# Klasifikasi Nuansa Emosi Film Berdasarkan Sinopsis Menggunakan Logistic Regression

Skye Kanahaya Endrawan[a1], Cokorda Pramartha[a2]

[a]Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana
Jalan Raya Kampus Udayana, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia
[1]skyeendrawan@email.com
[2]cokorda@unud.ac.id

***Abstract***

*With the rapid development of digital era, movie industry has consider to be a dominant form of entertainment. However, with thousands of movies released every year, audiences often face difficulties selecting movies based on their emotional preferences. This study proposes a classification approach to determine the emotional nuance of movies (happy, sad, tense) based on their synopsis. The dataset used in this study was sourced from Kaggle with 676.491 entries and labeled using a DistilBERT pre-trained emotion detection model from Hugging Face. After mapping the labeled entries and undersampling, 6.600 balanced samples were used. Following preprocessing and data splitting, TF-IDF text representation and Logistic Regression model were applied. The model achieved 76% accuracy on validation data and improved to 83% on test data, with macro F1-scores reflecting consistent performance across all classes. These results suggest that movie synopses contain sufficient emotional signals to be automatically classified using a lightweight and effective machine learning approach.*

***Keywords:*** *Text Processing, Logistic Regression, Classification, Movie Synopsos*

## 1. Pendahuluan

Dalam pesatnya perkembangan informasi digital saat ini, masyarakat memiliki akses yang sangat luas terhadap media hiburan. Film dianggap sebagai salah satu bentuk hiburan yang paling dikenal luas, serta telah mengalami pertumbuhan pesat dan memiliki pengaruh besar dalam kehidupan masyarakat [1]. Industri perfilman global berkembang sangat pesat dalam dua dekade terakhir.Setiap tahunnya, ribuan judul film dari berbagai genre dan negara dirilis ke pasar [2]. Namun, dengan banyaknya pilihan tersebut sering kali justru menimbulkan kebingungan di kalangan penonton dalam menentukan film yang ingin dilihat [3]

Sebuah film umumnya memiliki satu atau lebih genre untuk menggambarkan berbagai elemen dan karakteristik yang terkandung dalam film tersebut. Hal ini membuat klasifikasi genre menjadi tidak selalu jelas, karena batas antar genre sering kali tumpang tindih karena memiliki ciri-ciri yang mirip [4]. Untuk menjawab keterbatasan tersebut, ditawarkan pendekatan berbasis klasifikasi emosi film sebagai solusi. Tidak seperti genre yang bersifat kategoris dan lebih umum, emosi seperti *happy, sad,* dan *tense* memberikan deskripsi yang lebih personal dan kontekstual, serta dapat mencerminkan nuansa afektif yang lebih relevan dengan preferensi emosional yang ingin dirasakan oleh penonton [5].

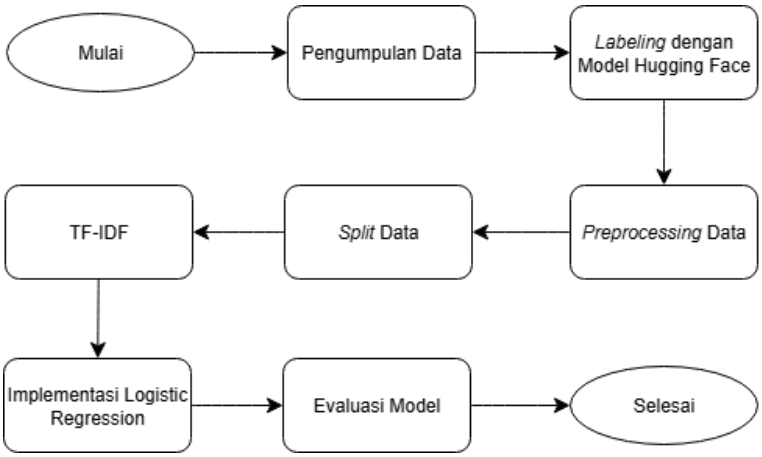
Studi terdahulu yang sudah dilakukan oleh [1] berfokus pada pengenalan emosi berdasarkan ulasan penonton, sehingga menggambarkan emosi yang dirasakan penonton (*induced emotion*), bukan emosi yang ditampilkan oleh film itu sendiri. Secara fundamental, hal tersebut berbeda dengan penelitian ini yang berfokus pada *perceived emotion* berdasarkan informasi dalam film itu sendiri, yakni melalui sinopsis. Sementara itu, penelitian lain seperti [6], [7]berfokus pada klasifikasi genre, tanpa menangkap dimensi emosional yang lebih subjektif dan halus dari narasi.

p-ISSN: 2986-3929
e-ISSN: 3032-1948

Penelitian ini menawarkan pendekatan klasifikasi nuansa emosi film berdasarkan data sinopsis, yang merupakan sumber naratif padat dan tersedia luas. Dibanding elemen lain dari film seperti judul, sinopsis mengandung informasi yang lebih besar mengenai alur dari sebuah film [8]. Klasifikasi nuansa emosi pada penelitian berikut memanfaatkan metode representasi teks TF-IDF serta model Logistic Regression. Penelitian [9] menunjukkan bahwa model Logistic Regression memiliki performa yang lebih baik dalam mengklasifikasi text dalam berbagai kelas emosi daripada model serupa

## 2. Metode Penelitian

Penelitian ini dirancang melalui tahapan sistematis mulai dari pengumpulan data hingga proses evaluasi model. Setiap tahap dirancang untuk memastikan bahwa proses berjalan efektif dan mendapatkan performa yang baik. Beberapa tahapan divisualisasikan pada diagram berikut.




**Gambar 1.** Alur Penelitian

### 2.1. Pengumpulan Data

*Dataset* pada penelitian ini merupakan data sekunder yang diperoleh dari *Kaggle* yaitu '*Letterboxd Movie Datasets*'. *Letterboxd* merupakan *social platform* untuk penyuka film di mana pengguna dapat berbagi mengenai pengalaman menonton mereka. *Dataset* ini terdiri dari id, judul, tahun rilis, *tagline*, sinopsis, durasi, *genre* serta *rating* dari film. Jumlah data yang diperoleh adalah sebanyak 676.491 data, namun dilakukan *random sampling* sehingga yang digunakan hanya 20.000 data untuk proses klasifikasi.

### 2.2. *Labeling* dengan Model *Hugging Face*

Pada *dataset*, setiap film belum memiliki label emosi. Oleh karena itu dilakukan pendekatan *pre-trained labeling* menggunakan model yang sudah dilatih sebelumnya untuk mengklasifikasikan emosi dari teks. Model yang digunakan adalah *distilbert-base-uncased-finetuned-emotion* yang berasal dari *Hugging Face*. Model ini merupakan versi ringan dari BERT yang telah dilatih secara khusus untuk mendeteksi emosi dari teks.

Secara teknis, model ini bekerja dengan mengkonversi teks *input* ke representasi numerik melalui proses tokenisasi. Kemudian token-token ini diproses oleh jaringan *transformer DistilBERT* untuk menghasilkan *embedding contextualized* dari teks. *Output* dari layer terakhir berupa vektor yang merepresentasikan kekuatan masing-masing kelas emosi yaitu *anger, fear, joy, love, sadness,* dan *surprise*. Vektor ini kemudian diproses oleh fungsi *softmax* untuk menghasilkan probabilitas dari setiap label emosi. Label dengan probabilitas tertinggi dianggap sebagai dominan dalam teks tersebut sehingga *output* label merupakan emosi dominan yang ditangkap dari *input* teks.

### 2.3. *Preprocessing* Data

*Preprocessing* data dilakukan untuk menyiapkan *dataset* sehingga dapat diproses pada tahap implementasi model. *Preprocessing* yang dilakukan dalam penelitian ini mencakup beberapa tahapan mulai dari penggabungan *csv dataset* hingga tahap *stemming.*

- **Persiapan *Dataset***
  Persiapan dataset yang dilakukan disini mencakup proses penggabungan *csv movies* dengan *csv genres,* menghapus data yang duplikat dan *null,* kemudian membuang kolom yang tidak diperlukan.

- ***Resampling***
  Tahap kedua dari *preprocessing* yaitu melakukan *resampling* terhadap 676.491 data menjadi 20.000 data yang siap digunakan dalam proses *labeling* dengan model *Hugging Face*. Proses *resampling* dilakukan random dengan parameter *random_state* yang digunakan adalah 42.

- ***Case Folding***
  *Case Folding* terdiri dari dua proses yaitu, proses mengubah teks sinopsis menjadi *lowercase* dan proses menghilangkan karakter selain huruf pada teks sinopsis.

- ***Tokenizing***
  *Tokenizing* dilakukan pada data yang sudah melalui *case folding*. Data hasil *case folding* diuraikan menjadi bagian kecil yang berupa kata atau karakter.

- ***Stopwords Removal***
  Pada proses ini dilakukan menghapusan kepada kata-kata umum yang sering muncul namun tidak memberikan makna penting. *Library stopwords* yang digunakan pada proses ini dalam bahasa Inggris, menyesuaikan bahasa yang digunakan pada *dataset.*

- ***Stemming***
  Tahap terakhir dari *preprocessing* data ini mengubah kata ke bentuk dasarnya. Proses ini dilakukan untuk mengurangi variasi bentuk kata sehingga dapat dibaca oleh model.

### 2.4. *Split* Data

Pada implementasi ini data dibagi menjadi 3 bagian yaitu *training*, *validation*, dan *test* dengan rasio 70:15:15.

### 2.5. *Label Encoding*

Model seperti Logistic Regression membutuhkan target dalam bentuk angka tunggal, bukan dalam format vektor. Sehingga dengan *Label Encoding*, setiap genre dikonversi menjadi nilai numerik yang sesuai untuk proses klasifikasi.

### 2.6. TF-IDF

TF-IDF *Vectorizer* mentransformasi teks ke fitur numerik dengan struktur data berbentuk matriks *sparse* (jumlah sampel, jumlah fitur) untuk menilai bobot setiap kata dalam sinopsis. TF-IDF digunakan untuk mereduksi *noise*, dan mengubah teks menjadi vektor numerik untuk diproses model *machine learning*.

### 2.7. Implementasi Logistic Regression

Logistic Regression merupakan algoritma klasifikasi yang memodelkan hubungan antara fitur masukan dan probabilitas dari kelas target dengan menggunakan fungsi sigmoid. Dalam konteks penelitian ini, setiap sinopsis film yang telah direpresentasikan dalam bentuk vektor fitur akan dipetakan ke dalam tiga kelas target emosi yaitu *happy*, *sad*, dan *tense*.

p-ISSN: 2986-3929
e-ISSN: 3032-1948

### 2.8. Evaluasi

Setelah proses pelatihan selesai, evaluasi performa model dilakukan pada data validasi dan data uji. Evaluasi mencakup metrik-metrik klasifikasi standar, yaitu akurasi (*accuracy*), presisi (*precision*), *recall*, dan F1-*score*, yang dihitung untuk masing-masing kelas. Selain itu, digunakan juga *confusion matrix* untuk memvisualisasikan distribusi prediksi yang benar dan salah antar kelas, sehingga dapat memberikan gambaran lebih rinci mengenai performa model pada tiap kategori emosi.

## 3. Hasil dan Diskusi

### 3.1. Pengumpulan Data

Seperti yang sudah disinggung sebelumnya, data yang digunakan pada penelitian ini diperoleh dari *Kaggle.* Data yang diperoleh dari *Kaggle* masih terpisah pada dua csv yang berbeda yaitu antara data film yang memuat kolom *id, name, date, tagline, description, minute, dan rating* dengan data genre dari film. Sehingga untuk proses awal dilakukan *merging* kepada dua csv berdasarkan *id.* Dari data gabungan dilakukan pengecekan data duplikat dan data *null.*

```
Jumlah Data: 676491
Jumlah Missing Pada Kolom
id                  0
name                4
date            29094
tagline        553762
description    102418
minute          95143
rating         587012
genre               0
dtype: int64
Data Duplikat:
0
```

**Gambar 2.** Hasil Pengecekan Data

Pada gambar 2 ditunjukkan hasil pengecekkan dengan total data sebanyak 676.491, kemudian beberapa kolom masih memiliki nilai *missing*, dan terdapat 0 duplikat. Kemudian data yang masih memiliki nilai *null* dihapus dan beberapa kolom yang tidak terpakai juga dihapus. Kolom yang tersisa ada kolom *id, name, description,* dan *genre*. Tahap terakhir pada persiapan data yaitu proses *resample* 20.000 data yang nantinya akan digunakan pada *pre-trained labeling.*

### *3.2. Labeling* dengan Model *Hugging Face*

Proses *pre-trained labeling* dilakukan dengan model DistilBERT dari *Hugging Face***.** Model memproses setiap teks sinopsis kemudian mengambil hasil prediksi dari semua label dengan skor probabilitas. Setelah skor diurutkan, model mengembalikan label dengan skor tertinggi sebagai hasil karena merupakan emosi dominan. Label hasil prediksi disimpan.

**Tabel 1.** Distribusi Emosi

| Emosi | Jumlah |
|---|---|
| *Joy* | 6842 |
| *Anger* | 5596 |
| *Fear* | 4585 |
| *Sadness* | 2232 |
| *Love* | 479 |

p-ISSN: 2986-3929
e-ISSN: 3032-1948

| Emosi | Jumlah |
|---|---|
| *Surprise* | 264 |
| *Unknown* | 2 |

Tabel 1 menunjukkan jumlah persebaran data setelah dilakukan *pre-trained labeling.* Dari persebaran tersebut kemudian dilakukan *mapping* label mosi menjadi hanya 3 kelas, yaitu *happy, tense,* dan *sad.* Komposisi *mapping* yaitu untuk label *joy* dan *love* ke kelas *happy,* selanjutnya label *anger* dan *fear* ke kelas *tense*, sementara label *sadness* ke kelas *sad.* Untuk label *surprise* tidak digunakan karena jumlah yang sedikit membuat distribusi tidak seimbang. Selain itu label *surprise* tidak cukup kuat untuk menggambarkan nuansa emosional dominan dari sinopsis.

**Tabel 2.** Distribusi *Mapping*

| ***Mapped*** | **Jumlah** |
|---|---|
| *Tense* | 10181 |
| *Happy* | 7321 |
| *Sad* | 2232 |

Dari hasil distribusi *mapping* seperti yang ditunjukkan pada Tabel 2, dilakukan *undersampling* agar data seimbang. pada setiap kelas *'mapped'* diambil 2.200 data, sehingga total data yang akan digunakan pada proses-proses selanjutnya yaitu sebanyak 6.600 data.

### 3.3. *Preprocessing* Data

Tahap selanjutnya yaitu *preprocessing* data, di mana seluruh data yang sudah diberi label melewati proses *case folding, tokenization, stopwords removal,* dan *stemming.* Hasil dari setiap proses *preprocessing* akan disimpan pada kolom baru yang diberi nama sesuai prosesnya, kemudian digunakan pada proses selanjutnya. Setelah semua tahapan *preprocessing* selesai, hasilnya disimpan pada *dataframe* baru seperti pada Gambar 2.

| lowercase | cleanpunct | tokens | stopword | stemming |
|---|---|---|---|---|
| a mother and a daughter in a bond beyond time,... | a mother and a daughter in a bond beyond time ... | [a, mother, and, a, daughter, in, a, bond, bey... | [mother, daughter, bond, beyond, time, waiting... | [mother, daughter, bond, beyond, time, wait, m... |
| "lovely morning but cold and frosty. one would... | lovely morning but cold and frosty one would... | [lovely, morning, but, cold, and, frosty, one,... | [lovely, morning, cold, frosty, one, would, th... | [love, morn, cold, frosti, one, would, think, ... |
| a documentary highlighting the 20 year history... | a documentary highlighting the 20 year history... | [a, documentary, highlighting, the, 20, year, ... | [documentary, highlighting, 20, year, history,... | [documentari, highlight, 20, year, histori, in... |
| the story of a young couple who are expecting ... | the story of a young couple who are expecting ... | [the, story, of, a, young, couple, who, are, e... | [story, young, couple, expecting, first, baby,... | [stori, young, coupl, expect, first, babi, see... |
| sophie (ambika mod) has spent weeks planning a... | sophie ambika mod has spent weeks planning a s... | [sophie, ambika, mod, has, spent, weeks, plann... | [sophie, ambika, mod, spent, weeks, planning, ... | [sophi, ambika, mod, spent, week, plan, specia... |

**Gambar 2.** *Preprocessing* Data

### 3.4. *Split* Data

Setelah melewati tahap *preprocessing*, dilakukan pembagian kepada seluruh data dengan rasio 70:15:15. Data dibagi menjadi tiga subset yaitu, data *training, validation,* dan *test*. Parameter *stratify* pada label (*y*) digunakan agar distribusi kelas emosi tetap seimbang di setiap subset. Hasil *split* data dapat dilihat pada Tabel 3.

**Tabel 3.** *Split* Data

| ***Mapped*** | ***Train Set*** | ***Validation Set*** | ***Test Set*** |
|---|---|---|---|
| ***Tense*** | 1540 | 330 | 330 |

p-ISSN: 2986-3929
e-ISSN: 3032-1948

| *Mapped* | *Train Set* | *Validation Set* | *Test Set* |
|---|---|---|---|
| ***Happy*** | 1540 | 330 | 330 |
| ***Sad*** | 1540 | 330 | 330 |
| **Total** | 4620 | 990 | 990 |

### *3.5. Label Encoding*

Pada data *train, validation,* dan *test* dilakukan proses *label encoding* untuk merubah label emosi dalam nilai numerik. Penelitian ini menggunakan 3 label di mana ada 3 label numerik yaitu 0 (*happy*), 1 (*sad*), dan 2 (*tense*). Potongan program pada proses *label encoding* dapat dilihat pada Gambar 3.

```
# Label Encoding
label_encoder = LabelEncoder()
train['label'] = label_encoder.fit_transform(train['mapped'])
validation['label'] = label_encoder.fit_transform(validation['mapped'])
test['label'] = label_encoder.fit_transform(test['mapped'])

# Menampilkan label dan nilai encodenya
for i, label in enumerate(label_encoder.classes_):
    print(f"{i}: {label}")
```

**Gambar 3.** *Label Encoding*

### 3.6. TF-IDF

Proses TF-IDF dilakukan ke seluruh subset data yang sudah dilakukan pembagian sebelumnya. Metode ini dilakukan untuk mengubah teks sinopsis menjadi vektor numerik yang dapat diproses oleh model. TF-IDF diimplementasikan dengan parameter *min_df=2* untuk menghapus kata yang terlalu jarang muncul, *max_df=0.8* untuk menghindari kata yang terlalu umum, serta *ngram_range=(1,2)* agar mempertimbangkan unigram dan bigram sehingga dapat menangkap konteks lokal dari pasangan kata. Potongan program pada pembobotan divisualisasikan pada Gambar 4.

```
# TF-IDF Vectorizer
tfidf = TfidfVectorizer(
    min_df=2,
    max_df=0.8,
    ngram_range = (1,2)
)

X_train = tfidf.fit_transform(X_train_text )
X_val = tfidf.transform(X_val_text)
X_test = tfidf.transform(X_test_text)
```

**Gambar 4.** TF-IDF

### 3.7. Implementasi Logistic Regression

Model klasifikasi pada penelitian ini dibangun menggunakan algoritma Logistic Regression yang diinisialisasi dengan beberapa parameter untuk mengoptimalkan performa. Parameter *max_iter=1000* digunakan untuk memastikan proses iterasi saat jumlah fitur tinggi. Kemudian untuk menghindari *overfitting* digunakan regularisasi L2 dengan parameter *penalty='12'* dan *C=1*. Potongan program dari implementasi Logistic Regression dapat dilihat pada Gambar 5.

p-ISSN: 2986-3929
e-ISSN: 3032-1948

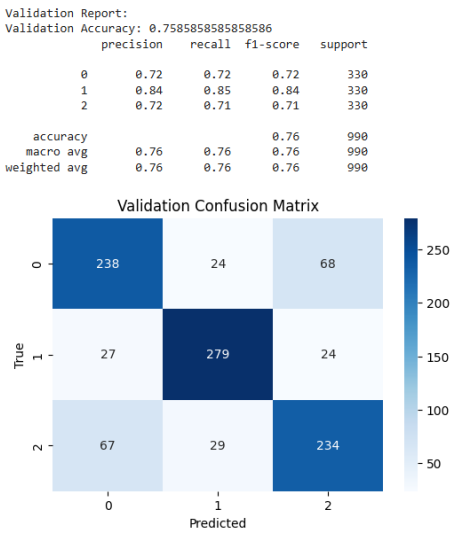
```
# Inisialisasi dan training model
model = LogisticRegression(
    max_iter=1000,
    penalty='l2',
    C=1,
    solver='lbfgs',
    random_state=42
)
model.fit(X_train, y_train)
```

**Gambar 5.** Implementasi Model

Optimasi bobot dilakukan menggunakan algoritma *lbfgs*, yaitu *solver* yang cocok untuk dataset berukuran sedang hingga besar. Proses pelatihan dilakukan pada data hasil representasi TF-IDF, yaitu *x_train* dan label target *y_train.*

### 3.8. Evaluasi

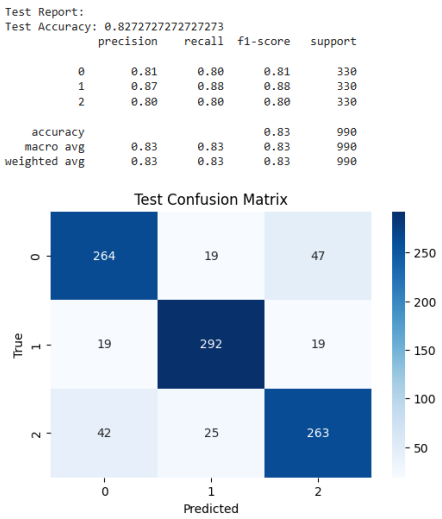
Setelah proses pelatihan selesai, model dievaluasi pada dua tahap, yaitu pada data validasi dan dta uji. Evaluasi dilakukan menggunakan metrik klasifikasi akurasi, *precision, recall,* dan F1-*score* untuk masing-masing kelas, serta *confusion matrix* untuk memvisualisasikan pola kesalahan prediksi antar kelas. Ketiga kelas target yang digunakan dalam klasifikasi adalah *happy, sad,* dan *tense* yang masing-masing diwakili oleh label numerik 0, 1, dan 2.




**Gambar 6.** Evaluasi Validasi

Pada data validasi yang divisualisasikan Gambar 6, model memperoleh akurasi sebesar 76%. Berdasarkan *classification report*, model menunjukkan performa terbaik pada kelas *tense* dengan F1-*score* 84%, sedangkan kelas *happy* dan *sad* memperoleh F1-*score* masing-masing 72% dan 71%. *Confusion matrix* menunjukkan bahwa sebagian besar klasifikasi terjadi antara kelas *happy* dan *sad*, yang mengindikasikan adanya tumpang tindih makna emosional dalam sinopsis kedua kelas tersebut.

p-ISSN: 2986-3929
e-ISSN: 3032-1948




**Gambar 7.** Evaluasi Uji

Pada data uji yang divisualisasikan Gambar 7, model menunjukkan peningkatan performa dengan akurasi mencapai 83% dan *macro* F1-*score* sebesar 83%. Kelas *tense* tetap menjadi yang paling akurat dengan F1-*score* 88%, sementara *happy* dan *sad* masing-masing memperoleh 81% dan 80%. *Confusion matrix* menunjukkan kesalahan klasifikasi yang lebih sedikit, menandakan bahwa model mampu melakukan generalisasi yang baik terhadap data baru.

## 4. Kesimpulan

Penelitian ini berhasil menunjukkan bahwa pendekatan klasifikasi nuansa emosi film berbasis sinopsis menggunakan TF-IDF dan Logistic Regression mampu mencapai akurasi dan generalisasi yang baik. Hasil akurasi sebesar 83% pada data uji serta nilai F1-*score* makro yang merata menunjukkan bahwa sinopsis film dapat memberikan informasi emosional yang cukup kuat untuk proses klasifikasi. Temuan ini mendukung potensi pemanfaatan pendekatan ini dalam sistem rekomendasi film berbasis preferensi emosional. Untuk penelitian selanjutnya, disarankan penggunaan metode *deep learning* serta perluasan pada label emosi agar mampu menangkap nuansa yang lebih kompleks dan kaya dari teks naratif film.

## Daftar Pustaka

[1] R. Na dan N. Sun, “Emotion Recognition and Classification of Film Reviews Based on Deep Learning and Multimodal Fusion,” *Wirel Commun Mob Comput*, vol. 2022, no. 1, hlm. 2024352, Jan 2022, doi: 10.1155/2022/2024352.

[2] A. R. Fadillah, K. K. Noviyanti, I. P. A. A. Wiguna, dan C. Pramartha, “Perancangan Ontologi Semantik: Representasi Digital Film Bioskop Indonesia,” *JELIKU (Jurnal Elektronik Ilmu Komputer Udayana)*, vol. 12, no. 4, hlm. 955, Mei 2024, doi: 10.24843/JLK.2024.v12.i04.p20.

[3] R. P. Kurnia dan Y. A. Atma, “Analisis Rekomendasi Film Dari Data Imdb Menggunakan Python,” *Device : Journal of Information System, Computer Science And Information Technology*, vol. 3, no. 2, hlm. 23–28, Des 2022, doi: 10.46576/DEVICE.V3I2.2698.

[4] P. Khant dan B. Tidke, “Multimodal Approach to Recommend Movie Genres Based on Multi Datasets,” *Indian J Sci Technol*, vol. 16, no. 30, hlm. 2304–2310, Agu 2023, doi: 10.17485/IJST/v16i30.1238.

[5] M. Muszynski *dkk.*, “Recognizing Induced Emotions of Movie Audiences from Multimodal Information,” *IEEE Trans Affect Comput*, vol. 12, no. 1, hlm. 36–52, Jan 2021, doi: 10.1109/TAFFC.2019.2902091.

[6] N. K. Rajput dan B. A. Grover, “A multi-label movie genre classification scheme based on the movie’s subtitles,” *Multimed Tools Appl*, vol. 81, no. 22, hlm. 32469–32490, Sep 2022, doi: 10.1007/S11042-022-12961-6/TABLES/5.

[7] N. Muslimah, I. Indriati, dan R. C. Wihandika, “Klasifikasi Film Berdasarkan Sinopsis dengan Menggunakan Improved K-Nearest Neighbor (K-NN),” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 3, no. 1, hlm. 196–204, 2019, Diakses: 4 Juli 2025. [Daring]. Tersedia pada: https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/4066

[8] Q. Hoang, “Predicting Movie Genres Based on Plot Summaries,” Jan 2018, doi: https://doi.org/10.48550/arXiv.1801.04813.

[9] F. M. Alotaibi, “Classifying Text-Based Emotions Using Logistic Regression,” *VAWKUM Transactions on Computer Sciences*, vol. 7, no. 1, hlm. 31–37, Apr 2019, doi: 10.21015/VTCS.V16I2.551.

p-ISSN: 2986-3929
e-ISSN: 3032-1948

Halaman ini sengaja dibiarkan kosong