



Twitter's Sentiment Analysis (Pilpres 2019 Core)

NLP - C
Noam Chomsky

27 September 2024

Meet the Team



Yuan Shafira

Your Role/Contribution

[linkedin.com/in/
yuanshafira](https://linkedin.com/in/yuanshafira)



Abdur Rozaq

Your Role/Contribution

[Abdur Rozaq](https://linkedin.com/in/Abdur-Rozaq)



Erwin Duadja

Jungler/Support/Feeder/Beban

[erwin-duadja-betha-sasana](https://linkedin.com/in/erwin-duadja-betha-sasana)



Fadhillah Rojabhy

Ketua Kelompok

[Fadhillah Rojabhy](https://linkedin.com/in/Fadhillah-Rojabhy)



Aldythya Nugraha

Your Role/Contribution

[Aldythya Nugraha](https://linkedin.com/in/Aldythya-Nugraha)

Outline

- Background & Problem Statement
- Objectives & Scope
- Data Collection & Data Understanding
- Preparation (Preprocessing)
- Sample Tweet (Preprocessing)
- Explore Data Analysis
- Model Development
- Training & Optimization
- Results



Background & Problem Statement

Sentiment Analysis:

- Teknik komputasional untuk menganalisis opini atau perasaan dalam teks.
- Mengidentifikasi sentimen positif, negatif, atau netral dari teks.
- Digunakan dalam berbagai aplikasi seperti analisis media sosial, pemasaran, politik, dan layanan pelanggan.

Project yang Dikembangkan:

- Mengembangkan sistem berbasis AI untuk klasifikasi sentimen pada Twitter.
- Dataset berasal dari tweet pengguna saat Pemilihan Presiden Indonesia 2019.
- Terdapat 1.815 data tweet dengan kategori sentimen: positif, netral, dan negatif.

Masalah Utama:

- Tantangan dalam mengklasifikasi sentimen dengan akurasi yang tinggi.
- Perlu memilih teknik dan algoritma yang tepat untuk memaksimalkan performa.
- Tantangan dalam menangani teks beragam dari media sosial yang sering kali tidak terstruktur.



Objectives & Scope

Tujuan Proyek:

- Melakukan eksperimen dengan dua algoritma utama: Random Forest dan Long Short-Term Memory (LSTM).
- Menguji berbagai teknik preprocessing dan vektorisasi untuk meningkatkan kualitas data input.
- Optimasi performa model menggunakan hyperparameter tuning.

Ruang Lingkup Proyek:

- Melakukan eksperimen dengan algoritma Random Forest & LSTM.
- Membandingkan performa algoritma melalui evaluasi model berdasarkan metrik akurasi dan lainnya.
- Melakukan optimasi model dengan menggunakan hyperparameter tuning.
- Publikasi hasil eksperimen dan model akhir ke platform Github.

Langkah Selanjutnya:

- Menentukan algoritma terbaik melalui evaluasi performa model.
- Menyusun laporan akhir yang berisi hasil analisis dan kesimpulan proyek.
- Publikasi kode dan hasil di Github untuk digunakan oleh komunitas atau pengembangan lebih lanjut.



Data Collection & Data Understanding

Data Collection

- **Sumber Data:** Dataset berisi tweet dari **Pemilihan Presiden Indonesia 2019**.
- **Jumlah Data:** Terdapat total 1.815 tweet yang dikategorikan ke dalam tiga jenis sentimen.
- **Kolom Dataset:**
 - **Unnamed:** Kolom index/nomor urut
 - **Sentimen:** Target/Dependent variable yang merepresentasikan sentimen (Positif, Netral, Negatif).
 - **Tweet:** Independent variable yang berisi teks tidak terstruktur dari tweet pengguna.
- **Tidak ada yang sama/ duplikat.**

```
print('Jumlah Data : ', len(df))

df = df.drop_duplicates()
print('Jumlah Data setelah menghapus data duplikat : ', len(df))
Executed at 2024.09.26 15:42:42 in 235ms

Jumlah Data : 1815
Jumlah Data setelah menghapus data duplikat : 1815
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1815 entries, 0 to 1814
Data columns (total 3 columns):
 #   Column      Non-Null Count Dtype  
 ---  -----      -----          ----- 
 0   Unnamed: 0   1815 non-null   int64  
 1   sentimen    1815 non-null   object  
 2   tweet       1815 non-null   object  
dtypes: int64(1), object(2)
memory usage: 42.7+ KB
```

Data Collection & Data Understanding

Data Understanding

- **Distribusi sentimen:**

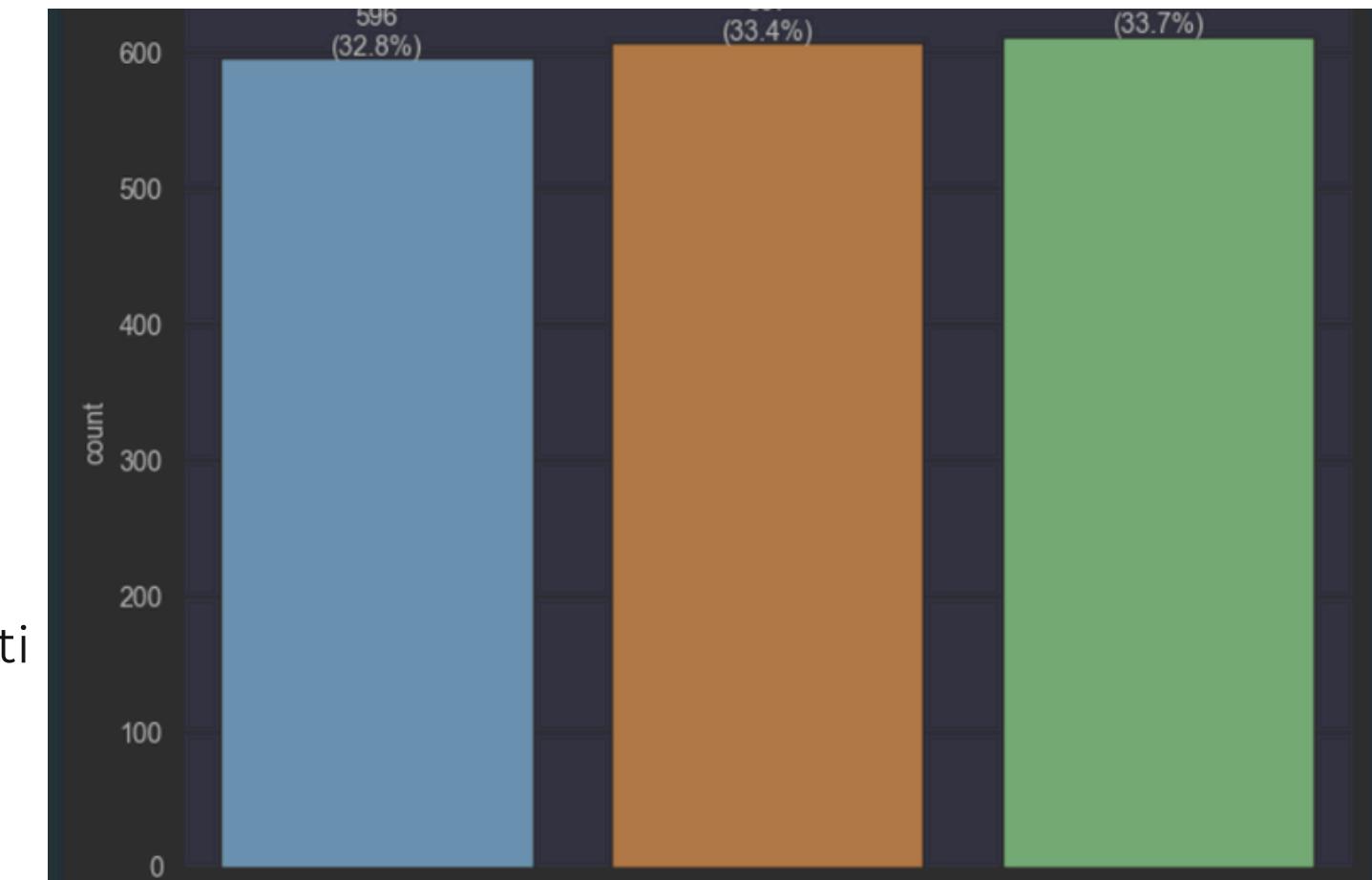
- Negatif: 596 tweet (32,8%)
- Netral: 607 tweet (33,4%)
- Positif: 612 tweet (33,7%)

- **Keseimbangan Kategori:** Dataset memiliki distribusi yang cukup merata antara tiga sentimen ini, yang membantu memastikan model klasifikasi sentimen tidak bias. Label sentimen yang ada cukup balance, sehingga tidak memerlukan oversampling maupun undersampling.

- **Tantangan:**

- **Teks Tidak Terstruktur:** Tweet terdiri dari bahasa yang tidak terstruktur dan sulit diproses secara langsung.
- **Tata Bahasa Tidak Baku & Bahasa Gaul:** Penggunaan slang, singkatan, dan tata bahasa yang tidak baku sangat umum.
- **Emoji & Emoticon:** Banyak tweet menggunakan emoji atau emoticon, yang dapat mempengaruhi hasil klasifikasi sentimen.
- **Noisy Words:** Terdapat banyak kata-kata yang tidak relevan, seperti singkatan dan typo (kesalahan ketik), yang menurunkan kualitas data untuk analisis.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1815 entries, 0 to 1814
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   Unnamed: 0   1815 non-null   int64  
 1   sentimen    1815 non-null   object  
 2   tweet       1815 non-null   object  
dtypes: int64(1), object(2)
memory usage: 42.7+ KB
```



Preparation (Preprocessing)

- **Teks Cleaning**

- **Drop Kolom yang Tidak Diperlukan:** Menghapus kolom Unnamed:0.
- **Konversi ke Lowercase:** Semua teks diubah menjadi huruf kecil.
- **Hapus @mentions & #hashtags:** Menghilangkan mention (@user) dan hashtag (#topic). untuk mention @prabowo, @jokowi, @sandiuno tidak.
- **Hapus angka:** Tidak berperngaruh ke sentimen
- **Hapus URLs:** Menghilangkan URL dalam teks.
- **Hapus Special Characters:** Karakter khusus seperti tanda baca dihapus.
- **Hapus Spasi Berlebih:** Menghilangkan spasi berlebih dalam teks.
- **Replace Slang words:** Mengubah kembali ke kata aslinya (baku)
- **Hapus Stopwords:** Stopwords umum dalam Bahasa Indonesia dihapus, namun stopwords yang dianggap memiliki nilai mengubah sentimen tidak kami hapus

```
~ exc_stopwords = [  
    'tidak', 'tak', 'belum', 'bukan', 'tanpa', 'jarang', 'kurang',  
    'baik', 'bisa', 'mungkin', 'boleh', 'masalah'  
]
```

- **Text Normalization**

- **Stemming:** Mengubah kata ke bentuk dasar.
- **Lemmatization:** Proses lemmatization untuk pemetaan kata.

- **Text Tokenization**

- **Word Tokenize:** Memisahkan teks menjadi kata-kata token individu.



Sample Tweet (Preprocessing)

- Index: 1144
- **Asli:**
 - Kesimpulannya perkembangan ekonomi, agraria dan kemaritiman menjadi fokus utama PAS bukan tidak peduli terhadap esport tapi masih ada masalah yg lebih d utamakan sama seperti halnya jokowi yg menomor satukan infrastruktur.
- **Cleaning:**
 - kesimpulannya perkembangan ekonomi agraria kemaritiman menjadi fokus utama pas bukan tidak peduli esport masalah lebih utamakan sama halnya jokowi menomor satukan infrastruktur
- **Stemming:**
 - simpul kembang ekonomi agraria maritim jadi fokus utama pas bukan tidak peduli esport masalah lebih utama sama hal jokowi nomor satu infrastruktur
- **Lemmatized:**
 - simpul kembang ekonomi agraria maritim jadi fokus utama pas bukan tidak peduli esport masalah lebih utama sama hal jokowi nomor satu infrastruktur
- **Tokenize:**
 - ['simpul', 'kembang', 'ekonomi', 'agraria', 'maritim', 'jadi', 'fokus', 'utama', 'pas', 'bukan', 'tidak', 'peduli', 'esport', 'masalah', 'lebih', 'utama', 'sama', 'hal', 'jokowi', 'nomor', 'satu', 'infrastruktur']
- **Sentimen:**
 - netral

Sample Tweet (Preprocessing)

- Index: 573

- **Asli:**

- @prabowo dan @sandiuno kalau diberi mandat rakyat gak akan ambil gaji. Gak di jiplak lagi tah
@Dennysiregar7 ڦڻ~,ڦڻ~,

- **Cleaning:**

- prabowo sandiuno kalau diberi mandat rakyat gak ambil gaji gak jiplak tah ڦڻ ڦڻ

- **Stemming:**

- prabowo sandiuno kalau beri mandat rakyat gak ambil gaji gak jiplak tah ڦڻ ڦڻ

- **Lemmatized:**

- prabowo sandiuno kalau beri mandat rakyat gak ambil gaji gak jiplak tah

- **Tokenize:**

- ['prabowo', 'sandiuno', 'kalau', 'beri', 'mandat', 'rakyat', 'gak', 'ambil', 'gaji', 'gak', 'jiplak', 'tah']

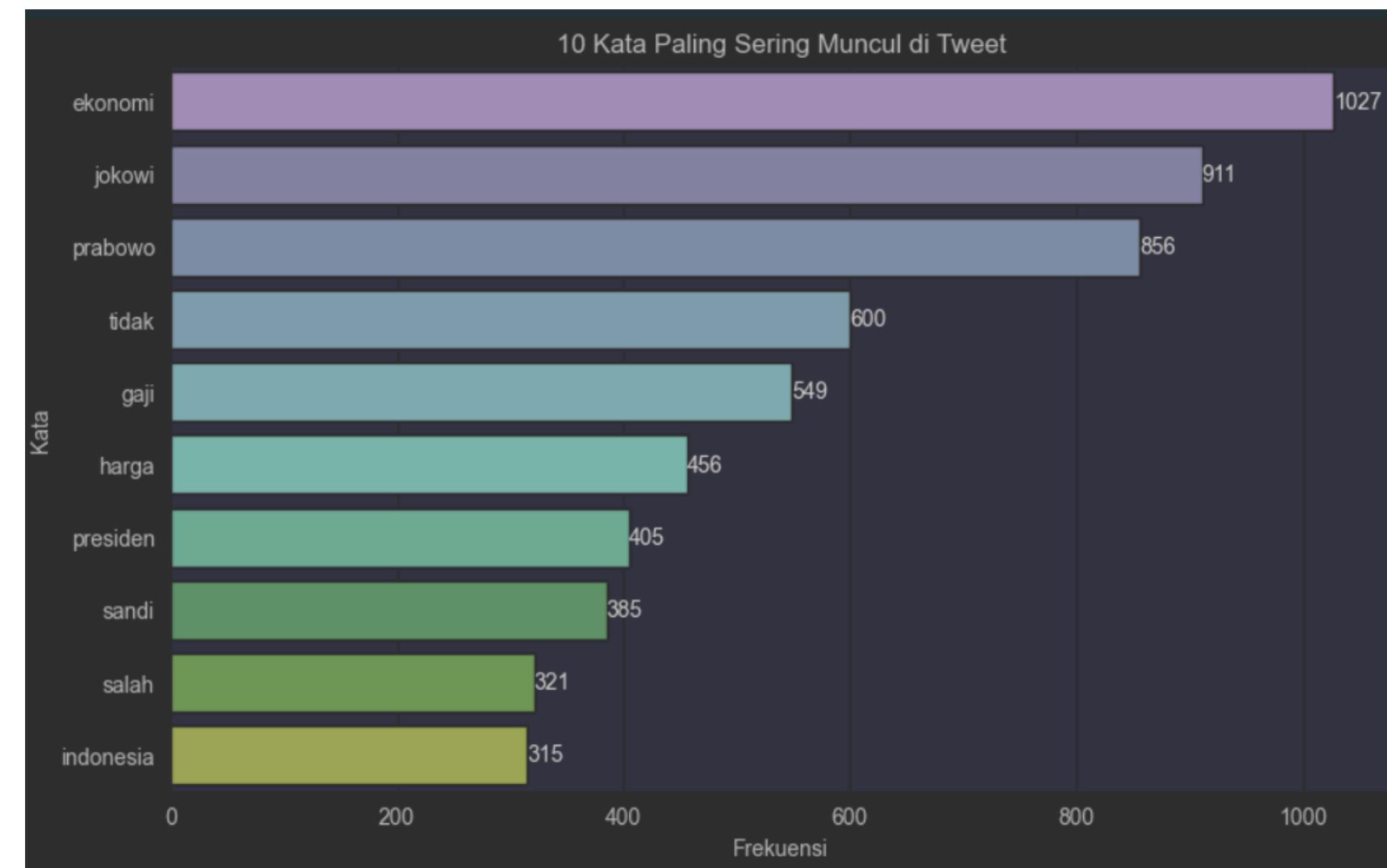
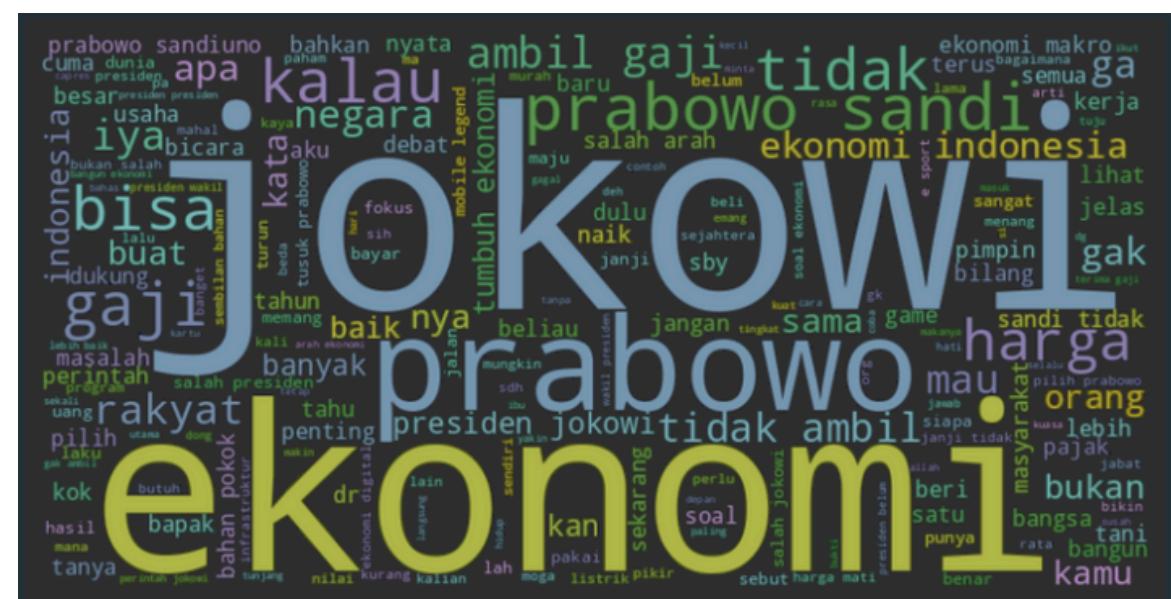
- **Sentimen:**

- positif

Exploratory Data Analysis

- 10 Kata Paling Banyak pada Tweet
 - Terbanyak:
 - Ekonomi : 1027
 - Paling Sedikit:
 - Indonesia : 315
 - Nama tokoh seperti Jokowi dan Prabowo muncul dengan frekuensi tinggi.
 - Isu gaji, harga, dan presiden juga sering dibicarakan.

- Wordcloud



Exploratory Data Analysis

sentimen word	Jml_Tweet	positif	netral	negatif	% Ngtf	% Neutr	% Pstf
ekonomi	1027	294	369	364	35.44	35.93	28.63
jokowi	911	320	329	262	28.76	36.11	35.13
prabowo	856	318	304	234	27.34	35.51	37.15
tidak	600	211	193	196	32.67	32.17	35.17
gaji	549	227	153	169	30.78	27.87	41.35
harga	456	170	148	138	30.26	32.46	37.28
presiden	405	144	132	129	31.85	32.59	35.56
sandi	385	160	151	74	19.22	39.22	41.56
salah	321	83	115	123	38.32	35.83	25.86
indonesia	315	120	106	89	28.25	33.65	38.10

Keterangan:

- 1.Jika ada pada tweet terdapat kata `sandi` dan `gaji` maka sekitar 41% tweet tersebut bersentimen positif.
- 2.Jika pada tweet terdapat kata `sandi` dan `jokowi` maka sekitar 36-39% tweet tersebut bersentimen netral
- 3.sementara jika pada tweet terdapat kata `salah` dan `ekonomi` maka sekitar 35-38% tweet tersebut bersentimen negatif

Model Development, Training, Optimization #Scheme 1

Train : 70%

Val : 15%

Test : 15%

Random Forest	LSTM (Long-Short Term Memory)
Random Forest - Stop Words - Word2Vec - Skip-gram - GridSearchCV	LSTM With Preprocessing
Random Forest - No Stop Words - Word2Vec - Skip-gram GridSearchCV	LSTM Without Preprocessing

```
param_grid = {  
    'n_estimators': [100, 300],  
    'max_depth': [10, 50],  
    'min_samples_split': [0.04, 0.08],  
    'min_samples_leaf': [0.01, 0.02],  
    'max_features': ['sqrt'],  
    'bootstrap': [True],  
    'oob_score': [True],  
}
```

Results #Scheme 1

accuracy	RF_SW_Word2Vec_SG_GSCV	RF_No_SW_Word2Vec_SG_GSCV	LSTM-Preprocessing	LSTM-No-Preprocessing
Train	0.70	0.72	0.99	0.52
Validation	0.57	0.49	0.46	0.51
Test	0.58	0.48	0.46	0.53

best model :

Random Forest - Stopwords - Word2Vec - Skip-gram - GridSearchCV

Classification Report (Train):				
	precision	recall	f1-score	support
negatif	0.66	0.83	0.73	429
netral	0.74	0.70	0.72	438
positif	0.72	0.56	0.63	403
accuracy			0.70	1270
macro avg	0.70	0.69	0.69	1270
weighted avg	0.70	0.70	0.69	1270

Classification Report (Validation):				
	precision	recall	f1-score	support
negatif	0.54	0.67	0.60	85
netral	0.53	0.63	0.58	75
positif	0.66	0.46	0.54	112
accuracy			0.57	272
macro avg	0.58	0.59	0.57	272
weighted avg	0.59	0.57	0.57	272

Classification Report (Test):				
	precision	recall	f1-score	support
negatif	0.52	0.66	0.58	82
netral	0.61	0.63	0.62	94
positif	0.62	0.47	0.54	97
accuracy			0.58	273
macro avg	0.59	0.59	0.58	273
weighted avg	0.59	0.58	0.58	273

Results Scheme #2

Random Forest

Random Forest Classification Report (Training Data):				Random Forest Classification Report (Validation Data):				Random Forest Classification Report (Test Data):				
	precision	recall	f1-score	support	precision	recall	f1-score	support	precision	recall	f1-score	support
negatif	1.00	1.00	1.00	481	0	0.42	0.89	0.57	56	neg	Add text cell	0.61
neutra	1.00	1.00	1.00	489	1	0.62	0.23	0.33	66	neutra	0.57	0.66
positif	1.00	1.00	1.00	482	2	0.74	0.49	0.59	59	positif	0.66	0.48
accuracy			1.00	1452	accuracy			0.52	181	accuracy		0.61
macro avg	1.00	1.00	1.00	1452	macro avg	0.60	0.54	0.50	181	macro avg	0.61	0.60
weighted avg	1.00	1.00	1.00	1452	weighted avg	0.60	0.52	0.49	181	weighted avg	0.61	0.60
Confusion Matrix (Training Data):				Confusion Matrix (Validation Data):				Confusion Matrix (Test Data):				
[[480 0 1]				[[50 2 4]				[[79 24 12]				
[1 488 0]				[45 15 6]				[19 78 21]				
[0 2 480]]				[23 7 29]]				[31 36 63]]				

LSTM

```

Epoch 1/10
46/46 71s 1s/step - accuracy: 0.4094 - loss: 1.0732 - val_accuracy: 0.4490 - val_loss: 1.0265
Epoch 2/10
46/46 12s 268ms/step - accuracy: 0.5420 - loss: 0.9423 - val_accuracy: 0.5482 - val_loss: 0.9883
Epoch 3/10
46/46 13s 293ms/step - accuracy: 0.7304 - loss: 0.6637 - val_accuracy: 0.5372 - val_loss: 1.0058
Epoch 4/10
46/46 19s 266ms/step - accuracy: 0.8356 - loss: 0.4326 - val_accuracy: 0.5785 - val_loss: 1.0790
Epoch 5/10
46/46 21s 272ms/step - accuracy: 0.9376 - loss: 0.1998 - val_accuracy: 0.5923 - val_loss: 1.1691
Epoch 6/10
46/46 19s 239ms/step - accuracy: 0.9508 - loss: 0.1375 - val_accuracy: 0.5620 - val_loss: 1.4141
Epoch 7/10
46/46 11s 228ms/step - accuracy: 0.9520 - loss: 0.1220 - val_accuracy: 0.5647 - val_loss: 1.3866
Epoch 8/10
46/46 21s 253ms/step - accuracy: 0.9663 - loss: 0.0785 - val_accuracy: 0.6061 - val_loss: 1.4691
Epoch 9/10
46/46 14s 310ms/step - accuracy: 0.9791 - loss: 0.0683 - val_accuracy: 0.5758 - val_loss: 1.5921
Epoch 10/10
46/46 18s 243ms/step - accuracy: 0.9828 - loss: 0.0478 - val_accuracy: 0.5565 - val_loss: 1.6188
12/12 1s 81ms/step - accuracy: 0.5605 - loss: 1.6302
LSTM Model Accuracy: 0.5564738512039185

```

Define the parameter grid for hyperparameter tuning

```

param_grid = {
    'n_estimators': [100, 200, 300],
    'max_depth': [10, 20, 30],
    'min_samples_split': [2, 5, 10]
}

```

Classification Report:				
	precision	recall	f1-score	support
negatif	0.61	0.70	0.65	115
neutra	0.61	0.63	0.62	118
positif	0.64	0.53	0.58	130
accuracy			0.62	363
macro avg	0.62	0.62	0.62	363
weighted avg	0.62	0.62	0.62	363
Confusion Matrix:				
[[81 21 13]				
[18 74 26]				
[34 27 69]]				

Results #3 (keras_tuner & RandomizedSearchCV)

```
25 rf_random_search = RandomizedSearchCV(estimator=rf_model, param_distributions=param_dist, n_iter=100, cv=3, verbose=2, random_state=42, n_jobs=-1)
26
27 rf_random_search.fit(X_train, y_train)
28
29 best_params = rf_random_search.best_params_
30 print("Best Parameters:", best_params)
31
32 best_rf_model = rf_random_search.best_estimator_
33
34 y_pred_rf = best_rf_model.predict(X_test)
35 rf_report = classification_report(y_test, y_pred_rf, target_names=['negatif', 'positif', 'netral'])
36 print(rf_report)
Executed at 2024.09.27 14:34:28 in 35s 242ms

    Fitting 3 folds for each of 100 candidates, totalling 300 fits
    Best Parameters: {'n_estimators': 800, 'min_samples_split': 10, 'min_samples_leaf': 4, 'max_features': 'log2', 'max_depth': 60, 'bootstrap': True}
        precision    recall   f1-score   support
      negatif     0.40     0.29     0.34     115
      positif     0.47     0.49     0.48     130
      netral      0.46     0.56     0.50     118
      accuracy          0.45     0.45     0.45     363
      macro avg     0.44     0.45     0.44     363
      weighted avg  0.44     0.45     0.44     363
```

```
51 model = tuner.hypermodel.build(best_hps)
52 history = model.fit(X_train_pad, y_train_pad, epochs=5, batch_size=64, validation_data=(X_test_pad, y_test_pad))
53
54 lstm_loss, lstm_accuracy = model.evaluate(X_test_pad, y_test_pad)
55 print(f"Test Loss: {lstm_loss}")
56 print(f"Test Accuracy: {lstm_accuracy}")
```

```
    Trial 10 Complete [00h 00m 23s]
    val_accuracy: 0.6143250465393066

    Best val_accuracy So Far: 0.641873300075531
    Total elapsed time: 00h 05m 51s

    The optimal number of units in the embedding layer is 96.
    The optimal number of units in the LSTM layer is 96.
    The optimal dropout rate for the LSTM layer is 0.2.
    The optimal recurrent dropout rate for the LSTM layer is 0.2.
    The optimal number of units in the dense layer is 160.
    The optimal dropout rate for the dense layer is 0.5.
```

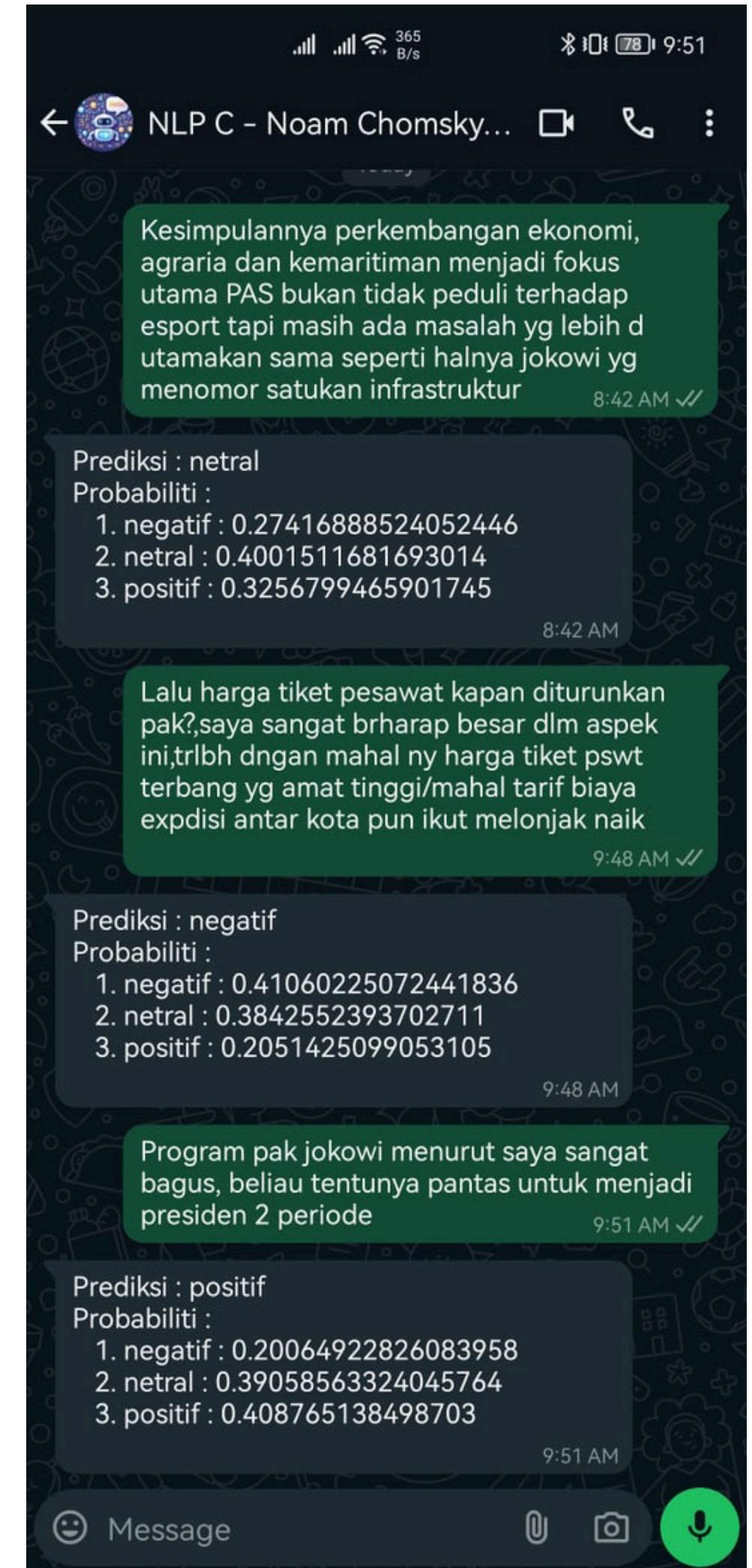
Real-world Application

- Aplikasi berupa chatbot whatsapp, yang dapat di akses melalui:
 - +62 877-8761-1391



- Spam filter email/message/whatsapp
- Review/customer experience di dunia retail/kesehatan/travel dll
- Sentimen berita dan rumor untuk bursa saham

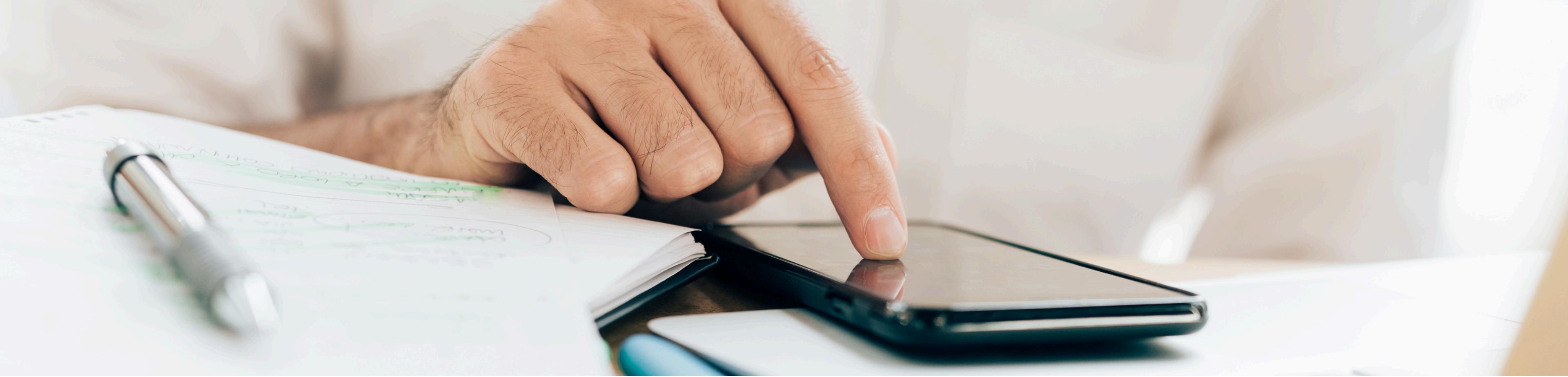
<https://github.com/rowjak/sentiment-analysis-pilpres-2019>



Conclusion & Future Improvement

- AI dapat sangat membantu pekerjaan terutama dalam konteks projects ini para social media analyst dan tim sukses/kampanye
- Banyak hal yang harus dilakukan terutama dalam tahap pre-processing untuk bisa memperoleh hasil maksimal
- Performa kedua model baik Randomforest dan LSTM sama-sama kurang baik untuk sentimen analisis twit
- Handle sarcasm
- Handle noisy words (singkatan dan typos) : fasttext atau transformers/pre-trained model

```
Executed at 2024.09.26 18:44:16 in 172ms
  rumah, zci, titik, masayarakat, aniesbasu, nonggo, makanan, simpang, tarik, banjir, grup, santai, ya, dat, rebel, ketajemukan,  
'kumpulin', 'imbauan', 'halilintar', 'subtansi', 'trilyunan', 'amatir', 'jarak', 'simpang', 'âœ', 'munculah', 'parlente', 'ril', 'angsur', 'trlihat', 'kapabilitas',  
'mala', 'rw', 'max', 'sono', 'pantesnya', 'relatif', 'perusahaann', 'lembek', 'animasi', 'mantaps', 'normatif', 'nmm', 'selaras', 'digembar', 'urai', 'mengigil', 'smi',  
'satire', 'mesem', 'ngajarin', 'jngan', 'preadient', 'mora', 'trhormatnya', 'pamrih', 'marjinal', 'padat', 'invest', 'multifungi', 'adaa', 'bengkalai', 'gausah',  
'rekamany', 'msyarakat', 'mncari', 'brniat', 'khidupn', 'sangkin', 'stuck', 'trbukti', 'tlh', 'timeout', 'gantiðÿ', 'cermat', 'yaelaah', 'proud', 'who', 'have', 'we',  
'time', 'usul', 'sifudan', 'insyaaallah', 'proporsional', 'hambur', 'watt', 'objektif', 'pangkas', 'akses', 'musolini', 'italia', 'rossi', 'mengkan', 'pamugkas',  
'yaaaaa', 'serahkn', 'gajinyaaaa', 'cawapresx', 'kinclong', 'nyapres', 'order', 'berkutet', 'klok', 'miraprojo', 'lapis', 'jepang', 'gblk', 'mantull', 'pancasaila',  
'nangkep', 'drasakan', 'dutarakan', 'bambanggg', 'kaw', 'gmna', 'nyediain', 'hargain', 'mudeng', 'maap', 'ðÿ', 'blaðÿ', 'pki', 'pade', 'nyampe', 'omzet', 'bankrut',  
'rekor', 'australia', 'korea', 'capek', 'ngnggur', 'sanggah', 'visioner', 'sekua', 'sekarn', 'bdain', 'didoain', 'moesa', 'asykar', 'sahabat', 'bgawal', 'meubelnya',  
'mosyantax', 'alhmdllh', 'prjuangan', 'pegang', 'makes', 'sadis', 'tilap', 'niru', 'halahhhh', 'halah', 'mantaappppp', 'kontra', 'alamat', 'digitech', 'tampal',  
'technologi', 'interest', 'oom', 'lainnnya', 'jejer', 'capressnya', 'wahhh', 'disalahkn', 'seprtii', 'tekorðÿ', 'adl', 'ditanyain', 'dngerin', 'nyo', 'andi', 'konon',  
'ptkiani', 'teladan', 'maksimal', 'menyalahkn', 'seblm', 'sesi', 'matt', 'jo', 'walkot', 'ws', 'riyadh', 'khalid', 'king', 'sambut', 'cerah', 'cuaca', 'ajarin',  
'aktifitas', 'kreatifitas', 'property', 'tookoh', 'dp', 'malaikat', 'kekanak', 'sultan', 'kebun', 'primer', 'agregat', 'pengeloalan', 'srek', 'tenar', 'ngehack',  
'tampar', 'wayang', 'sesat', 'gituh', 'ngingetin', 'institusi', 'kana', 'over', 'dibegoin', 'smesh', 'layang', 'ditungguin', 'nampaknya', 'alquran', 'salat', 'desember',  
'meni', 'diapain', 'bajar', 'kantor', 'pantry', 'ngabisin', 'sabdi', 'mantabkan', 'lainnyaðÿ', 'instrumen', 'insfrastruktur', 'sbenernya', 'aplqi', 'nqikuuuuuuuut',
```



Thank you, any question ?

Semangat!!!