BAB II
LANDASAN TEORI

2.1 Landasan Teori

2.1.1 Data Mining

Data Mining adalah sebuah proses untuk menemukan pola atau hubungan dari sejumlah besar data dalam basis data relasional yang kompleks. Proses ini juga dikenal sebagai serangkaian tahapan untuk menggali informasi yang bernilai tambah dan sebelumnya tidak diketahui. Informasi tersebut diperoleh dengan cara menganalisis serta mengenali pola-pola menarik dalam data yang ada di basis data (Siregar, Kom, Puspabhuana, Kom, & Kom, 2017). Data Mining sering kali digunakan untuk mengungkap pengetahuan yang tersembunyi di dalam basis data yang besar, sehingga juga dikenal dengan istilah *Knowledge Discovery in Databases* (Suhirman, 2023).

2.1.2 Text Preprocessing

Text Preprocessing adalah tahap awal dalam Text Mining yang bertujuan untuk mengubah data teks tidak terstruktur menjadi data terstruktur yang dapat dianalisis lebih lanjut. Tujuan preprocessing adalah menghasilkan indeks istilah yang dapat mewakili dokumen secara efektif (Kurniasari, Santoso, & Prahutama, 2021). Tahapan-tahapannya meliputi:

1. Case Folding

   Case folding adalah proses standarisasi teks dengan mengonversi seluruh huruf kapital dalam dokumen menjadi huruf kecil. Transformasi ini hanya berlaku pada karakter huruf dari 'a' hingga 'z', sementara karakter lain seperti tanda baca, koma, atau spasi yang dikenal sebagai delimiter atau karakter khusus dapat diabaikan atau dihapus menggunakan perintah dalam Python (Rofiqi & Akbar, 2024).

2. Stemming

   Stemming adalah proses menghilangkan imbuhan pada sebuah kata untuk mendapatkan bentuk dasar atau akar kata seperti "berlari" yang diubah menjadi "lari" atau "memasak" yang diubah menjadi "masak". Tujuan dari stemming adalah mengurangi variasi kata yang tidak diperlukan agar meningkatkan efisiensi dalam pemrosesan teks (Albab & Fawaiq, 2023).

3. Stopword Removal

   Stopword removal adalah proses seleksi kata-kata penting dengan menghilangkan kata-kata yang dianggap tidak memiliki manfaat dalam pemrosesan teks. Stopword biasanya terdiri dari kata-kata yang sering muncul, seperti kata ganti orang, "dan", "atau", "juga", "dari", "ke", dan sejenisnya. Proses ini dilakukan dengan membandingkan kata dalam dokumen dengan daftar stopword, di mana kata yang sesuai akan dihapus (Lestari & Saepudin, 2021).

4. Tokenizing

   Tokenizing adalah proses memecah kalimat menjadi unit-unit yang lebih kecil atau kata-kata terpisah. Pemisahan ini dilakukan berdasarkan spasi antar kata. Selain itu, angka, tanda baca, dan karakter khusus juga dapat dihapus pada tahap ini karena dianggap tidak berpengaruh dalam pemrosesan teks (Mualfah, Prihatin, & Firdaus, 2023).

2.1.3 Pelabelan Data

Pelabelan data adalah proses memberikan label atau anotasi pada data yang tidak berlabel untuk membantu komputer mengenali pola di dalamnya. Pelabelan ini penting dalam pembelajaran mesin untuk mengajarkan model membedakan antara berbagai kategori data, seperti pada pengenalan gambar, pemrosesan teks, dan lainnya (Alhaqq, Putra, & Ruldeviyani, 2022).

2.1.4 TF-IDF

TF-IDF adalah metode pengukuran untuk menentukan tingkat kepentingan kata dalam dokumen. *Term Frequency* (TF) menunjukkan seberapa sering suatu kata muncul dalam dokumen yang ditunjukkan pada rumus 2.1. Sedangkan *Inverse Document Frequency* (IDF) mengukur seberapa jarang suatu kata muncul di seluruh dokumen, memberikan bobot lebih pada kata-kata yang lebih unik atau jarang digunakan yang ditunjukkan pada rumus 2.2. Perhitungan TF-IDF yang ditunjukkan pada rumus 2.3. Metode ini menggabungkan dua konsep tersebut untuk menghasilkan bobot yang akurat (Husain, Sukirman, & SAJIAH, 2024).

$$TF_{d,t} = \frac{\text{jumlah munculnya kata t dalam dokumen Total jumlah}}{\text{keseluruhan kata dalam dokume}} \quad (2.1)$$

$$IDF = \log_2\left(\frac{D}{df}\right) \quad (2.2)$$

$$W_{d,t} = TF_{d,t} * IDF \quad (2.3)$$

2.1.5 Klasifikasi

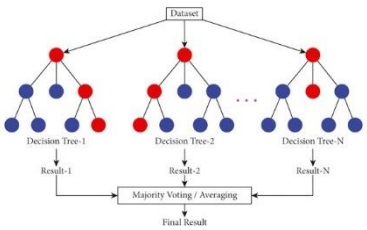
Klasifikasi adalah suatu cara mengelompokkan data berdasarkan ciri-cirinya, kemudian memperkirakan kelompok mana yang cocok dengan data baru tersebut (Sihombing & Yuliati, 2021). Dalam klasifikasi, model dibangun dengan menggunakan data yang sudah dilabeli (*supervised learning*) untuk memprediksi kelas dari data baru yang belum pernah dilihat sebelumnya (Mestika, Selan, & Qadafi, 2023). Contoh dari klasifikasi adalah ketika algoritma digunakan untuk mengkategorikan buah untuk pemeriksaan harga. Proses ini melibatkan pembelajaran pola dari data pelatihan, sehingga algoritma dapat memberikan prediksi yang akurat ketika diberikan data baru (Kristiawan, Somali, & Widjaja, 2020).

Klasifikasi biasanya melibatkan dua langkah utama yaitu pelatihan dan prediksi. Pada tahap pelatihan, model belajar dari dataset yang berisi input dan label yang sesuai. Pada tahap prediksi, model yang sudah dilatih akan menerapkan pengetahuan tersebut untuk memprediksi kelas dari data baru yang belum diberi

label (Burhanuddin, 2024). Klasifikasi machine learning banyak digunakan dalam berbagai aplikasi seperti pengenalan gambar, analisis teks, dan diagnosis medis (Arminarahmah & Mahalisa, 2024; Kristiawan et al., 2020; Pramann et al., 2023).

2.1.6 Random Forest

Random Forest adalah salah satu algoritma *ensemble learning* yang populer dalam machine learning, yang menggabungkan beberapa pohon keputusan (*decision trees*) untuk meningkatkan akurasi dan mengurangi overfitting. Random Forest bekerja dengan membangun sejumlah pohon keputusan dari subset acak data pelatihan dan menggabungkan hasil prediksi dari masing-masing pohon melalui voting mayoritas (untuk klasifikasi) atau rata-rata (untuk regresi) (Mario & Suryono, 2025). Proses ini dikenal sebagai bagging (*bootstrap aggregating*), yang membantu mengurangi varians dan risiko overfitting. Gambar 2.1 menunjukkan gambaran umum alur kerja Random Forest.



Gambar 2. 1 Alur Kerja Random Forest.

Dalam pembentukan Random Forest, setiap pohon keputusan dibangun dengan memilih subset fitur secara acak pada setiap split node. Hal ini memastikan bahwa setiap pohon memiliki keragaman yang tinggi, sehingga prediksi gabungan dari semua pohon cenderung lebih stabil dan akurat. Selain itu, Random Forest mampu menangani data dengan banyak fitur tanpa perlu reduksi dimensi, karena hanya subset fitur yang dipilih secara acak yang dipertimbangkan pada setiap split. Keunggulan lain dari Random Forest adalah kemampuannya untuk mengukur pentingnya fitur (*feature importance*), yang berguna untuk analisis lebih lanjut tentang kontribusi setiap fitur dalam proses prediksi. (Yulianto, Fanani, Affandy, & Aziz, 2024).

2.1.7 Teknik Evaluasi Model

Evaluasi model merupakan langkah krusial untuk menilai kinerja model machine learning. Salah satu alat evaluasi yang sering digunakan adalah confusion matrix, yang menampilkan performa model klasifikasi dengan membandingkan hasil prediksi model terhadap label aktual. Confusion matrix terdiri dari empat elemen utama: True Positive (TP), False Positive (FP), True Negative (TN), dan False Negative (FN). Berdasarkan confusion matrix, beberapa metrik evaluasi dapat dihitung, seperti akurasi, precision, recall, dan F1-score.

2.1.8 Visualisasi data

Visualisasi data adalah proses menyajikan data dalam bentuk grafis atau visual, seperti grafik, diagram, atau peta, untuk memudahkan pemahaman dan analisis informasi. Visualisasi data membantu dalam menemukan pola, tren, dan anomali dalam data yang mungkin tidak terlihat melalui analisis numerik biasa. Contoh umum visualisasi data adalah grafik batang, diagram garis, peta panas (heatmap), dan diagram sebar (scatter plot) (Sudipa et al., 2023).

Dalam konteks penelitian ilmiah, visualisasi data sangat penting karena seringkali peneliti bekerja dengan dataset yang sangat besar dan kompleks. Visualisasi membantu peneliti untuk melihat hubungan antar variabel dan memahami hasil dari model machine learning (Perkasa & Rahmatulloh, 2024). Dengan visualisasi, hasil prediksi tren penelitian, misalnya, dapat dengan mudah dipahami oleh penerbit jurnal untuk membantu mereka dalam mengambil keputusan yang lebih baik terkait topik publikasi yang relevan.

2.2 Penelitian Terkait

a. Penelitian oleh Ferisa Dwi Alfia Meisty, Dian Anggraeni, dan Mohamat Fatekurohman (2024) berjudul “Perbandingan Metode Naïve Bayes Classifier dengan Metode Random Forest pada Prediksi Rating Review Drama Korea”. Penelitian ini bertujuan untuk mengklasifikasikan review drama Korea ke dalam kategori Bagus, Tidak Bagus, atau Cukup Bagus, serta membandingkan

performa metode Naïve Bayes Classifier dan Random Forest. Data diperoleh dari IMD dan melalui tahap preprocessing, termasuk pembersihan teks dan pelabelan, yang kemudian dibagi menjadi data latih dan data uji. Evaluasi dilakukan menggunakan akurasi, precision, recall, dan F1-score, yang menunjukkan bahwa Random Forest memiliki akurasi lebih tinggi sebesar 89% dalam prediksi review dibandingkan dengan Naïve Bayes yang memiliki akurasi 86%. Dalam prediksi rating, Random Forest juga unggul dengan akurasi 41% dibandingkan Naïve Bayes yang mencapai 40%, sehingga disimpulkan bahwa Random Forest lebih baik dalam memprediksi rating review drama Korea (Meisty, Anggraeni, & Fatekurohman, 2024).

b. Penelitian yang dilakukan oleh Juwariyem, Sriyanto, Sri Lestari, dan Chairani, 2024, berjudul "Prediction of Stunting in Toddlers Using Bagging and Random Forest Algorithms" bertujuan untuk meningkatkan akurasi prediksi stunting pada balita dengan menerapkan teknik Bagging dan algoritma Random Forest. Penelitian ini menggunakan dataset sebanyak 10.001 data, dengan 7 atribut sebagai variabel input dan 1 atribut kelas sebagai target prediksi. Teknik Bagging digunakan untuk menangani ketidakseimbangan data, sedangkan Random Forest diterapkan untuk membangun model prediksi yang lebih akurat. Hasilnya menunjukkan bahwa model yang dikembangkan memiliki akurasi sebesar 91.98%, dengan precision kelas *yes* 91.72%, recall kelas *yes* 98.84%, precision kelas *no* 93.55%, dan recall kelas *no* 65.28%, sehingga metode ini dapat digunakan sebagai alat bantu dalam pengambilan keputusan terkait intervensi pencegahan stunting pada balita (Lestari, 2024).

c. Penelitian yang dilakukan oleh Saman Behrouzi, Zahra Shafaeipour Sarmoor, Khosrow Hajsadeghi, dan Kaveh Kavousi, 2020, berjudul “Predicting Scientific Research Trends Based on Link Prediction in Keyword Networks”. Penelitian ini bertujuan untuk memprediksi tren penelitian ilmiah masa depan dengan menganalisis jaringan kata kunci di bidang ilmu komputer. Dua metode utama yang digunakan adalah prediksi hubungan berbasis topologi

dan algoritma pembelajaran mesin, seperti Support Vector Machine (SVM) dan Random Forest. Hasilnya menunjukkan bahwa metode yang diterapkan dapat memprediksi tren penelitian dengan akurat, terutama pada jaringan kata kunci di konferensi ilmiah (Behrouzi, Sarmoor, Hajsadeghi, & Kavousi, 2020).

d. Penelitian yang dilakukan oleh Widya Apriliah, Ilham Kurniawan, Muhamad Baydhowi, dan Tri Haryati, 2021, berjudul "Prediksi Kemungkinan Diabetes pada Tahap Awal Menggunakan Algoritma Klasifikasi Random Forest". Tujuan dari penelitian ini adalah untuk merancang model yang dapat memprediksi kemungkinan terjadinya diabetes pada tahap awal dengan akurasi yang tinggi. Penelitian ini menggunakan tiga algoritma klasifikasi, yaitu Random Forest, Support Vector Machine (SVM), dan Naive Bayes, serta dataset yang diambil dari UCI Repository. Hasil evaluasi menunjukkan bahwa algoritma Random Forest memberikan akurasi tertinggi sebesar 97,88% dibandingkan algoritma lainnya, sehingga direkomendasikan sebagai metode terbaik untuk deteksi dini diabetes (Apriliah, Kurniawan, Baydhowi, & Haryati, 2021).

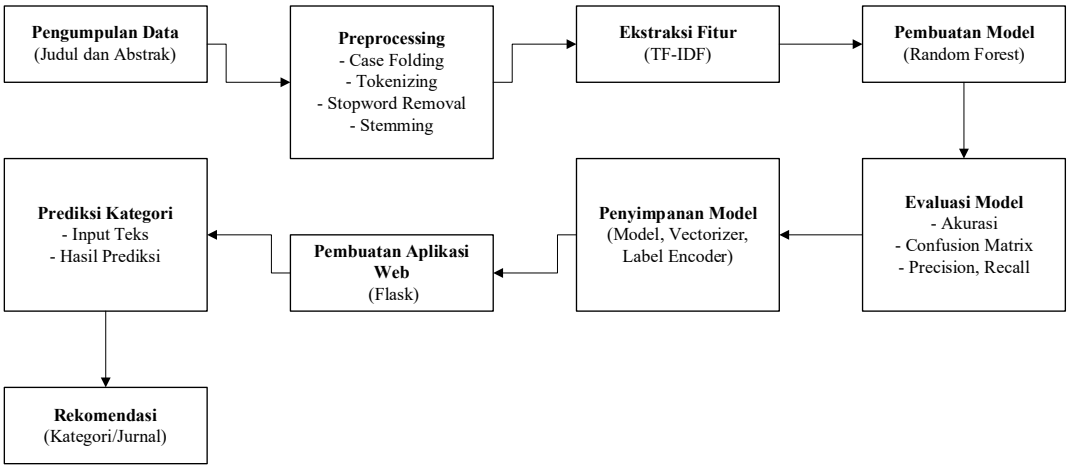
e. Penelitian yang dilakukan oleh Hendri Mahmud Nawawi, Agung Baitul Hikmah, Ali Mustopa, dan Ganda Wijaya, 2024, berjudul "Model Klasifikasi Machine Learning untuk Prediksi Ketepatan Penempatan Karir". Penelitian ini bertujuan untuk memprediksi penempatan karir yang sesuai berdasarkan data pendidikan dan pengalaman kerja. Lima algoritma diuji dalam penelitian ini, yaitu Random Forest, Decision Tree, Naive Bayes, K-Nearest Neighbor (KNN), dan Support Vector Machine (SVM) menggunakan dataset Job Placement dari Kaggle. Dari hasil evaluasi, Random Forest menunjukkan kinerja terbaik dengan akurasi sebesar 87% dan nilai AUC sebesar 0,93. Faktor yang paling berpengaruh dalam penempatan karir adalah "ssc_percentage" atau persentase ujian sekolah menengah (Nawawi, Hikmah, Mustopa, & Wijaya, 2024).

Tabel 2. 1 Penelitian Terkait

No	Peneliti	Judul Penelitian	Tujuan	Metode dan Data	Hasil
1	Ferisa Dwi Alfia Meisty, Dian Anggraeni, Mohamat Fatekurohman (2024)	Perbandingan Metode Naïve Bayes Classifier dengan Metode Random Forest pada Prediksi Rating Review Drama Korea	Mengklasifikasikan review drama Korea dan membandingkan performa Naïve Bayes dengan Random Forest	Data dari IMD, preprocessing teks, evaluasi dengan akurasi, precision, recall, dan F1-score	Random Forest unggul dengan akurasi 89% dibandingkan Naïve Bayes 86%; prediksi rating Random Forest (41%) lebih baik
2	Juwariyem, Sriyanto, Sri Lestari, Chairani (2024)	Prediction of Stunting in Toddlers Using Bagging and Random Forest Algorithms	Memprediksi kejadian stunting pada balita dengan meningkatkan akurasi prediksi menggunakan algoritma Bagging dan Random Forest	Menggunakan Bagging dan Random Forest pada dataset 10.001 data dengan 7 atribut dan 1 kelas	Akurasi model mencapai 91.98% dengan precision kelas yes 91.72%, recall kelas yes 98.84%, precision kelas no 93.55%, dan recall kelas no 65.28%
3	Saman Behrouzi, Zahra Shafaeipour Sarmoor, Khosrow Hajsadeghi, Kaveh Kavousi (2020)	Predicting Scientific Research Trends Based on Link Prediction in Keyword Networks	Memprediksi tren penelitian ilmiah dengan analisis jaringan kata kunci	Prediksi hubungan berbasis topologi dan pembelajaran mesin (SVM dan Random Forest)	Metode berhasil memprediksi tren penelitian dengan akurat
4	Widya Apriliah, Ilham Kurniawan, Muhamad Baydhowi, Tri Haryati (2021)	Prediksi Kemungkinan Diabetes pada Tahap Awal Menggunakan Algoritma Klasifikasi Random Forest	Memprediksi kemungkinan diabetes pada tahap awal dengan akurasi tinggi	Data dari UCI Repository, algoritma Random Forest, SVM, dan Naïve Bayes, evaluasi akurasi	Random Forest unggul dengan akurasi 97,88%, direkomendasikan untuk deteksi dini diabetes

5	Hendri Mahmud Nawawi, Agung Baitul Hikmah, Ali Mustopa, Ganda Wijaya (2024)	Model Klasifikasi Machine Learning untuk Prediksi Ketepatan Penempatan Karir	Memprediksi penempatan karir berdasarkan data pendidikan dan pengalaman kerja	Data dari Kaggle (Job Placement), algoritma Random Forest, Decision Tree, Naïve Bayes, KNN, dan SVM, evaluasi AUC	Random Forest unggul dengan akurasi 87% dan AUC 0,93; faktor utama: persentase ujian sekolah menengah (ssc_percentage)

2.3 Kerangka Berfikir



Gambar 2. 2 Kerangka Berfikir

1. Pengumpulan Data

Tahap ini merupakan langkah awal dalam penelitian. Dataset yang berisi dokumen teks (seperti judul dan abstrak penelitian) beserta kategori/labelnya dikumpulkan. Dataset ini menjadi fondasi untuk membangun model klasifikasi teks. Data yang digunakan harus relevan dan representatif agar model yang dihasilkan dapat bekerja dengan baik.

2. Preprocessing Teks

Setelah data terkumpul, langkah selanjutnya adalah membersihkan dan mempersiapkan teks agar siap diproses lebih lanjut. Preprocessing meliputi beberapa tahap:

- Case Folding: Mengubah semua teks menjadi huruf kecil untuk memastikan konsistensi.
- Menghapus Karakter Non-Alfabet: Menghilangkan simbol, angka, atau karakter khusus yang tidak relevan.
- Tokenizing: Memecah teks menjadi kata-kata individual (token).
- Stopword Removal: Menghapus kata-kata umum yang tidak memiliki makna signifikan (seperti "dan", "di", "yang").

- Stemming: Mengubah kata ke bentuk dasarnya (misalnya, "menulis" menjadi "tulis").
- Preprocessing ini penting untuk mengurangi noise dan memastikan teks siap untuk ekstraksi fitur.

3. Ekstraksi Fitur

Setelah teks dibersihkan, langkah selanjutnya adalah mengubah teks menjadi representasi numerik yang dapat diproses oleh model machine learning. Metode yang digunakan adalah TF-IDF (*Term Frequency-Inverse Document Frequency*), yang menghitung bobot kata berdasarkan frekuensi kemunculannya dalam dokumen dan keunikan kata tersebut di seluruh dokumen. Hasilnya adalah vektor numerik yang merepresentasikan teks.

4. Pembuatan Model

Pada tahap ini, model klasifikasi teks dibangun menggunakan algoritma Random Forest. Random Forest dipilih karena kemampuannya dalam menangani data teks dan ketahanannya terhadap *overfitting*. Dataset dibagi menjadi dua bagian: data latih (*train*) dan data uji (*test*). Model dilatih menggunakan data latih dan diuji menggunakan data uji untuk memastikan bahwa model dapat bekerja dengan baik pada data baru.

5. Evaluasi Model

Setelah model dibuat, performanya dievaluasi menggunakan beberapa metrik, seperti:

- Akurasi: Persentase prediksi yang benar.
- Precision: Proporsi prediksi positif yang benar.
- Recall: Proporsi data positif yang berhasil diprediksi.
- F1-Score: Rata-rata harmonik dari precision dan recall.

Selain itu, confusion matrix digunakan untuk menganalisis performa model pada setiap kategori. Evaluasi ini membantu menentukan apakah model sudah cukup baik atau perlu diperbaiki.

6. Penyimpanan Model

Jika model telah memenuhi kriteria performa yang diharapkan, model, *vectorizer* (TF-IDF), dan *label encoder* disimpan menggunakan *pickle* atau *joblib*. Penyimpanan ini memungkinkan model dan komponen pendukungnya digunakan kembali tanpa perlu melatih ulang. File-file ini nantinya akan diintegrasikan ke dalam aplikasi web.

7. Pembuatan Aplikasi Web

Untuk memungkinkan penggunaan model secara praktis, aplikasi web dibangun menggunakan framework Flask. Aplikasi ini dirancang dengan antarmuka pengguna yang sederhana, di mana pengguna dapat memasukkan teks (judul dan abstrak) melalui form input. Aplikasi web akan memproses teks tersebut menggunakan model yang telah disimpan dan menampilkan hasil prediksi kategori.

8. Prediksi

Pada tahap ini, aplikasi web sudah siap digunakan untuk prediksi. Pengguna memasukkan teks, dan aplikasi akan melakukan:

- Preprocessing teks (case folding, tokenizing, stopword removal, stemming).
- Ekstraksi fitur menggunakan TF-IDF.
- Prediksi kategori menggunakan model Random Forest.

Hasil prediksi kemudian ditampilkan kepada pengguna, memberikan informasi tentang kategori yang sesuai dengan teks yang dimasukkan.

9. Rekomendasi

Selain menampilkan hasil prediksi, aplikasi web juga dapat memberikan rekomendasi jurnal atau kategori berdasarkan hasil prediksi. Misalnya, jika teks diprediksi masuk ke kategori "Keuangan dan Perusahaan", aplikasi dapat merekomendasikan jurnal-jurnal terkait bidang tersebut. Fitur ini meningkatkan nilai tambah dari aplikasi yang dibangun.

2.4 Hipotesis Penelitian

1. H1: Algoritma Random Forest dapat diterapkan secara efektif untuk memprediksi kategori artikel berdasarkan fitur teks seperti judul dan abstrak, dengan akurasi yang memadai dan performa yang stabil.
2. H2: Sistem berbasis algoritma Random Forest mampu memberikan rekomendasi jurnal yang sesuai dengan scope keilmuan artikel berdasarkan hasil prediksi kategori, dengan tingkat presisi dan recall yang tinggi.
3. H3: Aplikasi berbasis web yang dikembangkan dapat meningkatkan efisiensi proses seleksi artikel dengan mengurangi waktu yang dibutuhkan untuk penempatan artikel ke jurnal yang sesuai, serta meminimalkan kesalahan akibat penempatan artikel ke jurnal yang tidak relevan.