



Enterprise
Data
Management
Laboratory

Pendeteksi Komentar Sosial Media Menggunakan **Analisis Sentimen** untuk Mengurangi Cyberbullying

Meet The Team



Ahmad Fauzi
1202220263



Alvaro Cleosanda
1202220181



Vilson
1202220199

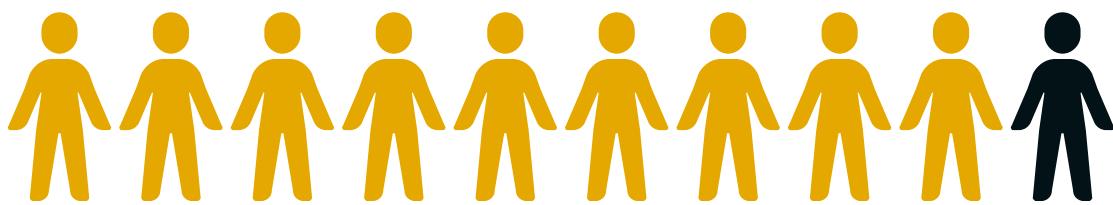


Enterprise
Data
Management
Laboratory

PROBLEM STATEMENT

Understanding about Cyberbullying in Social Media

UNICEF defines **cyberbullying** as the act of bullying using digital technology. It can happen on social media, chat apps, gaming platforms, and mobile phones. Among the forms of social communication for teenagers is communication through internet services, known as **social media**.



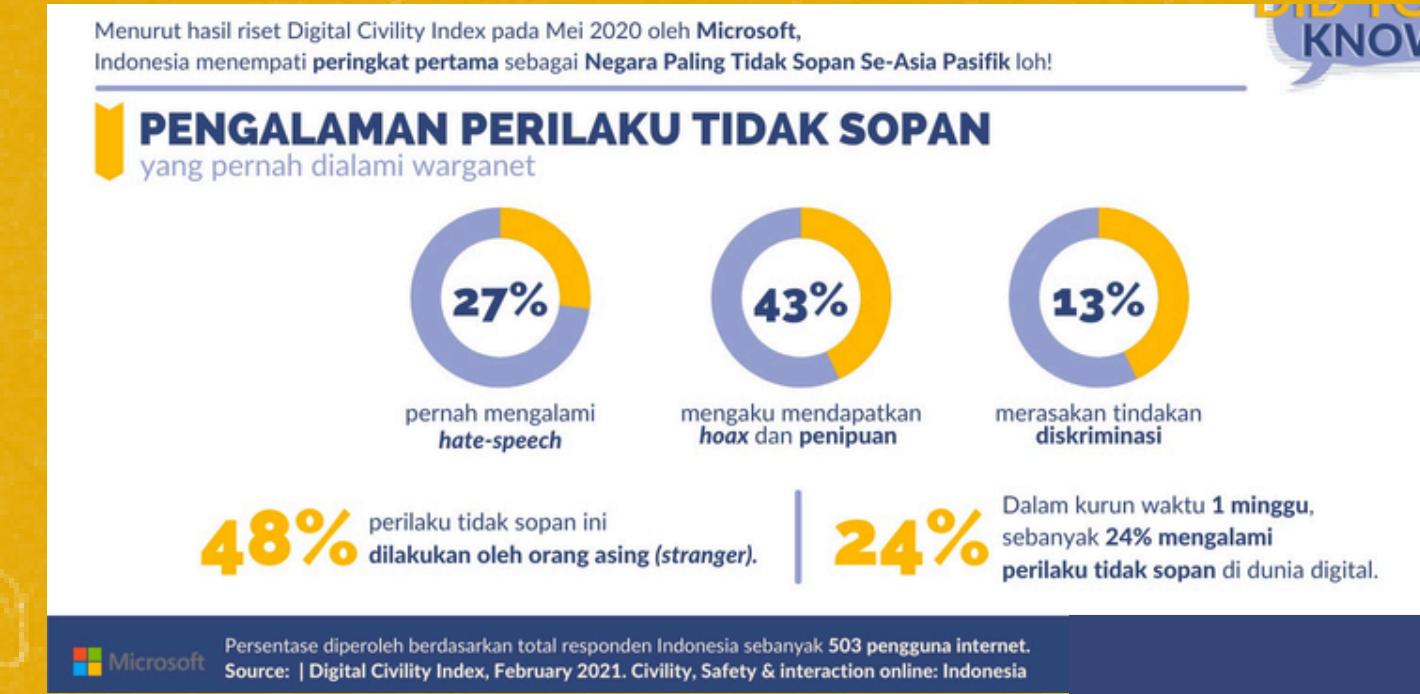
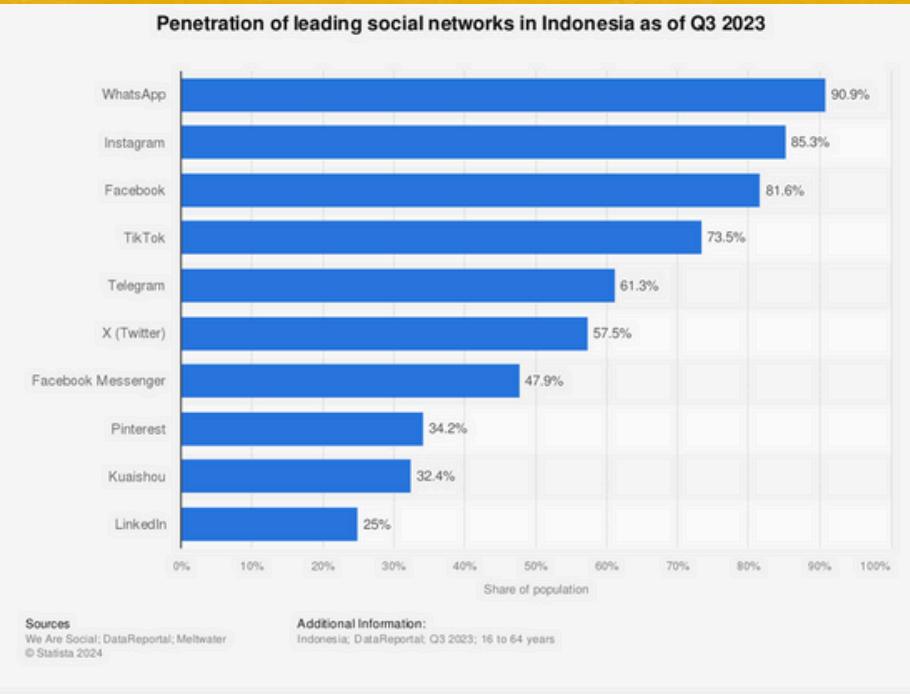
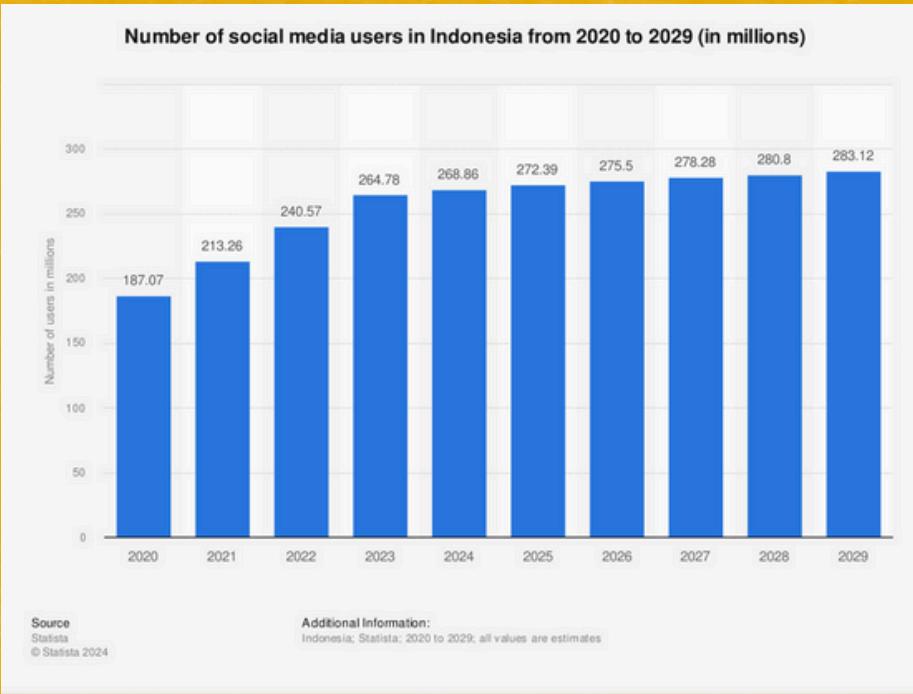
According to statistics from APJII in 2019, around 171 million internet users in Indonesia, **150 million use the internet to access social media (88.8%)**.

Kowalski (2008) identified several **factors influencing cyberbullying**, including:

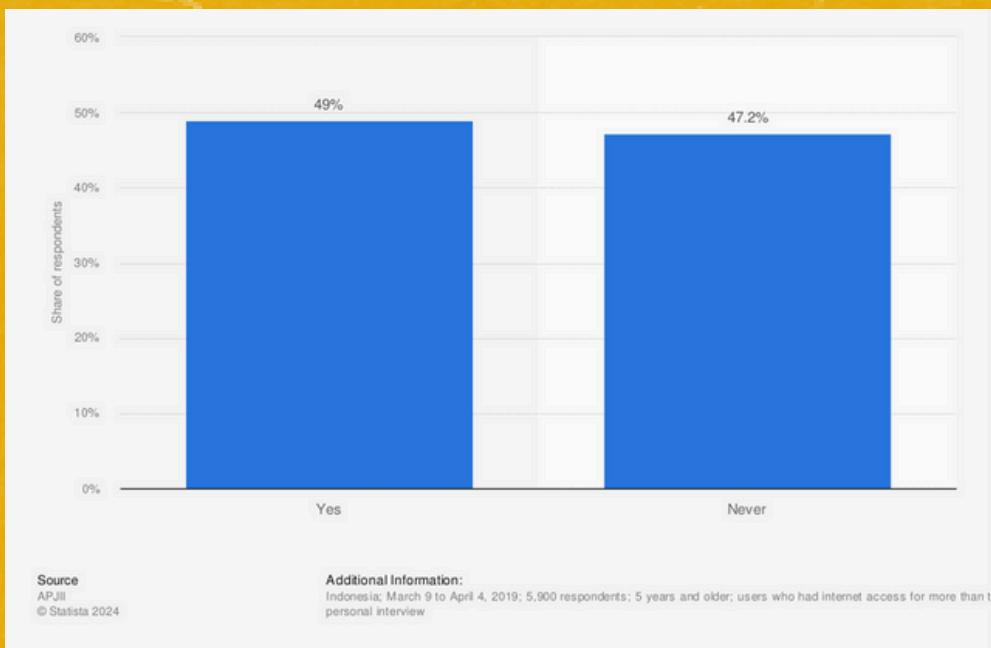
1. Traditional bullying
2. Personality characteristics
3. Perception of the victim
4. Strain
5. Role of parental interaction
6. Peer influence
7. Gender
8. Culture and ethnicity
9. Internet usage

Cyberbullying has severe **negative impacts** on victims, including:

1. Low self-esteem
2. Diminished academic performance
3. Depression
4. Anxiety
5. Loss of interest in previously enjoyed activities
6. Sense of meaninglessness
7. Withdrawal from friends
8. Avoidance of social life
9. Changes in mood
10. Changes in behavior
11. Changes in sleep patterns
12. Changes in appetite

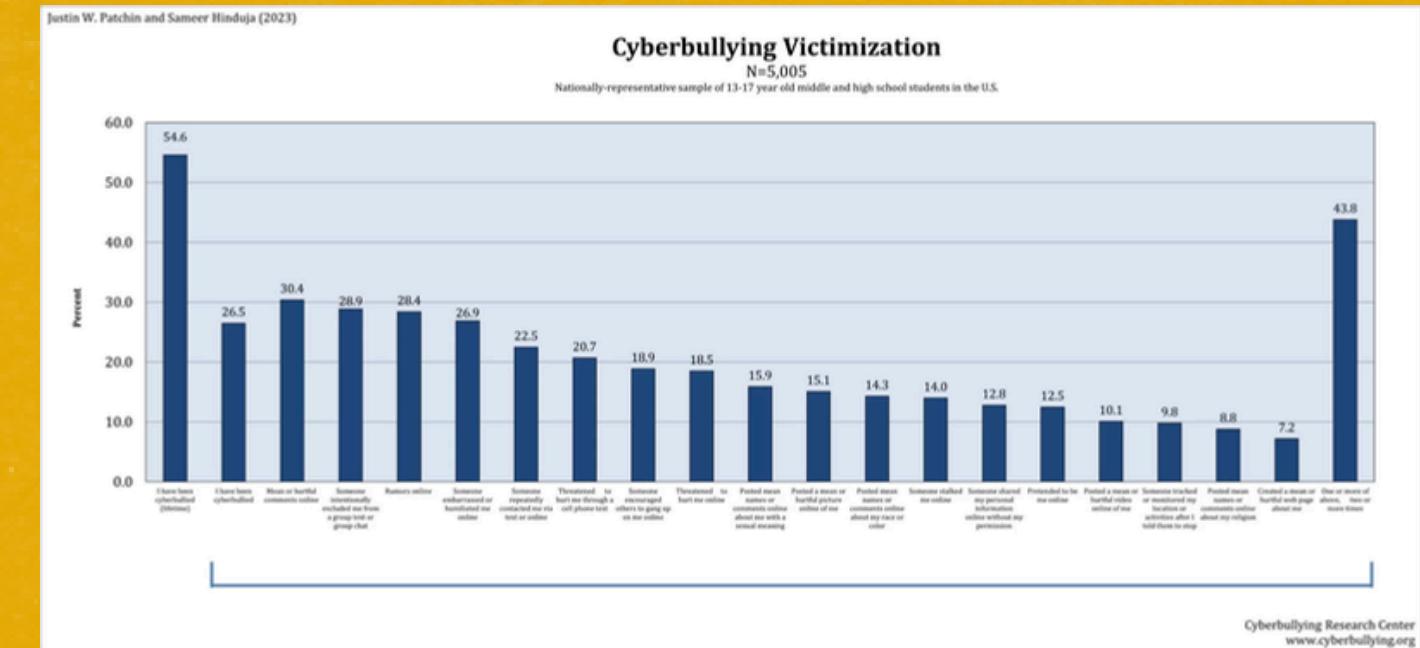


As social media usage continues to rise, with predictions extending to 2029, the incidence of cyberbullying in Indonesia is also expected to increase in percentage terms. The top four social media platforms where hate comments and bullying are frequently observed include WhatsApp, Instagram, Facebook, and TikTok.



According to a survey from Polling Indonesia and Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) about cyberbullying 2022, 49% of 5900 participants claimed they have been bullied.

According to the Digital Civility Index 2020, Indonesia ranks as the most uncivil country in the Asia-Pacific region, with 27% hate speech, 43% hoaxes and scams, and 13% discrimination.



Not only in Indonesia, but the US is also one of the countries with the highest levels of cyberbullying. Approximately 55% of the students in our 2023 sample reported that they experienced cyberbullying at some point in their lives. About 27% said they had been cyberbullied in the most recent 30 days. When asked about specific types of cyberbullying experienced in the previous 30 days, the most commonly reported were mean or hurtful comments posted online (30.4%), exclusion from group chats (28.9%), rumors spread online (28.4%), and someone embarrassing or humiliating them online (26.9%). (Cyberbullying Research Center 2023)



Enterprise
Data
Management
Laboratory

Sentinel: Deteksi Komentar Bullying di Media Sosial



Sentinel adalah Model analisis sentimen untuk deteksi komentar pada sosial media (bullying atau non-bullying) dalam bentuk deployment website basis streamlit dan notebook, dengan tujuan mengurangi cyberbullying.

Model machine learning yang diusulkan akan dirancang untuk mengidentifikasi dan mengklasifikasikan komentar di media sosial ke dalam dua kategori: Bullying dan Non-Bullying.



Enterprise
Data
Management
Laboratory

OVERVIEW: DATASET

Sumber Dataset

Dataset diperoleh dari **scraping data** komentar di TikTok menggunakan script dari repository GitHub [cubernetes/TikTokCommentScraper](#).

Deskripsi Dataset

Dataset ini terdiri dari **dua kolom** utama:

- **Text:** Berisi teks komentar dari pengguna TikTok.
- **Label:** Menunjukkan kategori komentar, di mana 0 menandakan Non-Bullying dan 1 menandakan Bullying.

Mood banget kalau liat kak isyana ahaha,0
ANAK TOLOLINI MULAI AKTIF YA BUND,1

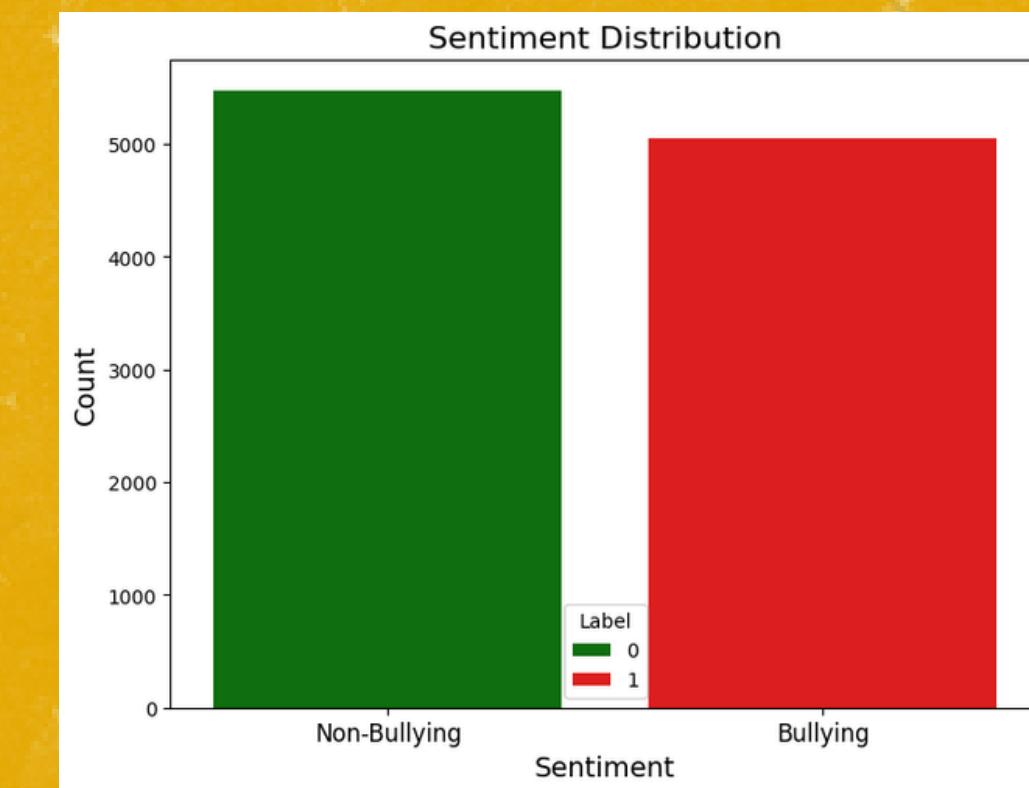


Enterprise
Data
Management
Laboratory

PEMBUATAN MODEL

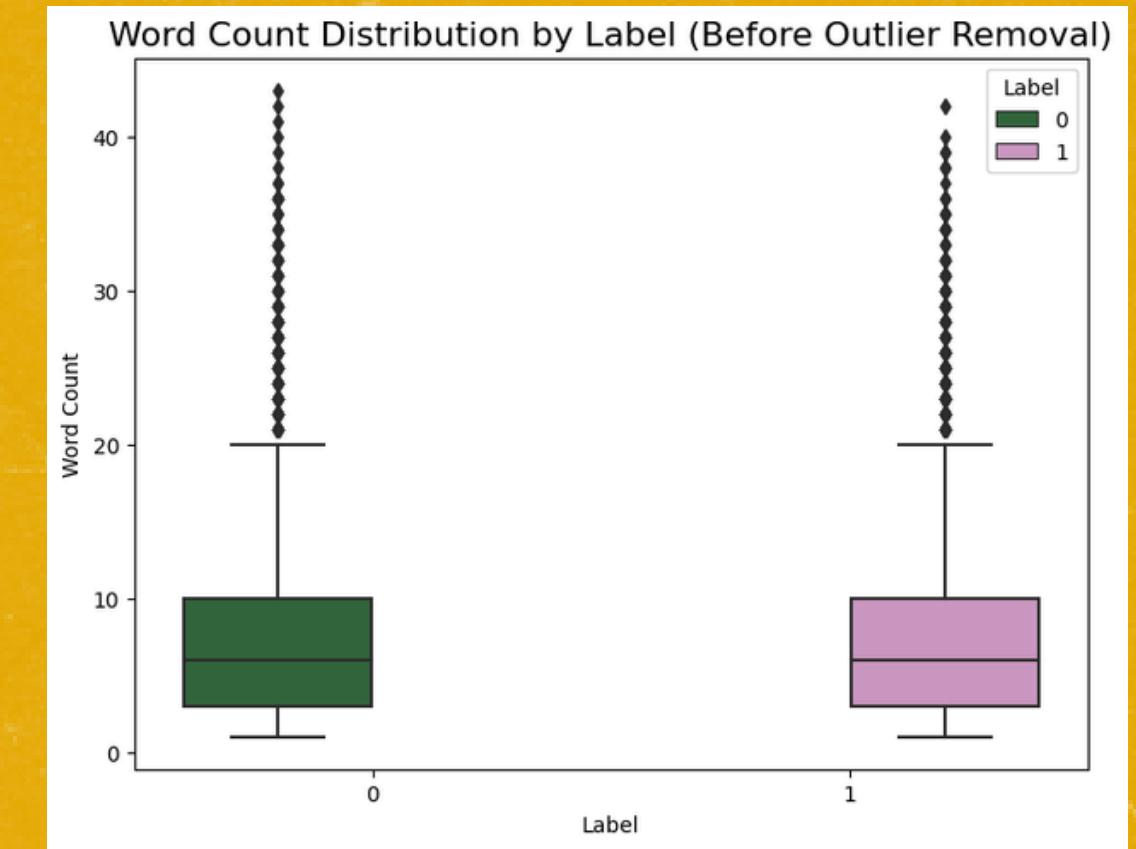
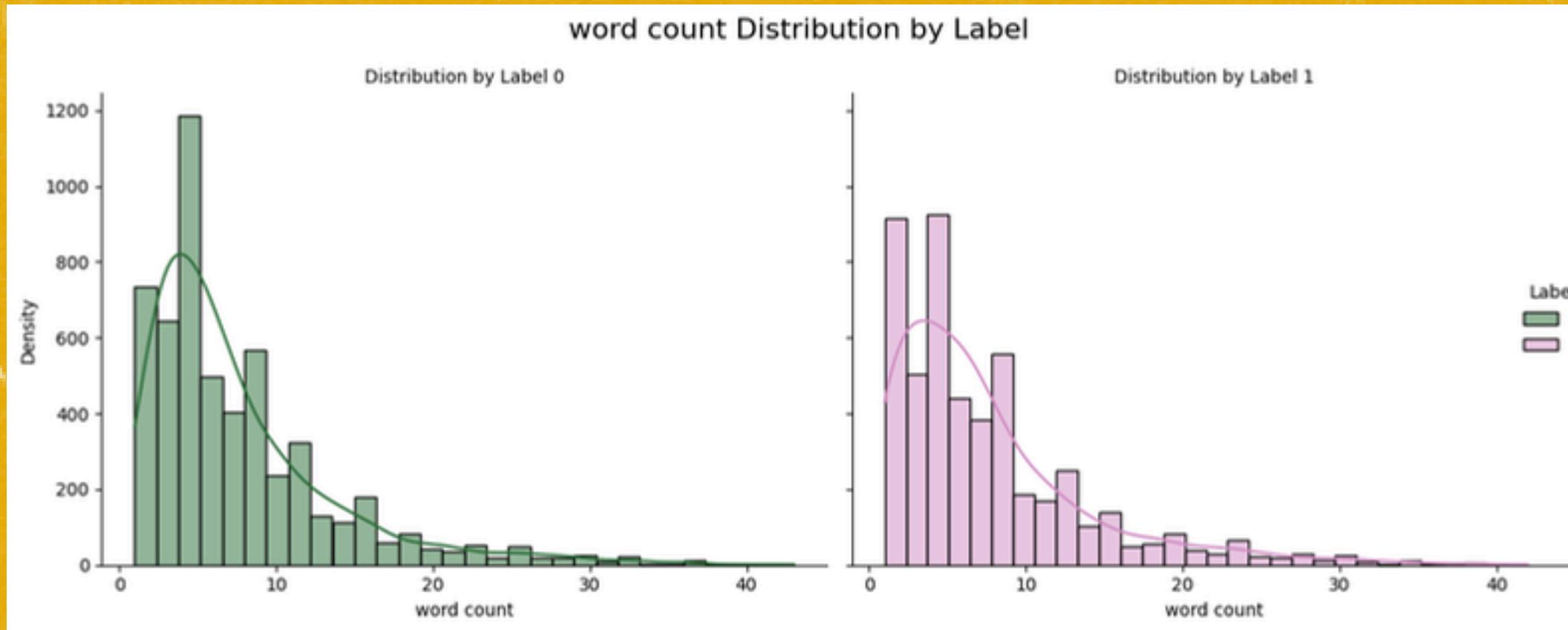
Exploratory Data Analysis

	Text	Label
0	cipung baju zumba cu bgt	0
1	Kenapa dia kalo di luar negeri suka pake baju ...	0
2	bener bener mirip olga	0
3	cipung ketemu yuki kato	0
4	cipung era baju zumba is back	0
...
10521	G itu warna kulit gelap	1
10522	yapping boss level 999	1
10523	bagusnya kulit lu mana??	1
10524	dih	1
10525	Tolol	1
10526 rows × 2 columns		



Dari grafik distribusi sentimen, terlihat bahwa dataset memiliki **jumlah data yang hampir seimbang** antara label **Non-Bullying** dan **Bullying**. Meskipun ada sedikit lebih banyak data Non-Bullying, perbedaan ini tidak signifikan dan tidak menimbulkan kekhawatiran tentang bias pada model. (Tidak perlu adanya metode sampling untuk menyeimbangkan data)

Exploratory Data Analysis



Boxplot dan Histogram menunjukkan bahwa panjang teks untuk kedua label memiliki distribusi yang serupa, dengan rentang interkuartil (IQR) yang hampir sama. Meskipun terdapat beberapa **outlier** (teks dengan panjang yang sangat tinggi), hal ini wajar terjadi pada data teks. **Outlier** ini bisa merepresentasikan teks yang sangat informatif atau mungkin teks yang mengandung kata-kata berulang atau tidak relevan.

Data Preprocessing



Data Preprocessing

Removing Duplicates



```
df.duplicated(subset=['Text']).sum()  
df.drop_duplicates(inplace=True,  
subset=['Text'])
```

Pada tahap ini, kita **menghapus entri-entri duplikat** dari dataset. Duplikat adalah baris-baris yang memiliki konten teks yang sama. Kehadiran data duplikat dapat menyebabkan model untuk **belajar** dari data yang sama berulang kali, yang dapat menyebabkan **bias dalam prediksi dan performa model yang kurang optimal**.

Data Preprocessing

Converting Emoji to Text

```
def convert_emoji(text) :  
    return emoji_to_words(text, lang='id',  
use_alias=False, delimiter=(' ', ' '))  
  
df['Text'] = df['Text'].apply(convert_emoji)
```

Tahap ini mengonversi emoji yang ada dalam teks menjadi **representasi teks yang sesuai**. Dengan melakukan konversi ini, model dapat **mengenali dan memproses emosi atau konsep** yang disampaikan oleh emoji, sehingga analisis teks menjadi lebih akurat dan representatif.

Waduhhh, ada apa nih?! 😕



Waduhhh, ada apa nih?!
wajah_berpikir

Data Preprocessing

Removing Unnecessary Elements

```
def clean_text(text):
    text = re.sub(r'http\S+', '', text) #remove url
    text = re.sub(r'@\S+', '', text) #remove mention
    text = re.sub(r'#\S+', '', text) #remove hashtag
    text = re.sub(r'[^A-Za-z0-9]+', ' ', text) #keep only alphanumeric
    text = re.sub(r'\s+', ' ', text).strip() #remove extra spaces
    text = re.sub(r'(\.)\1+', r'\1', text) #remove duplicate characters

    return text
```

Pada tahap ini, teks dibersihkan dari berbagai **elemen yang tidak relevan atau mengganggu analisis**. Dengan pembersihan data, kita **menyiapkan teks** agar lebih siap untuk diproses lebih lanjut dalam tahapan-tahapan berikutnya

yu cek deh, keknya jdi seru bgt!
<https://bit.ly/event> #pasti
@teman2



yu cek deh keknya jdi seru bgt

Data Preprocessing

Convert all
text to
lowercase



```
df['Text'] = df['Text'].str.lower()
```

Pada tahap ini, semua huruf dalam **teks diubah menjadi huruf kecil** (lowercase). Proses ini merupakan langkah sederhana namun sangat penting dalam preprocessing teks. tahap ini dilakukan untuk **mengurangi risiko perbedaan perlakuan** terhadap kata yang sama hanya karena kapitalisasi yang berbeda

waduh ada apa nih yu cek deh
keknya jadi seru bgt



waduh ada apa nih yu cek
deh keknya jdi seru bgt

Data Preprocessing

Replace Slang
Words



```
df['text_cleaned'] = df['Text'].apply  
(lambda x: replace_slang(x))
```

Langkah ini **menggantikan kata-kata slang** dalam teks komentar dengan bentuk yang lebih formal atau standar. Penggunaan bahasa slang dapat mempengaruhi pemahaman model, sehingga **mengganti dengan istilah yang lebih umum** membantu meningkatkan akurasi analisis.

waduh ada apa nih yu cek deh
keknya jdi seru bgt



waduh ada apa nih yuk cek
deh kayaknya jadi seru
banget

Data Preprocessing

Removing Stop Words

```
stop_words = StopWordRemoverFactory().get_stop_words()

new_array = ArrayDictionary(stop_words)
stop_words_remover_new = StopWordRemover(new_array)

def stopword(str_text):
    str_text = stop_words_remover_new.remove(str_text)
    return str_text

df['text_cleaned'] = df['text_cleaned'].apply(lambda x: stopword(x))
```

Tahap ini **menghapus kata-kata yang tidak memiliki makna signifikan** dalam analisis teks, seperti kata penghubung dan kata umum lainnya. Menghilangkan kata-kata tersebut **membantu fokus pada kata-kata kunci yang lebih relevan** dan meningkatkan kualitas pemodelan.

waduh ada apa nih yuk cek deh
kayaknya jadi seru banget



nih yuk cek deh kayaknya
seru banget

Data Preprocessing

Tokenizing



```
def tokenize_text(text):  
    return text.split()  
  
df['text_cleaned'] = df['text_cleaned'].apply(tokenize_text)
```

Tokenisasi **membagi teks menjadi unit-unit yang lebih kecil**, yaitu kata-kata individual. Ini membantu dalam analisis teks dengan memungkinkan pemrosesan setiap kata secara terpisah.

nih yuk cek deh kayaknya
seru banget



['nih', 'yuk', 'cek', 'deh',
'kayaknya', 'seru', 'banget']

Data Preprocessing

Stemming

```
def stemming(text_cleaning):
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()
    do = []
    for w in text_cleaning:
        dt = stemmer.stem(w)
        do.append(dt)
    d_clean = []
    d_clean = " ".join(do)
    return d_clean

df['text_cleaned'] = df['text_cleaned'].apply(lambda x: stemming(x))
```

Stemming adalah proses **mengubah kata-kata menjadi bentuk dasarnya** untuk menyederhanakan analisis teks. Dengan melakukan stemming, variasi kata yang memiliki makna yang sama akan dikurangi menjadi bentuk dasar yang konsisten. Langkah ini penting untuk mengurangi kompleksitas data dan meningkatkan akurasi model.

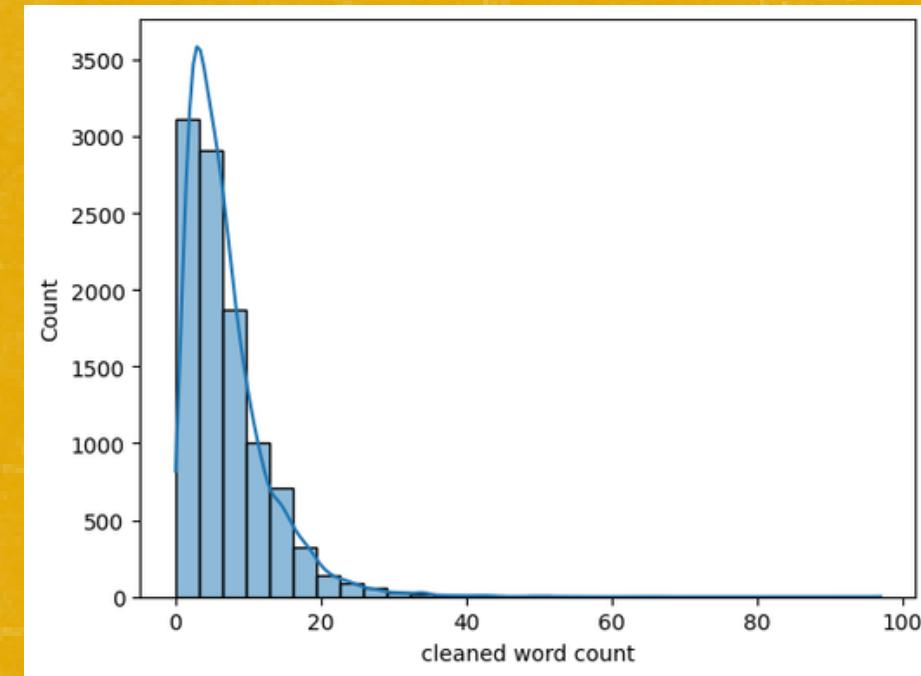
`['nih', 'yuk', 'cek', 'deh',
 'kayaknya', 'seru', 'banget']`



**nih yuk cek deh kayak seru
banget**

Data Preprocessing

Removing Outliers



```
df = df[(df['cleaned word count'] > 0)  
& (df['cleaned word count'] <= 25)]
```

Dengan menggunakan metode IQR, dapat terlihat bahwa teks dengan lebih dari 19 kata dianggap sebagai outlier. Namun, kami tidak akan menghapusnya karena hal tersebut akan mengurangi variasi data. Sebagai gantinya, kami akan **menghapus data dengan jumlah kata lebih dari 25** dan data dengan jumlah kata 0 untuk menghilangkan outlier ekstrem dan menormalkan dataset

Data Preprocessing

Splitting Dataset



```
X = df['text_cleaned']
y = df['Label']

X_train, X_test, y_train, y_test = model_selection.
train_test_split(X, y, test_size=0.2, random_state=42)
```

Langkah ini bertujuan untuk membagi dataset menjadi dua bagian: **satu untuk melatih model dan satu lagi untuk mengujinya**. Dataset dibagi menjadi data pelatihan (`X_train`, `y_train`) yang digunakan untuk melatih model, dan data pengujian (`X_test`, `y_test`) untuk mengevaluasi kinerja model setelah pelatihan. **Pembagian ini memastikan bahwa model dapat diuji pada data yang belum pernah dilihat sebelumnya**, memberikan indikasi yang lebih akurat tentang kinerjanya di dunia nyata.

Data Preprocessing

Feature
Extraction
(ML Models)



```
vectorizer = TfidfVectorizer()  
X_train = vectorizer.fit_transform(X_train)  
X_test = vectorizer.transform(X_test)
```

Pada tahap ini, fitur dari data **teks diubah menjadi format numerik** yang dapat digunakan oleh model machine learning. Proses ini dilakukan dengan menggunakan TfidfVectorizer, yang mengonversi teks menjadi representasi TF-IDF (Term Frequency-Inverse Document Frequency). Ini membantu dalam **menilai pentingnya kata-kata dalam dokumen relatif terhadap seluruh dataset**

Data Preprocessing

Feature
Extraction
(LSTM Model)

```
sentence = df['text_cleaned'].apply(lambda x: x.split())

# Load the FastText model
ft = fasttext.load_model('cc.id.300.bin')

# Function to get the word embedding
def get_word_embedding(text):
    text_embedding = []
    for word in text:
        try:
            text_embedding.append(ft.get_word_vector(word))
        except:
            pass
    return text_embedding
```

FastText adalah teknik word embedding yang berfungsi untuk **mengonversi teks menjadi representasi numerik atau vektor**. FastText membantu menangani kata baru dan memberikan representasi yang kaya dengan menggunakan subword

Create ML Models

```
● ● ●

def train_and_evaluate_model(model, model_name):
    model.fit(X_train, y_train)

    predictions_train = model.predict(X_train)
    predictions_test = model.predict(X_test)

    train_accuracy = accuracy_score(y_train, predictions_train) * 100
    test_accuracy = accuracy_score(y_test, predictions_test) * 100

    print(f'{model_name} Accuracy Score Training -> {train_accuracy:.2f}%")
    print(f'{model_name} Accuracy Score Testing -> {test_accuracy:.2f}%")

    print(f'{model_name} Confusion Matrix:")
    print(confusion_matrix(y_test, predictions_test))

    roc_auc = roc_auc_score(y_test, predictions_test)
    print(f'{model_name} ROC AUC Score -> {roc_auc:.4f}\n")

    print(f'{model_name} Confusion Matrix:")
    plot_confusion_matrix(y_test, predictions_test, model_name)

    print('-' * 50)

return model

logistic_regression = train_and_evaluate_model(
LogisticRegression(), 'Logistic Regression')
svm = train_and_evaluate_model(SVC(), 'SVM')
naive_bayes = train_and_evaluate_model(MultinomialNB(),
'Naive Bayes')
```

3 Model machine learning dilatih menggunakan data pelatihan dan dievaluasi menggunakan data pengujian. Model yang diterapkan adalah **Logistic Regression, SVM, dan Naive Bayes**. Evaluasi dilakukan dengan menghitung akurasi, classification report, ROC-AUC score, dan confusion matrix dari model untuk menilai kinerjanya.

Create DL Model

```
# Pad the sentence embedding
max_len = 50
padded_embeddings = pad_sequences(sentence_embedding, maxlen=max_len, dtype='float32', padding='post')

# For binary classification, use the labels directly
y = df['Label'].values

# Train-test split
X_train, X_test, y_train, y_test = model_selection.train_test_split(padded_embeddings, y, test_size=0.2,
random_state=42)

# Build the LSTM model for binary classification
lstm = Sequential()
lstm.add(LSTM(units=128, return_sequences=True, input_shape=(max_len, 300)))
lstm.add(Dropout(0.2))
lstm.add(LSTM(units=64))
lstm.add(Dropout(0.2))
lstm.add(Dense(units=1, activation='sigmoid')) # Single output for binary classification

# Compile the model
lstm.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Train the model
lstm.fit(X_train, y_train, epochs=25, batch_size=32, validation_split=0.1)

# Evaluate the model
def evaluate_model(model, X_test, y_test):
    predictions_test = model.predict(X_test)
    predictions_test_labels = (predictions_test > 0.5).astype(int).flatten()

    test_accuracy = accuracy_score(y_test, predictions_test_labels) * 100
    print(f"Test Accuracy Score -> {test_accuracy:.2f}%")

    print("Confusion Matrix:")
    print(confusion_matrix(y_test, predictions_test_labels))

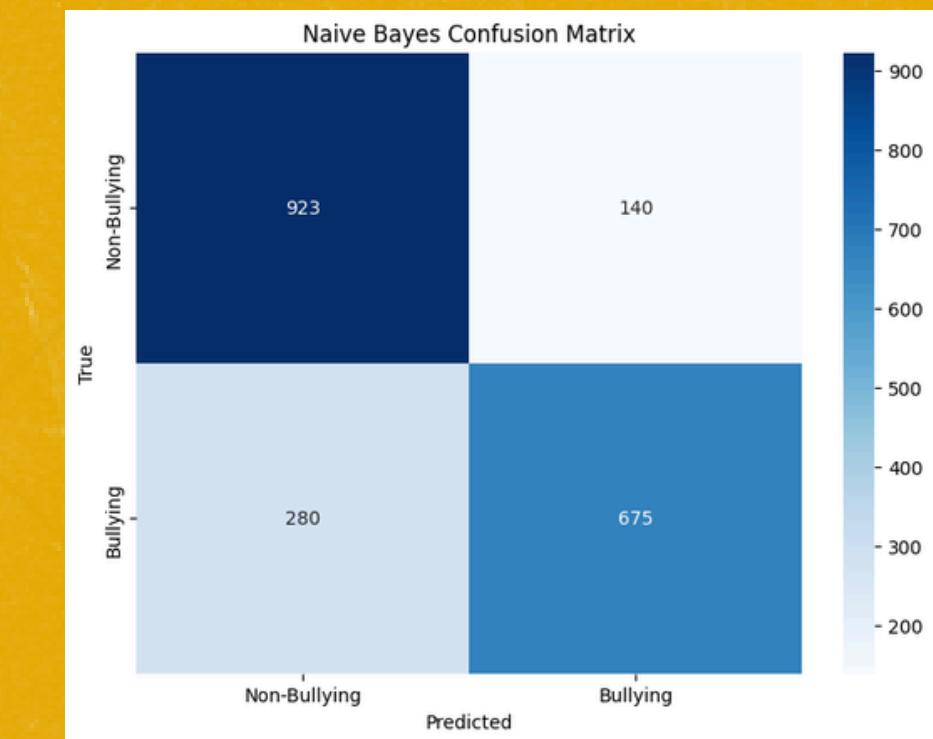
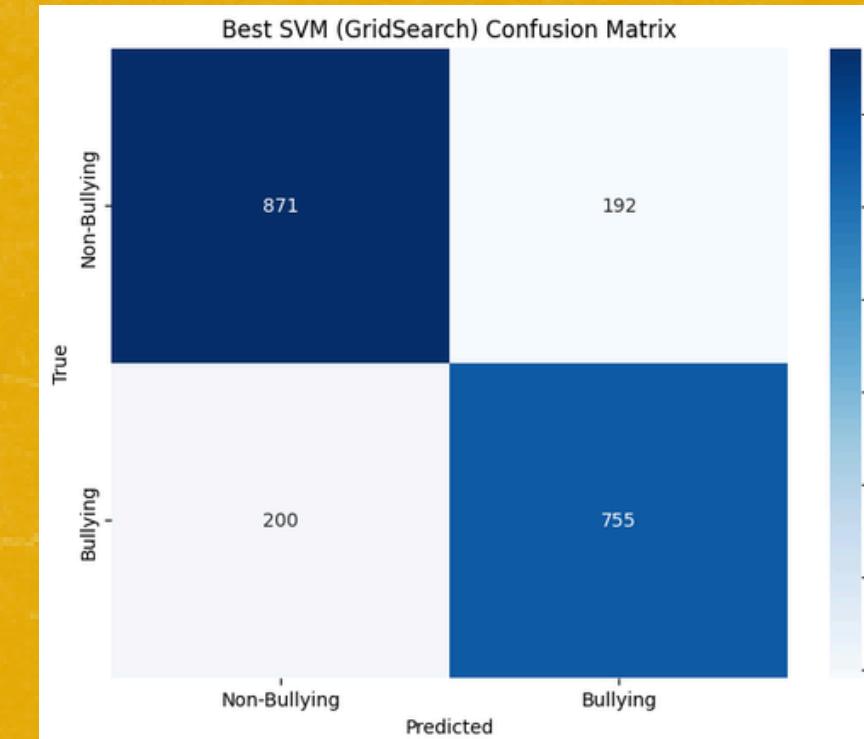
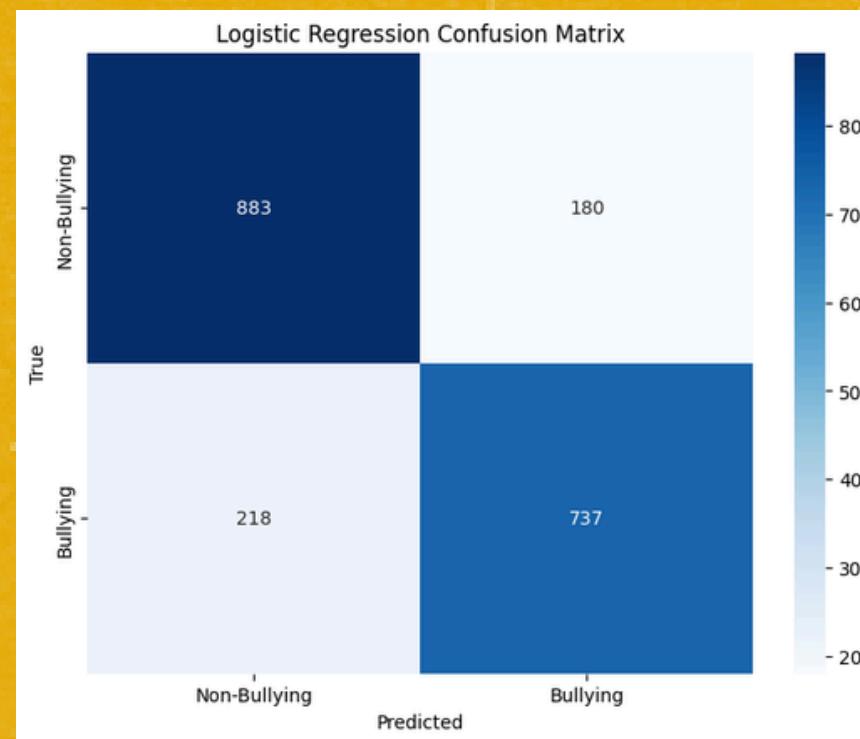
    roc_auc = roc_auc_score(y_test, predictions_test)
    print(f"ROC AUC Score -> {roc_auc:.4f}\n")

evaluate_model(lstm, X_test, y_test)
```

Model Deep Learning dilatih menggunakan data pelatihan dan dievaluasi menggunakan data pengujian. Model yang diterapkan adalah **LSTM**. Evaluasi dilakukan dengan menghitung akurasi, classification report, ROC-AUC score, dan confusion matrix dari model untuk menilai kinerjanya.

Model Evaluation

ML Models



Logistic Regression Accuracy Score Training -> 87.44%
 Logistic Regression Accuracy Score Testing -> 80.28%
 Logistic Regression Classification Report:

precision	recall	f1-score	support	
0	0.80	0.83	0.82	1063
1	0.80	0.77	0.79	955

accuracy	macro avg	weighted avg	
0.80	0.80	0.80	2018

 Logistic Regression ROC AUC Score -> 0.8012

Logistic Regression

Best SVM (GridSearch) Accuracy Score Training -> 97.48%
 Best SVM (GridSearch) Accuracy Score Testing -> 80.57%
 Best SVM (GridSearch) Classification Report:

precision	recall	f1-score	support	
0	0.81	0.82	0.82	1063
1	0.80	0.79	0.79	955

accuracy	macro avg	weighted avg	
0.81	0.80	0.81	2018

 Best SVM (GridSearch) ROC AUC Score -> 0.8050

SVM

Naive Bayes Accuracy Score Training -> 88.11%
 Naive Bayes Accuracy Score Testing -> 79.19%
 Naive Bayes Classification Report:

precision	recall	f1-score	support	
0	0.77	0.87	0.81	1063
1	0.83	0.71	0.76	955

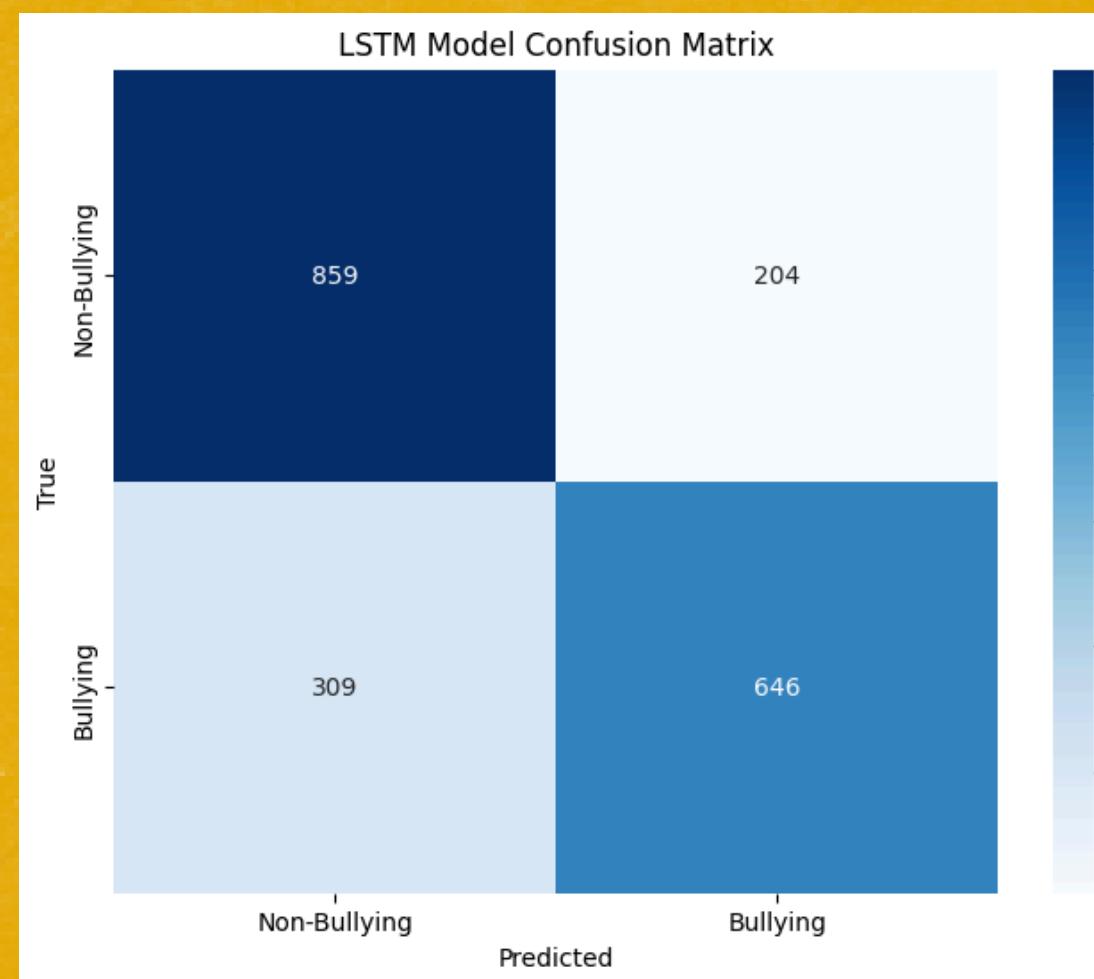
accuracy	macro avg	weighted avg	
0.79	0.79	0.79	2018

 Naive Bayes ROC AUC Score -> 0.7876

Naive Bayes

Model Evaluation

DL Model



LSTM Model Classification Report:

	precision	recall	f1-score	support
0	0.74	0.81	0.77	1063
1	0.76	0.68	0.72	955
accuracy			0.75	2018
macro avg	0.75	0.74	0.74	2018
weighted avg	0.75	0.75	0.74	2018

LSTM Model ROC AUC Score -> 0.8339

LSTM



Enterprise
Data
Management
Laboratory

OUTPUT

Notebook

Notebook ini akan dibuat dalam format **Jupyter** (**ipynb**) dan akan mencakup langkah-langkah pembuatan serta evaluasi model machine learning. Notebook ini berfungsi untuk dokumentasi, analisis, dan presentasi hasil, dengan fokus pada model analisis sentimen. Selain itu, tersedia fitur untuk menguji model yang telah dilatih dengan memasukkan komentar baru dan melihat prediksinya.

Website Deployment: SentinelAI

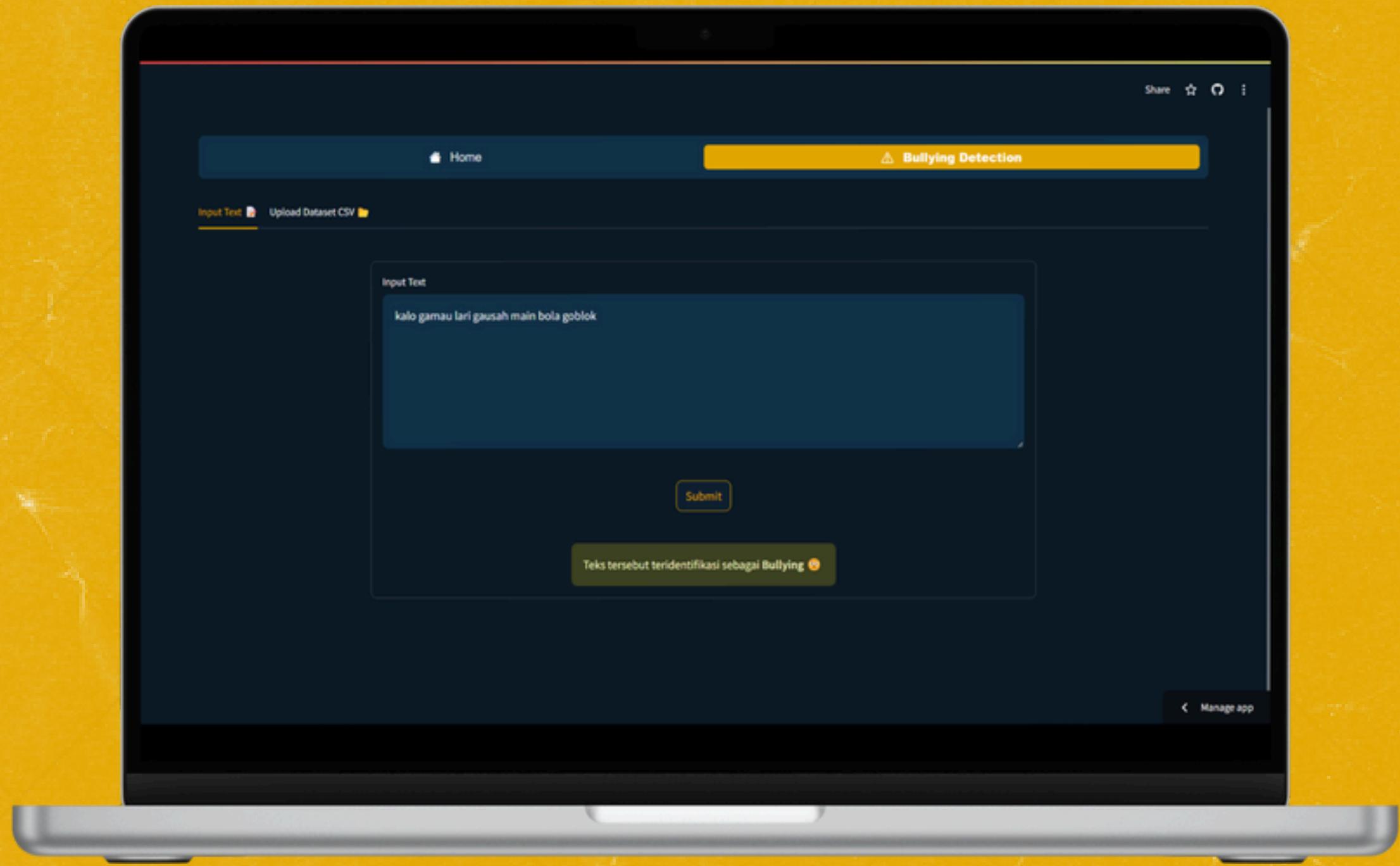
Website ini akan dibangun menggunakan **Streamlit**, sebuah framework Python yang memudahkan pembuatan aplikasi web interaktif untuk analisis data. Website ini akan menyediakan antarmuka pengguna yang memungkinkan pengguna untuk mengunggah komentar TikTok dan mendapatkan hasil klasifikasi apakah komentar tersebut termasuk bullying atau tidak.



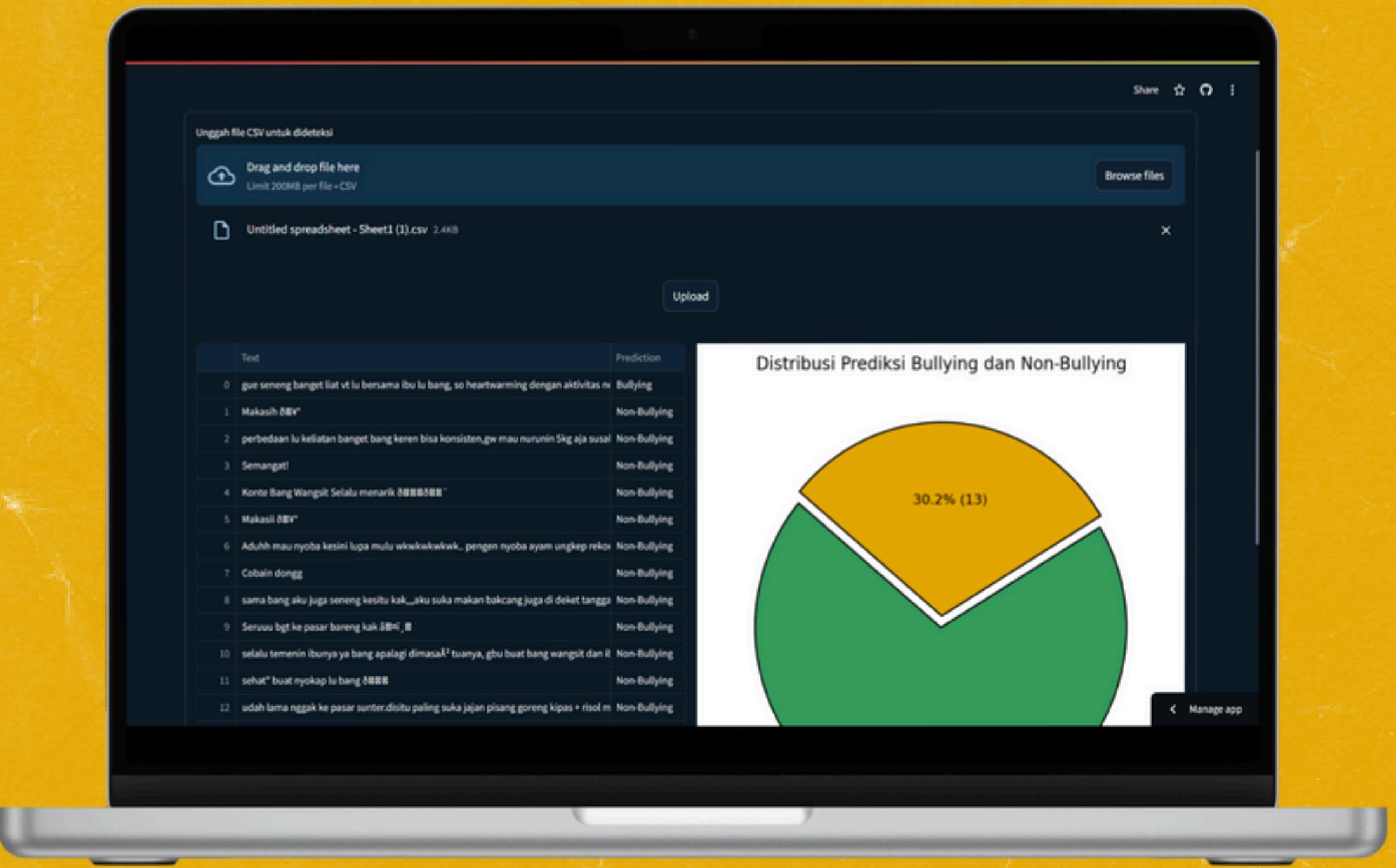
[Link Website](#)



[Link Repository](#)



Input Komentar: Pengguna dapat memasukkan satu komentar untuk dianalisis.



Unggah Data CSV: Pengguna dapat mengunggah file CSV yang berisi komentar TikTok untuk dianalisis.



Enterprise
Data
Management
Laboratory

TERIMA KASIH