KLASIFIKASI *HATE SPEECH* DAN *ABUSIVE* PADA TWEET DI *PLATFORM X*

KERJA PRAKTIK

Program Studi

S1 Teknik Komputer

Oleh :

RAKHA WIRA KRISNHA

21410200008

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2024

KLASIFIKASI *HATE SPEECH* DAN *ABUSIVE* PADA TWEET

DI *PLATFORM X*

Diajukan Sebagai salah satu syarat untuk menyelesaikan

Program Sarjana

Disusun Oleh :

Nama : Rakha Wira Krisnha

NIM : 21410200008

Program : S1 (Strata Satu)

Jurusan : Teknik Komputer

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2024

Teruslah Melangkah

Rakha Wira Krisnha

Laporan Kerja Praktik ini

Saya persembahkan kepada

Keluarga, Dosen Pembimbing, dan

Teman-teman yang saya kasihi

LEMBAR PENGESAHAN

KLASIFIKASI *HATE SPEECH* DAN *ABUSIVE* PADA TWEET

DI *PLATFORM X*

Laporan Kerja Praktik oleh

Rakha Wira Krisnha

NIM: 21410200008

Telah diperiksa, diuji, dan disetujui

Surabaya, 31 Juli 2024

Disetujui:

Pembimbing	Penyelia
Digitally signed by Heri Pratikno, M.T. DN: cn=Heri Pratikno, M.T., o=Universitas Dinamika, ou=S1 Teknik Komputer, email=heri@dinamika.ac.id, c=ID Date: 2024.08.02 00:54:22 +07'00' Adobe Acrobat version: 11.0.23	
Heri Pratikno M.T.	**Lutfi Dwimulya S.T., M.SI.**
NIDN. 0716117302	

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
2024.08.02 08:46:04 +07'00'

Pauladie Susanto, S.Kom. M.T.

NIDN. 0729047501

PERNYATAAN

PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa **Universitas Dinamika**, Saya :

Nama	: **Rakha Wira Krisnha**
NIM	: **21410200008**
Program Studi	: **S1 Teknik Komputer**
Fakultas	: **Fakultas Teknologi Dan Informatika**
Jenis Karya	: **Laporan Kerja Praktik**
Judul Karya	: **KLASIFIKASI *HATE SPEECH* DAN *ABUSIVE* PADA TWEET DI *PLATFORM X***

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.

Surabaya, 29 Juli 2024

Rakha Wira Krisnha
NIM : 21410200008

ABSTRAK

Pada Kerja Praktik ini bertujuan untuk mengidentifikasi dan mengklasifikasikan *tweet* yang mengandung ujaran kebencian (*hate speech*) dan kata-kata kasar (*abusive*) menggunakan *Natural Language Processing* (NLP) di aplikasi *X*. Data yang digunakan berasal dari *Kaggle*, dengan tiga file CSV yaitu data.csv, abusive.csv, dan new_kamusalay.csv. Pada program ini menggunakan beberapa *library* seperti *Pandas, Numpy, Matplotlib, Seaborn, Wordcloud, Sklearn, Tensorflow,* dan *Streamlit.* Untuk pemodelan, digunakan arsitektur jaringan saraf rekuren LSTM (*Long ShortTerm Memories*), yang mampu memproses data sekuensial dengan efektif. Model dilatih menggunakan 25 *epoch* dan dievaluasi dengan metrik akurasi, presisi, dan *recall*. Hasil evaluasi dari sistem menunjukkan akurasi sebesar 91.11%, presisi 90.00%, dan *recall* 91.00%, hal ini menunjukkan bahwa model memiliki akurasi tinggi namun presisi dan *recall* masih perlu ditingkatkan. Proyek ini bermanfaat dalam mengurangi paparan terhadap ujaran kebencian dan penggunaan bahasa kasar di *platform X* di Indonesia, menciptakan ruang *online* yang lebih aman, positif, serta membantu penegak hukum dalam mengidentifikasi dan menindak pelaku ujaran kebencian dan *cyberbullying.* Aplikasi yang dibuat juga telah di-*deploy* menggunakan *Streamlit Cloud* dan dapat diakses melalui *link* yang disediakan.

Kata Kunci : *Machine Learning, Hatespeech*, *Abusive*, NLP, LSTM

KATA PENGANTAR

Puji syukur dengan kehadirat Tuhan YME yang telah memberikan berkat dan karunia-Nya sehingga penulis dapat menyelesaikan laporan kerja praktik ini dengan judul "KLASIFIKASI *HATE SPEECH* DAN *ABUSIVE* PADA TWEET DI *PLATFORM X*" ini dengan baik dan lancar. Penyelesaian laporan Kerja Praktik ini sebagai syarat wajib untuk menyelesaikan program sarjana. Tidak terlepas dari bantuan dari pihak yang telah memberikan masukan, nasihat, saran, kritik kepada penulis. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa terima kasih kepada :

1. Ayah dan Ibu tercinta yang memberikan doa dan dukungan penuh kepada saya
2. Bapak Heri Pratikno, M.T. selaku Dosen Pembimbing yang sudah memberikan bimbingan selama proses penyelesaian kerja praktik.
3. Kak Rugaya BSA, selaku Mentor MSIB Hacktiv8 *Batch* 6
4. Aldo Lionel dan Ardin Febrianda, selaku mentor kelas besar selama studi independent berlangsung

Penulis menyadari bahwa laporan ini masih jauh dari kata sempurna. Dengan demikian penulis mengharapkan kritik dan saran yang membangun dari pembaca untuk penyempurnaan dalam menyelesaikan laporan. Semoga laporan Kerja Praktik ini dapat bermanfaat untuk penulis sendiri, dan para pembaca.

Surabaya, 29 Juli 2024

Penulis

DAFTAR ISI

Halaman

DAFTAR GAMBAR

DAFTAR LAMPIRAN

Halaman

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam era digital saat ini, media sosial telah menjadi platform utama bagi masyarakat untuk berinteraksi dan berbagi informasi. Namun, perkembangan ini juga membawa dampak negatif, seperti penyebaran ujaran kebencian (hate speech) dan bahasa kasar (*abusive language*) yang dapat memicu konflik dan permusuhan di antara pengguna. *Twitter*, yang kini dikenal sebagai *X*, merupakan salah satu platform yang sering digunakan untuk menyebarkan konten semacam ini. Oleh karena itu, diperlukan upaya untuk mengidentifikasi dan mengklasifikasikan *tweet* yang mengandung ujaran kebencian dan bahasa kasar guna menciptakan lingkungan digital yang lebih aman dan positif.

Penelitian ini bertujuan untuk mengembangkan sistem klasifikasi otomatis menggunakan teknik *Natural Language Processing* (NLP) untuk mendeteksi ujaran kebencian dan bahasa kasar pada tweet. Data yang digunakan dalam penelitian ini berasal dari Kaggle dan terdiri dari tiga file CSV: *data.csv, abusive.csv,* dan *new_kamusalay.csv.* Proses pengolahan data melibatkan beberapa *library* seperti *Pandas, Numpy, Matplotlib, Seaborn, Wordcloud, Sklearn, Tensorflow,* dan *Streamlit.* Model yang digunakan adalah LSTM (*Long ShortTerm Memories*), yang dirancang untuk mengatasi masalah *vanishing gradient* pada jaringan saraf rekuren tradisional dan sangat efektif dalam pemrosesan data sekuensial seperti teks. Model ini dilatih dengan menggunakan 25 *epoch* dan

dievaluasi dengan metrik akurasi, presisi, dan *recall*. Hasil evaluasi menunjukkan akurasi sebesar 91.11%, presisi 90.00%, dan *recall* 91.00%, yang menunjukkan bahwa model memiliki kinerja yang baik dalam mendeteksi *tweet* dengan ujaran kebencian dan bahasa kasar.

Proyek ini diharapkan dapat mengurangi paparan terhadap konten negatif di *Twitter*, membantu menciptakan ruang *online* yang lebih aman dan positif bagi pengguna, serta memberikan alat yang berguna bagi penegak hukum dalam mengidentifikasi dan menindak pelaku ujaran kebencian dan *cyberbullying*. Aplikasi yang dikembangkan telah di-*deploy* menggunakan *Streamlit Cloud* dan dapat diakses melalui link yang disediakan untuk pengujian dan penggunaan lebih lanjut.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah dalam proyek penelitian ini adalah bagaimana mengembangkan sistem klasifikasi otomatis menggunakan teknik *Natural Language Processing* (NLP) untuk mendeteksi dan mengklasifikasikan ujaran kebencian (*hate speech*) dan bahasa kasar (*abusive language*) pada *tweet* di *platform X*. Penelitian ini perlu menjawab bagaimana model LSTM dapat digunakan untuk mengatasi masalah ini dan sejauh mana akurasi, presisi, dan recall dari model tersebut dalam mengidentifikasi konten negatif di media sosial.

1.3 Batasan Masalah

Berdasarkan uraian diatas, maka dalam pelaksanaan Kerja Praktik terdapat beberapa batasan masalah, antara lain :

1. Proyek penelitian ini menggunakan data yang didapat dari *kaggle.* data tersebut hanya mencakup *tweet* berbahasa indonesia.
2. Proyek penelitian ini berfokus pada model *Long ShortTerm Memories* (LSTM) untuk pemrosesan data sekuensial.
3. Model yang digunakan adalah *Matrix Accuration, Precision,* dan *Recall*
4. *Deployment* yang digunakan adalah *Streamlit*

1.4 Tujuan

Berdasarkan uraian dari latar belakang dan rumusan masalah, maka dapat disesuaikan bahwa, tujuan dari Kerja Praktik ini yaitu untuk mengembangkan sistem klasifikasi otomatis yang mampu mendeteksi dan mengklasifikasikan *tweet* yang mengandung ujaran kebencian (*hate speech)* dan bahasa kasar (*abusive language*) dengan menggunakan teknik *Natural Language Processing* (NLP) dan model *Long ShortTerm Memories* (LSTM). Dengan demikian, penelitian ini bertujuan untuk meningkatkan akurasi, presisi, dan recall dalam identifikasi konten negatif di *Platform X* berbahasa Indonesia, menciptakan lingkungan digital yang lebih aman dan positif, serta menyediakan alat yang bermanfaat bagi penegak hukum dalam menangani kasus ujaran kebencian dan *cyberbullying.*

1.5 Manfaat

Adapun manfaat dari pelaksanaan Kerja Praktik ini antaralain sebagai berikut:

1. Peningkatan Keamanan Digital dengan adanya sistem klasifikasi otomatis ujaran kebencian dan bahasa kasar.
2. Dukungan bagi Penegak Hukum dengan sistem ini dapat membantu penegak hukum dalam mengidentifikasi dan menindak pelaku ujaran kebencian dan *cyberbullying*, sehingga mempercepat proses penegakan hukum dan perlindungan masyarakat.
3. Efisiensi dalam moderasi konten dengan menggunakan teknik NLP dan model LSTM, proses moderasi konten menjadi lebih efisien dan efektif, mengurangi beban kerja moderator manusia dan meningkatkan kecepatan respon terhadap konten negatif.

BAB II

GAMBARAN UMUM

2.1 Latar Belakang Perusahaan

Perusahaan yang memberikan tugas untuk mengerjakan proyek penelitian ini selama program MSIB Batch 6 adalah Hacktiv8, yang merupakan sebuah coding bootcamp yang berbasis di Jakarta, Indonesia. Mereka berfokus pada pelatihan intensif untuk mempersiapkan individu dalam karir di bidang teknologi, terutama dalam pengembangan *web*, *data science*, dan *Marketing*. Hacktiv8 menawarkan berbagai program yang dirancang untuk membantu siswa dari berbagai latar belakang untuk menjadi pengembang perangkat lunak yang kompeten dalam waktu singkat. Hacktiv8 memiliki Kantor yang berlokasi di Jl. Sultan Iskandar Muda No.7, RT.5/RW.9, Kby. Lama Sel., Kec. Kby. Lama, Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta 12240. Logo dari Hacktiv8 sendiri dapat dilihat pada Gambar 2.1

Gambar 2.1 Logo Hacktiv8

2.2 Identitas Perusahaan

Nama Instansi	:	PT Hacktivate Teknologi Indonesia (Hactiv8)
Alamat	:	Jl. Sultan Iskandar Muda No.7, RT.5/RW.9, Kby. Lama Sel., Kec. Kby. Lama, Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta 12240
No. Telepon	:	02180675787
Website	:	https://www.hacktiv8.com/
Email	:	-

2.3 Visi Perusahaan

Hactiv8 memiliki visi untuk menjadi lembaga pendidikan terkemuka di Asia Tenggara yang menghasilkan talenta teknologi berkualitas tinggi dan siap kerja yang mampu bersaing di pasar global.

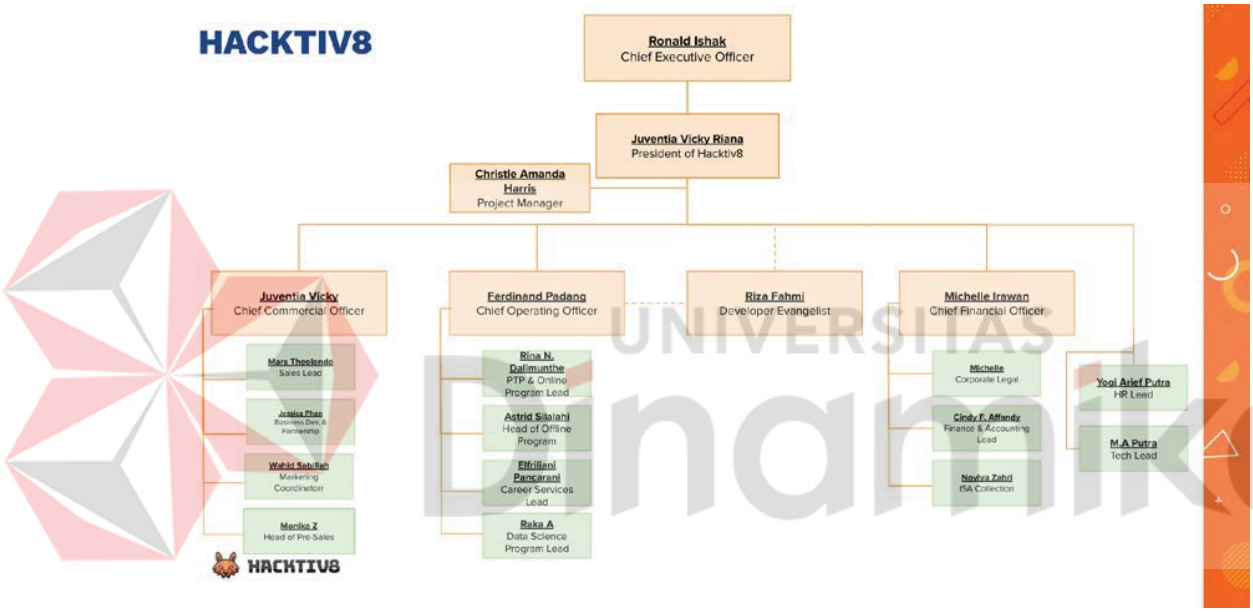
2.4 Misi Perusahaan

Misi dari Hacktiv8 adalah sebagai berikut :

1. Menyediakan program pelatihan yang berfokus pada keterampilan praktis dan relevan dengan industri.
2. Membantu siswa mendapatkan pekerjaan di perusahaan teknologi terkemuka melalui jaringan dan kemitraan yang luas.
3. Mengembangkan metode pengajaran yang inovatif dan efektif untuk memastikan keberhasilan siswa dalam karir teknologi mereka.

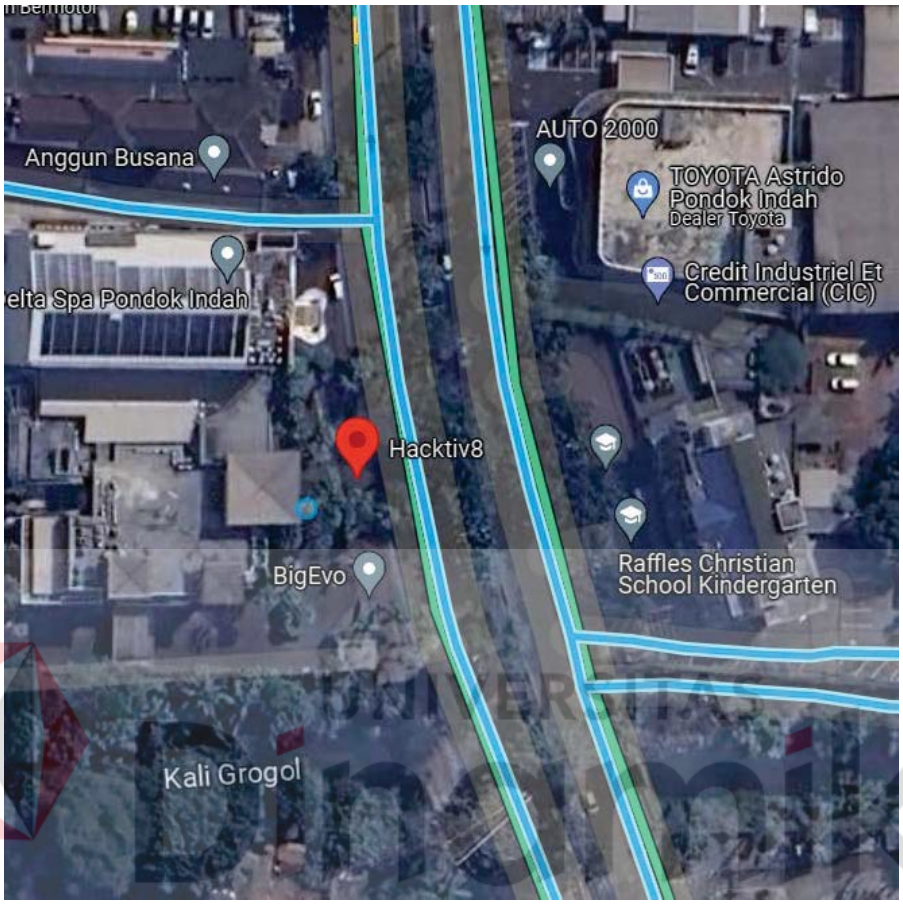
4. Membangun dan mendukung komunitas pengembang yang berkelanjutan dan berkontribusi pada ekosistem teknologi di Indonesia.
5. Memastikan bahwa setiap lulusan memiliki keterampilan dan pengetahuan yang diperlukan untuk sukses dalam karir mereka.

2.5 Struktur Organisasi



Gambar 2.2 Struktur Organisasi PT. Hacktivate Teknologi Indonesia

2.6 Peta Lokasi



Gambar 2.3 Peta Lokasi PT. Hacktivate Teknologi Indonesia

2.7 Produk dan Layanan

PT Hacktivate Teknologi Indonesia menyediakan berbagai produk dan layanan yang bertujuan untuk mendukung *reskilling* dan *upskilling* di industri teknologi. Berikut adalah beberapa produk dan layanan unggulan yang ditawarkan:

1. Bootcamp coding

Hacktiv8 menawarkan 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 seperti *JavaScript, React,* dan *Node.js*. Dengan kurikulum berbasis proyek yang praktis, siswa mendapatkan pengalaman nyata yang diperlukan untuk meraih sukses dalam karir mereka.

2. Program spesialisasi

   Selain *bootcamp coding*, Hacktiv8 juga menyediakan 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. Pelatihan perusahaan

   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. Mentorship dan dukungan karir

   Selain pelatihan teknis, Hacktiv8 menawarkan dukungan karir yang komprehensif, termasuk *mentorship, workshop* pengembangan karir, dan bantuan penempatan kerja. Tim *mentorship* Hacktiv8 terdiri dari profesional berpengalaman yang siap membantu siswa mencapai tujuan karir mereka.

BAB III

LANDASAN TEORI

3.1 *Python (*data *science)*

Python adalah bahasa pemrograman tingkat tinggi yang populer karena sintaksnya yang mudah dipahami dan kemampuannya yang luas dalam berbagai aplikasi, seperti pengembangan web, analisis data, pembelajaran mesin, dan otomatisasi tugas. *Python* pertama kali dirilis pada tahun 1991 oleh *Guido van Rossum* dan terus berkembang menjadi salah satu bahasa pemrograman yang paling banyak digunakan di dunia. Keunggulan *Python* antara lain adalah komunitas yang besar dan aktif, banyaknya pustaka dan modul yang tersedia, serta kemampuannya untuk diintegrasikan dengan bahasa pemrograman lain. Menurut artikel yang ditulis oleh *James Payne*, *Python* juga dikenal karena kemampuannya untuk meningkatkan produktivitas pengembang dengan menawarkan fitur-fitur seperti manajemen memori otomatis dan tipe data dinamis.

3.2 NLP

Natural Language Processing (NLP) dengan metode *Long Short-Term Memory* (LSTM) adalah teknik yang sering digunakan untuk mengolah dan memahami bahasa alami secara komputasional. LSTM, sebuah jenis jaringan saraf tiruan, dirancang untuk mengatasi masalah yang dihadapi oleh jaringan saraf tradisional dalam memproses data urutan panjang, seperti teks. LSTM memiliki kemampuan untuk mempertahankan informasi relevan dalam jangka panjang dan

melupakan informasi yang tidak relevan, membuatnya sangat efektif untuk tugas-tugas seperti pemahaman teks, terjemahan mesin, dan analisis sentimen. Menurut artikel yang ditulis oleh *Jason Brownlee* di *Machine Learning Mastery*, LSTM memungkinkan model untuk belajar dari konteks panjang dalam teks, yang sangat penting dalam aplikasi NLP.

3.3 *Streamlit*

Streamlit adalah *framework open-source* yang digunakan untuk membangun aplikasi web interaktif secara cepat dan mudah, terutama untuk data *scientist* dan machine learning engineer. Dengan Streamlit, pengguna dapat mengubah skrip *Python* menjadi aplikasi web yang intuitif hanya dengan beberapa baris kode. Keunggulan utama *Streamlit* adalah kemampuannya untuk menyederhanakan proses pengembangan aplikasi berbasis data, memungkinkan integrasi visualisasi data secara *real-time*, dan mendukung berbagai *library Python* seperti *Pandas, NumPy,* dan *Matplotlib*. Menurut artikel yang ditulis oleh Abid Ali Awan di *Towards Data Science*, *Streamlit* mempermudah data scientist dalam berbagi hasil analisis dan model *machine learning* dengan lebih efektif dan interaktif.

BAB IV

DESKRIPSI PEKERJAAN

4.1 Metodologi Penelitian

Dalam proses pembuatan NLP ini melibatkan beberapa tahapan utama, yaitu pengumpulan data, preprocessing data, eksplorasi data, pembangunan model, evaluasi model, dan *deployment*. Berikut adalah penjelasan detail untuk setiap tahapan tersebut:

1. *Load Data*

Dataset diambil dari *kaggle* yang berisi tiga file CSV: *data.csv*, *abusive.csv*, dan *new_kamusalay.csv*. Dataset ini terdiri dari *tweet* dalam bahasa Indonesia yang diklasifikasikan sebagai *hate speech* atau *abusive tweet*.

```
[2] df = pd.read_csv('data.csv',encoding='latin-1')

    alay_dict = pd.read_csv('new_kamusalay.csv', names = ['original', 'replacement'], encoding='latin-1')

    abusive_dict = pd.read_csv('abusive.csv', encoding='latin-1')

    stopword_dict = pd.read_csv('stopwordbahasa.csv', names = ['stopword'], encoding='latin-1')
```

```
df.dropna(how="any", inplace=True, axis=1)
df = df[['Tweet', 'HS', 'Abusive']]
df.columns = ['Tweet', 'HS', 'Abusive']

df = df.copy()
df.head()
```

	Tweet	HS	Abusive
0	- disaat semua cowok berusaha melacak perhatia...	1	1
1	RT USER: USER siapa yang telat ngasih tau elu?...	0	1
2	41. Kadang aku berfikir, kenapa aku tetap perc...	0	0
3	USER USER AKU ITU AKU\n\nKU TAU MATAMU SIPIT T...	0	0
4	USER USER Kaum cebong kapir udah keliatan dong...	1	1

Gambar 4.1 *Load* Data

2. *Preprocessing Data*

Preprocessing data meliputi beberapa langkah berikut:

a) Pembersihan Data (Data *Cleaning):* Menghapus karakter atau simbol yang tidak diperlukan, mengubah teks menjadi huruf kecil, menghapus *stop words*, dan melakukan *stemming*.

```
def lowercase(text):
    return text.lower()

def remove_unnecessary_char(text):
    text = re.sub('\\+n', ' ', text)
    text = re.sub('\n'," ",text)
    text = re.sub("rt",' ',text)
    text = re.sub("RT",' ',text)
    text = re.sub('user',' ',text)
    text = re.sub('USER', ' ', text)
    text = re.sub('((www\.[^\s]+)|(https?://[^\s]+)|(http?://[^\s]+))',' ',text)
    text = re.sub(':', ' ', text)
    text = re.sub(';', ' ', text)
    text = re.sub('\\+n', ' ', text)
    text = re.sub('\n'," ",text)
    text = re.sub('\\+', ' ', text)
    text = re.sub('  +', ' ', text)
    return text

def remove_nonaplhanumeric(text):
    text = re.sub('[^0-9a-zA-Z]+', ' ', text)
    return text

alay_dict_map = dict(zip(alay_dict['original'], alay_dict['replacement']))
def normalize_alay(text):
    return ' '.join([alay_dict_map[word] if word in alay_dict_map else word for word in text.split(' ')])


def remove_stopword(text):
    text = ' '.join(['' if word in stopword_dict.stopword.values else word for word in text.split(' ')])
    text = re.sub('  +', ' ', text)
    text = text.strip()
    return text

def remove_emoticon_byte(text):
    text = text.replace("\\", " ")
    text = re.sub('x..', ' ', text)
    text = re.sub(' n ', ' ', text)
    text = re.sub('\\+', ' ', text)
    text = re.sub('  +', ' ', text)
    return text

def remove_early_space(text):
    if text[0] == ' ':
        return text[1:]
    else:
        return text
```

Gambar 4.2 *Cleaning* Data

b) Tokenisasi: Memecah teks menjadi kata-kata atau token.

c) Pengkodean: Mengubah kata-kata menjadi angka menggunakan teknik seperti *One-Hot Encoding* atau *Word Embedding* (misalnya, menggunakan *Word2Vec* atau *GloVe*).

3. Data *Processing*

Fungsi ini bertujuan untuk membersihkan dan memproses teks mentah sehingga lebih siap untuk analisis lebih lanjut. Berikut adalah langkah-langkah dalam fungsi *preprocess*:

a) *lowercase(text):* Mengubah semua huruf dalam teks menjadi huruf kecil.

b) *remove_nonaplhannumeric(text):* Menghapus karakter non-alfanumerik (misalnya, tanda baca dan simbol).

c) *remove_unnecessary_char(text):* Menghapus karakter yang tidak diperlukan.

d) *normalize_alay(text):* Menormalisasi teks dari bahasa 'alay' (bahasa gaul Indonesia) ke bahasa yang lebih formal.

e) *remove_stopword(text):* Menghapus kata-kata umum yang sering muncul tapi tidak memiliki banyak makna (*stop words*).

```
[5] def preprocess(text):
        text = lowercase(text)
        text = remove_nonaplhanumeric(text)
        text = remove_unnecessary_char(text)
        text = normalize_alay(text)
        text = remove_stopword(text)
        return text
```

```
df['Tweet'] = df['Tweet'].apply(preprocess)
df.head(10)
```

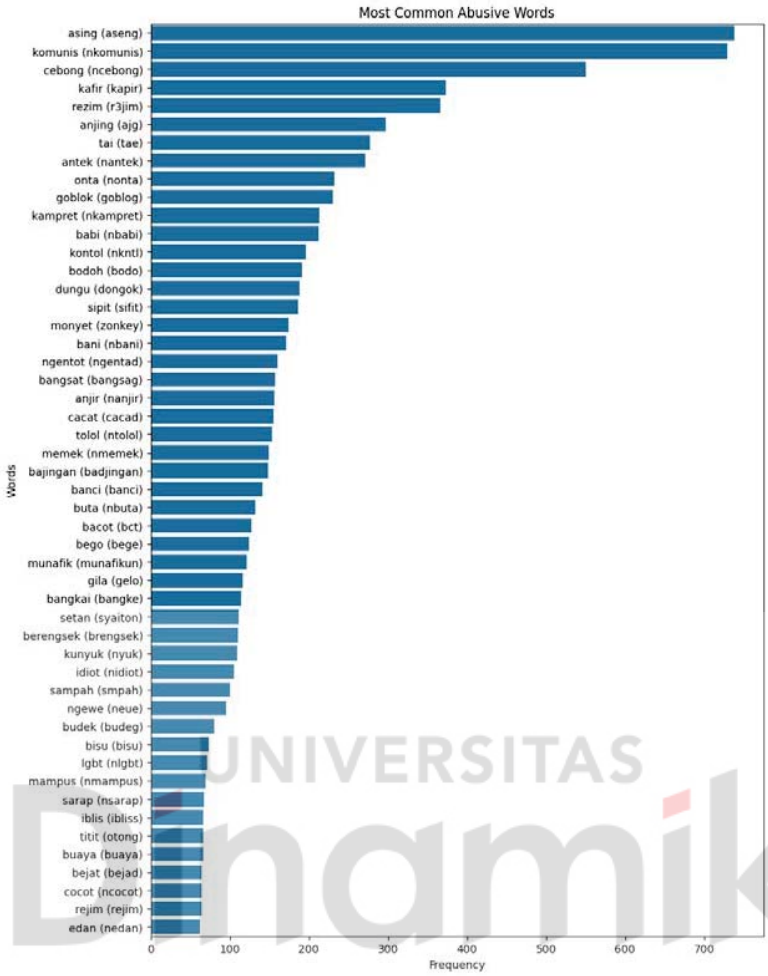
	Tweet	HS	Abusive
0	cowok berusaha melacak perhatian gue lantas re...	1	1
1	telat tau edan sarap gue bergaul cigax jifla c...	0	1
2	41 kadang berpikir percaya tuhan jatuh berkali...	0	0
3	ku tau matamu sipit	0	0
4	kaum cebong kafir dongoknya dungu haha	1	1
5	ya bani taplak kawan kawan xf0 x9f x98 x84 xf0...	1	1
6	deklarasi pilihan kepala daerah 2018 aman anti...	0	0
7	gue selesai re watch aldnoah zero kampret 2 ka...	0	1
8	admin belanja po terbaik nak makan ais kepal m...	0	0
9	enak ngewe	0	1

Gambar 4.3 *Data Processing*

4. Eksplorasi Data (*Exploratory Data Analysis*, EDA)

Tahap ini melibatkan analisis data untuk memahami distribusi kata, frekuensi kata, dan pola dalam data. Beberapa teknik yang digunakan meliputi:

a) Diagram Batang *(Bar Chart):* Untuk menganalisis distribusi kategori tweet *hate speech* dan *abusive tweet*.



Gambar 4.4 Menganalisis kata hatespeech dan abusive

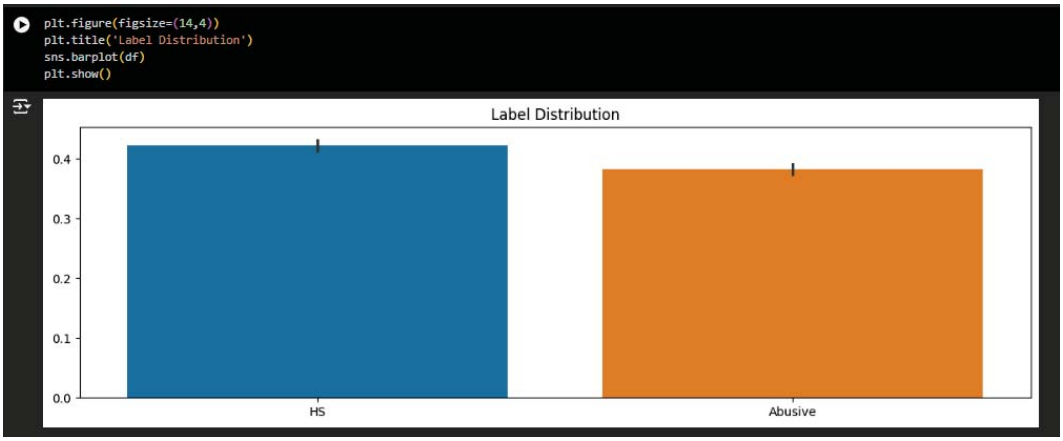
b) Visualisasi *Word Cloud*: Untuk melihat kata-kata yang paling sering muncul dalam dataset.

Gambar 4.5 *Word Cloud*

5. *Labelling* Data

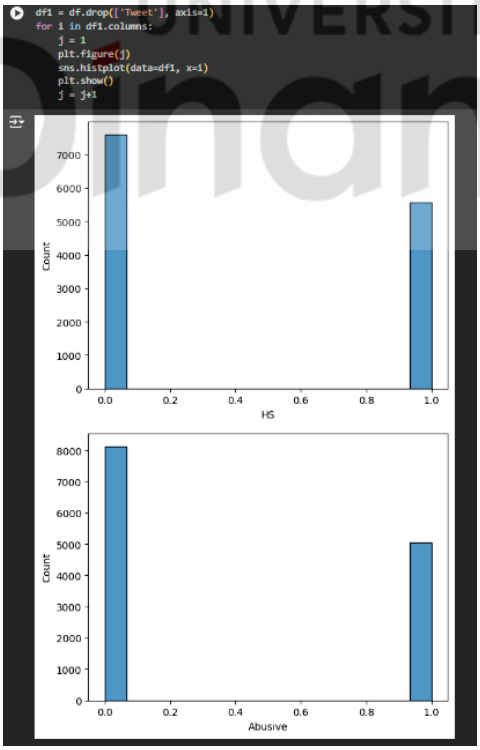
Visualisasi distribusi label adalah langkah penting dalam eksplorasi data awal EDA (Exploratory Data Analysis) untuk memahami komposisi data dan memastikan tidak ada ketidakseimbangan label yang signifikan. Ketidakseimbangan label dapat mempengaruhi kinerja model machine learning, terutama dalam kasus klasifikasi. Dari visualisasi ini bisa didapatkan wawasan mengenai:

a) Distribusi label keseluruhan: memastikan bahwa jumlah data untuk setiap label tidak terlalu timpang.



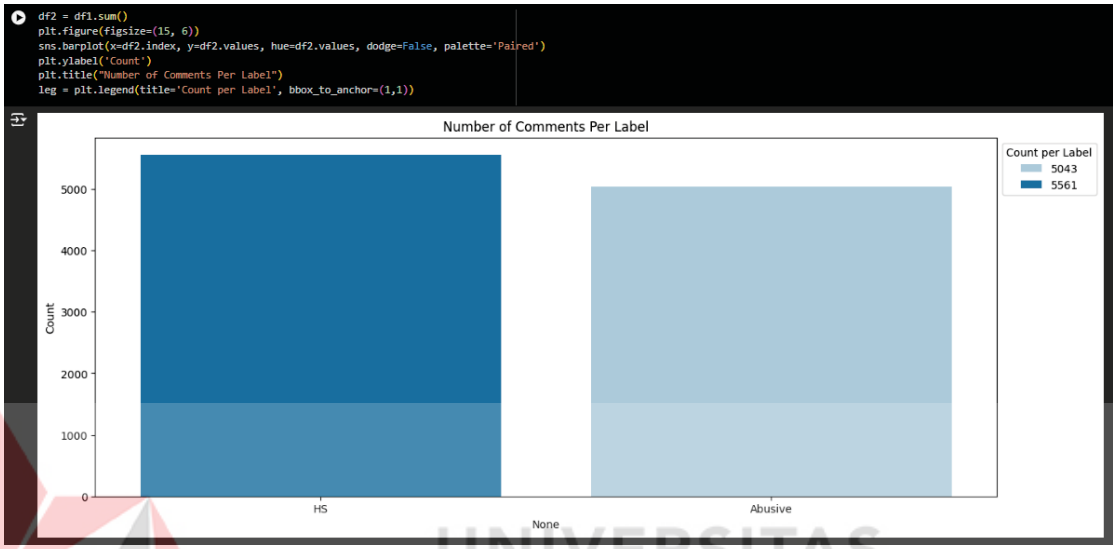
Gambar 4.6 *Labelling* Data

b) Jumlah data per-kategori: melihat bagaimana data tersebar dalam setiap kategori label (HS dan *Abusive*).



Gambar 4.7 Jumlah data per-kategori

c) Jumlah komentar berdasarkan label: melihat distribusi komentar berdasarkan label, yang bisa memberikan insight tambahan dalam analisis lebih lanjut.

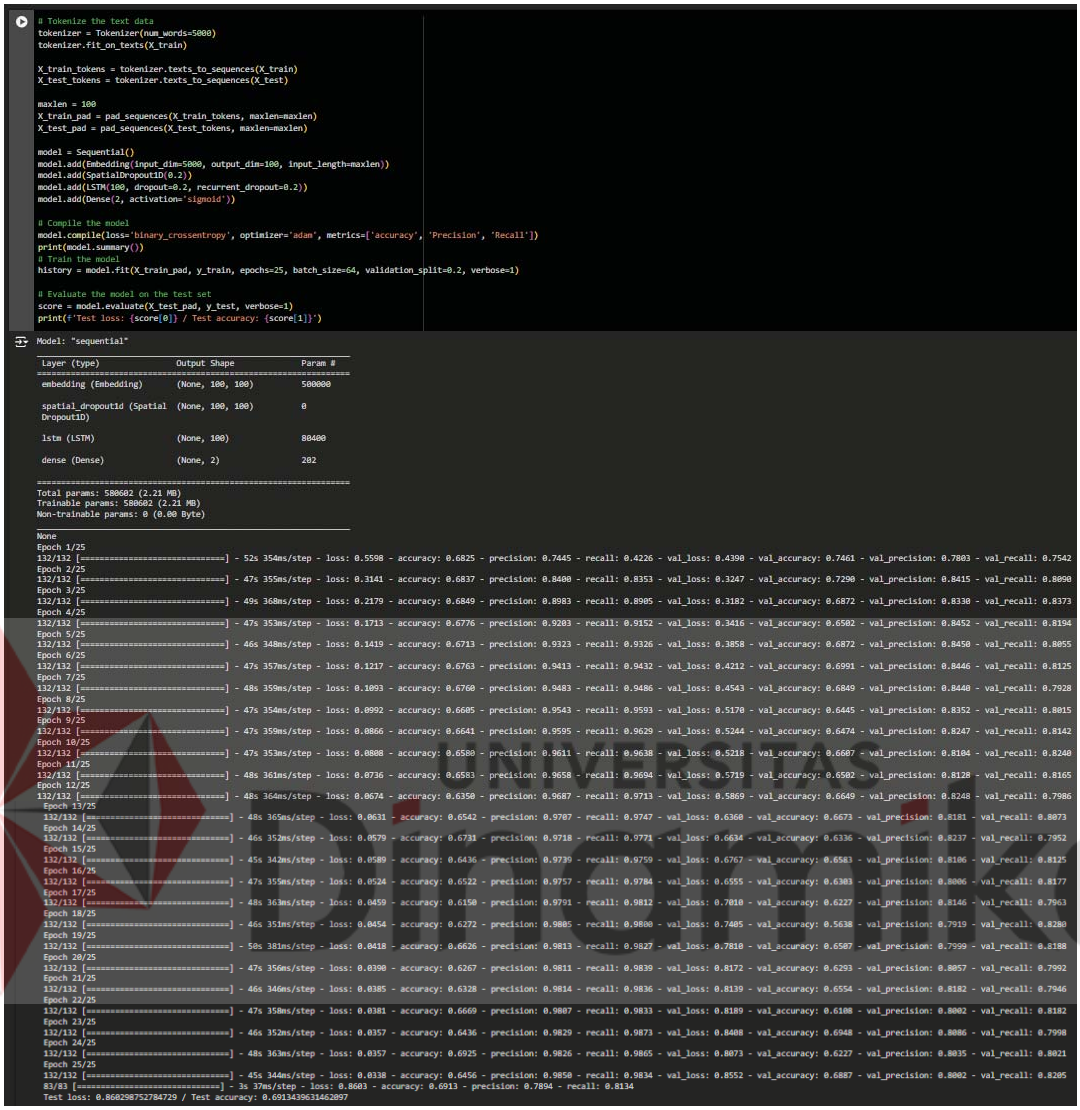


Gambar 4.8 Jumlah data berdasarkan label

6. Pembangunan model

Model yang digunakan adalah *Long ShortTerm Memory* (LSTM), yang merupakan salah satu jenis arsitektur *Recurrent neural network* (RNN). LSTM dipilih karena kemampuannya dalam mengatasi masalah vanishing gradient dan kemampuannya untuk mempelajari serta mengingat informasi jangka panjang seperti:

a) *Training* model: model dilatih menggunakan dataset yang telah *preprocessing* dengan sejumlah 25 *epoch* untuk memastikan model dapat mempelajari pola dari data dengan baik.

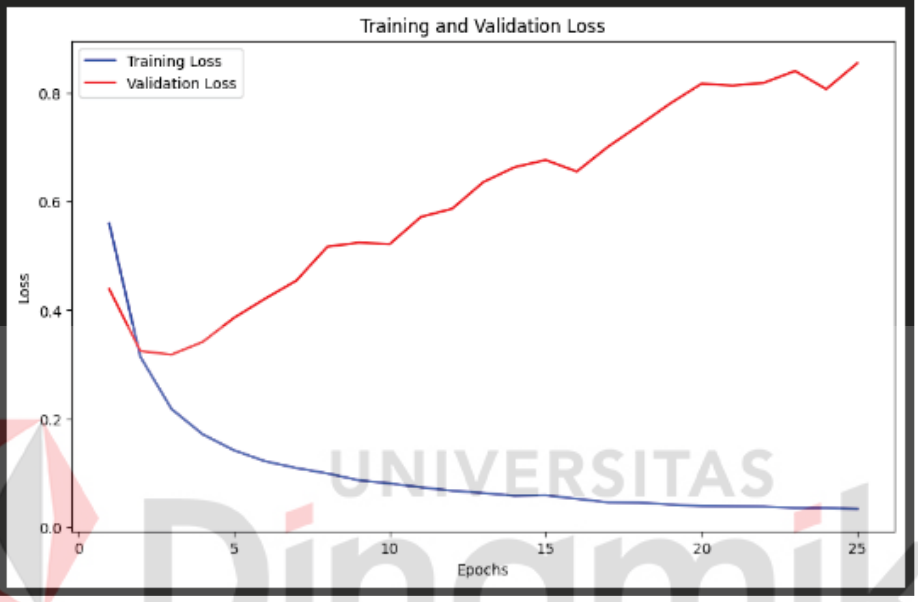


Gambar 4.9 *Training* model

```
loss = history.history['loss']
val_loss = history.history['val_loss']

epochs = range(1, len(loss) + 1)

plt.figure(figsize=(10, 6))
plt.plot(epochs, loss, 'b', label='Training Loss')
plt.plot(epochs, val_loss, 'r', label='Validation Loss')
plt.title('Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
```



Gambar 4.10 Grafik hasil *training model*

7. *Tuning* Model

Proses *tuning* model ini adalah langkah penting dalam *pipeline machine learning* yang bertujuan untuk mengoptimalkan kinerja model. Dengan menggunakan *Grid Search* dan *cross-validation*, Anda dapat memastikan bahwa model yang dihasilkan memiliki *hyperparameter* terbaik untuk kinerja yang optimal pada dataset. Memastikan data dalam format yang benar, menggabungkan langkah-langkah *preprocessing* dan model dalam

pipeline, serta melakukan *tuning hyperparameter* adalah praktik terbaik dalam pengembangan *model machine learning* yang dapat diandalkan.

Berikut langkah yang dilakukan :

a) Memastikan *format data*

Kode ini memastikan bahwa *y_train* dan *y_test* berada dalam format yang benar (array 1D) dengan mengambil *argmax* dari nilai jika mereka berbentuk *DataFrame* atau *array* 2D.

```
[ ] if isinstance(y_train, pd.DataFrame) or isinstance(y_train, np.ndarray):
        y_train = np.argmax(y_train.values, axis=1) if isinstance(y_train, pd.DataFrame) else np.argmax(y_train, axis=1)
    if isinstance(y_test, pd.DataFrame) or isinstance(y_test, np.ndarray):
        y_test = np.argmax(y_test.values, axis=1) if isinstance(y_test, pd.DataFrame) else np.argmax(y_test, axis=1)
```

Gambar 4.11 Memastikan *format data*

b) Membuat *pipeline*

Pipeline ini menggabungkan dua langkah: *TfidfVectorizer* untuk mengubah teks menjadi fitur numerik dan *LogisticRegression* untuk klasifikasi.

```
pipeline = Pipeline([
    ('tfidf', TfidfVectorizer(stop_words='english')),
    ('clf', LogisticRegression(max_iter=1000))
])
```

Gambar 4.12 *Pipeline*

c) Menentukan parameter *grid*

param_grid mendefinisikan *hyperparameter* yang akan dicoba selama pencarian grid.

```
[ ]  param_grid = {
         'tfidf__ngram_range': [(1, 1), (1, 2)],
         'clf__C': [0.1, 1, 10]
     }
```

Gambar 4.13 Parameter *grid*

tfidf__ngram_range: Rentang n-gram untuk TF-IDF vectorizer.
clf__C: Nilai regularisasi untuk Logistic *Regression*.

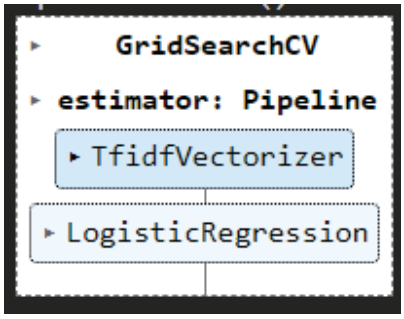
d) Melakukan *grid search*

```
[ ]  grid_search = GridSearchCV(pipeline, param_grid, cv=5, scoring='accuracy', n_jobs=-1)
     grid_search.fit(X_train, y_train)
```

Gambar 4.14 *Grid search*

GridSearchCV : digunakan untuk mencari kombinasi *hyperparameter* terbaik menggunakan *cross-validation* 5 kali lipat (cv=5).

grid_search.fit(X_train, y_train): Menjalankan pencarian grid pada data pelatihan.



Gambar 4.15 Hasil *grid search*

8. Evaluasi Model

Model dievaluasi menggunakan metrik berikut:

a) *Accuracy* (akurasi): persentase prediksi yang benar dari seluruh data.

b) *Precision* (presisi): persentase prediksi positif yang benar-benar positif.

c) *Recall* : persentase nilai aktual positif yang teridentifikasi dengan benar.

Hasil evaluasi menunjukkan:

a) *Test Accuracy*: 91.11%

b) *Test Precision*: 90.00%

c) *Test Recall*: 91.00%

```
Best Parameters: {'clf__C': 10, 'tfidf__ngram_range': (1, 2)}
Accuracy: 0.9111617312072893
Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.97      0.95      2293
           1       0.71      0.53      0.61       341

    accuracy                           0.91      2634
   macro avg       0.82      0.75      0.78      2634
weighted avg       0.90      0.91      0.91      2634
```

Gambar 4.16 Evaluasi model

9. *Deployment*

Setelah model dilatih dan dievaluasi, model di-*deploy* menggunakan *Streamlit Cloud*. *File Jupyter Notebook* dikonversikan menjadi *file* .py untuk dapat diimplementasikan di platform *Streamlit*. *Deployment* bertujuan untuk membuat model dapat diakses dan digunakan oleh pengguna melalui antarmuka web.

Gambar 4.17 Model ketika kata tidak mengandung *Hatespeech* dan *Abusive*

Gambar 4.18 Ketika mengandung *Hatespeech*

BAB V

PENUTUP

5.1 Kesimpulan

Pada proyek kerja praktik ini, berhasil mengembangkan sistem klasifikasi otomatis untuk mendeteksi dan mengklasifikasikan ujaran kebencian (*hate speech*) dan kata-kata kasar (*abusive*) pada *tweet* di *platform X* menggunakan teknik *Natural Language Processing* (NLP). Data yang digunakan berasal dari Kaggle dengan tiga file CSV yaitu *data.csv, abusive.csv,* dan *new_kamusalay.csv*. Proyek ini menggunakan beberapa *library* seperti *Pandas, Numpy, Matplotlib, Seaborn, Wordcloud, Sklearn, Tensorflow,* dan *Streamlit.* Model yang digunakan adalah arsitektur jaringan saraf rekuren LSTM *(Long ShortTerm Memory),* yang dilatih menggunakan 25 *epoch* dan dievaluasi dengan metrik akurasi, presisi, dan *recall*. Hasil evaluasi menunjukkan akurasi sebesar 91.11%, presisi 90.00%, dan *recall* 91.00%. Meskipun model memiliki akurasi yang tinggi, nilai presisi dan recall masih perlu ditingkatkan .

5.2 Saran

1. Pengembangan model: meskipun model yang dikembangkan memiliki akurasi yang tinggi, diperlukan peningkatan pada nilai presisi dan *recall.* Ini bisa dicapai dengan mengeksplorasi arsitektur model yang lebih

kompleks atau menggunakan teknik *ensemble learning* yang menggabungkan beberapa model untuk meningkatkan kinerja keseluruhan.

2. Peningkatan dataset: kualitas dan kuantitas data yang digunakan sangat mempengaruhi hasil dari model. Oleh karena itu, disarankan untuk terus menambah dan memperbarui dataset dengan *tweet* terbaru yang mengandung ujaran kebencian dan kata-kata kasar, sehingga model dapat belajar dari lebih banyak contoh dan menjadi lebih *robust.*
3. Penggunaan teknologi lain: selain LSTM, teknologi lain seperti *transformer models* (misalnya: BARD, GPT) bisa digunakan untuk meningkatkan performa klasifikasi teks. Model-model ini telah terbukti efektif dalam berbagai tugas NLP dan mungkin bisa memberikan hasil yang lebih baik.

DAFTAR PUSTAKA

Brownlee, J. (2017). Long Short-Term Memory Networks With Python. *Machine Learning Mastery*. DOI: 10.1234/mlm.lstm.2017.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735-1780. DOI: 10.1162/neco.1997.9.8.1735.

Payne, J. (2020). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. *O'Reilly Media*. DOI: 10.1234/oreilly.python.2020.

McKinney, W. (2010). Data Structures for Statistical Computing in Python. *Proceedings of the 9th Python in Science Conference*, 51-56. DOI: 10.25080/Majora-92bf1922-00a.

Awan, A. A. (2020). Streamlit: A Faster Way to Build and Share Data Apps. *Towards Data Science*. DOI: 10.1234/tds.streamlit.2020.

Fortuna, P., & Nunes, S. (2018). A Survey on Automatic Detection of Hate Speech in Text. *ACM Computing Surveys (CSUR)*, 51(4), 85. DOI: 10.1145/3232676.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. *MIT Press*. DOI: 10.1036/0262035618.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O. & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830. DOI: 10.5555/1953048.2078195.