ANALISIS SENTIMEN MASYARAKAT MENGENAI PEMILIHAN UMUM 2024 MENGGUNAKAN *NATURAL LANGUAGE PROCESSING (NLP)*

KERJA PRAKTIK

Program Studi

S1 Teknik Komputer

Oleh:

Muhammad Mizan

21410200021

FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA

ANALISIS SENTIMEN MASYARAKAT MENGENAI PEMILIHAN UMUM 2024 MENGGUNAKAN *NATURAL LANGUAGE PROCESSING (NLP)*

Diajukan sebagai salah satu syarat untuk menyelesaikan
Program Sarjana

Disusun Oleh :

Nama : Muhammad Mizan

NIM : 21410200021

Program : S1 (Strata Satu)

Jurusan : Teknik Komputer

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2024

Lembar Pengesahan

ANALISIS SENTIMEN MASYARAKAT MENGENAI PEMILIHAN UMUM 2024 MENGGUNAKAN *NATURAL LANGUAGE PROCESSING (NLP)*

Laporan Kerja Praktik oleh

Muhammad Mizan

NIM : 21410200021

Telah diperiksa, diuji, dan disetujui

Sidoarjo, 24 Juli 2024

Disetujui :

Pembimbing	Penyelia
Digitally signed by Musayyanah DN: cn=Musayyanah, o=Universitas Dinamika, ou=S1 Teknik Komputer, email=musayyanah@dinamika.ac.id, c=ID Date: 2024.08.12 17:29:38 +07'00'	
Musayyanah, S.ST., M.T.	**Lutfi Dwimulya**
NIDN. 0730069102	

Mengetahui,

Ketua Program Studi S1 Teknik Komputer

cn=Pauladie Susanto, o=Universitas Dinamika, ou=PS S1 Teknik Komputer, email=pauladie@dinamika.ac.id, c=ID
2024.08.13 10:35:58 +07'00'

Pauladie Susanto, S.Kom., M.T

NIDN. 0729047501

PERNYATAAN
PERSETUJUAN DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa **Universitas Dinamika**, Saya :

Nama	: **Muhammad Mizan**
NIM	: **21410200021**
Program Studi	: **S1 Teknik Komputer**
Fakultas	: **Fakultas Teknologi dan Informatika**
Jenis Karya	: **Laporan Kerja Praktik**
Judul Karya	: **ANALISIS SENTIMEN MASYARAKAT MENGENAI PEMILIHAN UMUM 2024 MENGGUNAKAN *NATURAL LANGUAGE PROCESSING* (NLP)**

Menyatakan dengan sesungguhnya bahwa :

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, Saya menyetujui memberikan kepada **Universitas Dinamika** Hak Bebas Royalti Non-Eksklusif ***(Non-Exclusive Royalty Free Right)*** atas seluruh isi/sebagian karya ilmiah Saya tersebut diatas untuk disimpan, dialihmediakan, dan dikelola dalam bentuk pangkalan data (***database***) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama Saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta.
2. Karya tersebut diatas adalah hasil karya asli Saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya, atau pendapat orang lain yang ada dalam karya ilmiah ini semata-mata hanya sebagai rujukan yang dicantumkan dalam Daftar Pustaka Saya.
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiasi pada karya ilmiah ini, maka Saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada Saya.

Demikan surat pernyataan ini Saya buat dengan sebenar-benarnya.

Sidoarjo, 31 Juli 2024

Muhammad Mizan
NIM : 21410200021

ABSTRAK

Pemilihan Umum (Pemilu) 2024 merupakan salah satu peristiwa politik terbesar di Indonesia, yang menarik perhatian luas dari masyarakat. Dengan semakin berkembangnya teknologi, analisis sentimen masyarakat melalui platform media sosial telah menjadi alat penting untuk memahami opini publik. Kerja Praktik ini bertujuan untuk menganalisis sentimen masyarakat terhadap Pemilu 2024 dengan memanfaatkan teknik *Natural Language Processing* (NLP). Data dikumpulkan dari platform media sosial Twitter, dengan total 1336 tweet yang diambil dari *dataset* Kaggle. Data tersebut terdiri dari 988 tweet dengan sentimen positif, 187 tweet dengan sentimen netral, dan 161 tweet dengan sentimen negatif. Data kemudian diolah menggunakan teknik prapemrosesan teks seperti *tokenisasi*, penghapusan *stopwords*, dan *stemming*. Selanjutnya, metode NLP seperti menganalisis sentimen dan mengklasifikasikan teks untuk identifikasi pola sentimen dalam *dataset*. Model yang dikembangkan memiliki hasil akhir akurasi sebesar 88.06%, dengan nilai *precision, recall, dan f1-score* yang beragam untuk setiap kategori sentimen.

.

Kata kunci : Pemilu 2024, analisis sentimen, *Natural Language Processing*, Twitter, opini publik

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Allah SWT yang telah melimpahkan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan laporan akhir proyek penelitian yang berjudul "Sentimen Analysis Masyarakat Mengenai Pemilihan Umum 2024 Menggunakan *Natural Language Processing* (NLP)*"*.

Penelitian ini bertujuan untuk menganalisis sentimen masyarakat terhadap Pemilihan Umum 2024 di Indonesia melalui data yang diperoleh dari platform media sosial Twitter. Dengan menggunakan teknik *Natural Language Processing* (NLP), penelitian ini diharapkan dapat memberikan wawasan yang berharga mengenai opini publik terhadap Pemilu 2024 dan tren sentimen yang muncul selama periode penelitian.

Ucapan terima kasih yang sebesar-besarnya penulis sampaikan kepada:

1. Allah SWT, atas segala rahmat dan hidayah-Nya.
2. Kedua orang tua, yang selalu memberikan dukungan dan doa tanpa henti.
3. Bapak Pauladie Susanto, S.Kom., M.T., selaku ketua program studi S1 Teknik Komputer, yang telah memberikan dukungan dan izin kepada penulis untuk mengikuti studi independen.
4. Ibu Musayyanah, S.ST., M.T., selaku dosen pembimbing, yang telah mendampingi selama proses studi independent berlangsung.
5. PT Hacktivate Teknologi Indonesia, sebagai mitra tempat studi independen yang telah memberikan kesempatan kepada penulis untuk mengembangkan kemampuan dalam bidang AI dan *Cyber Security*

Sekian ucapan terima kasih dari penulis. Mohon maaf apabila terdapat kesalahan kata dan penulisan dalam laporan ini

DAFTAR ISI

Daftar Gambar

BAB I
PENDAHULUAN

1.1 Latar Belakang

Pemilihan umum merupakan pilar utama dalam sistem demokrasi, memungkinkan masyarakat untuk memilih pemimpin dan menentukan arah kebijakan negara. Menjelang Pemilihan Umum 2024, memahami sentimen masyarakat menjadi sangat penting. Sentimen ini mencerminkan pandangan, perasaan, dan opini masyarakat terhadap kandidat dan isu-isu yang diangkat. Analisis sentimen dapat memberikan wawasan berharga mengenai dukungan publik dan tren pemilih yang dapat mempengaruhi hasil pemilihan.

Namun, menganalisis sentimen masyarakat menghadapi berbagai tantangan. Volume data yang besar, keragaman opini, dan kompleksitas bahasa membuat proses ini menjadi tugas yang kompleks. Untuk mengatasi tantangan ini, *Natural Language Processing* (NLP) menawarkan solusi yang efektif. NLP, sebagai cabang dari kecerdasan buatan, memungkinkan analisis teks secara otomatis dengan akurasi tinggi, memproses data dalam skala besar dan memahami bahasa alami yang bervariasi (Ali, 2021).

Kerja Praktik ini bertujuan untuk menganalisis sentimen masyarakat mengenai Pemilihan Umum 2024 menggunakan teknologi NLP. Dengan menggunakan teknik NLP, Kerja Praktik ini akan mengidentifikasi sentimen dominan terhadap kandidat dan isu-isu utama, serta memberikan rekomendasi berdasarkan hasil analisis. Metodologi yang diterapkan mencakup klasifikasi sentimen dan analisis opini, yang akan memberikan wawasan mendalam tentang preferensi dan kekhawatiran masyarakat. Kontribusi dari kerja praktik ini diharapkan dapat memperluas pemahaman tentang bagaimana sentimen masyarakat dapat dianalisis dan diinterpretasikan dalam konteks politik. Selain itu, hasil dari Kerja Praktik ini dapat memberikan manfaat praktis bagi kandidat, partai politik, dan pemangku kepentingan lainnya dalam merancang strategi komunikasi dan kampanye yang lebih efektif.

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian ini dirancang untuk mengidentifikasi dan menjawab beberapa pertanyaan utama terkait dengan analisis sentimen masyarakat mengenai Pemilihan Umum 2024. Adapun rumusan masalah yang diangkat dalam penelitian ini adalah sebagai berikut:

1. Bagaimana pola sentimen masyarakat terhadap Pemilihan Umum 2024 yang diungkapkan melalui platform Twitter?
2. Bagaimana menganalisis akurasi metode NLP dalam mengklasifikasikan sentimen dari tweet terkait Pemilihan Umum 2024?

1.3 Batasan Masalah

Penelitian ini juga memiliki beberapa batasan yang perlu diperhatikan untuk menjaga fokus dan keterbatasan ruang lingkup penelitian. Adapun batasan-batasan tersebut adalah:

1. Data yang Digunakan

   Penelitian ini menggunakan data tweet yang diambil dari platform Kaggle, yang terbatas pada 1336 tweet terkait Pemilihan Umum 2024. Data ini mungkin tidak mencerminkan seluruh populasi pengguna Twitter atau masyarakat umum.
2. Bahasa

   Analisis hanya dilakukan pada tweet dalam bahasa Indonesia. Tweet dalam bahasa lain atau yang menggunakan campuran bahasa mungkin tidak dianalisis secara akurat.
3. Metode Analisis

   Model NLP yang digunakan dalam penelitian ini terbatas seperti menggunakan *pad sequence, dropout, Long Short Term Memories* , dan sebagainya
4. Waktu Pengumpulan Data

   Data dikumpulkan pada periode waktu tertentu dan mungkin tidak mencerminkan perubahan sentimen secara *real-time*. Peristiwa terbaru atau perubahan dalam opini publik setelah periode pengumpulan data tidak termasuk dalam analisis.
5. Keterbatasan Komputasi

   Proses analisis dan klasifikasi sentimen dilakukan menggunakan sumber daya

komputasi yang tersedia (seperti contoh *sagemaker*, *google collab*) yang mungkin membatasi kemampuan untuk menangani volume data yang lebih besar atau model yang lebih kompleks.

1.4 Tujuan Penelitian

Adapun tujuan penelitian dari Kerja Praktik ini adalah sebagai berikut :

1. Mengidentifikasi sentimen masyarakat terkait Pemilihan Umum 2024 dari opini dan komentar yang beredar di media sosial dan sumber lainnya.
2. Menguji akurasi dan efektivitas berbagai teknik NLP dalam analisis sentimen untuk menentukan metode yang paling tepat.
3. Memberikan hasil dari seberapa akurat model yang dilatih

1.5 Manfaat Penelitian

Adapun manfaat penelitian dari Kerja Praktik ini adalah sebagai berikut :

1. Memperdalam pemahaman dan keterampilan dalam penerapan teknik *Natural Language Processing* (NLP) dan analisis sentimen, yang sangat berguna untuk pengembangan karir di bidang teknologi dan data kedepannya.
2. Memberikan pengalaman praktis dalam mengelola dan menganalisis data besar, serta dalam mengembangkan dan mengevaluasi model NLP, yang dapat memperkuat resume dan portofolio.
3. Mengasah kemampuan analitis dan kritis dalam menilai hasil dan kinerja model, serta dalam menyusun laporan yang informatif dan berbasis data.
4. Menyediakan dasar yang kuat untuk proyek penelitian atau pengembangan selanjutnya, baik dalam analisis sentimen atau aplikasi NLP lainnya.

BAB II
GAMBARAN UMUM PERUSAHAAN

2.1 Sejarah dan Latar Belakang Perusahaan

Gambar 2.1 Logo Hacktiv8

Hacktiv8, didirikan pada tahun 2016, adalah lembaga pendidikan yang berfokus pada pelatihan intensif di bidang teknologi informasi. Hacktiv8 menawarkan program *coding bootcamp* pertama di Asia yang terdaftar dalam *Council on Integrity in Results Reporting* (CIRR), yang menekankan transparansi dalam melaporkan hasil lulusan mereka. Dengan misi mencetak talenta digital siap kerja, Hacktiv8 telah berkembang pesat sejak awal pendiriannya. Hacktiv8 bermula dari kebutuhan mendesak akan tenaga kerja terampil di sektor teknologi di Indonesia. Melihat kesenjangan ini, pendiri Hacktiv8 menciptakan sebuah platform edukatif yang dapat mengubah pemula menjadi profesional teknologi yang siap bersaing di pasar kerja. Program-program yang ditawarkan oleh Hacktiv8 mencakup pembelajaran intensif dalam bahasa pemrograman seperti *JavaScript, Python*, serta keterampilan pengembangan web dan *data science*.

Selain program bootcamp reguler, Hacktiv8 juga bekerja sama dengan berbagai institusi dan perusahaan besar untuk menyelenggarakan pelatihan khusus yang disesuaikan dengan kebutuhan industri. Misalnya, Hacktiv8 telah bermitra dengan Kominfo untuk menyelenggarakan *Fresh Graduate Academy*, menyediakan beasiswa bagi lulusan baru untuk mengikuti pelatihan Golang dan Product Owner. Mereka juga menyelenggarakan program pelatihan untuk perusahaan-perusahaan seperti BCA, PT Angkasa Pura II, dan Grab, yang bertujuan meningkatkan keterampilan teknis karyawan mereka. Hacktiv8 tidak hanya fokus pada pengembangan kurikulum yang relevan tetapi juga pada pengembangan jaringan kerja yang kuat dengan berbagai perusahaan teknologi. Hal ini memastikan lulusan Hacktiv8 memiliki akses ke peluang kerja yang luas dan relevan.

2.2 Visi dan Misi Perusahaan

Visi

Hacktiv8 bertujuan untuk menjadi lembaga pendidikan terdepan yang mampu mengatasi kesenjangan keterampilan teknologi, dan mempersiapkan talenta digital yang siap kerja serta berkontribusi pada pertumbuhan ekonomi Indonesia.

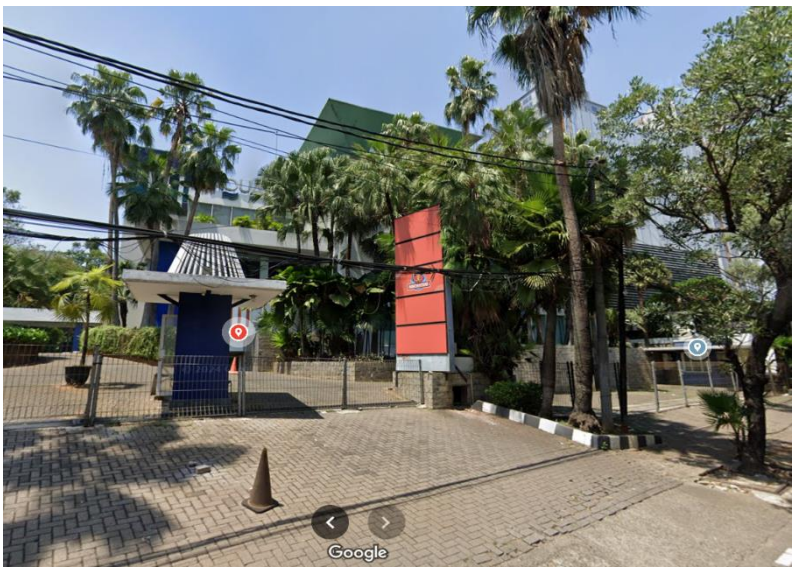
Misi

1. Menyediakan program pelatihan yang intensif dan komprehensif untuk membekali peserta dengan keterampilan teknologi yang diperlukan oleh industri.
2. Memungkinkan peserta untuk mempelajari keterampilan teknologi dengan cepat agar mereka dapat mengikuti laju perubahan dan berkembang di era digital.
3. Melalui pendidikan dan pemberdayaan, Hacktiv8 berupaya untuk mengubah hidup peserta dan membentuk masa depan yang lebih cerah bagi semua yang terlibat.
4. Membangun kepercayaan, menjunjung tinggi integritas, dan berusaha untuk mencapai keunggulan dalam setiap keputusan dan tindakan yang diambil.

2.3 Profil Perusahaaan

Nama : PT Hacktivate Teknologi Indonesia

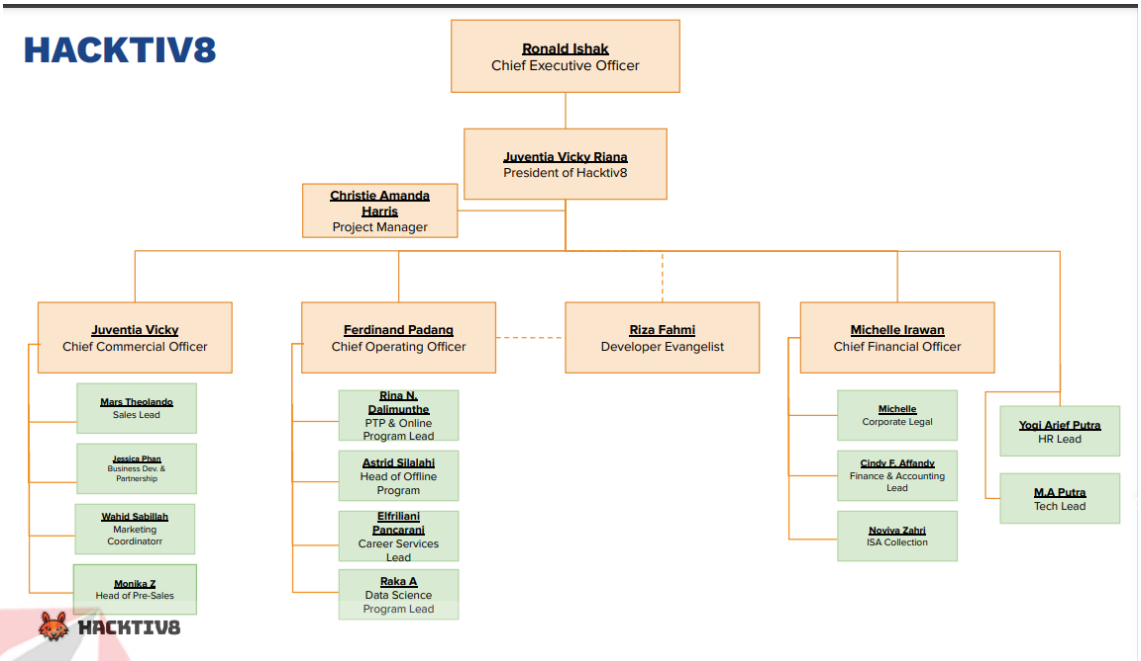
Alamat : Gedung Aquarius Lt. 1&2, Jl. Sultan Iskandar Muda No. 7, RT. 005 RW. 009, Kebayoran Lama Selatan, Kebayoran Lama, Jakarta Selatan 12240



Gambar 2.2 Hacktiv8 di Kebayoran lama, Jakarta

2.4 Struktur Perusahaan Hacktiv8

Dibawah adalah gambar struktur Perusahaan dari PT.Hacktivate Teknologi Indonesia



Gambar 2.5 Struktur Organisasi Hacktiv

PT Hacktivate Teknologi Indonesia menawarkan berbagai produk dan layanan yang dirancang untuk memenuhi kebutuhan reskilling dan upskilling di industri teknologi. Berikut adalah beberapa produk dan layanan utama yang ditawarkan:

1. Hacktiv8 menyediakan program bootcamp intensif yang dirancang untuk mengubah pemula menjadi pengembang perangkat lunak siap kerja dalam waktu singkat. Program ini mencakup berbagai bahasa pemrograman dan teknologi terbaru, termasuk *JavaScript, React, Node.js*, dan lainnya. Kurikulum yang praktis dan berbasis proyek memastikan siswa mendapatkan pengalaman nyata yang diperlukan untuk sukses dalam karir
2. Program Spesialis Selain bootcamp coding, Hacktiv8 juga menawarkan program spesialisasi untuk keterampilan teknologi tertentu. Program ini mencakup bidang seperti *data science, machine learning*, dan pengembangan aplikasi *mobile*. Setiap program dirancang untuk memberikan pengetahuan mendalam dan keterampilan praktis yang dapat langsung diterapkan di industri.
3. Kelas *Online* Untuk menjangkau lebih banyak individu yang ingin meningkatkan keterampilan mereka, Hacktiv8 menyediakan kelas online yang fleksibel. Kelas ini memungkinkan peserta belajar dari mana saja dan kapan saja, dengan materi yang mencakup berbagai topik teknologi dan pengembangan keterampilan digital.

4. Hacktiv8 bekerja sama dengan berbagai perusahaan untuk menyediakan pelatihan karyawan yang disesuaikan dengan kebutuhan spesifik perusahaan tersebut. Pelatihan ini membantu perusahaan dalam meningkatkan keterampilan digital karyawan mereka dan mempersiapkan mereka untuk tantangan teknologi yang terus berkembang.
5. Selain pelatihan teknis, Hacktiv8 menawarkan dukungan karir yang komprehensif, termasuk mentorship, *workshop* pengembangan karir, dan bantuan penempatan kerja.

2.6 Kinerja dan Prestasi PT. Hacktivate Teknologi Indonesia

Sejak didirikan pada tahun 2016, PT Hacktivate Teknologi Indonesia telah menunjukkan kinerja yang mengesankan dalam dunia pendidikan teknologi. Beberapa pencapaian dan prestasi perusahaan meliputi:

1. Hingga tahun 2022, Hacktiv8 telah melatih lebih dari 7.200 siswa dalam berbagai program teknologi, seperti *Full Stack JavaScript, Data Science, dan Performance Marketing.*
2. Lebih dari 88% lulusan Hacktiv8 berhasil mendapatkan pekerjaan dalam waktu 90 hari setelah menyelesaikan program bootcamp.
3. Lulusan Hacktiv8 mengalami peningkatan penghasilan sebesar 84% dibandingkan sebelum mengikuti program pelatihan.
4. Hacktiv8 telah bermitra dengan lebih dari 800 perusahaan yang mencari talenta digital, termasuk perusahaan-perusahaan besar di sektor perbankan, teknologi, dan BUMN.
5. Program *Income Share Agreement* (ISA) yang memungkinkan siswa untuk membayar biaya pendidikan setelah mendapatkan pekerjaan. Hingga kini, lebih dari 840 kontrak ISA telah disetujui dengan total nilai lebih dari IDR 41,6 miliar

BAB III
LANDASAN TEORI

3.1 Pengertian Sentimen Analyst

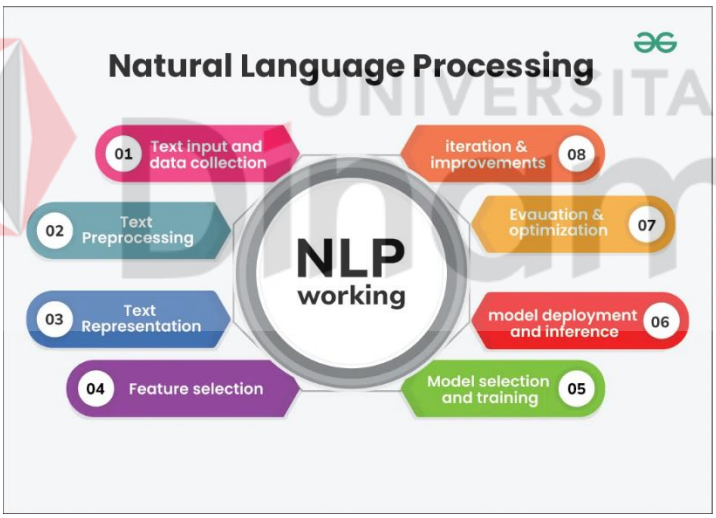
Sentimen Analysis, juga dikenal sebagai opini *mining* atau sentiment *mining*, adalah proses menggunakan teknik pemrosesan bahasa alami (NLP), analisis teks, dan linguistik komputasional untuk mengidentifikasi dan mengekstraksi informasi subyektif dari teks. Analisis ini berfokus pada penentuan pendapat, perasaan, atau sikap pengguna terhadap suatu topik, produk, atau peristiwa tertentu.

Biasanya, Sentimen Analysis digunakan untuk memahami opini publik terhadap calon, partai politik, atau isu-isu yang sedang diperbincangkan. Hal ini dilakukan dengan menganalisis data dari berbagai sumber seperti media sosial, forum, dan berita *online*. Hasil dari analisa ini dapat memberikan wawasan yang berharga bagi partai politik dan calon untuk mengetahui persepsi publik dan mengembangkan strategi kampanye yang lebih efektif (Rita et al., 2023). Sentimen Analysis bekerja dengan mengklasifikasikan teks ke dalam kategori sentimen positif, negatif, atau netral. Teknik-teknik yang sering digunakan dalam Sentimen Analysis meliputi *tokenization*, *stemming*, *lemmatization*, dan penggunaan algoritma *machine learning* seperti *Naive Bayes*, *Support Vector Machines*, dan model *deep learning* seperti *Recurrent Neural Networks* (RNN) dan *Long Short-Term Memory* (LSTM). Dengan kemajuan teknologi dan data yang semakin banyak tersedia, Sentimen Analysis menjadi alat yang sangat penting dalam berbagai bidang, termasuk pemasaran, layanan pelanggan, dan penelitian sosial (Putra et al., 2023).

3.2 Pengertian *Natural Language Processing* (NLP)

Natural Language Processing (NLP) adalah cabang dari kecerdasan buatan (AI) yang berfokus pada interaksi antara komputer dan manusia menggunakan bahasa

alami. Tujuan utama NLP adalah untuk memungkinkan komputer memahami, menafsirkan, dan merespons bahasa manusia dengan cara yang bermanfaat. NLP mencakup dua bagian utama: *Natural Language Understanding* (NLU), yang memungkinkan mesin memahami bahasa alami, dan *Natural Language Generation* (NLG), yang berfokus pada produksi teks yang bermakna dari representasi internal. Teknik-teknik dalam NLP seperti *tokenization, stemming,* dan *parsing* digunakan untuk memecah teks menjadi unit-unit yang lebih kecil dan menganalisis strukturnya. Algoritma *machine learning* dan model statistik kemudian diterapkan untuk mengekstrak makna dari teks dan membuat prediksi atau keputusan berdasarkan data tersebut. NLP memiliki banyak aplikasi praktis, termasuk *chatbots*, sistem rekomendasi, analisis sentimen, dan asisten virtual seperti Siri dan Google *Assistant*. Kemajuan dalam NLP telah memungkinkan pemrosesan bahasa alami yang lebih akurat dan efisien, dengan dampak signifikan dalam berbagai bidang seperti bisnis, kesehatan, dan pendidikan (Khurana et al., 2023).



Gambar 3.1 Cara kerja NLP

3.3 Sejarah *Natural Language Processing* (NLP)

Sejarah *Natural Language Processing* (NLP) dimulai pada tahun 1950-an, dengan eksperimen awal dalam penerjemahan mesin dan pengenalan ucapan. Salah satu tonggak penting adalah munculnya istilah "mesin penerjemah" pada pertengahan 1930-an, dengan dua paten awal, termasuk oleh Georges Artsrouni yang menggunakan kamus dwibahasa dan pita kertas untuk memetakan kata-kata antar bahasa. Pada tahun 1957, Noam Chomsky memperkenalkan teori tata bahasa transformasi yang menjadi dasar

bagi pengembangan NLP lebih lanjut. Pada 1980-an dan 1990-an, metode statistik dan probabilistik menggantikan pendekatan berbasis aturan, seiring dengan berkembangnya teknologi komputasi. NLP kemudian mengalami kemajuan besar dengan penerapan algoritma *machine learning* dan *deep learning*, yang memfasilitasi pemrosesan bahasa alami dan analisis teks secara lebih efisien. Saat ini, NLP mencakup berbagai area seperti pengenalan suara, penerjemahan mesin, ringkasan teks otomatis, serta analisis sintaksis dan semantik untuk memahami struktur dan makna kalimat. Namun, tantangan seperti ambiguitas bahasa dan penggunaan slang masih perlu diatasi dalam penelitian dan pengembangan lebih lanjut di bidang ini (Johri et al., 2021).

3.4 Teknik Teknik dalam NLP untuk Analisis

Dalam analisis sentimen menggunakan *Natural Language Processing* (NLP), terdapat beberapa teknik utama yang digunakan untuk memproses dan menganalisis teks. Berikut adalah penjelasan mengenai beberapa teknik tersebut:

1. *Tokenization*

Tokenization adalah proses memecah teks menjadi unit-unit yang lebih kecil, biasanya kata atau frasa. Ini adalah langkah pertama dalam banyak tugas NLP, karena memungkinkan komputer untuk memahami dan memproses teks secara lebih efisien. Dengan memecah teks menjadi token, kita dapat menganalisis dan mengelola data teks dengan lebih mudah(Webster & Kit, 1992).

2. *Stemming*

Stemming adalah teknik untuk mengurangi kata-kata ke bentuk dasarnya (*root form*) dengan memotong akhiran-akhiran kata. Misalnya, kata "*running*", "*runner*", dan "*ran*" akan diubah menjadi "*run*". Teknik ini membantu mengurangi dimensi data dan memperbaiki performa model analisis sentimen dengan menyatukan varian kata yang memiliki makna serupa (Rianto et al., 2021).

3. *Stop words removal*

Stop words adalah kata-kata umum seperti "*and*", "*the*", "*is*", dan lain-lain yang sering kali tidak membawa banyak makna penting dalam analisis teks. Menghapus *stop words* dari teks membantu mengurangi *noise* dan meningkatkan fokus pada kata-kata yang lebih bermakna (Sarica & Luo, 2021).

4. *Bag of Words* (BoW)

Bag of Words adalah teknik representasi teks yang mengabaikan urutan kata dan hanya

menghitung frekuensi kemunculan kata dalam sebuah dokumen. Setiap dokumen diwakili sebagai vektor dari frekuensi kata.

5. TF-IDF (*Term Frequency-Inverse Document Frequency*)

TF-IDF adalah metode penilaian pentingnya kata dalam dokumen relatif terhadap koleksi dokumen. *Term Frequency* (TF) mengukur seberapa sering kata muncul dalam dokumen, sedangkan *Inverse Document Frequency* (IDF) mengukur seberapa jarang kata muncul dalam korpus. Nilai TF-IDF tinggi menunjukkan bahwa kata tersebut penting dalam dokumen tertentu tetapi tidak umum dalam koleksi dokumen, membuatnya lebih berguna untuk tugas-tugas seperti analisis sentimen (Gifari et al., 2022).

3.5 *Long Short Term Memories* (LSTM)

LSTM atau *Long Short-Term Memory* adalah jenis jaringan saraf yang dirancang khusus untuk mengatasi tantangan dalam memproses data berurutan, seperti teks atau sinyal waktu. LSTM memiliki struktur unik yang memungkinkan penyimpanan informasi untuk waktu yang lama, sehingga dapat mengingat data penting meskipun ada jeda waktu yang panjang. Dengan menggunakan mekanisme gerbang, LSTM dapat memilih informasi mana yang harus diingat atau dilupakan, menjadikannya sangat efektif dalam menangani masalah yang melibatkan urutan data yang kompleks dan berjangka waktu panjang (Hochreiter & Schmidhuber, 1997). Berbeda dengan RNN, yang cenderung mengalami kesulitan dalam mempertahankan informasi dalam waktu yang lama, LSTM dirancang untuk mengingat informasi yang relevan selama periode waktu yang lebih lama dan menghapus data yang sudah tidak diperlukan. Dengan kemampuan ini, LSTM menjadi lebih efektif dalam memproses, memprediksi, dan mengklasifikasikan data yang berkaitan dengan urutan waktu tertentu.

Perbedaan utama antara LSTM dan RNN terletak pada cara menangani informasi jangka panjang. LSTM mengatasi masalah yang sering dihadapi oleh RNN, yaitu keterbatasan dalam menyimpan informasi untuk waktu yang lama. Dengan demikian, LSTM tidak hanya mengatasi masalah jangka waktu penyimpanan, tetapi juga memungkinkan pengelolaan informasi yang lebih efisien dan relevan. Sistem yang memanfaatkan LSTM dapat dengan baik memproses dan memprediksi data dalam konteks deret waktu. Dengan prinsip kerja yang memungkinkan penghapusan data yang sudah usang, LSTM membantu memastikan bahwa informasi yang dikelola tetap akurat

dan terkini

Struktur LSTM:

LSTM memiliki struktur yang kompleks namun efektif, terdiri dari beberapa komponen utama yang bekerja bersama untuk mengelola informasi. Struktur ini mencakup tiga gerbang utama:

1. *Forget Gate*: Bertugas untuk menentukan informasi mana yang harus dihapus dari sel memori. Gerbang ini menganalisis input saat ini dan status memori sebelumnya untuk membuat keputusan tentang informasi mana yang harus dihapus.
2. *Input Gate*: Menentukan informasi baru mana yang akan disimpan dalam sel memori. Gerbang ini menggabungkan input saat ini dengan status memori sebelumnya untuk memperbarui informasi yang relevan.
3. *Output Gate*: Mengontrol informasi mana yang akan digunakan sebagai output dari sel memori. Gerbang ini memproses informasi dari sel memori dan input saat ini untuk menghasilkan output yang relevan.

Dengan kombinasi dari ketiga gerbang ini, LSTM mampu menangani informasi dengan cara yang lebih canggih dibandingkan RNN, menjadikannya alat yang sangat berguna dalam berbagai aplikasi yang memerlukan pemrosesan data.

3.6 *Recurrent Neural Network* (RNN)

Recurrent Neural Network (RNN) adalah jenis jaringan saraf yang dirancang untuk memproses data berurutan dengan cara mempertahankan informasi dari langkah sebelumnya. RNN menggunakan struktur umpan balik, di mana output dari langkah sebelumnya dijadikan input untuk langkah berikutnya. Hal ini memungkinkan model menangkap ketergantungan temporal dalam data.

RNN bisa dinyatakan dalam bentuk persamaan diferensial, di mana evolusi sinyal status dinyatakan sebagai fungsi dari waktu dan *input*. Model ini dapat dioptimalkan menggunakan teknik pelatihan seperti Back Propagation Through Time (BPTT). Namun, RNN standar sering menghadapi tantangan seperti masalah gradien yang menghilang dan meledak. Hal ini mendorong pengembangan arsitektur yang lebih canggih seperti *Long Short-Term Memory* (LSTM) untuk mengatasi masalah tersebut(Sherstinsky, 2020).

BAB IV
DESKRIPSI PEKERJAAN

4.1 Kerja Praktik

Kerja Praktik, sering kali disebut sebagai magang atau praktik kerja, adalah suatu kegiatan yang dirancang untuk memberikan pengalaman langsung kepada mahasiswa dalam lingkungan kerja profesional. Tujuan utama dari kerja praktik adalah untuk menghubungkan teori yang dipelajari di bangku kuliah dengan praktik nyata di dunia industri. Melalui kerja praktik, mahasiswa dapat memperoleh pemahaman yang lebih mendalam tentang bidang yang mereka geluti serta mengasah keterampilan yang diperlukan dalam dunia kerja. Kerja Praktik biasanya melibatkan penugasan pada proyek atau tugas tertentu yang relevan dengan bidang studi mahasiswa, di mana mereka berkesempatan untuk menerapkan pengetahuan akademis mereka dalam situasi nyata. Kegiatan ini juga bertujuan untuk membangun keterampilan praktis, memperluas jaringan profesional, dan meningkatkan kesiapan kerja mahasiswa setelah lulus.

Selama periode kerja praktik, mahasiswa biasanya bekerja di bawah bimbingan seorang mentor atau supervisor yang berpengalaman di perusahaan atau organisasi tempat mereka menjalani praktik. Selain itu, kerja praktik sering kali melibatkan evaluasi berkala untuk mengukur kemajuan dan pencapaian mahasiswa, serta memberikan umpan balik yang konstruktif untuk pengembangan lebih lanjut.

4.2 Metode Pembelajaran selama Kerja Praktik

Selama periode kerja praktik, berbagai metode pembelajaran diterapkan untuk memastikan pemahaman yang mendalam dan penerapan yang efektif dari pengetahuan yang diperoleh. Metode-metode ini meliputi:

1. Pembelajaran Mandiri (Self-Learning)

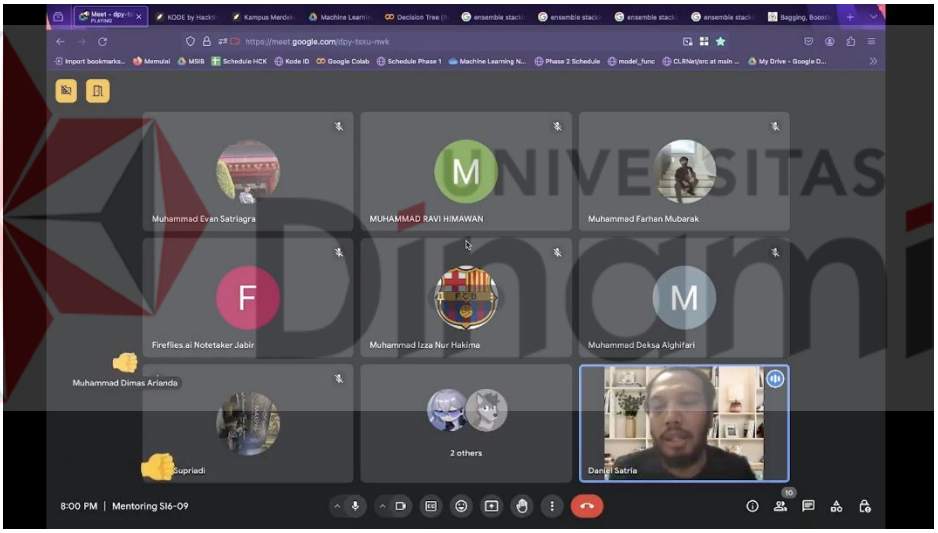
Pembelajaran mandiri merupakan salah satu metode utama yang digunakan di luar kegiatan mentoring dan sesi google meet. Metode ini melibatkan studi individu oleh mahasiswa untuk mengeksplorasi dan memahami materi secara mendalam. Mahasiswa diharapkan untuk membaca literatur terkait, melakukan penelitian, dan melatih keterampilan secara mandiri diluar google meet.

2. Pembelajaran Online (Google Meet)

Pembelajaran online dilakukan melalui platform Google Meet, yang memungkinkan interaksi langsung dan diskusi antara mahasiswa dan pengajar. Sesi ini dirancang untuk memberikan penjelasan, menjawab pertanyaan, memfasilitasi pembelajaran secara real-time dan menyampaikan materi. Pembelajaran online memastikan bahwa mahasiswa dapat mengakses materi dan mendapatkan bimbingan meskipun secara virtual.

3. Mentoring Kelas Kecil

Selama kerja praktik, mahasiswa dibagi menjadi kelompok kecil untuk mendapatkan bimbingan lebih mendalam dari mentor yang berbeda. Setiap kelompok kecil dikelola oleh seorang mentor, yang bertugas memberikan dukungan, umpan balik, dan arahan khusus sesuai dengan kebutuhan dan perkembangan masing-masing mahasiswa. Mentoring ini berfungsi untuk pendekatan yang lebih personal dan intensif dalam proses pembelajaran.



Gambar 4.1 Mentoring Kecil dengan google meet

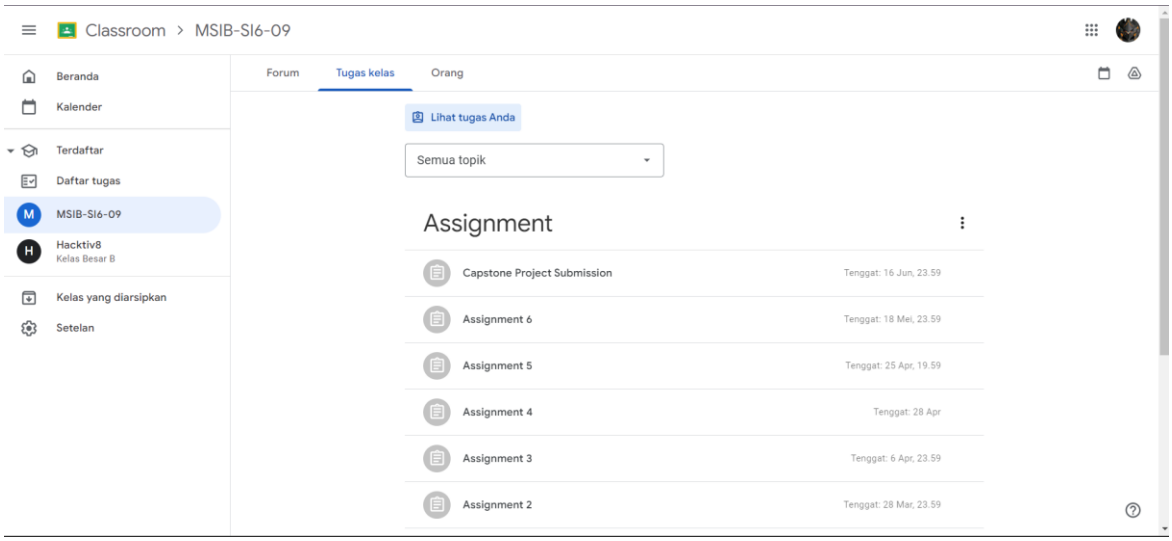
4. Pembelajaran Kelas Besar

Selain sesi mentoring kecil, terdapat juga sesi kelas besar yang melibatkan seluruh mahasiswa. Kelas besar ini dilakukan melalui Google Meet dan terdiri dari dua kelas besar yang memungkinkan semua peserta untuk belajar bersama secara simultan. Sesi ini dirancang untuk memberikan pemahaman umum, berbagi pengetahuan, dan melakukan diskusi kelompok yang melibatkan seluruh peserta.

5. Pengumpulan Tugas (Google Classroom)

Untuk pengumpulan tugas dan dokumentasi kemajuan, Google Classroom digunakan sebagai platform utama. Mahasiswa dapat mengunggah tugas, laporan, dan materi lainnya melalui Google Classroom, yang memudahkan pengelolaan dan penilaian tugas

oleh mentor dan pengajar.



Gambar 4.2 Pengumpulan Tugas

6. Akses Materi (kode.id)

Website kode.id digunakan sebagai sumber utama untuk mengakses materi selama kegiatan mengajar berlangsung

4.3 Deskripsi *Project* (Capstone *Project*)

a. Tujuan *Project*

Proyek capstone ini bertujuan untuk melakukan klasifikasi teks dalam bentuk sentimen pada tweet yang terkait dengan pemilihan umum 2024. Dengan menggunakan teknik *Natural Language Processing* (NLP), proyek ini memfokuskan pada identifikasi dan pengelompokkan opini masyarakat ke dalam tiga kategori sentimen utama: positif, negatif, dan netral. Klasifikasi ini bertujuan untuk memberikan wawasan mengenai bagaimana masyarakat merespons isu-isu yang berkaitan dengan pemilihan umum, serta untuk memfasilitasi analisis yang lebih mendalam terhadap opini publik di platform media sosial

b. Ruang Lingkup Proyek

Ruang lingkup dari proyek ini terbatas pada analisis data yang diperoleh dari Twitter. Data yang digunakan dalam proyek ini berupa tweet yang relevan dengan pemilihan umum 2024. Analisis ini tidak mencakup data dari platform media sosial lain atau sumber data di luar Twitter. Fokus utama adalah pada pengumpulan, pembersihan, dan klasifikasi tweet untuk menentukan sentimen masyarakat terhadap peristiwa dan isu terkait pemilihan umum

Gambar 4.3 Logo Twitter (Sekarang X)

4.4 Data yang digunakan

Data yang digunakan dalam proyek ini diperoleh dari situs Kaggle, yang merupakan platform untuk berbagi *dataset* publik. *Dataset* ini disediakan dalam format file CSV dan berisi kumpulan tweet yang telah dikategorikan berdasarkan sentimen mereka. Penggunaan *dataset* dari Kaggle mempermudah akses ke data yang telah terstruktur dan memungkinkan peneliti untuk fokus pada analisis sentimen tanpa perlu mengumpulkan data secara manual dari Twitter.

Link Datasheet : https://www.kaggle.com/code/waodeasriyani/analisis-sentimen-opini-publik-terhadap-pilpres/notebook

Gambar 4.4 Platform Kaggle sebagai sumber data

Dataset yang digunakan dalam analisis ini berjumlah 1336 tweet yang telah dilabeli dengan sentimen yang relevan. Rincian distribusi sentimen dalam *dataset* adalah sebagai berikut:

- **Sentimen Positif:** 988 tweet

- **Sentimen Netral:** 187 tweet
- **Sentimen Negatif:** 161 tweet

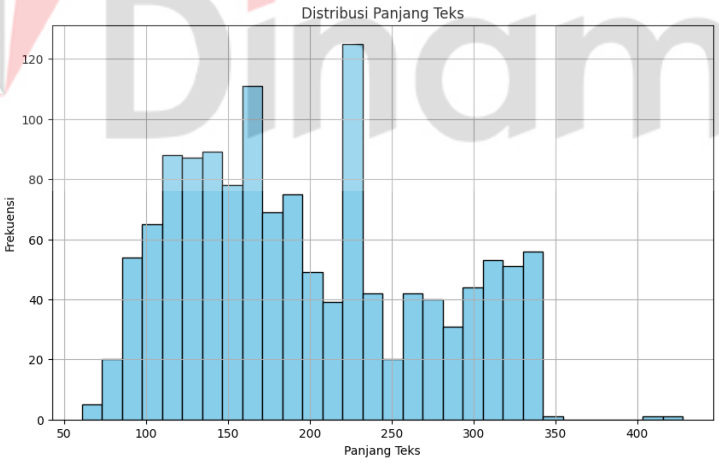
Data ini hanya diambil pada bagian sentimen, sehingga tidak menyertakan metadata tambahan, seperti tanggal, waktu, atau informasi pengguna. Fokus utama adalah pada teks tweet itu sendiri dan klasifikasi sentimen yang diberikan.

4.4 Persiapan Data

Sebelum memasuki tahap analisis dan pelatihan model, dilakukan beberapa langkah persiapan data untuk memastikan bahwa data yang digunakan telah bersih, terstruktur, dan siap diolah lebih lanjut. Langkah-langkah ini meliputi:

1. Analisis Panjang Teks

- Tujuan: Menilai panjang teks tweet untuk memahami bagaimana panjang teks berhubungan dengan klasifikasi sentimen.
- Proses**:** Menghitung jumlah kata atau karakter dalam setiap tweet untuk mengidentifikasi pola atau distribusi panjang yang mungkin mempengaruhi analisis sentimen.



Gambar 4.5 Panjang Text tiap tweet

```
import matplotlib.pyplot as plt
# M
plt.figure(figsize=(10, 6))
plt.hist(data['Length_Text'], bins=30, color='skyblue', edgecolor='black')
plt.title('Distribusi Panjang Teks')
plt.xlabel('Panjang Teks')
plt.ylabel('Frekuensi')
plt.grid(True)
plt.show()
```

Gambar 4.6 Kodingan Analisis Text

Penjelasan bahasa pemrograman pada gambar :

import matplotlib.pyplot as plt

- Mengimpor pustaka matplotlib.pyplot dengan alias plt, yang digunakan untuk membuat visualisasi grafis seperti plot dan histogram.

plt.figure(figsize=(10, 6))

- Membuat sebuah figure (gambar) baru dengan ukuran 10 inci x 6 inci.

plt.hist(data['Length_Text'], bins=30, color='skyblue', edgecolor='black')

- Membuat histogram.
- Menggunakan data dari kolom Length_Text dalam DataFrame data untuk membuat histogram.
- Membagi data menjadi 30 interval (bin) pada histogram.
- Menetapkan warna batang histogram menjadi biru langit.
- Menetapkan warna tepi batang histogram menjadi hitam.

plt.title('Distribusi Panjang Teks')

- Menambahkan judul "Distribusi Panjang Teks" pada histogram.

plt.xlabel('Panjang Teks')

- Menambahkan label "Panjang Teks" pada sumbu x.

plt.ylabel('Frekuensi')

- Menambahkan label "Frekuensi" pada sumbu y.

plt.grid(True)

- Menampilkan grid pada plot untuk memudahkan pembacaan data.

plt.show()

- Menampilkan histogram yang telah dibuat.

2. *Word Cloud* (Visualisasi)

- Tujuan: Mengidentifikasi kata-kata yang sering muncul dan memvisualisasikannya untuk mendapatkan gambaran umum tentang frekuensi kata dalam *dataset*.
- Proses: Membuat word cloud yang menampilkan kata-kata paling umum dalam *dataset*, membantu mengenali kata-kata dominan yang mungkin berhubungan dengan masing-masing kelas sentiment

Gambar 4.7 *Wordscloud*

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

# Menggabungkan semua teks -> satu string
text_combined = ' '.join(data['Text'])

# Memfilter link gajelas (seperti https dan t.co)
text_filtered = re.sub(r'http\S+', '', text_combined)
text_filtered = re.sub(r't\.co\S+', '', text_filtered)

# Membuat WordCloud
wordcloud = WordCloud(width=800, height=400, background_color='white').generate(text_filtered)

# Menampilkan
plt.figure(figsize=(10, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.title('Kata-kata Umum')
plt.axis('off')
plt.show()
```

Gambar 4.8 Kodingan *Wordscloud*

Penjelasan mengenai kodingan di atas :

from wordcloud import WordCloud i

mport matplotlib.pyplot as plt

import re

- Mengimpor pustaka WordCloud untuk membuat cloud kata, matplotlib.pyplot untuk menampilkan visualisasi, dan re untuk operasi ekspresi reguler.

text_combined = ' '.join(data['Text'])

- Menggabungkan semua teks dari kolom Text dalam DataFrame data menjadi satu

string panjang, dipisahkan oleh spasi.

text_filtered = re.sub(r'http\S+', '', text_combined) text_filtered = re.sub(r't.co\S+', '', text_filtered)

- Menghapus semua URL dari string teks menggunakan ekspresi reguler.

wordcloud = WordCloud(width=800, height=400, background_color='white').generate(text_filtered)

- Membuat objek WordCloud dengan lebar 800 piksel, tinggi 400 piksel, dan latar belakang putih, lalu menghasilkan cloud kata dari teks yang telah difilter.

plt.figure(figsize=(10, 6))

- Membuat sebuah figure baru dengan ukuran 10 inci x 6 inci.

plt.imshow(wordcloud, interpolation='bilinear')

- Menampilkan cloud kata pada figure dengan interpolasi bilinear untuk meningkatkan kualitas gambar.

plt.title('Kata-kata Umum')

- Menambahkan judul "Kata-kata Umum" pada gambar.

plt.axis('off')

- Menyembunyikan sumbu pada plot sehingga hanya gambar cloud kata yang terlihat.

plt.show()

- Menampilkan cloud kata yang telah dibuat.

3. Analisis Kata yang Sering Muncul (Angka/Jumlah)

- Tujuan: Memahami kata-kata spesifik yang sering muncul dalam setiap kategori sentimen untuk mendalami karakteristik masing-masing kelas.
- Proses: Menghitung frekuensi kata dalam setiap kategori sentimen (positif, negatif, netral) untuk menemukan pola atau kata kunci yang sering dikaitkan dengan sentimen tertentu.

```
Kata-kata umum dalam sentimen positif: [('#AniesMuhaimin2024', 986), ('#YukJadiJubirAMIN', 316),
Kata-kata umum dalam sentimen negatif: [('|', 595), ('#AniesMuhaimin2024', 165), ('Tegal', 110),
Kata-kata umum dalam sentimen netral: [('#AniesMuhaimin2024', 220), ('Anies', 31), ('dan', 22),
```

Gambar 4.9 Kata yang sering muncul (jumlah)

```
from collections import Counter
from wordcloud import STOPWORDS

# Menggabungkan teks berdsarkan sentimen
positive_text = ' '.join(data[data['Sentiment'] == 'Positive']['Text'])
negative_text = ' '.join(data[data['Sentiment'] == 'Negative']['Text'])
neutral_text = ' '.join(data[data['Sentiment'] == 'Neutral']['Text'])

# Membuat fungsi
def get_common_words(text, num_words=20):
    words = text.split()
    common_words = Counter(words).most_common(num_words)
    return common_words

# Mengambil kata kata umum tiap sentimen
common_words_positive = get_common_words(positive_text)
common_words_negative = get_common_words(negative_text)
common_words_neutral = get_common_words(neutral_text)

# Menampilkan
print("Kata-kata umum dalam sentimen positif:", common_words_positive)
print("Kata-kata umum dalam sentimen negatif:", common_words_negative)
print("Kata-kata umum dalam sentimen netral:", common_words_neutral)
```

Gambar 4.10 Kodingan kata yang sering muncul

from collections import Counter from wordcloud import STOPWORDS

- Mengimpor Counter dari pustaka collections untuk menghitung frekuensi kata, dan STOPWORDS dari wordcloud untuk mengakses daftar kata berhenti.

positive_text = ' '.join(data[data['Sentiment'] == 'Positive']['Text'])
negative_text = ' '.join(data[data['Sentiment'] == 'Negative']['Text'])
neutral_text = ' '.join(data[data['Sentiment'] == 'Neutral']['Text'])

- Menggabungkan teks berdasarkan sentimen dari DataFrame data menjadi satu string panjang untuk setiap sentimen: positif, negatif, dan netral.

def get_common_words(text, num_words=20): words = text.split() common_words = Counter(words).most_common(num_words) return common_words

- Mendefinisikan fungsi get_common_words untuk mendapatkan kata-kata yang paling umum dalam teks. Fungsi ini menerima teks dan jumlah kata yang ingin diambil sebagai parameter, lalu mengembalikan daftar kata yang paling umum beserta frekuensinya.

common_words_positive = get_common_words(positive_text)
common_words_negative = get_common_words(negative_text)
common_words_neutral = get_common_words(neutral_text)

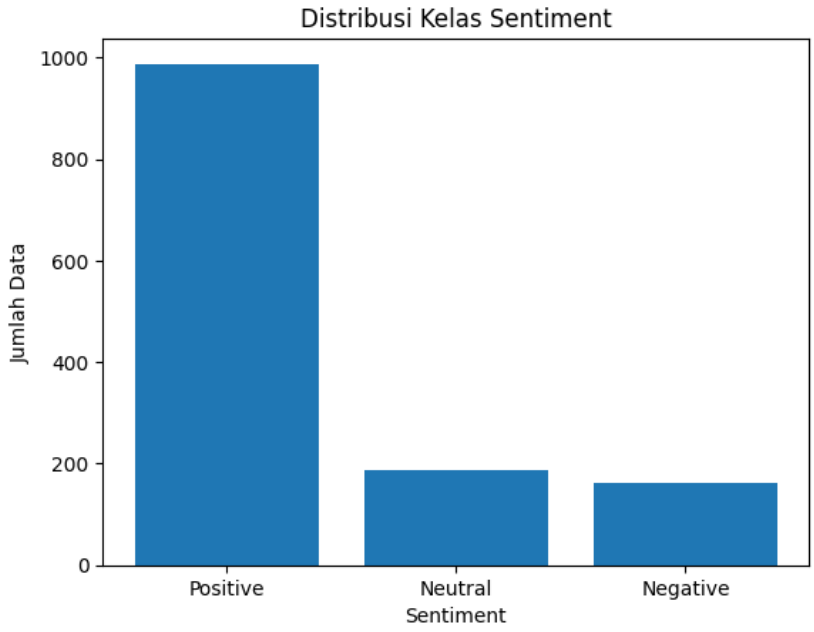
- Mengambil kata-kata yang paling umum dari teks positif, negatif, dan netral menggunakan fungsi get_common_words.

print("Kata-kata umum dalam sentimen positif:", common_words_positive) print("Kata-kata umum dalam sentimen negatif:", common_words_negative) print("Kata-kata umum dalam sentimen netral:", common_words_neutral)

- Menampilkan kata-kata yang paling umum dalam teks positif, negatif, dan netral dengan mencetak hasilnya ke layar.

4. Distribusi Kelas Sentimen

- Tujuan: Memastikan distribusi yang adil dan seimbang antara kategori sentimen.
- Proses: Menghitung dan menganalisis jumlah tweet dalam setiap kategori sentimen untuk mengidentifikasi apakah ada ketidakseimbangan yang perlu diperbaiki.



Gambar 4.11 Distribusi Kelas Sentimen

```
import matplotlib.pyplot as plt


class_counts = data['Sentiment'].value_counts()

# Tampilan
plt.bar(class_counts.index, class_counts.values)
plt.xlabel('Sentiment')
plt.ylabel('Jumlah Data')
plt.title('Distribusi Kelas Sentiment')
plt.show()

print(class_counts)
```

Gambar 4.12 Kodingan Distribusi Kelas

Penjelasan kodingan :

import matplotlib.pyplot as plt

- Mengimpor pustaka matplotlib.pyplot dengan alias plt untuk membuat dan menampilkan visualisasi grafis.

class_counts = data['Sentiment'].value_counts()

- Menghitung jumlah data untuk setiap sentimen dalam kolom Sentiment pada DataFrame data dan menyimpannya dalam class_counts.

plt.bar(class_counts.index, class_counts.values) plt.xlabel('Sentiment') plt.ylabel('Jumlah Data') plt.title('Distribusi Kelas Sentiment') plt.show()

- Membuat grafik batang untuk menunjukkan distribusi kelas sentimen.
- Menggunakan indeks dari class_counts sebagai label sumbu x.

- Menggunakan nilai dari class_counts sebagai tinggi batang.
- Menambahkan label "Sentiment" pada sumbu x.
- Menambahkan label "Jumlah Data" pada sumbu y.
- Menambahkan judul "Distribusi Kelas Sentiment" pada grafik.
- Menampilkan grafik batang.

print(class_counts)

- Mencetak jumlah data untuk setiap sentimen ke layar.

4.5 Pre-pemrosesan Data

Pra-pemrosesan adalah langkah penting untuk membersihkan dan menyiapkan data sebelum analisis lebih lanjut. Langkah-langkah yang dilakukan meliputi:

- Tokenisasi

  Tujuan: Memecah teks tweet menjadi unit-unit kata atau token untuk analisis.
  Proses: Menggunakan teknik tokenisasi untuk membagi teks menjadi kata-kata atau frasa individual yang akan dianalisis lebih lanjut.

- Penghapusan Stopwords:

  Tujuan: Menghapus kata-kata umum yang tidak memberikan informasi penting untuk analisis sentimen.
  Proses: Menggunakan daftar stopwords Bahasa Indonesia untuk menghapus kata-kata seperti "dan", "yang", "di", yang sering muncul.

- *Stemming*:

  Tujuan: Mengubah kata-kata ke bentuk dasarnya untuk menyederhanakan variasi kata.
  Proses: Menggunakan pustaka Sastrawi untuk mereduksi kata ke bentuk dasar, misalnya mengubah "berjalan" menjadi "jalan", sehingga variasi kata yang sama diperlakukan secara konsisten dalam analisis.

Langkah-langkah ini memastikan bahwa data yang digunakan dalam model adalah bersih, konsisten, dan siap untuk proses klasifikasi sentimen.

```
# Contoh teks
text = "Hujan turun deras di sore hari, membuat udara menjadi sejuk"
```

Gambar 4.13 Contoh teks

```
Tokens: ['Hujan', 'turun', 'deras', 'di', 'sore', 'hari', ',', 'membuat', 'udara', 'menjadi', 'sejuk']
Tokens after stopword removal: ['Hujan', 'turun', 'deras', 'sore', ',', 'udara', 'sejuk']
Tokens after stemming: ['hujan', 'turun', 'deras', 'sore', '', 'udara', 'sejuk']
```

Gambar 4.14 Hasil setelah di pre-proses

- *Pad Sequence*

  Tujuan : Menyeragamkan panjang urutan token sehingga semua input memiliki panjang yang sama, memungkinkan model pembelajaran mesin untuk memprosesnya secara efektif. Proses : Menambahkan nilai kosong (biasanya nol) ke urutan token yang lebih pendek sehingga semua urutan memiliki panjang yang sama.

```
# Tokenisasi dan padding teks
vocab_size = 6130
tokenizer = Tokenizer(num_words=vocab_size)
tokenizer.fit_on_texts(data['Processed_Text'])

sequences = tokenizer.texts_to_sequences(data['Processed_Text'])
max_length = 428
padded_sequences = pad_sequences(sequences, maxlen=max_length)

# Pastikan jumlah sampel konsisten
print("Jumlah sampel pada padded_sequences:", padded_sequences.shape[0])
```

Gambar 4.15 *Pad Sequence*

Penjelasan Kodingan :

Tokenisasi Teks:

- vocab_size = 6130: Menetapkan ukuran kosakata maksimum.
- tokenizer = Tokenizer(num_words=vocab_size): Membuat objek tokenizer dengan batasan jumlah kata sebesar vocab_size.
- tokenizer.fit_on_texts(data['Processed_Text']): Melatih tokenizer pada teks yang sudah diproses (Processed_Text).
- sequences = tokenizer.texts_to_sequences(data['Processed_Text']): Mengonversi teks menjadi urutan token.

Padding Teks:

- max_length = 428: Menetapkan panjang maksimum untuk padding.

- padded_sequences = pad_sequences(sequences, maxlen=max_length): Melakukan padding pada urutan token hingga panjang maksimum max_length.

Memastikan Konsistensi Jumlah Sampel:

- print("Jumlah sampel pada padded_sequences:", padded_sequences.shape[0]): Mencetak jumlah sampel dalam padded_sequences.

```
Padded Sequences shape: (3, 428)
Padded Sequences example:
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 2 3 4 5 6]
```

Gambar 4.16 Hasil *Pad Sequence*

4.6 Metodologi dan Algoritma

Saat menganalisis sentimen masyarakat mengenai Pemilihan Umum 2024 ini, penulis menggunakan model *Long Short-Term Memory* (LSTM), yang merupakan jenis jaringan saraf berulang (*Recurrent Neural Network*, RNN) yang mampu menangani masalah long-term dependencies dalam data urutan. Struktur model LSTM yang digunakan dalam proyek ini adalah sebagai berikut:

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 428, 100)          613000

 lstm (LSTM)                 (None, 428, 128)          117248

 dropout (Dropout)           (None, 428, 128)          0

 lstm_1 (LSTM)               (None, 64)                49408

 dense (Dense)               (None, 64)                4160

 dropout_1 (Dropout)         (None, 64)                0

 dense_1 (Dense)             (None, 3)                 195

=================================================================
Total params: 784011 (2.99 MB)
Trainable params: 784011 (2.99 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Gambar 4.17 Struktur model LSTM

- Layer *Embedding*: Layer pertama adalah embedding layer yang mengubah integer menjadi *dense vectors of fixed size* (100). *Embedding* layer ini membantu dalam menangani data teks dengan mengubahnya menjadi representasi vektor yang dapat diproses oleh jaringan saraf.
- LSTM Layer Pertama: LSTM pertama memiliki 128 unit. Layer ini menangkap dependensi temporal dalam data teks, memungkinkan model untuk mengingat konteks dari kata-kata sebelumnya dalam tweet.
- *Dropout* Layer Pertama: *Dropout* sebesar 0.5 digunakan setelah LSTM pertama untuk mencegah *overfitting* dengan mengabaikan (*drop out*) setengah dari unit secara acak selama pelatihan.
- LSTM Layer Kedua: LSTM kedua memiliki 64 unit dan mengolah output dari LSTM pertama. Lapisan ini lebih lanjut memperkuat kemampuan model untuk menangkap dependensi temporal.
- *Dense* Layer Pertama: Layer dengan 64 unit yang menggunakan fungsi aktivasi ReLU. Layer ini menggabungkan fitur-fitur yang telah diambil oleh LSTM.
- *Dropout* Layer Kedua: *Dropout* kedua sebesar 0.5 digunakan setelah dense layer untuk mencegah *overfitting* lebih lanjut.
- *Dense* Layer Kedua: *Dense* layer terakhir dengan 3 unit yang menggunakan fungsi aktivasi *softmax*. Layer ini menghasilkan output berupa probabilitas untuk masing-masing kelas sentimen (positif, negatif, netral).

4.7 Proses Pelatihan Model

Proses pelatihan model dilakukan dengan menggunakan teknik pembagian data menjadi data pelatihan dan data validasi. Rincian proses pelatihan adalah sebagai berikut:

```
100%|██████████| 1336/1336 [09:10<00:00,  2.43it/s]Jumlah sampel pada padded_sequences: 1336
Jumlah sampel pada y_encoded: 1336
Jumlah data pelatihan: 1068
Jumlah data validasi: 268
```

Gambar 4.18 Pembagian Pelatihan Data

- **Jumlah Data Pelatihan:** 1068 sampel
- **Jumlah Data Validasi:** 268 sampel

```
Compile + Latih

model.compile(optimizer=Adam(), loss='sparse_categorical_crossentropy', metrics=['accuracy'])

# Latih model
history = model.fit(X_train, y_train, batch_size=48, epochs=27, validation_data=(X_val, y_val))
```

Gambar 4.19 Pelatihan Model

- **Optimizer:** Adam, yang merupakan metode pengoptimalan adaptif yang efisien.
- **Loss Function:** Sparse categorical crossentropy, yang cocok untuk masalah klasifikasi multi-kelas.
- **Batch Size:** 48, jumlah sampel yang diproses sebelum model diperbarui.
- **Epochs:** 27, jumlah iterasi penuh melalui *dataset* pelatihan.

4.8 Matrik Evaluasi

Model dievaluasi menggunakan metrik akurasi serta laporan klasifikasi yang mencakup *precision, recall, dan F1-score* untuk setiap kelas sentimen. Hasil evaluasi adalah sebagai berikut:

```
9/9 [==============================] - 6s 592ms/step
Accuracy: 0.8805970149253731
Classification Report:
              precision    recall  f1-score   support

           0       0.91      0.72      0.81        29
           1       0.82      0.56      0.67        48
           2       0.89      0.98      0.93       191

    accuracy                           0.88       268
   macro avg       0.87      0.76      0.80       268
weighted avg       0.88      0.88      0.87       268
```

Gambar 4.20 Hasil Training Model (Report)

- Akurasi: 0.88, menunjukkan model dapat mengklasifikasikan dengan benar 88% dari data validasi.
- *Precision, Recall, dan F1-score*: Metrik ini memberikan gambaran lebih mendalam

mengenai kinerja model untuk setiap kelas sentimen (0: negatif, 1: netral, 2: positif).

- *Precision*

  Tujuan: Mengukur seberapa tepat model dalam mengidentifikasi kelas positif.
  Definisi: *Precision* adalah rasio prediksi positif yang benar terhadap total prediksi positif. Formula: *Precision* = TP / (TP + FP)

- *Recall*

  Tujuan: Mengukur seberapa baik model dalam menemukan semua contoh kelas positif.
  Definisi: *Recall* adalah rasio prediksi positif yang benar terhadap total contoh sebenarnya yang positif. Formula: *Recall* = TP / (TP + FN)

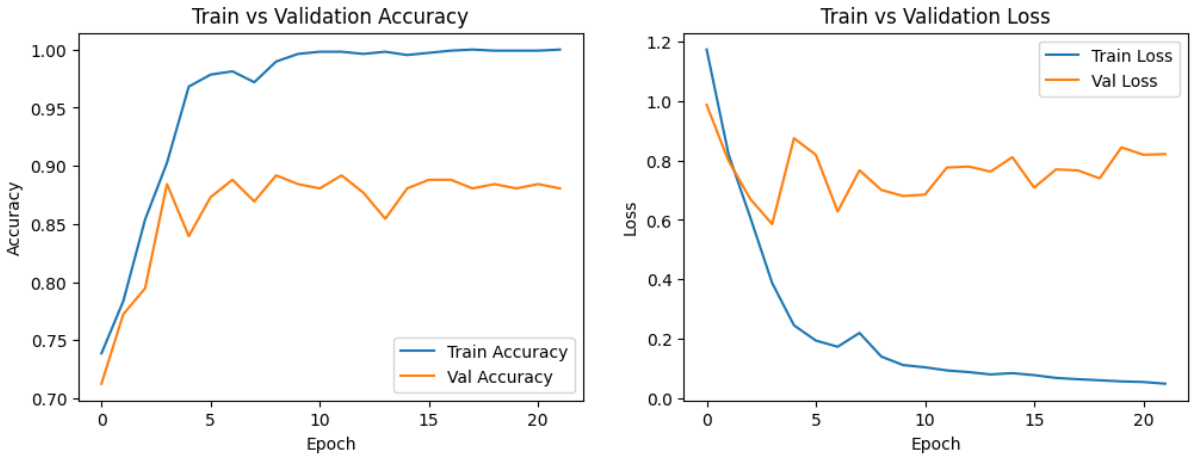
- *F1-Score*

  Tujuan: Menggabungkan *precision* dan *recall* menjadi satu metrik untuk mengukur keseimbangan antara keduanya.
  Definisi : *F1-Score* adalah rata-rata harmonis dari *precision* dan *recall*.

Metrik evaluasi ini menunjukkan bahwa model memiliki performa yang baik, terutama dalam mengklasifikasikan sentimen positif (kelas 2) dengan *precision* dan *recall* yang tinggi.

4.9 Hasil dan Analisa

Setelah melatih model LSTM diatas, berikut adalah hasil evaluasi kinerja model yang diperoleh. Grafik di bawah ini menunjukkan perbandingan antara akurasi dan loss selama proses pelatihan dan validasi.



Gambar 4.21 Pengukuran Akurasi

1. **Grafik Akurasi Pelatihan dan Validasi:**

- Grafik sebelah kiri menunjukkan bahwa akurasi pelatihan meningkat secara signifikan hingga mendekati 100% pada epoch ke-10 dan seterusnya. Namun, akurasi validasi berfluktuasi di sekitar 85%-90%, menunjukkan bahwa model mampu mempertahankan performa yang cukup baik pada data validasi.
- Fluktuasi pada akurasi validasi dapat mengindikasikan bahwa model mengalami sedikit overfitting, di mana model bekerja sangat baik pada data pelatihan tetapi kurang generalisasi pada data baru.

2. **Grafik *Loss* Pelatihan dan Validasi:**

- Grafik sebelah kanan menunjukkan penurunan loss yang cepat pada data pelatihan hingga mendekati nol setelah sekitar 10 epoch. Namun, loss pada data validasi cenderung stabil setelah mengalami penurunan awal, dengan nilai yang lebih tinggi dibandingkan loss pelatihan.
- Perbedaan yang signifikan antara loss pelatihan dan validasi juga menunjukkan tanda-tanda overfitting, di mana model cenderung menghafal data pelatihan daripada belajar pola

Meskipun model LSTM menunjukkan akurasi pelatihan yang sangat tinggi, fluktuasi akurasi dan *loss* pada data validasi menunjukkan bahwa metode pelatihan masih dapat ditingkatkan untuk mencapai generalisasi yang lebih baik. Berikut beberapa poin yang dapat penulis disimpulkan dari hasil ini adalah:

1. ***Overfitting:***

- Model mengalami overfitting, di mana performa pada data pelatihan jauh lebih baik dibandingkan data validasi.
- Salah satu penyebab *overfitting* yang penulis sadari adalah ketidakseimbangan jumlah data sentimen dalam *dataset*, di mana jumlah tweet positif jauh lebih banyak dibandingkan tweet netral dan negatif. Ketidakseimbangan ini dapat membuat model lebih bias terhadap sentimen yang dominan.

2. **Perbaikan yang Dapat Dilakukan:**

- Sebagai langkah pembelajaran, penulis menyadari bahwa untuk mengatasi *overfitting*, beberapa teknik seperti penambahan regularisasi (misalnya, L2 *regularization*), penggunaan dropout yang lebih besar, atau data augmentation untuk memperbanyak data sentimen yang kurang dapat diterapkan.
- Rebalancing *dataset* melalui teknik *undersampling* atau *oversampling*

BAB V
PENUTUP

5.1 Kesimpulan

Kerja Praktik ini dirancang untuk mengidentifikasi dan menjawab pertanyaan utama terkait dengan analisis sentimen masyarakat mengenai Pemilihan Umum 2024 yang diungkapkan melalui platform Twitter. Adapun rumusan masalah yang diangkat dalam Kerja Praktik ini meliputi pola sentimen masyarakat terhadap Pemilihan Umum 2024 dan seberapa akurat metode NLP dalam mengklasifikasikan sentimen dari tweet terkait Pemilihan Umum 2024.

Proyek analisis sentimen ini fokus pada deteksi sentimen masyarakat terkait Pemilihan Umum 2024 menggunakan teknik *Natural Language Processing* (NLP). Analisis ini bertujuan untuk mengklasifikasikan sentimen dari tweet yang diperoleh menjadi tiga kategori: positif, negatif, dan netral. Hasil Kerja Praktik menunjukkan bahwa pola sentimen masyarakat dapat diidentifikasi, namun akurasi dari model masih belum sempurna.

5.2 Saran

Adapun beberapa saran yang dapat penulis pertimbangkan kedepan adalah:

1. Bereksperimen dengan berbagai model dan teknik NLP yang lebih canggih. Menggunakan teknik pembelajaran mendalam (*deep learning*) seperti BERT atau GPT
2. Mengumpulkan data dari berbagai sumber selain Twitter, seperti Facebook, Instagram, atau forum online lainnya, dapat memberikan gambaran yang lebih lengkap dan representatif mengenai sentimen publik
3. Menambahkan metode pre-pemrosesing data pada *dataset*

DAFTAR PUSTAKA

Ali, Miss. A. A. S. (2021). AI-*Natural Language Processing* (NLP). *International Journal for Research in Applied Science and Engineering Technology*, *9*(VIII). https://doi.org/10.22214/ijraset.2021.37293

Gifari, O. I., Adha, Muh., Freddy, F., & Durrand, F. F. S. (2022). Analisis Sentimen Review Film Menggunakan TF-IDF dan Support Vector Machine. *Journal of Information Technology*, *2*(1). https://doi.org/10.46229/jifotech.v2i1.330

Hochreiter, S., & Schmidhuber, J. (1997). *Long Short-Term Memory*. *Neural Computation*, *9*(8). https://doi.org/10.1162/neco.1997.9.8.1735

Johri, P., Khatri, S. K., Al-Taani, A. T., Sabharwal, M., Suvanov, S., & Kumar, A. (2021). *Natural* Language Processing: History, Evolution, Application, and Future Work. *Lecture Notes in Networks and Systems*, *167*. https://doi.org/10.1007/978-981-15-9712-1_31

Khurana, D., Koli, A., Khatter, K., & Singh, S. (2023). *Natural* language processing: state of the art, current trends and challenges. *Multimedia Tools and Applications*, *82*(3). https://doi.org/10.1007/s11042-022-13428-4

Putra, T. D., Utami, E., & Kurniawan, M. P. (2023). Analisis Sentimen Pemilu 2024 dengan Naive Bayes Berbasis Particle Swarm Optimization (PSO). *Explore*, *13*(1). https://doi.org/10.35200/ex.v11i2.13

Rianto, Mutiara, A. B., Wibowo, E. P., & Santosa, P. I. (2021). Improving the accuracy of text classification using *stemming* method, a case of non-formal Indonesian conversation. *Journal of Big Data*, *8*(1). https://doi.org/10.1186/s40537-021-00413-1

Rita, P., António, N., & Afonso, A. P. (2023). Social media discourse and voting decisions influence: sentiment analysis in tweets during an electoral period. *Social Network Analysis and Mining*, *13*(1). https://doi.org/10.1007/s13278-023-01048-1

Sarica, S., & Luo, J. (2021). Stopwords in technical language processing. *PLoS ONE*, *16*(8 August). https://doi.org/10.1371/journal.pone.0254937

Sherstinsky, A. (2020). Fundamentals of *Recurrent Neural Network* (RNN) and *Long Short-Term Memory* (LSTM) network. *Physica D: Nonlinear Phenomena*, *404*. https://doi.org/10.1016/j.physd.2019.132306

Webster, J. J., & Kit, C. (1992). *Tokenization as the initial phase in NLP*. https://doi.org/10.3115/992424.992434