

AI Bootcamp TETI
2024

INOVASI TEKNOLOGI IDENTIFIKASI TWEET OTOMATIS GUNA MENCIPTAKAN MITIGASI BENCANA BERKELANJUTAN BERBASIS AI-NLP

Presented by : Group Five



OUR TEAM

Leader

Fidelya Fredelina

Member

Adinda Putri Romadhon

Fatimah Nadia Eka Putri

Firda Salmah Atiatul Maula

Nashatra Aqila Ramadhani

OVERVIEW

01

Background

02

Problem
Identification

03

Solution

04

Machine Learning
Process

05

Improvements

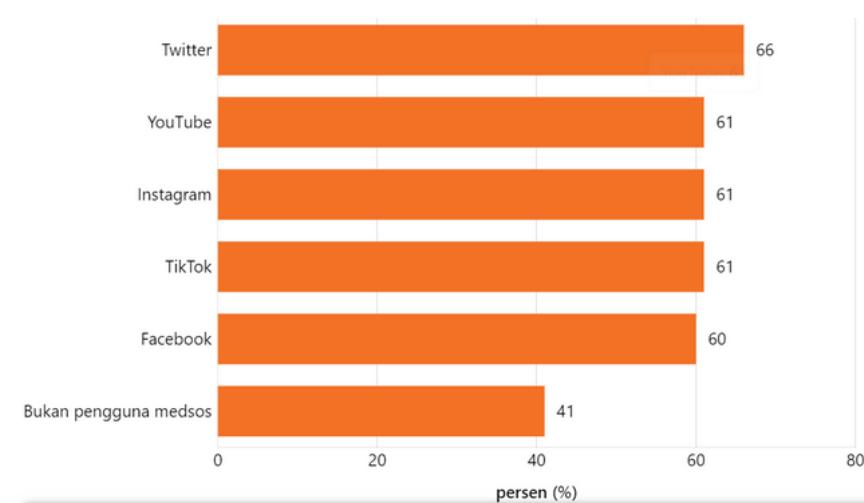
06

Conclusion



Tantangan Mitigasi Bencana di Era Digital

Proporsi Pengguna Media Sosial yang Khawatir Terhadap Perbedaan Berita Asli dan Palsu di Internet (2023)



Reuters Institute menyebut, proporsi yang menyentuh 66% dari total pengguna platform dengan nama baru X itu.

Informasi Berlebihan dan Tidak Terverifikasi

Media sosial seperti Twitter memungkinkan informasi tersebar dengan cepat, tetapi fluktuasi konten yang terjadi di media sosial dapat menyebabkan **pengaburan informasi penting**. Misalnya, ketika isu bencana alam muncul, konten trending mungkin tidak relevan atau bahkan bertentangan dengan informasi bencana tersebut.



Peran Twitter sebagai platform utama interaksi dan berbagi informasi

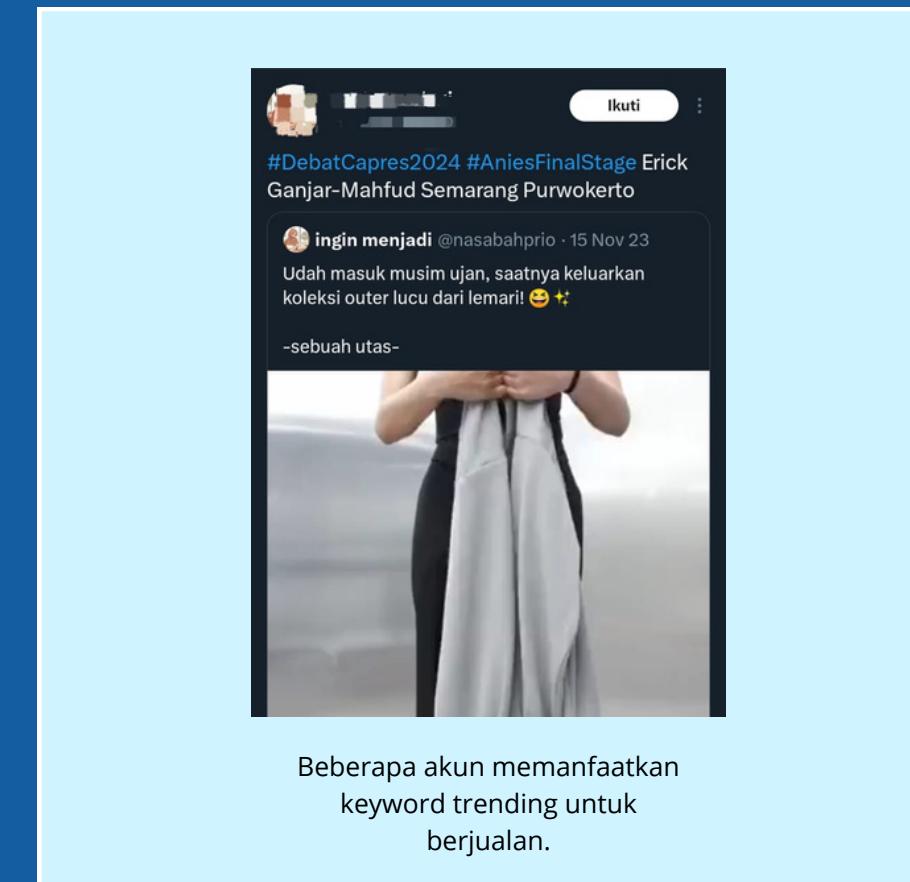
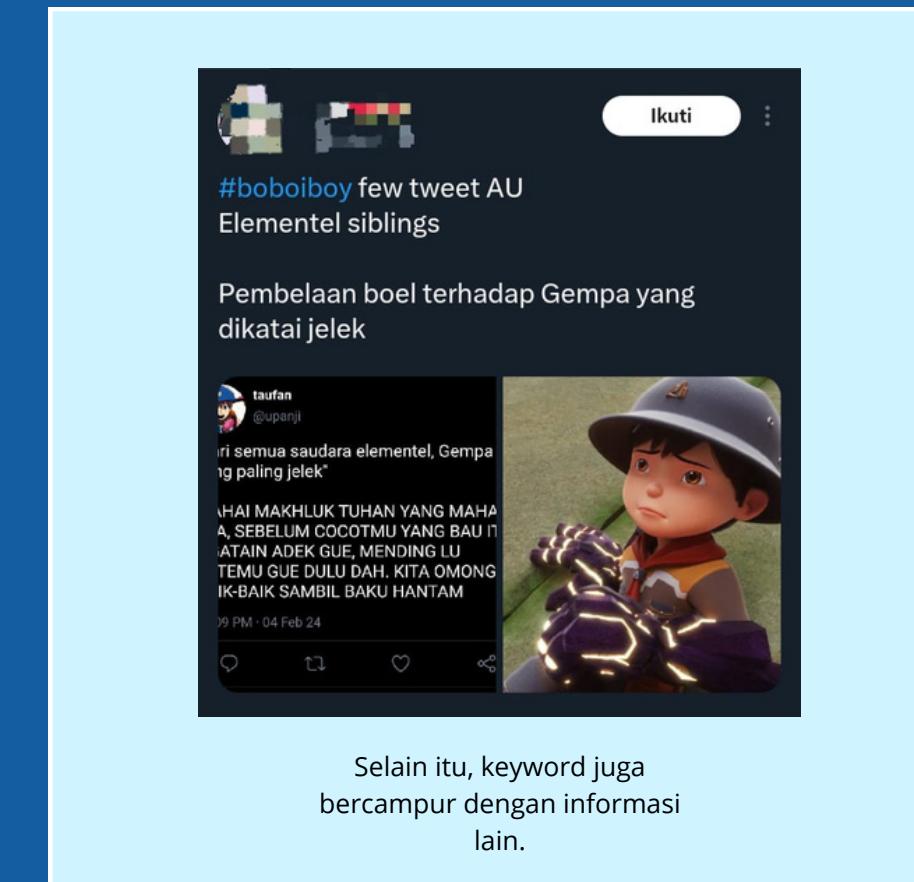
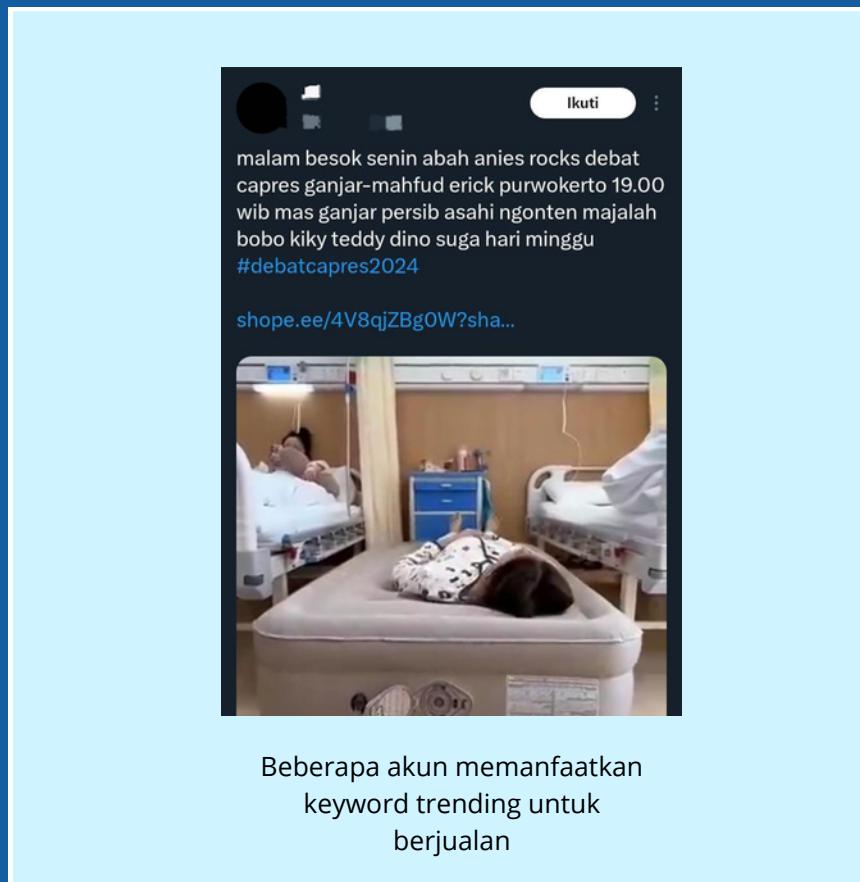
Twitter telah menjadi salah satu platform utama di mana pengguna dapat berinteraksi secara real-time dengan orang lain.

Perkembangan Teknologi AI dalam Pemrosesan Bahasa Alami

Perkembangan teknologi AI dalam NLP tidak hanya terbatas pada penelitian akademis, tetapi juga telah **diintegrasikan ke dalam aplikasi-aplikasi nyata** seperti asisten virtual, analisis sentimen dalam media sosial, penerjemahan otomatis, dan lain-lain. Ini menunjukkan **dampak positif** dari kemajuan teknologi AI dalam meningkatkan pengalaman pengguna dan efisiensi komunikasi.

PROBLEM IDENTIFICATION

Twitter, yang awalnya diciptakan untuk memfasilitasi respons cepat terhadap peristiwa darurat dan menyebarkan informasi dengan kecepatan, sekarang disalahgunakan dengan fluktuasi konten tidak relevan seperti pornografi, hoaks, dan isu yang tidak terkait. Akibatnya, informasi bencana yang seharusnya krusial menjadi terabur dan terlupakan.



TUJUAN

Mengembangkan sebuah sistem AI yang secara otomatis dapat mengidentifikasi dan menganalisis tweet yang berkaitan dengan mitigasi bencana dengan berfokus pada hashtag yang sedang trend.

MANFAAT

1. Identifikasi cepat
2. Pelaporan akurat
3. Respons efektif
4. Mengurangi dampak negatif konten tidak relevan
5. Meningkatkan kesadaran publik



SOLUSI: IDENTIFIKASI OTOMATIS DENGAN AI-NLP

Metode

Pentingnya informasi yang akurat dalam respons bencana mendorong penggunaan kecerdasan buatan (AI) dengan algoritma dan analisis NLP. Dengan pendekatan terstruktur, AI dapat mengidentifikasi pola-pola penting dalam informasi sosial, menyajikan data relevan dan terpercaya kepada pihak berwenang. Hal ini tidak hanya meningkatkan efisiensi mitigasi bencana, tetapi juga melindungi masyarakat dari dampak merugikan informasi yang salah atau tidak relevan.

Dalam upaya meningkatkan respons bencana, identifikasi otomatis menggunakan kecerdasan buatan (AI) memainkan peran kunci. Algoritma melibatkan langkah-langkah seperti pengumpulan data, preprocessing, analisis sentimen, dan pengidentifikasi pola untuk menyajikan informasi relevan dan terpercaya. Dengan demikian, algoritma ini berkontribusi pada efisiensi dan ketepatan respons dalam mitigasi bencana.

Konsep penggunaan AI

SOLUSI: IDENTIFIKASI OTOMATIS DENGAN AI-NLP

NLP

Pada projek kali ini, kami memanfaatkan berbagai metode dan framework dalam Natural Language Processing untuk memproses data tekstual yang kami dapatkan. Kami memanfaatkan tokenizing dan stop words untuk memproses tweet sebelum kemudian di-vectorize untuk dimasukkan ke dalam model AI.

Binary
Classification

Pada intinya masalah yang kami hadapi adalah masalah binary classification, di mana kami mempunya 2 target class yang harus diprediksi. Target class tersebut berupa 1 (Disaster) dan 0 (Non-Disaster). Menggunakan classification model seperti Logistic Regression, SVM, dan Random Forest Classifier, kami mencoba untuk memprediksi apakah suatu tweet dengan keyword tertentu relevan atau tidak relevan dalam konteks bencana. Akurasi model dihitung menggunakan ROC AUC dan F1 Score.

MACHINE LEARNING PROCESS

DATA CLEANING

- metode **duplicated()**

```
[ ] raw_df[raw_df.duplicated(subset=['text', 'target'])]
```

	id	keyword	location	text	target
103	103	aftershock	fresno, ca	thinking about how some1 spat blood at me duri...	0
292	292	annihilation	NaN	[AUS] Vic Bushfire Appeal Doggos Animal Fa...	0
771	771	bioterror	Planet Earth	The human cull (from 38mins30secs https://t.co...	0
787	787	bioterror	NaN	Does the push for mass vaccination point towar...	0
808	808	bioterror	United States	Does the push for vaccination point to a stage...	0

menemukan **baris-baris duplikat** dalam DataFrame `raw_df` berdasarkan kolom '`text`' dan '`target`'. Jika ada baris yang memiliki teks dan target yang sama, baris-baris duplikat tersebut akan ditampilkan.

- metode **isnull()**

```
[ ] raw_df.isnull().sum()
```

id	0
keyword	0
location	3382
text	0
target	0
<code>dtype: int64</code>	

Ini berguna untuk mengetahui apakah ada nilai null dalam kolom-kolom tertentu.

- metode **drop_duplicates()**

```
[ ] raw_df.drop_duplicates(subset=['text', 'target'], inplace=True)
print(raw_df.shape)
```

(11225, 5)

Selanjutnya, **menghapus baris-baris duplikat** dari DataFrame menggunakan metode `drop_duplicates()`. Parameter `subset=['text', 'target']` digunakan untuk menentukan kolom-kolom yang digunakan sebagai kunci untuk mengidentifikasi duplikat. Sedangkan, parameter `inplace=True` digunakan untuk menyimpan perubahan pada DataFrame yang ada tanpa perlu membuat salinan baru.

- metode **dropna()**

```
[ ] raw_df.dropna(subset=['text', 'target'], inplace=True)
```

Ini berguna untuk menghapus baris-baris yang memiliki nilai null dalam kolom '`text`' dan '`target`' dari DataFrame.

MACHINE LEARNING PROCESS

EXPLORATORY DATA ANALYSIS (EDA) & VISUALIZING

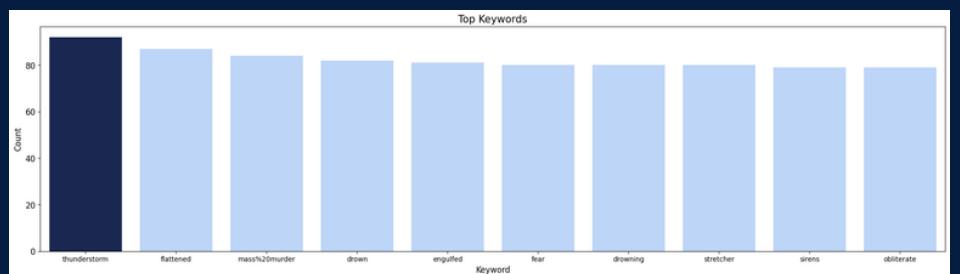
```
[ ] # exploration to count the amount of data for each keyword  
keyword_count_df = raw_df['keyword'].value_counts().reset_index(name='count')  
print(keyword_count_df)
```

index	count	
0	thunderstorm	92
1	flattened	87
2	mass%20murder	84
3	drown	82
4	engulfed	81
..
214	electrocuted	15
215	rainstorm	11
216	siren	10
217	deluged	10
218	tsunami	6

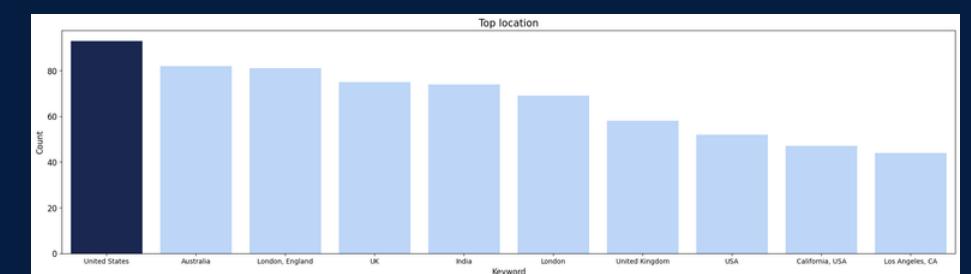
[219 rows x 2 columns]

eksplorasi data untuk menghitung jumlah data yang terkait dengan setiap kata kunci (keyword) dalam dataset menggunakan teknik-teknik yang umum digunakan dalam pengolahan bahasa alami (Natural Language Processing - NLP).

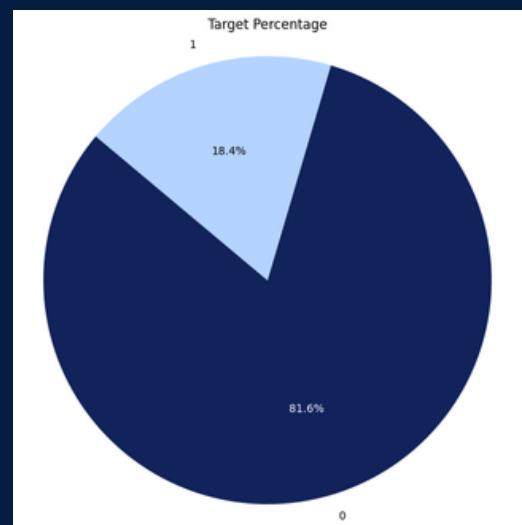
```
# visualizing the keyword_count
```



```
# exploration to count the amount  
of data for each location
```



```
# visualizing the target_count
```



MACHINE LEARNING PROCESS

PREPROCESSING

word preprocessing

```
def preprocess(text):
    # Check if the input is a string
    if isinstance(text, str):
        text = text.lower()
        text = re.sub(r'[^a-zA-Z0-9\s]', '', text)
        text = re.sub(r'https://\S+', '', text)
        text = re.sub(r'#', '', text)
        return text
    else:
        # If not a string
        return str(text)
```

tokenizing

```
# Tokenizing
nltk.download('punkt')
df['tokens'] = df['clean_text'].apply(nltk.word_tokenize)
```

stop words

```
# Function to remove stop words
def remove_stopwords(df, column_name):
    # Apply the stop word removal to each row in the specified column
    df['cleaned_column'] = df[column_name].apply(lambda tokens: [word for word in tokens if word.lower() not in stop_words])

    return df
df = remove_stopwords(df, 'tokens')
```

result

				text	target	clean_text	tokens	cleaned_column
0	0			Communal violence in Bhainsa, Telangana. "Ston...	1	communal violence in bhainsa telangana stones ...	[communal, violence, in, bhainsa, telangana, s...	[communal, violence, bhainsa, telangana, stone...
1	1			Telangana: Section 144 has been imposed in Bha...	1	telangana section 144 has been imposed in bhai...	[telangana, section, 144, has, been, imposed, ...	[telangana, section, 144, imposed, bhainsa, ja...
2	2			Arsonist sets cars ablaze at dealership https:...	1	arsonist sets cars ablaze at dealership httpst...	[arsonist, sets, cars, ablaze, at, dealership, ht...	[arsonist, sets, cars, ablaze, dealership, htt...
3	3			Arsonist sets cars ablaze at dealership https:...	1	arsonist sets cars ablaze at dealership httpst...	[arsonist, sets, cars, ablaze, at, dealership, ht...	[arsonist, sets, cars, ablaze, dealership, htt...
4	4			"Lord Jesus, your love brings freedom and pard...	0	lord jesus your love brings freedom and pardon...	[lord, jesus, your, love, brings, freedom, freedom, and...	[lord, jesus, love, brings, freedom, pardon, f...
...
11365	11365			Media should have warned us well in advance. T...	0	media should have warned us well in advance th...	[media, should, have, warned, us, well, advance, well, in, ad...	[media, warned, us, well, advance, wrecked, wh...
11366	11366			i feel directly attacked 💀 i consider moonbin ...	0	i feel directly attacked i consider moonbin a...	[i, feel, directly, attacked, i, consider, moonbin, a...	[feel, directly, attacked, consider, moonbin, ...
11367	11367			i feel directly attacked 💀 i consider moonbin ...	0	i feel directly attacked i consider moonbin a...	[i, feel, directly, attacked, i, consider, moonbin, a...	[feel, directly, attacked, consider, moonbin, ...
11368	11368			ok who remember "outcast" nd the "dora" au?? T...	0	ok who remember outcast nd the dora au those a...	[ok, who, remember, outcast, nd, the, dora, au, au, wrec...	[ok, remember, outcast, nd, dora, au, au, wrec...

Pada proses preprocessing, data subyek melalui tahap penting pengolahan untuk meningkatkan kualitasnya. Ini melibatkan penghapusan tanda baca, link, dan hashtag. Selain itu, dilakukan tokenization untuk memecah data menjadi unit kecil, dan stop words, kata-kata umum yang kurang relevan, dihapus. Dengan langkah-langkah ini, data menjadi lebih bersih dan siap untuk dianalisis oleh algoritma identifikasi otomatis dengan lebih efisien.

MACHINE LEARNING PROCESS

SPLITTING & VECTORIZING

splitting

```
x = df['joined_column']
y = df['target']
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, stratify=y)
```

vectorizing

```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf= TfidfVectorizer()
X_train = tfidf.fit_transform(X_train)
X_test = tfidf.transform(X_test)
```

Setelah preprocessing, langkah vectorizing mengubah data teks menjadi representasi vektor numerik, memungkinkan algoritma memproses informasi. Selanjutnya, proses splitting membagi data menjadi pelatihan (training set) dan pengujian (testing set) untuk melatih dan menguji model identifikasi otomatis. Dengan langkah ini, model dapat belajar dan dinilai dengan baik pada data yang berbeda.

MACHINE LEARNING PROCESS METRICS

Metrics Function

```
from sklearn.metrics import accuracy_score, roc_auc_score, roc_curve, f1_score
def metrics(y_true, y_pred):
    # print('Accuracy Score: ', accuracy_score(y_true, y_pred))
    print('AUC ROC Score: ', roc_auc_score(y_true, y_pred))
    print('F1 Score: ', f1_score(y_true, y_pred))
```

- AUC ROC: mengukur kualitas dari model klasifikasi dalam memisahkan kelas positif dan negatif.
- F1 Score: ukuran yang mencakup baik recall maupun precision dalam satu nilai.

Confusion Matrix Function

Confusion matrix digunakan sebagai alat untuk evaluasi kinerja model klasifikasi.

```
from sklearn.metrics import confusion_matrix
def cfmatrix(y_true, y_pred):
    confusion_matrix_df = pd.DataFrame((confusion_matrix(y_true, y_pred)), ('Non-Disaster', 'Disaster'), ('Non-Disaster', 'Disaster'))

    plt.figure()
    heatmap = sns.heatmap(confusion_matrix_df, annot=True, annot_kws={'size': 14}, fmt='d', cmap='YlGnBu')
    heatmap.yaxis.set_ticklabels(heatmap.yaxis.get_ticklabels(), rotation=0, fontsize=14)
    heatmap.yaxis.set_ticklabels(heatmap.xaxis.get_ticklabels(), rotation=0, fontsize=14)

    plt.title('Confusion Matrix', fontsize=18, color='darkblue')
    plt.ylabel('True label', fontsize=14)
    plt.xlabel('Predicted label', fontsize=14)
    plt.show()
```

MACHINE LEARNING PROCESS

MODELS USED

Logistic Regression

Model: Algoritma statistik yang memprediksi probabilitas suatu peristiwa terjadi

Kelebihan: Mudah dipahami, stabil, tahan outlier.

Kekurangan: Membutuhkan data besar, sensitif terhadap multikolinearitas, kurang akurat untuk data kompleks.

```
[ ] from sklearn.linear_model import LogisticRegression

# Fitting
logreg = LogisticRegression()
logreg.fit(x_train, y_train)

# Prediction
y_pred_train = logreg.predict(x_train)
y_pred_test = logreg.predict(x_test)
```

Random Forest Classifier

Model: Algoritma ensemble learning yang menggabungkan beberapa pohon keputusan untuk meningkatkan akurasi.

Kelebihan: Akurasi tinggi, menangani data kompleks, mudah diinterpretasikan.

Kekurangan: Kompleksitas model, membutuhkan data besar, lama pelatihan.

```
[ ] from sklearn.ensemble import RandomForestClassifier
# Fitting
random_forest = RandomForestClassifier(max_depth = 10, n_estimators = 50, class_weight = 'balanced')
random_forest.fit(x_train, y_train)

# Prediction
y_pred_train = random_forest.predict(x_train)
y_pred_test = random_forest.predict(x_test)
```

SVM

Model: Algoritma klasifikasi dan regresi yang mencari hyperplane dengan margin maksimum.

Kelebihan: Akurasi tinggi, menangani data nonlinear, efisien dalam dimensi tinggi.

Kekurangan: Lama pelatihan, sensitif terhadap outlier, sulit diinterpretasikan.

```
▶ from sklearn.model_selection import GridSearchCV
from sklearn.svm import SVC

# Define the parameter grid
param_grid = {
    'C': [0.1, 1, 10],
    'gamma': [0.01, 0.1, 1],
    'kernel': ['linear', 'rbf']
}

# Create an SVM model
svm_model = SVC()

# Create GridSearchCV instance
grid_search = GridSearchCV(svm_model, param_grid, cv=3, scoring='roc_auc', verbose=2, n_jobs=-1)

# Fit the model to the data
grid_search.fit(x_train, y_train)

# Print the best parameters and corresponding accuracy
print("Best Parameters: ", grid_search.best_params_)
print("Best ROC AUC: {:.2f}%".format(grid_search.best_score_ * 100))

# Evaluate the best model on the test set
best_model = grid_search.best_estimator_
test_accuracy = best_model.score(x_test, y_test)
print("Test ROC AUC of Best Model: {:.2f}%".format(test_accuracy * 100))
```

```
[ ] # Fitting
svm_best = SVC(C=10.0, gamma=1, kernel='rbf')
svm_best.fit(x_train, y_train)

# Prediction
y_pred_train = svm_best.predict(x_train)
y_pred_test = svm_best.predict(x_test)
```

MACHINE LEARNING PROCESS

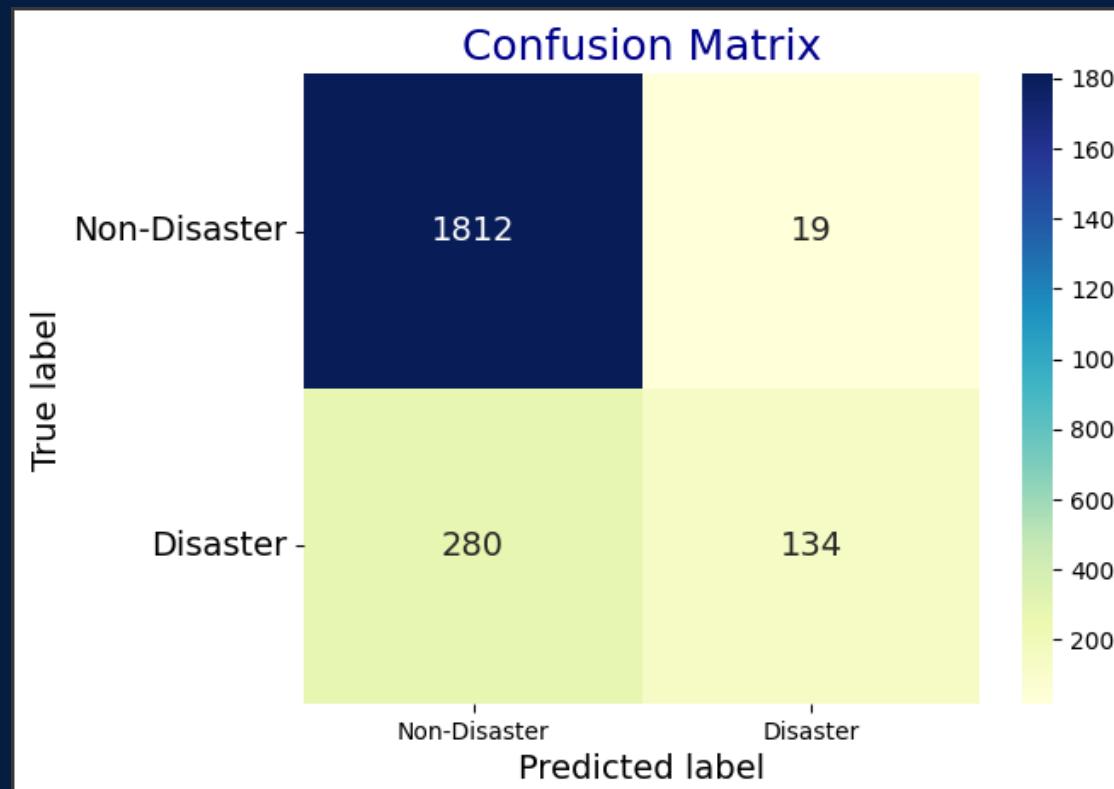
MODELS INSIGHT

Logistic Regression

Metrics for Train Dataset
AUC ROC Score: 0.70701823958723
F1 Score: 0.5841458245019076

Metrics for Test Dataset
AUC ROC Score: 0.6566473271647446
F1 Score: 0.47266313932980597

Hasilnya lumayan, nilai False Positive kecil

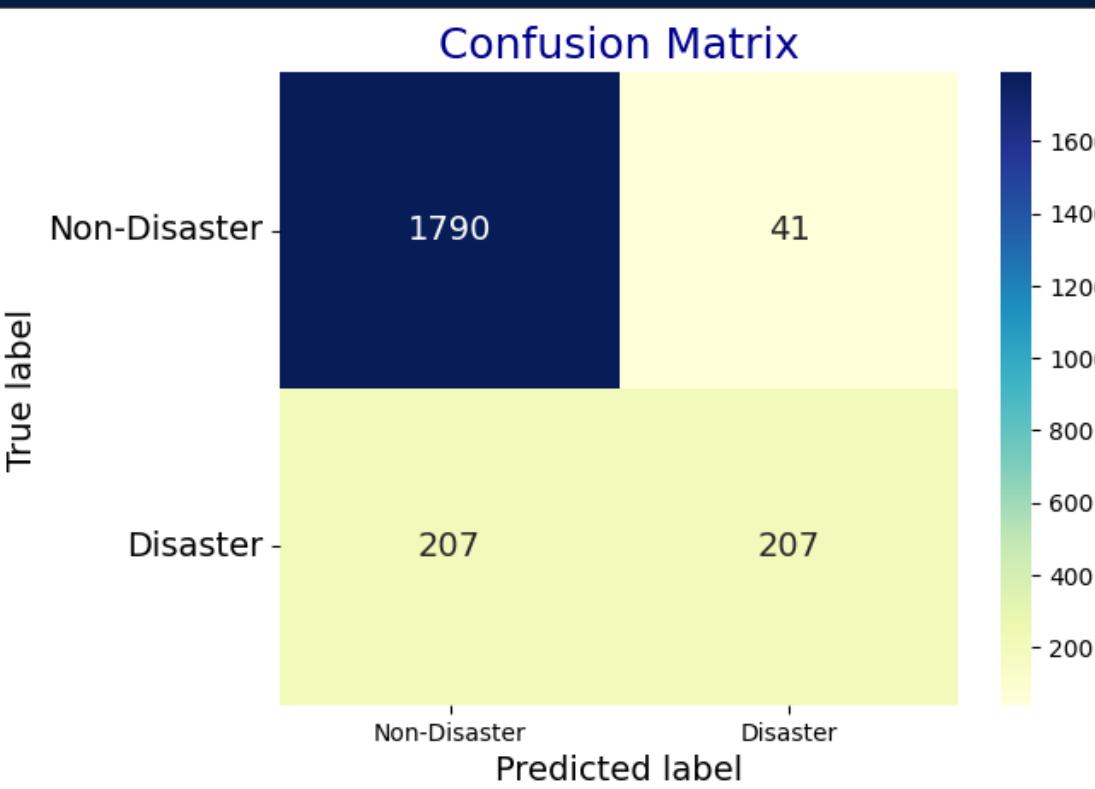


SVM

Metrics for Train Dataset
AUC ROC Score: 0.9993964996982498
F1 Score: 0.9993961352657005

Metrics for Test Dataset
AUC ROC Score: 0.738803932277444
F1 Score: 0.6253776435045317

Terjadi overfitting pada model

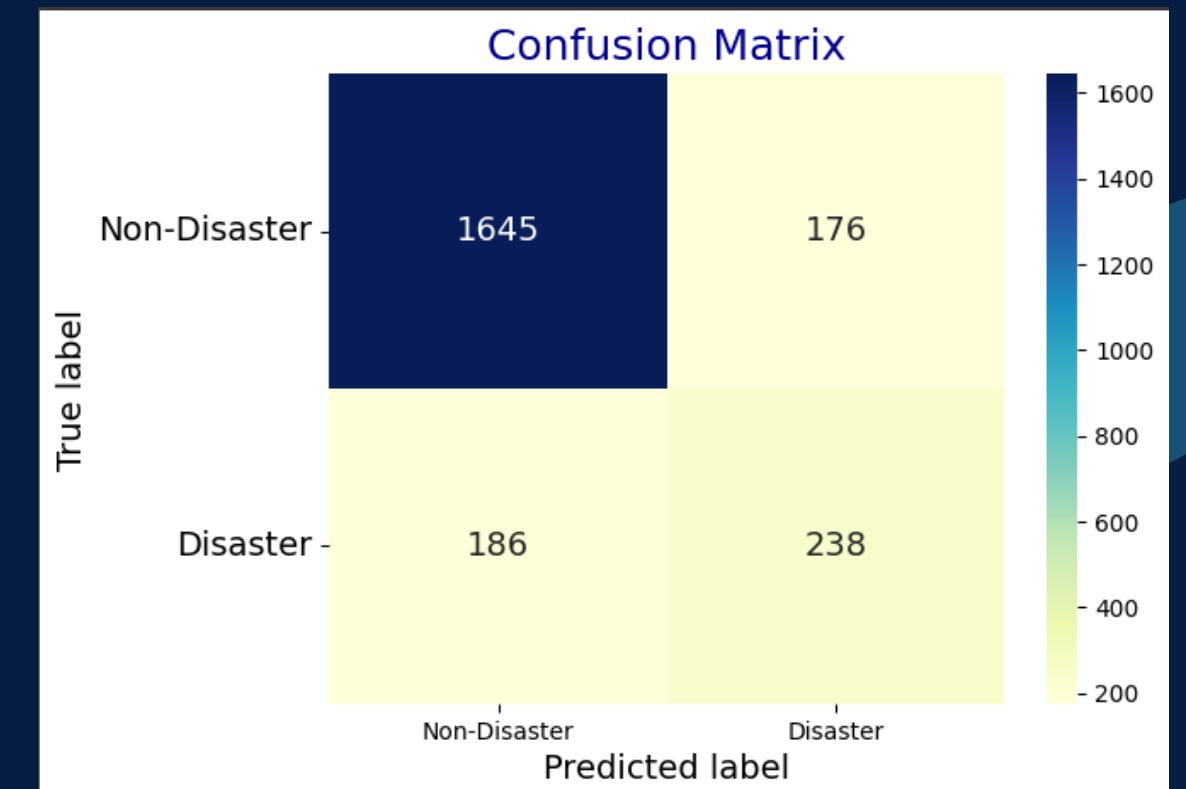


Random Forest Classifier

Metrics for Train Dataset
AUC ROC Score: 0.7996493962401017
F1 Score: 0.6897868278714603

Metrics for Test Dataset
AUC ROC Score: 0.736647696541316
F1 Score: 0.5680190930787591

Tidak ada over/underfit



MACHINE LEARNING PROCESS

K-FOLD CROSS VALIDATION

```
models = []
models.append('Logistic Regression', logreg)
models.append('SVM', svm_best)
models.append('Random Forest Classifier', random_forest))

from sklearn import model_selection
acc_score=[]
auc_score=[]
pre_results=[]
f1_results=[]
names=[]
result_col=pd.DataFrame(columns=['Algorithm','recall','ROC','Precision','f1_score'])
result_col

i=0
for name, model in models:
    kfold=model_selection.KFold(n_splits=10)
    names.append(name)
    cv_acc_results = model_selection.cross_val_score(model, X_train, y_train, cv=kfold, scoring="recall")
    cv_auc_results = model_selection.cross_val_score(model, X_train, y_train, cv=kfold, scoring="roc_auc")
    cv_pre_results = model_selection.cross_val_score(model, X_train, y_train, cv=kfold, scoring="precision")
    cv_f1_results = model_selection.cross_val_score(model, X_train, y_train, cv=kfold, scoring="f1")
    acc_score.append(cv_acc_results)
    auc_score.append(cv_auc_results)
    pre_results.append(cv_pre_results)
    f1_results.append(cv_f1_results)

    result_col.loc[i]=[name,round(cv_acc_results.mean()*100,2),
                      round(cv_auc_results.mean()*100,2),
                      round(cv_pre_results.mean()*100,2),
                      round(cv_f1_results.mean()*100,2)]
    i+=1

result_col.sort_values(by=['ROC'],ascending=False)
```

Result for x_train, y_train

	Algorithm	recall	ROC	Precision	f1_score
1	SVM	45.91	90.52	89.66	60.64
0	Logistic Regression	25.91	89.85	92.65	40.40
2	Random Forest Classifier	50.55	81.99	59.82	57.16

Model SVM memiliki nilai ROC tertinggi

Result for x_test, y_test

	Algorithm	recall	ROC	Precision	f1_score
1	SVM	22.22	84.85	93.47	35.53
0	Logistic Regression	7.75	84.57	100.00	14.08
2	Random Forest Classifier	28.44	73.79	69.60	42.47

IMPROVISASI



- Pengembangan teknologi untuk bisa mendekripsi dalam bahasa indonesia.
- Pembaruan dataset secara berkala.
- Pengembangan dan pelatihan model machine learning lebih lanjut.
- Implementasi teknologi pada platform X.

- Adanya imbalance data
- Dataset hanya tersedia dalam bahasa inggris
- Nilai yang berbeda jauh untuk metrik akurasi recall dan f1 score dengan ROC dan precision



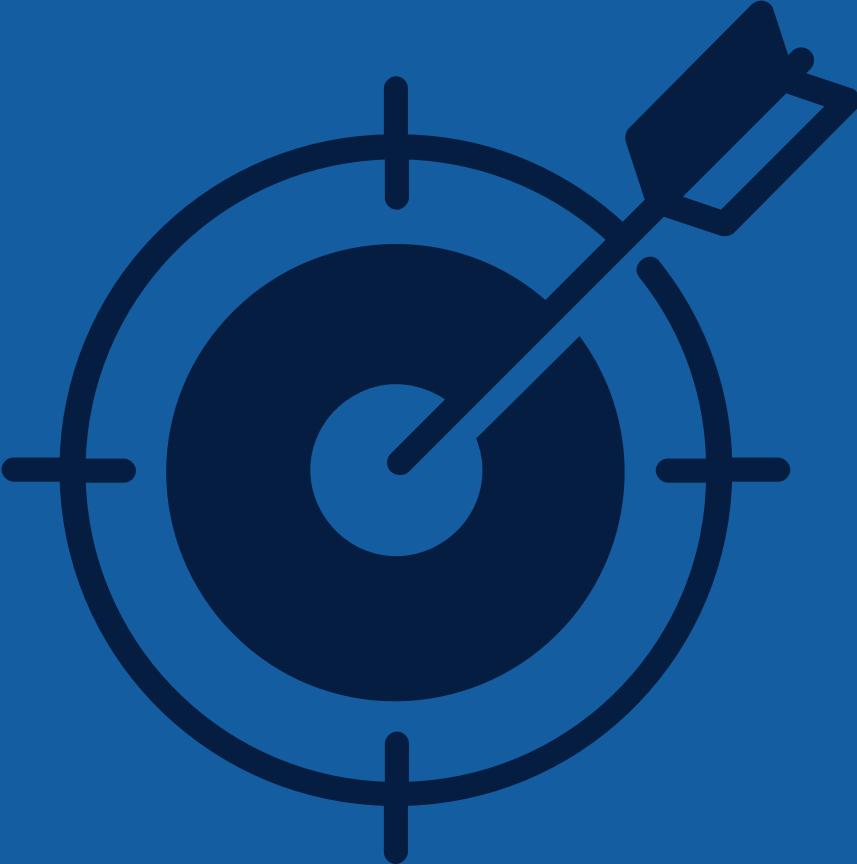
KENDALA

IMPLEMENTATION STRATEGY



Notifikasi Darurat

Setiap tweet yang terdeteksi sebagai tweet bencana dapat dikirimkan sebagai pemberitahuan otomatis kepada pihak berwenang atau tim penanggulangan bencana sehingga penanganan pasca bencana dapat berlangsung lebih cepat dan efektif.



Optimalisasi Informasi

Tweet dengan label non-disaster akan otomatis tertimbun atau terhapus dan tweet disaster akan naik sehingga informasi mengenai bencana dapat tersampaikan dengan baik ketika dilakukan pencarian berdasarkan keyword.



Analisis Tren Bencana

Memanfaatkan data historis terkait tweet bencana untuk menganalisis tren dan mengembangkannya untuk prediksi ke depan.



Pencegahan Hoax

Dalam pengembangan lebih lanjut, X dapat mengintegrasikan teknologi ini ke dalam platform dan memberi label disaster atau non-disaster sehingga pembaca dapat menghindari informasi palsu.

Conclusion

- Twitter merupakan salah satu media sosial paling populer yang biasa digunakan untuk melaporkan kejadian bencana
- Sayangnya banyak konten tidak relevan yang dapat mengganggu proses pelaporan bencana
- Sistem berbasis NLP yang kami ajukan dapat membantu mengklasifikasikan apakah sebuah tweet merupakan tweet yang relevan dengan bencana atau tidak relevan
- Namun sistem ini masih belum sempurna dan ke depannya perlu dikembangkan lebih lanjut

Thank You

Colab Notebook

