

**DETEKSI SARKASME KALIMAT BERBAHASA INGGRIS  
PADA MEDIA SOSIAL TWITTER MENGGUNAKAN MODEL  
*ROBUSTLY OPTIMIZED BERT APPROACH (ROBERTA)***

**LAPORAN TUGAS AKHIR**

Laporan Ini Disusun Untuk Memenuhi Salah Satu Syarat Memperoleh Gelar Sarjana Strata 1 (S1) Pada Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang



**Disusun Oleh :**

**NAMA : HILDA PUTRI CAHYANIGRUM**

**NIM : 32602100053**

**FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM SULTAN AGUNG  
SEMARANG**

**2025**

**FINAL PROJECT**  
**SARCASM DETECTION IN ENGLISH SENTENCES ON**  
**TWITTER SOCIAL MEDIA USING THE ROBUSTLY**  
**OPTIMIZED BERT APPROACH (ROBERTA) MODEL**

Proposed to complete the requirements for obtaining a Bachelor's Degree (S1) in Informatics Engineering, Faculty of Industrial Technology, Sultan Agung Islamic University, Semarang.



**FAKULTAS TEKNOLOGI INDUSTRI**  
**UNIVERSITAS ISLAM SULTAN AGUNG**  
**SEMARANG**

**2025**

LEMBAR PENGESAHAN  
TUGAS AKHIR

DETEKSI SARKASME KALIMAT BERBAHASA INGGRIS PADA  
MEDIA SOSIAL TWITTER MENGGUNAKAN MODEL *ROBUSTLY  
OPTIMIZED BERT APPROACH* (ROBERTA)

HILDA PUTRI CAHYANINGRUM  
NIM 32602100053

Telah dipertahankan di depan tim penguji ujian sarjana tugas akhir  
Program Studi Teknik Informatika  
Universitas Islam Sultan Agung  
Pada tanggal : 27/02/2025

TIM PENGUJI UJIAN SARJANA :

Mustafa, ST, MM, M. Kom

NIK. 210610040  
(Ketua Penguji)

11/03/2025

Moch Taufik, ST., MIT

NIK. 210604034  
(Anggota Penguji)

10/03/2025

Badie'ah ST., M. Kom

NIK. 210615044  
(Pembimbing)

10/03/2025

Semarang, 13/03/2025

Mengetahui,

Kaprodi Teknik Informatika  
Universitas Islam Sultan Agung



Moch. Taufik, S.T., M.IT  
0604034

## SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : Hilda Putri Cahyaningrum

NIM : 32602100053

Judul Tugas Akhir : Deteksi Sarkasme Kalimat Berbahasa Inggris pada Media Sosial Twitter Menggunakan Model *Robustly Optimized BERT Approach* (RoBERTa)

Dengan bahwa ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apabila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, 13 Maret 2025

Yang Menyatakan,



Hilda Putri Cahyaningrum

## PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama : Hilda Putri Cahyaningrum

NIM : 32602100053

Program Studi : Teknik Informatika

Fakultas : Teknologi Industri

Alamat Asal : Jepara

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul : Deteksi Sarkasme Kalimat Berbahasa Inggris pada Media Sosial Twitter Menggunakan Model *Robustly Optimized BERT Approach* (RoBERTa) Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak bebas Royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dan pangkalan data dan dipublikasikan diinternet dan media lain untuk kepentingan akademis selama tetap menyantumkan nama penulis sebagai pemilik hak cipta. Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari terbukti ada pelanggaran Hak Cipta/Plagiarisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan agung.

Semarang, 13 Maret 2025

Yang menyatakan,



Hilda Putri Cahyaningrum

## KATA PENGANTAR

Dengan mengucapkan syukur Alhamdulillah atas kehadiran Allah swt atas limpahan rahmat dan karunia-Nya kepada penulis, sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “Deteksi Sarkasme Kalimat Berbahasa Inggris pada Media Sosial Twitter Menggunakan Model Robustly Optimized BERT Approach (RoBERTa)”. Tugas akhir ini disusun sebagai salah satu syarat untuk memperoleh gelar sarjana strata 1 (S1) diprogram studi teknik informatika, fakultas teknologi industri, universitas islma sultan agung semarang.

Penulisan tugas akhir ini tidak lepas dari dukungan dari berbagai pihak. Oleh karena itu, saya ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada :

1. Rektor UNISSULA Bapak Prof. Dr. H. Gunarto, S.H., M.H yang mengizinkan penulis menimba ilmu dikampus ini.
2. Dekan Fakultas Teknologi Industri Ibu Dr. Ir. Novi Marlyana, S.T., M.T., IPU., ASEAN. Eng.
3. Dosen Pembimbing penulis Ibu Badie’ah, S.T., M. Kom yang telah sabar membimbing saya hingga akhir dan memberikan banyak nasehat dan saran.
4. Ucapan terima kasih yang sebesar-besarnya kepada Ibu dan bapak yang selalu melimpahkan kasih sayang yang tulus, doa yang tiada henti, dukungan yang begitu besar dalam kehidupan penulis, serta kepercayaan untuk saya dapat melanjutkan studi dan menyelesaikannya sampai akhir.
5. Para sahabat, teman-teman yang telah memberikan begitu banyak bantuan, semangat, inspirasi dan diskusi progres penyusunan Tugas Akhir.

Saya menyadari bahwa skripsi ini masih jauh dari sempurna. Oleh karena itu, saya mengharapkan kritik dan saran yang membangun bagi para pembaca. Semoga skripsi ini dapat memberikan manfaat bagi perkembangan ilmu pengetahuan dan memberikan inspirasi bagi para pembaca.

Semarang, 27 Februari 2025

Hilda Putri Cahyaningrum

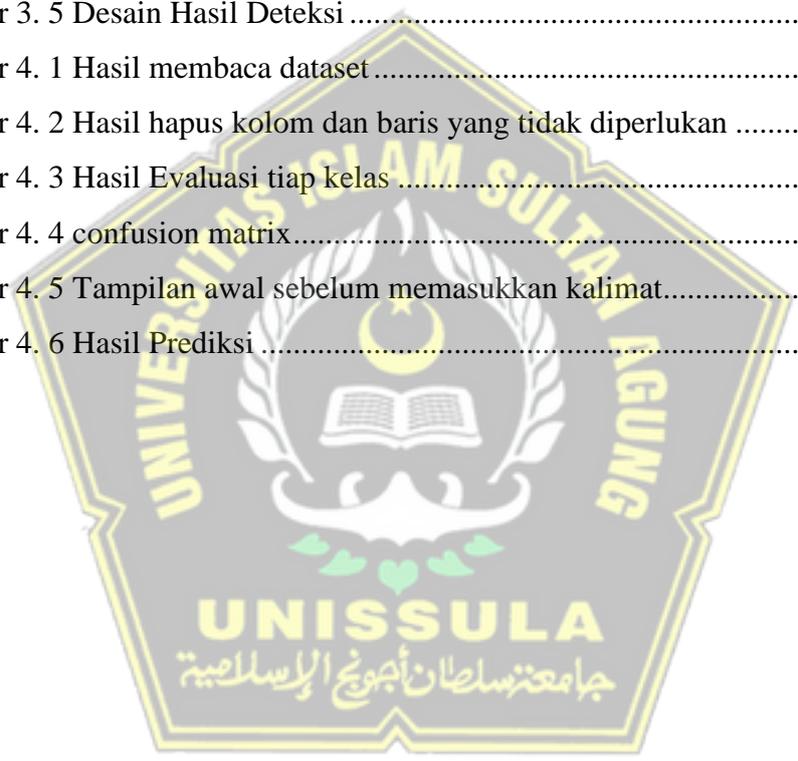
## DAFTAR ISI

<b>HALAMAN JUDUL .....</b>	<b>i</b>
<b>LEMBAR PENGESAHAN TUGAS AKHIR.....</b>	<b>iii</b>
<b>SURAT PERNYATAAN KEASLIAN TUGAS AKHIR.....</b>	<b>iv</b>
<b>PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH .....</b>	<b>v</b>
<b>KATA PENGANTAR.....</b>	<b>vi</b>
<b>DAFTAR ISI.....</b>	<b>vii</b>
<b>DAFTAR GAMBAR.....</b>	<b>ix</b>
<b>DAFTAR TABEL .....</b>	<b>x</b>
<b>ABSTRAK .....</b>	<b>xi</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1    Latar Belakang .....	1
1.2    Rumusan Masalah.....	2
1.3    Batasan Masalah.....	2
1.4    Tujuan .....	3
1.5    Manfaat .....	3
1.6    Sistematika Penulisan .....	3
<b>BAB II TINJAUAN PUSTAKA DAN DASAR TEORI.....</b>	<b>5</b>
2.1    Tinjauan Pustaka.....	5
2.2    Dasar Teori.....	9
2.2.1 <i>Natural Language Processing (NLP)</i> .....	9
2.2.2 <i>Multi-Class Classification</i> .....	9
2.2.3 <i>Trasformer</i> .....	10
2.2.4 <i>Bidirectional Encoder Representations from Transformers (BERT)</i> 12	
2.2.5 <i>Robustly Optimized BERT Pretraining Approach (RoBERTa)</i> .....	13
2.2.6    Metriks Evaluasi.....	14
<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>17</b>

3.1	Metode Penelitian.....	17
3.1.1	Studi Literatur .....	17
3.1.2	Pengumpulan Data .....	17
3.1.3	Pemodelan Sistem .....	18
3.2	Analisis Sistem.....	20
3.3	Analisis Kebutuhan .....	20
3.4	Desain <i>User Interface</i> .....	24
3.4.1	Tampilan awal.....	24
3.4.2	Memasukkan kalimat .....	25
3.4.3	Hasil deteksi .....	26
<b>BAB IV HASIL DAN ANALISIS PENELITIAN .....</b>		<b>27</b>
4.1	Persiapan data.....	27
4.2	<i>Preprocessing</i> .....	28
4.2.1	<i>Cleansing Data</i> .....	28
4.2.2	<i>Tokenization</i> .....	32
4.3	Optimasi model.....	34
4.3.1	Membagi data.....	34
4.3.2	<i>Parameter</i> .....	34
4.4	<i>Fine Tuning</i> .....	35
4.5	<i>Hasil</i> .....	36
4.6	Tampilan <i>User interface</i> .....	38
<b>BAB V KESIMPULAN DAN SARAN .....</b>		<b>39</b>
5.1	Kesimpulan .....	39
5.2	Saran.....	39
<b>DAFTAR PUSTAKA</b>		
<b>LAMPIRAN</b>		

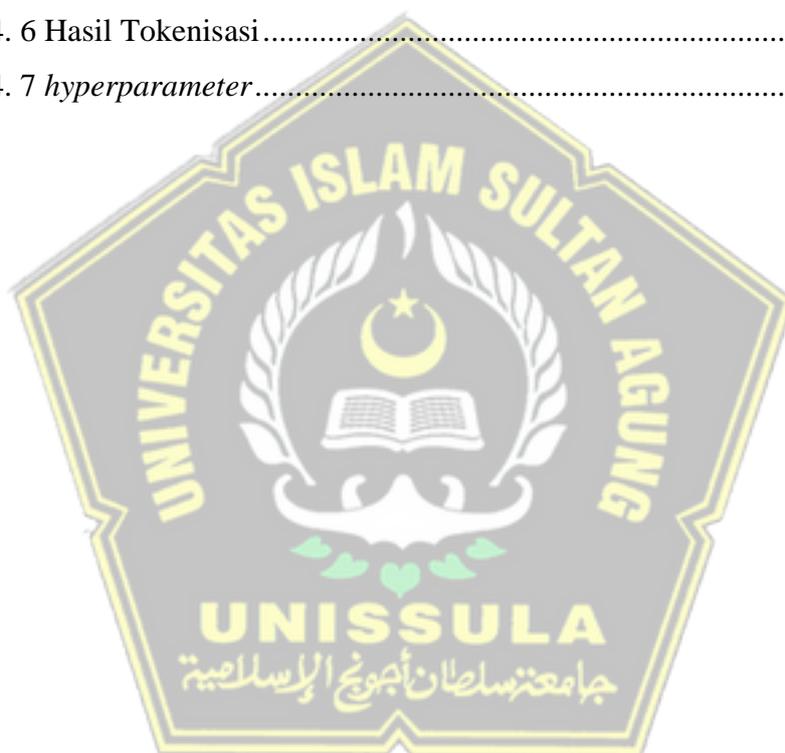
## DAFTAR GAMBAR

Gambar 2. 1 arsitektur tranformer(Vaswani dkk., 2017) .....	10
Gambar 2. 2 <i>Pre-training</i> dan <i>Fine-Tuning</i> BERT .....	13
Gambar 3. 1 <i>flowchart</i> metode penelitian .....	17
Gambar 3. 2 Pemodelan Sistem .....	18
Gambar 3. 3 Desain Tampilan Awal <i>User Interface</i> .....	24
Gambar 3. 4 Desain Input kalimat .....	25
Gambar 3. 5 Desain Hasil Deteksi .....	26
Gambar 4. 1 Hasil membaca dataset .....	27
Gambar 4. 2 Hasil hapus kolom dan baris yang tidak diperlukan .....	28
Gambar 4. 3 Hasil Evaluasi tiap kelas .....	36
Gambar 4. 4 confusion matrix.....	37
Gambar 4. 5 Tampilan awal sebelum memasukkan kalimat.....	38
Gambar 4. 6 Hasil Prediksi .....	38



## DAFTAR TABEL

Tabel 1. 1 Tabel Sistematika Penulisan .....	3
Tabel 4. 1 <i>Replace Special Karakter</i> .....	29
Tabel 4. 2 menghapus emoji .....	30
Tabel 4. 3 hapus <i>tag username</i> .....	30
Tabel 4. 4 mengganti kontraksi kata bahasa inggris .....	31
Tabel 4. 5 Hapus <i>stopworsd, lower casting, dll</i> .....	32
Tabel 4. 6 Hasil Tokenisasi .....	33
Tabel 4. 7 <i>hyperparameter</i> .....	35



## ABSTRAK

Bahasa memegang peranan vital sebagai alat komunikasi yang membedakan manusia, dengan fungsi utamanya sebagai pertukaran informasi. Di era digital ini, media sosial telah merevolusi cara interaksi, namun penggunaan yang tidak bijak, seperti penggunaan sarkasme dalam bermedia sosial dapat berdampak negatif. Sarkasme, sebagai gaya bahasa sindiran yang tajam, menjadi tantangan dalam komunikasi digital, terutama di platform seperti Twitter (X) dengan keterbatasan karakter dan ketiadaan isyarat non-verbal. Penelitian ini bertujuan untuk menerapkan dan menganalisis kinerja model RoBERTa dalam mendeteksi sarkasme dalam percakapan berbahasa Inggris di Twitter (X). Sarkasme, sebagai bentuk komunikasi yang kompleks dan seringkali ambigu, dapat menimbulkan kesalahpahaman dan bahkan pelanggaran hukum di ruang digital. Melalui analisis mendalam terhadap dataset yang terdiri dari empat kategori (ironi, sarkasme, figuratif, dan reguler), penelitian ini menganalisis kemampuan model RoBERTa dalam mengidentifikasi nuansa makna yang halus. Hasil penelitian menunjukkan bahwa model mencapai akurasi keseluruhan sebesar 89%, dengan performa yang sangat baik dalam mengidentifikasi ironi, sarkasme, dan kalimat reguler. Penelitian ini menekankan pentingnya pengembangan sistem deteksi sarkasme yang lebih akurat untuk meminimalkan dampak negatif dari miskomunikasi di media sosial dan mendukung keamanan digital.

Kata kunci : RoBERTa, Deteksi Sarkasme, Media Sosial

### ABSTRACT

*Language plays a vital role as a communication tool that differentiates humans, with its main function being the exchange of information. In this digital era, social media has revolutionized the way of interaction, but unwise use, such as the use of sarcasm in social media, can have a negative impact. Sarcasm, as a sharp satirical style of language, is a challenge in digital communication, especially on platforms like Twitter (X) with limited characters and the absence of non-verbal cues. This research aims to apply and analyze the performance of the RoBERTa model in detecting sarcasm in English conversations on Twitter (X). Sarcasm, as a complex and often ambiguous form of communication, can lead to misunderstandings and even legal violations in digital spaces. Through in-depth analysis of a dataset consisting of four categories (irony, sarcasm, figurative, and regular), this research analyzes the ability of the RoBERTa model to identify subtle nuances of meaning. The results showed that the model achieved an overall accuracy of 89%, with excellent performance in identifying irony, sarcasm, and regular sentences.. This research emphasizes the importance of developing more accurate sarcasm detection systems to minimize the negative impact of miscommunication on social media and support digital security.*

*Keywords: RoBERTa, Sarcasm Detection, Social Media*

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Bahasa berperan menjadi alat komunikais yang vital yang sering digunakan dan membedakan antara manusia satu dengan yang lainnya. Fungsi utama Bahasa adalah sebagai alat bertukar informasi. Sebagai alat komunikasi yang dinamis Bahasa dapat di ekspresikan dalam bentuk verbal maupun tertulis, baik dalam komunikasi tatap muka maupun melalui media sosial (Andriarsih & Asriyani, 2020). Dalam era digital seperti sekarang ini media sosial telah merevolusi cara kita berinteraksi menjadi media komunikasi *online* dan dapat mengakses informasi apa saja dari pengguna di seluruh dunia. Akan tetapi, penggunaan media sosial yang tidak bijak dapat memberikan dampak buruk bagi orang lain bahkan pada diri sendiri, seperti penggunaan kata-kata sarkasme saat bersosial media.

Sarkasme merupakan suatu gaya bahasa yang memiliki ciri penggunaan kata sindiran yang bertujuan untuk melukai perasaan orang lain. Dibandingkan dengan ironi dan sinisme, sarkasme Tingkat kepahitannya lebih tinggi dan cenderung lebih kasar (Hilmawan, 2022). Penggunaan kata sarkasme yang memiliki arti demikian kompleks dalam praktiknya menjadi tantangan dalam komunikais digital dan bersosial media, terutama pada *platform* seperti twitteratau sekarang X. Twitter atau sekarang X merupakan salah satu *platorm* media sosial yang memungkinkan pengguna untuk berbagi pesan singkat atau biasa disebut *tweet* yang memiliki keterbatasan karakter dengan maksimal 280 karakter. Terbatasnya karakter yang dapat digunakan terlebih lagi tidak adanya isyarat non-verbal seperti ekspresi wajah atau nada suara menjadikan pesan yang disampaikan menjadi tidak jelas yang sulit diartikan sehingga munculnya salah tafsir atau bahkan multi tafsir, terutama erat kaitannya dengan penggunaan sarkasme yang memperumit Upaya pemahaman makna yang sebenarnya dari *tweet* yang dibaca.(Fitrianto & Editya, 2024)

Terjadinya salah tafsir terhadap ketajaman sakasme dapat melukai perasaan orang lain dan justru rentan terhadap penyalahgunaan dan pelanggaran

hukum. Seperti yang diatur dalam UU ITE No. 11 Tahun 2008 yang diubah dengan UU No. 19 Tahun 2016 Pasal 27 ayat 1 atau ayat 3 sarkasme dapat masuk ke dalam salah satu kategori pelanggarannya (Sarwono, 2024). Untuk itu pentingnya dalam memahami kalimat berbentuk sarkasme, dengan memahami penggunaan sarkasme di media sosial merupakan salah satu bentuk Upaya kita berkontribusi terhadap kemandirian digital dan perlindungan *public* dari potensi miskomunikasi yang merugikan. Hal ini sejalan dengan Upaya nasional maupun internasional untuk mendorong kemandirian dan ketertiban di ruang digital.

Seiring meningkatnya interaksi melalui *platform online*, deteksi sarkasme menjadi semakin penting untuk meminimalkan dampak negatif dari salah tafsir. Di sinilah penelitian ini berperan, yaitu untuk membantu memahami urgensi dan relevansi penerapan teknologi pemrosesan Bahasa alami, khususnya model RoBERTa dalam mengidentifikasi sarkasme penelitian ini juga bertujuan untuk menjawab kebutuhan akan sistem yang dapat menangkap makna tersembunyi dalam komunikasi digital, terutama dalam konteks twitter atau sekarang X yang sering menjadi wadah ekspresi publik.

## 1.2 Rumusan Masalah

Rumusan berdasarkan identifikasi masalah penelitian ini yaitu :

1. Bagaimana penerapan teknologi pemrosesan bahasa alami, khususnya model RoBERTa, dapat membantu dalam mengidentifikasi sarkasme di media sosial dan meminimalkan dampak negatif dari salah tafsir?
2. Sejauh mana akurasi dan efisiensi model pemrosesan bahasa alami, seperti RoBERTa, dalam mendeteksi sarkasme di berbagai konteks komunikasi digital?

## 1.3 Batasan Masalah

Adapun Batasan masalah dalam tugas akhir ini adalah :

1. Model hanya dapat mendeteksi tiga *class* yaitu *irony*, *sarcasm*, dan *regular*.
2. Sistem deteksi hanya dapat mendeteksi kalimat berbahasa inggris.
3. Analisis terbatas pada teks tanpa emoji dan karakter khusus lainnya.

#### 1.4 Tujuan

Tujuan dari Penelitian ini yaitu :

1. Menerapkan model RoBERTa untuk membangun sistem yang dapat mendeteksi kalimat sarkasme berbahasa Inggris dalam penggunaan media sosial
2. Menganalisis kinerja sistem dalam mengklasifikasi kata berdasarkan label yang sudah ditentukan.

#### 1.5 Manfaat

1. Membantu memilah informasi ambigu dengan mengidentifikasi keberadaan sarkasme dalam teks.
2. Mengurangi penyebaran informasi palsu yang disamarkan sebagai opini atau fakta dengan mendeteksi sarkasme yang digunakan
3. Mencegah miskomunikasi yang dapat menimbulkan kerugian dengan memahami konteks sarkasme dalam komunikasi *online*
4. Mendukung upaya untuk menciptakan ruang digital yang lebih aman dan tertib.

#### 1.6 Sistematika Penulisan

Pada penelitian ini menerapkan sistematika penulisan sebagai berikut :

Tabel 1. 1 Tabel Sistematika Penulisan

BAB I : Pendahuluan

Bab pertama akan berisi latar belakang yang membahas dan menjelaskan mengenai urgensi dari penelitian. Pembahasan dan penjelasan yang akan dibagi menjadi beberapa yaitu latar belakang, perumusan masalah, pembatasan masalah, tujuan, manfaat, dan sistematika penulisan.

BAB II : Tinjauan Pustaka dan Dasar Teori

Bab dua akan berisi tinjauan Pustaka dan dasar teori. Tujuan

bab ini adalah untuk menunjukkan Pustaka penelitian sebelumnya dan dasar teori yang akan digunakan pada penelitian.

BAB III : Metode Penelitian

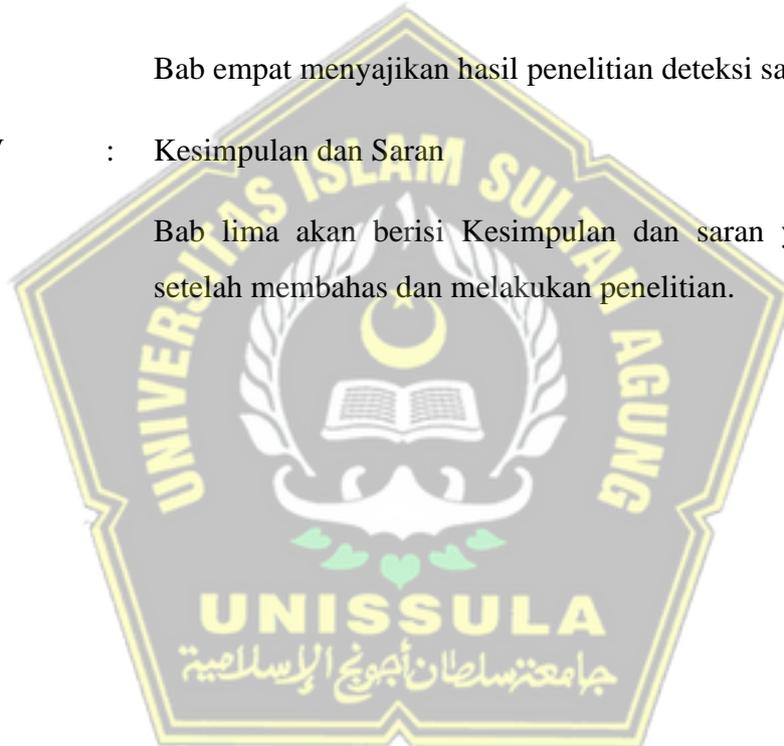
Bab tiga membahas *metode* yang akan digunakan dalam penelitian termasuk perancangan yang berupa *flowchart*.

BAB IV : Hasil dan Analisis Penelitian

Bab empat menyajikan hasil penelitian deteksi sarcasm

BAB V : Kesimpulan dan Saran

Bab lima akan berisi Kesimpulan dan saran yang dibuat setelah membahas dan melakukan penelitian.



## BAB II

### TINJAUAN PUSTAKA DAN DASAR TEORI

#### 2.1 Tinjauan Pustaka

Penelitian dalam bidang pemrosesan bahasa alami (NLP) terus berkembang pesat, terutama dalam tugas analisis sentimen dan klasifikasi teks. Beberapa penelitian terbaru telah menunjukkan kemajuan signifikan dalam memanfaatkan model-model *deep learning* dan transformer untuk mencapai hasil yang lebih akurat dan relevan.

Salah satu contohnya pada penelitian (Kusrini & Sudarmawan, 2020) tentang peningkatan akurasi klasifikasi sentimen ulasan makanan amazon dengan *bidirectional LSTM* dan *BERT embedding*. Penelitian ini berhasil meningkatkan akurasi dalam mengklasifikasikan sentimen ulasan makanan di Amazon menggunakan kombinasi model *deep learning Bidirectional LSTM* dan model bahasa BERT. Model BiLSTM mampu menangkap konteks kalimat secara menyeluruh, sementara BERT memberikan representasi kata yang sangat baik berdasarkan konteksnya dengan akurasi sebesar 93%. Hasil penelitian menunjukkan bahwa model ini mampu mencapai akurasi yang jauh lebih tinggi dibandingkan dengan metode-metode sebelumnya. Hal ini mengindikasikan bahwa kombinasi kedua model ini sangat efektif dalam memahami bahasa alami dan dapat diaplikasikan pada berbagai tugas pemrosesan bahasa lainnya.

Selanjutnya, penelitian oleh (Alwan dkk., 2022) membahas pengembangan sebuah aplikasi deteksi judul berita *clickbait* berbasis *Chrome extension* menggunakan model *Natural Language Processing (NLP) Multilingual BERT*. Tujuannya adalah untuk membantu pengguna mengidentifikasi berita *clickbait* yang seringkali menyesatkan. Hasil penelitian menunjukkan bahwa model *Multilingual BERT uncased* berhasil mencapai akurasi 92% dalam mendeteksi *clickbait*. Aplikasi ini dibangun dengan arsitektur REST API untuk menghubungkan antar komponen dan menggunakan JavaScript untuk pengembangan ekstensinya. Dengan demikian, pengguna dapat dengan mudah mendeteksi *clickbait* hanya dengan menandai judul berita yang mencurigakan.

Selain analisis sentimen dan deteksi *clickbait*, model NLP juga dapat diterapkan dalam sistem rekomendasi berita penelitian yang dilakukan oleh (Juarto & Suganda, 2021) mengembangkan sebuah sistem rekomendasi berita yang lebih cerdas dengan menggabungkan dua metode populer, yaitu *neural collaborative filtering* dan *Sentence BERT*. Metode *neural collaborative filtering* memanfaatkan interaksi pengguna dengan berita sebelumnya untuk memberikan rekomendasi yang lebih personal, sementara *Sentence BERT* digunakan untuk memahami makna dari judul dan isi berita. Dengan menggabungkan kedua metode ini, sistem rekomendasi dapat memberikan saran berita yang tidak hanya relevan dengan preferensi pengguna, tetapi juga sesuai dengan isi berita yang sedang tren atau menarik perhatian pengguna. Hasil penelitian menunjukkan bahwa metode *hybrid* ini memberikan hasil yang lebih baik dibandingkan dengan metode tradisional, terutama dalam hal akurasi rekomendasi berita yang relevan. Secara sederhana, sistem ini dapat membantu pengguna menemukan berita yang menarik dan sesuai dengan minat mereka secara lebih efektif.

Penelitian lain yang menarik adalah penelitian oleh (Basbeth & Fudholi, 2024) bertujuan untuk mengklasifikasi emosi dalam teks bahasa Indonesia menggunakan model *deep learning*, khususnya BERT, RoBERTa, dan DistilBERT. Data yang digunakan berasal dari Twitter dan dilabeli dengan lima emosi dasar. Hasil penelitian menunjukkan bahwa model DistilBERT-*freeze* memberikan akurasi dan F1-*score* yang paling baik dalam mengklasifikasi emosi pada teks bahasa Indonesia. Hal ini disebabkan oleh penyetelan *hyperparameter* yang tepat dan teknik *fine-tuning* yang efektif. Penelitian ini berkontribusi pada pengembangan model bahasa alami untuk memahami emosi dalam teks bahasa Indonesia, yang memiliki potensi aplikasi yang luas dalam berbagai bidang, seperti analisis sentimen, pemasaran, dan layanan pelanggan.

Pada penelitian yang dilakukan oleh (Fitrianto & Editya, 2024), klasifikasi tweet sarkasme berbahasa Indonesia pada *platform X* (sebelumnya Twitter) menggunakan metode *deep learning* dengan fokus pada model-model turunan BERT, yaitu IndoBERT, BERT, RoBERTa, dan BERT Multilingual. Penelitian ini bertujuan untuk mengatasi kesulitan dalam mendeteksi sarkasme pada teks

yang tidak memiliki isyarat non-verbal, yang dapat berdampak pada analisis sentimen dan penggunaan sebagai bukti hukum. Hasil penelitian menunjukkan bahwa model IndoBERT, yang telah dirancang khusus untuk bahasa Indonesia, mencapai performa terbaik dengan skor F1 sebesar 95% pada dataset pengujian, mengungguli model-model BERT lainnya.

Penelitian lain yang dilakukan oleh (Syahwaluddin & Alita, 2024) membahas tentang efektivitas teknik *oversampling* SMOTE dalam meningkatkan kinerja model klasifikasi ujaran kebencian yang menggunakan BERT, di mana masalah ketidakseimbangan kelas menjadi tantangan utama. Hasil penelitian menunjukkan bahwa penerapan SMOTE meningkatkan akurasi model secara keseluruhan dari 85% menjadi 88%, terutama dalam meningkatkan *precision* untuk kelas minoritas seperti *Offensive*, meskipun *recall* mengalami penurunan. Pada kelas *Neither*, *F1-score* mengalami peningkatan, menunjukkan perbaikan dalam keseimbangan antara *precision* dan *recall*. Kinerja pada kelas mayoritas *Hate* tetap stabil, menunjukkan bahwa SMOTE tidak mengganggu kinerja model pada kelas yang sudah dominan. Secara keseluruhan, penerapan SMOTE memberikan manfaat yang signifikan dalam menangani ketidakseimbangan kelas, menghasilkan model klasifikasi yang lebih akurat.

Pada penelitian lain yang dilakukan oleh (Rendragraha dkk., 2021) yang juga membahas deteksi bahasa dengan mengeksplorasi penggunaan model *Transformer*, khususnya BERT, untuk mendeteksi bahasa kasar dalam komentar berita *online* berbahasa Indonesia, yang mana penggunaan internet dan interaksi online semakin meningkat, termasuk munculnya komentar dengan bahasa kasar yang merugikan. Penelitian ini membandingkan kinerja model BERT yang dilatih dari awal (*scratch*) dengan model *pre-trained* BERT Multilingual, dengan hasil menunjukkan bahwa BERT *Multilingual* mencapai performa lebih baik dengan *Macro Average F1 Score* sebesar 54% dibandingkan dengan 50% pada model *scratch*. Penelitian ini juga mengidentifikasi tantangan dalam deteksi bahasa kasar, seperti variasi penulisan kata kasar dan penggunaan bahasa daerah, serta memberikan saran untuk pengembangan model lebih lanjut, termasuk peningkatan dataset dan teknik pelatihan.

Penelitian lain yang dilakukan oleh (Putri dkk., 2024) menganalisis sentimen berbasis aspek dan deteksi emosi dari ulasan pelanggan *coffee shop* di Palangka Raya menggunakan *deep learning*, khususnya model BERT dan LSTM. Data ulasan dari 103 *coffee shop* dikumpulkan dari Google Maps, diproses melalui serangkaian langkah *preprocessing*, dan dianotasi untuk aspek (*Food and Drinks, Ambience, Place, Service*), sentimen (positif, negatif, netral), dan emosi (*joy, anticipation, trust, dll.*). Hasil penelitian menunjukkan bahwa aspek "service" paling sering dibahas, sentimen positif mendominasi, dan emosi yang paling sering muncul adalah *joy, anticipation, dan trust*. Model BERT mencapai akurasi 99% dalam klasifikasi sentimen, menunjukkan efektivitas *deep learning* dalam memahami persepsi pelanggan dan memberikan wawasan bagi pengelola *coffee shop* untuk meningkatkan kualitas layanan.

Penelitian lain tentang analisis sentimen yang dilakukan oleh (Adinda dkk., 2025) menganalisis sentimen masyarakat Surabaya terhadap layanan kesehatan menggunakan model RoBERTa, dengan fokus pada aspek BPJS, antrian, tenaga kesehatan, dan fasilitas kesehatan. Data dikumpulkan dari media sosial dan berita, kemudian diproses melalui pra-pemrosesan dan tokenisasi. Hasilnya menunjukkan sentimen netral mendominasi pada isu antrian, sentimen positif tertinggi pada fasilitas kesehatan, dan sentimen negatif tertinggi pada tenaga kesehatan, dengan akurasi model mencapai 87.2%, memberikan wawasan penting bagi peningkatan layanan kesehatan di Surabaya.

Terakhir penelitian oleh (Chandradev dkk., 2022) berhasil mengimplementasikan model *deep learning* BERT, khususnya SmallBERT, untuk menganalisis sentimen pada ulasan hotel di Indonesia. Model yang telah dilatih dengan dataset besar mampu mengklasifikasikan sentimen ulasan dengan akurasi yang tinggi. Hasil penelitian menunjukkan bahwa mayoritas ulasan hotel memiliki sentimen positif. Penelitian ini memberikan kontribusi penting dalam memahami preferensi pelanggan hotel dan dapat membantu industri perhotelan dalam meningkatkan kualitas layanannya. Metode *fine-tuning* yang diterapkan pada model SmallBERT terbukti efektif dalam meningkatkan kinerja model dalam menganalisis sentimen teks dalam bahasa Indonesia.

Secara keseluruhan, dapat diambil kesimpulan bahwa penelitian-penelitian ini menunjukkan betapa pentingnya model *deep learning* dalam tugas pemrosesan bahasa alami. Model *deep learning* dan *transformer* seperti BERT, RoBERTa dan DistilBERT ini mampu memahami konteks dan makna teks dengan lebih baik. Oleh karena itu, penelitian ini diharapkan dapat memberikan kontribusi lebih lanjut dalam pengembangan model bahasa alami untuk deteksi sarkasme, khususnya data teks bahasa Inggris dari Twitter.

## 2.2 Dasar Teori

### 2.2.1 *Natural Language Processing (NLP)*

*Natural Language Processing (NLP)* adalah cabang kecerdasan buatan yang fokus pada interaksi manusia-mesin melalui bahasa alami. NLP memungkinkan komputer untuk memahami, menghasilkan, dan merespons bahasa manusia, sehingga menjadi dasar berbagai aplikasi seperti asisten virtual, chatbot, dan penerjemah bahasa. Kemajuan pesat NLP, terutama didorong oleh model seperti BERT dan GPT, memungkinkan pemahaman konteks bahasa yang lebih baik. Namun, NLP masih menghadapi tantangan seperti ambiguitas bahasa, bias data, dan isu etika. Pengembangan NLP yang etis dan inklusif sangat penting untuk masa depan teknologi ini, yang diharapkan dapat menjembatani interaksi manusia-mesin dan membuka peluang baru di berbagai sektor. (Horacek, 2024)

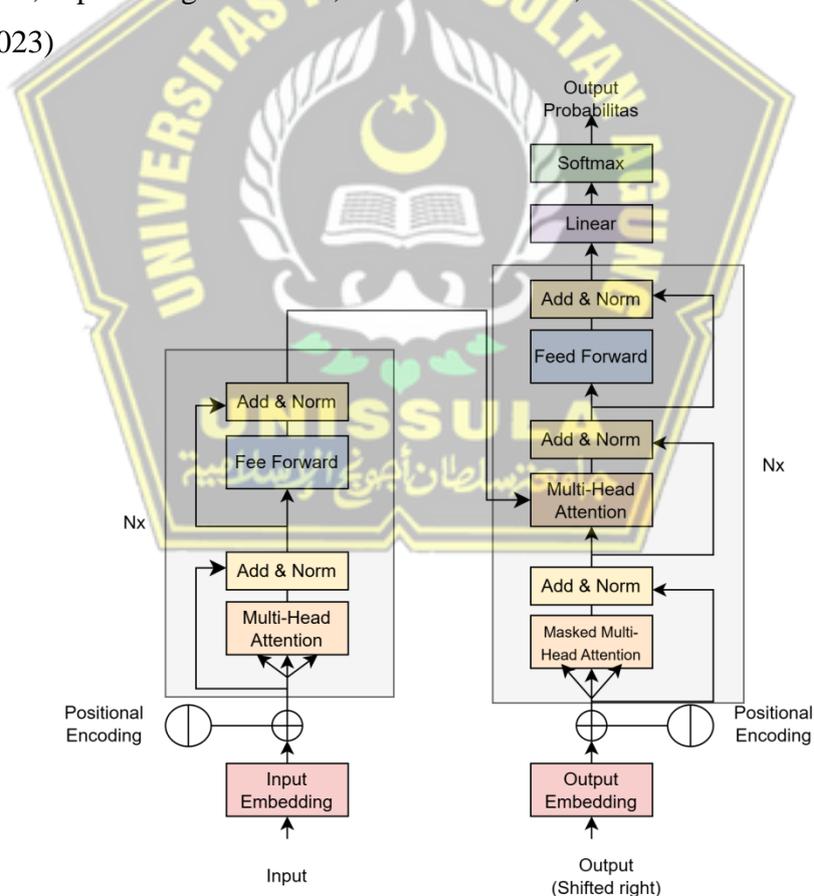
### 2.2.2 *Multi-Class Classification*

Klasifikasi multi-kelas adalah tugas penting dalam pembelajaran mesin yang melibatkan pengkategorian data ke dalam salah satu dari tiga kelas atau lebih. Dengan kata lain teks atau data hanya dapat dikategorikan ke dalam satu kelas dari tiga atau lebih kelas yang memiliki keistimewaan masing-masing, dan masalah ini sering muncul dalam berbagai aplikasi seperti pengenalan gambar, analisis teks, dan bioinformatika. Tantangan utama dalam klasifikasi multi-kelas terletak pada peningkatan kompleksitas seiring bertambahnya jumlah kelas. Kompleksitas ini sebagian besar disebabkan oleh heterogenitas batas keputusan,

yaitu perbedaan dalam kompleksitas dan karakteristik batas yang memisahkan kelas-kelas yang berbeda.(Moral dkk., 2022)

### 2.2.3 *Trasformer*

*Trasformer* adalah inovasi besar dalam bidang *Natural Language Processing* (NLP). Dengan mengganti mekanisme rekursif dengan mekanisme *attention*, mekanisme ini memungkinkan model untuk memperhitungkan hubungan antar kata secara bersamaan bukan secara berurutan. Sehingga model ini mampu mengatasi keterbatasan model-model sebelumnya. Hasilnya, *trasnformer* memungkinkan pelatihan yang lebih cepat dan menghasilkan model lebih akurat, terutama untuk tugas-tugas yang melibatkan data teks yang kompleks, seperti ringkasan teks, analisis sentimen, deteksi sentimen.(Mohiuddin dkk., 2023)



Gambar 2. 1 arsitektur tranformer(Vaswani dkk., 2017)

*Trasformer* menggunakan struktur *encoder-decoder* namun dengan inovasi seperti terlihat pada gambar. Dimana *encoder* mengubah input menjadi

representasi numerik, lalu *decoder* menggunakan representasi ini untuk menghasilkan kalimat baru (Subakan dkk., 2021) . Arsitektu *encoder-decoder* yang masing-masing terdiri dari enam lapisan identik yang ditumpuk. Pada bagian *encoder*, setiap lapisan memiliki dua sub-lapisan utama: mekanisme *perhatian mandiri multi-head* yang memungkinkan model untuk fokus pada berbagai bagian input secara bersamaan dan *jaringan feed-forward* yang sepenuhnya terhubung. Setiap sub-lapisan dilengkapi dengan koneksi sisa dan normalisasi lapisan. Bagian *decoder* juga memiliki enam lapisan identik dengan dua sub-lapisan yang sama dengan encoder, ditambah sub-lapisan *perhatian multi-head* yang beroperasi pada output encoder. Sub-lapisan perhatian mandiri pada decoder dimodifikasi dengan *masking* untuk memastikan prediksi hanya bergantung pada output sebelumnya, menghasilkan output secara auto-regresif. Inti dari mekanisme ini adalah fungsi *perhatian* (attention) yang memetakan *query* dan sekumpulan pasangan *key-value* menjadi *output* berupa jumlah berbobot, di mana *query*, *key*, *value*, dan *output* semuanya berbentuk vektor. (Vaswani dkk., 2017)

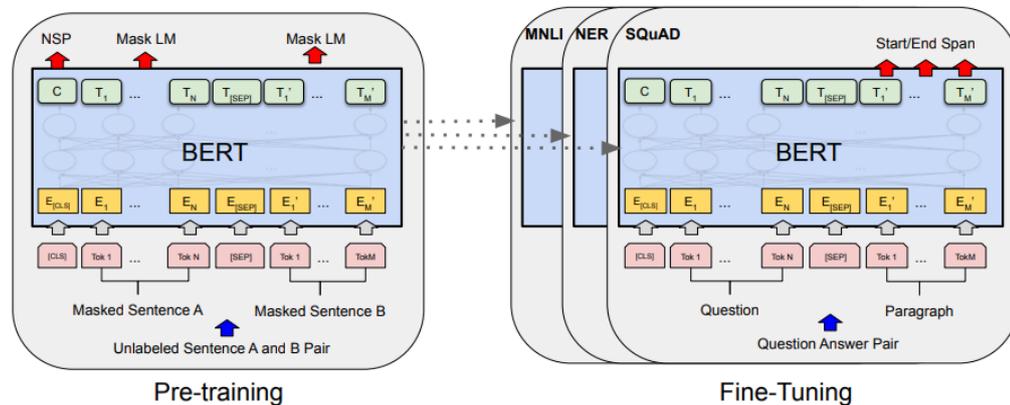
Model *Transformer* menggunakan mekanisme *Scaled Dot-Product Attention*, di mana *output* dihitung sebagai jumlah berbobot dari *value* berdasarkan *query* dan *key*, diskalakan dengan akar kuadrat dimensi *key* ( $\sqrt{d_k}$ ) untuk mencegah gradien yang terlalu kecil. Perhatian ini dihitung secara paralel untuk sekumpulan *query*, *key*, dan *value* yang dipaketkan dalam matriks. Model ini menggunakan *multi-head attention*, di mana *query*, *key*, dan *value* diproyeksikan secara linear ke dimensi yang lebih rendah sebelum dilakukan perhitungan *attention*, hasilnya digabungkan dan diproyeksikan kembali. Multi-head attention memungkinkan model untuk memperhatikan informasi dari berbagai subruang representasi secara bersamaan. Dalam encoder-decoder, *query* berasal dari lapisan decoder sebelumnya, dan *key* serta *value* berasal dari output encoder. Encoder dan decoder keduanya menggunakan *self-attention*, di mana *query*, *key*, dan *value* berasal dari lapisan yang sama. Pada decoder, *masking* diterapkan pada *self-attention* untuk mencegah aliran informasi ke kiri. Selain sub-lapisan *attention*, setiap lapisan encoder dan decoder juga mengandung jaringan *feed-forward* yang sepenuhnya terhubung. Model ini menggunakan

*embedding* yang dipelajari untuk mengubah token input dan output menjadi vektor, dan *bobot embedding* dibagikan dengan transformasi linear *pra-softmax*. Untuk memanfaatkan urutan sekuens, *positional encoding* sinusoidal ditambahkan ke *embedding* input, memungkinkan model untuk mempelajari hubungan antar posisi.

#### **2.2.4 Bidirectional Encoder Representations from Transformers (BERT)**

Model BERT bekerja dengan cara mempelajari hubungan antar kata dalam suatu kalimat secara kontekstual menggunakan arsitektur *Transformer*. Kalimat yang akan diproses diubah menjadi urutan *token*, di mana setiap *token* mewakili satu kata. Urutan *token* ini kemudian dimasukkan ke dalam *encoder BERT*. *Encoder BERT* terdiri dari beberapa lapisan yang masing-masing berisi mekanisme *multi-head attention* dan *feed-forward neural network*. *Multi-head attention* memungkinkan model untuk memperhatikan kata-kata lain dalam kalimat saat memproses suatu kata tertentu, sehingga memahami konteks kata secara keseluruhan. *Output* dari *encoder* kemudian diproses oleh lapisan *softmax* untuk menghasilkan probabilitas prediksi. Fungsi aktivasi *softmax* ini memungkinkan model untuk memilih kata yang paling mungkin sesuai dengan konteksnya. Secara keseluruhan, BERT dirancang untuk memahami bahasa alami dengan sangat baik, sehingga dapat digunakan untuk berbagai tugas pemrosesan bahasa alami, seperti klasifikasi teks, pertanyaan-jawaban, dan generasi teks.

Model BERT (*Bidirectional Encoder Representations from Transformers*) memiliki pendekatan yang unik dalam mempelajari bahasa alami. Proses pembelajarannya terbagi menjadi dua tahap utama: *Pre-Training* dan *Fine-tuning*. (Devlin dkk., 2019)



Gambar 2. 2 Pre-training dan Fine-Tuning BERT

Prosedur *pre-training* dan *fine-tuning* BERT menggunakan arsitektur yang sama (kecuali lapisan output) dan parameter model pra-latih yang sama untuk inialisasi berbagai tugas hilir. Model BERT menggunakan dua metode utama dalam melakukan *pre-training* atau pelatihan awal. Metode yang pertama disebut *Masked Language Model* (MLM) atau Model Bahasa Bertopeng. Sederhananya, model ini akan menyembunyikan beberapa kata secara acak dalam sebuah kalimat, lalu tugasnya adalah menebak kata-kata yang disembunyikan tersebut. Tujuannya agar model dapat memahami konteks kalimat secara mendalam. Metode yang kedua adalah *Next Sentence Prediction* (NSP) atau Prediksi Kalimat Selanjutnya. Dalam metode ini, model akan diberikan dua kalimat dan tugasnya adalah menebak apakah kalimat kedua merupakan lanjutan dari kalimat pertama. Tujuannya agar model dapat memahami hubungan antar kalimat. Setelah itu masuk ketahap *fine-tuning*, selama *fine-tuning* semua parameter disesuaikan. Simbol khusus [CLS] ditambahkan di awal setiap contoh input, dan [SEP] adalah token pemisah (misalnya, memisahkan pertanyaan/jawaban). (Devlin dkk., 2019)

### 2.2.5 Robustly Optimized BERT Pretraining Approach (RoBERTa)

RoBERTa adalah model bahasa bentuk modifikasi, pengoptimalan, dan evaluasi dari model BERT yang sebelumnya memiliki beberapa keterbatasan yang diakibatkan oleh *undertraining* dan pilihan *hyperparameter* yang kurang optimal. Beberapa perubahan utama dalam model RoBERTa adalah karena peningkatan

performa yang tidak signifikan pada tugas *downstream* maka dilakukan penghapusan tugas *Next Sentence Prediction* (NSP) , yang awalnya digunakan untuk memahami hubungan antar bagian teks pada model BERT. Selain itu untuk menggantikan pendekatan *static masking* pada BERT yang mengganti data secara permanen, pada RoBERTa menggunakan metode *dynamic masking* yang mana dapat mengubah secara acak pola mask pada tugas *Masked Language Modeling* (MLM) secara *real time* tiap kali data dilatih.

Pada model BERT dataset yang dilatih berukuran 16 GB, sedangkan pada Model RoBERTa dataset yang dilatih memiliki ukuran yang lebih besar dan lebih beragam dengan peningkatan data sebesar 160 GB. Selain itu RoBERTa juga menggunakan ukuran Batch yang lebih besar dengan pelatihan yang lebih lama untuk memaksimalkan performa. Dengan pelatihan yang lebih panjang dan optimalisasi hyperparameter, model RoBERTa dapat mencapai hasil yang paling baik pada beberapa *benchmark* seperti *GLUE*, *RACE*, dan *SQUAD*. Maka dari itu dengan arsitektur yang sama peningkatan performa model dapat dicapai dengan mengoptimalkan proses *pretraining*. (Liu dkk., 2019)

### 2.2.6 Metriks Evaluasi

Dalam *machine learning*, prediksi hasil dari data yang ada terbagi menjadi klasifikasi (kategori) dan regresi (angka), di mana klasifikasi, baik biner (dua kategori) atau multi-kelas (lebih dari dua kategori), menggunakan algoritma untuk menghasilkan probabilitas setiap kategori dan memilih yang paling mungkin, dengan evaluasi model menggunakan indikator kinerja untuk membandingkan keakuratan prediksi. Metrik evaluasi, terutama yang berasal dari *confusion matrix*, adalah alat penting untuk menilai dan meningkatkan kinerja model klasifikasi multi-kelas.

#### 2.2.6.1 Confusion matrix

*Confusion matrix* adalah tabel silang yang membandingkan klasifikasi aktual dengan prediksi model, di mana kolom menunjukkan prediksi dan baris menunjukkan kebenaran, dengan kelas-kelas diurutkan sama di kedua sumbu,

sehingga elemen yang diklasifikasikan dengan benar terletak di diagonal utama, menunjukkan kesesuaian antara model dan kebenaran aktual.(Grandini dkk., 2020)

### 2.2.6.2 Precision dan recall

Presisi adalah metrik yang mengukur seberapa tepat model klasifikasi dalam memprediksi kelas positif. Ini dihitung sebagai rasio antara jumlah prediksi positif yang benar (*True Positive*) dan jumlah total prediksi positif (*True Positive* + *False Positive*). Dengan kata lain, presisi memberi tahu kita seberapa yakin kita dapat mempercayai prediksi positif yang diberikan oleh model. Jika presisi tinggi, berarti model cenderung membuat prediksi positif yang benar, dan kesalahan prediksi positif (*False Positive*) relatif sedikit.

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

*Recall*, atau sensitivitas, adalah metrik yang mengukur kemampuan model klasifikasi untuk menemukan semua elemen positif yang sebenarnya dalam data. Ini dihitung sebagai rasio antara jumlah prediksi positif yang benar (*True Positive*) dan jumlah total elemen positif aktual (*True Positive* + *False Negative*). Dengan kata lain, recall memberi tahu kita seberapa baik model dapat mengidentifikasi semua kasus positif yang ada, meminimalkan risiko melewatkan kasus positif (*False Negative*). Semakin tinggi nilai *recall*, semakin baik model dalam menemukan semua elemen positif.

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

### 2.2.6.3 Accuracy

Akurasi, metrik populer dalam klasifikasi multi-kelas yang dihitung langsung dari *confusion matrix*, mengukur proporsi prediksi yang benar dari total prediksi, dengan mempertimbangkan jumlah *True Positive* dan *True Negative* dibagi jumlah semua elemen matriks; secara sederhana, akurasi adalah probabilitas prediksi model yang benar untuk unit acak, di mana setiap unit

memiliki bobot sama, sehingga akurasi cocok ketika fokus pada prediksi individu yang benar tanpa memperhatikan distribusi kelas, tetapi akurasi dapat menyembunyikan kesalahan klasifikasi pada kelas dengan sedikit unit dalam dataset yang tidak seimbang, meskipun intuitif dan mudah dipahami, dengan nilai antara 0 dan 1, dan selisihnya dari 1 disebut tingkat kesalahan klasifikasi.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

#### 2.2.6.4 F1- Score

F1-Score, yang juga dihitung dari *confusion matrix*, menggabungkan presisi dan recall menggunakan rata-rata harmonik, memberikan nilai antara 0 dan 1, di mana 1 adalah nilai terbaik; ini dapat diartikan sebagai rata-rata tertimbang presisi dan recall, memberikan bobot yang sama pada keduanya, dan berguna untuk menemukan keseimbangan optimal antara keduanya, baik dalam klasifikasi biner maupun multi-kelas; rata-rata harmonik memberikan bobot lebih besar pada kelas yang lebih kecil dan model dengan nilai presisi dan *recall* yang serupa, serta sangat sensitif terhadap nilai presisi atau *recall* yang mendekati 0, karena rata-rata harmonik memberi bobot lebih besar pada nilai yang lebih rendah.

$$F1 - Score = 2 \times \left( \frac{precision \cdot recall}{precision + recall} \right) \quad (4)$$

## BAB III METODOLOGI PENELITIAN

### 3.1 Metode Penelitian

Pada bab ini akan membahas tentang gambaran menyeluruh metodologi penelitian yang akan digunakan, termasuk desain penelitian, pengumpulan data, analisis, dan interpretasi hasilnya. Pada penelitian ini metode penelitian yang digunakan adalah metode *supervised learning* dan akan menggunakan model RoBERTa. Keluaran dari sistem ini adalah berupa hasil klasifikasi untuk setiap kalimat atau tweet yang diuji, setiap label akan disertai dengan skor probabilitas yang menunjukkan seberapa yakin model terhadap prediksi tersebut.



Gambar 3. 1 *flowchart* metode penelitian

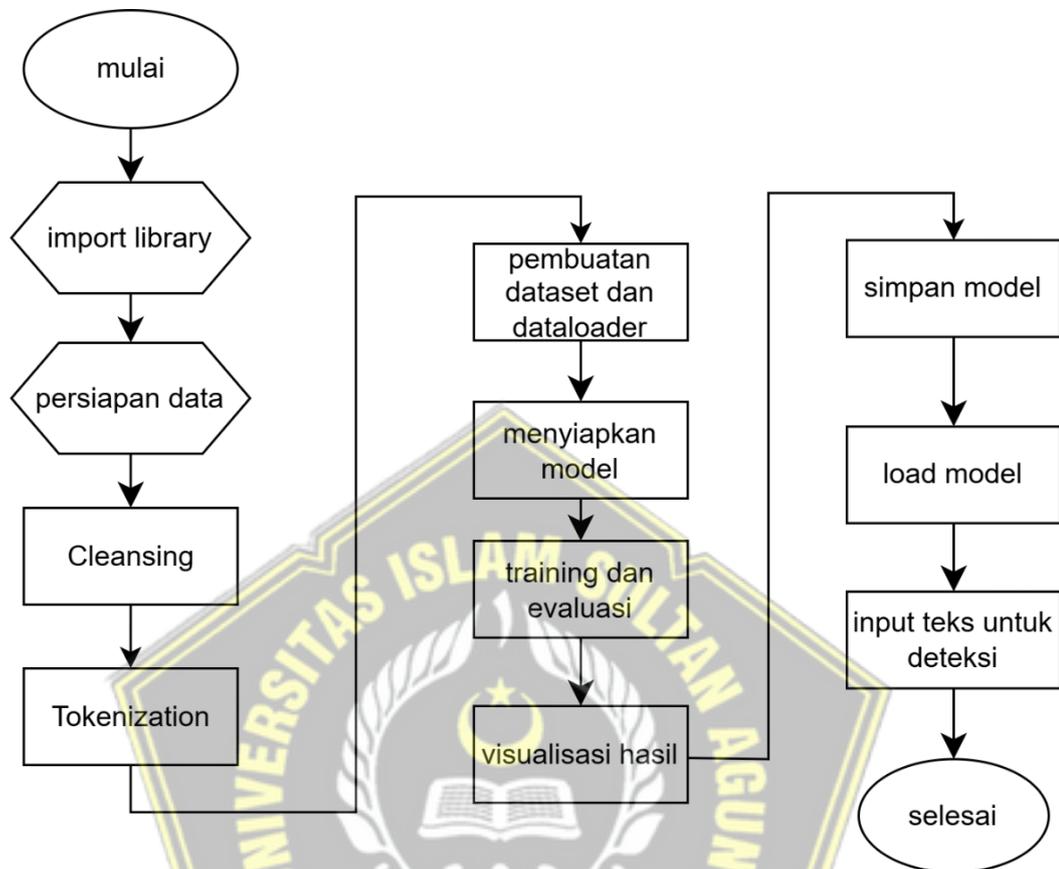
#### 3.1.1 Studi Literatur

Pada tahap studi literatur dilakukan untuk mengumpulkan informasi dan teori yang mendukung. Beberapa sumber yang digunakan meliputi penelitian terdahulu terkait deteksi sarkasme dengan model *deep learning* (BERT, RoBERTa, atau arsitektur lainnya), konsep dasar NLP, dan dataset yang tersedia di berbagai platform. Studi literatur bertujuan untuk memahami tantangan, Teknik, serta perkembangan terkini dalam deteksi sarkasme, sehingga dapat digunakan sebagai dasar dalam perancangan model.

#### 3.1.2 Pengumpulan Data

Dalam penelitian ini, pengumpulan data dilakukan melalui pemanfaatan sumber utama dataset sarkasme publik yang telah dilabeli dan tersedia di Kaggle, yang memberikan fondasi data yang terstruktur dan relevan, data dari Kaggle dicari yang sesuai dengan fokus pada topik dengan lingkup penelitian, memperluas variasi dan kedalaman data yang dianalisis.

### 3.1.3 Pemodelan Sistem



Gambar 3. 2 Pemodelan Sistem

Pada tahap pemodelan terdiri dari beberapa proses yaitu *import library*, persiapan data, *preprocessing*, pelatihan model, dan evaluasi.

1. Mulai: Tahap ini menandai awal dari proses keseluruhan. Ini adalah titik di mana semua tahapan selanjutnya akan dimulai.
2. Import Library: Pada tahap ini, semua pustaka (*library*) yang diperlukan untuk menjalankan program diimpor. Pustaka ini biasanya berisi fungsi dan alat yang akan digunakan dalam tahapan selanjutnya, seperti untuk pemrosesan data, pembuatan model, dan visualisasi. Pada penelitian ini *library* yang di import diantaranya adalah *library torch* untuk membangun arsitektur, *library numpy* pustaka untuk operasi numerik, *library pandas* untuk manipulasi dan analisis data, *library re* untuk pencocokan pola teks, *library string* modul untuk operasi string, *library nltk* pustaka untuk pemrosesan

bahasa alami (NLP), library matplotlib dan seaborn untuk visualisasi data, library transformers menyediakan model-model transformer yang sudah dilatih sebelumnya, sklearn.metrics modul untuk metrik evaluasi model.

3. **Persiapan Data:** Tahap ini melibatkan pengumpulan dan penataan data yang akan digunakan untuk pelatihan dan pengujian model. Pada tahap ini membaca data dari file CSV menggunakan pandas, kemudian menghapus kolom dan baris yang tidak diperlukan seperti baris dan kolom yang mengandung unnamed, NAN, dan class yang tidak dipakai dalam penelitian.
4. **Cleansing:** Pada tahap ini, data yang telah dikumpulkan dibersihkan dari noise atau informasi yang tidak relevan. Menghapus karakter khusus dan emoji. Menangani kontraksi (misalnya, "can't" menjadi "cannot"). Menghapus tag HTML dan URL. Menghapus tanda baca dan mengubah teks menjadi huruf kecil. Menghapus stopwords (kata-kata umum).
5. **Tokenization:** Pada tahap ini, teks dipecah menjadi unit-unit yang lebih kecil yang disebut token. Token bisa berupa kata, frasa, atau karakter. Tokenisasi diperlukan agar model dapat memahami dan memproses teks.
6. **Pembuatan Dataset dan Dataloader:** Dataset dibuat dari data yang telah di tokenisasi. Dataloader kemudian digunakan untuk memuat data ke dalam model dalam batch selama pelatihan dan pengujian.
7. **Menyiapkan Model:** Pada tahap ini, arsitektur model dipilih dan parameter-parameternya diinisialisasi. Memuat model RoBERTa yang sudah dilatih sebelumnya dari hugging face transformers.
8. **Training dan Evaluasi:** Model dilatih menggunakan data pelatihan. Selama pelatihan, model belajar untuk menyesuaikan parameternya agar dapat membuat prediksi yang akurat. Setelah pelatihan selesai, model dievaluasi menggunakan data pengujian untuk mengukur kinerjanya.
9. **Visualisasi Hasil:** Pada tahap ini, hasil evaluasi divisualisasikan menggunakan grafik atau diagram. Membuat plot loss pelatihan dan validasi serta membuat confusion matrix untuk visualisasi kinerja model. Visualisasi ini membantu dalam memahami kinerja model dan mengidentifikasi area yang perlu ditingkatkan.

10. Simpan Model: Model yang telah dilatih dan dievaluasi disimpan ke dalam file. Ini memungkinkan model untuk digunakan kembali di masa mendatang tanpa perlu melatihnya dari awal.
11. Load Model: Model yang telah disimpan dimuat kembali ke dalam memori. Ini diperlukan untuk menggunakan model dalam tugas inferensi, seperti deteksi teks.
12. Input Teks untuk Deteksi: Teks baru dimasukkan ke dalam model yang telah dimuat. Model kemudian memproses teks dan menghasilkan prediksi, seperti label atau kategori.
13. Selesai: Tahap ini menandai akhir dari proses keseluruhan. Semua tahapan telah selesai, dan model telah digunakan untuk melakukan tugas yang diinginkan.

### 3.2 Analisis Sistem

Analisis sistem dilakukan untuk mengevaluasi performa model yang telah dibangun dalam mendeteksi sarkasme. Evaluasi ini mencakup pengukuran performa model menggunakan metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *F1-score*. Selain itu, hasil klasifikasi dianalisis lebih dalam dengan menggunakan confusion matrix untuk melihat distribusi prediksi model antara label *sarcasm*, *irony*, *figurative* dan *regular*. Visualisasi hasil *confusion matrix* dibuat menggunakan *heatmap* dengan pustakan visual python *seaborn* dan *matplotlib*.

### 3.3 Analisis Kebutuhan

Pada tahap analisis kebutuhan, penulis mengidentifikasi perangkat apa saja yang dibutuhkan selama proses penelitian agar dalam pembuatan model deteksi sarkasme dapat berjalan dengan baik. Beberapa *tools software* maupun *hardware* yang digunakan yaitu :

1. Python 3.11.0

Python 3.11.0 adalah versi spesifik dari bahasa pemrograman Python yang dirilis dengan fokus utama pada peningkatan performa, memberikan

kecepatan eksekusi kode yang lebih tinggi dibandingkan versi sebelumnya. Selain itu, versi ini juga memperkenalkan fitur-fitur baru yang meningkatkan efisiensi dan kemudahan penggunaan bagi para pengembang, serta penyempurnaan dalam penanganan kesalahan dan tipe data. Dengan peningkatan kinerja dan fitur-fitur baru tersebut, Python 3.11.0 menjadi pilihan yang sangat baik untuk pengembangan berbagai aplikasi, mulai dari pengembangan web, analisis data, kecerdasan buatan, hingga otomatisasi tugas.

## 2. *Library PyTorch*

*PyTorch* adalah kerangka kerja pembelajaran mesin sumber terbuka yang dikembangkan oleh Facebook's AI Research lab. *Library* ini sangat populer di kalangan peneliti dan pengembang karena fleksibilitas dan kemudahan penggunaannya. *PyTorch* menyediakan alat untuk membangun dan melatih jaringan saraf tiruan, dengan dukungan untuk komputasi GPU yang mempercepat proses pelatihan. Kegunaan utamanya adalah dalam pengembangan aplikasi kecerdasan buatan, seperti pengenalan gambar, pemrosesan bahasa alami, dan pembelajaran mendalam.

## 3. *Library Numpy*

*NumPy* adalah *library* dasar untuk komputasi numerik dalam Python. *Library* ini menyediakan dukungan untuk array multidimensi dan berbagai fungsi matematika yang efisien. *NumPy* sering digunakan dalam analisis data, komputasi ilmiah, dan pembelajaran mesin, karena kemampuannya untuk memanipulasi data numerik dengan cepat dan mudah.

## 4. *Library Pandas*

*Pandas* adalah *library* yang kuat untuk analisis dan manipulasi data. *Library* ini menyediakan struktur data yang fleksibel, seperti *DataFrame*, yang memungkinkan pengguna untuk bekerja dengan data tabular dengan mudah. *Pandas* juga menyediakan berbagai fungsi untuk membersihkan, mengubah, dan menganalisis data, sehingga sangat berguna dalam eksplorasi data dan persiapan data untuk pembelajaran mesin.

5. *Library re (regular expression)*

*Library re* menyediakan dukungan untuk ekspresi reguler dalam Python. Ekspresi reguler adalah urutan karakter yang mendefinisikan pola pencarian. *Library* ini digunakan untuk mencari, mencocokkan, dan memanipulasi teks berdasarkan pola tertentu. Kegunaan utamanya adalah dalam pemrosesan teks, seperti ekstraksi informasi, validasi data, dan penggantian teks.

6. *Library String*

*Library string* pada python menyediakan konstanta dan kelas untuk memanipulasi string. *Library* ini menyediakan berbagai alat untuk operasi string umum, seperti mengubah huruf besar/kecil, menghapus spasi, dan mencari *substring*.

7. *Library NLTK*

NLTK adalah *library* yang populer untuk pemrosesan bahasa alami (NLP) dalam Python. *Library* ini menyediakan berbagai alat dan dataset untuk tugas-tugas NLP, seperti tokenisasi, stemming, lemmatisasi, analisis sentimen, dan klasifikasi teks. NLTK sangat berguna dalam pengembangan aplikasi yang berhubungan dengan pemahaman bahasa alami. (Rossum Guido van & the Python development team, 2017)

8. *Library Matplotlib*

Matplotlib adalah *library* untuk membuat visualisasi data statis, animasi, dan interaktif di Python. pyplot adalah kumpulan fungsi dalam Matplotlib yang menyediakan antarmuka seperti MATLAB untuk membuat *plot*. *Library* ini sering digunakan untuk membuat grafik, diagram, dan visualisasi data lainnya.

9. *Library Seaborn*

*Seaborn* adalah *library* visualisasi data yang dibangun di atas Matplotlib. *Seaborn* menyediakan antarmuka tingkat tinggi untuk membuat visualisasi statistik yang menarik dan informatif. *Library* ini sangat berguna dalam eksplorasi data dan analisis statistik, karena kemampuannya untuk menghasilkan visualisasi yang kompleks dengan mudah.

#### 10. *Library Scikit-learn*

*Scikit-learn* adalah *library* pembelajaran mesin yang menyediakan berbagai algoritma untuk klasifikasi, regresi, pengelompokan, dan reduksi dimensi. *Library* ini juga menyediakan alat untuk evaluasi model, pemilihan fitur, dan pra-pemrosesan data. *Scikit-learn* sangat populer di kalangan praktisi pembelajaran mesin karena kemudahan penggunaan dan kelengkapan fiturnya.

#### 11. *Visual Studio Code (VS Code)*

*VS Code* adalah *editor* kode sumber yang dikembangkan oleh Microsoft. *VS code* dipilih karena ringan, cepat, dan memiliki banyak fitur yang dapat disesuaikan. *VS Code* mendukung berbagai bahasa pemrograman dan menyediakan fitur-fitur seperti *debugging*, kontrol versi Git, dan ekstensi yang dapat memperluas fungsionalitasnya. Kegunaan utamanya adalah untuk menulis, mengedit, dan mengelola kode sumber secara efisien

#### 12. *Google Colaboratory (Colab)*

*Google Colab* adalah lingkungan *notebook Jupyter* yang berjalan di *cloud*. *Tools* ini memungkinkan pengguna untuk menulis dan menjalankan kode Python melalui *browser*, dengan akses gratis ke sumber daya komputasi seperti GPU dan TPU. *Colab* sangat berguna untuk pembelajaran mesin, analisis data, dan pendidikan, karena memudahkan kolaborasi dan berbagi kode. Kegunaan utamanya adalah untuk eksperimen dan pengembangan model pembelajaran mesin, serta untuk menjalankan analisis data interaktif. Sehingga sangat cocok untuk digunakan dalam penggunaan model berat seperti transformers.

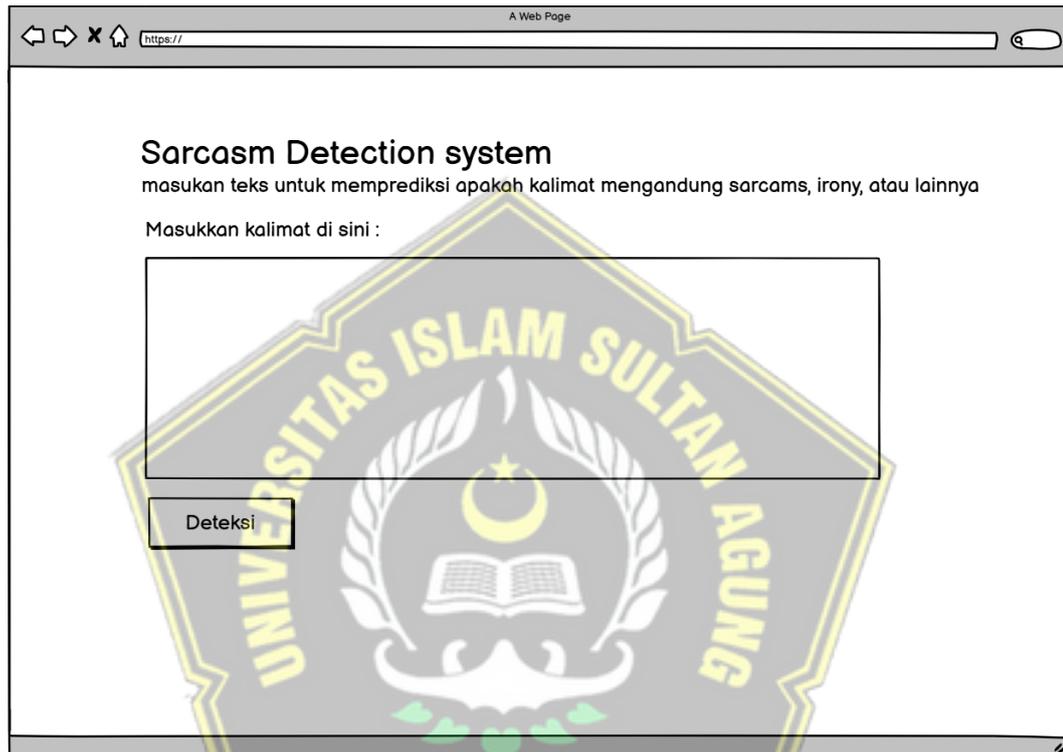
#### 13. *Streamlit*

*Streamlit* adalah pustaka Python sumber terbuka yang memudahkan pembuatan aplikasi *web* interaktif untuk pembelajaran mesin dan ilmu data. *Tools* ini memungkinkan pengembang untuk mengubah *skrip* Python menjadi aplikasi web yang dapat dibagikan dengan cepat. *Streamlit* sangat berguna untuk membuat dasbor, visualisasi data, dan aplikasi demo model pembelajaran mesin. Kegunaan utamanya adalah untuk memvisualisasikan

data dan model, serta untuk membuat aplikasi interaktif yang mudah digunakan.

### 3.4 Desain *User Interface*

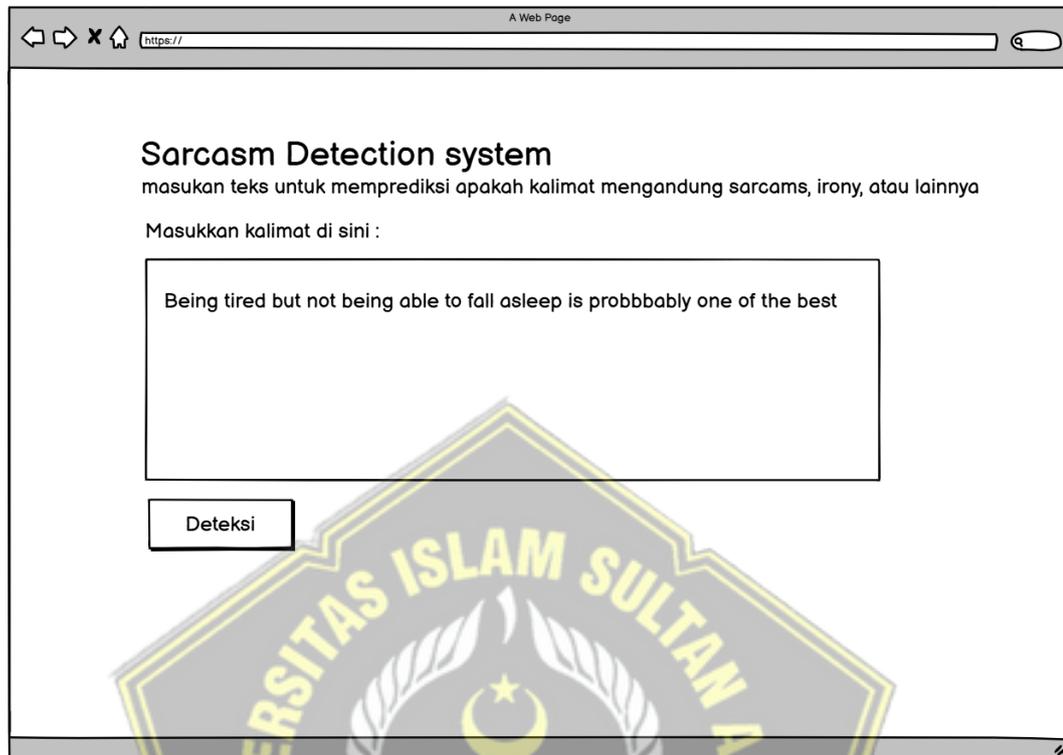
#### 3.4.1 Tampilan awal



Gambar 3. 3 Desain Tampilan Awal *User Interface*

Gambar 3.3 merupakan tampilan awal *user interface*. Pada tampilan awal ini terdapat judul dan *text area* dimana *user* dapat memasukkan kalimat atau teks yang ingin diidentifikasi. Kemudian ada tombol deteksi yang apabila diklik akan menampilkan hasil deteksi dari kalimat yang telah di inputkan.

### 3.4.2 Memasukkan kalimat



A Web Page

https://

## Sarcasm Detection system

masukan teks untuk memprediksi apakah kalimat mengandung sarcams, irony, atau lainnya

Masukkan kalimat di sini :

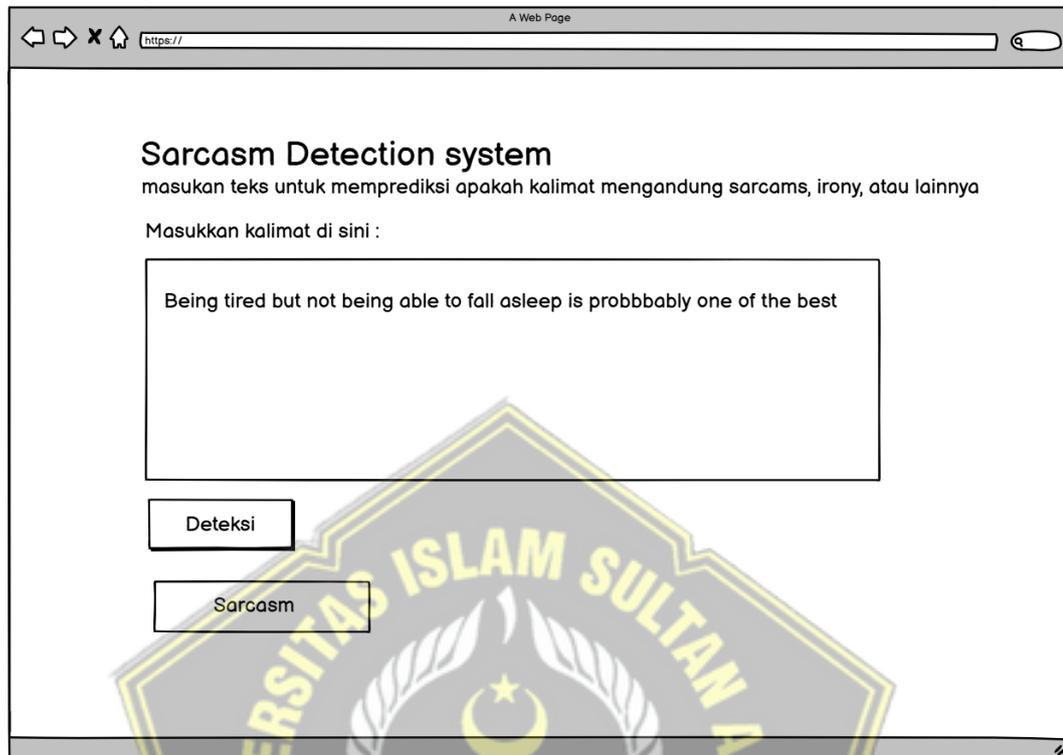
Being tired but not being able to fall asleep is probbably one of the best

Deteksi

Gambar 3. 4 Desain *Input* kalimat

Gambar 3.4 merupakan tampilan setelah memasukkan kalimat yang ingin di deteksi apakah kalimat mengandung *sarcasm*, *irony*, *fugurative*, atau *regular*.

### 3.4.3 Hasil deteksi



A Web Page

https://

## Sarcasm Detection system

masukan teks untuk memprediksi apakah kalimat mengandung sarcams, irony, atau lainnya

Masukkan kalimat di sini :

Being tired but not being able to fall asleep is probbably one of the best

Deteksi

Sarcasm

Gambar 3. 5 Desain Hasil Deteksi

Gambar 3.5 menunjukkan desain untuk tampilan hasil deteksi dari kalimat yang telah di inputkan setelah mengklik tombol deteksi hasil akan ditampilkan.

## BAB IV

### HASIL DAN ANALISIS PENELITIAN

#### 4.1 Persiapan data

```
0 I love waking up to the sound of Doris Kearns ... figurative NaN
1 Just saw some caravans caravanning. #irony figurative NaN
2 New wellness program launched at work Tuesday... figurative NaN
3 When you get curious about a company a LinkedI... figurative NaN
4 @WashTimes: #Leesburg clothing store clerk inj... figurative NaN

Unnamed: 3 Unnamed: 4 Unnamed: 5 Unnamed: 6 Unnamed: 7 Unnamed: 8 \
0 NaN NaN NaN NaN NaN NaN
1 NaN NaN NaN NaN NaN NaN
2 NaN NaN NaN NaN NaN NaN
3 NaN NaN NaN NaN NaN NaN
4 NaN NaN NaN NaN NaN NaN
```

Gambar 4.1 Hasil membaca dataset

Pada gambar 4.1 menunjukkan hasil langkah awal membaca dan menampilkan data menggunakan *library Pandas* di Python. Baris pertama `file_path = "/content/train.xlsx"` mendefinisikan variabel `file_path` yang berisi *path* atau lokasi file Excel yang akan dibaca. Kemudian, baris selanjutnya `df = pd.read_excel(file_path)` menggunakan fungsi `read_excel()` dari *Pandas* untuk membaca data dari file Excel tersebut dan menyimpannya dalam sebuah *DataFrame* bernama `df`. *DataFrame* ini merupakan struktur data tabel yang mirip dengan tabel di *database* atau *spreadsheet*. Terakhir, perintah `print(df.head())` akan menampilkan lima baris pertama dari *DataFrame* `df`, memberikan gambaran sekilas tentang isi dan format data yang baru saja dibaca. *Output* yang ditampilkan menunjukkan beberapa kolom seperti *'tweets'*, *'class'*, dan kolom-kolom lain yang berisi nilai NaN (*Not a Number*), yang mungkin menunjukkan data yang hilang atau belum diisi.

```

↳
      tweets      class
0  I love waking up to the sound of Doris Kearns ... figurative
1      Just saw some caravans caravanning. #irony figurative
2  New wellness program launched at work Tuesday.... figurative
3  When you get curious about a company a LinkedI... figurative
4  @WashTimes: #Leesburg clothing store clerk inj... figurative

```

Gambar 4. 2 Hasil hapus kolom dan baris yang tidak diperlukan

Gambar 4.2 menunjukkan hasil dari proses menghapus kolom-kolom yang namanya mengandung kata "*Unnamed*" karena diasumsikan tidak relevan. Selanjutnya, baris-baris yang memiliki nilai NaN (*Not a Number*) dihapus untuk memastikan data yang lengkap. Kemudian, kode mendefinisikan daftar label yang *valid* dan hanya mempertahankan baris-baris yang labelnya sesuai dengan daftar tersebut, kemungkinan untuk memfilter data berdasarkan kategori tertentu. Terakhir, kode menampilkan lima baris pertama dari *DataFrame* yang telah dibersihkan dan diproses, memberikan gambaran sekilas tentang bagaimana data tersebut sekarang terlihat. *Output* yang ditampilkan menunjukkan kolom '*tweets*' dan '*class*' dengan beberapa baris contoh data yang telah diproses.

## 4.2 Preprocessing

Data yang telah diperoleh sebelumnya akan diproses dalam tahap *preprocessing*. Dalam pengolahan data terutama bidang *natural language processing* (NLP) atau pengolahan Bahasa alami, *preprocessing* adalah serangkaian Langkah yang dilakukan untuk membersihkan dan mempersiapkan data sebelum digunakan dalam model yang akan digunakan dalam proses deteksi sarkasme. Dengan melakukan *preprocessing* dapat mengurangi *noise* atau gangguan dalam data, menstandarkan *format* data, serta dapat meningkatkan kualitas dan efisiensi model yang akan digunakan. Adapun Langkah-langkah *preprocessing* yang dilakukan dalam penelitian ini adalah sebagai berikut :

### 4.2.1 Cleansing Data

Pada proses pembersihan data mendefinisikan fungsi `clean_text` untuk membersihkan data *tweet* dari elemen-elemen yang tidak diinginkan. Pertama

menghapus URL dari kalimat. Selanjutnya, mengubah semua karakter menjadi huruf kecil, menghilangkan karakter *non*-alfabetik yang akan diganti dengan spasi, merapikan spasi berlebih, menghapus username yang diawali dengan '@', menghilangkan simbol '#' dari hastag, menghilangkan *emoji* dan *emoticon*, terakhir teks akan dipecah menjadi kata untuk *stopwords* Bahasa Inggris akan dihilangkan, kemudian kata yang tersisa akan digabungkan lagi menjadi *string* yang bersih. Sebagai langkah akhir proses *cleaning* akan diterapkan pada kolom *tweets* dalam *DataFrame* *df* untuk menghasilkan kolom 'clean\_text'.

Membersihkan teks *tweet* dari karakter khusus yang tidak diinginkan atau ada salah interpretasi. Menggantikan pola teks tertentu dengan *string* kosong atau *string* yang diinginkan. Tujuan dari pembersihan ini adalah untuk memperbaiki tampilan teks *tweet* sehingga lebih mudah dibaca dan dianalisis. Beberapa karakter yang ditangani berupa karakter *Unicode* yang tidak valid, simbol yang tidak relevan, dan kesalahan *encoding*. Setelah pembersihan karakter, teks *tweet* menjadi lebih konsisten dan bisa diproses lebih lanjut.

Tabel 4. 1 *Replace Special Karakter*

Sebelum <i>cleaning</i>	Sesudah <i>cleaning</i>
#News #æ-°é» Asian shares win reprieve on ECB signal, U.S ☹☹. jobs next focus <a href="http://t.co/PEjsT4keCP">http://t.co/PEjsT4keCP</a>	#news # Asian shares win reprieve ECB signal, U.S ☹☹. jobs next focus <a href="http://t.co/PEjsT4keCP">http://t.co/PEjsT4keCP</a>
Trop de zÃ~le... RT @mashable 'Pixels' copyright notices took down the studio's own trailer <a href="http://t.co/1eoaCLjEpC">http://t.co/1eoaCLjEpC</a>	Trop de zele RT @mashable 'Pixels' copyright notices took down the studio's own trailer <a href="http://t.co/1eoaCLjEpC">http://t.co/1eoaCLjEpC</a>
â€~Even-Keeledâ€™™ White House Staffer Arrested After Shooting At Boyfriend With His Own Gun	Even-Keeled White House Staffer Arrested After Shooting At Boyfriend With His Own Gun

Pada tabel 4.1 menunjukkan contoh kalimat yang sebelum melalui tahap *cleansing replace special* karakter dan pada sisi kanan merupakan kalimat setelah tahap *cleansing special* karakter.

Proses pembersihan teks unuk menghilangkan *emoji* apabila di teks terdapat *emoji*. Pola emoji didefinisikan ke dalam *emoji\_pattern* berisi rentang karakter yang umumnya digunakan. Kemudian dikompilasi menjadi *regular expression* untuk digunakan. Fungsi `sub` digunakan untuk mengganti kemunjulan *emoji* dengan *sting* kosong.

Tabel 4. 2 menghapus emoji

Sebelum <i>cleaning</i>	Sesudah <i>cleaning</i>
#News #æ-°é—» Asian shares win reprevie on ECB signal, U.S ☹️. jobs next focus <a href="http://t.co/PEjsT4keCP">http://t.co/PEjsT4keCP</a>	#news # Asian shares win reprevie ECB signal, U.S jobs next focus <a href="http://t.co/PEjsT4keCP">http://t.co/PEjsT4keCP</a>
It's 9 o'clock and I'm having a fucking wonderful day 😞 😞	It's 9 o'clock and I'm having a fucking wonderful day
I'm glad nobody can read minds 😊 😞	I'm glad nobody can read minds

Pada tabel 4.2 menunjukkan contoh kalimat yang sebelum melalui tahap *cleansing* hapus emoji dan pada sisi kanan merupakan kalimat setelah tahap *cleansing* menghapus emoji.

Mengganti *username mention* dan menggantinya dengan *string* kosong dengan menemukan '@' kemudian akan di hapus atau diganti dengan *string* kosong.

Tabel 4. 3 hapus tag *username*

Sebelum <i>cleaning</i>	Sesudah <i>cleaning</i>
@Lauren_Southern from their assholes of course. It's such a wealth	from their assholes of course. It's such a wealth of knowledge didn't you

of knowledge didn't you know??	know??
@HabibKhanT mullah omer's death revelation made by an entity	mullah omers death revelation made by an entity
@fmanjoo @HerringWSJ lol white people and their endangered animals amirite? Hunters provide them with income	lol white people and their endangered animals amirite? Hunters provide them with income

Pada tabel 4.3 menunjukkan contoh kalimat yang sebelum melalui tahap *cleansing* hapus tag *username* dan pada sisi kanan merupakan kalimat setelah tahap *cleansing* menghapus tag *username*.

Mengganti kontraksi kata yang digunakan alam sehari-hari dengan mengubah teks *tweet* dengan format yang lebih standar seperti *he's*" menjadi "*he is*", "*there's*" menjadi "*there is*", "*won't*" menjadi "*will not*", dan sebagainya.

Tabel 4. 4 mengganti kontraksi kata bahasa inggris

Sebelum <i>cleaning</i>	Sesudah <i>cleaning</i>
I've been waiting all summer for this day	I have been waiting all summer for this day
@Lauren_Southern from their assholes of course. It's such a wealth of knowledge didn't you know??	from their assholes of course. It is such a wealth of knowledge did not you know??
I'm glad nobody can read minds 😁😬	I am glad nobody can read minds

Pada tabel 4.3 menunjukkan contoh kalimat yang sebelum melalui tahap *cleansing* mengganti kontraksi kata dan pada sisi kanan merupakan kalimat setelah tahap *cleansing* mengganti kontraksi kata.

Membersihkan dan memproses teks *tweet* Menghapus "*Character entity references*" Menggantikan kode seperti "*&amp;*;" dengan karakter yang sesuai, dalam hal ini "&", Menghapus semua tag HTML yang mungkin ada dalam teks *tweet*, sehingga hanya teks aslinya yang tersisa, Menghapus semua jenis URL yang mungkin ada dalam teks *tweet*, baik yang dimulai dengan "*http*" maupun

"https", termasuk *URL* pendek seperti "t.co", Menghapus semua tanda baca dan karakter khusus yang tidak diinginkan dari teks *tweet*, Mengubah semua huruf dalam teks *tweet* menjadi huruf kecil agar teks lebih konsisten dan mudah diolah, Menghapus kata-kata umum yang tidak memiliki banyak makna penting dalam analisis teks, seperti "the", "a", "is", dll, Menggabungkan kembali kata-kata yang tersisa setelah penghapusan *stopwords* menjadi sebuah *string* teks Kembali.

Tabel 4. 5 Hapus *stopworsd*, *lower casting*, dll

Sebelum <i>cleaning</i>	Sesudah <i>cleaning</i>
#News #æ-°é—» Asian shares win	news asian shares win reprieve ecb
reprieve on ECB signal, U.S ☹️. jobs	signal jobs next focus
next focus <a href="http://t.co/PEjsT4keCP">http://t.co/PEjsT4keCP</a>	
@Lauren_Southern from their assholes	assholes course such wealth
of course. It's such a wealth of	knowledge know
knowledge didn't you know??	
Well I'd be glad to stick around for the	Well glad stick around Funny &
#Funny & amp	

Pada tabel 4.5 menunjukkan contoh kalimat yang sebelum melalui tahap *cleansing* hapus *stopworsd*, *lower casting*, dll, pada sisi kanan merupakan kalimat setelah tahap *cleansing* hapus *stopworsd*, *lower casting*, dll.

#### 4.2.2 Tokenization

Pada proses *tokenization* ini *tokenizer* RoBERTa akan memecah kalimat menjadi *token-token* numerik yang dapat dipahami model. Tokenisasi adalah proses pemecahan teks menjadi unit-unit yang lebih kecil yang disebut token, yang bisa berupa kata, subkata, atau karakter, tergantung pada metode tokenisasi yang digunakan. Dalam konteks pemrosesan bahasa alami (NLP), tokenisasi adalah langkah awal yang krusial untuk mengubah teks mentah menjadi format yang dapat dipahami oleh model komputer. Tokenisasi dapat dilakukan dengan berbagai cara, seperti memisahkan teks berdasarkan spasi, tanda baca, atau

menggunakan algoritma yang lebih kompleks seperti *Byte-Pair Encoding* (BPE) yang digunakan oleh model seperti RoBERTa. Tokenisasi yang efektif sangat penting karena memengaruhi kualitas representasi teks dan pada akhirnya, kinerja model NLP dalam tugas-tugas seperti klasifikasi teks, terjemahan mesin, atau analisis sentimen.

Inisialisasi *tokenizer* untuk model RoBERTa yang telah dilatih sebelumnya ("*roberta-base*"). *Tokenizer* ini akan digunakan untuk memproses teks *input* sebelum diberikan ke model RoBERTa. Dalam kode ini, *tokenizer* diunduh secara otomatis dari model "*roberta-base*" yang tersedia di *Hugging Face*.

Tabel 4. 6 Hasil Tokenisasi

Sebelum tokenisasi	Setelah tokenisasi
not appreciate opinions just that they often remind me of those little air fresheners you hang in cars superficially pleasant but ultimately contributing nothing of substance	[ not ] [ appreciate ] [opinions] [just] [that] [they] [often] [remind] [me] [of] [those] [little] [air] [fresheners] [you] [hang] [in] [cars] [superficially] [pleasant] [but] [ultimately] [contributing] [nothing] [of] [substance]
People always find ways to complain about everything even when it's unnecessary	[people] [always] [find] [ways] [complain] [everything] [even] [unnecessary]
The movie had a great plot but the execution was completely disappointing	[movie] [great] [plot] [execution] [completely] [disappointing]
Sometimes the simplest solutions are the best ones yet people ignore them	[sometimes] [simplest] [solutions] [best] [ones] [yet] [people] [ignore]
His speech sounded impressive but lacked any real meaning or depth	[speech] [sounded] [impressive] [lacked] [real] [meaning] [depth]
She gave a lengthy explanation that did not answer the actual question	[gave] [lengthy] [explanation] [answer] [actual] [question]
Everyone praises his leadership skills but fails to notice his lack of responsibility	[everyone] [praises] [leadership] [skills] [fails] [notice] [lack] [responsibility]

Sebelum tokenisasi	Setelah tokenisasi
The design looked modern yet felt completely out of place in the setting	[design] [looked] [modern] [felt] [completely] [place] setting]

### 4.3 Optimasi model

#### 4.3.1 Membagi data

Membagi dataset menjadi dua bagian: data latih (train) dan data uji (test). Data ini disimpan dalam variabel X dan label kelasnya dalam variabel y. Sebanyak 80% data digunakan untuk melatih model, sementara 20% sisanya digunakan untuk pengujian. Pembagian ini dilakukan secara konsisten setiap kali proses dijalankan untuk memastikan hasil yang stabil. Setelah pembagian, fitur data latih disimpan dalam X\_train, fitur data uji dalam X\_test, sedangkan label untuk masing-masing bagian disimpan dalam y\_train dan y\_test. Jadi dari jumlah dataset sebanyak 37.824, data latih digunakan sebanyak 30.259, dan data uji sebanyak 7.565

#### 4.3.2 Parameter

Dalam pemodelan, *parameter* adalah nilai-nilai yang menentukan karakteristik atau perilaku suatu model, berbeda dengan data yang digunakan untuk melatih model. *Parameter* berfungsi untuk mengatur bentuk dan kompleksitas model, mengendalikan perilakunya, serta memungkinkan model untuk belajar. Dalam *machine learning*, *parameter* model dipelajari atau diestimasi berdasarkan data pelatihan untuk mencari nilai optimal yang meminimalkan kesalahan atau memaksimalkan kinerja model.

*hyperparameter* adalah *parameter* yang nilainya ditentukan sebelum proses pelatihan model dimulai dan berperan penting dalam mengendalikan proses pembelajaran serta arsitektur model. Penggunaan *hyperparameter* yang tepat dapat mempengaruhi kinerja model dan memastikan model dapat bekerja secara optimal. *Hyperparameter* yang digunakan pada penelitian ini dapat dilihat pada tabel 4.1

Tabel 4. 7 hyperparameter

<i>Hyperparameter</i>	<b>Nilai</b>
Learning Rate	2e-5 (0.00002)
Batch Size	16
Epoch	5
Weight Decay	0.01
Logging steps	10
Save Total Limit	2
Early Stopping	3

*Hyperparameter* yang digunakan dalam penelitian ini merupakan hasil dari serangkaian uji coba dan eksperimen yang dilakukan beberapa kali untuk menemukan kombinasi terbaik yang dapat menghasilkan kinerja model yang optimal. Setiap nilai hyperparameter, seperti *learning rate*, *batch size*, *epoch*, dan lainnya, dipilih melalui proses pencarian yang cermat untuk meminimalkan kesalahan dan memaksimalkan performa model. Dengan menguji berbagai nilai pada masing-masing *hyperparameter*, kami memastikan bahwa konfigurasi yang dipilih memberikan hasil yang paling baik dalam meningkatkan akurasi dan efisiensi model.

#### 4.4 *Fine Tuning*

*Fine tuning* adalah proses melatih kembali model *pretrained* (yang sudah terlatih sebelumnya) dengan dataset yang lebih spesifik untuk tugas tertentu. Dalam konteks model RoBERTa, *fine tuning* memungkinkan model bahasa besar ini diadaptasi ke tugas-tugas yang lebih spesifik seperti klasifikasi sentimen, *named entity recognition* (NER), atau *question answering*. Langkah-langkah *fine tuning* pada model RoBERTa meliputi penyiapan dataset yang relevan dengan

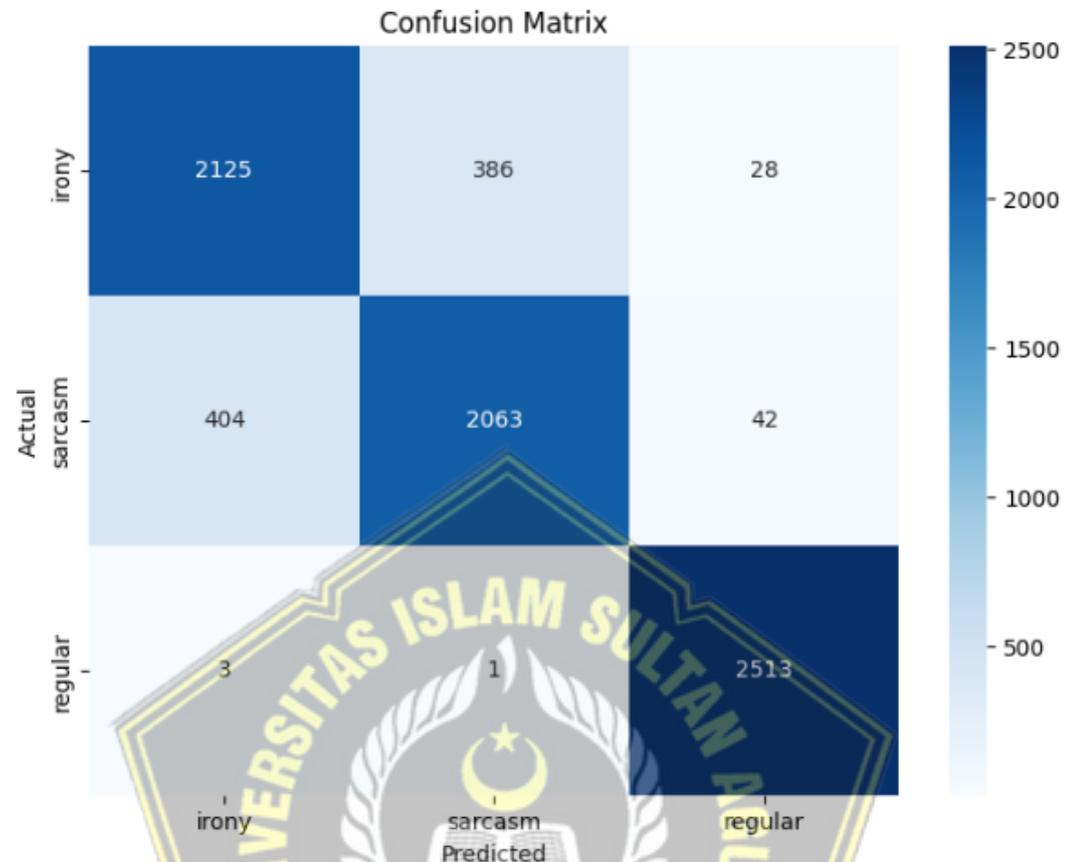
tugas yang ingin diselesaikan, memilih model RoBERTa yang sesuai, menambahkan lapisan spesifik tugas di atas model RoBERTa, melatih model RoBERTa yang telah dimodifikasi dengan dataset yang telah disiapkan, dan mengevaluasi kinerja model yang telah di-*fine tuning* menggunakan metrik yang sesuai dengan tugas yang diberikan.

#### 4.5 Hasil

Epoch	Training Loss	Validation Loss	Accuracy	Precision	Recall	F1
1	0.344900	0.283092	0.882089	0.883905	0.882089	0.881092
2	0.270500	0.278776	0.885261	0.885937	0.885261	0.884659
3	0.216600	0.342670	0.885790	0.884646	0.885790	0.885123
4	0.107600	0.496456	0.885129	0.885129	0.885129	0.884379
5	0.381600	0.532752	0.884865	0.883836	0.884865	0.884286

Gambar 4. 3 Hasil Evaluasi tiap kelas

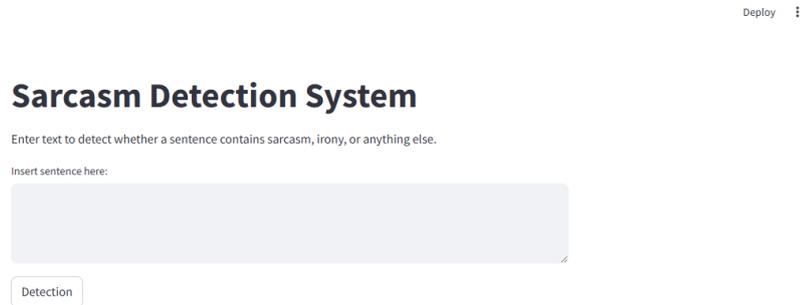
Gambar 4.9 ini menampilkan metrik kinerja model machine learning selama 5 epoch pelatihan, mencakup *Training Loss*, *Validation Loss*, *Accuracy*, *Precision*, *Recall*, dan *F1-score*. Pada epoch awal, model menunjukkan tren positif dengan penurunan *Training Loss* dan *Validation Loss*, serta peningkatan metrik akurasi lainnya. Namun, setelah epoch ke-3, terlihat adanya peningkatan signifikan pada *Validation Loss*, yang menandakan potensi overfitting. Meskipun demikian, secara keseluruhan, model menunjukkan performa yang cukup baik dengan akurasi di atas 88% di setiap epoch



Gambar 4. 4 *confusion matrix*

Pada gambar 4.10 *Confusion matrix* di atas menggambarkan kinerja model klasifikasi dalam memprediksi tiga kelas: ironi, sarkasme, dan reguler. Matriks ini menunjukkan bahwa model memiliki kinerja yang sangat baik dengan akurasi tinggi, terlihat dari nilai-nilai di diagonal utama yang jauh lebih besar dibandingkan nilai-nilai di luar diagonal. Secara khusus, model berhasil memprediksi 2125 sampel ironi dengan benar, 2063 sampel sarkasme dengan benar, dan 2513 sampel reguler dengan benar. Terdapat kesalahan klasifikasi, yaitu 386 sampel ironi yang diprediksi sebagai sarkasme, 28 sampel ironi terdeteksi sebagai reguler, 404 sampel sarkasme tereteksi sebagai ironi, 42 sampel sarkasme terdeteksi sebagai reguler, satu sampel reguler yang diprediksi sebagai sarkasme, dan 3 sampel reguler terdeteksi sebagai ironi. Secara keseluruhan, model ini menunjukkan kemampuan yang baik dalam membedakan antara ketiga kelas tersebut.

## 4.6 Tampilan *User interface*



Gambar 4. 5 Tampilan awal sebelum memasukkan kalimat

Gambar 4.11 merupakan tampilan awal sistem deteksi *sarcasm* dengan streamlit sebelum memasukkan kalimat yang ingin dideteksi.



Gambar 4. 6 Hasil Prediksi

Gambar 4.12 merupakan tampilan sistem setelah dimasukkan kalimat dan model dapat mendeteksi kalimat yang dimasukkan sebagai *sarcasm*.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Berdasarkan hasil penelitian yang sudah dilakukan tentang deteksi sarkasme kalimat berbahasa Inggris pada media sosial Twitter dari hasil evaluasi model, performa yang dicapai sangat tinggi dengan akurasi, *precision*, *recall*, dan *F1-score* di atas 89%. *Training loss* yang mencapai nol pada *epoch* 4 dan 5 menunjukkan bahwa model sepenuhnya menyesuaikan diri dengan data latih, sementara *validation loss* yang terus menurun mengindikasikan kemampuan model dalam melakukan prediksi pada data validasi. Selain itu, *confusion matrix* menunjukkan bahwa model mampu mengklasifikasikan tiga kelas (ironi, sarkasme, dan reguler) dengan sangat baik, dari total 7565 sampel. Meskipun demikian, perlu dilakukan evaluasi lebih lanjut untuk memastikan tidak terjadi *overfitting* dan model tetap dapat melakukan generalisasi dengan baik pada data baru.

#### 5.2 Saran

Berdasarkan hasil dari penelitian maka saran untuk penelitian selanjutnya adalah :

1. Untuk penelitian selanjutnya, disarankan untuk menerapkan teknik regularisasi seperti *dropout* atau *L2 regularization* untuk mengatasi kemungkinan *overfitting*, karena *training loss* pada model ini mencapai nol.
2. Mencoba model yang lebih kompleks atau menggunakan *transfer learning* untuk melihat apakah hasilnya bisa lebih baik.
3. Menggunakan dataset yang lebih besar dan lebih beragam dapat membantu model lebih memahami berbagai jenis ironi dan sarkasme.

## DAFTAR PUSTAKA

- Adinda, W., Ashifa, N., Susanto, S. S., Fauziah, A., & Nurcahyawati, V. (2025). *Analisis Sentimen Tanggapan Masyarakat terhadap Layanan Kesehatan di Kota Surabaya dengan RoBERTa*. 4(1), 139–149. <https://doi.org/10.31284/p.semtik.2025-1.6900>
- Alwan, D. A., Agung, G., Gde, N., & Nabila, O. G. (2022). *Implementasi Deteksi Judul Berita Clickbait Berbahasa Indonesia dengan pre-trained model Multilingual BERT Pada Aplikasi Berbasis Chrome Extension*. 2.
- Andriarsih, L., & Asriyani, W. (2020). *ANALISIS PENGGUNAAN BAHASA SARKASME PADA KOMENTAR NETIZEN DI MEDIA SOSIAL INSTAGRAM*. 3.
- Basbeth, F., & Fudholi, D. H. (2024). *Klasifikasi Emosi Pada Data Text Bahasa Indonesia Menggunakan*. 8(April), 1160–1170. <https://doi.org/10.30865/mib.v8i2.7472>
- Chandradev, V., Agus, I. M., Suarjaya, D., & Bayupati, I. P. A. (2022). *Analisis Sentimen Review Hotel Menggunakan Metode Deep Learning BERT*. 107–116.
- Devlin, J., Chang, M.-W., Lee, K., Google, K. T., & Language, A. I. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Naacl-Hlt* 2019, *Mlm*, 4171–4186. <https://aclanthology.org/N19-1423.pdf>
- Fitrianto, R. A., & Editya, A. S. (2024). *Klasifikasi Tweet Sarkasme Pada Platform X Menggunakan Bidirectional Encoder Representations from Transformers*. 6(3), 366–371.
- Grandini, M., Bagli, E., & Visani, G. (2020). *Metrics for Multi-Class Classification: an Overview*. 1–17.
- Hilmawan, M. D. (2022). *Deteksi Sarkasme Pada Judul Berita Berbahasa Inggris Menggunakan Algoritme Bidirectional LSTM* (Vol. 2, Nomor 1, hal. 46–51).
- Horacek, H. (2024). Natural language processing. *Computer Physics Communications*, 61(1–2), 76–92. [https://doi.org/10.1016/0010-4655\(90\)90107-C](https://doi.org/10.1016/0010-4655(90)90107-C)

- Juarto, B., & Suganda, A. (2021). *Neural Collaborative with Sentence BERT for News Recommender System*. 5(December), 448–455.
- Kusrini, & Sudarmawan. (2020). *Peningkatan Akurasi Klasifikasi Sentimen Ulasan Makanan Amazon dengan June*.  
<https://doi.org/10.35585/inspir.v10i1.2568>
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 1. <http://arxiv.org/abs/1907.11692>
- Mohiuddin, K., Welke, P., Alam, M. A., Martin, M., Alam, M. M., Lehmann, J., & Vahdati, S. (2023). Attention Is All You Need. *International Conference on Information and Knowledge Management, Proceedings, Nips*, 4752–4758. <https://doi.org/10.1145/3583780.3615497>
- Moral, P. Del, Nowaczyk, S., & Pashami, S. (2022). Why Is Multiclass Classification Hard? *IEEE Access*, 10(August), 80448–80462. <https://doi.org/10.1109/ACCESS.2022.3192514>
- Putri, Oktaviani Enjela, Pranatawijaya, Viktor H., & Kristianti, N. (2024). *ANALISIS SENTIMEN BERBASIS ASPEK DAN DETEKSI EMOSI PADA COFFEE SHOP PALANGKA RAYA MENGGUNAKAN DEEP LEARNING*. 4(September), 67–77.
- Rendragraha, A. D., Bijaksana, M. A., & Romadhony, A. (2021). Pendekatan Metode Transformers untuk Deteksi Bahasa Kasar dalam Komentar Berita Online Indonesia. *e-Proceeding of Engineering*, 8(2), 3385–3395.
- Rossum Guido van, & the Python development team. (2017). *The Python Library Reference Release 3.6.1*. 3–3. <https://cz8023.cn/book/book/python/library.pdf>
- Subakan, C., Ravanelli, M., Cornell, S., Bronzi, M., & Zhong, J. (2021). Attention is all you need in speech separation. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, 2021-June*, 21–25. <https://doi.org/10.1109/ICASSP39728.2021.9413901>
- Syahwaluddin, R., & Alita, D. (2024). Penerapan Oversampling Pada Klasifikasi Ujaran Kebencian Menggunakan Bidirectional Encoder Representations

from Transformers. *Indonesian Journal of Computer Science*, 13(4), 284–301. <http://ijcs.stmikindonesia.ac.id/ijcs/index.php/ijcs/article/view/3135>

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 2017-Decem(Nips)*, 5999–6009.

