

Analisis Sentimen Dengan Metode K-Nearest Neighbor, Random Forest, Binary Classification Dan Recurrent Neural Network terhadap Review Gojek Pada Playstore

Azmi Taqiuddin Syah
School of Electrical Engineering
Telkom University, Indonesia
miuddinsyah@telkomuniversity.ac.id

Terryanda Naufaldo Arhaby
School of Electrical Engineering Telkom
University, Indonesia
terryanda@student.telkomuniversity.ac.id

Raihanna Fawaz
School of Electrical Engineering Telkom
University, Indonesia
raihanafw@student.telkomuniversity.ac.id

Abstract— Analisis sentimen terhadap layanan online sangat penting, terutama pada aplikasi Gojek. Dalam menganalisis sentimen dari review pengguna Gojek di Playstore, terdapat metode klasifikasi yang digunakan yaitu K-Nearest Neighbors (KNN), Random Forest, Binary Classification, dan Recurrent Neural Network (RNN). Metode KNN digunakan untuk menemukan kesamaan dengan tetangga terdekat. Random Forest digunakan untuk memperoleh hasil klasifikasi dengan mengadopsi penggabungan pohon keputusan. Binary Classification memisahkan data menjadi kelas positif dan negatif. RNN mampu memodelkan urutan data untuk mempermudah review. Setelah mengevaluasi keempat metode tersebut, terdapat kelebihan dan kekurangan dari masing-masing metode.

Keyword: Binary Classification, K-Nearest Neighbor, Random Forest, Recurrent Neural Network, Review, Sentimen.

I. PENDAHULUAN

Dalam era digital yang semakin berkembang, data yang dihasilkan oleh pengguna aplikasi mobile seperti Gojek telah menjadi sumber informasi berharga bagi pemangku kepentingan, termasuk perusahaan dan konsumen. Di antara beragam metode analisis data, analisis sentimen telah menjadi fokus utama, karena memungkinkan pemahaman yang lebih dalam tentang persepsi dan sikap pengguna terhadap produk atau layanan tertentu.

Dalam konteks ini, penelitian ini mengeksplorasi pendekatan analisis sentimen yang berfokus pada ulasan pengguna Gojek di platform Play Store. Kami memperkenalkan tiga metode yang umum digunakan: K-Nearest Neighbor (KNN), Random Forest, dan Binary Classification. Setiap metode ini memiliki pendekatan yang unik dalam mengolah data dan menghasilkan pemahaman tentang sentimen pengguna.

Pertama, K-Nearest Neighbor (KNN) adalah algoritma yang bekerja berdasarkan prinsip bahwa pengguna yang memiliki preferensi atau opini serupa cenderung berada dalam jarak yang dekat dalam ruang fitur. Kami akan menggunakan pendekatan ini untuk mengelompokkan ulasan-ulasan pengguna berdasarkan kemiripan fitur-fitur tertentu, dan dari situ, mengidentifikasi pola sentimen yang mendasarinya.

Kedua, Random Forest adalah salah satu metode yang berbasis pada teknik ensemble learning, yang menggabungkan prediksi dari beberapa model pembelajaran mesin (dalam hal ini, pohon keputusan). Kami akan menerapkan Random Forest untuk melakukan klasifikasi ulasan-ulasan pengguna menjadi kategori sentimen positif, negatif, atau netral, dengan memanfaatkan kekuatan pengambilan keputusan kolektif dari sejumlah besar pohon keputusan.

Ketiga, Binary Classification adalah pendekatan yang melibatkan pembuatan model yang memprediksi apakah suatu ulasan akan memiliki sentimen positif atau negatif. Dengan menggunakan teknik ini, kami akan mengidentifikasi fitur-fitur penting dalam ulasan pengguna yang menjadi indikator kuat untuk menentukan apakah suatu sentimen dianggap positif atau negatif.

Penelitian ini tidak hanya akan memberikan wawasan tentang persepsi pengguna terhadap Gojek, tetapi juga membandingkan efektivitas dari ketiga metode tersebut dalam melakukan analisis sentimen. Diharapkan hasil dari penelitian ini dapat memberikan pemahaman yang lebih mendalam bagi perusahaan teknologi seperti Gojek dalam meningkatkan kualitas layanan mereka, serta bagi para peneliti dalam pengembangan metodologi analisis sentimen yang lebih baik di masa depan.

II. KARYA TERKAIT

- "Analisis Sentimen Terhadap Pemindahan Ibu Kota Indonesia Pada Media" dari UIN Syarif Hidayatullah Jakarta. Penelitian ini mencakup pengukuran performa metode K-Nearest Neighbor (K-NN) dan data pelatihan menggunakan pendekatan lexicon dalam klasifikasi data uji.

- "Implementasi Algoritma K-Nearest Neighbor (K-NN) Untuk Analisis Sentimen Publik Pembelajaran Online" dari Universitas Teknokrat Indonesia. Penelitian ini menggunakan data tweet sebanyak 1825 data tweet dalam Bahasa Indonesia yang dikumpulkan dari 1

- "Perbandingan Metode Random Forest dan K-Nearest Neighbors Terhadap Sentimen Opini Publik terhadap Promo Gojek" dari Universitas Pembangunan Nasional Veteran Jakarta. Penelitian ini diharapkan bisa mendapatkan informasi akan sentimen opini publik terhadap promo Gojek serta mengetahui performa metode K-Nearest Neighbors dan Random Forest. st

- "Analisa Sentimen Terhadap Review Fintech Dengan Metode Naive Bayes Classifier dan K-Nearest Neighbor" dari Neliti. Penelitian ini menggunakan metode eksperimen dengan menggunakan Algoritma Naive Bayes Classifier dan K-Nearest Neighbor.

- "Analisis Sentimen Pada Ulasan Aplikasi Gojek Menggunakan Metode Support Vector Machine (SVM) dan K Nearest Neighbor (KNN)" dari Universitas Negeri Semarang. Tujuan penelitian ini adalah melakukan klasifikasi pada ulasan Aplikasi Gojek menggunakan metode Support Vector Machine (SVM) dan K Nearest Neighbor (KNN).

III. METODE PENELITIAN

A. Dataset Review Gojek

Dataset "Review Gojek" yang akan digunakan dalam penelitian ini berisi informasi tentang ulasan pengguna aplikasi Gojek di platform Play Store. Data tersebut mencakup beberapa atribut, antara lain:

1. Username: Nama pengguna yang memberikan ulasan.
2. Review/Content: Isi ulasan yang diberikan oleh pengguna.
3. Score: Nilai skor yang diberikan oleh pengguna untuk aplikasi Gojek.
4. Date: Tanggal ulasan diberikan oleh pengguna.
5. App Version: Versi aplikasi Gojek yang digunakan oleh pengguna saat memberikan ulasan.

B. Metode Analisis Sentimen

Pada penelitian ini, kami akan menggunakan metode analisis sentimen untuk mengidentifikasi pola sentiment dalam ulasan-ulasan pengguna. Tiga metode yang akan kami terapkan adalah K-Nearest Neighbor (KNN), Random Forest, dan Binary Classification.

1. K-Nearest Neighbor (KNN): Kami akan menerapkan algoritma KNN untuk mengelompokkan ulasan-ulasan pengguna berdasarkan kemiripan fitur-fitur tertentu. Dengan menggunakan pendekatan ini, kami akan mencoba mengidentifikasi pola sentiment yang mendasarinya dengan memperhatikan ulasan-ulasan yang serupa.
2. Random Forest: Kami akan menggunakan teknik Random Forest untuk melakukan klasifikasi ulasan-ulasan pengguna ke dalam kategori sentimen positif, negatif, atau netral. Dengan memanfaatkan kekuatan pengambilan keputusan kolektif dari sejumlah besar pohon keputusan, kami akan mencoba mengklasifikasikan ulasan-ulasan tersebut berdasarkan sentiment yang terkandung di dalamnya.
3. Binary Classification: Kami juga akan menerapkan pendekatan Binary Classification untuk memprediksi apakah suatu ulasan memiliki sentiment positif atau negatif. Dengan menggunakan teknik ini, kami akan mengidentifikasi fitur-fitur penting dalam ulasan pengguna yang menjadi indikator kuat untuk menentukan sentimentnya.
4. Recurrent Neural Network: Kami juga akan menerapkan pendekatan jenis jaringan saraf tiruan yang menggunakan data sekuensial atau data deret waktu

Melalui penelitian ini, diharapkan dapat diperoleh pemahaman yang lebih mendalam tentang sentimen pengguna terhadap aplikasi Gojek, serta perbandingan efektivitas ketiga metode analisis sentimen yang digunakan. Hal ini diharapkan dapat memberikan wawasan yang berharga bagi Gojek dalam meningkatkan kualitas layanan mereka, serta bagi peneliti dalam pengembangan metodologi analisis sentimen yang lebih baik di masa depan.

IV. SISTEM DESAIN DAN GAMBARAN UMUM

Dataset dengan Skenario Sebagai berikut:

```
import pandas as pd

df = pd.read_csv('GojekAppReviewV4.0.0-V4.9.3_Cleaned.csv')
df.head()
```

	userName	content	score	at	appVersion
0	Yuga Edit	akun gopay saya di blok	1	2022-01-21 10:52:12	4.9.3
1	ff buruk	Lambat sekali sekarang ini bosssku apk gojek g...	3	2021-11-30 15:40:38	4.9.3
2	Anisa Suci Rahmayuliani	Kenapa sih dari kemarin sy buka aplikasi gojek...	4	2021-11-29 22:58:12	4.9.3
3	naoki yakuza	Baru download gojek dan hape baru trus ditop u...	1	2022-09-03 15:21:17	4.9.3
4	Trio Sugianto	Mantap	5	2022-01-15 10:05:27	4.9.3

Dengan data sebanyak 225001

	userName	content	score	at	appVersion
224997	Sad Garner	Gofood Biaya lain2ya gak ngotak mending hujan2...	1	2023-02-15 09:37:58	4.0.0
224998	fadhil fadil	Yok lah	5	2021-12-11 12:28:20	4.0.0
224999	g sugianto	Sempurna	5	2022-03-14 01:45:39	4.0.0
225000	J I H A D ' B E	GOJEK LAMA LAMA GAK JELAS LAGI PESEN MASA MAP...	1	2021-12-24 08:48:51	4.0.0
225001	Ria Heria	Good	5	2022-03-17 14:33:34	4.0.0

Lalu Ambil Kolom yang dibutuhkan

```
# ambil kolom yg dibutuhkan
df = df[df['appVersion'].str.startswith("4.8")]
df = df.loc[:, ['userName', 'content', 'score']]
df.head()
```

	userName	content	score
133	Abu karim aljabbar Mkatiksaidi	Ramah banget	5
134	Fathan Mubina	Setelah update kok nggak bisa dibuka	4
135	Nyauw Jin Fie	Good	5
136	Tanaka Kun	Good	5
137	Anton S.	Sangat membantu	5

Lalu Hapus Duplikasi dan berikan stopwords Indonesia seperti foto dibawah ini

```
# Hapus duplikasi
df = df.drop_duplicates(subset=['content'], drop_duplicates())

# stopwords
stop_words = stopwords.words('indonesian') + stopwords.words('english') + ["yg", "gak", "ngisii", "udah", "di", "sok", "nya", "arg", "ata", "bpk", "ga"]
df['content'] = df['content'].apply(lambda x: [word.lower() for word in word_tokenize(x) if (word.isalpha() and word.lower() not in stop_words)])

# normalisasi teks
df['content'] = df['content'].apply(lambda x: ' '.join(x))
df.head()
```

Lalu Dataset akan menjadi

	userName	content	score
133	Abu karim aljabbar Mkatiksaidi	ramah banget	5
134	Fathan Mubina	update nggak dibuka	4
135	Nyauw Jin Fie	good	5
136	Tanaka Kun	good	5
137	Anton S.	membantu	5

Lalu lakukan stemming

```
# Stemming
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

stemmer = StemmerFactory().create_stemmer()
df['content'] = df['content'].apply(lambda x: ' '.join([stemmer.stem(word) for word in x.split()]))
df.head(5)
```

	userName	content	score
133	Abu karim aljabbar Mkatiksaidi	ramah banget	5
134	Fathan Mubina	update nggak buka	4
135	Nyauw Jin Fie	good	5
136	Tanaka Kun	good	5
137	Anton S.	bantu	5

Lalu lakukan Labeling seperti dibawah ini

Label	Score
'kecewa':	-0.4,
'rugi':	-1
'buruk':	-0.6,
'jelek':	-0.6,
'lelet':	-0.7,

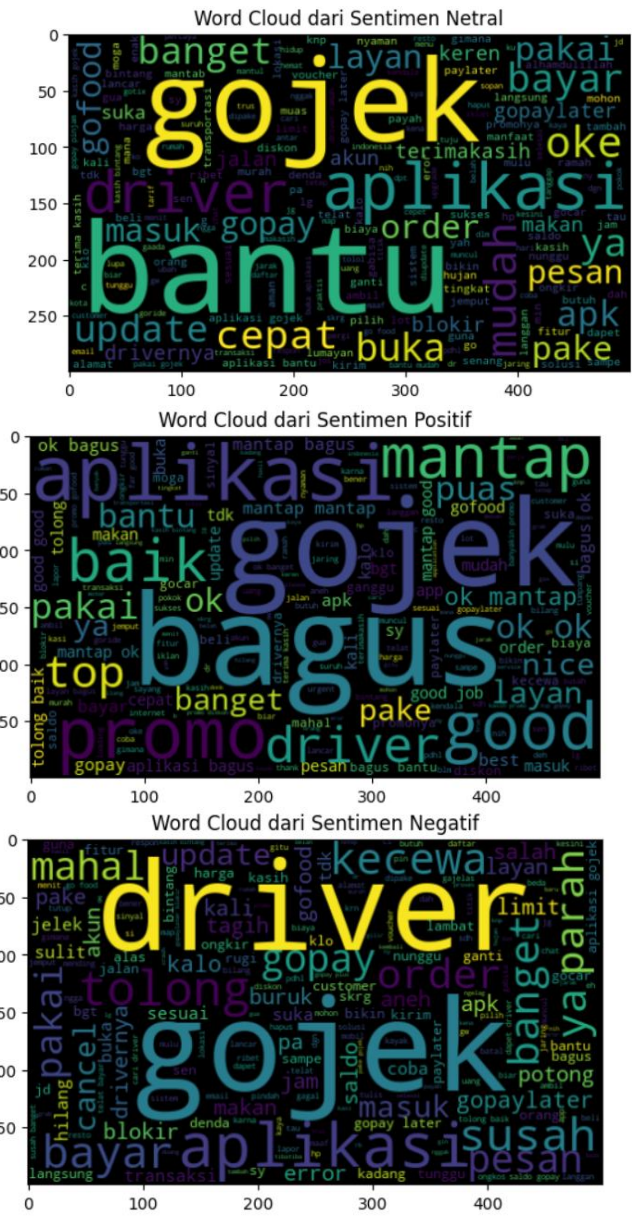
'gagal':	-0.5,
'parah':	-0.6,
'mahal':	-0.3,
'tolong':	-0.1,
'hilang':	-0.3,
'gajelas':	-0.3,
'gj':	-0.3,
'promo':	0.6,
'kadang':	-0.1,
'maling':	-0.5,
'ganggu':	0.3,
'sedot':	-0.5,
'bagus':	0.5,
'pulsa':	0
'potong':	-1
'baik':	0.5,
'kntl':	-1
'ngelag':	-0.8,
'salah':	-0.5,
'bintang':	0
'benerin':	-0.4,
'lambat':	-0.8,
'siput':	-0.4,
'mati':	-0.7,
'minimal':	-0.3,
'susah':	-0.6,
'nagih':	-0.6,
'capek':	-0.7,
'kacau':	-0.3,
'tagih':	-0.3,
'mantap':	1
'puas':	0.9,
'sampah':	-0.5,
'sulit':	-0.6,
'aneh':	-0.4,

Dengan Labeling ini disimpulkan sentiment sebagai berikut

```
#['sentimen'] = df['content'].apply(lambda x: 'Positif' if analyzer.polarity_scores(x)['compound'] > 0 else ('negatif' if analyzer.polarity_scores(x)['compound'] < -0.5 else 'Netral'))
df
```

	userName	content	score	sentimen
133	Abu karim aljabbar Mkatiksaidi	ramah banget	5	Netral
134	Fathan Mubina	update nggak buka	4	Netral
135	Nyauw Jin Fie	good	5	Positif
136	Tanaka Kun	good	5	Positif
137	Anton S.	bantu	5	Netral
...
8219	Malik Azis	gopay mantap	5	Positif
8220	Zunus	buka aplnya hapus download beberpa kali restar...	1	Netral
8221	Mutiara Purnama	bye bye gojek alih ijo lah mantapp	1	Netral
8222	Melanie Anggi	fitur gomed ya	5	Netral
8223	Ade Nurul Hidayah	go food mahal ah	1	Negatif

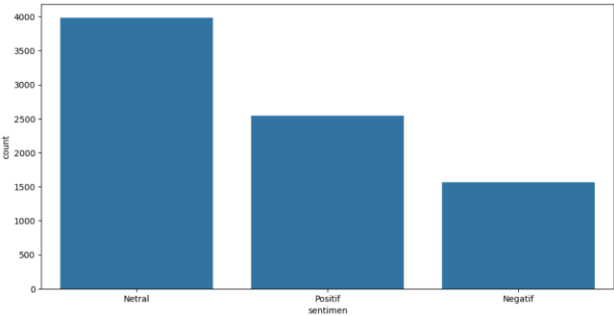
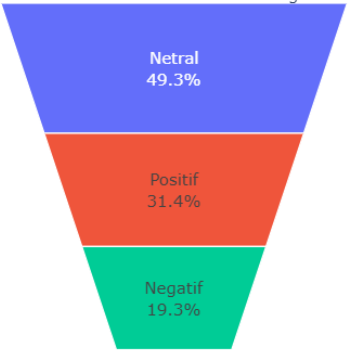
A. Visualisasi Data dengan Word Cloud



B. Kesimpulan Hasil Labeling

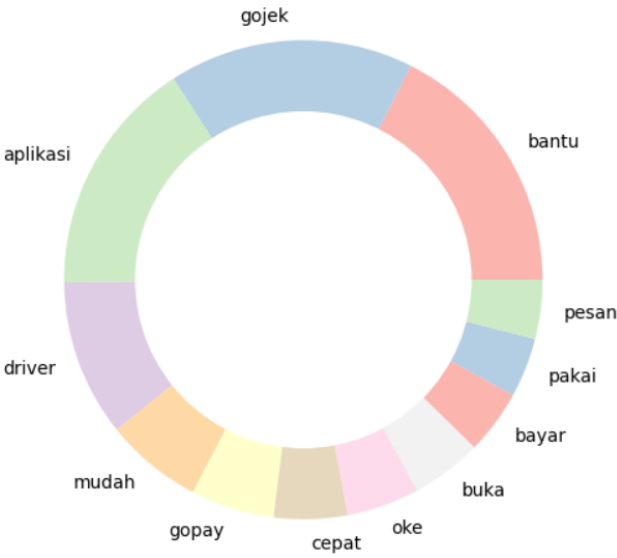
	sentimen	content
1	Netral	3987
2	Positif	2541
0	Negatif	1562

Funnel-Chart dari Distribusi target

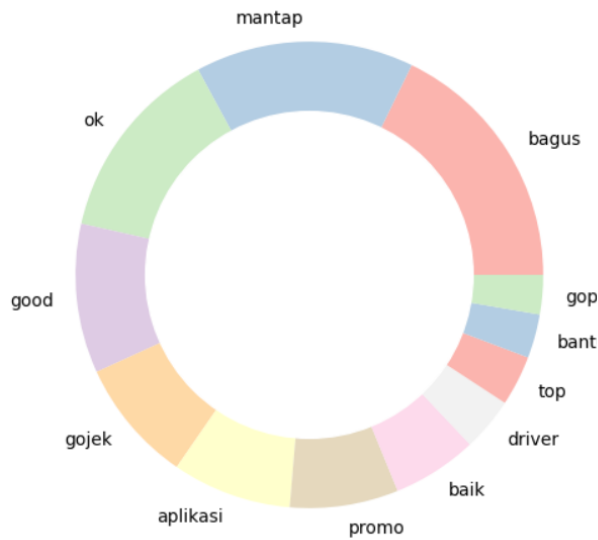


C. Donut Plot setiap Sentimen

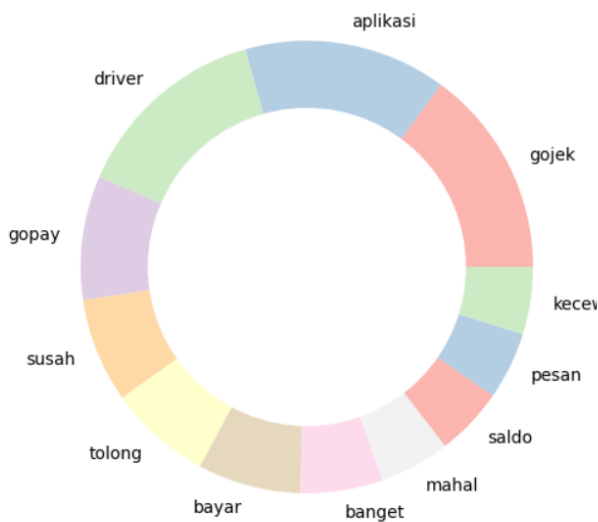
Donut Plot dari Sentimen Netral



Donut Plot dari Sentimen positif



Donut Plot dari Sentimen negatif



D. Splitting

```
# Splitting
from sklearn.model_selection import train_test_split

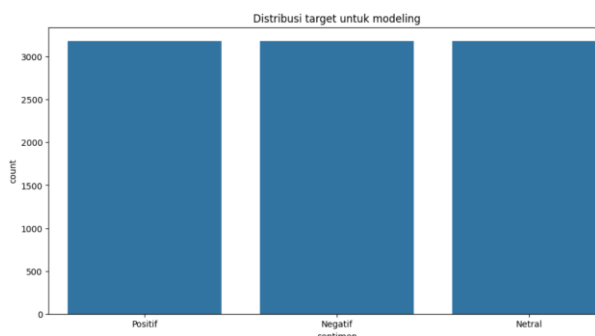
X_train, X_test, y_train, y_test = train_test_split(X_train, y_train, test_size=0.2, random_state=42)
X_train.shape, X_test.shape

((6472, 6961), (1618, 6961))

from imblearn.over_sampling import SMOTE

smote = SMOTE(random_state=42)
X_train, y_train = smote.fit_resample(X_train, y_train)

plt.figure(figsize=(12, 6))
sns.countplot(x=y_train)
plt.title('Distribusi target untuk modeling')
plt.show()
```



E. Random Forest Classifier

```
# list parameters
rf_param_grid = {'n_estimators': [50, 100, 200],
                 'max_depth': [None, 10, 20, 30],
                 'min_samples_split': [2, 5, 10],
                 'min_samples_leaf': [1, 2, 4]}

rf_model = RandomizedSearchCV(RandomForestClassifier(random_state=42), rf_param_grid, cv=5, scoring='accuracy', random_state=42)
rf_model.fit(X_train, y_train)
```

Best Parameters for Random Forest:
{
'n_estimators': 100,
'min_samples_split': 5,
'min_samples_leaf': 1,
'max_depth': None
}

Classification Report for Random Forest (Tuned):

	precision	recall	f1-score	support
Negatif	0.85	0.92	0.89	310
Netral	0.95	0.97	0.96	803
Positif	0.98	0.90	0.94	505
accuracy			0.94	1618
macro avg	0.93	0.93	0.93	1618
weighted avg	0.94	0.94	0.94	1618

Confusion Matrix for Random Forest Classifier

	negative	neutral	positive
Truth negative	286	19	5
neutral	21	778	4
positive	28	20	457

F. K-Nearest Neighbor

```
# Normalisasi fitur
scaler = StandardScaler(with_mean=False)
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Pemilihan Parameter K
knn_param_grid = {'n_neighbors': [3, 5, 7, 9]} # Contoh nilai K yang akan diuji

# Inisialisasi model KNN
knn_model = GridSearchCV(KNeighborsClassifier(), knn_param_grid, cv=5, scoring='accuracy')
knn_model.fit(X_train_scaled, y_train)
```

Best Parameters for KNN: {'n_neighbors': 3}

Classification Report for KNN:

	precision	recall	f1-score	support
Negatif	0.46	0.51	0.49	310
Netral	0.75	0.68	0.71	803
Positif	0.70	0.76	0.73	505
accuracy			0.67	1618
macro avg	0.64	0.65	0.64	1618
weighted avg	0.68	0.67	0.68	1618

	negative	neutral	positive
truth negative	159	109	42
truth neutral	136	547	120
truth positive	48	72	385

Binary Classification data yang kita ambil yaitu positif dan negatif dan neutral di drop

Confusion Matrix for Binary Classification (Logistic Regression)

		Predicted	
		Negative	Positive
Truth	Negative	305	7
	Positive	43	466

```
# Plotting
models = ['Random Forest', 'KNN', 'Logistic Regression']
accuracies = [accuracy_rf, accuracy_knn, accuracy_logreg]
print(accuracies)
colors = ['red', 'blue', 'green']

plt.figure(figsize=(10, 6))
plt.bar(models, accuracies, color=colors)
plt.xlabel('Models')
plt.ylabel('Accuracy')
plt.title('Comparison of Accuracy between Random Forest, KNN, and Logistic Regression')
plt.ylim(0, 1) # Set batas y-axis dari 0 hingga 1
plt.show()
```

Model	Accuracy
Random Forest	0.92
KNN	0.81
Logistic Regression	0.94

```
#set stop word
stop_words = stopwords.words('indonesian')

stop_words = set(["ya", "dg", "rt", "dgn", "ny", "d", "klo",
                  "kalo", "amp", "blar", "bikin", "bilang",
                  "gak", "ga", "nyo", "ini", "sh",
                  "sl", "tau", "tdk", "tuh", "utk", "ya",
                  "jd", "jgn", "sdh", "aja", "n", "t",
                  "nyg", "hehe", "pen", "u", "nan", "loh", "rt",
                  "kamp", "yah", "lho", "yuk", "tdk", "rb", "dg",
                  "guys", "mantean", "toh", "amp",
                  "de", "ai", "sa", "lha", "loh", "gustii",
                  "ora", "bessque", "gk", "dab", "e", "ak", "y", "w",
                  "yg", "nah", "nih", "huftttt", "lah", "aq", "aw"])

def preprocess_text(review):
    if isinstance(review, str):
        review = re.sub(r'http://s*', '', review) # menghapus link website
        review = re.sub(r'SD\d+', '', review).strip() # proses menghapus nomor
        review = re.sub(r'[A-Za-z]', '', review) # proses menghapus simbol huruf
        review = review.translate(str.maketrans("", "", string.punctuation)) # menghapus tanda baca
        review = review.lower() # proses case folding
        review = ''.join(word for word in review.split() if word not in stop_words) # proses stop word
        return review
    else:
        return ""

data['review'] = data['review'].apply(lambda x: preprocess_text(x))
```

```

• X = data['review'].tolist()

#Preprocessing pada kelas label
Y = to_categorical(data['rating'])
print("Shape pada sentiment:",Y.shape)

Shape pada sentiment: (1000, 6)

#Tokenizing and change text to sequence

max_features = 2500
MAX_SEQUENCE_LENGTH = 50

tokenizer = Tokenizer(num_words=max_features, split=' ')
tokenizer.fit_on_texts(X)

word_index = tokenizer.word_index
print("Found %s unique tokens" % len(word_index),'\n')
print("Index: "+str(tokenizer.word_index),"\\n")

X = tokenizer.texts_to_sequences(X)
X = pad_sequences(X, maxlen=MAX_SEQUENCE_LENGTH, padding='post', truncating='post')
X[:2]

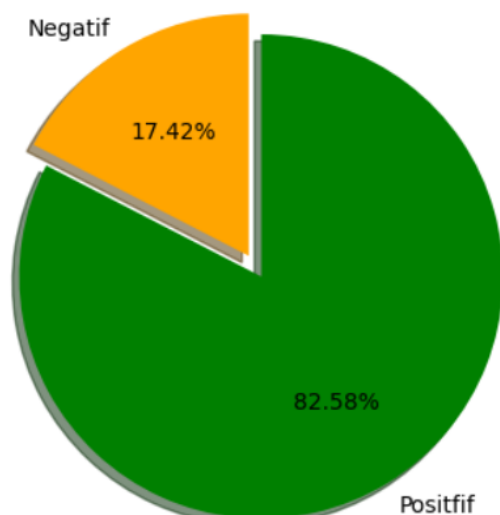
```

```
found 2332 unique tokens
```

```
Index: {'saya': 1, 'di': 2, 'gajejk': 3, 'dam': 4, 'aplikasi': 5, 'biusa': 6, 'sangat': 7, 'ada': 8, 'bagus': 9, 'tidak': 10, 'ini': 11,  
array([[ 76,   20,    3,  2892,    0,    0,    0,    0,    0,    0,    0,  
         0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,  
         0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0],  
       [458,  46,  44, 11893,  41,  3,  50, 208,  57,    0,    0,    0,  
         0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,  
         0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0],  
       [ 0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,  
         0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0])}
```

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 50, 200)	68308200
lstm (LSTM)	(None, 200)	320800
dense (Dense)	(None, 6)	1206
Total params: 68,630,206		
Trainable params: 322,006		
Non-trainable params: 68,308,200		
None		
#split dataset		
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.1, random_state = 0, stratify		
print(X_train.shape,Y_train.shape)		
print(X_test.shape,Y_test.shape)		
(900, 50) (900, 6)		
(100, 50) (100, 6)		
#cross training		
batch_size = 64		
history = model.fit(X_train, Y_train, epochs=100, validation_data=(X_test, Y_test), verbose = 1)		
epoch 1/100		
29/29 [=====] - 11s 268ms/step - loss: 1.2401 - accuracy: 0.5644 - val_loss: 1.1818 - val_accuracy: 0.57		
epoch 2/100		
29/29 [=====] - 7s 256ms/step - loss: 1.1431 - accuracy: 0.5911 - val_loss: 1.1702 - val_accuracy: 0.57		
epoch 3/100		
29/29 [=====] - 8s 275ms/step - loss: 1.1286 - accuracy: 0.5964 - val_loss: 1.1405 - val_accuracy: 0.59		
epoch 4/100		
29/29 [=====] - 7s 258ms/step - loss: 1.0322 - accuracy: 0.6311 - val_loss: 0.9987 - val_accuracy: 0.63		
epoch 5/100		
29/29 [=====] - 7s 249ms/step - loss: 0.9179 - accuracy: 0.7011 - val_loss: 0.9471 - val_accuracy: 0.70		
epoch 6/100		
29/29 [=====] - 8s 265ms/step - loss: 0.8701 - accuracy: 0.7233 - val_loss: 0.9235 - val_accuracy: 0.68		
epoch 7/100		
29/29 [=====] - 8s 333ms/step - loss: 0.8169 - accuracy: 0.7511 - val_loss: 0.9675 - val_accuracy: 0.70		
epoch 8/100		
29/29 [=====] - 8s 273ms/step - loss: 0.7735 - accuracy: 0.7711 - val_loss: 0.8869 - val_accuracy: 0.70		
epoch 9/100		
29/29 [=====] - 8s 280ms/step - loss: 0.7284 - accuracy: 0.7800 - val_loss: 0.9274 - val_accuracy: 0.72		
epoch 10/100		
29/29 [=====] - 8s 268ms/step - loss: 0.7008 - accuracy: 0.7864 - val_loss: 0.8872 - val_accuracy: 0.73		
epoch 11/100		
29/29 [=====] - 8s 277ms/step - loss: 0.6527 - accuracy: 0.7909 - val_loss: 0.9322 - val_accuracy: 0.73		
epoch 12/100		
29/29 [=====] - 8s 271ms/step - loss: 0.6394 - accuracy: 0.8011 - val_loss: 0.9593 - val_accuracy: 0.70		
epoch 13/100		
...		
Confusion Matrix RNN		
[[18 0 1 1 6]		
[5 0 0 0 1]		
[2 0 0 0 3]		
[0 0 0 0 6]		
[6 0 0 2 49]]		
	precision	recall f1-score support
class 0	0.58	0.69 0.63 26
class 1	0.00	0.00 0.00 6
class 2	0.00	0.00 0.00 5
class 3	0.00	0.00 0.00 6
class 4	0.75	0.86 0.80 57
accuracy		0.67 100
macro avg	0.27	0.31 0.29 100
weighted avg	0.58	0.67 0.62 100

GojekD App Sentiment RNN



V. Kesimpulan

Dari penelitian yang dilakukan, kami menyimpulkan bahwa dengan menggunakan empat metode klasifikasi yaitu, KNN, Random Forest, Binary Classification, dan RNN untuk menganalisis sentimen review pengguna aplikasi Gojek di Playstore. Hasil dari penelitian ini memperoleh persentase sentimen positif sebesar 82,58% dan persentase komentar negatif sebesar 17,42%. Dari besar persentase, dapat dilihat bahwa mayoritas pengguna aplikasi Gojek memberikan review yang positif dan sedikit dari pengguna aplikasi Gojek memberikan review yang negatif. Sehingga, dapat disimpulkan bahwa Gojek merupakan aplikasi yang dapat membuat pelanggan puas dengan kualitas dan layanannya.

REFRENSI

M. Saifurrido, Martanto, U. Hayati, "ANALISIS ALGORITMA K-NEAREST NEIGHBOR TERHADAP SENTIMEN PENGGUNA APLIKASI SHOPEE", Vol. 10, No. 1, 2024.

<https://journal.nurulfikri.ac.id/index.php/JIT/article/view/1054>

E. Fitri, Y. Yuliani, S.Rosyida, W. Gata, "ANALISIS SENTIMEN TERHADAP APLIKASI RUANGGURU MENGGUNAKAN ALGORITMA NAÏVE BAYES, RANDOM FOREST, DAN SUPPORT VECTOR MACHINE", Vol. 18, No. 1, July 2020.

<https://journals.usm.ac.id/index.php/transformatika/article/view/2317>

C. Morama, D. E. Ratnawati, I. Arwani. "ANALISIS SENTIMEN BERBASIS ASPEK TERHADAP ULASAN HOTEL TENTREM YOGYAKARTA MENGGUNAKAN ALGORITMA RANDOM FOREST CLASSIFIER" Vol. 6, No. 4, April 2022.

<https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/download/10908/4809>

R. Sanusi, F. D. Astuti, I. Y. Buryadi. "ANALISIS SENTIMEN PADA TWITTER TERHADAP PROGRAM KARTU PRA KERJA DENGAN RECURRENT NEURAL NETWORK", Vol. 5, No. 2, Maret 2021.

<https://ejournal.akakom.ac.id/index.php/jiko/article/view/645/157>