

BAB I

PENDAHULUAN

1.1.Latar Belakang Masalah

Perkembangan teknologi digital yang pesat telah mengubah lanskap industri media secara signifikan di Indonesia. Berbagai *platform* seperti televisi, internet dan media sosial semakin terintegrasi, konsumen memiliki akses yang lebih fleksibel dan personal terhadap konten media. Fenomena ini menghadirkan tantangan baru bagi stasiun TV di Indonesia untuk berinovasi dan menarik untuk mempertahankan audiens mereka. Platform seperti Netflix, Vidio, Disney+ Hotstar, Prime Video, dan Catchplay kini bersaing ketat untuk menarik dan mempertahankan pelanggan. Dalam lanskap kompetisi yang dinamis ini, pemahaman terhadap preferensi dan kepuasan pengguna menjadi kunci utama untuk keberlangsungan bisnis.

Salah satu sumber data yang paling autentik, *real time*, dan mudah diakses untuk mengukur kepuasan pengguna adalah ulasan pengguna di platform digital seperti *Google Play Store*. Ulasan ini berisi sentimen (opini atau perasaan) yang mencerminkan pengalaman langsung pengguna terhadap kualitas layanan, konten, dan kinerja aplikasi. Analisis sentimen bertujuan mengelompokkan (klasifikasi) polaritas teks ke dalam kategori positif, negatif, atau netral.

Meskipun ulasan pengguna sangat melimpah, masih terdapat kesenjangan dalam penelitian yang ada. Belum optimalnya perbandingan kinerja antara dua algoritma *Machine learning* (ML) yang populer seperti Naive Bayes dan Support Vector Machine (SVM) dalam konteks klasifikasi sentimen ulasan aplikasi *Over The Top* (OTT) berbahasa Indonesia. Selain itu, jarang ada penelitian yang secara eksplisit menghubungkan hasil klasifikasi sentimen positif tersebut dengan metrik bisnis seperti rating aplikasi di *Google Play Store*. Analisis ini diperlukan untuk menguji hipotesis mengenai seberapa besar pengaruh sentimen terhadap popularitas aplikasi.

Untuk mengatasi kesenjangan tersebut, penelitian ini mengusulkan metode klasifikasi sentimen otomatis menggunakan pendekatan *Supervised Learning*. Metode yang digunakan adalah algoritma Naive Bayes dan SVM. Dengan membandingkan kedua metode, diharapkan dapat mengidentifikasi kelebihan dan kelemahan masing-masing, serta memberikan wawasan bagi pengembang aplikasi untuk meningkatkan pengalaman pengguna dan kualitas produk di pasar yang kompetitif. Data ulasan diambil dari *Google Play Store*. Data penelitian disajikan dalam laporan yang menyeluruh dan menyediakan pemahaman yang lebih baik tentang efektivitas klasifikasi sentimen. Selanjutnya, penelitian menggunakan Koefisien Korelasi Pearson untuk mengukur hubungan linier antara persentase sentimen positif dengan jumlah rating aplikasi. Salah satu metode statistik yang umum digunakan untuk mengukur hubungan antara dua variabel adalah korelasi Pearson. Korelasi Pearson mengukur kekuatan dan arah hubungan. Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi metodologis dalam ranah *Natural Language Processing* (NLP) dan memberikan wawasan strategis yang jelas bagi penyedia layanan OTT.

1.2. Perumusan Masalah

Berdasarkan uraian latar belakang yang menunjukkan perlunya pemahaman sistematis terhadap sentimen pengguna aplikasi OTT, permasalahan utama dalam tugas akhir ini dirumuskan sebagai berikut: Membangun model klasifikasi sentimen pada ulasan aplikasi OTT (Netflix, Vidio, Disney+ Hotstar, Prime Video, dan Catchplay) di *Google Play Store* menggunakan Algoritma Naive Bayes dan Support Vector Machine (SVM). Pada penelitian ini juga mengukur perbandingan kinerja kedua algoritma tersebut berdasarkan metrik evaluasi seperti akurasi dan *recall*. Selanjutnya, penelitian ini juga menganalisis apakah terdapat korelasi yang signifikan antara hasil sentimen positif yang diperoleh dengan jumlah rating dari aplikasi-aplikasi OTT tersebut.

1.3. Batasan Masalah

Pada penelitian ini membatasi permasalahan diantaranya

1.3.1 Objek Penelitian

Objek penelitian adalah lima Aplikasi OTT yaitu Netflix, Vidio, Disney+ Hotstar, Prime Video, dan Catchplay.

1.3.2 Ulasan

Tiap Aplikasi OTT akan diambil 1000 ulasan dari *Google Play Store* dengan kriteria ulasan yang diambil adalah dari negara Indonesia, disortir yang paling relevan, dan ulasan dengan skor satu sampai lima (semua skor)

1.3.3 Model Klasifikasi

Pemodelan klasifikasi ulasan menggunakan *machine learning* Naive Bayes dan SVM. Hasil klasifikasi akan dievaluasi dengan matrik evaluasi.

1.3.4 Analisa Korelasi

Analisa Korelasi Pearson digunakan untuk menganalisa hubungan hasil klasifikasi sentimen dengan rating aplikasi OTT pada *Google Play Store*.

1.4. Tujuan Dan Manfaat

1.4.1 Tujuan

- a) Membangun sebuah model klasifikasi sentimen untuk ulasan aplikasi OTT di *Google Play Store*. Model klasifikasi sentimen menggunakan algoritma *Naive Bayes* dan SVM, untuk mengkategorikan ulasan menjadi sentimen positif, negatif, atau netral.
- b) Menganalisis dan membandingkan sentimen dari lima aplikasi OTT yang diteliti (Netflix, Vidio, Disney+ Hotstar, Prime Video, dan Catchplay+) berdasarkan hasil klasifikasi.
- c) Menghubungkan hasil klasifikasi sentimen dengan *rating* aplikasi untuk menguji hipotesis bahwa sentimen ulasan memiliki korelasi dengan popularitas suatu aplikasi.

1.4.2 Manfaat

a) Manfaat Untuk Mahasiswa

Penelitian ini memberikan manfaat yang signifikan bagi Mahasiswa sebagai berikut:

- 1) Menjadi sarana untuk menerapkan dan memperdalam pemahaman teoretis mengenai bidang *Natural Language Processing (NLP)*, *Data Mining*, dan *ML* yang telah diperoleh selama masa perkuliahan.
- 2) Memberikan pengalaman langsung dalam pengolahan data teks dalam skala besar, implementasi Algoritma *Naive Bayes* dan *Support Vector Machine (SVM)*, serta analisis statistik menggunakan Koefisien Korelasi *Pearson*.
- 3) Memenuhi salah satu syarat kelulusan dan penyelesaian studi di Program Studi Informatika, Fakultas Sains, Teknologi dan Kesehatan, Universitas Sahid Surakarta.

b) Manfaat Untuk Universitas Sahid Surakarta

- 1) Hasil penelitian ini dapat memperkaya khazanah ilmiah di lingkungan Universitas Sahid Surakarta, khususnya dalam bidang Teknik Informatika dan *Data Science*.
- 2) Laporan Tugas Akhir ini dapat dijadikan sebagai bahan bacaan, referensi, atau studi komparasi bagi Mahasiswa dan Dosen Program Studi Informatika yang akan melakukan penelitian sejenis di masa mendatang.
- 3) Turut berkontribusi dalam publikasi ilmiah yang dapat meningkatkan peringkat dan kredibilitas Program Studi Informatika di mata publik dan lembaga akreditasi.

c) Manfaat Untuk *End User*

- 1) Menyediakan pemahaman yang objektif dan terstruktur tentang sentimen pasar dari ulasan pengguna. Hasil perbandingan kinerja model, sentimen terklasifikasi, dan

analisis korelasi dapat digunakan untuk membantu penyedia layanan OTT.

- 2) Hasil perbandingan sentimen dari lima aplikasi OTT dapat menjadi informasi yang transparan dan terukur untuk membantu calon pengguna aplikasi OTT mengevaluasi kualitas layanan secara digital sebelum membuat keputusan untuk berlangganan atau mengunduh aplikasi.

1.5. Metodologi Penelitian

Penelitian ini menggunakan metode klasifikasi dengan mengikuti Metode *Knowledge Discovery in Databases (KDD)*. Metode KDD adalah mekanisme untuk menemukan informasi penting dari data yang sudah ada dalam basis data. KDD dan *data mining* adalah metode yang digunakan untuk menemukan informasi tersembunyi dari data yang sangat besar.

1.5.1 Pengumpulan Data Dengan Teknik *Web Scraping*

Pengumpulan Data merupakan tahapan pertama dalam KDD, pada tahap ini dilakukan pengumpulan, seleksi, dan pelabelan data menggunakan bahasa pemrograman *Python*. Dalam tahapan ini, data yang terbaru dipilih. Untuk analisis ini, data yang dipilih adalah ulasan aplikasi Netflix, Vidio, Disney+ Hotstar, Prime Video, dan Catchplay+ dari *Google Play Store*. Data harus mencakup baik ulasan positif dan negatif untuk memastikan analisis yang komprehensif.

Tahap pengumpulan data dilakukan dengan memanfaatkan teknik *Web Scraping* pada platform *Google Play Store*, yang berfungsi sebagai sumber primer untuk ulasan pengguna. Proses ini berfokus pada 5 aplikasi OTT. Aplikasi OTT yang diteliti adalah Netflix, Vidio, Prime Video, Disney+ Hotstar, dan Catchplay.

Untuk tujuan pengambilan data ini, digunakan library *google_play_scraper* di lingkungan pemrograman *Python*. Parameter yang ditetapkan dalam proses *Scraping* adalah sebagai berikut:

- a) Data diakses menggunakan *package name* unik setiap aplikasi, sebagai contoh kode yang digunakan menargetkan ulasan untuk aplikasi dengan Nama Paket seperti Tabe 1.1. Paket Aplikasi.

Tabel 1.1 Paket Aplikasi

Aplikasi	Nama Paket
Netflix	com.netflix.mediaclient
Vidio	com.vidio.android
Disney+ Hotstar	in.startv.hotstar.dplus
Prime Video	com.amazon.avod.thirdpartyclient
Catchplay	com.catchplay.asiaplay

- b) Proses *scraping* diatur secara spesifik untuk mengambil ulasan berbahasa Indonesia dan berlokasi di Indonesia guna memastikan relevansi data dengan konteks lokal.
- c) Jumlah ulasan yang ditargetkan untuk diambil adalah sebanyak 1.000 ulasan untuk setiap aplikasi. Jumlah ulasan ini memastikan ketersediaan data yang memadai untuk pelatihan model ML.
- d) Ulasan disortir berdasarkan kategori untuk mendapatkan ulasan yang paling relevan bagi pengguna. Semua ulasan kemudian diambil mulai dari ulasan berbintang 1 sampai 5 untuk diikutsertakan dalam analisis.

1.5.2 *Text processing*

Text Preprocessing adalah tahap awal yang bertujuan untuk mengolah dan mengubah data ke dalam format yang sesuai sebelum dilakukan analisis lebih lanjut. Langkah-langkah yang dilakukan meliputi:

a) Seleksi Kolom

Dengan menggunakan teknik Seleksi Kolom Eksplisit / *Explicit Column Selection* dilakukan penetapan kolom yang digunakan. Kolom *content* yang berisi ulasan pengguna dan kolom skor yang berisi *Label* sentimen. Proses ini bertujuan untuk mengisolasi variabel-variabel yang benar-benar dibutuhkan untuk analisis sentimen dan efisiensi pemrosesan data, dengan membuang kolom-kolom metadata yang tidak relevan.

b) Pelabelan Data

Tahap pelabelan data merupakan bagian dari pra-pemrosesan yang bertujuan untuk mengkonversi skor bintang (rating) dari *Google Play Store* menjadi kategori sentimen yang diskrit. Metode ini disebut pelabelan berdasarkan skor (*Score Based Labeling*).

Proses pelabelan dilakukan dengan mendefinisikan sebuah fungsi khusus yang memetakan skor ulasan dari 1 hingga 5 ke dalam tiga kelas sentimen. Tiga kelas sentimen adalah Positif, Negatif, dan Netral. Aturan pemetaan yang diterapkan pada setiap ulasan dapat dilihat pada Tabel 1.2. Kategori *Label Sentimen*.

Tabel 1.2. Kategori *Label Sentimen*

Batas Nilai	Kategori <i>Label Sentimen</i>
1 atau 2	Negatif
3	Netral
4 atau 5	Positif

c) *Cleaning*

Pada tahap ini, data dibersihkan dengan menghilangkan elemen yang tidak relevan, seperti kolom, tanda baca dan spasi kosong, yang tidak memberikan kontribusi terhadap klasifikasi. Berikut adalah tahapan dari proses *cleaning*:

1) Pemeriksaan Data Hilang

Pemeriksaan Data Hilang / *Missing Value Check* adalah teknik yang dilakukan untuk mengidentifikasi keberadaan nilai-nilai yang kosong atau tidak terdefinisi (*null* atau *NaN*) dalam *DataFrame*.

2) Penghapusan Data Hilang

Teknik yang digunakan pada tahap ini adalah *Eksklusi Listwise (Listwise Deletion)*. Teknik ini memilih untuk menghapus seluruh baris data yang memiliki nilai kosong pada kolom yang ditentukan.

Proses ini secara spesifik menargetkan kolom *Label* yang berisi kategori sentimen (*Label Content*) dan menghapus semua baris ulasan yang gagal mendapatkan *Label (missing Label)*. Penghapusan ini dilakukan secara permanen pada *DataFrame* untuk menjamin bahwa seluruh data yang tersisa memiliki nilai sentimen yang valid dan siap untuk diolah.

3) Verifikasi Data Hilang

Setelah data yang hilang dihapus, dilakukan langkah verifikasi untuk memastikan integritas data terjaga.

4) Menghapus karakter Non-Alfabet dan Simbol

Langkah ini melakukan pembersihan terhadap gangguan (*noise*) khas data daring di kolom *content*. Dengan menggunakan pola *Regex* untuk mencari dan menghapus elemen-elemen seperti karakter @, http, https, dan tanda baca.

5) Menghapus Angka

Selanjutnya menghapus angka pada kolom *content* dengan *RegEx* untuk mengurangi *noise* dan meningkatkan efektivitas model ML.

6) Membuat kolom *text_clean*

Hasil pembersihan dari kolom *content* disimpan dalam kolom baru yang diberi nama kolom *text_clean*. Proses atau langkah selanjutnya mengolah data pada kolom *text_clean*.

d) *Case Folding*

Proses *Case Folding* adalah tahap kunci dalam *Text Processing* yang bertujuan untuk menciptakan konsistensi dalam seluruh dataset ulasan. Tanpa *Case Folding*, sebuah kata dapat dihitung berkali-kali hanya karena perbedaan kapitalisasi (misalnya, "SAYA" dan "saya").

Teknik yang digunakan adalah Konversi *String Penuh (Full String Conversion)* yang mengaplikasikan metode bawaan *Python/Pandas*. Metode bawaan *Python/Pandas* adalah *.str.lower()*.

e) *Stopword Removal*

Kata-kata yang umum dan tidak relevan, seperti "dan", "atau", "adalah", dihapus dari data. Hal ini dilakukan agar fokus analisis hanya pada kata-kata yang memiliki makna penting.

Teknik ini mengandalkan *library Natural Language Toolkit (NLTK)* dan kamus (corpus) *stopwords* Bahasa Indonesia.

- a. *Python* mengunduh *library stopwords* yang memuat daftar kata henti baku berbahasa Indonesia dari NLTK.

Daftar ini berisi kata-kata umum seperti "yang," "dan," "adalah," dan "di," yang frekuensinya tinggi namun tidak menentukan sentimen Positif, Negatif, atau Netral.

- b. Teks yang sudah bersih pada kolom *text_clean* kemudian diolah menggunakan fungsi *List Comprehension* dan *Filtering* untuk memecah ulasan menjadi kata-kata / *token* menggunakan metode *.split()* pada *Python*.
- c. Setiap kata dalam ulasan pada kolom *text_clean* diperiksa dan hanya kata-kata yang tidak ditemukan dalam daftar *stopword* yang dipertahankan dan digabungkan kembali menjadi sebuah kalimat.
- d. Selanjutnya ulasan yang telah melalui proses *Stopword Removal* akan disimpan kedalam kolom baru yang diberi nama kolom *text_StopWord*.

f) *Tokenisasi*

Tokenisasi atau *Tokenizing* adalah proses fundamental yang memecah suatu *string* teks menjadi unit-unit yang lebih kecil dan bermakna yang disebut *token* (biasanya adalah kata-kata individual). Tujuannya adalah untuk mengubah kalimat panjang menjadi daftar kata-kata, yang merupakan format input yang diperlukan untuk proses pembobotan fitur (seperti TF-IDF) dan pelatihan model ML.

Kalimat dipecah menjadi kata-kata individual yang disebut *token*. Proses ini memisahkan kata-kata dalam teks berdasarkan spasi, sehingga setiap kata dapat dianalisis secara terpisah.

Teknik yang digunakan adalah *Word Tokenization* yang memanfaatkan fungsi *word_tokenize()* dari *library* NLTK.

- a. Kode *Python* menerapkan fungsi *word_tokenize()* pada kolom ulasan yang sudah bersih. Kolom ulasan yang sudah bersih adalah kolom *text_StopWord*. Setiap baris teks pada kolom *text_StopWord* dipecah menjadi list kata-kata.
- b. Hasil dari proses *Tokenisasi* disimpan dalam kolom baru yang diberi nama kolom *text_tokens*. Kolom *text_tokens* menyimpan ulasan sebagai serangkaian daftar kata-kata. Contohnya, ulasan "aplikasi bagus sekali" diubah menjadi ['aplikasi', 'bagus', 'sekali'].

g) *Stemming*

Tahap *Stemming* adalah langkah akhir dalam *Text Processing* yang bertujuan untuk mengubah kata-kata berimbuhan (seperti prefiks, sufiks, atau infiks) menjadi bentuk kata dasar atau akarnya. Tujuannya adalah untuk mengurangi redudansi fitur, memastikan bahwa variasi kata yang memiliki arti sentimen serupa misalnya, "menonton," "ditonton," "tontonan" diperlakukan sebagai satu entitas tunggal "tonton" oleh model.

Kata-kata dalam teks dipotong ke bentuk dasarnya, seperti mengubah "bermain" menjadi "main". Ini membantu mengurangi variasi kata yang memiliki akar yang sama, sehingga data lebih konsisten dan memperkuat hasil analisis sentimen.

Teknik yang digunakan adalah *Stemming* Bahasa Indonesia, diimplementasikan oleh *library Sastrawi*

- a. *Python* menginisiasi objek *Stemmer* dari *Sastrawi* yang bertugas memproses kata-kata.
- b. Kemudian mengiterasi seluruh *token* unik yang ada di kolom *text_tokens* dan menyimpannya di dalam kamus.
- c. Setelah semua kata unik terkumpul, *Python* memproses setiap kata unik tersebut sekali saja menggunakan fungsi *stemmer Sastrawi* dan menyimpan hasil kata dasarnya kembali ke kamus.
- d. Terakhir, fungsi *get_stemmed_term* digunakan untuk memetakan kembali setiap *token* dalam ulasan dengan kata dasar yang sudah tersimpan di dalam kamus. Hasil akhirnya digabungkan kembali menjadi sebuah kalimat dan disimpan dalam kolom baru bernama *text_stemindo*.

1.5.3 Transformasi

Pada tahap transformasi data, *CountVectorizer* digunakan untuk mengonversi teks menjadi vektor numerik berdasarkan frekuensi kata. Selanjutnya, metode TF-IDF diterapkan untuk memberikan bobot lebih besar pada kata-kata penting dalam ulasan tertentu dan bobot lebih kecil pada kata-kata umum.

1.5.4 Klasifikasi

Proses klasifikasi melibatkan pengolahan data dengan menggunakan algoritma untuk mencapai tujuan tertentu. Dalam penelitian ini, tujuan utamanya adalah mengelompokkan data dengan algoritma *Naïve Bayes* dan SVM. Pada masing-masing Algoritma ini akan membagi data menjadi dua bagian: data untuk melatih model (*training*) dan data untuk menguji hasilnya (*testing*).

Pada penelitian ini akan dilakukan tiga kali skenario pengujian pada masing-masing algoritma dengan perbedaan presentase pada data training dan data testing seperti pada Tabel 1.3.

Pengujian Klasifikasi untuk mendapatkan model data pengujian terbaik.

Tabel 1.3. Skenario Pengujian Klasifikasi

Skenario	Data Training	Data Testing
1	70%	30%
2	80%	20%
3	90%	10%

1.5.5 Evaluasi Model

Evaluasi melibatkan mengevaluasi hasil analisis sentimen untuk memastikan akurasi dan reliabilitas. Langkah ini melibatkan penggunaan metode evaluasi yang sesuai untuk menentukan seberapa baik algoritma *Naïve Bayes* dan SVM dalam mengklasifikasikan sentimen ulasan aplikasi OTT untuk masing-masing skenario pengujian. Dengan demikian, dapat diketahui seberapa efektif algoritma dalam mengidentifikasi sentimen positif dan negatif dalam ulasan, serta seberapa baik algoritma dalam memprediksi sentimen ulasan yang belum pernah dilihat sebelumnya.

1.5.6 Analisis Korelasi

Setelah proses klasifikasi sentimen selesai dan persentase ulasan positif untuk setiap aplikasi telah diperoleh, langkah selanjutnya adalah menganalisis korelasi antara sentimen tersebut dengan *rating* aplikasi. Analisis ini bertujuan untuk membuktikan hipotesis bahwa sentimen ulasan memiliki hubungan signifikan dengan tingkat popularitas aplikasi di pasar.

Metode yang digunakan untuk mengukur korelasi ini adalah Koefisien Korelasi *Pearson*. Koefisien ini dipilih karena

kemampuannya dalam mengukur kekuatan dan arah hubungan linier antara dua variabel kuantitatif. Variabel yang akan diukur adalah persentase sentimen positif (sebagai variabel X) dan *rating* Aplikasi di *Google Play Store* (sebagai variabel Y) dari kelima aplikasi OTT yang diteliti. Aplikasi OTT yang diteliti adalah Netflix, Vidio, Disney+ Hotstar, Prime Video, dan Catchplay+.

Langkah-langkah yang dilakukan adalah sebagai berikut:

a) Pengumpulan Data:

Data persentase sentimen positif diambil dari hasil klasifikasi yang telah dilakukan pada tahap sebelumnya. Sementara itu, *rating* aplikasi dicatat langsung dari halaman resmi setiap aplikasi di *Google Play Store*.

b) Perhitungan Koefisien *Pearson*:

Data yang telah terkumpul akan dihitung menggunakan rumus matematis Koefisien Korelasi *Pearson* (r). Hasil perhitungan akan menghasilkan nilai antara -1 hingga +1. Nilai yang mendekati +1 menunjukkan adanya korelasi positif yang kuat, artinya semakin tinggi persentase sentimen positif, semakin tinggi pula *rating* aplikasi. Sebaliknya, nilai yang mendekati -1 menunjukkan korelasi negatif, dan nilai yang mendekati 0 menunjukkan tidak adanya hubungan linier.

c) Interpretasi Hasil:

Hasil dari perhitungan koefisien korelasi akan diinterpretasikan untuk menarik kesimpulan. Analisis akan berfokus pada seberapa kuat hubungan antara sentimen pengguna dan popularitas aplikasi. Hasil ini akan memberikan wawasan yang lebih mendalam mengenai pengaruh *word of mouth* digital terhadap keputusan pengguna dalam mengunduh sebuah aplikasi. Temuan ini diharapkan dapat memberikan masukan praktis bagi

penyedia layanan OTT dalam memahami pentingnya menjaga sentimen positif di kalangan penggunanya.

1.6.Sistematika Penulisan

Untuk memberikan gambaran yang jelas mengenai alur penyusunan Tugas Akhir ini, laporan disusun dalam beberapa bab dengan sistematika sebagai berikut:

1. BAB I PENDAHULUAN

Bab ini berisi uraian mengenai Latar Belakang Masalah, Perumusan Masalah, Batasan Masalah, Tujuan dan Manfaat Penelitian, Metodologi Penelitian, serta Sistematika Penulisan sebagai gambaran umum isi laporan.

2. BAB II LANDASAN TEORI

Bab ini memuat teori-teori yang relevan sebagai dasar penyusunan penelitian, meliputi tinjauan pustaka, kerangka pemikiran, serta hipotesis. Seluruh teori yang disajikan merupakan teori pendukung bagi pelaksanaan dan pembahasan Penelitian Tugas Akhir.

3. BAB III ANALISIS DAN PERANCANGAN SISTEM

Bab ini menjelaskan analisis sistem yang meliputi sistem berjalan dan sistem yang diusulkan, serta perancangan sistem yang mencakup perancangan proses bisnis, struktur data, diagram perancangan, dan rancangan antarmuka.

4. BAB IV IMPLEMENTASI DAN ANALISIS HASIL

Bab ini berisi implementasi sistem berdasarkan rancangan yang telah dibuat, pengujian terhadap sistem, serta analisis hasil pengujian untuk menilai keberhasilan dan kesesuaian sistem dengan kebutuhan.

5. BAB V SIMPULAN DAN SARAN

Bab ini memuat simpulan penelitian yang diperoleh dari hasil pengembangan sistem, serta saran untuk penyempurnaan dan pengembangan lebih lanjut di masa mendatang.

BAB II LANDASAN TEORI

2.1 Tinjauan Pustaka

2.1.1. Klasifikasi

Klasifikasi merupakan teknik yang paling umum diterapkan pada *data mining*. Proses klasifikasi data melibatkan proses *learning* dan *training*. Dalam belajar (*learning*) data pelatihan (*training*) data dianalisis dengan algoritma klasifikasi. Dalam proses klasifikasi pengujian data dilakukan dengan menggunakan perkiraan akurasi dari aturan klasifikasi (Rosyani, Saprudin and Amalia, 2021).

2.1.2. Analisis Sentimen

Analisis sentimen adalah metode yang digunakan untuk mengidentifikasi dan memahami perasaan atau pendapat yang terdapat dalam teks. Ulasan pengguna merupakan sumber informasi yang penting karena dapat memberikan pemahaman tentang tingkat kepuasan atau ketidakpuasan pengguna terhadap suatu aplikasi (Randy Suryono, 2024).

2.1.3. *Google Play Store*

Google memiliki sebuah layanan yang dikenal dengan *Play Store* yang menyediakan konten digital seperti *game*, aplikasi, *film*, musik, dan buku dengan kategori yang beragam. Salah satu fitur yang terdapat pada *Play Store* adalah fitur rating dan ulasan dimana pengguna produk dari *Play Store* dapat memberikan opini mereka terhadap produk yang telah mereka gunakan (Nurian, 2023).

2.1.4. *Data Mining*

Data mining adalah suatu kegiatan analisa data untuk mencari suatu pola tertentu, dengan jumlah data yang besar dan bertujuan untuk menghasilkan informasi yang dapat digunakan dan

dikembangkan lebih lanjut. *Data mining* merupakan inti dari proses *Knowledge Discovery Databases (KDD)*, meliputi dugaan algoritma yang mengeksplor data, membangun model dan menemukan pola yang belum diketahui. KDD merupakan penyelesaian masalah dengan menganalisa data yang ada pada *database* dengan data tersimpan secara elektronik dan pencariannya dilakukan otomatis seperti pada Komputer (Suliman, 2021).

2.1.5. *Layanan Over The Top (OTT)*

Layanan OTT adalah Layanan konten berupa data, informasi, atau multimedia yang berjalan melalui jaringan internet. Beberapa penyedia layanan OTT juga memiliki konten yang hanya diproduksi dan didistribusi pada *platform* mereka sendiri, atau disebut juga konten original. Konten yang orisinal disebut memiliki nilai jual tersendiri yang berpengaruh pada loyalitas pelanggan OTT. Pelanggan melihat kemampuan sistem pada konten OTT terkait keandalan, daya tahan, akurasi, dan kemudahan dalam penggunaan, atau dalam bahasa lainnya kualitas konten. Konsistensi dalam memberikan konten yang berkualitas tinggi pada pelanggan dapat membantu meningkatkan keunggulan kompetitif (Sugiarto and Syech Abu Bakar, 2023).

2.1.6. *Naïve Bayes*

Naïve Bayes merupakan salah satu algoritma *data mining* untuk mengklasifikasi suatu objek menggunakan probabilitas dan statistik, serta dapat memprediksi peluang berdasarkan dari pengalaman sebelumnya. Ciri dari algoritma *Naïve Bayes* ini adalah independensi yang sangat kuat (*naïve*) dari masing-masing kondisi atau kejadian. Algoritma ini termasuk dalam *supervised learning* karena membutuhkan *data training* sebelum proses klasifikasi (Nugraha and Gustian, 2024). Secara umum rumus dasar persamaan teorema *Bayes* disajikan pada Persamaan (1).

$$P(H|X) = \frac{P(H) \times P(X|H)}{P(X)} \quad (1)$$

Sumber: (Nugraha and Gustian, 2024)

Keterangan :

X : Data dengan kelas yang belum diketahui

H : Hipotesis X merupakan suatu kelas spesifik

P(H|X) : Probabilitas H berdasarkan kondisi X

P(H) : Probabilitas pada hipotesis H (prior)

P(X|H) : Probabilitas X berdasarkan kondisi H

P(X) : Probabilitas X (data sampel yang diamati)

2.1.7. *Support Vector Machines (SVM)*

Support Vector Machines adalah suatu metode / algoritma yang handal dalam menyelesaikan masalah klasifikasi data. Permasalahan SVM dipecahkan dengan menyelesaikan persamaan *Lagrangian* yang merupakan bentuk dual dari SVM melalui *quadratic programming*. SVM mencari grid untuk menemukan kombinasi *hyperparameter* terbaik seperti C (parameter toleransi kesalahan), tipe kernel, dan gamma, SVM digunakan untuk membuat garis pemisah (*hyperplane*) antara dua kelas atau lebih. Proses klasifikasi berdasarkan model pembelajaran SVM dievaluasi menggunakan matriks konfusi. Hal ini memberikan gambaran komprehensif tentang performa model saat memisahkan kelas yang berbeda (Maulana *et al.*, 2024). Dalam SVM, rumus untuk memprediksi kelas dari suatu sampel x menggunakan Persamaan (2).

$$f(x) = \text{sign}(w * x + b) \quad (2)$$

Sumber:(Randy Suryono, 2024)

Keterangan :

$f(x)$: Fungsi keputusan yang mengklasifikasikan sampel x

w : Vektor bobot

x : Vektor fitur input

b : Bias

$\text{sign}(-)$: Fungsi tanda yang menghasilkan *Label* kelas

2.1.8. Evaluasi Model

Proses evaluasi model merupakan langkah kritis dalam menilai kinerja model, yang dilakukan dengan mengukur performanya menggunakan dataset uji. Dataset uji merupakan sekumpulan data yang sebelumnya tidak diperkenalkan atau digunakan dalam tahap pelatihan dan validasi model. Evaluasi ini memberikan gambaran objektif tentang kemampuan model dalam menghadapi situasi dunia nyata dan mengidentifikasi sejauh mana model dapat diandalkan dalam tugas yang diberikan.

Pada tahap evaluasi digunakan *metrics* evaluasi untuk mengevaluasi kinerja modeling dari sistem rekomendasi. *Metrics* evaluasi merupakan parameter yang digunakan untuk menilai kualitas model atau algoritma pembelajaran mesin. Beberapa metrik evaluasi telah diterapkan diantaranya *Accuracy*, *Precision*, *Recall*, dan *F1-Score*. Masing-masing metrik evaluasi ini memiliki rumusan khusus yang digunakan untuk mengukur berbagai aspek performa model (Sugandi and Sarmini, 2024). Rumus perhitungan akurasi, presisi, *recall*, dan *F1-Score* ditunjukkan pada Persamaan (3)-Persamaan (6).

a. *Accuracy* (Akurasi)

$$Accuracy = \frac{TP+TN}{(TP+FN)+(FP+TN)} \quad (3)$$

b. *Precision* (Presisi)

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

c. *Recall* (*Recall* atau Sensivitas)

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

d. *F1-Score*

$$F1 - Score = \frac{2TP}{2TP+FN+FP} \quad (6)$$

Sumber :(Sugandi and Sarmini, 2024):

Keterangan :

True Positive (TP) : Jumlah ulasan yang benar-benar Positif dan diprediksi Positif oleh model

True Negative (TN) : Jumlah ulasan yang benar-benar Negatif dan diprediksi Negatif oleh model

False Positif (FP) : Jumlah ulasan yang sebenarnya Negatif tetapi salah diprediksi Positif (Model terlalu optimis)

False Negative (FN) : Jumlah ulasan yang sebenarnya Positif tetapi salah diprediksi Negatif (Model kehilangan ulasan baik)

2.1.9. Analisa Korelasi *Pearson*

Analisis korelasi *Pearson* disebut juga dengan korelasi *product moment* adalah analisis untuk mengukur keeratan hubungan secara linier antara dua variable yang mempunyai distribusi normal. Menganalisis data tentang hubungan penggunaan media gambar, dan

untuk menentukan apakah variable X dengan variable Y terdapat hubungan yang signifikan, maka Persamaan (7) menggambarkan rumus korelasi *product moment* (Sari, Hadiati and Sihotang, 2023):

$$r_{xy} = \frac{N \cdot \sum XY - (\sum X) \cdot (\sum Y)}{\sqrt{\{(N \cdot \sum X^2 - (\sum X)^2)(N \cdot \sum Y^2 - (\sum Y)^2)\}}}$$

Sumber : (Sari, Hadiati and Sihotang, 2023) (7)

Keterangan:

r_{xy} : Angka indeks korelasi antara variabel X dengan variabel Y

N : Jumlah Sampel

$\sum X^2$: Jumlah kuadrat variabel X

$\sum Y^2$: Jumlah kuadrat variabel Y

$\sum XY$: Jumlah hasil perkalian antara skor X dan skor Y

$\sum X$: Jumlah variabel X

$\sum Y$: Jumlah variabel Y

2.1.10. Bahasa Pemrograman *Python*

Python adalah bahasa pemrograman tingkat tinggi yang terkenal karena sintaksisnya yang sederhana, mudah dipahami, dan fleksibel. Dikembangkan oleh Guido van Rossum, *Python* memiliki berbagai kegunaan, termasuk pengembangan perangkat lunak, analisis data, kecerdasan buatan, pengembangan web, dan lebih banyak lagi. Bahasa pemrograman ini menjadi pilihan utama bagi banyak pengembang karena kombinasi kejelasan sintaksis dan kemampuan untuk menangani berbagai tugas pemrograman (Wilyani, Arif and Aslimar, 2024).

2.1.11. *Machine Learning*

Mesin Pembelajaran / ML adalah cabang dari *Artificial Intelligence (AI)* yang fokus belajar dari data. ML fokus pada pengembangan sistem yang mampu belajar secara “mandiri” tanpa harus berulang kali diprogram manusia.

Mesin Pembelajaran terbagi menjadi beberapa algoritma yang berbeda-beda dan setiap dari algoritma tersebut memiliki fungsi dan tujuannya masing-masing. Beberapa contoh dari algoritma ML yang paling populer antara lain adalah *Supervise Learning, Unsupervised Learning, Semi-supervised Learning, & Reinforcement Learning* (Alfarizi *et al.*, 2023).

2.2 Kerangka Pemikiran

Layanan *Over The Top (OTT)* media service telah menjadi bagian tak terpisahkan dari kehidupan sehari-hari, dan platform seperti Netflix, Vidio, Disney+ Hotstar, Prime Video, dan Catchplay+ bersaing untuk menarik dan mempertahankan pengguna. Salah satu sumber data yang autentik untuk memahami kepuasan pengguna adalah ulasan di *Google Play Store*. Analisis ulasan dalam jumlah besar secara manual tidak efisien.

Oleh karena itu, penelitian ini bertujuan untuk mengklasifikasi sentimen ulasan menggunakan metode ML. Penelitian ini akan menerapkan algoritma *Naive Bayes* dan *Support Vector Machine (SVM)* untuk mengategorikan ulasan menjadi sentimen positif, negatif, atau netral. Setelah klasifikasi, proses selanjutnya dilakukan evaluasi untuk membandingkan kinerja kedua algoritma.

Selanjutnya, penelitian ini menganalisis korelasi antara persentase sentimen positif dari ulasan dengan *rating* aplikasi. Analisis ini menggunakan Koefisien Korelasi *Pearson* untuk mengukur hubungan linier antara kedua variabel. Hasilnya diharapkan dapat membuktikan hipotesis bahwa sentimen ulasan memiliki hubungan signifikan dengan tingkat popularitas aplikasi di pasar.

Dengan demikian, kerangka pemikiran ini menunjukkan alur logis dari masalah, pendekatan solusi, hingga kontribusi yang diharapkan dari penelitian ini.

2.3 Hipotesis

Penelitian ini menguji dugaan mengenai perbandingan kinerja antara dua algoritma klasifikasi. Berdasarkan sifat data ulasan teks yang memiliki frekuensi kata yang terdistribusi secara independen, dapat diduga bahwa algoritma *Naïve Bayes* lebih efektif dibandingkan dengan *Support Vector Machine*. Hipotesis penelitian H_0 (Hipotesis Nol) dan H_1 (Hipotesis Alternatif) dan dirumuskan sebagai berikut:

a. H_0 (Hipotesis Nol):

Tidak terdapat perbedaan yang signifikan dalam tingkat akurasi antara Algoritma *Naive Bayes* dan Algoritma *Support Vector Machine* dalam klasifikasi sentimen ulasan aplikasi OTT.

b. Hipotesis Alternatif (H_1):

Tingkat akurasi Algoritma *Naive Bayes* lebih tinggi secara signifikan dibandingkan dengan Algoritma *Support Vector Machine* dalam klasifikasi sentimen ulasan aplikasi OTT.