

KLASIFIKASI TWEET BENCANA

KELOMPOK 5



DISASTER DETECTION SQUAD

**When chaos tweets, these are the people who read
between the flames:**

1

Ahmad Izza
11220940000006

2

Dani Hidayat
11220940000014

MISSION: DISASTER CLASSIFIED

1

**Exploratory Data
Analysis**

2

Preprocessing

3

Vectorization

4

Modeling:
Naive Bayes
Logistic Regression

5

Model Interpretability:
Analisis kesalahan klasifikasi
Named Entity Extraction
POS-based Feature Scoring

PROBLEM STATEMENT



THE PROBLEM

- Twitter kini menjadi salah satu sumber utama informasi saat terjadi bencana.
- Masyarakat bisa melaporkan kejadian secara **real-time** melalui tweet.
- Namun, tidak semua tweet dengan kata seperti "**ablaze**" benar-benar menggambarkan bencana, bisa jadi hanya **metafora**.
- **Contoh:**
 - 🔥 "The concert was ablaze!" → **bukan bencana**
 - 🔥 "The entire forest is ablaze." → **bencana**
- **Tujuan:** Membangun model (**Naive Bayes & Logistic Regression**) yang dapat mendeteksi apakah sebuah tweet benar-benar mengandung informasi bencana atau tidak.

DATASET

Dataset yang digunakan berasal dari Kaggle, berisi **10.876** tweet berbahasa Inggris yang terbagi menjadi **3.263** data test dan **7.613** data train yang telah diklasifikasi secara manual.

Label:



1: Tweet terkait bencana nyata

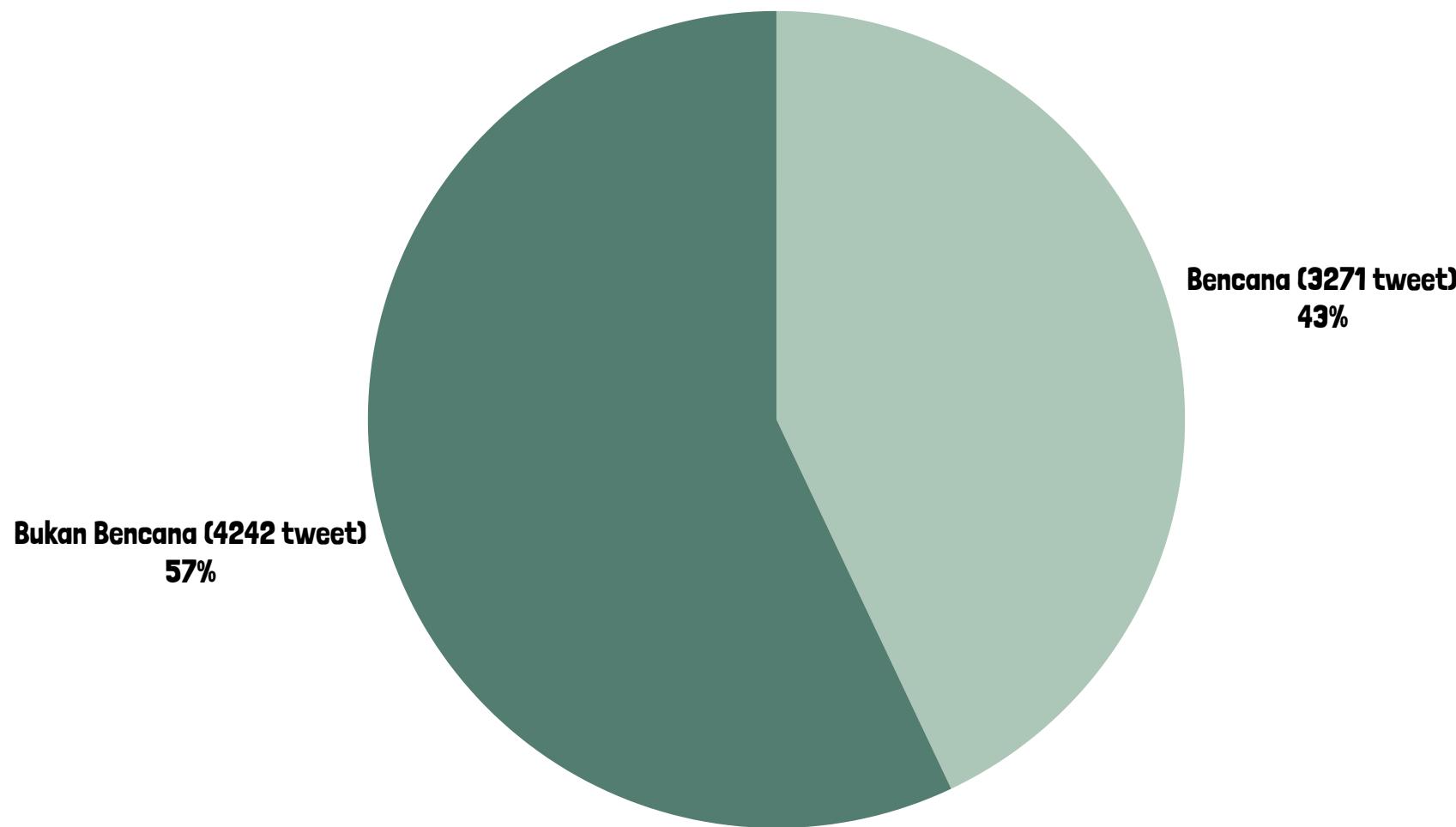


0: Tweet tidak terkait bencana



EXPLORATORY DATA

DISTRIBUSI TWEET



Contoh Tweet

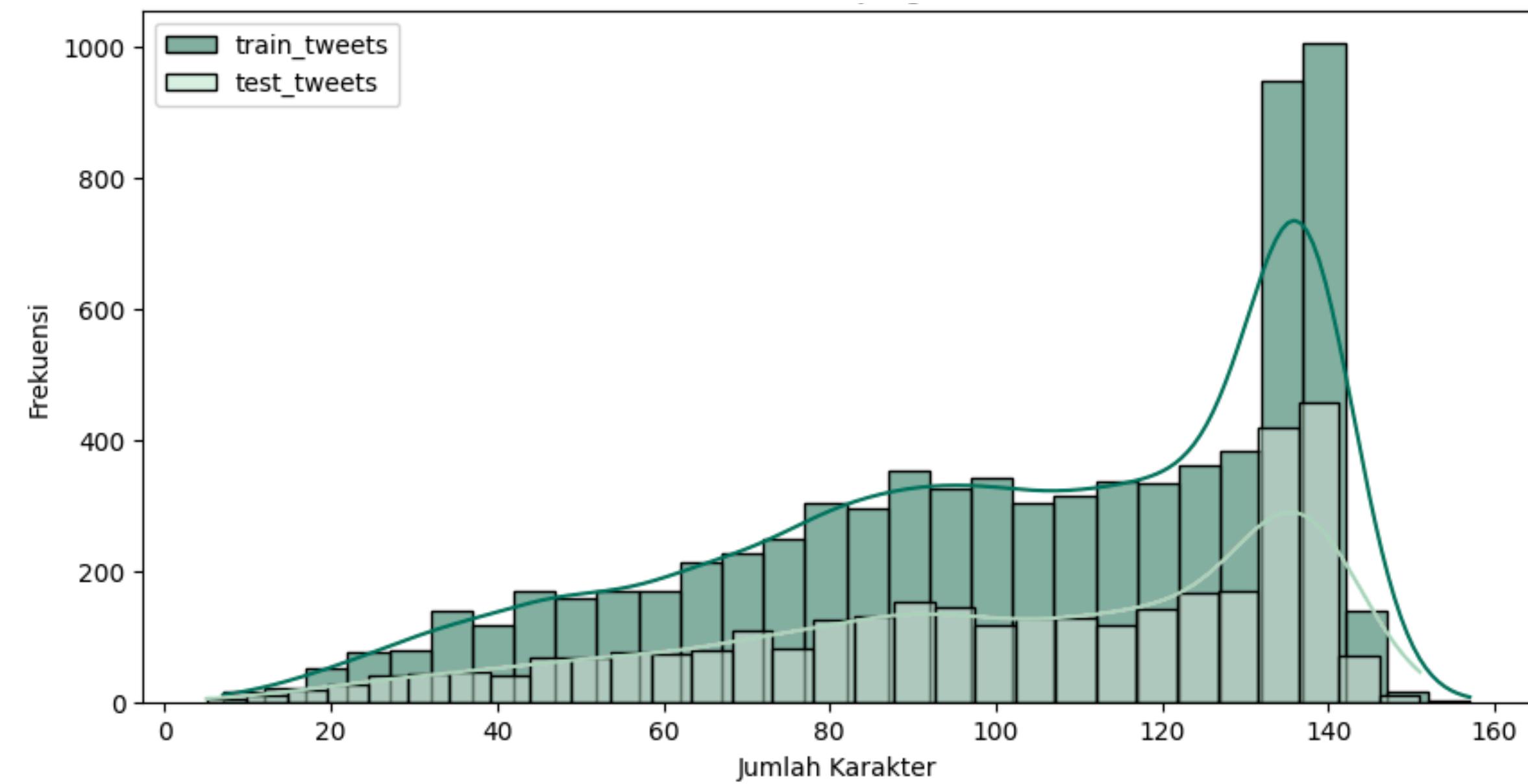
Bukan tentang bencana:

- "**Total tweet fail! You are so beautiful inside and out Blaze On!**"
- "**Shark boy and lava girl for the third time today..."**
- "**My brains going to explode i need to leave this house..."**

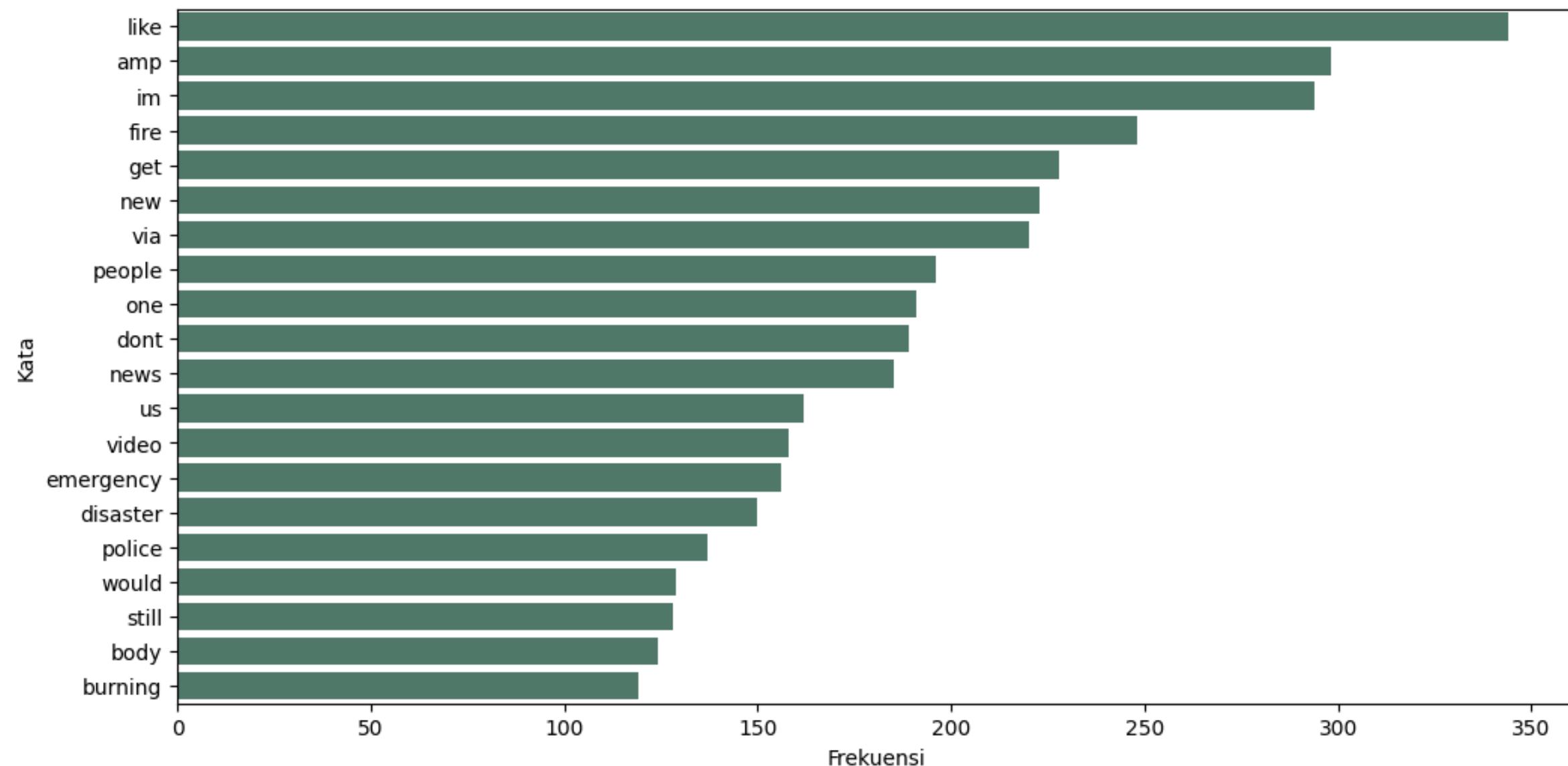
Tentang bencana:

- "**Forest fire near La Ronge Sask. Canada**"
- "**70 Years After Atomic Bombs Japan Still Struggles...**"
- "**Russian 'food crematoria' provoke outrage...**"

DISTRIBUSI PANJANG TWEET



KATA PALING SERING MUNCUL





DATA PREPROCESSING

LANGKAH-LANGKAH PREPOCESSING

1

**Mengubah semua teks
menjadi huruf kecil
(lowercase)**

2

**Menghapus URL, tanda
baca, angka, dan
karakter khusus**

3

**Menghapus stopwords
(kata umum yang tidak
bermakna)**

4

**Melakukan lemmatization
(mengubah kata ke
bentuk dasar)**

5

**Tokenisasi (memecah
teks
menjadi kata-kata)**

CONTOH TWEET SEBELUM & SESUDAH PREPROCESSING:

BUKAN BENCANA

Tweet Asli: Had an awesome time visiting the CFC head office the ancop site and ablaze. Thanks to Tita Vida for taking care of us ??

Clean Tweet: awesome time visiting cfc head office ancop site ablaze thanks tita vida taking care u

BENCANA

Tweet Asli: Our Deeds are the Reason of this #earthquake May ALLAH Forgive us all

Clean Tweet: deed reason earthquake may allah forgive u

TEXT VECTORIZATION

TFIDFVECTORIZER

	fire	like	im	get	amp	via	new	one	dont	people
0	0.00000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000
1	0.22082	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000
2	0.00000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000
3	0.00000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.305331
4	0.00000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000

DENGAN PENGANTURAN:

- **ngram_range=(1, 2):**
 - Artinya: model akan mempertimbangkan **unigram** (kata tunggal) dan **bigram** (2 kata berurutan).
- **max_features=10.000:**
 - hanya akan menyimpan maksimal **10.000 kata/fitur** paling penting berdasarkan frekuensi di seluruh dokumen.

TF-IDF digunakan untuk mengubah teks menjadi fitur numerik berbobot, agar bisa digunakan oleh model machine learning.

FEATURE SELECTION

Metode: Chi-Square (SelectKBest)

Jumlah fitur terbaik dipilih: 3000 dari 10.000

Tujuan:

menghindari curse of dimensionality

Mengurangi fitur tidak relevan

Meningkatkan efisiensi dan performa model

DATA SPLITTING

Setelah proses Feature Selection (SelectKBest), data dibagi menjadi:

- **80% data latih → X_train, y_train**
- **20% data uji → X_test, y_test**
- **Menggunakan stratify=y untuk menjaga distribusi label tetap seimbang**

Note:

- **Data yang digunakan hanya berasal dari train (dataset pelatihan dari Kaggle).**
- **Dataset test dari Kaggle tidak digunakan dalam pelatihan maupun evaluasi karena tidak memiliki label target.**
- **Evaluasi model hanya dilakukan pada data uji hasil pembagian dari data train.**



TRAINING & EVALUATION

LOGISTIC REGRESSION



EVALUASI BASE MODEL

Metric	Kelas 0 (Bukan Bencana)	Kelas 1 (Bencana)
Precision	0.78	0.85
Recall	0.91	0.66
F1-Score	0.84	0.74

- **Accuracy: 80%**
- **Model cenderung lebih baik mengenali non-bencana.**

HYPERPARAMETER TUNING

Menggunakan GridSearchCV dengan 5-fold cross-validation.

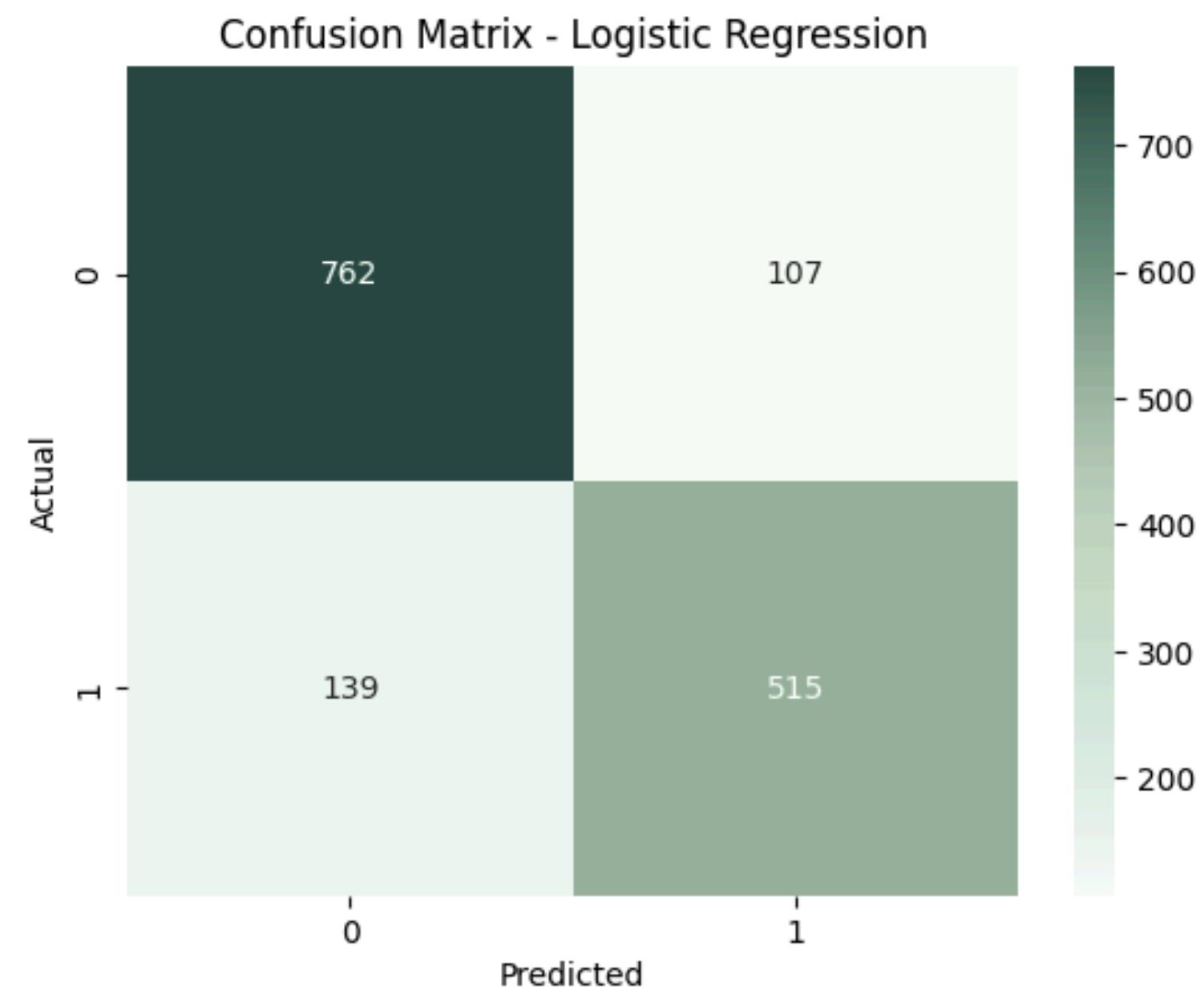
Parameter	Nilai yang Dicoba	Penjelasan
Penalty	l1, l2, elasticnet	penalty atau regularisasi digunakan untuk mencegah overfitting dengan menambahkan “penalty” terhadap kompleksitas model, khususnya terhadap nilai besar dari koefisien (parameter) model.
C	0.01, 0.1, 1, 10, 100	Merupakan kebalikan dari kekuatan regularisasi. Nilai C yang kecil berarti regularisasi kuat (model lebih sederhana untuk mencegah overfitting), sedangkan nilai C yang besar berarti regularisasi lemah (model lebih kompleks).
Solver	'liblinear', 'lbfgs', 'newton-cg', 'newton-cholesky', 'sag', 'saga'	adalah metode numerik yang digunakan untuk menyelesaikan optimisasi fungsi loss

EVALUASI BEST MODEL

- Parameter terbaik:
 - **C=1, penalty="l2", solver='liblinear'**
- Test Accuracy: **84%**
- Tuning berhasil menyeimbangkan kinerja kedua kelas (disaster & non-disaster).

Metric	Kelas 0 (Bukan Bencana)	Kelas 1 (Bencana)
Precision	0.85	0.83
Recall	0.88	0.79
F1-Score	0.86	0.81

CONFUSION MATRIX - BEST MODEL





TRAINING & EVALUATION

NAIVE BAYES

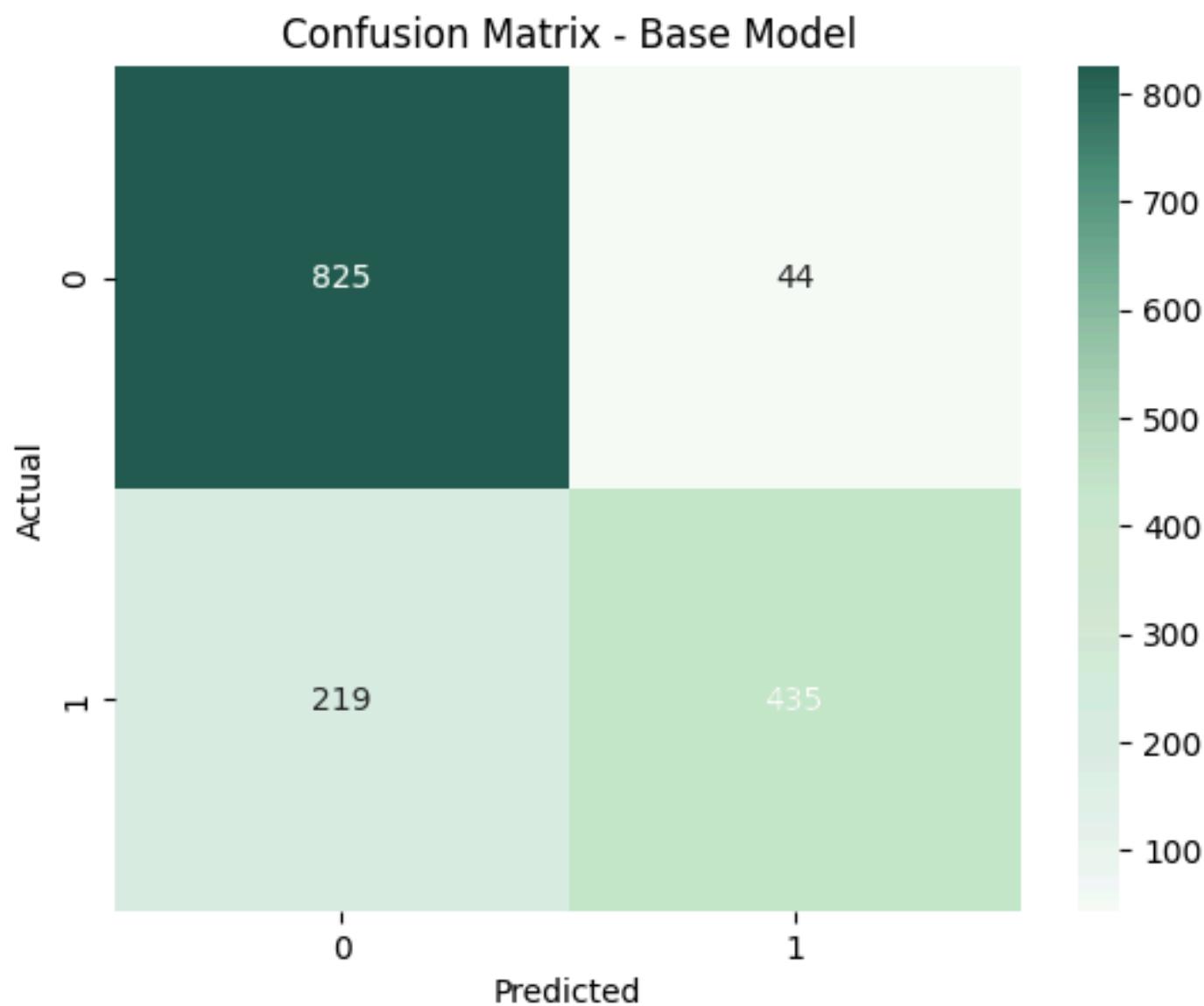


EVALUASI BASE MODEL

Metric	Kelas 0 (Bukan Bencana)	Kelas 1 (Bencana)
Precision	0.79	0.91
Recall	0.95	0.67
F1-Score	0.86	0.77

- **Accuracy: 82.7%**
- **Model cenderung lebih baik mengenali non-bencana, tapi banyak false negative untuk bencana.**

CONFUSION MATRIX - BASE MODEL



HYPERPARAMETER TUNING

Menggunakan GridSearchCV dengan 5-fold cross-validation.

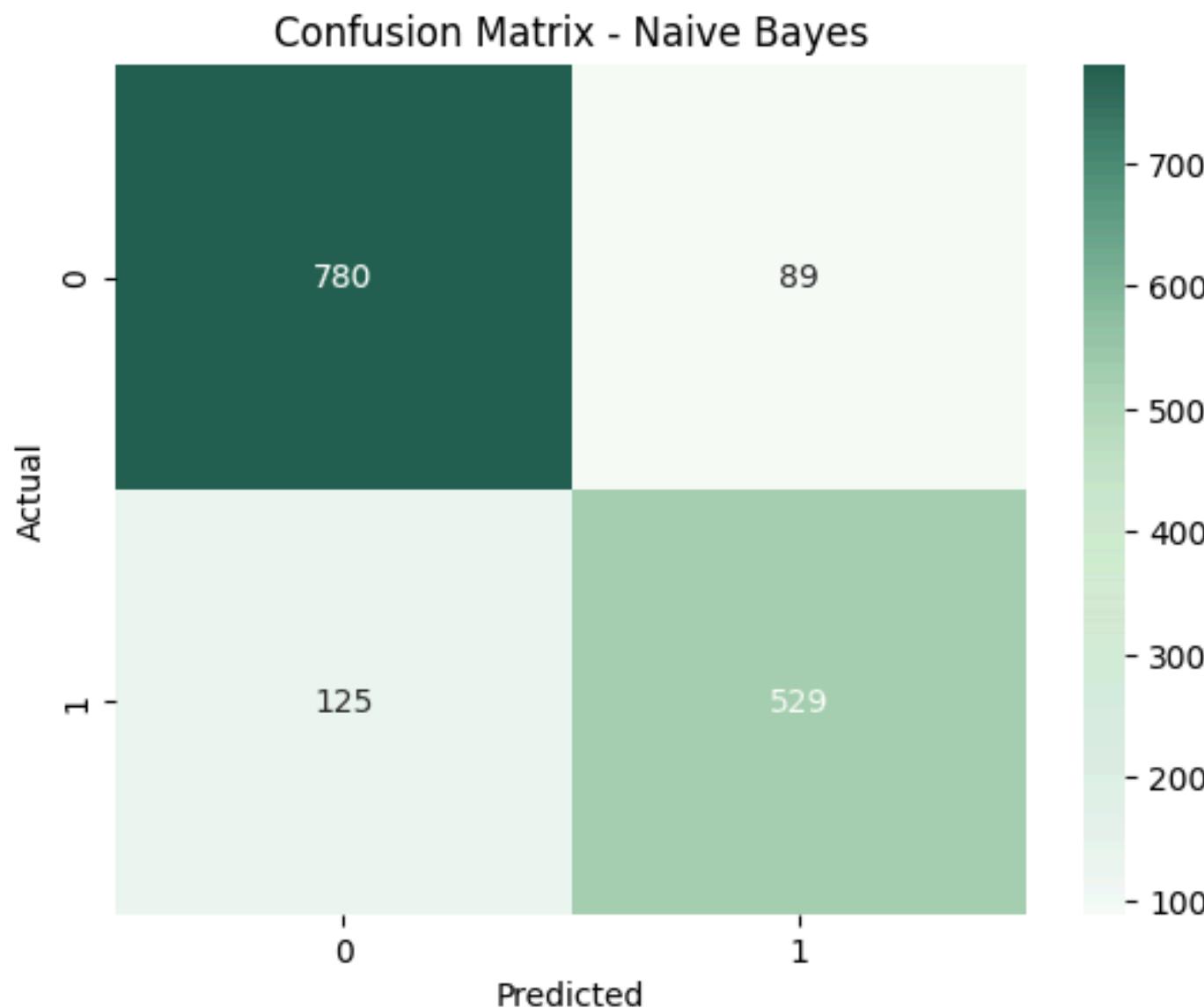
Parameter	Nilai yang Dicoba	Penjelasan
alpha	0.01, 0.1, 0.5, 1.0, 2.0	Parameter smoothing. Nilai yang lebih kecil membuat model lebih tajam terhadap perbedaan fitur antar kelas, sedangkan nilai besar membuat prediksi lebih general.
fit_prior	True, False	Jika True, model belajar proporsi kelas dari data. Jika False, tidak.
class_prior	[0.5,0.5], [0.7,0.3], [0.9,0.1] (hanya jika fit_prior=True)	Proporsi manual untuk kelas 0 dan 1, misal memberi bobot lebih ke bencana.

EVALUASI BEST MODEL

- Parameter terbaik:
 - **alpha = 0.01, fit_prior = True, class_prior = [0.5, 0.5]**
- Test Accuracy: **85.9%**
- Tuning berhasil menyeimbangkan kinerja kedua kelas (disaster & non-disaster).

Metric	Kelas 0 (Bukan Bencana)	Kelas 1 (Bencana)
Precision	0.86	0.86
Recall	0.90	0.81
F1-Score	0.88	0.83

CONFUSION MATRIX - BEST MODEL



ANALISIS KESALAHAN KLASIFIKASI

ANALISIS KESALAHAN KLASIFIKASI – NAIVE BAYES

SEHARUSNYA BENCANA, DIPREDIKSI BUKAN

- I waited 2.5 hours to get a cab my feet are bleeding
- @KatRamsland Yes I'm a bleeding heart liberal.
- @beckyfeigin I defs will when it stops bleeding!
- @Benjm1 @TourofUtah @B1Grego saw that pileup on TV
keep racing even bleeding
- Watch how bad that fool get burned in coverage this
year. Dat dude is all-pro practice squad material
- Destruction magic's fine just don't go burning down
any buildings.
- @zourryart I forgot to add the burning buildings and
screaming babies
- into fucking buildings (2/2)

SEHARUSNYA BUKAN BENCANA, DIPREDIKSI BENCANA

- Ali you flew planes and ran into burning buildings why
are you making soup for that man child?!
#BooRadleyVanCullen
- if firefighters acted like cops they'd drive around
shooting a flamethrower at burning buildings
- I crashed my car into a parked car the other day...
- Had a minute alone with my crush??...it was an
overrated experience...smh
- Ari's hints and snippets will be the death of me.
- tomorrow will be the death of me
- @johndcgow heard this few days ago while driving and
near crashed the car from laughing to much

ANALISIS KESALAHAN KLASIFIKASI – LOGISTIC REG.

SEHARUSNYA BENCANA, DIPREDIKSI BUKAN

- it wasnt a very big stab but it was a deep stab and theres like blood everywhere
- If a truckload of soldiers will be blown up nobody panics but when one little lion dies everyone loses their mind
- Even though BSG had been sufficiently hyped up for me in all the years I somehow delayed watching it I was utterly utterly blown away.
- drake killing this dude and tea bagging the dead body at this point

SEHARUSNYA BUKAN BENCANA, DIPREDIKSI BENCANA

- I crashed my car into a parked car the other day...
- VIDEO: Slain Mexican Journalist Unknowingly Predicted His Own Death
- Article by Michael Jackman at Metro Times Detroit
- Ashes 2015: Australia Û's collapse at Trent Bridge among worst in history: England bundled out Australia for 60 ...
- Ali you flew planes and ran into burning buildings why are you making soup for that man child?!
- The Burning Legion has RETURNED!



DETEKSI LOKASI, WAKTU, DAN ORGANISASI



POS TAGGING – PROPER NOUNS EXTRACTION

- Menggunakan POS Tagging untuk mengekstrak kata benda khusus (Proper Noun: NNP, NNPS) dari tweet yang diklasifikasikan sebagai bencana.
- Fokus: Tweet dengan target = 1 (berisi informasi tentang bencana).
- Tokenisasi + POS Tagging + Ambil Proper Noun:

```
def extract_proper_nouns(text):  
    tokens = nltk.word_tokenize(text)  
    tagged = nltk.pos_tag(tokens)  
    proper_nouns = [word for word, tag in tagged if tag in ['NNP', 'NNPS']]  
    return proper_nouns  
  
df_POS = disaster_df.copy()  
df_POS['proper_nouns'] = df_POS['text'].apply(extract_proper_nouns)
```

- Hitung frekuensi proper noun

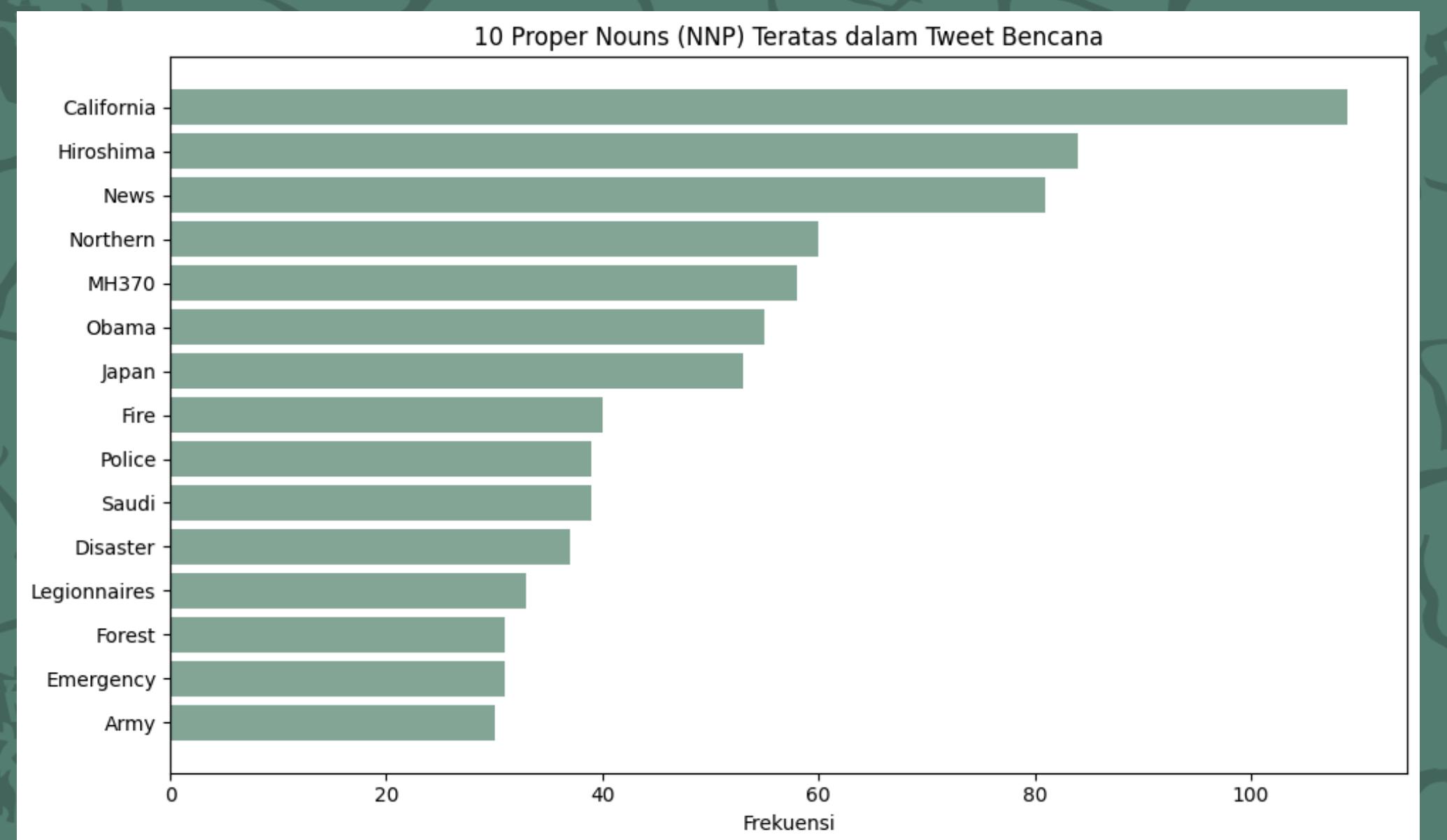
```
from collections import Counter  
  
all_nouns = sum(df_POS['proper_nouns'], []) # Gabungkan semua kata  
noun_counts = Counter(all_nouns)  
  
# Buang kata pendek (<4 huruf)  
for key in list(noun_counts):  
    if len(key) < 4:  
        del noun_counts[key]  
  
top_nouns = noun_counts.most_common(15)
```

- Contoh hasil:

- Tweet: Nearly 50 thousand people affected by floods in Paraguay
- Proper Nouns: [Paraguay]

TOP 10 NNP TWEET BENCANA

(POS TAGGING – PROPER NOUNS EXTRACTION)



- Kata seperti **California**, **Hiroshima**, dan **Japan** paling sering muncul, menunjukkan lokasi kejadian.
- nama tokoh seperti "**Obama**" dan entitas seperti "**MH370**" juga muncul, yang dapat merepresentasikan konteks atau berita besar yang terkait bencana.
- Kehadiran kata seperti "**News**", "**Police**", dan "**Emergency**" menunjukkan bahwa tweet sering mengandung elemen pemberitaan atau respons instansi.

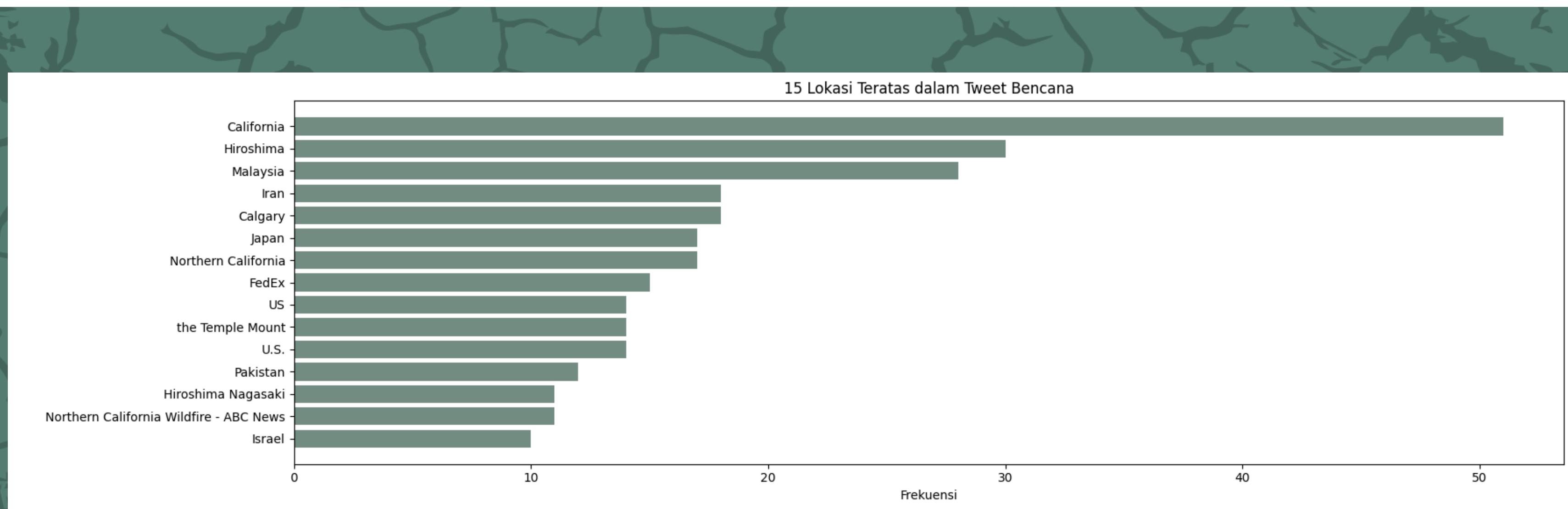
NAMED ENTITY RECOGNITION (NER) – BIO TAGGING

- Menggunakan model spaCy untuk mengenali entitas bernama dalam tweet (lokasi, waktu, organisasi, event).
- Format BIO (Beginning, Inside, Outside) digunakan untuk memberi label setiap token dalam teks.
- Dikembangkan dua fungsi:
 - ner_bio(): Melabeli setiap token dengan informasi entitas (misal: lokasi/waktu/organisasi/event).
 - ner_bio2(): mengekstrak entitas penting → lokasi, waktu, organisasi, event

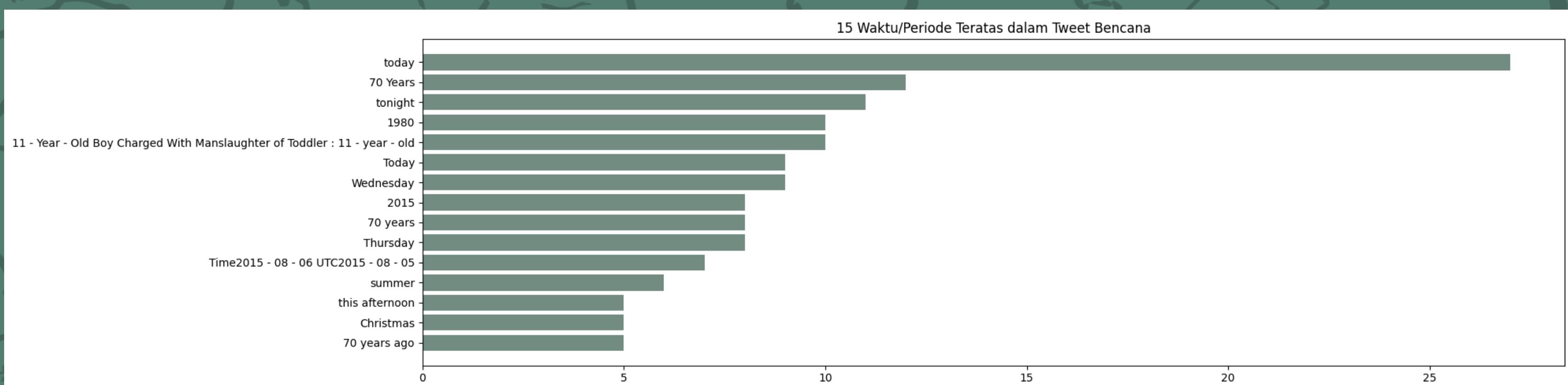
```
def ner_bio(text):  
    doc = nlp(text) # Proses teks dengan spaCy  
    tokens = [token.text for token in doc]  
    tags = []  
    for token in doc:  
        ent = token.ent_iob_ # Ambil tag BIO ('B', 'I', 'O')  
        label = token.ent_type_ # Ambil tipe entitas (GPE, ORG, dll.)  
        if ent == 'O':  
            tags.append('O') # Token bukan entitas  
        else:  
            tags.append(f"{ent}-{label}") # Contoh: B-GPE  
    return list(zip(tokens, tags))
```

```
def ner_bio2(text):  
    doc = nlp(text)  
    # Simpan hasil entitas ke dalam list  
    loc, datetime, org, event = [], [], [], []  
    for token in doc:  
        label = token.ent_type_  
        # Klasifikasi berdasarkan jenis entitas  
        if label in ['GPE', 'LOC']:  
            loc.append(str(token))  
        if label in ['DATE', 'TIME']:  
            datetime.append(str(token))  
        if label == 'ORG':  
            org.append(str(token))  
        if label == 'EVENT':  
            event.append(str(token))  
    # Gabungkan hasil dalam bentuk string  
    return " ".join(loc), " ".join(datetime), " ".join(org), " ".join(event)
```

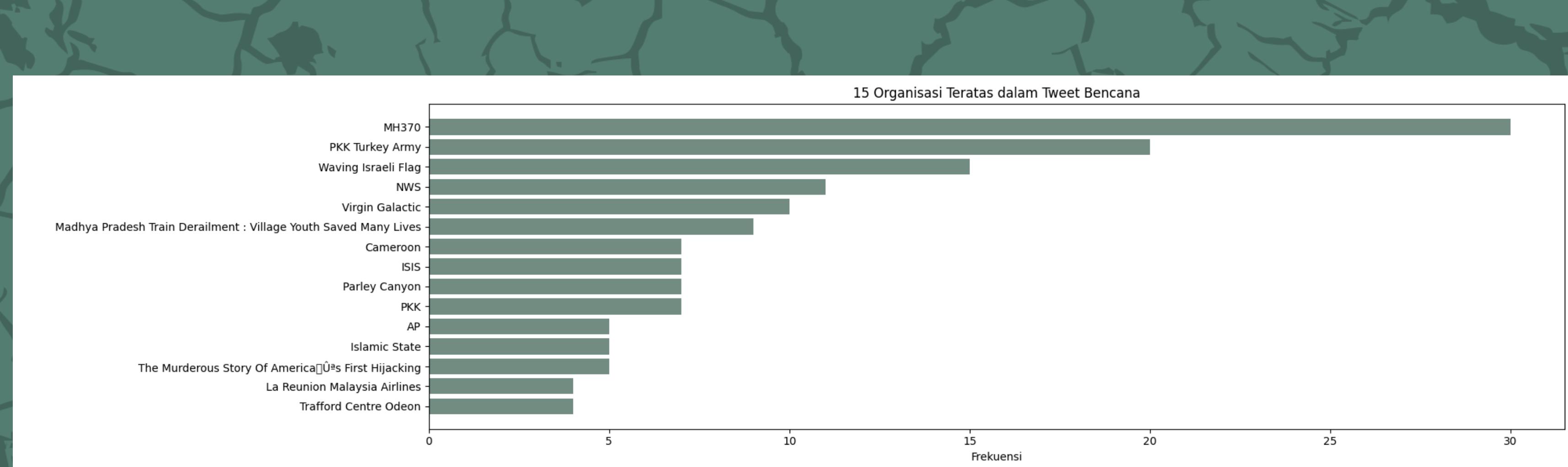
TOP 15 LOKASI TERATAS



TOP 15 WAKTU/PERIODE TERATAS



TOP 15 ORGANISASI TERATAS



INSIGHT

- **Lokasi paling sering disebut:**
 - California, Hiroshima, dan Japan muncul paling dominan dalam tweet bencana.
- **Waktu/Periode penting (dari NER):**
 - Kata seperti today, tonight, dan Wednesday sering digunakan, menandakan urgensi dan kejadian yang baru terjadi.
- **Organisasi/lembaga terkait:**
 - MH370, PKK Turkey Army, dan Virgin Galactic menjadi organisasi atau entitas yang paling sering dikaitkan dengan tweet bencana.
- **Perbandingan pendekatan:**
 - POS Tagging mendeteksi kata benda khusus (NNP) namun bisa kurang kontekstual.
 - NER BIO memberi label yang lebih spesifik seperti lokasi (LOC), waktu (TIME), dan organisasi (ORG), sehingga lebih informatif dan terstruktur.

KESIMPULAN

Metric	Logistic Regression	Naive Bayes
Precision	84%	86%
Recall	83%	85%
F1-Score	83%	86%
Akurasi	83.7%	85.9%

- Model Naive Bayes menunjukkan performa yang lebih unggul dibanding Logistic Regression pada seluruh metrik evaluasi.
- Kedua model membuktikan bahwa metode machine learning masih sangat efektif dan efisien dalam menyelesaikan tugas klasifikasi teks, terutama saat dipadukan dengan feature engineering dan pre-processing yang tepat



THANK YOU!

I HOPE YOU LEARN SOMETHING NEW TODAY!