PIKIRIN pendidikan sama kesehatan dulu ga si, daripada mbg gajelas gini	Asbun nih Mending PIKIRIN pendidikan sama kesehatan dulu ga si, daripada mbg gajelas gini

3. Mengganti Bahasa Gaul dan Singkatan

Pengguna media sosial sering menulis dengan bahasa informal yang mengandung slang dan singkatan. Agar maknanya bisa dipahami, kata-kata ini perlu diganti dengan bentuk lengkapnya menggunakan kamus manual. Contohnya, “ty” menjadi “terima kasih” dan “omg” menjadi “ya tuhan”. Contoh mengganti bahasa gaul dan singkatan dapat dilihat pada tabel 2.4.

Tabel 2.4 Sampel Mengganti Bahasa Gaul dan Singkatan

No	Sebelum Mengganti Bahasa Gaul dan Singkatan	Mengganti Bahasa Gaul dan Singkatan
1	Aku seneng bet sama makan bergizi gratisss ini!!! :D, lanjot 02 pak prabowo #MBGKEREN	Aku seneng bet sama makan bergizi gratisss ini!!! :D, lanjot 02 pak prabowo #MBGKEREN
2	Asbun nih Mending PIKIRIN pendidikan sama kesehatan dulu ga si, daripada mbg gajelas gini	Asal bunyi nih Mending PIKIRIN pendidikan sama kesehatan dulu ga si, daripada mbg gajelas gini

4. Mengganti kontraksi

Kata-kata kontraksi seperti “nggamau” dan “nggabisa” diganti dengan bentuk lengkapnya, seperti “tidak mau” dan “tidak bisa”. Ini penting agar tokenisasi menjadi lebih tepat dan memudahkan teknik penggantian negasi yang dilakukan kemudian. Contoh mengganti kontraksi dapat dilihat pada tabel 2.5.

Tabel 2.5 Sampel Mengganti Kontraksi

No	Sebelum Mengganti kontraksi	Mengganti kontraksi
1	Aku seneng bet sama makan bergizi gratisss ini!!! :D, lanjot 02 pak prabowo #MBGKEREN	Aku seneng bet sama makan bergizi gratisss ini!!! :D, lanjot 02 pak prabowo #MBGKEREN
2	Asal bunyi nih Mending PIKIRIN pendidikan sama kesehatan dulu ga si, daripada mbg gajelas gini	Asal bunyi nih Mending PIKIRIN pendidikan sama kesehatan dulu ga sih, daripada mbg tidak jelas gini

5. Menghapus Angka

   Angka biasanya tidak mengandung sentimen sehingga dihapus. Namun, penghapusan ini dilakukan setelah penggantian slang karena beberapa slang seperti "gr8" (great) mengandung angka. Ada juga yang berpendapat angka bisa membantu klasifikasi. Contoh menghapus angka dapat dilihat pada tabel 2.6.

Tabel 2.6 Sampel Menghapus Angka

No	Sebelum Menghapus Angka	Menghapus Angka
1	Aku seneng bet sama makan bergizi gratisss ini!!! :D, lanjot 02 pak prabowo #MBGKEREN	Aku seneng bet sama makan bergizi gratisss ini!!! :D, lanjot pak prabowo #MBGKEREN
2	Asal bunyi nih Mending PIKIRIN pendidikan sama kesehatan dulu ga sih, daripada mbg tidak jelas gini	Asal bunyi nih Mending PIKIRIN pendidikan sama kesehatan dulu ga sih, daripada mbg tidak jelas gini

6. Mengganti Pengulangan Tanda Baca

   Tanda baca seperti tanda tanya, tanda seru, dan titik yang diulang diubah menjadi *tag* khusus, misalnya "???" menjadi "*multiQuestionMark*". Ini menunjukkan emosi yang kuat dan dilakukan sebelum penghapusan tanda baca. Hal ini dilakukan agar model dapat membaca emosi pada tanda baca yang berlebihan seperti tanda baca yang berulang. Contoh mengganti pengulangan tanda baca dapat dilihat pada tabel 2.7.

Tabel 2.7 Sampel Pengulangan Tanda Baca

No	Sebelum Mengganti Pengulangan Tanda Baca	Mengganti Pengulangan Tanda Baca
1	Aku seneng bet sama makan bergizi gratisss ini!!! :D, lanjot pak prabowo #MBGKEREN	Aku seneng bet sama makan bergizi gratisss ini *multiExclamationMark* :D, lanjot pak prabowo #MBGKEREN
2	Asal bunyi nih Mending PIKIRIN pendidikan sama kesehatan dulu ga sih, daripada mbg tidak jelas gini	Asal bunyi nih Mending PIKIRIN pendidikan sama kesehatan dulu ga sih, daripada mbg tidak jelas gini

7. Mengganti Negasi dengan Antonim

   Jika dalam kalimat ditemukan kata "tidak" dan kata berikutnya memiliki antonim, maka frasa tersebut diganti dengan antonim yang tepat, misalnya "tidak baik" diganti menjadi "buruk". Pendekatan ini jarang digunakan tapi membantu memperjelas sentimen. Contoh mengganti negasi dengan antonim dapat dilihat pada tabel 2.8.

Tabel 2.8 Sampel Mengganti Negasi dengan Antonim

No	Sebelum Mengganti Negasi dengan Antonim	Mengganti Negasi dengan Antonim
1	Aku seneng bet sama makan bergizi gratisss ini *multiExclamationMark* :D, lanjot pak prabowo #MBGKEREN	Aku seneng bet sama makan bergizi gratisss ini *multiExclamationMark* :D, lanjot pak prabowo #MBGKEREN
2	Asal bunyi nih Mending PIKIRIN pendidikan sama kesehatan dulu ga sih, daripada mbg tidak jelas gini	Asal bunyi nih Mending PIKIRIN pendidikan sama kesehatan dulu ga sih, daripada mbg ambigu gini

8. Menghapus Tanda Baca

   Tanda baca sering dihapus karena tidak berpengaruh pada analisis sentimen, meski kadang tanda baca seperti tanda seru dapat menunjukkan emosi kuat. Penghapusan tanda baca ini merupakan teknik klasik dalam pemrosesan teks. Contoh menghapus tanda baca dapat dilihat pada tabel 2.9.

Tabel 2.9 Sampel Menghapus Tanda Baca

No	Sebelum Menghapus Tanda Baca	Menghapus Tanda Baca
1	Aku seneng bet sama makan bergizi gratisss ini *multiExclamationMark* :D, lanjot pak prabowo #MBGKEREN	Aku seneng bet sama makan bergizi gratisss ini *multiExclamationMark* D lanjot pak prabowo #MBGKEREN

No	Sebelum Menghapus Tanda Baca	Menghapus Tanda Baca
2	Asal bunyi nih Mending PIKIRIN pendidikan sama kesehatan dulu ga sih, daripada mbg ambigu gini	Asal bunyi nih Mending PIKIRIN pendidikan sama kesehatan dulu ga sih daripada mbg ambigu gini

9. Menangani Kata Kapital (Huruf Besar Semua)

   Kata-kata yang semua hurufnya kapital dan panjangnya lebih dari dua karakter dianggap mengekspresikan emosi kuat. Kata tersebut diberi awalan "ALL_CAPS_" agar tetap dikenali oleh model. Contoh menangani kata kapital dapat dilihat pada tabel 2.10.

Tabel 2.10 Sampel Menangani Kata Kapital (Huruf Besar Semua)

No	Sebelum Menangani Kata Kapital (Huruf Besar Semua)	Menangani Kata Kapital (Huruf Besar Semua)
1	Aku seneng bet sama makan bergizi gratisss ini *multiExclamationMark* D lanjot pak prabowo #MBGKEREN	Aku seneng bet sama makan bergizi gratisss ini *multiExclamationMark* D lanjot pak prabowo #MBGKEREN
2	Asal bunyi nih Mending PIKIRIN pendidikan sama kesehatan dulu ga sih daripada mbg ambigu gini	Asal bunyi nih Mending ALL_CAPS_PIKIRIN pendidikan sama kesehatan dulu ga sih daripada mbg ambigu gini

10. Mengubah Huruf Menjadi Kecil Semua

    Semua kata diubah menjadi huruf kecil agar kata yang sama dengan bentuk berbeda dapat digabung dan mengurangi dimensi data. Contoh mengubah huruf menjadi kecil semua dapat dilihat pada tabel 2.11.

Tabel 2.11 Sampel Mengubah Huruf Menjadi Kecil Semua

No	Sebelum Mengubah Huruf Menjadi Kecil Semua	Mengubah Huruf Menjadi Kecil Semua
1	Aku seneng bet sama makan bergizi gratisss ini *multiExclamationMark* D lanjot pak prabowo #MBGKEREN	aku seneng bet sama makan bergizi gratisss ini *multiexclamationmark* d lanjot pak prabowo #mbgkeren
2	Asal bunyi nih Mending ALL_CAPS_PIKIRIN pendidikan sama kesehatan dulu ga sih daripada mbg ambigu gini	asal bunyi nih mending all_caps_pikirin pendidikan sama kesehatan dulu ga sih daripada mbg ambigu gini

11. Menghapus Kata *Stopword*

    Kata-kata umum seperti “dan”, “atau”, “yang” yang sering muncul namun kurang bermakna dalam analisis sentimen dihapus. Daftar *stopword* biasanya diambil dari pustaka seperti NLTK. Contoh menghapus kata *stopword* dapat dilihat pada tabel 2.12.

Tabel 2.12 Sampel Menghapus Kata *Stopword*

No	Sebelum Menghapus Kata *Stopword*	Menghapus Kata *Stopword*
1	seneng bet sama makan bergizi gratisss ini *multiexclamationmark* d lanjot pak prabowo #mbgkeren	senang makan bergizi gratisss multiExclamationMark d lanjot prabowo #mbgkeren
2	asal bunyi nih mending all_caps_pikirin pendidikan sama kesehatan dulu ga sih daripada mbg ambigu gini	asal bunyi all_caps_pikirin pendidikan kesehatan mbg ambigu

12. Mengganti Kata yang Berulang-ulang (*Elongated Words*)

    Kata yang sengaja atau tidak sengaja diulang hurufnya seperti “kerennnnn” dikembalikan ke bentuk asli “keren” agar tidak dianggap kata berbeda oleh model. Contoh mengganti kata yang berulang-ulang dapat dilihat pada tabel 2.13.

Tabel 2.13 Sampel Mengganti Kata yang Berulang-ulang (*Elongated Words*)

No	Sebelum Mengganti Kata yang Berulang-ulang (*Elongated Words*)	Mengganti Kata yang Berulang-ulang (*Elongated Words*)
1	senang makan bergizi gratisss multiExclamationMark d lanjot prabowo #mbgkeren	senang makan bergizi gratis multiExclamationMark d lanjot prabowo #mbgkeren
2	asal bunyi all_caps_pikirin pendidikan kesehatan mbg ambigu	asal bunyi all_caps_pikirin pendidikan kesehatan mbg ambigu

13. Koreksi Ejaan

    Kesalahan ejaan yang sering terjadi di teks informal seperti “kalean” diperbaiki menjadi “kalian” menggunakan alat koreksi ejaan otomatis, terlebih pada media sosial seperti X, karena banyak sekali kata yang tidak baku. Hal ini dilakukan untuk mempermudah model dalam melakukan klasifikasi. Contoh koreksi ejaan dapat dilihat pada tabel 2.14.

Tabel 2.14 Sampel Koreksi Ejaan

No	Sebelum Koreksi Ejaan	Koreksi Ejaan
1	senang makan bergizi gratis multiExclamationMark d lanjot prabowo #mbgkeren	senang makan bergizi gratis multiExclamationMark d lanjut prabowo #mbgkeren
2	asal bunyi all_caps_pikirin pendidikan kesehatan mbg ambigu	asal bunyi all_caps_pikirin pendidikan kesehatan mbg ambigu

14. Lematisasi

Kata-kata yang mengalami perubahan bentuk akibat infleksi diubah ke bentuk dasar atau lema biasanya dilakukan dengan menggunakan *library*, sesuai dengan kamus. Contohnya, "berlari" menjadi "lari". Contoh lematisasi dapat dilihat pada tabel 2.15.

Tabel 2.15 Sampel Lematisasi

No	Sebelum Lematisasi	Lematisasi
1	senang sekali makan bergizi gratis multiExclamationMark d lanjut prabowo #mbgkeren	senang makan gizi gratis multiExclamationMark d lanjut prabowo #mbgkeren
2	asal bunyi all_caps_pikirin pendidikan kesehatan mbg ambigu	asal bunyi all_caps_pikirin didik sehat mbg ambigu

15. Stemming

Hampir sama dengan lematisasi, hanya saja caranya lebih mudah, dimana kata-kata dipotong akhiran atau imbuhan sehingga menjadi bentuk dasar (*stem*) guna mengurangi variasi kata dan dimensi data. Contoh *Stemming* dapat dilihat pada tabel 2.16.

Tabel 2.16 Sampel *Stemming*

No	Sebelum *Stemming*	*Stemming*
1	senang sekali makan bergizi gratis multiExclamationMark d lanjut prabowo #mbgkeren	senang makan gizi gratis multiExclamationMark d lanjut prabowo #mbgkeren
2	asal bunyi all_caps_pikirin pendidikan kesehatan mbg ambigu	asal bunyi all_caps_pikirin didik sehat mbg ambigu

2.5.2 Tokenisasi

Tokenisasi merupakan tahap awal penting dalam pemrosesan bahasa alami yang bertujuan memecah teks mentah menjadi unit-unit yang lebih kecil seperti kata, frasa, atau

simbol, yang disebut *token*. Proses ini menjadi fondasi bagi berbagai tugas NLP karena semua tahap berikutnya seperti *stemming*, *lemmatization*, dan *vectorization* bekerja berdasarkan token yang dihasilkan. Metode tokenisasi berkembang dari pendekatan berbasis spasi dan aturan hingga teknik modern seperti *Byte Pair Encoding* (Sennrich dkk., 2015) dan *WordPiece* yang digunakan dalam BERT (Devlin dkk., 2018). Teknik tokenisasi *subword* ini sangat penting untuk menangani kata-kata yang tidak ada dalam kosakata (*out-of-vocabulary*). Menurut Fatima dkk. (2025) dalam penelitiannya tentang NLP, mengatakan bahwa tokenisasi membantu mengidentifikasi batas-batas kata dalam kalimat, memungkinkan sistem untuk menangani teks panjang menjadi format yang lebih terstruktur dan dapat dianalisis secara semantik. Contoh tokenisasi dapat dilihat pada tabel 2.17.

Tabel 2.17 Sampel Tokenisasi

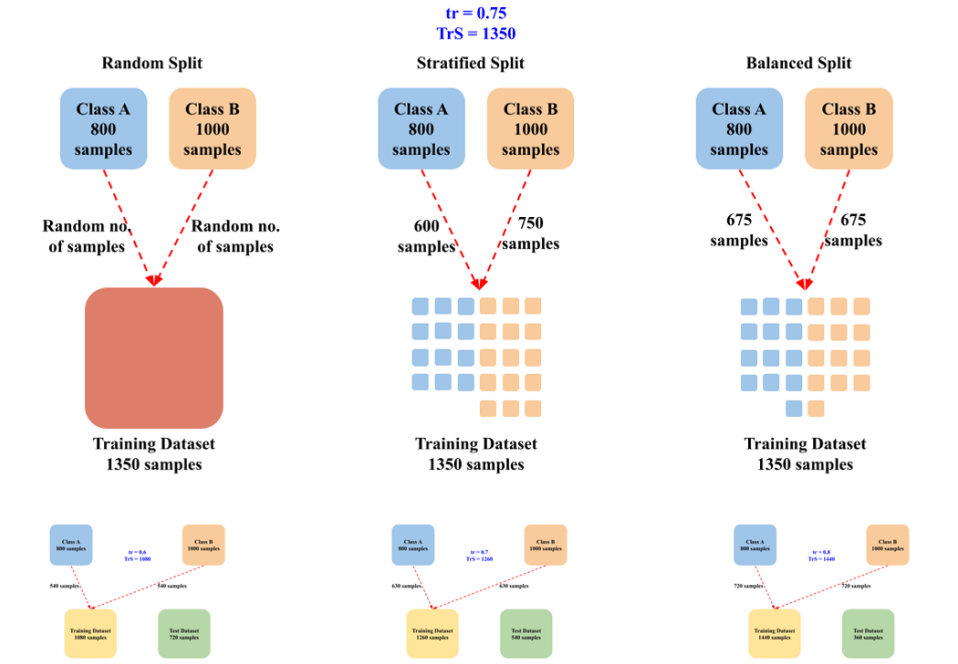
No	Sebelum Tokenisasi	Tokenisasi
1	senang makan gizi gratis multiExclamationMark d lanjut prabowo #mbgkeren	[“senang”, “makan”, “gizi”, “gratis”, “multiExclamationMark”, “d”, “lanjut”, “prabowo”, “#mbgkeren”]
2	asal bunyi all_caps_pikirin didik sehat mbg ambigu	[“asal”, “bunyi”, “all_caps_pikirin”, “didik”, “sehat”, “mbg”, “ambigu”]

2.6 Pembagian Data

Pembagian data merupakan tahap penting dalam *pipeline* pembelajaran mesin yang bertujuan untuk memastikan bahwa model yang dilatih memiliki kemampuan generalisasi yang baik terhadap data yang belum pernah dilihat. Secara umum, dataset dibagi menjadi tiga subset utama: data latih (*training set*), data validasi (*validation set*), dan data uji (*test set*). Data latih digunakan untuk membangun model, data validasi digunakan untuk menyetel parameter dan mencegah *overfitting*, sedangkan data uji digunakan sebagai acuan akhir untuk mengevaluasi performa model.

Pembagian data dapat dibagi kedalam 3 cara menurut Kohavi (1995). Gambar 2.1 mengilustrasikan tiga pendekatan umum dalam proses pembagian data pada pembelajaran mesin, yaitu *random split*, *stratified split*, dan *balanced split*. Pada metode *random split*, data dari masing-masing kelas dibagi secara acak tanpa mempertimbangkan proporsi kelas, sehingga berpotensi menghasilkan distribusi data yang tidak seimbang di subset pelatihan. Metode ini kurang ideal untuk dataset dengan distribusi kelas yang timpang karena dapat menyebabkan model bias terhadap kelas mayoritas. Sementara itu, metode *stratified split*

membagi data dengan menjaga proporsi masing-masing kelas agar konsisten di setiap subset. Meskipun stratifikasi dapat menghasilkan subset yang lebih representatif, distribusi absolut antar kelas tetap tidak seimbang, sehingga masih menyisakan potensi bias.



(a) Balanced split with train ratio 0.6. (b) Balanced split with train ratio 0.7. (c) Balanced split with train ratio 0.8.

Gambar 2.1 Visualisasi Pembagian Data *Random Split, Stratified Split dan Balaned Splt* (Kohavi, 1995)

Balanced split merupakan solusi terhadap ketimpangan ini, yaitu teknik yang secara sengaja memilih jumlah sampel yang sama dari setiap kelas untuk subset pelatihan maupun pengujian, meskipun jumlah total data antar kelas tidak seimbang. Pendekatan ini efektif dalam mengurangi bias kelas dan sering digunakan dalam skenario klasifikasi yang menghadapi masalah *class imbalance*. Gambar juga menunjukkan penerapan *balanced split* dengan berbagai rasio pelatihan (0.6, 0.7, dan 0.8), di mana jumlah sampel yang dipilih dari tiap kelas disesuaikan agar tetap seimbang, sementara ukuran total subset disesuaikan dengan rasio pelatihan yang digunakan. Pendekatan ini digunakan dalam penelitian untuk memastikan bahwa model tidak terlalu dipengaruhi oleh dominasi kelas tertentu selama proses pelatihan, serta meningkatkan kemampuan generalisasi model terhadap data uji.

Lebih lanjut, Kohavi (1995) dalam studi klasiknya memperkenalkan *k-fold cross-validation*, yaitu pendekatan yang membagi dataset ke dalam k subset (*fold*). Dalam

pendekatan ini setiap subset secara bergiliran digunakan sebagai data validasi sementara sisanya menjadi data latih. Teknik ini terbukti meningkatkan stabilitas dan akurasi estimasi performa, terutama pada dataset berukuran kecil atau tidak seimbang.

2.7 *Undersampling*

Undersampling merupakan salah satu pendekatan penyeimbangan data dengan cara mengurangi jumlah sampel dari kelas mayoritas untuk menyetarakan distribusi kelas, sebagaimana dijelaskan oleh He & Garcia (2009) dalam tinjauan komprehensif mengenai pembelajaran dari data tidak seimbang. Dalam konteks pembelajaran mesin, *undersampling* digunakan untuk mengatasi bias klasifikasi akibat dominasi kelas mayoritas, yang sering menyebabkan model mengabaikan pola dari kelas minoritas. Teknik ini bekerja dengan memangkas jumlah data dari kelas mayoritas, sehingga menghasilkan distribusi data yang lebih seimbang dan memungkinkan model mengenali pola dari kedua kelas secara setara. Meskipun metode ini sederhana dan efisien, salah satu tantangan utama dari *undersampling* adalah potensi kehilangan informasi penting, yang dapat menurunkan akurasi secara keseluruhan. Pemilihan strategi *undersampling* seperti *random undersampling*, *NearMiss*, atau *cluster-based undersampling* perlu disesuaikan dengan karakteristik dataset dan tujuan klasifikasi (Govindrajan, 2025). Contoh teknik undersampling sebagai berikut.

1. *Random Undersampling* (RUS)

   Random undersampling merupakan teknik pengurangan jumlah data pada kelas mayoritas secara acak hingga jumlahnya seimbang dengan kelas minoritas. Metode ini sederhana dan cepat diterapkan, namun memiliki kelemahan karena berpotensi membuang informasi penting yang dapat memengaruhi kinerja model. Teknik ini biasanya digunakan sebagai *baseline* sebelum mencoba metode *undersampling* yang lebih kompleks.
2. *NearMiss*

   NearMiss merupakan metode *undersampling* berbasis jarak yang memilih sampel mayoritas berdasarkan kedekatannya dengan sampel minoritas. Terdapat beberapa variasi, seperti *NearMiss-1* yang memilih sampel mayoritas dengan rata-rata jarak terdekat ke k tetangga minoritas, *NearMiss-2* yang memilih sampel dengan jarak terjauh dari tetangga mayoritas, dan *NearMiss-3* yang memilih sejumlah k sampel mayoritas terdekat untuk setiap sampel minoritas. Pendekatan ini dapat membantu

mempertahankan batas keputusan yang lebih representatif dibandingkan metode acak.

3. *Cluster-based Undersampling*

   Cluster-based undersampling dilakukan dengan membagi data mayoritas menjadi beberapa cluster, misalnya menggunakan algoritma *K-Means*, kemudian memilih perwakilan dari setiap cluster. Metode ini bertujuan menjaga keberagaman karakteristik data mayoritas sehingga informasi penting tetap terjaga meskipun jumlah sampel dikurangi.

4. *Custom Undersampling*

   Custom undersampling adalah teknik pengurangan data kelas mayoritas yang dirancang khusus sesuai dengan karakteristik dan kebutuhan dataset yang digunakan. Tidak seperti metode umum seperti RUS, *NearMiss*, atau *cluster-based*, pendekatan ini memungkinkan peneliti menentukan kriteria seleksi sampel berdasarkan atribut tertentu, distribusi label, atau hasil analisis awal terhadap data. Misalnya, penentuan sampel mayoritas yang dihapus dapat mempertimbangkan skor kemiripan antar dokumen, tingkat relevansi, atau hasil prediksi awal model. Keunggulan metode ini adalah fleksibilitasnya untuk mempertahankan data yang dianggap penting bagi kinerja model, sehingga mengurangi risiko kehilangan informasi krusial. Namun, kekurangannya adalah memerlukan analisis mendalam dan waktu lebih lama dalam tahap perancangan dibandingkan metode *undersampling* yang bersifat umum.

2.8 *Term Frequency – Inverse Document Frequency* (TF-IDF)

Term Frequency–Inverse Document Frequency (TF-IDF) merupakan teknik representasi teks yang menilai pentingnya sebuah kata dalam sebuah dokumen relatif terhadap seluruh koleksi dokumen. Konsep ini secara formal diperkenalkan oleh Salton & Buckley (1988) dan telah terbukti efektif dalam berbagai aplikasi klasifikasi teks dan pencarian informasi. TF-IDF sendiri memiliki rumus umum dimana untuk sebuah *term* t dalam dokumen d, TF-IDF dapat dihitung dengan persamaan 2.1, 2.2 dan 2.3:

$$TF - IDF(t,d) = TF(t,d) \times IDF(t) \quad (2.1)$$

TF (*Term Frequency)*:

$TF(t,d) = \frac{f_{t,d}}{\sum_k f_{k,d}}$ ..... (2.2)

Keterangan:

$f_{t,d}$: Jumlah kemunculan term t dalam dokumen d

IDF (*Inverse Document Frequency*):

$IDF(t) = \log\left(\frac{N}{1+n_t}\right)$ ..... (2.3)

Keterangan:

N : Jumlah total dokumen dalam *corpus*

n_t : Jumlah dokumen yang mengandung *term t*

Dengan catatan, 1 dalam penyebut digunakan untuk menghindari pembagian dengan nol, jika nilai n 0.

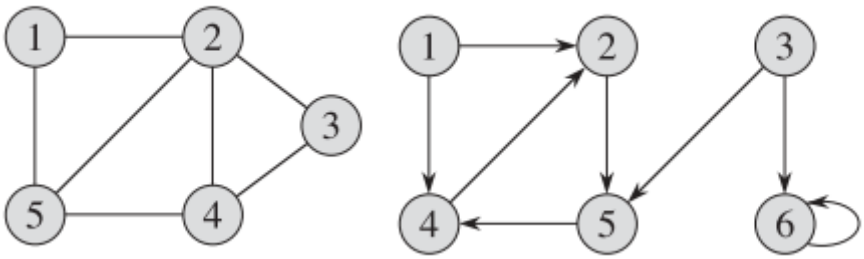
Berbagai studi menunjukkan bahwa TF-IDF tidak hanya sederhana dan efisien, tetapi juga memberikan performa tinggi dalam klasifikasi teks, termasuk pada tugas-tugas seperti deteksi ujaran kebencian, analisis ulasan, dan pemfilteran informasi berbasis konten (Wasid & Abdullah, 2025).

2.9 Graf

Graf merupakan struktur data yang digunakan untuk merepresentasikan hubungan antar objek. Secara formal, sebuah graf terdiri dari $G = (V, E)$ yang merupakan himpunan simpul (*vertices*) dan sisi (*edges*) yang menghubungkan pasangan simpul. Graf dapat bersifat terarah (*directed*) maupun tak terarah (*undirected*), dan dapat berbobot atau tidak berbobot. Dalam konteks ilmu komputer dan komputasi modern, graf banyak digunakan dalam berbagai aplikasi seperti jaringan sosial, pemetaan jalan, perambatan informasi, dan pemrosesan bahasa alami.

Menurut Cormen dkk. (2009) graf adalah struktur data yang terdiri dari sekumpulan simpul (*node* atau *vertex*) dan sekumpulan sisi (*edge*) yang menghubungkan pasangan simpul tersebut. Dalam representasinya, graf dapat disajikan menggunakan *adjacency list* maupun *adjacency matrix*. Pada *adjacency list*, setiap simpul memiliki daftar yang berisi simpul-simpul tetangganya, sehingga efisien untuk graf yang jarang (*sparse graph*). Sementara itu, *adjacency matrix* menggunakan matriks dua dimensi di mana setiap elemen

menunjukkan ada atau tidaknya sisi antara dua simpul; representasi ini cocok untuk graf yang padat (*dense graph*). Contoh graf dapat dilihat pada gambar 2.2.

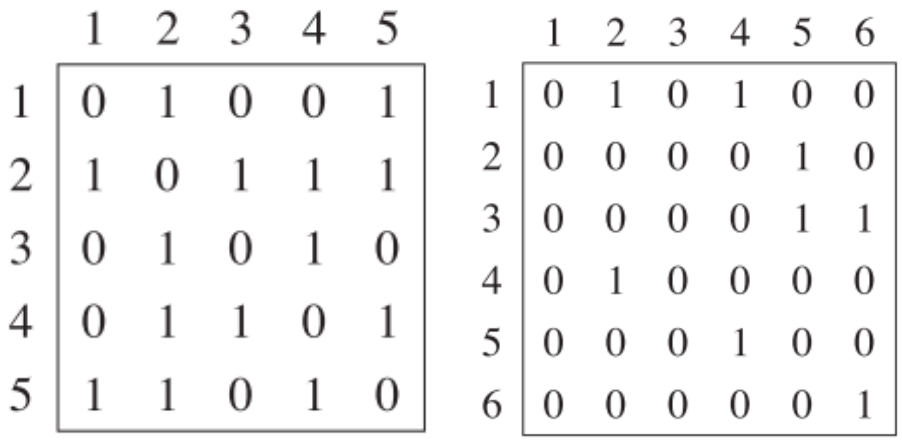


Gambar 2.2 Kiri Merupakan Graf Tidak Terarah (*Undirected Graph*) dan Kanan Graf Terarah (D*irected Graph/Digraph)* (Cormen dkk., 2009)

Jenis graf dapat dibedakan berdasarkan beberapa sifat graf berbobot (*weighted graph*) memiliki nilai (bobot) pada setiap sisi yang biasanya merepresentasikan jarak, biaya, atau kapasitas. Sebaliknya, graf tidak berbobot (*unweighted graph*) tidak memiliki nilai pada sisi-sisinya. Selain itu, graf juga dapat bersifat terarah (*directed graph/digraph*), di mana setiap sisi memiliki arah dari satu simpul ke simpul lain, atau tak terarah (*undirected graph*), di mana sisi hanya menghubungkan dua simpul tanpa arah tertentu (Cormen dkk., 2009).

2.10 *Adjacency Matrix* pada Graf

Adjacency matrix adalah salah satu metode representasi graf yang menggunakan sebuah matriks dua dimensi berukuran |V| × |V|, di mana |V| adalah jumlah simpul (*vertex*) pada graf. Setiap baris dan kolom pada matriks ini merepresentasikan satu simpul pada graf (Cormen dkk., 2009). Jika terdapat sebuah sisi dari simpul *u* ke simpul *v*, maka elemen pada baris *u* dan kolom *v* diisi dengan nilai 1 (atau nilai bobot jika graf berbobot), jika tidak ada sisi, maka diisi dengan 0. Untuk graf tak berarah, *adjacency matrix* bersifat simetris, sedangkan untuk graf berarah, matriksnya tidak harus simetris. Contoh *adjacency matrix* dapat dilihat pada gambar 2.3.



	1	2	3	4	5
1	0	1	0	0	1
2	1	0	1	1	1
3	0	1	0	1	0
4	0	1	1	0	1
5	1	1	0	1	0

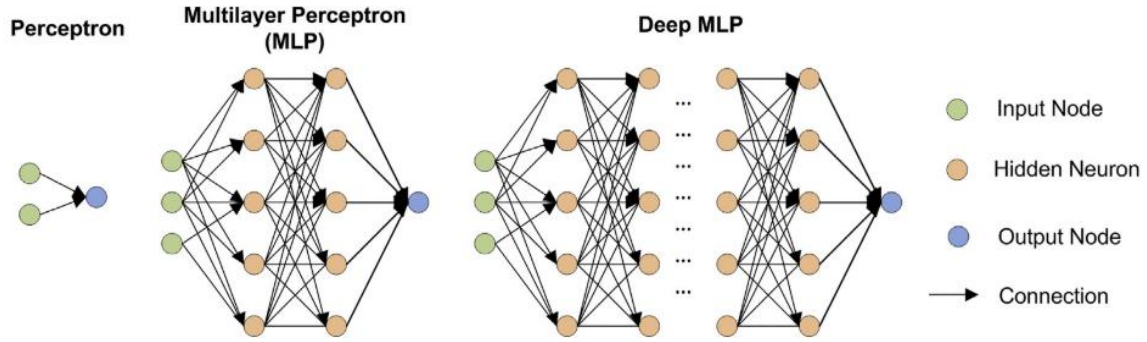
	1	2	3	4	5	6
1	0	1	0	1	0	0
2	0	0	0	0	1	0
3	0	0	0	0	1	1
4	0	1	0	0	0	0
5	0	0	0	1	0	0
6	0	0	0	0	0	1

Gambar 2.3 Representasi *Adjacency Matrix* dari Graf di Gambar 2.3 (Cormen dkk., 2009)

Kelebihan utama dari *adjacency matrix* adalah kemudahan dan kecepatan dalam mengecek keberadaan sebuah sisi antara dua simpul, yaitu hanya memerlukan waktu konstan O(1). Namun, representasi ini kurang efisien dalam hal penggunaan memori untuk graf yang jarang (*sparse graph*). Hal ini dikarenakan matriks tetap harus berukuran |V| × |V| meskipun banyak elemen yang bernilai nol (Cormen dkk., 2009).

2.11 Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan (JST) pertama kali diperkenalkan oleh Mcculloch & Pitts (1943) melalui model matematis sederhana yang mensimulasikan cara kerja neuron biologis dengan logika *Boolean*, menandai awal dari pendekatan komputasional terhadap pemrosesan informasi saraf. Sejak saat itu, konsep JST berkembang menjadi kerangka dasar dari berbagai sistem pembelajaran mesin modern. JST adalah sistem komputasi yang terinspirasi dari cara kerja otak manusia, terdiri dari jaringan *node* (neuron) yang tersusun dalam satu atau lebih lapisan (Grekousis, 2019). Lapisan ini terdiri dari lapisan input, satu atau lebih lapisan tersembunyi, dan lapisan output, di mana setiap neuron menerima input, mengalikan dengan bobot, dan menerapkan fungsi aktivasi untuk menghasilkan output, sehingga memungkinkan jaringan belajar secara otomatis dari data melalui proses pembobotan antar neuron (Indolia dkk., 2018). Visualisasi dari JST dapat dilihat pada gambar 2.4.



Gambar 2.4 Visualisasi Jaringan Syaraf Tiruan (Grekousis, 2019)

Salah satu pengembangan dari JST klasik adalah jaringan yang dirancang untuk memproses data tidak berstruktur seperti graf, yaitu *Graph Neural Network* (GNN). Dalam konteks ini, *Graph Convolutional Network* (GCN) memanfaatkan prinsip dasar JST seperti pembobotan, fungsi aktivasi, dan lapisan berlapis (*multi-layer*), namun dikombinasikan dengan struktur ketetanggaan antar simpul dalam graf. Selain itu, model GCN sering dilengkapi dengan lapisan *Multi-Layer Perceptron* (MLP) pada bagian akhir sebagai pengklasifikasi, menjadikannya tetap bagian dari paradigma jaringan saraf tiruan.

2.11.1 Fungsi Aktivasi

Fungsi aktivasi adalah komponen penting dalam jaringan saraf tiruan karena memungkinkan jaringan untuk mempelajari hubungan non-linear antar data. Fungsi ini mengubah output dari neuron berdasarkan nilai input terakumulasinya. Dalam studi bertajuk *"Bimodal Sentiment Analysis Based on a Pre-Trained Model and Masked Attention Fusion"* oleh Cai dkk. (2025), fungsi aktivasi seperti *Leaky* ReLU digunakan secara eksplisit dalam model BiGRU (*Bidirectional Gated Recurrent Unit*) untuk meningkatkan proses ekstraksi fitur token dalam tugas analisis sentimen berbasis multimodal. Penelitian ini menunjukkan bahwa pemilihan fungsi aktivasi yang tepat sangat krusial dalam memaksimalkan performa representasi dalam domain NLP, terutama saat menangani sinyal kompleks seperti teks dan suara. Setiap lapisan yang ada di jaringan syaraf tiruan, akan mengubah hasil kalkulasi neuron dengan fungsi aktivasi yang dapat dilihat pada persamaan 2.4.

$$y = \emptyset z_i \quad (2.4)$$

Keterangan:

$\emptyset$: Fungsi aktivasi dan

z_i : Nilai pada neuron ke i.

Berikut merupakan beberapa fungsi aktivasi yang umumnya digunakan dalam JST.

1. Fungsi Aktivasi Linear

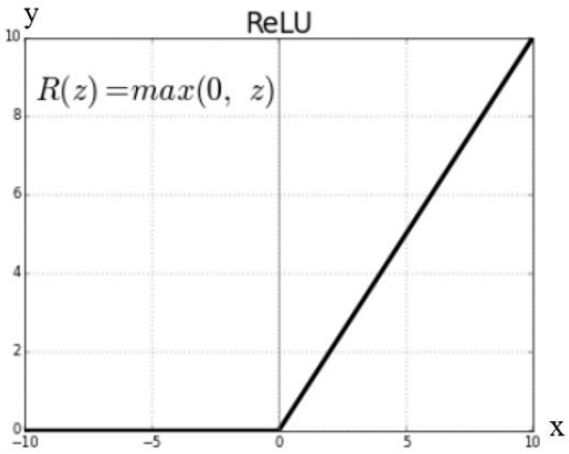
   Fungsi aktivasi linear merupakan suatu fungsi yang menghasilkan nilai output yang sama dengan nilai input. Rumus fungsi aktivasi linear dapat dilihat pada persamaan 2.5:

   $f(z_i) = z_i$ .....(2.5)

   Keterangan:

   z_i : Nilai pada *neuron* ke-i

2. Fungsi Aktivasi *Rectified Linear Unit* (ReLU)



Gambar 2.5 Fungsi Aktivasi ReLU

Fungsi aktivasi ReLU pertama kali diperkenalkan secara eksplisit dalam konteks deep learning oleh Glorot dkk. (2011) dalam penelitian mereka yang berjudul *Deep Sparse Rectifier Neural Networks*. Dalam studi ini, ReLU diperkenalkan sebagai alternatif yang lebih efisien dibandingkan fungsi aktivasi tradisional seperti *sigmoid* atau *tanh*, karena sifatnya yang sederhana dan kemampuannya dalam menghasilkan aktivasi yang jarang (*sparse activation*). Fungsi aktivasi ReLU akan mengubah masukan berupa bilangan negatif ke dalam nilai nol dan mengakumulasikan masukan berupa bilangan positif ke bilangan itu sendiri. Visualisasi fungsi aktivasi ReLU

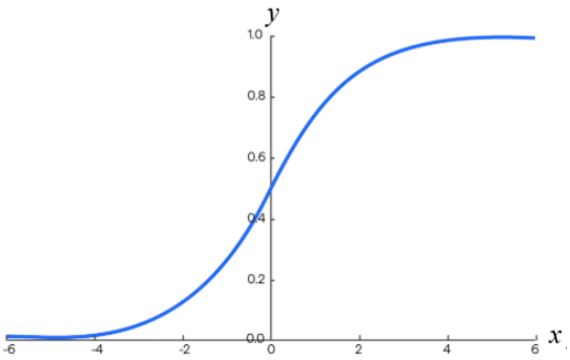
dapat dilihat pada gambar2.5 dan rumus untuk fungsi aktivasi ReLU dapat dituliskan pada persamaan 2.6.

$$f(z_i) = \max(0, z_i) \quad (2.6)$$

Keterangan:

z_i : Nilai pada neuron ke-i

3. Fungsi Aktivasi *SoftMax*



Gambar 2.6 Fungsi Aktivasi *Softmax*

Fungsi *softmax* merupakan fungsi aktivasi yang umum digunakan dalam model klasifikasi multi-kelas seperti *Multinomial Logistic Regression* dan jaringan saraf tiruan. Fungsi ini memperluas regresi logistik biner ke kasus multi-kelas dengan menghasilkan distribusi probabilitas atas seluruh kelas yang saling eksklusif. Bishop (1995) adalah tokoh yang memperkenalkan secara luas tentang fungsi *softmax* menjelaskan bahwa *softmax* menghitung probabilitas kelas dengan menormalisasi eksponensial dari skor logit, sehingga nilai keluaran menjadi positif dan totalnya satu. Hal ini menjadikan *softmax* ideal untuk memetakan vektor output menjadi distribusi probabilitas yang dapat ditafsirkan secara statistik sebagai peluang keanggotaan kelas. Visualisasi fungsi aktivasi *SoftMax* dapat dilihat pada gambar2.6 dan rumus fungsi aktivasi *softmax* dapat dilihat pada persamaan 2.7.

$$f(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}} \quad (2.7)$$

Keterangan:

z_i : Nilai pada *neuron* ke-i,

z_j : Nilai dari neuron ke-j dan

n : Jumlah total kelas.

2.11.2 *Cross-Entropy Loss*

Fungsi *loss* berperan penting dalam mengarahkan proses pembelajaran jaringan saraf tiruan. Salah satu fungsi *loss* yang paling banyak digunakan dalam tugas klasifikasi adalah *cross-entropy loss*, yang pertama kali diperkenalkan dalam konteks pelatihan jaringan saraf oleh Rumelhart dkk. (1986) dalam makalah klasik mereka mengenai algoritma *backpropagation*. Fungsi ini mengukur selisih antara distribusi probabilitas hasil prediksi dan label sebenarnya, dan secara efektif mendorong model untuk menghasilkan output probabilistik yang mendekati target. *Cross-entropy loss* terbukti sangat efektif untuk klasifikasi multi-kelas karena dapat memperbesar penalti untuk kesalahan prediksi yang jauh dari label sebenarnya. Rumus fungsi *Cross-entropy loss* dapat dilihat pada persamaan 2.8.

$$\mathcal{L} = -\sum_{i=1}^{C} y_1 \log(\hat{y}_i) \quad \text{.....(2.8)}$$

Keterangan:

C : Jumlah kelas

y_1 : Label sebenarnya (*ground truth*) dalam bentuk *one-hot encoding*

$\hat{y}_i$: Probabilitas prediksi dari model untuk kelas ke-i (*output* dari *softmax*).

2.11.3 *Batch Normalization*

Batch Normalization diperkenalkan oleh Ioffe & Szegedy (2015) sebagai teknik untuk mempercepat dan menstabilkan proses pelatihan jaringan neural dalam dengan mengurangi *internal covariate shift*, yaitu perubahan distribusi input di setiap lapisan selama pelatihan. Untuk *input* aktivasi $x = \{x^1, x^2, x^3, \dots, x^m\}$ dari sebuah *layer* dalam *mini-batch* berukuran m. Dengan menormalisasi input untuk setiap *mini-batch* agar memiliki rata-rata nol dan varians satu, teknik ini membantu model untuk konvergen lebih cepat dan memungkinkan penggunaan *learning rate* yang lebih tinggi tanpa menyebabkan ketidakstabilan. *Batch Normalization* juga bertindak sebagai bentuk regularisasi, yang dapat mengurangi kebutuhan akan teknik regularisasi lain seperti *dropout*.

2.11.4 *Dropout*

Dropout adalah salah satu teknik regularisasi yang paling populer dalam jaringan saraf tiruan. *Dropout* pertama kali diperkenalkan secara sistematis oleh Srivastava dkk. (2014). *Dropout* bekerja dengan cara menonaktifkan secara acak sejumlah unit neuron pada saat

pelatihan, sehingga mencegah model terlalu bergantung pada fitur tertentu dan memaksa penyebaran informasi ke berbagai jalur representasi dalam jaringan. Pendekatan ini secara tidak langsung mensimulasikan pelatihan pada banyak model jaringan yang berbeda dan menggabungkan hasilnya, sehingga membantu mengurangi *overfitting*. Ketika model digunakan untuk inferensi, semua neuron kembali diaktifkan dan bobotnya disesuaikan. Dalam formulasi matematisnya, *dropout* menerapkan masking dengan distribusi *Bernoulli* pada output dari suatu *layer*, di mana unit hanya aktif dengan probabilitas tertentu. *Dropout* dituliskan di persamaan 2.9.

$z = (W_x + b) \cdot r$ dengan $r \sim Bernoulli(p)$ .....(2.9)

Keterangan:

r : *Masking* vector (1 atau 0) dipilih secara acak untuk setiap unit dan

p : Probabilitas unit untuk dipertahankan.

2.11.5 *Optimizer AdamW*

Optimizer merupakan komponen penting dalam pelatihan jaringan syaraf tiruan karena berfungsi mengatur pembaruan bobot berdasarkan turunan fungsi *loss*. Salah satu *optimizer* adaptif yang banyak digunakan adalah *Adam* (*Adaptive Moment Estimation*), namun penelitian oleh Loshchilov & Hutter (2017) menunjukkan bahwa implementasi regularisasi L2 (*weight decay*) dalam *Adam* tidak setara dengan *weight decay* dalam *Stochastic Gradient Descent* (SGD). Untuk mengatasi hal ini, mereka mengusulkan *AdamW*, sebuah perbaikan dari *Adam* yang memisahkan secara eksplisit proses regularisasi dari langkah optimisasi utama, sehingga *weight decay* tidak tercampur dalam pembaruan gradien. Secara matematis, pembaruan parameter pada *AdamW* dilakukan dengan dua langkah terpisah dan dapat dituliskan dalam persamaan 2.10.

$\theta_{i+1} = \theta_t - \eta \cdot \hat{m}_t - \eta \cdot \lambda \cdot \theta_t$ .....(2.10)

Keterangan:

θ_t : Parameter pada iterasi ke-t,

$\hat{m}_t$: Estimasi gradien momen pertama yang sudah dikoreksi,

η : Koefisien *weight decay*

Berbeda dengan *Adam* klasik yang mencampurkan *weight decay* dalam perhitungan gradien *loss*, *AdamW* menerapkan *decay* langsung pada parameter, terlepas dari gradien *loss* itu sendiri. Pendekatan ini memungkinkan pemilihan nilai *weight decay* yang tidak bergantung pada setting *learning rate*, serta terbukti meningkatkan performa generalisasi model dalam berbagai eksperimen *image classification* dan NLP (Zhou dkk., 2024). Oleh karena itu, dalam penelitian ini, *AdamW* dipilih sebagai *optimizer* karena stabilitasnya dalam pelatihan model berbasis fitur TF-IDF yang bersifat spars dan berdimensi tinggi.

2.11.6 *Label Smoothing*

Label smoothing adalah teknik regularisasi yang diperkenalkan oleh Szegedy dkk. (2015) untuk meningkatkan kemampuan generalisasi model *neural network*. Dalam klasifikasi multi-kelas, target label biasanya direpresentasikan sebagai vektor *one-hot* (misalnya [0, 0, 1, 0]), yang mengasumsikan keyakinan penuh (100%) terhadap satu kelas. *Label smoothing* mengurangi keyakinan ini secara eksplisit dengan menyebarkan sebagian kecil dari probabilitas ke kelas lain. Jika y adalah vektor target *one-hot* dan $\in$ adalah faktor *smoothing*, maka label yang diubah y_{ls} dapat dituliskan dalam persamaan 2.11.

$$y_{ls} = (1 - \epsilon) \cdot y + \frac{\epsilon}{K} \quad \text{.....(2.11)}$$

Keterangan:

K : Jumlah kelas

$\in$: Nilai *smoothing* (misal 1)

Tujuan dari teknik ini adalah mengurangi *overconfidence* pada model terhadap satu kelas dan mendorong distribusi probabilitas yang lebih lembut, yang terbukti secara empiris dapat meningkatkan akurasi dan ketahanan terhadap *overfitting*. Dalam penelitian ini, *label smoothing* digunakan bersama fungsi *Cross Entropy Loss* untuk memperkuat kemampuan generalisasi model pada data sentimen yang memiliki distribusi label tidak seimbang.

2.11.7 *Residual Connection*

Residual connection merupakan inovatif yang pertama kali diperkenalkan oleh He dkk. (2015) melalui konsep *Residual Network* (ResNet). *Residual connection*

memungkinkan informasi untuk melompati satu atau lebih *layer* melalui mekanisme *shortcut* atau *identity mapping*. *Residual Connection* dapat dituliskan dalam persamaan 2.12.

$$y = \mathcal{F}(x, \{W_i\}) + x \quad \text{.....(2.12)}$$

Keterangan:

x : Input ke blok residual

$\mathcal{F}(x, \{W_i\})$: Fungsi *non-linear* yang biasa terdiri dari beberapa *layer* konvolusi dan aktivasi

y : Output yang diperoleh dengan menambah input secara langsung ke hasil transformasi $\mathcal{F}$

Ide utama dari *residual learning* adalah mengatasi degradasi performa pada jaringan yang semakin dalam, di mana akurasi mulai menurun karena kesulitan dalam mengoptimasi lapisan-lapisan dalam. Dengan menggunakan *residual connection*, gradien dapat dipropagasi lebih baik ke lapisan awal, sehingga mempercepat konvergensi dan mencegah masalah *vanishing gradient*. Dalam penelitian ini, *residual connection* digunakan pada beberapa *layer* GCN untuk meningkatkan stabilitas pelatihan serta mempertahankan representasi fitur antar *layer*.

2.11.8 *Scheduler* Pembelajaran*: Cosine Annealing Warm Restarts*

Cosine Annealing Warm Restarts, juga dikenal sebagai SGDR (*Stochastic Gradient Descent with Warm Restarts*), adalah teknik penjadwalan *learning rate* yang diperkenalkan oleh Loshchilov & Hutter (2016) untuk meningkatkan performa pelatihan jaringan saraf dalam. Teknik ini menggunakan fungsi *cosine annealing* untuk secara perlahan mengurangi *learning rate* hingga nilai minimum, lalu melakukan *restart* secara periodik dengan mengembalikan *learning rate* ke nilai awal. Tujuannya adalah agar model dapat melompat keluar dari *local minima* dan tetap menjaga dinamika pelatihan yang adaptif. Fungsi penjadwalan *learning rate* dalam SGDR dapat dituliskan dalam persamaan 2.13.

$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min})\left(1 + cos\left(\frac{T_{cur}}{T}\pi\right)\right) \quad \text{.....(2.13)}$$

Keterangan:

η_t : Learning rate pada iterasi ke-t,

η_{min} dan η_{max} : Batas bawah dan atas learning rate,

T_{cur} : Jumlah iterasi sejak restart terakhir,

T_i : Panjang siklus saat ini,

Dalam penelitian ini, digunakan *Cosine Annealing Warm Restarts* untuk menjaga kestabilan dan adaptivitas pembelajaran model GCN, serta mempercepat konvergensi tanpa mengandalkan *learning rate* yang konstan.

2.11.9 *Gradient Clipping*

Gradient clipping adalah teknik regularisasi yang digunakan untuk mengatasi masalah *exploding gradients* dalam pelatihan jaringan syaraf dalam. Teknik ini bekerja dengan membatasi (*clip*) besar norm dari gradien ketika melebihi nilai tertentu, untuk mencegah perubahan parameter yang ekstrem dan menjaga stabilitas pelatihan. Dalam praktiknya, *gradient clipping* dapat dituliskan dalam persamaan 2.14.

$$if\left|\left|g\right|\right|_2 > \theta, g \leftarrow \theta \cdot \frac{g}{||g||_2} \quad (2.14)$$

Keterangan:

g : Vektor gradien,

θ : Batas maksimum norm,

$||g||_2$: Norm Ecludian (L2) dari gradien

Penggunaan teknik ini pertama kali didiskusikan secara eksplisit oleh Chung dkk. (2014) dalam studi evaluasi terhadap model RNN dan LSTM. Mereka menunjukkan bahwa *gradient clipping* mencegah pelatihan menjadi tidak stabil, khususnya pada jaringan dalam atau sekuensial. Dalam penelitian ini, digunakan *clipping* dengan batas norm satu (1) untuk mencegah gradien yang terlalu besar saat melatih *Graph Convolutional Network* (GCN) agar proses pembelajaran tetap terkendali dan konvergen.

2.11.10 *Class Weights / Imbalanced*

Klasifikasi dengan data yang tidak seimbang (*imbalanced data*) merupakan masalah, model cenderung bias terhadap kelas mayoritas karena distribusi label yang tidak merata. Salah satu teknik paling umum untuk mengatasi hal ini adalah penerapan *class weights*, yaitu memberikan bobot lebih tinggi kepada kelas minoritas selama proses pelatihan. Dengan cara

ini, kesalahan prediksi terhadap kelas minoritas akan memberikan penalti yang lebih besar dalam fungsi *loss*, sehingga mendorong model untuk belajar memperhatikan representasi kelas tersebut. *Class weighting* dapat dituliskan dalam persamaan 2.15.

$$\mathcal{L} = -\sum_{i=1}^{N} w_{y_i} \cdot log(p_{y_i}) \quad \text{.....(2.15)}$$

Keterangan:

w_{y_i} : Bobot kelas untuk label ke-y_i,

p_{y_i} : Probabilitas yang diprediksi oleh model untuk label y_i,

N : Jumlah data

Studi oleh Shivashankar & Martini (2025) menunjukkan bahwa penggunaan strategi *class weighting* dapat secara efektif memperbaiki ketidakseimbangan label dalam tugas klasifikasi teknikal, menghasilkan peningkatan yang signifikan pada performa model, terutama terhadap akurasi kelas minoritas. Oleh karena itu, dalam penelitian ini juga diterapkan *class weighting* untuk menangani data sentimen yang tidak seimbang, di mana kelas positif, netral, dan negatif memiliki frekuensi kemunculan yang berbeda.

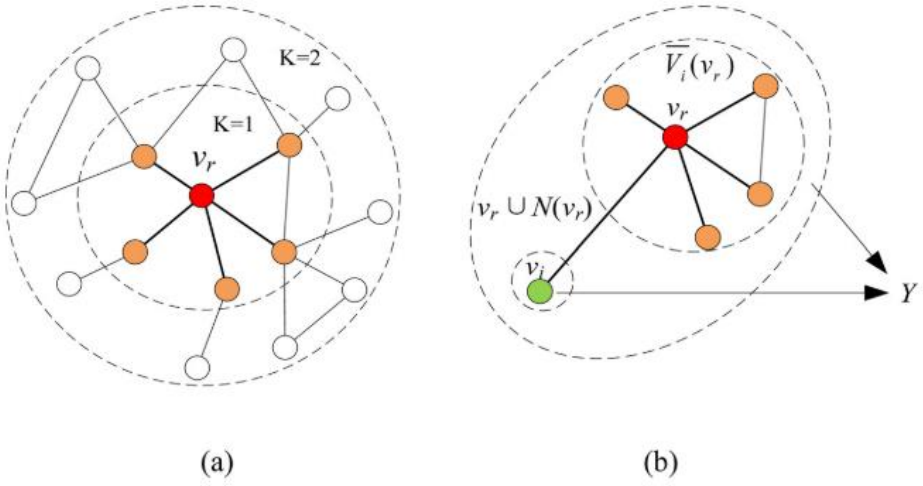
2.11.11 *Early stopping*

Early stopping adalah teknik regularisasi yang digunakan dalam pelatihan jaringan syaraf tiruan untuk mencegah *overfitting* terhadap data pelatihan. Teknik ini memantau performa model pada data validasi dan secara otomatis menghentikan pelatihan ketika performa validasi mulai menurun, meskipun *loss* pelatihan masih terus menurun. Dengan menghentikan pelatihan pada saat yang tepat, model dapat memperoleh generalisasi yang lebih baik terhadap data yang belum pernah dilihat sebelumnya. Menurut Prechelt (1997) *early stopping* dianggap sebagai bentuk regularisasi implisit karena membatasi jumlah pembaruan parameter yang bisa menyebabkan *overfit* terhadap noise dalam data pelatihan. Dalam praktiknya, pelatihan model dihentikan setelah sejumlah iterasi tanpa perbaikan signifikan pada metrik validasi, misalnya *F1-score* atau akurasi. Dalam penelitian ini, *early stopping* diterapkan untuk menghentikan pelatihan model secara otomatis ketika performa validasi tidak menunjukkan peningkatan selama beberapa *epoch*, guna menjaga efisiensi pelatihan dan menghindari *overfitting*.

2.12 *Graph Neural Network*

Graph Neural Network (GNN) adalah model jaringan saraf pertama yang secara eksplisit dirancang untuk memproses data berbasis graf, di mana relasi antar entitas direpresentasikan dalam bentuk simpul dan sisi. Model ini diperkenalkan oleh Scarselli dkk. (2009) sebagai pendekatan yang memungkinkan pembelajaran langsung dari struktur graf, baik dalam bentuk berarah maupun tak berarah, dengan mempertimbangkan informasi lokal dari *node* dan tetangganya melalui mekanisme propagasi pesan (*message passing*). GNN melakukan proses iteratif yang menggabungkan representasi fitur *node* dengan informasi dari tetangga terdekat hingga konvergen, dan menghasilkan output yang dapat digunakan untuk klasifikasi *node*, prediksi *edge*, atau tugas graf lainnya. Pendekatan ini menjadi fondasi utama bagi berbagai model graf modern seperti GCN, GraphSAGE, dan GAT, yang memanfaatkan kemampuan GNN dalam menangkap struktur dan topologi kompleks dari data tidak terstruktur.

Graph Neural Network (GNN) merupakan pendekatan pembelajaran mesin berbasis graf yang mampu secara efektif menangkap ketergantungan topologi dan atribut spasial antar *node*, menjadikannya sangat tepat untuk memodelkan struktur jaringan kompleks dan dinamis seperti dalam sistem komunikasi berbasis ruang, udara, dan darat (Liu dkk., 2025). GNN menggunakan *node* untuk merepresentasikan setiap simpul, dan *edge* untuk menggambarkan koneksi antar simpul, di mana masing-masing *node* memuat atribut seperti posisi, panjang, ketebalan, serta hubungan dengan objek lain. Sementara itu, *edge* merepresentasikan koneksi spasial maupun fungsional antar garis (Han dkk., 2025). GNN mengoperasikan pembelajaran melalui mekanisme *message passing*, di mana setiap *node* secara iteratif mengumpulkan informasi dari tetangganya melalui *edge*, menggabungkannya melalui fungsi agregasi, dan memperbarui representasi dirinya, sehingga memungkinkan propagasi informasi secara efisien dalam struktur graf (Sarkar dkk., 2023).



Gambar 2.7 Visualisasi Graf Dalam Proses Penggabungan Informasi Zhang dkk. (2022)

Gambar 2.7 merupakan visualisasi yang menggambarkan proses penggabungan informasi dalam jaringan graf, di mana sebuah *node* target memperoleh representasi yang lebih kaya melalui agregasi informasi dari tetangganya dalam radius tertentu (*multi-hop neighborhood*). Dalam konteks ini, *node* tetangga (baik yang langsung maupun tidak langsung) menyumbangkan fitur yang kemudian diproses melalui mekanisme agregasi. Informasi yang dikumpulkan tersebut digunakan untuk memprediksi label *node* target, dengan mempertimbangkan pula pengaruh kausal dari *node* lain yang relevan. Proses ini mencerminkan bagaimana struktur graf dan hubungan antar node memainkan peran penting dalam pembelajaran representasi dan klasifikasi pada data berbasis graf. Modul *message passing* dalam *Graph Neural Network* (GNN) secara umum melibatkan tiga tahap utama: pembuatan pesan antar *node*, agregasi pesan, dan pembaruan status *node*. Proses ini dituliskan dalam persamaan 2.16, persamaan 2.17 dan persamaan 2.18 (Ferriol-Galmés dkk., 2022).

$$M_{ij}^{*} = m\left(h_i^t, h_j^t, e_{ij}\right) \quad \text{.....(2.16)}$$

Keterangan:

h_i^t : Representasi fitur *node* i pada waktu ke – t

h_j^t : Representasi fitur *node* j tetangga dari i

e_{ij} : Fitur sisi/*edge* antara *node* i dan j

$m(\cdot)$: Fungsi *message function*, misalnya *neural network, concatenation* atau fungsi linier

$$M_i^{t+1} = aggr\left(M_{ij}^{*}\right) \quad (2.17)$$

Keterangan:

$aggr$: Fungsi agregasi seperti *mean*, *sum*, *max*. menggabungkan semua pesan masuk ke *node* i

M_{ij}^{*} : Fungsi *message* m menyandikan informasi antara *node* i dan tetangganya j

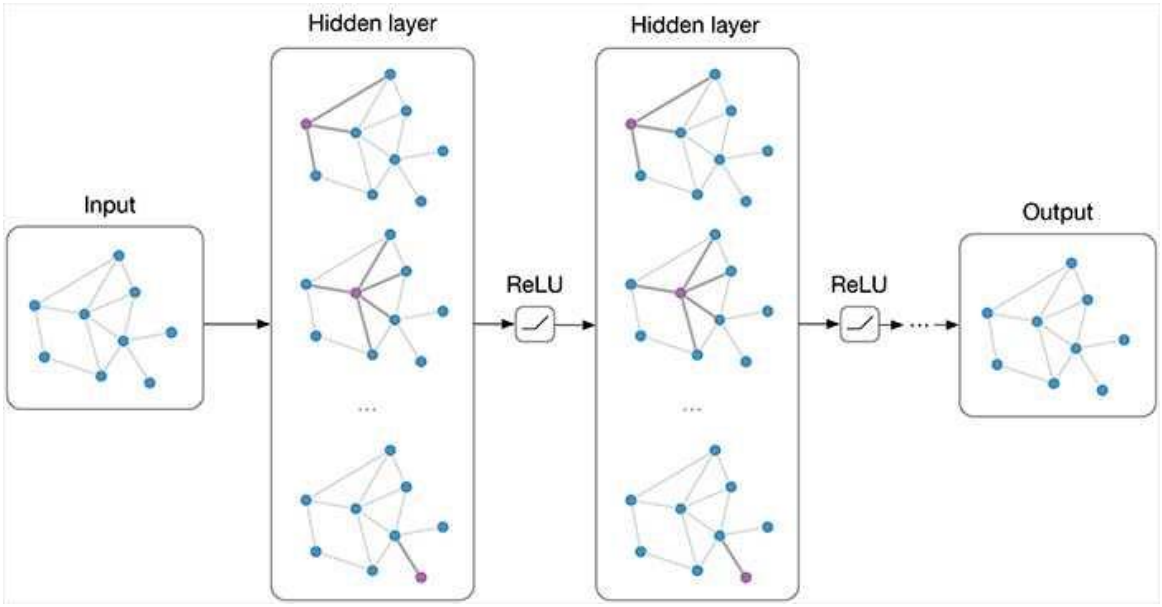
$$h_i^{t+1} = u(h_i^t, M_i^{t+1}) \quad (2.18)$$

Keterangan:

u : Fungsi *update* memperbarui representasi *node* i berdasarkan status sebelumnya dan pesan agregatnya. Fungsi u bisa berupa MLP (*Multi-Layer Perceptron*), GRU/LSTM unit (untuk *Gated* GNN), fungsi kombinasi linier.

h_j^t : Representasi fitur *node* j tetangga dari i

M_i^{t+1} : Hasil agregasi dari semua pesan masuk ke node i pada langkah waktu t+1



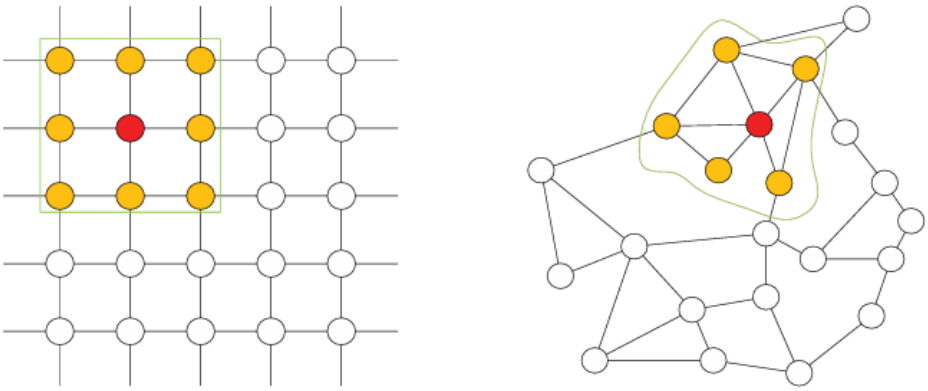
Gambar 2.8 Visualisasi GNN dalam JST (Phan dkk., 2023)

Graph Neural Network (GNN) dalam konteks analisis sentimen memungkinkan model untuk memahami konteks sentimen yang tersebar secara struktural dalam teks, dengan

mengubah dependensi sintaksis atau hubungan semantik antar kata menjadi struktur graf, sehingga memungkinkan klasifikasi opini yang lebih akurat dan kontekstual (Phan dkk., 2022). Oleh karena itu, penerapan GNN dalam analisis sentimen menjadi penting, terutama untuk meningkatkan pemahaman kontekstual terhadap opini pengguna yang tersebar dalam struktur kalimat kompleks, sehingga mendukung akurasi klasifikasi sentimen pada data teks secara signifikan. Visualisasi dari GNN pada JST dapat dilihat pada gambar 2.8.

2.13 *Graph Convolutional Network*

Graph Convolutional Network (GCN) memperluas operasi konvolusi tradisional ke *domain non-Euclidean* dengan memungkinkan setiap *node* dalam graf untuk mengagregasi dan mentransformasikan informasi dari tetangganya, sehingga menjadikannya efektif dalam tugas-tugas yang melibatkan data berbasis graf seperti klasifikasi *node* dan prediksi hubungan (Zhang dkk., 2019). *Graph Convolutional Network* (GCN) secara fundamental merupakan pendekatan pembelajaran transduktif, di mana model memanfaatkan informasi dari semua *node* (baik yang berlabel maupun tidak) selama proses pelatihan. Artinya, GCN tidak melakukan generalisasi ke *node* yang benar-benar baru (seperti pada *inductive learning*), tetapi belajar pada seluruh struktur graf secara menyeluruh. Pendekatan ini memungkinkan GCN untuk memanfaatkan hubungan topologis antar *node* untuk meningkatkan akurasi prediksi label, terutama dalam tugas klasifikasi *node* semi-supervised.



Gambar 2.9 Kiri CNN dan Kanan GCN (Zhang dkk., 2019).

Gambar 2.9 sisi kiri memperlihatkan bahwa pada data reguler, seperti citra, jumlah tetangga dari setiap *node* bersifat tetap dan beraturan, sehingga konvolusi dapat dilakukan dengan cara klasik. Namun pada sisi kanan struktur graf, setiap *node* memiliki jumlah

tetangga yang berbeda-beda, sehingga GCN menggunakan strategi konvolusi berbasis struktur graf lokal dari masing-masing *node* (Zhang dkk., 2019). GCN bekerja melalui proses propagasi, yaitu dengan mengagregasi informasi dari node tetangga melalui operasi konvolusi pada graf, sehingga setiap *node* dapat memperbarui representasinya berdasarkan fitur dirinya sendiri dan fitur dari *node* tetangganya secara iteratif pada setiap *layer* jaringan (Phan dkk., 2023).

Struktur dalam *Graph Convolutional Network* (GCN) direpresentasikan secara matematis menggunakan *adjacency matrix*, yaitu matriks persegi berukuran n x n, dimana n adalah jumlah *node* atau simpul. Elemen $A_{ij} = 1$ menunjukkan adanya hubungan(*edge*) dari simpul i ke simpul j, dan 0 jika tidak ada. Untuk memastikanbahwa setiap simpul dapat mempertahankan informasi dirinya sendiri dalam proses propagasi, maka digunakan *self-loop*, yaitu dengan penambahan hubungan dari setiap simpul ke dirinya sendiri. Diimplementasikan dengan menambahkan *identity matrix I* ke *adjency matrix* awal, dapat dituliskan dalam persamaan 2.19.

$$\tilde{A} = A + I \quad \text{.....(2.19)}$$

Keterangan:

A : Matriks ketetanggaan (*adjacency matrix*) asli berukuran $n \times n$

I : Matriks identitas untuk menambahkan *self-loop*

$\tilde{A}$: Matriks ketetanggaan yang telah ditambahkan *self-loop*, agar setiap simpul mempertahankan informasi dirinya sendiri

Normalisasi simetris terhadap *adjency matrix* tersebut dilakukan agar pembaruan informasi antar simpul tidak bergantung pada derajat atau jumlah tetangga suatu simpul. Normalisasi ini menggunakan *degree matrix D,* yaitu matriks diagonal yang menyimpan jumlah koneksi atau *degree* tiap simpul, dapat dituliskan dalam persamaan 2.20, GCN kemudian menggunakan *adjency matrix* ternomalisasi dituliskan dalam persamaan 2.21.

$$\widetilde{D}_{ii} = \sum_j \tilde{A}_{ij} \quad \text{.....(2.20)}$$

Keterangan:

$\tilde{A}_{ij}$: Elemen dari adjacency matrix $\tilde{A}$ yang sudah ditambahkan *self-loop*.

$\widetilde{D}_{ii}$: Elemen diagonal dari matriks derajat (degree matrix) hasil penjumlahan semua koneksi simpul i, termasuk *self-loop*.

$\sum_j \tilde{A}_{ij}$: Menjumlahkan semua hubungan (edge) dari simpul i ke tetangganya j, termasuk dirinya sendiri.

$$\hat{A} = \widetilde{D}^{-1/2}\tilde{A}\widetilde{D}^{-1/2} \quad \text{.....(2.21)}$$

Normalisasi ini digunakan setiap fitur simpul agar diperbarui secara proporsional terhadap bobot hubungan dengan tetangganya, tanpa terlalu mendominasi simpul yang banyak maupun sedikit koneksinya. Rumus utama propagasi dari *Graph Convolutional Network* (GCN) secara umum dituliskan dalam persamaan 2.22 (Kipf & Welling, 2016).

$$H^{(i+1)} = \sigma(\widetilde{D}^{-\frac{1}{2}}\tilde{A}\widetilde{D}^{-\frac{1}{2}}H^{(i)}\mathrm{W}^{(i)}) \quad \text{.....(2.22)}$$

Keterangan:

$\tilde{A} = A + I$: *Adjacency matrix* dengan *self-loop,* artinya setiap *node* juga melihat dirinya sendiri saat agregasi

$\widetilde{D}$: *Diagonal degree matrix* dari $\tilde{A}$, digunakan untuk normalisasi agar nilai agregasi tidak berat sebelah ke *node* dengan banyak tetangga

$H^{(i)}$: Representasi semua *node* di layer ke-i

$\mathrm{W}^{(i)}$: Matriks bobot yang dilatih, mengubah dimensi fitur

σ : Fungsi aktivasi (misal ReLU)

Salah satu permasalahan yang sering muncul dalam penggunaan *Graph Convolutional Networks* (GCNs) adalah fenomena *over-smoothing*, yaitu ketika representasi node menjadi terlalu homogen seiring bertambahnya jumlah lapisan (*depth*) pada model. Hal ini menyebabkan hilangnya informasi diskriminatif antar *node*, sehingga menurunkan performa klasifikasi. Penyebab utama *over-smoothing* adalah proses propagasi fitur yang berulang kali mencampurkan informasi dari tetangga tanpa kontrol, yang pada akhirnya membuat semua node memiliki *embedding* yang mirip satu sama lain. Menurut penelitian oleh Cai dkk. (2025), *over-smoothing* terjadi karena *embedding homogenization* yang disebabkan oleh *repeated neighborhood aggregation* dalam lapisan GCN. Mereka menekankan bahwa semakin dalam model GCN, semakin besar risiko representasi menjadi tidak terpisahkan antara kelas. Untuk mengatasi hal ini, berbagai strategi seperti penggunaan

residual connection, normalisasi, dan pembatasan jumlah lapisan digunakan untuk menjaga kualitas representasi node. *Pseudocode* untuk GCN dapat dilihat pada tabel 2.18.

Tabel 2.18 *Pseudocode* GCN *Layer*

```
GLOBAL VARIABLE: LAYER_UIDS = empty dictionary

FUNCTION get_layer_uid(layer_name):
    IF layer_name not in LAYER_UIDS:
        set LAYER_UIDS[layer_name] = 1
        RETURN 1
    ELSE:
        increment LAYER_UIDS[layer_name] by 1
        RETURN LAYER_UIDS[layer_name]

FUNCTION sparse_dropout(x, keep_prob, noise_shape):
    random_tensor = keep_prob + random_uniform(noise_shape)
    dropout_mask = floor(random_tensor) as boolean
    pre_out = retain_sparse(x, dropout_mask)
    RETURN pre_out * (1 / keep_prob)

FUNCTION dot(x, y, sparse):
    IF sparse:
        RETURN sparse_matrix_multiply(x, y)
    ELSE:
        RETURN matrix_multiply(x, y)


CLASS Layer:
    PROPERTIES:
        - name
        - vars (dictionary of trainable weights)
        - logging (boolean)
        - sparse_inputs (boolean)

    CONSTRUCTOR(kwargs):
        IF name not given:
            name = class_name + unique_id_from(get_layer_uid)
        SET logging from kwargs
        sparse_inputs = False

    METHOD _call(inputs):
        RETURN inputs

    METHOD __call__(inputs):
        WITH name_scope(name):
            IF logging AND not sparse_inputs:
                log_histogram("inputs", inputs)
            outputs = _call(inputs)
            IF logging:
                log_histogram("outputs", outputs)
        RETURN outputs

    METHOD _log_vars():
        FOR each var in vars:
            log_histogram(var)
```

```
CLASS Dense EXTENDS Layer:
    PROPERTIES:
        - dropout
        - act (activation function)
        - sparse_inputs
        - featureless
        - bias
        - num_features_nonzero (helper for sparse dropout)

    CONSTRUCTOR(input_dim,    output_dim,    placeholders,    dropout,
sparse_inputs, act, bias, featureless):
        CALL superclass constructor
        IF dropout is set:
            self.dropout = placeholders["dropout"]
        ELSE:
            self.dropout = 0

        Initialize weights as glorot(input_dim, output_dim)
        IF bias is True:
            Initialize bias as zeros(output_dim)

        IF logging enabled:
            log_vars()

    METHOD _call(inputs):
        x = inputs

        IF sparse_inputs:
            x = sparse_dropout(x, 1 - dropout, num_features_nonzero)
        ELSE:
            x = dropout(x, rate = 1 - dropout)

        output = dot(x, weights, sparse=sparse_inputs)

        IF bias:
            output += bias

        RETURN act(output)


CLASS GraphConvolution EXTENDS Layer:
    PROPERTIES:
        - dropout
        - act
        - support (list of adjacency matrices)
        - sparse_inputs
        - featureless
        - bias
        - num_features_nonzero

    CONSTRUCTOR(input_dim,    output_dim,    placeholders,    dropout,
sparse_inputs, act, bias, featureless):
        CALL superclass constructor
        IF dropout is set:
            self.dropout = placeholders["dropout"]
        ELSE:
            self.dropout = 0

        support = placeholders["support"]
```

```
        FOR each support matrix:
            Initialize weight matrix (input_dim, output_dim) with glorot
        IF bias:
            Initialize bias as zeros(output_dim)

        IF logging enabled:
            log_vars()

    METHOD _call(inputs):
        x = inputs

        IF sparse_inputs:
            x = sparse_dropout(x, 1 - dropout, num_features_nonzero)
        ELSE:
            x = dropout(x, rate = 1 - dropout)

        supports = empty list

        FOR each support matrix i:
            IF not featureless:
                pre_sup = dot(x, weight_i, sparse=sparse_inputs)
            ELSE:
                pre_sup = weight_i
            support_result = dot(support[i], pre_sup, sparse=True)
            append support_result to supports

        output = sum(all supports)

        IF bias:
            output += bias

        RETURN act(output)
```

Proses propagasi pada *Graph Convolutional Network* (GCN) dapat diformulasikan dengan persamaan $H^{(i+1)} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(i)}\mathrm{W}^{(i)})$, di mana $H^{(i)}$ merepresentasikan fitur *node* pada *layer* ke-l, $\mathrm{W}^{(i)}$adalah matriks bobot yang dapat dilatih, $\tilde{A}$ merupakan matriks ketetanggaan yang telah dinormalisasi, σ adalah fungsi aktivasi non-linear seperti ReLU. Implementasi persamaan tersebut tercermin dalam *pseudocode* kelas *GraphConvolution*. Pada bagian `pre_sup = dot(x, weight_i, ...)`, dilakukan transformasi fitur *node* yang sesuai dengan operasi $H^{(i)}\mathrm{W}^{(i)}$. Selanjutnya, baris `support_result = dot(support[i], pre_sup, sparse=True)` merepresentasikan tahap agregasi informasi dari *node* tetangga, yaitu operasi $\tilde{A}(H^{(i)}\mathrm{W}^{(i)})$. Apabila terdapat lebih dari satu matriks *support*, maka hasil agregasi dijumlahkan dengan `output = sum(all supports)`, sesuai dengan $\sum_i H^{(i)}\mathrm{W}^{(i)}$. Kemudian, bias ditambahkan melalui `output += bias`, dan tahap terakhir adalah penerapan fungsi aktivasi `act(output)` yang sesuai dengan σ(·). Dengan demikian, *pseudocode* tersebut merupakan bentuk implementasi langsung dari rumus propagasi GCN, yang

menggabungkan transformasi fitur, agregasi tetangga, penambahan bias, serta aktivasi non-linear.

Program ini membangun struktur dasar untuk *Graph Convolutional Network* (GCN) dengan *TensorFlow*. Pertama, ada fungsi `get_layer_uid` yang memberikan ID unik untuk setiap *layer* agar nama *layer* tidak bentrok. Fungsi `sparse_dropout` digunakan untuk melakukan *dropout* pada tensor yang berbentuk *sparse*, sementara fungsi `dot` berfungsi sebagai pembungkus untuk operasi perkalian matriks, baik *dense* maupun *sparse*. Selanjutnya, terdapat kelas dasar layer yang menjadi *blueprint* semua *layer*. Kelas ini mengatur nama *layer*, variabel yang dimiliki, serta *logging* menggunakan *TensorBoard*. Dua kelas turunan utama dari *layer* adalah *dense* dan *GraphConvolution*. Kelas *Dense* merepresentasikan *fully connected layer* standar, dengan dukungan untuk *dropout*, aktivasi non-linear, serta opsi bias. Kelas *GraphConvolution* adalah inti dari GCN, di mana input fitur diproses bersama dengan matriks *adjacency* (*support*) untuk melakukan konvolusi pada struktur graf. Prosesnya melibatkan *dropout* pada input, perkalian bobot dengan input (atau langsung bobot jika tanpa fitur), lalu dikalikan lagi dengan matriks *adjacency*. Hasil dari semua support dijumlahkan, ditambah bias jika ada, lalu dilewatkan ke fungsi aktivasi.

2.14 *Cosine Similarity* dan Pembentukan Graf

Cosine similarity merupakan metode umum dalam pembandingan kemiripan antar teks yang mengukur sudut kosinus antara dua vektor dalam ruang berdimensi tinggi, dan sering digunakan dalam representasi teks serta pencocokan dokumen dalam berbagai aplikasi *text mining* (Ding dkk., 2024). Secara umum rumus *cosine similarity* dituliskan dalam persamaan 2.23.

$$\cos(\theta) = \frac{\vec{A}\cdot\vec{B}}{||\vec{A}||\cdot||\vec{B}||} \quad \text{.....(2.23)}$$

Keterangan:

$\vec{A}\cdot\vec{B}$: Hasil perkalian *dot product* antara vektor A dan B

$||\vec{A}||$: Norma (panjang) dari vektor A

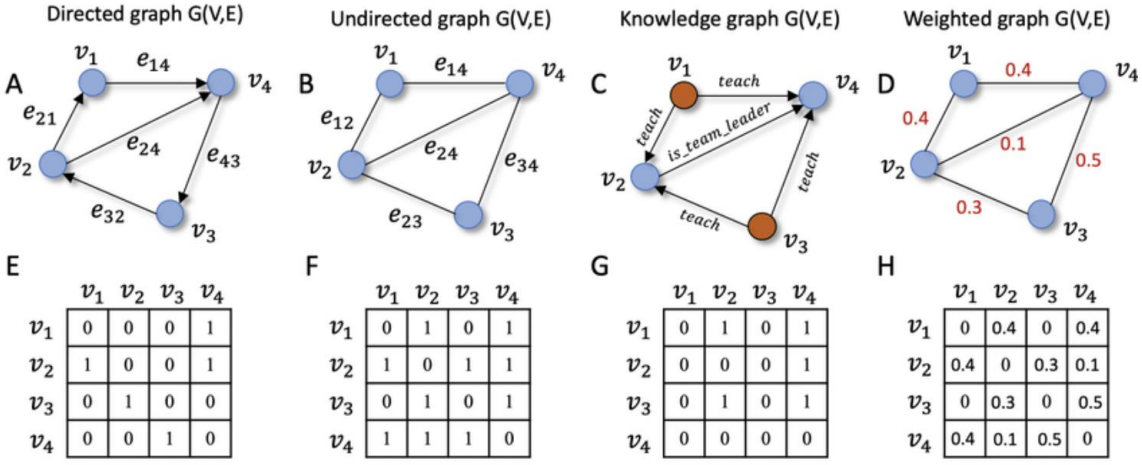
$||\vec{B}||$: Norma dari vektor B

Representasi teks berbasis TF-IDF menggunakan *cosine similarity* untuk mengukur tingkat kemiripan semantik antara dua dokumen. Kemiripan ini kemudian dimanfaatkan untuk membentuk struktur graf dokumen, di mana setiap dokumen dihubungkan dengan dokumen lain yang memiliki skor kemiripan tertinggi di atas ambang batas tertentu.

2.15 Pembuatan Graf Menggunakan *Adjacency Matrix*

Adjacency matrix memiliki pengaruh langsung terhadap kemampuan model GCN dalam menyebarkan dan menangkap informasi structural (Xiang dkk., 2025). *Adjacency matrix* berperan sebagai representasi hubungan antar *node* dalam graf, yang mengatur bagaimana informasi mengalir antar simpul pada tiap lapisan GCN.

Gambar 2.10 merupakan visualisasi dari matriks yang biasanya direpresentasikan dalam bentuk matriks biner atau berbobot, di mana entri A_{ij} menunjukkan keberadaan dan kekuatan koneksi antara *node* i dan *node* j. Pada penelitian ini, *adjacency matrix* dibentuk berdasarkan kemiripan *cosine* antar vektor TF-IDF, sehingga hubungan antar dokumen dalam graf mencerminkan kedekatan semantik antar teks. Matriks ini kemudian digunakan dalam proses propagasi fitur pada GCN untuk memperkuat representasi dokumen berdasarkan struktur global dari graf.



Gambar 2.10 Visualisasi Representasi Graf Menggunakan *Adjacency Matrix*

2.16 Evaluasi Model Klasifikasi

Evaluasi performa merupakan tahapan krusial dalam pengembangan model *machine learning* yang bertujuan untuk mengukur seberapa baik model memahami atau memprediksi

data. Dalam tugas klasifikasi, berbagai metrik digunakan untuk menilai kualitas prediksi model, di antaranya akurasi, *precision*, *recall*, dan *F1-score*. Metrik-metrik tersebut dihitung berdasarkan *confusion matrix,* yaitu sebuah tabel yang menunjukkan jumlah prediksi yang benar dan salah untuk setiap kelas. Pada klasifikasi multikelas seperti sentimen positif, netral, dan negatif, *confusion matrix* diperluas untuk mencakup tiga label, dan setiap sel merepresentasikan prediksi terhadap satu kelas terhadap label sebenarnya. Komponen dasar dari *confusion matrix* mencakup *True Positive* (TP), *False Positive (*FP), *True Negative* (TN), dan *False Negative* (FN), yang menjadi dasar untuk menghitung metrik-metrik evaluasi lainnya seperti akurasi dan *F1-score* (Sokolova & Lapalme, 2009). Rumus dari metrik akurasi dapat dilihat pada persamaan 2.24 dan gambar *confusion matrix* dapat dilihat pada tabel 2.19.

Tabel 2.19 Tabel *Confusion Matrix*

		Kelas sebenarnya	
		Positive	*Negative*
Prediksi	*Positive*	*True Positive*(TP)	*False Positive*(FP)
	Negative	*False Negative*(FP)	*True Negative*(TN)

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.24)$$

Precision dan *recall* merupakan dua metrik penting yang diperoleh dari *confusion matrix* dan digunakan secara luas untuk mengevaluasi performa model klasifikasi, khususnya pada data yang tidak seimbang. *Precision* mengukur seberapa banyak prediksi positif yang benar-benar relevan, sedangkan *recall* menilai seberapa besar bagian dari data positif yang berhasil dikenali oleh model. Kedua metrik ini memberikan pemahaman yang lebih dalam terhadap keseimbangan antara kesalahan tipe I (*false positive*) dan tipe II (*false negative*), menjadikannya esensial dalam berbagai aplikasi seperti pendeteksian objek, diagnosis medis, dan sistem deteksi anomali (Soyak dkk., 2025). Rumus persamaan dari metrik *Precision* dan *Recall* dapat dilihat pada persamaan 2.25 dan persamaan 2.26.

$$Precision = \frac{TP}{TP+FP} \quad (2.25)$$

$$Recall = \frac{TP}{TP+FN} \quad (2.26)$$

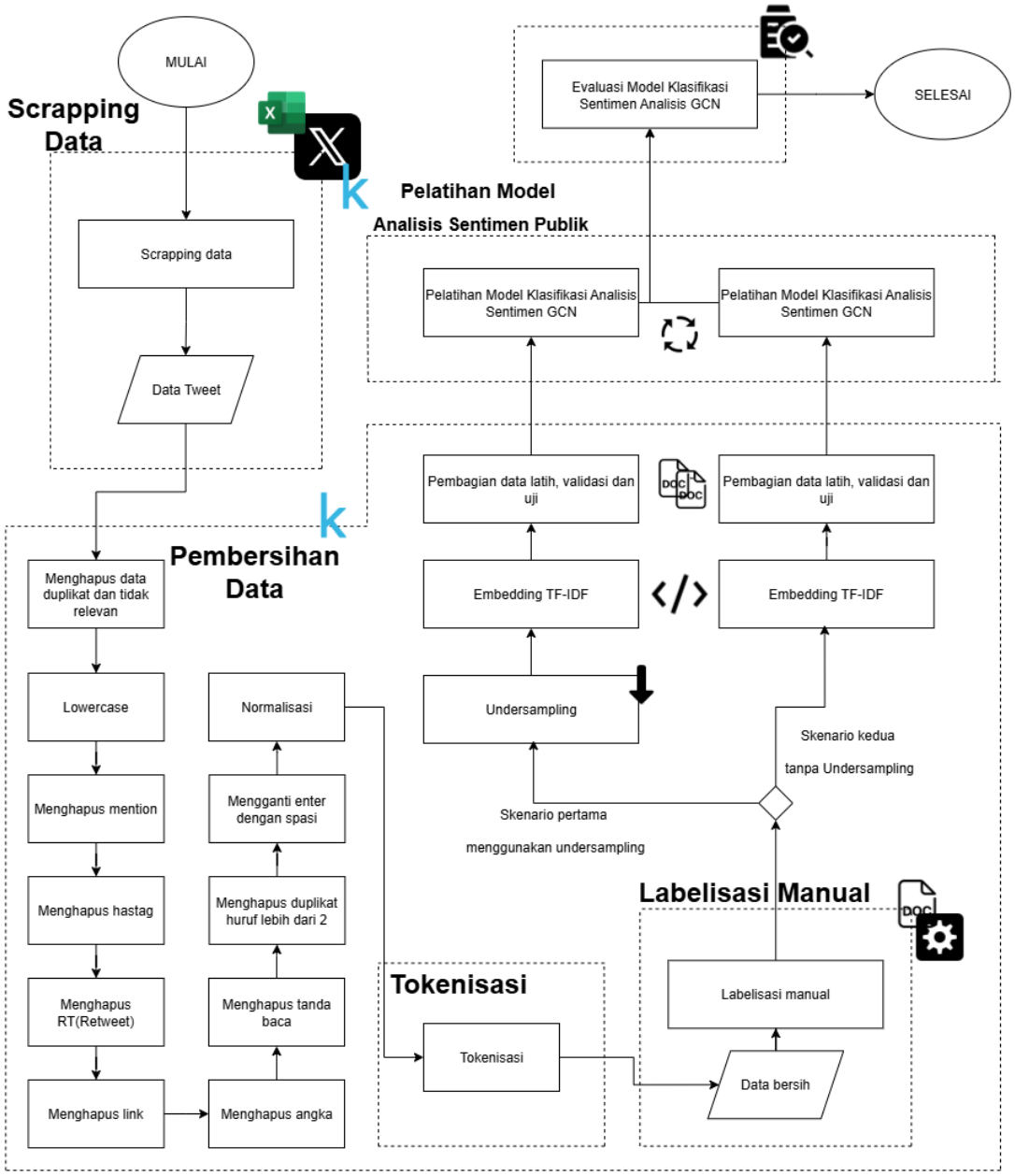
F1-score merupakan metrik evaluasi yang digunakan untuk menyeimbangkan *precision* dan *recall* dalam tugas klasifikasi, terutama saat menghadapi ketidakseimbangan

kelas. Sebagai rata-rata harmonik dari *precision* dan *recall*, *F1-score* memberikan gambaran yang lebih adil terhadap kinerja model ketika distribusi kelas tidak simetris. Dalam berbagai penelitian, *F1-score* dipandang sebagai indikator yang lebih informatif dibanding akurasi saja, karena tetap mempertimbangkan kesalahan tipe I dan tipe II secara bersamaan dalam pengambilan keputusan klasifikasi (Nguyen-Duc dkk., 2025). Rumus persamaan *F1-Score* dapat dilihat pada persamaan 2.27.

$$F1 - score = 2\frac{Precision}{Recall} \quad \text{.....(2.27)}$$

BAB III
METODE PENELITIAN

Bab ini membahas mengenai metodologi penelitian yang dilakukan pada penelitian Analisis Sentimen Publik untuk Makan Bergizi Gratis pada Media Sosial X (*Twitter*) menggunakan GCN dengan pendeketan TF-IDF. Terdapat dua skenario, skenario pertama menggunakan teknik *undersampling* dan skenario kedua tanpa teknik *undersampling.* Penelitian klasifikasi sentimen publik dengan model GCN dilakukan melalui beberapa tahapan yang akan digambarkan dengan diagram pada gambar 3.1.



Gambar 3.1 Proses Tahapan Penelitian

3.1 *Scrapping Data*

Data hasil *scrapping* yang dihasilkan untuk penelitian ini adalah data berupa cuitan dari media sosial X pada rentang waktu tanggal 1 November 2024 hingga 31 Maret 2025. Data diambil menggunakan sistem pencarian dengan kata kunci yaitu "makan bergizi gratis" dan "makan siang gratis" dengan filter Bahasa Indonesia (id). Dataset terdiri dari 15 kolom yakni *conversation_id, created_at, favorite_count, full_text, id_str, image_url, in_reply_to_screen_name, lang, location, quote_count, reply_count, retweet_count, tweet_url, user_id_str, username*. Jumlah dataset keseluruhan sebanyak 9.249 baris. Namun hanya kolom full_text yang akan dipakai untuk penelitian, maka kolom lain akan dihapus karena tidak digunakan. Contoh hasil data pengumpulan untuk topik makan bergizi gratis dapat dilihat tabel 3.1.

Tabel 3.1 Tabel Contoh Hasil *Scrapping Data*

No	Text
1	Kepala BP Taskin Budiman Sudjatmiko mengatakan Program Makan Bergizi Gratis (MBG) merupakan salah satu upaya yang dilakukan Presiden Prabowo Subianto untuk mengentaskan kemiskinan. Ria SW #dood Bandung Amorim #SquidGame2 Kurir Aceh https://t.co/P7fXx0hejo
2	@usabperning_ @aniesbaswedan @feryfarhati Makanya kamu ini berhak mendapt makan siang gratis biar melek otaknya
3	Mana makan siang itu? Yg katanya buah sayur susu tp diganti sm daon kelor plus ultramimi wkwkwk lawak. Apa blg maksi gratis klo lahan singkong n jagungnya sukses? NO NO NO NO TIDAK SEMUDAH ITU FERGUSO! Bro pikir ini minecraft kali yak
4	@worksfess Kebanyakan liat medsos.. padahal makan siang gratis cuma ada dikampanye..
5	@rendi078 @bublenjun @fIoyinn Kocak lu itu seminar bukan seminar sembarangan sekelas anak kampus bege. Pikiran dan dunia lu noh sesempit uji coba makan siang gratis dan bagi2 susu gratis aowkaokwo menolak pengetahuan
...	...
...	...
...	...
9248	@okkymadasari Maka dari itu.. Berharap bergizi mah jauhhhh Program makan siang gratis itu tujuannya hanya untuk MENCEGAH KELAPARAN
9249	Kepala BP Taskin Budiman Sudjatmiko mengatakan Program Makan Bergizi Gratis (MBG) merupakan salah satu upaya yang dilakukan Presiden Prabowo Subianto untuk mengentaskan kemiskinan. Ria SW #dood Bandung Amorim #SquidGame2 Kurir Aceh https://t.co/P7fXx0hejo

3.2 Pembersihan Data

Pembersihan data merupakan tahap dalam pengolahan data yang bertujuan untuk membersihkan, mengubah, dan menyiapkan data mentah agar siap digunakan dalam proses analisis atau pemodelan. Tahapan yang dilakukan dalam Pembersihan data pada penelitian ini yaitu data cleaning (*lowercase*, menghapus *mention*, menghapus *hastag*, menghapus RT (*retweet*), menghapus *link*, menghapus angka, menghapus simbol dan tanda baca, menghapus duplikat huruf lebih dari dua, mengganti enter dengan spasi), normalisasi (memperbaiki salah kata dan singkatan). Tahap *stopword removal* dan *lemmezation* tidak dilakukan karena bisa menghilangkan konteks dan mengubah sentimen dari kalimat.

Hasil dataset yang didapat sebelumnya malalui tahap pembersihan awal, pembersihan data meliputi menghilangkan duplikat dan data yang tidak relevan. Proses pembersihan data tidak relevan ini dilakukan dengan cara membersihkan data yang tidak mengandung kata kunci penelitian. Sedangkan proses pembersihan data duplikat dilakukan dengan cara menghapus duplikasi jika terdapat 70% kesamaan dalam data cuitan. Setelah pembersihan awal ini, data yang tersisa adalah 5.979 baris dari sebelumnya sebanyak 9.249 baris. Contoh data duplikat atau mirip dapat dilihat pada tabel 3.2.

Tabel 3.2 Cuitan Duplikat atau Mirip

No	full_text
1	Program Makan Bergizi Gratis (MBG) di Papua sangat membantu anak-anak mendapatkan asupan gizi yang sehat! @kompascom #MakanBergiziGratis #MBG #PapuaSehat #GiziAnak #PendidikanSehat #DukungMBG https://t.co/Kf8iWGuTYf
2	Program Makan Bergizi Gratis (MBG) di Papua sangat membantu anak-anak mendapatkan asupan gizi yang diperlukan untuk tumbuh sehat dan optimal. @kompascom #MakanBergiziGratis #MBG #PapuaSehat #GiziAnak #PendidikanSehat #DukungMBG https://t.co/ZhUw4Y5j31
3	Program makan bergizi gratis adalah bagian dari upaya untuk mewujudkan Indonesia yang bebas dari masalah gizi buruk dan stunting. #MakanBergiziBangunNegeri https://t.co/KxCBh7Vydw
4	Program makan bergizi gratis adalah bagian dari upaya untuk mewujudkan Indonesia yang bebas dari masalah gizi buruk dan stunting. #MakanBergiziBangunNegeri https://t.co/z7b61HSfVY
5	Program makan bergizi gratis adalah investasi terbaik untuk masa depan bangsa. #MakanBergiziBangunNegeri
6	Program makan bergizi gratis adalah investasi terbaik untuk masa depan Indonesia yang lebih cerah. #MakanBergiziBangunNegeri

Pembersihan data masih memiliki banyak data duplikat dimana banyak data yang memiliki kemiripan. Contohnya dapat dilihat pada tabel 3.2, baris 1 mirip dengan baris 2, baris 3 mirip dengan baris 4, baris 5 mirip dengan baris 6. Data ini harus dihapus untuk menyeimbangkan dataset juga mengurangi *noise* karena tidak memberikan banyak informasi. Kemudian diproses untuk membersihkan data teks dari elemen-elemen yang tidak relevan. Seperti *lowercase*, menghapus *mention* (@username), *hashtag* (#), *retweet* (RT), tautan (*link*), angka, simbol dan tanda baca, serta melakukan normalisasi teks dengan menghapus duplikasi huruf lebih dari dua, serta mengganti karakter baris baru (*enter*) dengan spasi agar data menjadi lebih konsisten dan siap untuk proses ke tahap selanjutnya. Setelah pembersihan awal ini, data yang tersisa adalah 5.979 baris, artinya banyak data yang mirip atau duplikat sebanyak 3.270 baris data yang memiliki kemiripan dengan data lainnya dan data yang tidak relevan.

Tabel 3.3 Data Sebelum dan Sesudah Pembersihan Data

Sebelum Tahap Pembersihan Data	Setelah Tahap Pembersihan Data
@KotaNusantara Nutrisi yang baik bikin anak Indonesia siap bersaing di masa depan semua berkat Makan Bergizi Gratis	nutrisi yang baik bikin anak indonesia siap bersaing di masa depan semua berkat makan bergizi gratis
@PNS_Ababil Akwoakwoakowak MAMAM tuh Program Asbun makan banyak duit	ababil akwoakwoakowak makan itu program asal bunyi makan banyak uang
Ketua Komisi XI DPR: Makan Bergizi Gratis Peluang UMKK Naik Kelas @DPR_RI https://t.co/e9R8XmjlNW #dprri	ketua komisi xi dewan perwakilan rakyat makan bergizi gratis peluang umkk naik kelas ri

Tabel 3.3. merupakan data hasil pembersihan dari elemen-elemen yang tidak relevan, seperti *lowercase*, menghapus *mention* (@username), *hashtag* (#), *retweet* (RT), tautan (*link*), angka, simbol dan tanda baca, serta melakukan normalisasi teks dengan menghapus duplikasi huruf lebih dari dua, serta mengganti karakter baris baru (*enter*) dengan spasi kemudian dilakukan proses tokenisasi agar bisa diolah oleh TF-IDF.

3.2.1 Tokenisasi

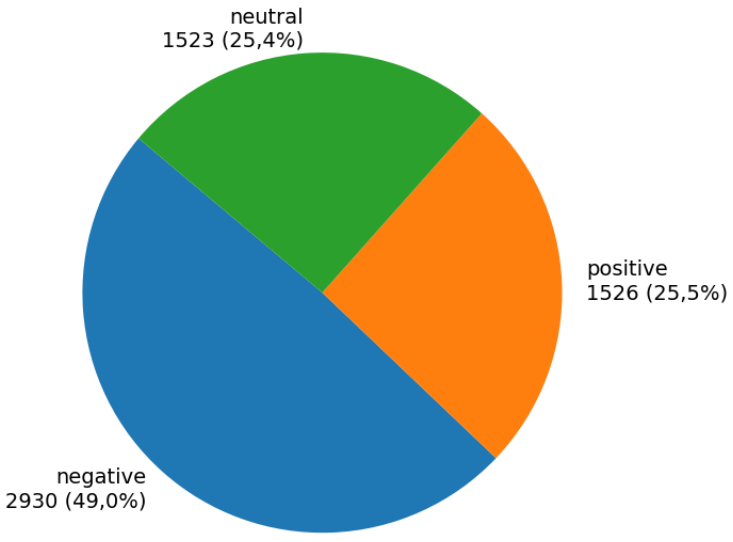
Tabel 3.4 menunjukkan hasil proses tokenisasi terhadap data teks yang telah melalui tahap pembersihan. Pada kolom “Sebelum Tahap Tokenisasi” terlihat kalimat utuh hasil dari tahap normalisasi, seperti penghapusan karakter tidak relevan, simbol, angka, serta duplikasi huruf dan karakter baris baru. Setelah itu, teks diproses menggunakan teknik tokenisasi, yang

memecah kalimat menjadi satuan kata-kata terpisah atau token. Hasil dari tokenisasi ditampilkan pada kolom “Setelah Tahap Tokenisasi”, berupa daftar kata dalam bentuk *array*. Proses ini bertujuan agar setiap kata dapat dianalisis secara individual pada tahap penghitungan bobot menggunakan TF-IDF. Token-token tersebut menjadi representasi dasar dari teks yang akan diolah dalam tahap selanjutnya dalam analisis data.

Tabel 3.4 Data Sebelum dan Sesudah Tokenisasi

Sebelum Tahap Tokenisasi	Setelah Tahap Tokenisasi
nutrisi yang baik bikin anak indonesia siap bersaing di masa depan semua berkat makan bergizi gratis	[“nutrisi”, “yang”, “baik”, “bikin”, “anak”, “Indonesia”, “siap”, “bersaing”, “di”, “masa”, “depan”, “semua”, “berkat”, “makan”, “bergizi”, “gratis”]
ababil akwoakwoakowak makan itu program asal bunyi makan banyak uang	[“ababil”, “akwoakwoakowak”, “makan”, “itu”, “program”, “asal”, “bunyi”, “makan”, “banyak”, “uang”]
ketua komisi xi dewan perwakilan rakyat makan bergizi gratis peluang umkk naik kelas ri	[“ketua”, “komisi”, “xi”, “dewan”, “perwakilan”, “rakyat”, “makan”, “bergizi”, “gratis”, “peluang”, “umkk”, “naik”, “kelas”, “ri”]

3.2.3 Labelisasi Manual



Gambar 3.2 Hasil Distribusi Sentimen Labelisasi Manual

Proses labelisasi dilakukan untuk menentukan polaritas sentimen dari setiap cuitan terhadap program “Makan Bergizi Gratis” di platform media sosial X. Kategori sentimen dibagi ke dalam tiga kelas, yaitu positif, netral, dan negatif. Untuk efisiensi dan konsistensi,

pelabelan dilakukan secara manual. Data yang didapat dari tahap labelisasi manual ini adalah 5.979 baris yang telah dilabeli, dimana positif sebanyak 1.526 baris, netral sebanyak 1.523 baris dan negatif sebanyak 2.930 baris. Hasil distribusi labelisasi dapat dillihat pada gambar 3.2.

3.2.4 *Undersampling*

Undersampling dilakukan untuk menyeimbangkan distribusi kelas dalam data. Keputusan ini diambil karena dataset mengandung banyak *noise*, dan pemilihan data dilakukan dengan strategi *undersampling* berbasis pemilihan kalimat terpanjang dari masing-masing kelas yang akan di *undersampling*. Pendekatan ini didasarkan pada asumsi bahwa kalimat yang lebih panjang cenderung mengandung informasi yang lebih kaya dan kontekstual. Teknik *undersampling* dipilih dengan pertimbangan bahwa kelas minoritas, yaitu kelas positif, menunjukkan tingkat keragaman yang rendah. Sebagian besar teks dalam kelas ini memiliki struktur dan ekspresi yang serupa, sehingga penggunaan teknik *oversampling* atau metode sintetik lainnya dikhawatirkan hanya akan mereplikasi pola yang sama tanpa menambah variasi informasi.

Pengurangan data pada kelas mayoritas (netral dan negatif) dinilai lebih aman karena kelas-kelas ini memiliki konten yang lebih bervariasi, sehingga sebagian data dapat dihilangkan tanpa kehilangan representasi keseluruhan. Proses *undersampling* ini tidak dilakukan secara acak, melainkan dengan strategi *undersampling* berbasis pemilihan kalimat terpanjang dari masing-masing kelas yang akan di *undersampling*. Pendekatan ini didasarkan pada asumsi bahwa kalimat yang lebih panjang cenderung mengandung informasi yang lebih kaya dan kontekstual. Hasil akhir dari proses ini adalah distribusi seimbang, dengan masing-masing kelas (positif, netral, dan negatif) terdiri dari 1.523 baris data. Distribusi label sebelum dan setelah teknik *undersampling* data dapat dilihat pada gambar 3.3.