ISSN: 1829-667X / E-ISSN: 2460-9021
DOI:10.31515/telematika.v21i2. 10867

Classification of Indonesian Tale Categories using Support Vector Machine and FastText Feature Extraction

Klasifikasi Kategori Dongeng Indonesia Menggunakan Metode *Support Vector Machine* dan Ekstraksi Fitur *FastText*

Helena Nurramdhani Irmanda[1], Ria Astriratma[2], Ati Zaidiah[3], Muhammad Rahman Hadi[4], Nayandra Agastia Putra[5]

[1,2,3] Sistem Informasi, Universitas Pembangunan Nasional Veteran Jakarta, Indonesia

[1*]helenairmanda@upnvj.ac.id, [2]astriratma@upnvj.ac.id, [3]atizaidiah@upnvj.ac.id, [4]2110512047@mahasiswa.upnvj.ac.id**,** [5]2110512051@mahasiswa.upnvj.ac.id

Informasi Artikel
Received: January 2024
Revised: April 2024
Accepted: May 2024
Published: June 2024

Abstract

Purpose: The purpose of this work is to develop a model to classify the various kinds of Indonesian folktales and to assess how well the support vector machine (SVM) approach and fastText feature extraction perform.
Design/methodology/approach: The first phase of the study process is the gathering of data, namely the fairy tale dataset that has been annotated with categorizations for each genre of fairy tale. Following the collection of data, the pre-processing step is conducted. The purpose of the pre-processing step is to prepare the data for further processing in the subsequent stage. Following the completion of the preprocessing step, the training data and testing data are segregated. The subsequent step involves doing feature extraction using fastText. Moreover, the classification process is conducted using the Support Vector Machine (SVM) approach in order to get the ultimate outcome of the modeling process. The last phase involves assessing the performance of the constructed model.
Findings/result: The accuracy of the classification model for Indonesian fairy tale categories is 85%. This result aligns with a precision of 85%, recall of 85%, and an F1-score of 86%, all of which indicate consistent performance.
Originality/value/state of the art: Previous researchs have not conducted any studies on the categorization of types of Indonesian fairy tales.

Abstrak

Tujuan: Tujuan dari penelitian ini adalah untuk membuat model yang dapat mengkategorikan jenis dongeng di

Keywords:one; two; three
Kata kunci: satu; dua; tiga

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI:10.31515/telematika.v21i2. 10867

Indonesia serta menganalisis kinerja metode *support vector machines* (SVM) dan ekstraksi fitur *fastext*.
Perancangan/metode/pendekatan: Proses penelitian dimulai dengan pengumpulan kumpulan dongeng yang telah dilabeli dengan jenis dongeng masing-masing. Setelah pengumpulan data, dilakukan tahap praproses yang bertujuan untuk menyiapkan data untuk diolah pada tahap berikutnya.. Setelah melakukan tahapan preprocessing, maka dilakukan pemisahan data *training* dan data *testing*. Tahapan berikutnya adalah melakukan ekstraksi fitur dengan *fastext.* Selanjutnya, klasifikasi dilakukan dengan metode SVM untuk mendapatkan hasil permodelan. Tahap terakhir yaitu evaluasi performansi dari model yang telah dibuat.
Hasil: Akurasi model klasifikasi kategori dongeng Indonesia adalah 85%. Hasil ini sejalan dengan nilai presisi 85%, recall 85%, dan F1-score 86% yang juga menunjukkan hasil yang konsisten.
Keaslian/ *state of the art*: Penelitian mengenai klasifikasi kategori dongeng Indonesia belum pernah dilakukan oleh peneliti sebelumnya

1. Pendahuluan

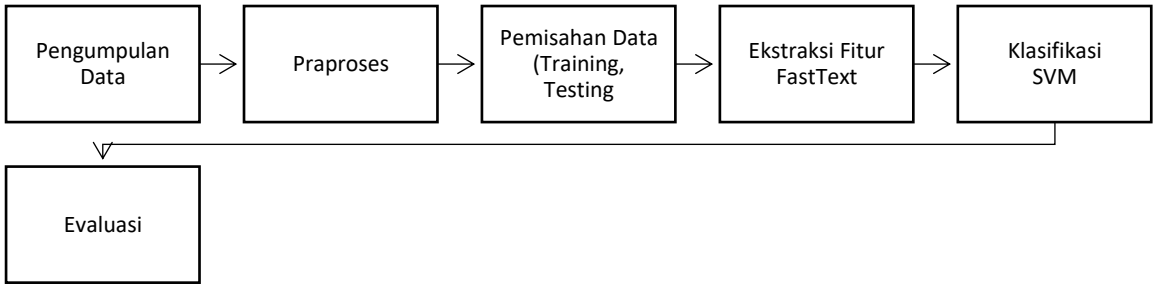
Dongeng adalah segala bentuk cerita yang diceritakan secara turun-temurun sejak dahulu [1]. Tradisi ini berkembang menjadi pengantar tidur yang sering dibacakan orang tua kepada anak-anak mereka. Cerita dongeng penuh dengan pesan moral yang dapat dipetik manfaatnya. Membaca dongeng dapat membantu anak-anak menjadi lebih tertarik untuk membaca, mendorong mereka untuk berpikir kritis dan kreatif, memberi pelajaran dengan cara yang tidak menggurui, serta mempererat hubungan antara anak dan orang tua [2]. Berdasarkan konsep-konsep tersebut, dongeng menjadi salah satu cara yang menyenangkan bagi anak untuk belajar karena memiliki cerita yang menarik sehingga anak-anak lebih mudah menyerap pengetahuan [3]. Di Indonesia, dongeng dibagi ke dalam beberapa kategori seperti fabel, legenda, mitos, sage, parabel, dan jenaka. Setiap kategori memiliki ciri khasnya masing-masing, namun masyarakat seringkali kebingungan membedakan antara kategori-kategori tersebut [4]. Penelitian sebelumnya telah mencoba mengklasifikasikan teks cerita, namun terdapat keterbatasan seperti penggunaan dataset yang kecil, kurangnya fitur representasi yang kaya, dan metode klasifikasi yang belum optimal [5]. Oleh karena itu, diperlukan suatu sistem yang lebih akurat dan efisien untuk mengklasifikasikan kategori dongeng di Indonesia.

Pendekatan text mining merupakan salah satu cara untuk mengklasifikasikan dongeng. Text mining digunakan untuk memperoleh informasi dari teks dengan memperhatikan pola dan tren menggunakan teknik statistik [6]. Proses klasifikasi teks melalui text mining dapat menggunakan berbagai metode seperti Naive Bayes, *k-nearest neighbor*, *decision tree*, dan *support vector machine* (SVM). Beberapa penelitian menemukan bahwa SVM menunjukkan performa terbaik dalam klasifikasi teks [7], namun tetap bergantung pada teknik ekstraksi fitur yang digunakan. Ekstraksi fitur merupakan proses penting dalam klasifikasi teks yang bertujuan

untuk mengubah format teks tidak terstruktur menjadi format yang terstruktur sehingga algoritma pembelajaran mesin dapat bekerja lebih efektif [8]. Salah satu teknik ekstraksi fitur yang handal adalah FastText. FastText adalah model yang digunakan untuk klasifikasi teks dan representasi kata yang efisien dan cepat berbasis model bag-of-words tradisional [9]. Keunggulan FastText terletak pada kemampuannya memperoleh informasi urutan kata dalam suatu kalimat hingga tingkat tertentu, memberikan representasi kalimat yang lebih akurat dibandingkan model *bag-of-words* tradisional [10]. Penelitian terdahulu menunjukkan bahwa kombinasi SVM dan FastText dapat meningkatkan kinerja klasifikasi teks dibandingkan dengan metode seperti TF-IDF [11]. Berdasarkan latar belakang ini, penelitian ini bertujuan untuk mengembangkan model klasifikasi kategori Dongeng Indonesia menggunakan metode *Support Vector Machine* (SVM) dan ekstraksi fitur FastText. Penelitian ini juga bertujuan untuk menganalisis performa model yang dihasilkan, dengan harapan memberikan solusi yang lebih akurat untuk klasifikasi kategori dongeng di Indonesia.

2. Metode/Perancangan

Metode penelitian yang digunakan melibatkan beberapa tahapan, yang mencakup pengumpulan data, praproses data, pemisahan data menjadi *dataset* pelatihan dan pengujian, ekstraksi fitur menggunakan FastText, dan klasifikasi menggunakan *Support Vector Machine* (SVM), serta Evaluasi. Rancangan tahapan ini dapat dilihat pada **Gambar 1**.



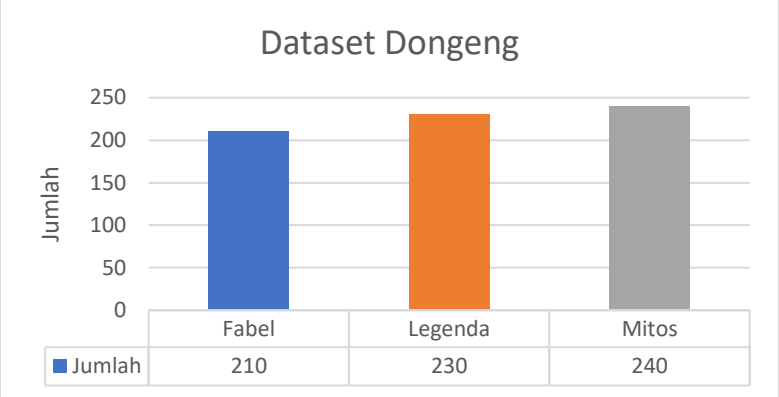
Gambar 1. Tahapan Pengembangan Model Kategori Dongeng Indonesia

Berdasarkan **Gambar 1,** tahapan ini mencakup proses mulai dari pengumpulan data hingga evaluasi model. Pada tahap pengumpulan data, teks dongeng dikumpulkan sebagai bahan untuk klasifikasi. Praproses data melibatkan pembersihan dan transformasi teks agar siap untuk analisis lebih lanjut. Data kemudian dibagi menjadi set pelatihan dan pengujian untuk menguji model pada data yang belum pernah dilihat selama pelatihan, sehingga memberikan gambaran lebih akurat tentang kinerja model pada data baru. Ekstraksi fitur dengan FastText digunakan untuk menghasilkan representasi numerik yang akan diproses oleh model SVM. Langkah terakhir adalah evaluasi performa model, yang bertujuan untuk menilai sejauh mana model dapat secara efektif mengklasifikasikan kategori dongeng yang berbeda.

2.1. Pengumpulan

Dataset dongeng yang sudah dilabeli dengan kategorinya masing-masing digunakan sebagai input sistem ini diantaranya Fabel, Legenda, dan Mitos. Dataset yang dikumpulkan sebanyak 680 data yang telah dilabelkan yang terdiri atas 210 fabel, 230 legenda, dan 240 mitos. Proporsi data dongeng digambarkan pada **Gambar 2**.

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI:10.31515/telematika.v21i2. 10867



Gambar 2. Proporsi Dataset Dongeng

Pada Tabel 1 dideskripsikan contoh data dongeng yang telah diberikan label.

Tabel 1. Contoh Dataset Dongeng

Cerita	Kategori
Seekor Kelinci sombong selalu menyombongkan kecepatannya kepada hewan-hewan lain. Kura-Kura, yang lambat tapi cerdas, menantang Kelinci untuk balapan. Kelinci setuju dengan yakin akan menang dengan mudah. Namun, dia terlalu percaya diri sehingga dia memutuskan untuk tidur sejenak selama balapan. Kura-Kura berlari perlahan tapi stabil hingga akhirnya mencapai garis finish pertama kali. Kelinci terbangun dan menyadari kesalahannya. Dari situlah dia belajar untuk tidak sombong dan menghargai setiap tantangan.	0 (Fabel)
Asal usul Gunung Papandayan, gunung berapi di Jawa Barat, berkaitan dengan legenda yang menceritakan tentang Ki Santang, seorang pangeran yang kuat dan bijaksana. Ia jatuh cinta pada Dewi Rengganis, putri raja dari Kerajaan Salakanagara. Namun, cinta mereka ditentang oleh Raja Salakanagara, dan perang pun tak terhindarkan. Ki Santang tewas dalam pertempuran, tetapi sebelum meninggal, ia mengutuk gunung yang kini dikenal sebagai Gunung Papandayan. Gunung ini meletus, menghancurkan sekitarnya, dan mengubur wilayah tersebut di bawah abu vulkanik. Legenda ini melambangkan pengorbanan cinta dan hubungan manusia dengan alam yang kuasa.	1 (Legenda)
Dewi Sri mengisahkan tentang Dewi yang melambangkan dewi hasil bumi dan pertanian dalam mitologi Jawa. Dewi Sri dikenal sebagai penjaga panen dan keberlimpahan. Dikatakan bahwa setiap kali panen sukses, Dewi Sri menghuni lumbung-lumbung padi dan memberikan keberuntungan kepada petani. Cerita ini mengajarkan nilai-nilai ketergantungan manusia pada alam, keberlimpahan, dan pentingnya menghormati alam serta mengelola sumber daya dengan bijaksana.	2 (Mitos)

2.2. Praproses

Praproses dalam konteks penambangan teks merujuk pada proses transformasi teks sebelum analisis. Proses ini melibatkan identifikasi unit (misalnya, kata dan frasa) yang akan digunakan (tokenisasi), menghapus konten yang tidak relevan [12]. Tahap praproses pada penelitian ini terdiri atas beberapa sub tahapan antara lain case folding, tokenization, *stopword removal*, dan *stemming*.

2.2.1. Case Folding

Case folding bertujuan mengkonversi karakter pada teks dongeng menjadi huruf kecil (*lowercase*) atau huruf besar (*uppercase*) untuk mengatasi perbedaan dalam penulisan huruf besar dan kecil dalam teks, sehingga memungkinkan pembandingan dan pencarian teks yang lebih konsisten [13].

Tabel 2. Hasil sebelum dan sesudah *case folding*

Sebelum Case Folding	**Setelah Case Folding**
Seekor Kelinci sombong selalu menyombongkan kecepatannya kepada hewan-hewan lain. Kura-Kura, yang lambat tapi cerdas, menantang Kelinci untuk balapan. Kelinci setuju dengan yakin akan menang dengan mudah. Namun, dia terlalu percaya diri sehingga dia memutuskan untuk tidur sejenak selama balapan. Kura-Kura berlari perlahan tapi stabil hingga akhirnya mencapai garis finish pertama kali. Kelinci terbangun dan menyadari kesalahannya. Dari situlah dia belajar untuk tidak sombong dan menghargai setiap tantangan.	seekor kelinci sombong selalu menyombongkan kecepatannya kepada hewan-hewan lain. kura-kura, yang lambat tapi cerdas, menantang kelinci untuk balapan. kelinci setuju dengan yakin akan menang dengan mudah. namun, dia terlalu percaya diri sehingga dia memutuskan untuk tidur sejenak selama balapan. kura-kura berlari perlahan tapi stabil hingga akhirnya mencapai garis finish pertama kali. kelinci terbangun dan menyadari kesalahannya. dari situlah dia belajar untuk tidak sombong dan menghargai setiap tantangan.

2.2.2. Special Text and Number Removal

Proses ini bertujuan untuk menghapus spesial teks seperti emoji dan simbol serta angka dari sebuah teks. Langkah-langkah yang bisa dilakukan adalah identifikasi karakter yang termasuk emoji dan symbol, angka termasuk karakter special seperti tab, enter, dll menggunakan ekspresi reguler. Pada **Tabel 3** dicontohkan proses *Special Text* and *Number removal* yang dilakukan.

Tabel 3. Hasil sebelum dan sesudah *Special Text* and *Number removal*

Sebelum *Special Text dan Number Removal*	**Setelah *Special Text dan Number Removal***
seekor kelinci sombong selalu menyombongkan kecepatannya kepada hewan-hewan lain. kura-kura, yang lambat tapi cerdas, menantang kelinci untuk balapan. kelinci setuju dengan yakin akan menang dengan mudah. namun, dia terlalu percaya diri sehingga dia memutuskan untuk tidur sejenak selama balapan. kura-kura berlari perlahan tapi stabil hingga akhirnya mencapai garis finish pertama kali. kelinci terbangun dan menyadari kesalahannya. dari situlah dia belajar untuk tidak sombong dan menghargai setiap tantangan.	seekor kelinci sombong selalu menyombongkan kecepatannya kepada hewan-hewan lain. kura-kura, yang lambat tapi cerdas, menantang kelinci untuk balapan. kelinci setuju dengan yakin akan menang dengan mudah. namun, dia terlalu percaya diri sehingga dia memutuskan untuk tidur sejenak selama balapan. kura-kura berlari perlahan tapi stabil hingga akhirnya mencapai garis finish pertama kali. kelinci terbangun dan menyadari kesalahannya. dari situlah dia belajar untuk tidak sombong dan menghargai setiap tantangan.

2.2.3. Punctuation Removal

Proses ini bertujuan untuk menghapus tanda baca atau menghilangkan simbol-simbol tertentu dari teks. Tujuannya adalah untuk menghilangkan *noise* dan mempersiapkan data untuk analisis lebih lanjut. Misalnya, tanda baca seperti tanda seru atau tanda tanya dapat dihapus karena tidak memberikan informasi penting saat mengklasifikasikan teks[14]. Hasil sebelum dan sesudah proses *punctuation removal* ditunjukan pada **Tabel 4**.

Tabel 4. Hasil sebelum dan sesudah *Punctuation Removal*

Sebelum *Punctuation Removal*	**Setelah *Punctuation Removal***
seekor kelinci sombong selalu menyombongkan kecepatannya kepada hewan-hewan lain. kura-kura, yang lambat tapi cerdas, menantang kelinci untuk balapan. kelinci setuju dengan yakin akan menang dengan mudah. namun, dia terlalu percaya diri sehingga dia memutuskan untuk tidur sejenak selama balapan. kura-kura berlari perlahan tapi stabil hingga akhirnya mencapai garis finish pertama kali. kelinci terbangun dan menyadari kesalahannya. dari situlah	seekor kelinci sombong selalu menyombongkan kecepatannya kepada hewan-hewan lain kura-kura yang lambat tapi cerdas, menantang kelinci untuk balapan kelinci setuju dengan yakin akan menang dengan mudah namun, dia terlalu percaya diri sehingga dia memutuskan untuk tidur sejenak selama balapan kura-kura berlari perlahan tapi stabil hingga akhirnya mencapai garis finish pertama kali kelinci terbangun dan menyadari kesalahannya dari situlah dia

dia belajar untuk tidak sombong dan menghargai setiap tantangan.	belajar untuk tidak sombong dan menghargai setiap tantangan.

2.2.4. *Tokenization*

Tokenisasi adalah proses membagi teks menjadi "token". Token bisa berupa kata, frasa, atau bagian-bagian lain dari teks yang memiliki makna. Hasil sebelum dan sesudah proses *tokenization* ditunjukan pada **Tabel 5**.

Tabel 5. Hasil sebelum dan sesudah *Tokenization*

Sebelum *Tokenization*	**Setelah *Tokenization***
seekor kelinci sombong selalu menyombongkan kecepatannya kepada hewan-hewan lain kura-kura yang lambat tapi cerdas, menantang kelinci untuk balapan kelinci setuju dengan yakin akan menang dengan mudah namun, dia terlalu percaya diri sehingga dia memutuskan untuk tidur sejenak selama balapan kura-kura berlari perlahan tapi stabil hingga akhirnya mencapai garis finish pertama kali kelinci terbangun dan menyadari kesalahannya dari situlah dia belajar untuk tidak sombong dan menghargai setiap tantangan.	['seekor', 'kelinci', 'sombong', 'selalu', 'menyomqbongkan', 'kecepatannya', 'kepada', 'hewanhewan', 'lain', 'kurakura', 'yang', 'lambat', 'tapi', 'cerdas', 'menantang', 'kelinci', 'untuk', 'balapan', 'kelinci', 'setuju', 'dengan', 'yakin', 'akan', 'menang', 'dengan', 'mudah', 'namun', 'dia', 'terlalu', 'percaya', 'diri', 'sehingga', 'dia', 'memutuskan', 'untuk', 'tidur', 'sejenak', 'selama', 'balapan', 'kurakura', 'berlari', 'perlahan', 'tapi', 'stabil', 'hingga', 'akhirnya', 'mencapai', 'garis', 'finish', 'pertama', 'kali', 'kelinci', 'terbangun', 'dan', 'menyadari', 'kesalahannya', 'dari', 'situlah', 'dia', 'belajar', 'untuk', 'tidak', 'sombong', 'dan', 'menghargai', 'setiap', 'tantangan']

2.2.5. Stopword Removal

Stopword removal merupakan tahap dalam pemrosesan bahasa alami (NLP) untuk menghapus kata-kata umum yang disebut "*stopword*" dari teks [13]. Stopword adalah kata-kata yang umumnya muncul dalam teks tetapi cenderung tidak memiliki nilai makna yang signifikan dalam analisis teks atau pemrosesan Bahasa. Contohnya "di", "dan", "yang", "dari", "ke", "untuk", "dengan". Hasil sebelum dan sesudah proses *stopword removal* ditunjukan pada **Tabel 6**.

Tabel 6. Hasil sebelum dan sesudah *Stopword Removal*

Sebelum *Stopword Removal*	**Setelah *Stopword Removal***
['seekor', 'kelinci', 'sombong', 'selalu', 'menyombongkan', 'kecepatannya', 'kepada', 'hewanhewan', 'lain', 'kurakura', 'yang', 'lambat', 'tapi', 'cerdas', 'menantang', 'kelinci', 'untuk', 'balapan', 'kelinci', 'setuju', 'dengan', 'yakin', 'akan', 'menang', 'dengan', 'mudah', 'namun', 'dia', 'terlalu', 'percaya', 'diri', 'sehingga', 'dia', 'memutuskan', 'untuk', 'tidur', 'sejenak', 'selama', 'balapan', 'kurakura', 'berlari', 'perlahan', 'tapi', 'stabil', 'hingga', 'akhirnya', 'mencapai', 'garis', 'finish', 'pertama', 'kali', 'kelinci', 'terbangun', 'dan', 'menyadari', 'kesalahannya', 'dari', 'situlah', 'dia', 'belajar', 'untuk', 'tidak', 'sombong', 'dan', 'menghargai', 'setiap', 'tantangan']	['seekor', 'kelinci', 'sombong', 'menyombongkan', 'kecepatannya', 'hewanhewan', 'kurakura', 'lambat', 'cerdas', 'menantang', 'kelinci', 'balapan', 'kelinci', 'setuju', 'menang', 'mudah', 'percaya', 'memutuskan', 'tidur', 'balapan', 'kurakura', 'berlari', 'perlahan', 'stabil', 'mencapai', 'garis', 'finish', 'kali', 'kelinci', 'terbangun', 'menyadari', 'kesalahannya', 'situlah', 'belajar', 'sombong', 'menghargai', 'tantangan']

2.2.6. *Stemming*

Stemming adalah teknik untuk menemukan kata dasar dari kata turunan dengan menghilangkan semua awalan, infiks, akhiran, dan konfiks, yang merupakan kombinasi awalan dan akhiran. [15]. Tujuan dari *stemming* adalah untuk mengurangi variasi kata ke bentuk dasarnya agar kata-

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI:10.31515/telematika.v21i2. 10867

kata dengan akar yang sama dapat dianggap sama secara kasar dalam analisis teks . Hasil sebelum dan sesudah proses *Stemming* ditunjukan pada **Tabel 7**.

Tabel 7. Hasil sebelum dan sesudah *Stemming*

Sebelum *Stemming*	**Setelah *Stemming***
['seekor', 'kelinci', 'sombong', 'menyombongkan', 'kecepatannya', 'hewanhewan', 'kurakura', 'lambat', 'cerdas', 'menantang', 'kelinci', 'balapan', 'kelinci', 'setuju', 'menang', 'mudah', 'percaya', 'memutuskan', 'tidur', 'balapan', 'kurakura', 'berlari', 'perlahan', 'stabil', 'mencapai', 'garis', 'finish', 'kali', 'kelinci', 'terbangun', 'menyadari', 'kesalahannya', 'situlah', 'belajar', 'sombong', 'menghargai', 'tantangan']	['seekor', 'kelinci', 'sombong', 'sombong', 'cepat', 'hewanhewan', 'kurakura', 'lambat', 'cerdas', 'tantang', 'kelinci', balap, 'kelinci', 'setuju', 'menang', 'mudah', 'percaya', 'putus', 'tidur', 'balap', 'kurakura', 'lari', 'perlahan', 'stabil', 'capai', 'garis', 'finish', 'kali', 'kelinci', 'bangun', 'sadar', 'salah', 'situ', 'ajar', 'sombong', 'harga', 'tantang']

2.3. Pemisahan Data

Pemisahan data merujuk pada pembagian data yang digunakan dalam model pembelajaran mesin. Data ini dibagi menjadi dua bagian: data *training* yang digunakan untuk membangun model dan data *testing* yang digunakan untuk mengevaluasi kemampuan prediksi model[16]. Tujuan utama dari pemisahan data ini adalah untuk menghindari *overfitting* (penyesuaian berlebihan) pada model yang dibangun dan untuk menguji kinerja model secara objektif. Pada penelitian ini, 20% data digunakan untuk pengujian dan 80% data digunakan untuk pelatihan. Pemisahan data menggunakan *library train test split*.

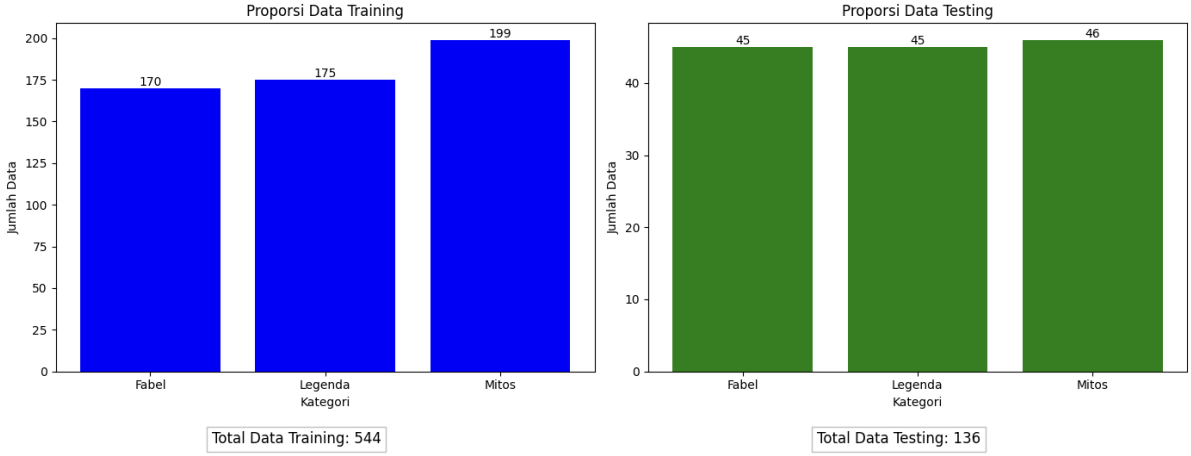
```
# Train Test Split Function
X = data[['Cerita','stemming']]
y = data['Kategori']
def split_train_test(data_df, test_size=0.2, shuffle_state=True):
    X_train, X_test, Y_train, Y_test = train_test_split(X, y, shuffle=shuffle_state, test_size=test_size, stratify=y, random_state=42)
    X_train = X_train.reset_index()
    X_test = X_test.reset_index()
    Y_train = Y_train.to_frame()
    Y_train = Y_train.reset_index()
    Y_test = Y_test.to_frame()
    Y_test = Y_test.reset_index()
    return X_train, X_test, Y_train, Y_test

# Call the train_test_split
X_train, X_test, Y_train, Y_test = split_train_test(data)
```

Gambar 3. Implementasi pemisahan data di Python

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI:10.31515/telematika.v21i2. 10867

Berdasarkan hasil pemisahan data didapatkan data *training* sebanyak 544 dan data testing *sebanyak* 136 dengan proporsi seperti yang ditampilkan pada **Gambar 4**.



Gambar 4. Proporsi setiap kategori pada data *training* dan data *testing*

2.4. Ekstraksi Fitur dengan FastText

Karena komputer tidak dapat mengolah data selain data numerik, ekstraksi fitur digunakan untuk menggali informasi potensial dan menampilkan kata-kata sebagai vektor fitur.. Ekstraksi fitur pada penelitian ini menggunakan metode FastText. Metode ini menghubungkan setiap karakter ngram dengan representasi vektor, sedangkan jumlah representasi vektor adalah jumlah kata. Selain itu, kata-kata yang memiliki akar yang sama akan berada lebih dekat satu sama lain. Dengan demikian, FastText memungkinkan pengelompokan kata-kata berdasarkan kemiripan leksikal, yang dapat meningkatkan efisiensi dan akurasi dalam model klasifikasi teks [17].

2.5. Klasifikasi SVM

Salah satu metode klasifikasi yang menggunakan prinsip pencarian *hyperplane* dengan margin terbesar adalah *Support Vector Machine* (SVM). Pada metode ini terdapat suatu garis yang memisahkan data antar kelas atau kategori disebut *hyperplane*. SVM dipilih untuk mengklasikfikasi kategori dongeng karena SVM merupakan algoritma yang sukses secara empiris dalam klasifikasi teks dan memiliki dasar teoritis yang kuat. Setelah praproses dan ekstraksi fitur, data traning digunakan untuk melatih model SVM. Proses ini memetakan data ke dalam ruang berdimensi-n dan mencari *hyperplane* yang dapat memisahkan data ke dalam kelas yang berbeda dengan margin maksimum[18].

2.6. Evaluasi

Selanjutnya, hasil klasifikasi dievaluasi untuk mendapatkan nilai akurasi, yang akan digunakan untuk menilai apakah model klasifikasi yang dibuat layak digunakan. Nilai akurasi berasal dari jumlah data uji yang benar, yang terdiri dari *True Positive* (TP) dan *True Negative* (TN), bersama dengan jumlah data uji keseluruhan.

$$Akurasi = \frac{TP + TN}{TP + TN + FN + FP} \quad (1)$$

Keterangan

TP: *True positive*

TN: True negative

FP: *False positive*

FN: *False negative*

Perhitungan dilakukan juga untuk *precision, recall* dan F1 *Score* untuk setiap kelas jenis dongeng dengan tujuan mengevaluasi keberhasilan model prediksi yang dapat dilihat pada Persamaan 9 [19], Persamaan 10 [19], dan Persamaan 11 [19].

$$Precision = \frac{TP}{TP+FP}\ x\ 100\% \quad (2)$$

$$Recall = \frac{TP}{TP+FN}\ x\ 100\% \quad (3)$$

$$F1 = \frac{2\ x\ Recall\ x\ Precision}{Recall+Precision} \quad (4)$$

3. Hasil dan Pembahasan

3.1. Hasil Ekstraksi Fitur FastText

FastText adalah salah satu algoritma pemrosesan bahasa alami untuk melakukan banyak tugas pemrosesan bahasa seperti klasifikasi teks, *clustering* kata, dan lainnya. Salah satu fitur utama dari *fastText* adalah kemampuannya untuk menghasilkan representasi vektor kata yang dapat digunakan dalam berbagai tugas pemrosesan bahasa. Pada penelitian ini, *FastText* sebagai ekstraksi fitur menerapkan gensim *library*, yang menyajikan perhitungan *FastText* untuk membentuk vektor yang berisikan bobot kemiripan. Dengan menggunakan Fungsi Gensim FastText, representasi data kata dalam bentuk vektor akan dihasilkan, seperti yang ditunjukkan dalam **Tabel 8**, yang mengandung kata dengan hasil vektor.

Tabel 8. Contoh nilai vektor hasil ekstraksi fitur dengan FastText

Kata	Vektor
legenda	-0.000482
danau	0.000166
berani	0.000165
bekas	0.000004

.

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI:10.31515/telematika.v21i2. 10867

3.2. Hasil Klasifikasi SVM

Proses klasifikasi teks dengan SVM melibatkan beberapa langkah penting. Pertama, model SVM diinisialisasi dengan pengaturan *kernel* RBF. Kernel RBF dipilih karena kemampuannya untuk menangani data yang tidak dapat dipisahkan secara linear. *Kernel* RBF memetakan data input ke ruang fitur berdimensi lebih tinggi, di mana sebuah *hyperplane* dapat lebih mudah memisahkan kategori-kategori yang berbeda. SVM bekerja dengan mencari *hyperplane* yang optimal untuk memisahkan data latih ke dalam kategori yang berbeda. *Hyperplane* ini adalah garis atau bidang yang membagi ruang fitur sehingga data dari kategori yang berbeda berada pada sisi yang berbeda dari *hyperplane* tersebut. Pada tahap ini, SVM akan mencoba untuk memaksimalkan *margin*, yaitu jarak terdekat antara titik data dari masing-masing kategori ke *hyperplane*. *Margin* yang lebih besar berarti model lebih yakin dalam memisahkan data.

Proses pelatihan melibatkan penyesuaian parameter model seperti penalti kesalahan klasifikasi dan koefisien *kernel*, untuk mengoptimalkan akurasi klasifikasi. SVM mempelajari pola dalam vektor fitur, menyesuaikan *hyperplane* hingga mencapai konfigurasi yang memisahkan data dengan akurasi terbaik. Setelah model dilatih dengan data latih, model SVM siap digunakan untuk mengklasifikasikan teks-teks baru berdasarkan fitur-fitur yang telah diekstraksi. Model ini dapat memprediksi kategori dari teks dongeng baru dengan menentukan sisi mana dari *hyperplane* teks tersebut berada. Implementasi klasifikasi ini dilakukan menggunakan pustaka *scikit-learn* dalam Python, yang mendukung SVM dengan berbagai kernel dan parameterisasi yang fleksibel. Kode untuk implementasi SVM dapat dilihat pada **Gambar 6**.

```
start_time = time.time()
clf_decision_fastText = svm.SVC()
clf_decision_fastText.fit(X_train_ft, Y_train_ft)
test_predictions_fastText = clf_decision_fastText.predict(X_test_ft)
print("Time taken to fit the model with fastText vectors: " + str(time.time() - start_time))
```

Gambar 5. Implementasi klasifikasi SVM di python

Beberapa contoh output Hasil klasifikasi SVM ditunjukkan pada **Tabel 9.**

Tabel 9. Hasil Klasifikasi SVM

Cerita	Kategori_Prediksi	Kategori_Sebenarnya
Seekor Kelinci sombong selalu menyombongkan kecepatannya kepada hewan-hewan lain. Kura-Kura, yang lambat tapi cerdas, menantang Kelinci untuk balapan. Kelinci setuju dengan yakin akan menang dengan mudah. Namun, dia terlalu percaya diri sehingga dia memutuskan untuk tidur sejenak selama balapan. Kura-Kura berlari perlahan tapi stabil hingga akhirnya mencapai garis finish pertama kali. Kelinci terbangun dan menyadari kesalahannya. Dari situlah dia belajar untuk tidak sombong dan menghargai setiap tantangan.	0	0
Asal usul Gunung Papandayan, gunung berapi di Jawa Barat, berkaitan dengan legenda yang menceritakan tentang Ki Santang, seorang pangeran yang kuat dan bijaksana. Ia jatuh cinta pada Dewi Rengganis, putri raja dari Kerajaan Salakanagara. Namun, cinta mereka	1	1

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI:10.31515/telematika.v21i2. 10867

ditentang oleh Raja Salakanagara, dan perang pun tak terhindarkan. Ki Santang tewas dalam pertempuran, tetapi sebelum meninggal, ia mengutuk gunung yang kini dikenal sebagai Gunung Papandayan. Gunung ini meletus, menghancurkan sekitarnya, dan mengubur wilayah tersebut di bawah abu vulkanik. Legenda ini melambangkan pengorbanan cinta dan hubungan manusia dengan alam yang kuasa.		
Dewi Sri mengisahkan tentang Dewi yang melambangkan dewi hasil bumi dan pertanian dalam mitologi Jawa. Dewi Sri dikenal sebagai penjaga panen dan keberlimpahan. Dikatakan bahwa setiap kali panen sukses, Dewi Sri menghuni lumbung-lumbung padi dan memberikan keberuntungan kepada petani. Cerita ini mengajarkan nilai-nilai ketergantungan manusia pada alam, keberlimpahan, dan pentingnya menghormati alam serta mengelola sumber daya dengan bijaksana.	2	2

3.3. Evaluasi

Pada proses ini akan dilakukan menggunakan data pengujian yang tidak pernah dilihat model selama pelatihan, mengukur kinerja model dengan metrik seperti akurasi, *presisi*, *recall*, dan F1-score. Implemetasi *classification report* di python dapat dilihat pada **Gambar 7**.

```
print("CLASSIFICATION REPORT TESTING")
print("============================")
test_predictions_fastText = clf_decision_fastText.predict(X_test_ft)
print(classification_report(Y_test_ft,test_predictions_fastText))
```

Gambar 6. Implementasi *Classification Report* di python

Matriks evaluasi yang didapatkan ditunjukkan pada **Tabel 10.**

Tabel 10. Confussion Matrix

		Aktual		
		Fabel	**Legenda**	**Mitos**
Prediksi	Fabel	43	0	5
	Legenda	0	41	7
	Mitos	2	7	39

Berdasarkan *confussion matrix* **Tabel 10** diperoleh nilai evaluasi akurasi, precision, recall, F1 score yang digambarkan pada **Tabel 11**.

Tabel 11. *Classification Report*

Metrics	**Fabel**	**Legenda**	**Mitos**	**Rata-rata**
Akurasi	95%	85%	76%	85%
Presisi	96%	85%	76%	86%
Recall	90%	85%	81%	85%
F1-Score	92%	85%	79%	86%

Berdasarkan **Tabel 11** dapat disimpulkan bawa nilai akurasi rata-rata keseluruhan adalah 85%. Nilai akurasi tertinggi diperoleh dari kategori fabel, kemudian nilai presisi rata-rata sebesar 86%, recall 85% dan F1-score 86%.

4. Kesimpulan dan Saran

Berdasarkan hasil klasifikasi dongeng Indonesia menggunakan metode SVM dan ekstraksi fitur FastText, model mencapai akurasi 85%. Hasil ini didukung oleh nilai presisi 85%, recall 85%, dan F1-score 86%, yang menunjukkan kinerja model yang konsisten. Namun, akurasi pada kategori mitos tercatat lebih rendah, yaitu sebesar 76%, yang menunjukkan perlunya peningkatan lebih lanjut, terutama dalam proses labeling untuk memisahkan antara legenda dan mitos dengan lebih baik.

Daftar Pustaka

[1] P. P. Ardini, "Pengaruh Dongeng dan Komunikasi Terhadap Perkembangan Moral Anak Usia 7-8 Tahun," *Jurnal Pendidikan Anak*, vol. 1, no. 1, 2015, doi: 10.21831/jpa.v1i1.2905.

[2] R. Rukiyah, "Dongeng, Mendongeng, dan Manfaatnya," *Anuva*, vol. 2, no. 1, p. 99, 2018, doi: 10.14710/anuva.2.1.99-106.

[3] U. D. Rosada, "Memperkuat Karakter Anak melalui Dongeng berbasis Media Visual," *Children Advisory Research and Education)*, vol. 04, no. 1, pp. 42–49, 2016.

[4] R. T. Rakhman, Y. A. Piliang, H. A. Ahmad, and I. Gunawan, "Pemetaan Jenis Dongeng Nusantara Dalam Infografis," *ANDHARUPA: Jurnal Desain Komunikasi Visual & Multimedia*, vol. 7, no. 01, pp. 59–78, 2021.

[5] O. Somantri and M. Khambali, "Feature Selection Klasifikasi Kategori Cerita Pendek Menggunakan Naïve Bayes dan Algoritme Genetika," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*, vol. 6, no. 3, pp. 301–306, 2017, doi: 10.22146/jnteti.v6i3.332.

[6] A. Deolika, K. Kusrini, and E. T. Luthfi, "Analisis Pembobotan Kata Pada Klasifikasi Text Mining," *Jurnal Teknologi Informasi*, vol. 3, no. 2, p. 179, 2019, doi: 10.36294/jurti.v3i2.1077.

[7] C. Darujati and A. B. Gumelar, "Pemanfaatan teknik supervised untuk klasifikasi teks bahasa indonesia," *Jurnal Bandung Text Mining*, vol. 16, no. 1, pp. 1–5, 2012.

[8] A. Tabassum and R. R. Patil, "A survey on text pre-processing & feature extraction techniques in natural language processing," *International Research Journal of Engineering and Technology (IRJET)*, vol. 7, no. 06, pp. 4864–4867, 2020.

[9] T. Yao, Z. Zhai, and B. Gao, "Text classification model based on fasttext," in *2020 IEEE International Conference on Artificial Intelligence and Information Systems (ICAIIS)*, 2020, pp. 154–157.

[10] A. Nurdin, B. A. S. Aji, A. Bustamin, and Z. Abidin, "Perbandingan Kinerja Word Embedding Word2Vec, Glove, Dan Fasttext Pada Klasifikasi Teks," *Jurnal Tekno Kompak*, vol. 14, no. 2, pp. 74–79, 2020.

[11] M. M. Kusairi *et al.*, "SVM Method with FastText Representation Feature for Classification of Twitter Sentiments Regarding the Covid-19 Vaccination Program 1,2," vol. x, no. 02, pp. 140–150, 2022.

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI:10.31515/telematika.v21i2. 10867

[12] L. Hickman, S. Thapa, L. Tay, M. Cao, and P. Srinivasan, "Text preprocessing for text mining in organizational research: Review and recommendations," *Organ Res Methods*, vol. 25, no. 1, pp. 114–146, 2022.

[13] A. W. Pradana and M. Hayaty, "The effect of stemming and removal of stopwords on the accuracy of sentiment analysis on indonesian-language texts," *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, pp. 375–380, 2019.

[14] T. Siddiqui and others, "Sarcasm detection from twitter database using text mining algorithms," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 11, pp. 1916–1924, 2021.

[15] Y. Permana, A. Emarilis, and others, "Stemming Analysis Indonesian Language News Text with Porter Algorithm," in *Journal of Physics: Conference Series*, 2021, p. 12019.

[16] Q. H. Nguyen *et al.*, "Influence of data splitting on performance of machine learning models in prediction of shear strength of soil," *Math Probl Eng*, vol. 2021, pp. 1–15, 2021.

[17] S. Thavareesan and S. Mahesan, "Sentiment lexicon expansion using Word2vec and fastText for sentiment prediction in Tamil texts," in *2020 Moratuwa engineering research conference (MERCon)*, 2020, pp. 272–276.

[18] N. Kalcheva, M. Karova, and I. Penev, "Comparison of the accuracy of SVM kemel functions in text classification," in *2020 International Conference on Biomedical Innovations and Applications (BIA)*, 2020, pp. 141–145.

[19] S. Adinugroho and Y. A. Sari, *Implementasi Data Mining Menggunakan Weka*. Universitas Brawijaya Press, 2018.