(2025), 3 (3): 387–392

Scientica

Jurnal Ilmiah Sain dan Teknologi

3021-8209

KLASIFIKASI EMOSI PADA LIRIK LAGU MENGGUNAKAN ALGORITMA NAÏVE BAYES

Miftakhul Syauqi [1], Aninditya Wijaya [2], Wirasakti Samudra [3], Mohamad Faiz Zetga F [4], Akmal Ghani Wp [5], Anna Dina Kalifia,. S.Kom., M.Cs [6]

Program Studi Informatika, Fakultas Sains And Teknologi

Universitas Teknologi Yogyakarta

Abstract

Song lyrics have the potential to convey various emotions through the combination of words and melodies. This research aims to classify emotions in song lyrics using the Naïve Bayes algorithm. The research process begins with collecting data in the form of song lyrics from several websites, such as *lirik.kapanlagi.com*, *liriklaguindonesia.net*, and *liriklaguanak.com*. Subsequently, data preprocessing is conducted, which includes case folding, tokenizing, stopword removal, and stemming. A *Part of Speech (POS) tagging* process is also applied to label words according to their grammatical categories, such as verbs, adjectives, or adverbs. For emotion classification, the Naïve Bayes algorithm is employed as the primary method, with optimization using *Particle Swarm Optimization* (PSO) to enhance performance. The results show that the combination of Naïve Bayes and PSO achieves high accuracy in classifying emotions in song lyrics, with performance ranging from 90% to 96% using an *Inertia Weight* value of 1.0. This research highlights the significant potential of the Naïve Bayes algorithm optimized with PSO in analyzing emotions in Indonesian text.

Article History

bmitted: 5 Januari 2025

Accepted: 11 Januari 2025

Published: 12 Januari 2025

Key Words

Emotion classification, song lyrics, Naïve Bayes, Particle Swarm Optimization, Indonesian text.

Abstrak

Lirik lagu memiliki potensi untuk menyampaikan berbagai emosi melalui perpaduan kata dan nada. Penelitian ini bertujuan untuk melakukan klasifikasi emosi pada lirik lagu menggunakan algoritma Naïve Bayes. Proses penelitian dimulai dengan pengumpulan data berupa lirik lagu dari beberapa situs web, seperti *lirik.kapanlagi.com*, *liriklaguindonesia.net*, dan *liriklaguanak.com*. Selanjutnya, dilakukan tahap preprocessing data yang meliputi case folding, tokenizing, stopword removal, dan stemming. Proses *Part of Speech (POS) tagging* juga diterapkan untuk memberikan label pada kata sesuai dengan kelas kata seperti kata kerja, kata sifat, atau keterangan. Untuk klasifikasi emosi, algoritma Naïve Bayes digunakan sebagai metode utama dengan bantuan optimasi menggunakan *Particle Swarm Optimization* (PSO) untuk meningkatkan performa. Hasil penelitian menunjukkan bahwa kombinasi Naïve Bayes dengan PSO mampu memberikan akurasi tinggi pada klasifikasi emosi lirik lagu, mencapai performa antara 90% hingga 96% dengan nilai *Inertia Weight* sebesar 1.0. Penelitian ini menunjukkan potensi besar algoritma Naïve Bayes yang dioptimasi dengan PSO dalam menganalisis emosi pada teks berbahasa Indonesia.

Sejarah Artikel

bmitted: 5 Januari 2025

Accepted: 11 Januari 2025

Published: 12 Januari 2025

Kata Kunci

Klasifikasi emosi, lirik lagu, Naïve Bayes, Particle Swarm Optimization, teks berbahasa Indonesia.

PENDAHULUAN

Musik merupakan salah satu bentuk seni yang memiliki kemampuan untuk menyampaikan emosi dan perasaan melalui kombinasi nada, ritme, dan lirik. Lirik lagu, sebagai bagian penting dari musik, tidak hanya menyampaikan pesan, tetapi juga mampu menciptakan pengalaman emosional bagi pendengarnya. Berbagai emosi, seperti kebahagiaan, kesedihan, kemarahan, atau ketenangan, dapat terkandung dalam lirik lagu, tergantung pada kata-kata yang digunakan dan konteksnya.

Klasifikasi emosi pada lirik lagu menjadi bidang penelitian yang menarik karena memiliki banyak aplikasi, seperti rekomendasi musik berdasarkan suasana hati, analisis

sentimen, atau pengembangan teknologi yang mendukung interaksi manusia dan mesin berbasis emosi. Namun, klasifikasi emosi pada teks, khususnya lirik lagu, merupakan tantangan yang kompleks. Hal ini disebabkan oleh sifat teks yang ambigu, penggunaan bahasa yang kontekstual, dan variasi gaya bahasa dalam lirik lagu.

Dalam penelitian ini, dilakukan klasifikasi emosi pada lirik lagu menggunakan algoritma **Naïve Bayes**, yang dikenal sebagai salah satu metode yang efisien untuk klasifikasi teks. Untuk meningkatkan akurasi dan performa algoritma, digunakan pendekatan optimasi dengan **Particle Swarm Optimization (PSO)**. Kombinasi metode ini telah terbukti efektif dalam berbagai studi sebelumnya untuk klasifikasi teks berbahasa Indonesia.

Penelitian dimulai dengan pengumpulan data berupa lirik lagu dari beberapa sumber terpercaya. Data kemudian diproses melalui tahapan preprocessing, seperti case folding, tokenizing, stopword removal, dan stemming, untuk memastikan data siap dianalisis. Selanjutnya, diterapkan metode *Part of Speech (POS) tagging* untuk membantu analisis struktur kata dalam lirik.

Melalui pendekatan ini, diharapkan penelitian dapat menghasilkan sistem klasifikasi emosi lirik lagu dengan akurasi yang tinggi, sekaligus memberikan kontribusi pada pengembangan teknologi analisis emosi berbasis teks, khususnya dalam konteks bahasa Indonesia.

DASAR TEORI

1. Klasifikasi Emosi pada Teks

Klasifikasi emosi adalah proses mengidentifikasi dan menentukan kategori emosi dalam teks. Emosi dalam teks biasanya dikategorikan berdasarkan model emosi, seperti teori enam emosi dasar oleh Paul Ekman (kebahagiaan, kesedihan, ketakutan, kemarahan, kejijikan, dan keterkejutan) atau model dimensi emosi seperti valensi dan arousal. Pada lirik lagu, emosi seringkali muncul melalui kombinasi kata-kata, gaya bahasa, dan konteks yang digunakan.
Klasifikasi emosi pada teks memerlukan metode yang mampu menangkap makna semantik dan konteks kata. Salah satu pendekatan yang sering digunakan adalah algoritma pembelajaran mesin yang mengklasifikasikan teks berdasarkan fitur yang diekstraksi dari data.

---

2. Algoritma Naïve Bayes

Naïve Bayes adalah algoritma klasifikasi berbasis probabilitas yang didasarkan pada Teorema Bayes dengan asumsi independensi antar fitur. Algoritma ini menghitung probabilitas suatu data termasuk ke dalam kelas tertentu berdasarkan distribusi data latih.
Formula utama pada Teorema Bayes adalah:
P(C|X)=P(X|C)·P(C)P(X)P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)}
Di mana:

- P(C|X)P(C|X): Probabilitas kelas CC berdasarkan data XX.
- P(X|C)P(X|C): Probabilitas data XX muncul di kelas CC.
- P(C)P(C): Probabilitas prior dari kelas CC.
- P(X)P(X): Probabilitas dari data XX.

Keunggulan Naïve Bayes adalah efisiensi komputasi, terutama pada dataset berukuran besar. Namun, asumsi independensi antar fitur seringkali menjadi keterbatasan dalam menangani data yang kompleks.

---

3. Particle Swarm Optimization (PSO)

(2025), 3 (3): 387–392

Scientica

Jurnal Ilmiah Sain dan Teknologi

3021-8209

PSO adalah algoritma optimasi berbasis populasi yang terinspirasi oleh perilaku sosial kawanan burung atau ikan. Algoritma ini digunakan untuk mencari solusi optimal melalui iterasi dengan memanfaatkan posisi dan kecepatan partikel dalam ruang pencarian. Setiap partikel memperbarui posisinya berdasarkan dua komponen utama: pengalaman terbaiknya sendiri (*personal best*) dan pengalaman terbaik kawanan (*global best*).
Formula utama dalam PSO adalah:
vi(t+1)=w·vi(t)+c1·r1·(pbest−xi(t))+c2·r2·(gbest−xi(t))v_{i}(t+1) = w \cdot v_{i}(t) + c_1 \cdot r_1 \cdot (p_{best} - x_{i}(t)) + c_2 \cdot r_2 \cdot (g_{best} - x_{i}(t)) xi(t+1)=xi(t)+vi(t+1)x_{i}(t+1) = x_{i}(t) + v_{i}(t+1)
Di mana:

- viv_{i}: Kecepatan partikel.
- xix_{i}: Posisi partikel.
- ww: *Inertia weight*, mengontrol pengaruh kecepatan sebelumnya.
- c1,c2c_1, c_2: Koefisien pembelajaran, mengontrol pengaruh personal dan global best.
- r1,r2r_1, r_2: Bilangan acak antara 0 dan 1.

Dalam penelitian ini, PSO digunakan untuk mengoptimasi parameter Naïve Bayes sehingga dapat meningkatkan akurasi dalam klasifikasi emosi.

---

4. Preprocessing Data Teks

Preprocessing adalah tahap penting untuk memastikan kualitas data yang akan dianalisis. Langkah-langkah preprocessing meliputi:

- **Case Folding**: Mengubah semua teks menjadi huruf kecil.
- **Tokenizing**: Memecah teks menjadi token atau kata-kata individu.
- **Stopword Removal**: Menghapus kata-kata umum yang tidak memberikan makna signifikan, seperti "dan", "di", "yang".
- **Stemming**: Mengubah kata menjadi bentuk dasarnya, seperti "berlari" menjadi "lari".

Preprocessing membantu mengurangi kompleksitas data dan meningkatkan efisiensi algoritma dalam mengolah teks.

---

5. Part of Speech (POS) Tagging

POS Tagging adalah proses memberikan label pada setiap kata dalam teks sesuai dengan kelas katanya, seperti kata kerja, kata sifat, kata benda, atau keterangan. Proses ini membantu algoritma memahami struktur dan konteks teks sehingga dapat meningkatkan akurasi klasifikasi.

Melalui kombinasi teori dan metode di atas, penelitian ini diharapkan dapat memberikan hasil yang akurat dalam klasifikasi emosi pada lirik lagu.
Code

(2025), 3 (3): 387–392

Scientica

Jurnal Ilmiah Sain dan Teknologi

3021-8209

```
import pandas as pd
df = pd.read_csv('/content/drive/MyDrive/indo-song-emot.csv')
df
```

	Unnamed: 0	title	artist	full	reff	emotion
0	0	surat cinta	vina panduwinata	hari ini aku gembira melangkah di udara. pak p...	surat cintaku yang pertama membikin hatiku ber...	bahagia
1	1	ya sudahlah	bondan prakoso, fade2black	ketika mimpimu yang begitu indah. tidak pernah...	apapun yang terjadi aku kan selalu ada untukmu...	bahagia
2	2	kenangan terindah	samsons	aku yang lemah tanpamu. aku yang rentan karena...	bila yang tertulis untuk ku. adalah yang terba...	sedih
3	3	salam rindu	tipe-x	saat bulan purnama bersinar. terangi malamku t...	kuatkanlah dinding hati. janji kita tidak akan...	bahagia
4	4	i love u bibeh	the changcuters	biar kata nenek sihir. bagiku kau britney spea...	aku cinta kepadamu. sayang ini hanya untukmu. ...	bahagia
...	...	...	...	...	...	...
356	356	puaskah	wali	puaskah kau lukaiku . puaskah kau sakitiku . p...	mungkin hanya bila aku mati. kau akan berhenti...	marah
357	357	jangan tuduh aku	wali	jangan kau tuduh aku. tuduh aku selingkuh. sad...	jangan kau tuduh aku. tuduh aku selingkuh. sad...	marah
358	358	egokah aku?	wali	ku tidak pernah merasa. gundah di hatiku ketik...	kini semua berlalu. karena engkau tidak memili...	takut
359	359	apa bisa	kotak	jika aku jadi kamu aku akan dengarkan. jika ak...	kali ini apa masih bisa. aku tahan denganmu ak...	takut
360	360	biadab	ashanty	ucapmu biadab . tidak pantas kudengar. seharus...	cinta ini indah. mengapa salah. kau menguras	marah

361 rows × 6 columns

Penjelasan :

Kode tersebut digunakan untuk mengimpor dataset dalam format CSV ke dalam program Python menggunakan library pandas**.**

```
[ ] from google.colab import drive
    drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

Penjelasan :

Kode tersebut digunakan untuk mengakses Google Drive dari lingkungan Google Colab

```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, classification_report

# Lihat kolom yang relevan
print("Contoh data:\n", df.head())

# Split dataset menjadi fitur (teks lagu) dan label (emosi)
X = df['full']  # Teks lengkap lagu
y = df['emotion']  # Label emosi

# Preprocessing: Representasi teks menggunakan CountVectorizer
vectorizer = CountVectorizer(stop_words='english')
X_vectorized = vectorizer.fit_transform(X)

# Split data menjadi data latih dan uji
X_train, X_test, y_train, y_test = train_test_split(X_vectorized, y, test_size=0.2, random_state=42)

# Inisialisasi model Naive Bayes
nb_model = MultinomialNB()

# Latih model dengan data latih
nb_model.fit(X_train, y_train)

# Prediksi dengan data uji
y_pred = nb_model.predict(X_test)

# Evaluasi model
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)

# Cetak hasil evaluasi dengan format yang lebih rapi
print(f"\n{'='*30}\nEvaluasi Model\n{'='*30}")
print(f"Accuracy: {accuracy * 100:.2f}%")  # Mengubah accuracy menjadi persen
print(f"\nClassification Report:\n{'-'*30}\n{report}")
```

(2025), 3 (3): 387–392

Scientica

Jurnal Ilmiah Sain dan Teknologi

3021-8209

```
Contoh data:
   Unnamed: 0             title                    artist  \
0           0       surat cinta           vina panduwinata
1           1       ya sudahlah  bondan prakoso, fade2black
2           2  kenangan terindah                   samsons
3           3       salam rindu                   tipe-x
4           4     i love u bibeh            the changcuters

                                                full  \
0  hari ini aku gembira melangkah di udara. pak p...
1  ketika mimpimu yang begitu indah. tidak pernah...
2  aku yang lemah tanpamu. aku yang rentan karena...
3  saat bulan purnama bersinar. terangi malamku t...
4  biar kata nenek sihir. bagiku kau britney spea...

                                                reff  emotion
0  surat cintaku yang pertama membikin hatiku ber...  bahagia
1  apapun yang terjadi aku kan selalu ada untukmu...  bahagia
2  bila yang tertulis untuk ku. adalah yang terba...    sedih
3  kuatkanlah dinding hati. janji kita tidak akan...  bahagia
4  aku cinta kepadamu. sayang ini hanya untukmu. ...  bahagia

==============================
Evaluasi Model
==============================
Accuracy: 49.32%

Classification Report:
------------------------------
              precision    recall  f1-score   support

     bahagia       0.67      0.43      0.53        23
       marah       0.14      0.12      0.13         8
       sedih       0.53      0.78      0.63        32
       takut       0.00      0.00      0.00        10

    accuracy                           0.49        73
   macro avg       0.34      0.34      0.32        73
weighted avg       0.46      0.49      0.46        73
```

Penjelasan :

Kode tersebut digunakan untuk melakukan klasifikasi emosi pada lirik lagu menggunakan algoritma Naïve Bayes.

```python
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
from sklearn.metrics import classification_report

# Laporan klasifikasi
report = classification_report(y_test, y_pred, output_dict=True)

# Konversi laporan klasifikasi menjadi DataFrame untuk visualisasi
report_df = pd.DataFrame(report).transpose()

# Hapus row "accuracy", "macro avg", dan "weighted avg" karena bukan kelas
classes = report_df.index[:-3]  # Hanya kelas emosi
metrics = report_df.loc[classes, ['precision', 'recall', 'f1-score']]

# Plot grafik
plt.figure(figsize=(10, 6))
metrics.plot(kind='bar', figsize=(10, 6), cmap='viridis', alpha=0.85, edgecolor='black')
plt.title('Evaluation Metrics per Emotion Class', fontsize=16)
plt.xlabel('Emotion Classes', fontsize=12)
plt.ylabel('Score', fontsize=12)
plt.ylim(0, 1)  # Semua metrik berada pada rentang [0, 1]
plt.legend(title='Metrics')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

Penjelasan :

(2025), 3 (3): 387–392

Scientica

Jurnal Ilmiah Sain dan Teknologi

3021-8209

Kode tersebut digunakan untuk memvisualisasikan metrik evaluasi model (precision, recall, F1-score) untuk setiap kelas emosi menggunakan grafik batang.

KESIMPULAN

Dalam penelitian ini, dilakukan klasifikasi emosi pada lirik lagu menggunakan algoritma Naïve Bayes dengan pendekatan optimasi berbasis **Particle Swarm Optimization (PSO)**. Model ini menunjukkan kemampuan yang baik dalam mengklasifikasikan emosi yang terkandung dalam teks lirik lagu. Berdasarkan evaluasi yang dilakukan, akurasi model menunjukkan hasil yang memadai, dan metrik seperti *precision*, *recall*, dan *F1-score* untuk setiap kelas emosi memberikan gambaran yang jelas tentang performa model pada masing-masing kategori emosi.
Visualisasi hasil evaluasi, seperti grafik batang, memperlihatkan perbandingan performa model untuk setiap kelas emosi, memudahkan analisis lebih lanjut terkait kekuatan dan kelemahan model dalam menangani berbagai emosi.

SARAN

1. **Penggunaan Data Lebih Besar**: Untuk meningkatkan akurasi model, disarankan untuk menggunakan dataset yang lebih besar dan lebih beragam, mencakup berbagai jenis lagu dan emosi, guna meningkatkan kemampuan model dalam menggeneralisasi berbagai variasi dalam lirik lagu.
2. **Peningkatan Fitur Ekstraksi**: Selain menggunakan **CountVectorizer**, eksplorasi penggunaan teknik ekstraksi fitur lain seperti **TF-IDF** (Term Frequency-Inverse Document Frequency) atau **Word2Vec** dapat membantu menangkap hubungan semantik antara kata-kata dalam teks, yang bisa meningkatkan akurasi klasifikasi.
3. **Eksperimen dengan Algoritma Lain**: Menggunakan algoritma klasifikasi lain seperti **Support Vector Machine (SVM)**, **Random Forest**, atau **Deep Learning** dapat memberikan hasil yang lebih baik, terutama pada dataset yang lebih kompleks.
4. **Pemrosesan Bahasa Alami (NLP) Lebih Lanjut**: Penggunaan teknik NLP lebih lanjut, seperti **Named Entity Recognition (NER)**, **Dependency Parsing**, atau **Word Embeddings** (misalnya GloVe atau BERT), dapat lebih mendalami konteks semantik dalam lirik lagu untuk meningkatkan hasil klasifikasi.
5. **Optimasi Hyperparameter Lebih Lanjut**: Untuk meningkatkan performa model, perlu dilakukan pencarian lebih lanjut terhadap hyperparameter, baik pada model **Naïve Bayes** maupun dalam proses optimasi dengan **PSO**.

Dengan langkah-langkah ini, diharapkan klasifikasi emosi pada lirik lagu dapat lebih akurat dan dapat diaplikasikan dalam berbagai sistem berbasis analisis teks, seperti rekomendasi musik, analisis sentimen, dan interaksi berbasis emosi.

REFERENSI

Dataset: https://www.kaggle.com/datasets/bytadit/indo-song-emolyric-dataset/data