BAB II
KAJIAN LITERATUR

2.1 Tinjauan Pustaka

Pada bagian ini berisikan landasan teori dan akan dijelaskan tinjauan pustaka yang berkaitan dengan penelitian yang dilakukan.

2.1.1 Prediksi Kesuksesan Film

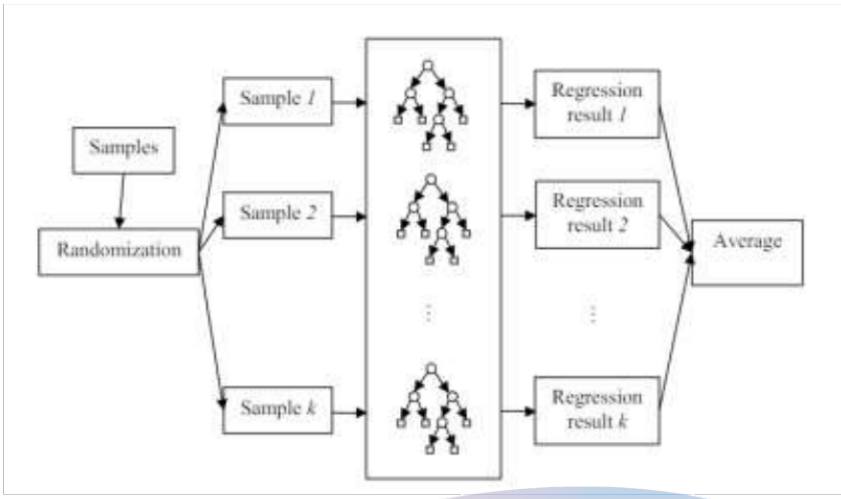
Memprediksi kesuksesan film telah menjadi semakin canggih melalui penerapan *machine learning* dan teknik analisis data. Berbagai model telah dikembangkan untuk menilai faktor-faktor yang mempengaruhi kinerja film, termasuk analisis sentimen dan atribut film [1], [10]. Model hibrida yang menggabungkan fitur film dan analisis sentimen juga mencapai akurasi yang mengesankan menunjukkan efektivitas mengintegrasikan data media sosial dengan metrik tradisional [4].

Sementara kemajuan dalam pemodelan prediktif ini menawarkan wawasan yang berharga, ketidakpastian yang melekat pada preferensi audiens dan dinamika pasar tetap menjadi tantangan, menunjukkan bahwa tidak ada model yang dapat menjamin kesuksesan secara 100%. Namun, menggunakan model dapat memberikan pengetahuan yang lebih mendalam tentang kemungkinan kesuksesan sebuah film. Ini dapat membantu pembuat film mengetahui potensi kesuksesan sebuah film, daripada hanya menebak tanpa informasi apapun.

2.1.2 Random Forest

Algoritma *Random Forest* digunakan untuk memprediksi kesuksesan film dengan menganalisis berbagai faktor sebelum film dirilis [11]. Dengan menerapkan algoritma ini, model dapat memprediksi *rating* film berdasarkan karakteristik film dan preferensi masyarakat dari data sebelumnya [12]. Algoritma ini efektif dalam mengatasi keterbatasan fitur selama tahap produksi, sehingga membantu produser membuat keputusan yang lebih baik sebelum film dirilis [13].

Random Forest terdiri dari beberapa *decision tree* yang digabungkan. Preferensi dan pengaruh antar pengguna dipelajari lalu model keseluruhan digunakan untuk prediksi. Sampel untuk setiap *decision tree* dipilih secara acak, begitu pula variabel untuk setiap subset fitur. Hasil regresi dari setiap pohon berbeda dan dirata-ratakan untuk mendapatkan hasil akhir dari *Random Forest* [14]. Alur kerja dari *Random Forest* dapat dilihat pada Gambar 2.1.



Gambar 2.1 Alur Kerja *Random Forest* [14]

2.1.3 Gradient Boosting

Algoritma *Gradient Boosting* bekerja dengan membangun model prediksi secara berurutan melalui serangkaian model prediksi lemah, seperti *decision tree*, untuk mengurangi kesalahan dan meningkatkan akurasi [12], [15]. Dengan menggunakan *Gradient Boosting*, industri film dapat memprediksi kesuksesan film sebelum dirilis, membantu pengambilan keputusan strategis terkait pemasaran dan waktu perilisan [12]. Algoritma ini memberikan bobot pada faktor-faktor seperti anggaran, aktor, *rating* dan data media sosial untuk memprediksi kesuksesan film berdasarkan informasi historis, sehingga meningkatkan akurasi prediksi [16]. *Pseudocode* dari *Gradient Boosting* dapat dilihat pada Gambar 2.2.

Algorithm 1 Friedman's Gradient Boost algorithm

Inputs:

- input data $(x, y)_{i=1}^{N}$
- number of iterations M
- choice of the loss-function $\Psi(y, f)$
- choice of the base-learner model $h(x, \theta)$

Algorithm:

1: initialize $\widehat{f}_0$ with a constant
2: **for** $t = 1$ to M **do**
3: compute the negative gradient $g_t(x)$
4: fit a new base-learner function $h(x, \theta_t)$
5: find the best gradient descent step-size ρ_t:

$$\rho_t = \arg\min_{\rho} \sum_{i=1}^{N} \Psi\left[y_i, \widehat{f}_{t-1}(x_i) + \rho h(x_i, \theta_t)\right]$$

6: update the function estimate:

$$\widehat{f}_t \leftarrow \widehat{f}_{t-1} + \rho_t h(x, \theta_t)$$

7: **end for**

Gambar 2.2 *Pseudocode Gradient Boosting* [17]

Pada Gambar 2.2. tertera algoritma *Gradient Boosting* karya Friedman. Algoritma ini digunakan untuk membangun model prediksi yang kuat dengan menggabungkan beberapa model prediksi lemah. Inputnya terdiri dari pasangan data (x, y), dimana x adalah vektor fitur dan y adalah variabel target, serta jumlah iterasi yang diinginkan, M. Algoritma ini juga memerlukan pemilihan fungsi kerugian $\psi(y, f)$ dan model pembelajar dasar $h(x, \theta)$.

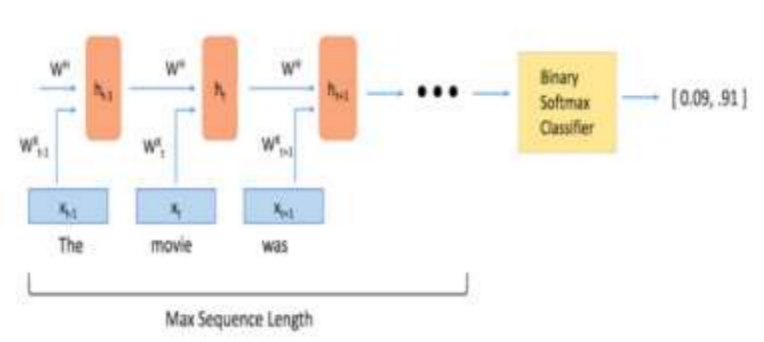
Prosesnya dimulai dengan menginisialisasi estimasi fungsi $\widehat{f_0}$ dengan sebuah konstanta. Untuk setiap iterasi dari 1 hingga M, algoritma menghitung gradien negatif $g_t(x)$ dari fungsi kerugian terhadap estimasi fungsi saat ini. Kemudian, algoritma mempelajari model pembelajar dasar baru, $h(x, \theta_t)$, yang sesuai dengan gradien negatif tersebut. Setelah itu, algoritma menentukan ukuran langkah terbaik ρ_t melalui pencarian gradien turun pada fungsi kerugian. Dengan ukuran langkah ini, estimasi fungsi diperbarui dengan menambahkan hasil kali ρ_t dengan model pembelajar dasar yang baru dipelajari. Proses ini diulangi hingga mencapai jumlah iterasi yang ditentukan [17].

2.1.4 LSTM (Long-Short Term Memory)

Algoritma *Long-Short Term Memory* efektif dalam menganalisis sentimen ulasan film karena mampu menangkap ketergantungan urutan dalam data teks [18]. Jaringan LSTM unggul dalam memproses urutan teks panjang dengan menangani masalah gradien yang menghilang melalui mekanisme gerbang, seperti *forgetting gate*, yang memungkinkan model mempertahankan informasi penting dalam jangka waktu panjang [19].

LSTM dikembangkan untuk mengatasi masalah gradien yang meledak dan menghilang yang sering terjadi saat melatih RNN tradisional. Salah satu keunggulan LSTM dibandingkan dengan RNN, model Markov tersembunyi dan metode pembelajaran lainnya adalah ketidakpekaannya terhadap panjang celah.

Terdapat beberapa arsitektur unit LSTM. Arsitektur khas terdiri dari sel (bagian memori dari unit LSTM), dan tiga ”pengatur” atau *gate*, yang mengendalikan aliran informasi di dalam unit LSTM: *input gate, output gate* dan *forget gate*. Beberapa variasi unit LSTM mungkin tidak memiliki satu atau lebih *gate* tersebut atau bahkan dapat menghasilkan *gate* yang berbeda. Ilustrasi arsitektur LSTM dapat dilihat pada Gambar 2.3.



Gambar 2.3 Arsitektur LSTM [20]

Bentuk persamaan untuk langkah maju dari sebuah unit LSTM dengan *forgetting gate* tertera pada persamaan (1), (2), (3), (4), (5) dan (6) berikut ini.

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \quad (1)$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \quad (2)$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \quad (3)$$

$$\tilde{c}_t = \sigma_g(W_c x_t + U_c h_{t-1} + b_c) \quad (4)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \quad (5)$$

$$h_t = o_t \circ \sigma_h(c_t) \quad (6)$$

dimana nilai awalnya adalah $c_0 = 0$ dan $h_0 = 0$ dan operator $\circ$ menunjukkan produk Hadamard (produk berdasarkan elemen). Subskrip t mengindeks langkah waktu. Dalam model ini, σ adalah fungsi aktivasi sigmoid, $tanh$ adalah fungsi aktivasi tangen hiperbolik, x_t adalah input pada waktu t. Sedangkan $W_i, W_c, W_f, W_o, U_i, U_c, U_f, U_o$ adalah matriks bobot untuk mengatur input dan b_i, b_c, b_f, b_o adalah vektor bias [20].

2.1.5 Pengujian Tingkat Keakuratan Hasil Prediksi

Setelah tahapan prediksi menggunakan metode yang tertera telah dilakukan, selanjutnya akan dilakukan proses evaluasi terhadap hasil prediksi dengan menggunakan RMSE (*Root Mean Squared Error*) dan persentase akurasi. RMSE telah digunakan sebagai metrik statistik standar untuk mengukur kinerja model dalam studi penelitian meteorologi, kualitas udara dan iklim [21]. Rumus RMSE dapat dilihat pada persamaan (7) berikut ini.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}} \quad \text{.....(7)}$$

dengan $\hat{y}_i$ adalah nilai yang diprediksi, y_i adalah nilai dari yang diobservasi dan n adalah jumlah observasi. Semakin kecil RMSE, maka hasil yang didapatkan semakin baik.

Lalu dari RMSE dapat dihitung persentase akurasi dengan menggunakan persamaan 8 berikut ini.

$$Accuracy\ Percentation = \left(1 - \frac{RMSE}{R}\right) \times 100 \quad \text{.....(8)}$$

dengan R adalah rentang variabel target yang dalam kasus ini merupakan 10, karena *rating* dari film berkisar antara 0 sampai dengan 10.

2.2 Penelitian Terdahulu

Pada Januari 2022 Agarwal dan Venugopal [10] menggunakan *dataset* IMDb *extensive* dari Kaggle dan menerapkan berbagai algoritma termasuk *Simple Linear Regression, Multiple Linear Regression* dan *Artificial Neural Network*. Fitur-fitur yang digunakan mencakup *rating*, *genre, rating voter* teratas, total suara, durasi dan tanggal rilis. Mereka menemukan bahwa *Artificial Neural Network* memberikan hasil terbaik dengan akurasi 86%. Namun, penelitian tersebut tidak menggunakan beberapa fitur kategorikal.

Pada tahun 2023 Sindhu dan Shamsi [1] melakukan penelitian menggunakan *dataset* IMDb serta *scrapping* data dari Facebook terhadap 210 film. Mereka membandingkan dua algoritma yaitu *Linear Regression* dan SVM. Fitur-fitur yang digunakan adalah yang berhubungan dengan Facebook yakni *director FB likes, actor FB likes, actress FB likes, movie FB likes, movie budget, likes on FB movie page* dan *sentiment score of FB movie page*. Hasilnya menunjukkan bahwa SVM mendapatkan akurasi 84% ketika skor sentimen dari Facebook ditambahkan sebagai sebuah fitur. Ini menunjukkan bahwasannya skor sentimen juga penting dalam memprediksi kesuksesan sebuah film. Namun, kekurangan dari penelitian tersebut adalah *dataset* yang digunakan terlalu sedikit karena hanya mencakup 210 film.

Pada tahun 2020, Ruwantha dan Kumara [2] melakukan klasifikasi terhadap unggahan Twitter untuk memprediksi kesuksesan film. *Dataset* yang digunakan adalah *scrapping* unggahan Twitter tentang 500 film. Mereka menggunakan algoritma LSTM. Atribut yang digunakan adalah *tweet* dan *label*. Hasilnya menunjukkan bahwa LSTM mendapatkan akurasi

83,97%. Pada bagian kesimpulan, disarankan penelitian selanjutnya menggunakan metode *ensemble* untuk memprediksi kesuksesan film.

Pada Maret 2023 Tripathi *et al.* [4] menggunakan *dataset* IMDb serta *dataset* ulasan film. Mereka menerapkan beberapa algoritma termasuk *Simple Regression Tree, Random Forest, Linear Regression* dan beberapa klasifikasi seperti *Linear SVC* dan *Naive Bayes Classifier*. Fitur-fitur yang digunakan untuk prediksi meliputi *genre,* durasi, anggaran, popularitas kru dan rasio aspek. Hasilnya menunjukkan bahwa *Linear Regression* memberikan performa terbaik untuk prediksi menggunakan fitur film, sementara *Linear SVC* adalah algoritma terbaik untuk analisis sentimen dengan akurasi 88,47%. Namun, penelitian tersebut kurang lengkap dalam penggunaan fitur. Fitur seperti tahun rilis, pemeran serta negara produksi belum ada terlihat digunakan.

Pada Oktober 2023 Gandasari *et al.* [5] menggunakan *dataset* Netflix yang diperoleh dari hasil pengambilan data dengan Python. Mereka menerapkan beberapa algoritma termasuk *Random Forest, Naive Bayes* dan *K-Nearest Neighbor*. Fitur-fitur yang digunakan mencakup jenis film, tahun rilis, sertifikasi usia, durasi, *genre* dan beberapa metrik seperti skor IMDb dan popularitas TMDB. Hasilnya menunjukkan bahwa *Random Forest* memiliki akurasi tertinggi sebesar 81,59%. Namun, penelitian tersebut juga belum terlihat menggunakan fitur pemeran dan negara produksi.

Berdasarkan penjabaran keterkaitan penelitian-penelitian yang telah dilakukan sebelumnya, maka kontribusi dari penelitian ini adalah melakukan prediksi kesuksesan film berdasarkan data fitur dan komentar *trailer* dengan model *ensemble* (*Random Forest+Gradient Boosting*) dan LSTM. Hasil prediksi diharapkan dapat membantu produser film untuk mengatur strategi pemasaran terhadap film setelah *trailer* diunggah.