RISET INFORMATIKA

PROPOSAL PENELITIAN

“Identifikasi Marka Linguistik Pembeda Teks Generatif AI dan Teks Manusia Menggunakan SHAP (SHapley Additive exPlanations)”

Dosen Pendamping:

Dr. Basuki Rahmat, S.Si. MT

Disusun Oleh:

Muhammad Hidayat Nurwahid (22081010300)

Universitas Pembangunan Nasional “Veteran” Jawa Timur

Surabaya

2025

BAB I

PENDAHULUAN

1.1.Latar Belakang

Kehadiran model bahasa generatif seperti model GPT dan Gemini telah mengubah pendekatan dalam pembuatan konten secara fundamental (Saha, 2025). Teknologi ini mampu menghasilkan teks dengan kualitas yang mirip dengan tulisan manusia (Eapen dkk., 2023). Sehingga, mengaburkan batas antara karya orisinal dan hasil rekayasa mesin. Kemampuan ini, meski inovatif, membawa implikasi serius yang menuntut perhatian.

Implikasinya terasa di berbagai sektor krusial. Dalam jurnalisme, risiko penyebaran misinformasi semakin tinggi. Di sektor komersial, rekayasa ulasan produk dapat merusak kepercayaan konsumen. Namun, tantangan terbesarnya mungkin terjadi di dunia akademik, di mana integritas karya ilmiah terancam oleh praktik plagiarisme yang semakin canggih dan sulit dilacak (Eke, 2023).

Sebagai respons, berbagai alat deteksi konten AI mulai bermunculan. Sayangnya, mayoritas alat ini beroperasi layaknya "kotak hitam" (*black box*). Alat-alat tersebut hanya memberikan vonis akhir "AI" atau "Manusia" tanpa disertai penjelasan yang dapat diverifikasi. Pendekatan ini melahirkan dilema baru: ketika seorang mahasiswa dituduh menggunakan AI, tidak ada bukti konkret yang bisa disajikan selain label dari sebuah sistem yang tidak transparan. Ketiadaan akuntabilitas ini membuat alat deteksi konvensional tidak dapat diandalkan dalam pengambilan keputusan yang adil.

Di sinilah letak urgensi penelitian ini. Masalah mendasarnya bukanlah ketiadaan detektor, melainkan ketiadaan penjelasan. Kita perlu beralih dari pertanyaan apakah sebuah teks ditulis oleh AI, ke pertanyaan mengapa sebuah teks dianggap demikian. Untuk itu, penelitian ini akan memanfaatkan *Explainable AI* (XAI) melalui metode SHAP (*SHapley Additive exPlanations*). Tujuannya adalah untuk "membongkar" kotak hitam tersebut, mengidentifikasi "sidik jari" atau marka linguistik yang secara konsisten menjadi pembeda antara teks AI dan tulisan manusia, dan pada akhirnya, menciptakan dasar untuk sistem deteksi yang transparan dan dapat dipertanggungjawabkan.

1.2.Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, penelitian ini berupaya menjawab pertanyaan-pertanyaan berikut:

1. Marka linguistik apa saja (meliputi aspek leksikal, sintaksis, dan stilistik) yang menjadi pembeda paling signifikan antara teks buatan *Generative AI* dan teks tulisan manusia?
2. Bagaimana metode SHAP dapat digunakan untuk mengidentifikasi dan memvisualisasikan kontribusi setiap marka linguistik terhadap keputusan model dalam mengklasifikasikan sebuah teks?
3. Sejauh mana konsistensi marka linguistik tersebut dapat diandalkan pada genre teks yang berbeda, seperti teks berita dan esai kreatif?

1.3.Tujuan

Sejalan dengan rumusan masalah, tujuan yang ingin dicapai dalam penelitian ini adalah:

1. Membangun sebuah model *machine learning* yang mampu mengklasifikasikan teks generatif AI dan teks manusia dengan performa terukur.
2. Mengidentifikasi serta mengurutkan marka-marka linguistik paling berpengaruh yang menjadi dasar prediksi model klasifikasi.
3. Menyajikan hasil analisis SHAP dalam bentuk visualisasi yang intuitif untuk menjelaskan cara kerja detektor secara transparan.

1.4.Batasan Masalah

Agar penelitian ini dapat berjalan fokus dan mendalam, ruang lingkupnya dibatasi pada beberapa aspek berikut:

1. **Bahasa dan Generalisasi Temuan:** Penelitian ini secara spesifik menggunakan *dataset* dalam Bahasa Inggris. Konsekuensinya, marka linguistik yang nantinya ditemukan sebagai pembeda tidak dapat digeneralisasi atau diterapkan secara langsung pada teks berbahasa Indonesia. Hal ini disebabkan oleh adanya perbedaan fundamental dalam struktur sintaksis, morfologi, dan stilistika antara kedua bahasa tersebut.
2. **Metode Penjelasan:** Metode *Explainable AI* (XAI) yang digunakan secara spesifik adalah SHAP (*SHapley Additive exPlanations*). Penelitian ini tidak akan melakukan perbandingan dengan metode XAI lainnya seperti LIME atau Anchors.
3. **Cakupan Fitur:** Analisis dibatasi pada fitur-fitur linguistik komputasional yang telah ditentukan sebelumnya, yang mencakup variabel leksikal (kosakata), sintaksis (struktur kalimat), dan stilistik (gaya bahasa).

BAB II

TINJAUAN PUSTAKA

2.1.Penelitian Terdahulu

Penelitian yang dilakukan oleh Ganesh Jawahar, Muhammad Abdul-Mageed, dan Laks V.S. Lakshmanan (2020) dengan judul "*Automatic Detection of Machine Generated Text: A Critical Survey*" menjelaskan bahwa upaya membedakan teks buatan mesin dari tulisan manusia pada dasarnya dibingkai sebagai sebuah tugas klasifikasi. Kajian ini mengidentifikasi bahwa pendekatan paling canggih saat ini adalah dengan melakukan *fine-tuning* pada model bahasa seperti RoBERTa , yang terbukti mampu mencapai akurasi deteksi yang sangat tinggi, yaitu sekitar 95% dalam mengidentifikasi teks yang dihasilkan oleh model GPT-2. Namun, penelitian tersebut juga menyoroti kelemahan fundamental dari detektor-detektor ini: mereka seringkali beroperasi sebagai "kotak hitam" yang tidak transparan. Secara spesifik, ditemukan bahwa detektor otomatis cenderung lemah dalam mengenali kesalahan semantik atau kontradiksi dalam teks, suatu aspek yang justru lebih mudah diidentifikasi oleh manusia. Temuan ini menggarisbawahi adanya kebutuhan mendesak untuk mengembangkan detektor yang tidak hanya akurat, tetapi juga dapat diinterpretasikan (*interpretable*), sehingga dapat memberikan penjelasan yang dapat dipahami manusia di balik setiap keputusannya (Jawahar dkk., 2020).

Penelitian yang dilakukan oleh Mahdi Dhaini, Wessel Poelman, dan Ege Erdogan (2023) dengan judul "*Detecting ChatGPT: A Survey of the State of Detecting ChatGPT-Generated Text*" menjelaskan bahwa deteksi teks buatan ChatGPT merupakan tugas krusial yang umumnya didekati sebagai masalah klasifikasi biner untuk membedakan antara tulisan manusia dan mesin. Kajian ini mengidentifikasi bahwa berbagai metode, mulai dari klasifikasi berbasis *perplexity* hingga model *transformer* yang di-*fine-tune* seperti DistilBERT, telah dikembangkan untuk tugas ini. Namun, penelitian tersebut juga menyoroti bahwa sekadar menghasilkan label klasifikasi tidaklah cukup, dan ada kebutuhan mendesak untuk memahami bagaimana model deteksi mengambil keputusannya. Secara spesifik, ditemukan bahwa beberapa pendekatan mulai mengintegrasikan teknik *Explainable AI* seperti SHAP untuk memberikan penjelasan lokal dan skor kepentingan fitur, yang dapat mengungkap kontribusi kata-kata tertentu terhadap prediksi model. Temuan ini menggarisbawahi bahwa pemanfaatan teknik *explainability* sangat berharga, tidak hanya untuk men-*debug* detektor, tetapi juga untuk

memfasilitasi analisis mendalam mengenai perbedaan gaya penulisan antara manusia dan ChatGPT (Dhaini dkk., 2023).

Penelitian yang dilakukan oleh Nuzhat Noor dan Islam Prova (2024) dengan judul "*Detecting AI Generated Text Based on NLP and Machine Learning Approaches*" menjelaskan bahwa pendeteksian teks buatan AI merupakan sebuah tugas klasifikasi yang krusial untuk mengatasi berbagai implikasi etis, hukum, dan sosial. Untuk membedakan antara tulisan manusia dan teks yang dihasilkan secara terprogram, penelitian ini menerapkan beberapa pendekatan *machine learning*, yaitu *XGB Classifier*, SVM, dan model *deep learning* berbasis arsitektur *Transformer*, BERT. Hasil pengujian menunjukkan bahwa model BERT memberikan performa paling unggul dengan akurasi tertinggi mencapai 93%, melampaui *XGB Classifier* (84%) dan SVM (81%). Keunggulan signifikan BERT ini disebabkan oleh kemampuannya yang lebih baik dalam memahami konteks kata secara dua arah dan menangkap nuansa serta pola-pola halus dalam data teks yang menjadi penanda khas dari konten buatan AI. Temuan ini memperkuat argumen bahwa arsitektur canggih seperti BERT merupakan solusi yang paling menjanjikan untuk mengembangkan detektor AI yang akurat dan andal (Noor & Prova, 2024).

Penelitian yang dilakukan oleh Anjana Priyatham Tatavarthi, Faranak Abri, dan Nada Attar (2025) dengan judul "*AI-Generated Text Detection and Source Identification*" menjelaskan bahwa deteksi teks buatan AI dapat dipecah menjadi dua tugas praktis: klasifikasi biner (manusia vs. AI) dan identifikasi sumber model AI (multikelas). Dalam penelitian ini, berbagai model *deep learning* dievaluasi, termasuk LSTM dan model *transformer* seperti BERT dan RoBERTa, dengan menggunakan teknik *embedding* yang berbeda. Hasilnya menunjukkan bahwa untuk tugas klasifikasi biner, model LSTM yang dikombinasikan dengan *embedding* BERT mencapai performa terbaik dengan akurasi dan skor F1 sebesar 97%. Sementara itu, untuk tugas identifikasi sumber multikelas yang lebih kompleks, model berbasis *transformer* terbukti jauh lebih unggul daripada model RNN, dengan RoBERTa mencapai akurasi dan skor F1 tertinggi sebesar 88%. Temuan ini menggarisbawahi keunggulan arsitektur *transformer* yang telah dilatih sebelumnya (*pretrained*) dalam menangkap pola linguistik yang halus untuk membedakan antar sumber AI yang berbeda (Tatavarthi dkk., 2025).

2.2.Kecerdasan Buatan Generatif

Kecerdasan Buatan Generatif (*Generative Artificial Intelligence* atau GAI) adalah cabang kecerdasan buatan yang secara fundamental berfokus pada penciptaan konten baru dan orisinal seperti teks, gambar, atau kode, alih-alih hanya menganalisis data yang sudah ada (Corchado

dkk., 2023). Perbedaan utamanya dengan AI diskriminatif terletak pada tujuannya; jika AI diskriminatif bertujuan untuk klasifikasi atau prediksi, GAI belajar dari distribusi probabilitas data dalam jumlah besar untuk menghasilkan sampel data baru yang serupa namun unik (Banh & Strobel, 2023). Proses ini biasanya dipicu oleh *prompt* atau instruksi dari pengguna, yang kemudian ditafsirkan oleh model untuk menghasilkan keluaran yang diinginkan (Kalota, 2024).

Kemampuan GAI didukung oleh kemajuan *Deep Learning*, terutama melalui arsitektur *Deep Generative Models* (DGM) seperti *Generative Adversarial Networks* (GANs), *Variational Autoencoders* (VAEs), dan arsitektur *Transformer* yang menjadi dasar bagi *Large Language Models* (LLM) (Noor & Prova, t.t.). Teks yang dihasilkan oleh LLM memiliki ciri khas: cenderung sangat terorganisir dan formal dengan sedikit kesalahan gramatikal, namun di sisi lain memiliki keragaman leksikal yang lebih rendah dan sering kali kurang spesifik secara emosional (Dhaini dkk., 2023). Model AI juga terbukti lemah dalam mendeteksi kesalahan semantik atau kontradiksi internal dalam teks yang dihasilkannya (Tatavarthi dkk., 2025).

Meskipun memiliki potensi besar, GAI dihadapkan pada tantangan signifikan, termasuk kecenderungan untuk mereplikasi bias dari data pelatihannya, menghasilkan informasi yang salah secara faktual atau halusinasi, serta potensi penyalahgunaan untuk menyebarkan misinformasi (Lv, 2023). Salah satu tantangan terbesar adalah sifatnya sebagai "kotak hitam" (*black box*), di mana proses internal pengambilan keputusannya tidak transparan. Kurangnya transparansi ini mendorong kebutuhan mendesak akan *Explainable AI* (XAI) untuk membangun kepercayaan dan memastikan penggunaan teknologi ini secara bertanggung jawab (Jawahar dkk., 2020).

2.3.*Natural Language Processing* (NLP)

Pemrosesan Bahasa Alami atau *Natural Language Processing* (NLP) adalah bidang ilmu komputer yang bertujuan agar mesin dapat memahami, memanipulasi, dan menghasilkan bahasa manusia, baik dalam bentuk teks maupun ucapan, untuk melakukan tugas-tugas yang bermanfaat (Chowdhury, 2003; Hirschberg & Manning, 2019). Fondasi NLP sendiri berakar pada berbagai disiplin ilmu, mulai dari ilmu komputer, linguistik, hingga kecerdasan buatan, yang semuanya berkontribusi untuk menjembatani kesenjangan komunikasi antara manusia dan mesin (Chowdhury, 2003). Secara praktis, NLP berfokus pada perancangan program komputer yang dapat mempelajari sintaksis dan makna dari bahasa manusia, memprosesnya, kemudian memberikan keluaran yang relevan dan efisien (Jain dkk., 2018). Tujuan utamanya

dapat dibagi menjadi tiga kategori: membantu komunikasi antarmanusia seperti pada sistem penerjemahan mesin, memfasilitasi interaksi antara manusia dan mesin seperti pada agen percakapan, serta menganalisis data linguistik dalam jumlah besar untuk diekstrak informasinya (Hirschberg & Manning, t.t.).

2.4.Tahapan Prapemrosesan Text (*Text Preprocessing*)

Prapemrosesan teks merupakan sebuah tahapan fundamental yang secara langsung menentukan kualitas dan akurasi hasil akhir dalam analisis teks. Pada dasarnya, data teks mentah bersifat tidak terstruktur dan perlu "dibersihkan" serta disiapkan sebelum dapat diolah oleh model komputer. Proses persiapan ini melibatkan beberapa langkah kunci, seperti normalisasi teks untuk memastikan konsistensi (misalnya, mengubah seluruh teks menjadi huruf kecil agar kata 'Analisis' dan 'analisis' dianggap sama). Selanjutnya, dilakukan tokenisasi, yaitu proses memecah kalimat menjadi unit-unit yang lebih kecil seperti kata atau frasa, yang menjadi dasar bagi analisis lebih lanjut. Setelah itu, tahapan penghapusan *stop words* (*stop words removal*) dilakukan untuk menyaring kata-kata umum yang tidak memiliki banyak makna kontekstual (contohnya 'the', 'it', 'and'), sehingga model dapat lebih fokus pada istilah-istilah yang signifikan. Melalui serangkaian tahapan ini, data teks yang awalnya mentah diubah menjadi format yang terstruktur dan siap analisis, sehingga prapemrosesan bukanlah sekadar langkah teknis, melainkan sebuah fondasi krusial yang menjamin keandalan dan ketepatan wawasan yang diperoleh (Chai, 2023a).

2.4.1. Normalisasi Teks (*Text Normalization*)

Normalisasi teks adalah sebuah proses standardisasi yang bertujuan untuk mengubah data teks yang beragam dan tidak konsisten menjadi sebuah bentuk kanonis atau standar (Chai, 2023b; Scannell, 2014). Dalam pemrosesan bahasa alami (NLP), pendekatan modern seringkali memperlakukan normalisasi sebagai masalah terjemahan mesin (*machine translation*), di mana teks yang belum standar (misalnya, teks dengan ejaan lama atau tidak baku) dianggap sebagai "bahasa sumber" dan teks dalam bentuk standarnya dianggap sebagai "bahasa target". Menurut penelitian oleh Scannell (2014), proses ini dapat dimodelkan secara efektif menggunakan model statistik yang merupakan varian dari IBM Model 1, yang tidak memerlukan penataan ulang urutan kata karena kemiripan struktur antara kedua "bahasa" tersebut (Scannell, 2014). Model ini terdiri dari dua komponen utama yang bekerja secara sinergis: Model Bahasa (*Language Model*) dan Model Terjemahan (*Translation Model*).

Komponen pertama, Model Bahasa, berfungsi untuk memastikan bahwa hasil normalisasi merupakan urutan kata yang runut, alami, dan sesuai dengan kaidah bahasa standar. Pembangunannya dimulai dengan mengumpulkan *dataset* teks standar, yang kemudian disaring menggunakan perangkat lunak pemeriksa tata bahasa untuk memilih teks yang paling patuh pada aturan. Dari *dataset* yang telah bersih ini, sebuah model *n-gram* secara spesifik model *trigram* yang menghitung probabilitas sebuah kata berdasarkan dua kata sebelumnya dilatih untuk memahami pola bahasa yang benar.

Komponen kedua, Model Terjemahan, bertugas menghitung probabilitas sebuah kata tidak standar dinormalisasikan menjadi kata standar. Karena data *dataset* paralel seringkali terbatas, model ini menggunakan pendekatan hibrida yang mengandalkan kamus bilingual (leksikon) berkualitas tinggi dan serangkaian aturan perubahan ejaan (*spelling rules*) yang telah terdefinisi. Probabilitasnya dihitung secara sederhana; jika sebuah pemetaan memerlukan penerapan n aturan ejaan, probabilitasnya akan dikalikan dengan sebuah "faktor penalti" yang telah dioptimalkan untuk meminimalkan kesalahan.

Saat proses normalisasi dijalankan pada sebuah kalimat baru (*decoding process*), sistem akan memprosesnya kata per kata dari kiri ke kanan. Untuk setiap kata, semua kemungkinan hipotesis normalisasi beserta probabilitasnya dihasilkan oleh Model Terjemahan. Probabilitas gabungan dari setiap hipotesis kalimat kemudian dihitung dengan skor dari Model Bahasa. Demi efisiensi, sistem akan melakukan pemangkasan (*pruning*), yaitu hanya mempertahankan hipotesis dengan probabilitas tertinggi jika ada beberapa hipotesis yang memiliki dua kata terakhir yang sama. Terakhir, setelah seluruh kalimat diproses, hipotesis dengan probabilitas kumulatif tertinggi dipilih sebagai hasil normalisasi akhir. Dengan demikian, pendekatan ini secara efektif mengintegrasikan kekuatan data statistik dari *dataset* besar dengan pengetahuan linguistik eksplisit untuk menghasilkan teks yang terstandarisasi secara akurat.

2.4.2. Tokenisasi Teks (*Text Tokenization*)

Tokenisasi adalah langkah fundamental dalam pemrosesan bahasa alami yang bertujuan untuk memecah teks menjadi unit-unit yang dapat diproses oleh model. Dalam sebuah penelitian yang berjudul “Unpacking Tokenization: Evaluating Text Compression and its Correlation with Model Performance”, secara spesifik membahas dan menggunakan *Byte Pair Encoding* (BPE), sebuah algoritma tokenisasi yang pada dasarnya bertujuan untuk kompresi data (Goldman dkk., 2024). Proses ini dimulai dengan menginisialisasi kosakata (*vocabulary*) dengan unit simbol paling dasar, yaitu semua karakter individual yang ada dalam *dataset*

pelatihan. Setelah inisialisasi, algoritma secara iteratif mencari pasangan token bersebelahan yang paling sering muncul di seluruh *dataset*, lalu menggabungkan pasangan tersebut menjadi satu token baru dan menambahkannya ke dalam kosakata. Proses penggabungan ini terus diulang hingga ukuran kosakata mencapai batas yang telah ditentukan sebelumnya, misalnya 32.000 token.

Secara teoretis, jurnal tersebut berpendapat bahwa tokenisasi berbasis kompresi seperti BPE dapat dipandang sebagai bentuk pemodelan bahasa yang paling sederhana, yaitu model 0-gram. Pemodelan bahasa pada umumnya bertujuan untuk memaksimalkan probabilitas sebuah teks, yang dihitung sebagai produk dari probabilitas setiap token dalam urutan tersebut, seperti yang ditunjukkan dalam rumus:

$$P(x) = \prod_k P(x_k|\mathrm{x}_{1:k-1}),$$

di mana x_k adalah token ke-k dalam sebuah teks x. Dengan melakukan kompresi, BPE meminimalkan jumlah token dalam sebuah urutan, yang secara efektif membatasi seberapa rendah nilai produk probabilitas di atas. Dalam kerangka 0-gram, probabilitas setiap token dianggap seragam dan tidak bergantung pada konteks sama sekali, sehingga dapat dirumuskan sebagai:

$$P(x_i) = |V|^{-1},$$

di mana $|V|$ adalah total ukuran kosakata (*vocabulary size*). Dengan demikian, tujuan BPE untuk memaksimalkan kompresi (meminimalkan jumlah token) sejalan dengan tujuan pemodelan bahasa untuk memaksimalkan probabilitas teks, menjadikannya metrik yang andal untuk kualitas tokenisasi.

2.4.3. *Stop Words Removal*

Stop words removal adalah salah satu tahapan penting dalam pra-pemrosesan teks (*text preprocessing*) yang bertujuan untuk menyaring dan menghapus kata-kata yang sering muncul namun tidak memiliki makna signifikan (Hidayat dkk., 2021). Kata-kata ini umumnya berupa kata hubung (*conjunction*) yang berfungsi untuk menyambungkan kalimat atau frasa. Kata-kata yang termasuk dalam kategori *stop word* dapat mencakup sekitar 20-30% dari total kata dalam sebuah dokumen (Kannan & Gurusamy, 2014).

Proses ini dilakukan karena kehadiran kata-kata tersebut dalam jumlah besar dapat mengganggu analisis. Dengan menghapusnya, model analisis sentimen diharapkan dapat

menjadi lebih efektif dan efisien, karena dapat lebih fokus pada kata-kata kunci yang mengandung sentimen atau makna utama. Tahapan *stop words removal*, bersama dengan proses lain seperti *stemming* dan *tokenizing*, memiliki pengaruh yang sangat besar terhadap hasil akhir dari analisis sentimen, karena kualitas data yang bersih sangat menentukan akurasi model yang dihasilkan.

2.5.Linguistik Komputasional dan Ekstraksi Fitur Teks

Linguistik Komputasional, yang juga dikenal sebagai *Natural Language Processing* (NLP), adalah bidang ilmu interdisipliner yang berfokus pada perancangan teknologi bahasa. Bidang ini bersifat dinamis; terkadang lebih dekat dengan Linguistik, namun saat ini lebih condong ke arah Ilmu Komputer dan *Machine Learning* (Church & Liberman, 2021). Menurut Manning (2015), linguistik komputasional tidak hanya tentang penerapan metode *machine learning* terbaik, melainkan lebih menekankan pada pemecahan masalah domain fundamental yang terkait dengan bahasa, seperti semantik dan pragmatik. Dalam praktiknya, bidang ini menganalisis berbagai tingkat fitur linguistik untuk memodelkan bahasa manusia, termasuk fitur leksikal, sintaksis, dan stilistik, dengan menggunakan berbagai pendekatan seperti *distributed representations* dan *Universal Dependencies*, yang dianggap sebagai salah satu arah pengembangan yang menjanjikan di masa depan (Church & Liberman, 2021; Manning, 2015).

2.5.1. Fitur Leksikal

Ekstraksi fitur leksikal adalah sebuah pendekatan yang mengandalkan kamus atau korpus yang telah ada, yang dikenal sebagai leksikon, untuk mengidentifikasi dan mengekstrak fitur dari data teks. Pendekatan ini bekerja dengan cara memetakan kata-kata dalam sebuah teks ke dalam kamus leksikon yang sudah berisi daftar kata beserta bobot atau nilai sentimennya. Tahapan utamanya adalah menghitung skor polaritas (*polarity score*) untuk setiap dokumen atau kalimat. Proses ini diawali dengan memberikan skor sentimen pada setiap kata dalam teks yang cocok dengan entri di dalam kamus leksikon. Selanjutnya, keseluruhan skor sentimen dari kata-kata tersebut dijumlahkan untuk menghasilkan satu skor polaritas total untuk teks tersebut. Nilai akhir inilah yang kemudian digunakan sebagai fitur numerik untuk mengklasifikasikan teks ke dalam kategori emosi atau sentimen tertentu, misalnya positif jika skornya lebih dari nol, dan negatif jika skornya kurang dari nol (Nurkasanah & Hayaty, 2022).

2.5.2. Fitur Sintaksis

Fitur sintaksis merujuk pada pola-pola gramatikal yang diekstrak dari sebuah teks untuk mengidentifikasi sentimen. Alih-alih hanya berfokus pada kata-kata tunggal, pendekatan ini menganalisis struktur kalimat secara lebih mendalam untuk menangkap hubungan antar kata yang sering kali menjadi indikator subjektivitas yang kuat (Duric, 2011).

Proses ekstraksi fitur sintaksis tidak memiliki satu rumus perhitungan tunggal, melainkan dilakukan melalui metode identifikasi pola. Salah satu cara yang umum adalah dengan menggunakan pola ekstraksi berbasis *Part-of-Speech* (POS). Pendekatan ini diawali dengan memberi label POS pada setiap kata dalam teks. Selanjutnya, pola-pola gramatikal yang dianggap intuitif sebagai penanda sentimen didefinisikan, misalnya ketika sebuah kata sifat (*adjective*) muncul berdekatan dengan kata benda (*noun*), yang biasanya menandakan bahwa kata sifat tersebut memodifikasi kata benda dengan nuansa subjektif (Duric, 2012).

Pendekatan yang lebih canggih yang diusulkan dalam penelitian Duric (2011) adalah menggunakan model statistik seperti HMM-LDA untuk memisahkan kata secara otomatis ke dalam kelas sintaksis (kata-kata pengubah/*modifier*) dan kelas semantik (entitas atau topik). Dari kelas-kelas ini, kata-kata yang paling representatif kemudian dipilih sebagai fitur. Proses seleksi ini dilakukan dengan memilih kata-kata teratas berdasarkan probabilitas kumulatifnya hingga mencapai ambang batas η yang telah ditentukan. Hal ini dapat dirumuskan sebagai berikut:

Pertama, dihitung fungsi distribusi kumulatif untuk sebuah kata w_i dalam kelas c_j:

$$F_j(w_i) = \sum_{P_{c_j}(w) \geq P_{c_j}(w_i)} P_{cj}(w)$$

Kemudian, himpunan kata (W_{c_j}) untuk kelas c_j didefinisikan sebagai semua kata yang fungsi distribusi kumulatifnya lebih kecil atau sama dengan ambang batas :

$$W_{c_j} = \{w_i | F_j(w_i) \leq \eta$$

Meskipun demikian, salah satu tantangan utama dalam menggunakan fitur sintaksis adalah bahwa struktur gramatikal dalam teks informal seperti ulasan sering kali tidak konsisten, yang dapat membuat pola-pola yang telah didefinisikan secara manual menjadi kurang efektif (Duric, 2012).

2.5.3. Fitur Stilistik

Fitur stilistik merujuk pada pola-pola fitur leksikal, sintaksis, dan diskursus dalam sebuah teks yang secara kolektif membentuk gaya penulisan. Analisis ini bertujuan untuk mengukur variasi linguistik yang dapat membedakan gaya individu, kelompok, atau register komunikatif tertentu, seperti berita atau percakapan. Penelitian yang dilakukan oleh Alkiek et al. (2025) didasarkan pada kerangka *Multidimensional Analysis* (MDA) *Biber*, dengan dua metode ekstraksi: berbasis aturan (*rule-based*) dan berbasis neural (Alkiek dkk., 2025). Proses ekstraksi ini tidak menggunakan rumus matematis yang kompleks, melainkan serangkaian langkah komputasi untuk mengidentifikasi konstruksi linguistik.

Metode ekstraksi berbasis aturan (BIBERPLUS) mengutamakan presisi linguistik, dimulai dengan *parsing* sintaksis menggunakan SpaCy untuk mendapatkan struktur kalimat. Setelah itu, serangkaian aturan yang dibuat manual diterapkan untuk mengidentifikasi dan menghitung 96 fitur stilistik Biber, seperti penggunaan *passive voice*, pronomina, dan penanda sikap (*stance markers*). Perhitungan fitur ini dapat dilakukan dalam dua mode. Untuk teks pendek, digunakan mode penghitungan biner yang hanya mencatat ada atau tidaknya sebuah fitur dalam segmen teks tertentu. Perhitungannya dirumuskan sebagai berikut:

$$Binary\ Count = \frac{Jumlah\ segmen\ yang\ mengandung\ fitur}{Total\ jumlah\ segmen}$$

Untuk teks yang lebih panjang, digunakan mode frekuensi reguler yang menghitung rata-rata kemunculan fitur per jumlah token tertentu. Sementara itu, metode ekstraksi berbasis neural (NEUROBIBER) menggunakan model *Transformer* (RoBERTa) untuk memprediksi ada atau tidaknya 96 fitur tersebut dengan kecepatan tinggi. Untuk teks yang lebih panjang dari kapasitas masukan model, teks dipecah menjadi beberapa bagian, dan hasil prediksinya diagregasi. Pada akhirnya, kedua metode ini menghasilkan sebuah vektor fitur 96-dimensi yang merepresentasikan profil stilistik dari sebuah teks untuk analisis lebih lanjut.

2.6.*Machine Learning* untuk Klasifikasi Teks

Secara umum, klasifikasi teks menggunakan *machine learning* adalah proses fundamental dalam *Natural Language Processing* (NLP) yang bertujuan mengotomatisasi penyortiran data tekstual ke dalam kategori-kategori yang telah ditentukan. Pendekatan ini menjadi krusial karena volume dokumen digital yang dihasilkan setiap hari sangat besar, sehingga klasifikasi manual menjadi proses yang mahal, tidak efisien, dan memakan waktu. Dengan memanfaatkan algoritma *machine learning*, sebuah sistem dapat "dilatih" menggunakan data yang sudah berlabel untuk mengenali pola, yang kemudian memungkinkannya untuk mengklasifikasikan

dokumen baru secara mandiri (Hassan dkk., 2022). Dalam penelitian ini menggunakan jenis *supervised machine learning* dengan algoritma gabungan antara *logistic regression* dan *eXtreme Gradient Boosting* (XGBoost).

2.6.1. *Supervised Machine Learning*

Supervised learning adalah cabang dari *machine learning* yang bertujuan untuk memprediksi atau mengklasifikasikan sebuah variabel hasil (*outcome*) yang telah ditentukan sebelumnya. Disebut *'supervised'* karena keberadaan variabel hasil ini 'mengawasi' proses analisis untuk mempelajari pola dari data. Dalam praktiknya, terdapat berbagai algoritma yang dapat digunakan, yang sering kali dibandingkan dengan model statistik tradisional seperti regresi logistik yang berfungsi sebagai *baseline* untuk mengevaluasi performa. Salah satu pendekatan yang populer adalah metode *ensemble*, yang menggabungkan beberapa model untuk menghasilkan prediksi yang lebih akurat dan stabil. Contohnya adalah *Random Forest*, yang membangun banyak *decision tree* dan menggabungkan hasilnya untuk mengurangi risiko *overfitting*. Pendekatan *ensemble* lain yang lebih canggih adalah *Super Learning*, yang secara sistematis mengombinasikan berbagai jenis algoritma (misalnya, *decision tree*, *support vector machines*, dan regresi) ke dalam satu algoritma komposit tunggal untuk mengoptimalkan performa prediksi. Selain itu, terdapat pula algoritma lain seperti *Gradient Boosting*, yang juga sering digunakan dalam literatur (Jiang dkk., 2020).

2.6.2. *Gradient Boosting Machines* (GBM)

Gradient Boosting Machine (GBM) adalah sebuah teknik *ensemble* yang bertujuan untuk menggabungkan beberapa model pembelajar lemah (*weak learners*), biasanya *decision tree*, menjadi satu model pembelajar kuat (*strong learner*) secara iteratif (Sibindi dkk., 2023). Algoritma ini bekerja sebagai teknik optimisasi yang mencari sebuah model aditif untuk meminimalkan fungsi kerugian (*loss function*). Langkah-langkah dalam GBM dapat diuraikan sebagai berikut:

Proses dimulai dengan membuat sebuah estimasi konstan awal, $f_0(x)$, yang meminimalkan fungsi kerugian.

$$f_0(x) = \arg min_\alpha \sum_{i=1}^{n} L(y_i, \alpha)$$

Untuk setiap iterasi , algoritma secara sekuensial menambahkan sebuah *base learner* baru, , untuk memperbaiki model sebelumnya.

$$f_t(x) = f_{t-1}(x) + \rho_t h_t(x)$$

di mana ρ_t adalah bobot atau laju pembelajaran (*learning rate*) untuk pohon ke-*t*.

Alih-alih melatih pohon baru pada data asli, GBM melatih setiap pohon pada kesalahan atau *pseudo-residual* dari model sebelumnya. *Pseudo-residual* ini dihitung sebagai gradien negatif dari fungsi kerugian.

$$r_{ti} = -\left[\frac{\partial L(y_i, f(x))}{\partial f(x)}\right]_{f(x)=f_{t-1}(x)}$$

Tujuannya adalah agar setiap pohon baru dapat "mengejar" dan memperbaiki kesalahan yang dibuat oleh gabungan pohon-pohon sebelumnya.

Bobot ρ_t untuk setiap pohon baru dihitung dengan menyelesaikan masalah optimisasi pencarian garis (*line search*).

2.6.3. *Extreme Gradient Boosting* (XGBoost)

Extreme Gradient Boosting (XGBoost) adalah implementasi yang lebih canggih dan teroptimalkan dari kerangka *Gradient Boosting*. Algoritma ini dikenal karena kecepatan dan performanya yang tinggi, terutama karena adanya regularisasi untuk mengontrol kompleksitas model dan mencegah *overfitting* secara efektif (Sibindi dkk., 2023). Langkah-langkah dalam XGBoost dapat diuraikan sebagai berikut:

Sama seperti GBM, model prediksi XGBoost, $\hat{y}_i$, adalah penjumlahan dari hasil prediksi sejumlah *P decision tree*.

$$\hat{y}_i^{XG} = \sum_{p=1}^{P} f_p\,(x_i)$$

Perbedaan utama XGBoost terletak pada fungsi objektifnya, yang secara eksplisit menyertakan istilah regularisasi (Ω) untuk menghukum kompleksitas model.

$$Obj = \sum_i L(y_i, \hat{y}_i^{XG}) + \sum_p \Omega(f_p)$$

Istilah regularisasi ini didefinisikan sebagai:

$$\Omega(f_p) = \alpha T + \frac{1}{2}\lambda \|w\|^2$$

di mana T adalah jumlah daun dalam pohon, w adalah bobot daun, serta α dan λ adalah parameter penalti yang mengontrol kompleksitas.

Model dilatih secara aditif. Pada setiap iterasi j , pohon baru f_j ditambahkan untuk meminimalkan fungsi objektif. Fungsi ini didekati menggunakan ekspansi *Taylor* orde kedua untuk optimisasi yang lebih cepat.

Untuk menemukan pemisahan (*split*) terbaik pada setiap simpul pohon, XGBoost menggunakan skor pengurangan kerugian (*loss reduction*) yang dihitung sebagai berikut:

$$Gain = \frac{1}{2}\left[\frac{(\sum_{i\in I_L} g_i)^2}{\sum_{i\in I_L} h_i + \lambda} + \frac{(\sum_{i\in I_R} g_i)^2}{\sum_{i\in I_R} h_i + \lambda} - \frac{(\sum_{i\in I} g_i)^2}{\sum_{i\in I} h_i + \lambda}\right] - \alpha$$

di mana I_L dan I_R adalah himpunan instans di simpul kiri dan kanan setelah pemisahan, g_i dan h_i adalah gradien orde pertama dan kedua dari fungsi kerugian. Pemisahan dengan *gain* tertinggi akan dipilih.

2.6.4. *Logistic Regression*

Regresi Logistik adalah sebuah algoritma klasifikasi yang banyak digunakan dalam *machine learning*, terutama untuk menangani masalah klasifikasi biner dengan memodelkan pengaruh variabel independen terhadap variabel dependen (Uçkan & Karabulut, 2024). Metode ini menggunakan fungsi logistik atau sigmoid untuk mengestimasi probabilitas suatu variabel target masuk ke dalam kategori tertentu. Prosesnya dimulai dengan membentuk kombinasi linear dari variabel-variabel independen $(x_1, x_2, \dots, x_n)$ beserta koefisiennya $(\beta_1, \beta_2, \dots, \beta_n)$ dan sebuah intersep (β_0), yang dirumuskan sebagai berikut:

$$x = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

Hasil dari kombinasi linear ini kemudian dimasukkan ke dalam fungsi logistik untuk diubah menjadi nilai probabilitas antara 0 dan 1. Estimasi probabilitas bahwa variabel dependen () akan bernilai 1 dihitung dengan rumus:

$$P(y = 1|x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n)}}$$

Pada dasarnya, regresi logistik menggunakan sebuah batas keputusan linear untuk memisahkan titik-titik data dan memprediksi probabilitas terjadinya suatu peristiwa tertentu.

2.7.*Explainable Artificial Intelligence* (XAI)

Explainable Artificial Intelligence (XAI) adalah sebuah bidang dalam kecerdasan buatan yang mempromosikan seperangkat alat, teknik, dan algoritma yang dapat menghasilkan penjelasan yang dapat ditafsirkan, intuitif, dan dapat dipahami manusia atas keputusan yang dibuat oleh sistem AI. Tujuan utama XAI adalah untuk meningkatkan transparansi, kepercayaan, dan keadilan dalam proses pengambilan keputusan oleh model *machine learning*, terutama pada model *deep neural network* yang sifatnya kompleks dan sering dianggap sebagai "kotak hitam" (*black-box*). Sebuah penjelasan (*explanation*) dalam konteks XAI merupakan informasi meta tambahan yang dihasilkan baik oleh model itu sendiri maupun oleh algoritma eksternal, yang berfungsi untuk menggambarkan pentingnya atau relevansi sebuah fitur masukan terhadap keputusan klasifikasi tertentu. Dengan adanya XAI, keputusan yang dihasilkan oleh model AI tidak hanya akurat, tetapi juga dapat diverifikasi dan dipahami oleh berbagai pihak, mulai dari pengembang, ahli di bidang terkait, hingga pengguna awam (Das & Rad, 2020).

2.8.*SHapley Additive exPlanations* (SHAP)

SHapley Additive exPlanations (SHAP) adalah sebuah metode dalam XAI yang menggunakan pendekatan teori permainan (*game theory*) untuk menjelaskan prediksi individual dari sebuah model *machine learning*. SHAP menghitung kontribusi setiap fitur terhadap prediksi dengan menganggap setiap fitur sebagai "pemain" dalam sebuah permainan koalisi. Metode ini mendistribusikan "pembayaran" (selisih antara prediksi aktual dan prediksi rata-rata) secara adil kepada setiap fitur sesuai dengan kontribusi marginalnya. Model penjelasan SHAP (g) diformulasikan sebagai fungsi aditif linear dari fitur, yang secara matematis dapat dituliskan sebagai berikut (Das & Rad, 2020):

$$g(z') = \phi_0 + \sum_{j=1}^{M} \phi_j z'_j$$

Di z' mana adalah vektor koalisi (representasi biner dari ada atau tidaknya sebuah fitur), M adalah jumlah maksimal fitur, ϕ_0 adalah nilai prediksi dasar (*base value*), dan ϕ_j adalah nilai Shapley yang merepresentasikan kontribusi atau atribusi dari fitur ke-j.

Proses dalam algoritma KernelSHAP dimulai dengan mengambil sebuah instans data yang akan dijelaskan, lalu secara acak membuat berbagai "koalisi" fitur dengan menghilangkan sebagian fitur dari instans tersebut. Selanjutnya, model black-box yang asli digunakan untuk membuat prediksi pada setiap sampel koalisi yang telah dibuat. Setiap sampel koalisi ini

kemudian diberi bobot menggunakan *SHAP kernel*, yang memberikan bobot lebih besar pada koalisi yang hanya memiliki sedikit atau sangat banyak fitur. Setelah itu, sebuah model linear yang dapat diinterpretasikan dilatih pada sampel-sampel koalisi ini, dengan prediksi dari model *black-box* sebagai target dan bobot dari *SHAP kernel* sebagai bobot sampel. Pada akhirnya, koefisien dari model linear yang telah dilatih inilah yang kemudian dianggap sebagai nilai Shapley (ϕ_j), yang merepresentasikan kontribusi setiap fitur terhadap prediksi.

2.9.Metrik Evaluasi Model Klasifikasi

Evaluasi performa model klasifikasi dilakukan dengan menggunakan serangkaian metrik yang dihitung dari matriks kebingungan (*confusion matrix*). Matriks ini membandingkan hasil prediksi model dengan nilai aktual, yang dikategorikan ke dalam empat komponen dasar: *True Positive* (TP), yaitu saat nilai aktual dan prediksi sama-sama positif; *True Negative* (TN), yaitu saat nilai aktual dan prediksi sama-sama negatif; *False Positive* (FP) atau *Type I error*, yaitu saat prediksi positif padahal nilai aktualnya negatif; dan *False Negative* (FN) atau *Type II error*, yaitu saat prediksi negatif padahal nilai aktualnya positif (Sathyanarayanan, 2024).

Dari komponen-komponen ini, beberapa metrik utama dapat dihitung. Akurasi adalah metrik yang paling umum, mengukur fraksi prediksi yang benar dari keseluruhan prediksi. Metrik ini dihitung dengan rumus:

$$Akurasi = \frac{TP + TN}{TP + FP + TN + FN}$$

Presisi (*Precision*) mengukur fraksi dari prediksi positif yang benar-benar positif, yang berguna ketika biaya *False Positive* tinggi. Rumusnya adalah sebagai berikut:

$$Presisi = \frac{TP}{TP + FP}$$

Selanjutnya adalah *Recall*, yang juga dikenal sebagai Sensitivitas (*Sensitivity*), mengukur proporsi dari kasus positif aktual yang berhasil diidentifikasi dengan benar oleh model. Metrik ini sangat penting ketika biaya *False Negative* tinggi, seperti dalam diagnosis medis. *Recall* dihitung dengan rumus:

$$Recall = \frac{TP}{TP + FN}$$

Terakhir, *F1-Score* digunakan untuk menyeimbangkan antara presisi dan *recall*, terutama pada kasus *dataset* yang tidak seimbang (*imbalanced*). Metrik ini adalah rata-rata harmonik

(*harmonic mean*) dari presisi dan *recall*, yang memberikan bobot lebih pada nilai yang lebih rendah. Nilai terbaik untuk F1-Score adalah 1 dan nilai terburuknya adalah 0. *F1-Score* dihitung dengan rumus berikut:

$$F1\ Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall}$$

BAB III

METODOLOGI PENELITIAN

Pada bab ini akan diuraikan metodologi penelitian yang digunakan, mencakup peralatan yang dibutuhkan, perancangan sistem, alur kerja, serta teknik analisis data. Metodologi ini menjadi landasan untuk menjawab rumusan masalah yang telah ditetapkan.

3.1.Jenis Peneltian

Penelitian ini menggunakan pendekatan penelitian terapan (*applied research*). Fokus utamanya adalah untuk memecahkan masalah praktis dan spesifik yang dihadapi di dunia nyata. Dalam konteks ini, masalah praktis tersebut adalah kegagalan alat deteksi AI konvensional yang beroperasi sebagai "kotak hitam" (*black box*), sehingga tidak memiliki transparansi dan akuntabilitas. Oleh karena itu, penelitian ini tidak berhenti pada pengembangan teori, melainkan bertujuan untuk menghasilkan solusi yang dapat diaplikasikan secara langsung, yaitu sebuah dasar untuk sistem deteksi yang transparan dan dapat dipertanggungjawabkan.

Untuk mencapai tujuan terapan tersebut, penelitian ini akan dilaksanakan dengan desain eksperimental kuantitatif. Sifat eksperimental ini terwujud melalui serangkaian langkah yang terstruktur dan terkontrol. Penelitian akan melibatkan pembangunan model *machine learning* untuk mengklasifikasikan teks, yang kemudian akan dianalisis dalam sebuah lingkungan yang terkendali. Eksperimen ini mencakup pengujian model pada *dataset* yang dirancang khusus (terdiri dari teks manusia dan AI) dan penerapan metode SHAP secara sistematis.

Secara konkret, eksperimen akan berfokus pada manipulasi dan pengukuran variabel-variabel yang jelas. Model *machine learning* akan bertindak sebagai subjek eksperimen, sementara "perlakuan"-nya adalah data teks dari berbagai genre (berita dan esai kreatif). Variabel yang diukur adalah kontribusi marka linguistik (leksikal, sintaksis, dan stilistik) terhadap keputusan model. Melalui analisis SHAP, penelitian ini secara kuantitatif akan mengidentifikasi, mengukur, dan memvisualisasikan "sidik jari" linguistik, sehingga menjawab rumusan masalah secara empiris.

3.2.Peralatan

Peralatan yang digunakan dalam penelitian ini terdiri dari perangkat lunak (*software*) dan perangkat keras (*hardware*).

1. Kebutuhan Perangkat Lunak (*Software*)

   Perangkat lunak utama yang digunakan adalah:

   - Bahasa Pemrograman: Python 3.11
   - Lingkungan Pengembangan: Jupyter Notebook, Microsoft Visual Studio Code
   - *Library* Utama:
     - Manipulasi Data: Pandas, NumPy
     - Pemrosesan Teks & Ekstraksi Fitur: NLTK, spaCy, textstat
     - Pemodelan *Machine Learning*: Scikit-learn, XGBoost
     - *Explainable AI* (XAI): SHAP
     - Visualisasi Data: Matplotlib, Seaborn
2. Kebutuhan Perangkat Keras (*Hardware*)

   Penelitian ini menggunakan laptop dengan spesifikasi sebagai berikut:

   - CPU: AMD Ryzen 5 5500U @ 2.10GHz
   - RAM: 16 GB
   - Sistem Operasi: Windows 11 Home 64-bit

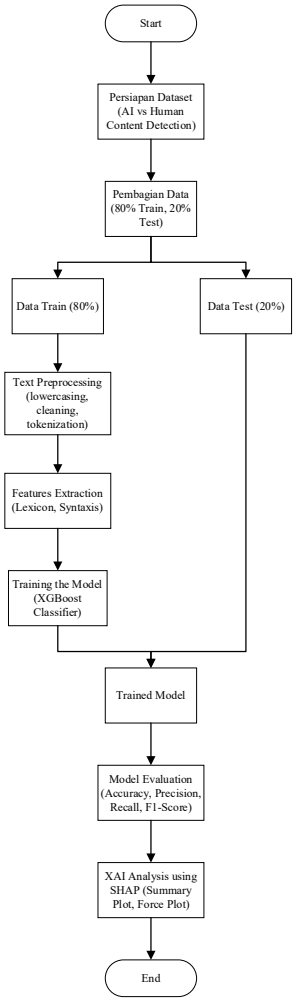
3.3.Pengumpulan Data

Penelitian ini menggunakan data sekunder yang diperoleh dari Kaggle. *Dataset* yang digunakan adalah *dataset* "AI vs Human Content Detection" yang dikurasi oleh Pratyush Puri. *Dataset* ini secara unik tidak menyajikan teks mentah, melainkan menyediakan 17 fitur linguistik dan stilometrik yang telah diekstraksi sebelumnya dari setiap sampel teks. Kolom-kolom tersebut meliputi content_type, word_count, character_count, sentence_count, lexical_diversity, avg_sentence_length, avg_word_length, punctuation_ratio, flesch_reading_ease, gunning_fog_index, grammar_errors, passive_voice_ratio, predictability_score, burstiness, dan sentiment_score. Variabel target untuk klasifikasi adalah kolom label, di mana nilai '1' menandakan konten yang dihasilkan oleh AI dan '0' menandakan konten yang ditulis oleh manusia. Pendekatan berbasis fitur ini memungkinkan analisis stilometrik yang mendalam dan pengembangan model klasifikasi yang cepat tanpa memerlukan pemrosesan bahasa alami yang ekstensif dari awal.

3.4.Perancangan Sistem Klasifikasi

Perancangan sistem bertujuan untuk membangun sebuah alur kerja yang dapat mengklasifikasikan teks buatan AI dan manusia, kemudian menganalisis proses pengambilan keputusan model tersebut. Secara umum, perancangan sistem dibagi menjadi dua bagian

utama: sistem latih (*training*) dan sistem uji (*testing*), yang keduanya akan melalui tahapan prapemrosesan dan ekstraksi fitur. Alur kerja penelitian secara keseluruhan dapat dilihat pada Gambar 3.1.



Gambar 3.1. Flowchart Alur Kerja Penelitian

3.4.1. Prapemrosesan Teks

Tahap prapemrosesan teks merupakan langkah fundamental dalam alur kerja *Natural Language Processing* (NLP) yang bertujuan untuk mengubah data teks mentah yang tidak terstruktur menjadi format yang bersih, konsisten, dan terstruktur. Sebagaimana dijelaskan dalam dasar teori, model *machine learning* tidak dapat memproses bahasa alami secara langsung, sehingga diperlukan serangkaian proses untuk membersihkan dan menstandarisasi data. Tahap ini krusial untuk mengurangi *noise* (gangguan) dan variasi data yang tidak relevan, yang pada akhirnya dapat meningkatkan efektivitas dan akurasi model pada tahap klasifikasi.

Proses prapemrosesan dalam penelitian ini meliputi beberapa langkah utama. *Input* dari tahap ini adalah korpus teks mentah dari *dataset* yang telah dikumpulkan. Proses pertama

adalah Normalisasi Teks melalui *lowercasing*, yaitu mengubah seluruh karakter dalam teks menjadi huruf kecil. Langkah ini penting untuk memastikan konsistensi, sehingga kata seperti "Human", "human", dan "HUMAN" dianggap sebagai token tunggal yang sama oleh model, bukan sebagai tiga kata yang berbeda. Selanjutnya, dilakukan Pembersihan Teks dengan menghapus elemen-elemen yang tidak memiliki nilai semantik untuk tugas klasifikasi ini, seperti karakter khusus (tanda baca berlebih), angka, dan spasi ganda. *Output* yang dihasilkan dari tahap prapemrosesan ini adalah kumpulan teks yang bersih dan seragam, yang siap untuk melalui proses Ekstraksi Fitur Linguistik pada tahap berikutnya.

3.4.2. Ekstraksi Fitur Linguistik

Tahap ekstraksi fitur linguistik merupakan inti dari penelitian ini, di mana ciri-ciri kebahasaan yang membedakan antara teks generatif AI dan teks manusia diukur dan diubah menjadi representasi numerik. Sebagaimana disinggung dalam dasar teori Linguistik Komputasional, proses ini mengubah data teks yang tidak terstruktur menjadi format tabel terstruktur yang dapat diolah oleh algoritma *machine learning*. Fitur-fitur ini akan menjadi variabel independen (X) yang menjadi masukan bagi model klasifikasi. Proses ini sangat krusial karena kualitas dan relevansi fitur yang diekstraksi akan secara langsung memengaruhi kinerja model dalam membedakan kedua jenis teks.

1. Ekstraksi Fitur Leksikal

   Fitur leksikal berfokus pada analisis kosa kata yang digunakan dalam teks. Berdasarkan teori linguistik komputasional, karakteristik leksikal seringkali menjadi indikator awal perbedaan gaya penulisan. *Input* untuk tahap ini adalah teks yang telah melalui proses prapemrosesan (teks bersih). Proses yang dilakukan meliputi perhitungan metrik-metrik berikut:

   - Jumlah Kata Unik: Mengukur seberapa banyak kosa kata yang berbeda digunakan dalam teks.
   - Panjang Rata-rata Kata: Mengindikasikan kecenderungan penggunaan kata-kata pendek atau panjang.
   - *Type-Token Ratio* (TTR): Mengukur kekayaan kosa kata atau variasi leksikal. Teori sebelumnya mengemukakan bahwa teks yang dihasilkan AI cenderung memiliki TTR yang lebih rendah, mengindikasikan kosa kata yang lebih monoton dan kurang bervariasi dibandingkan tulisan manusia.

Output dari tahap ini adalah nilai-nilai numerik yang merepresentasikan karakteristik leksikal dari setiap teks.

2. Ekstraksi Fitur Sintaksis

   Fitur sintaksis menganalisis struktur dan kompleksitas kalimat dalam teks. Studi tentang gaya penulisan seringkali melibatkan evaluasi struktur kalimat, karena dapat mencerminkan tingkat pemahaman dan kedalaman ekspresi. *Input* untuk tahap ini juga adalah teks bersih. Proses yang dilakukan meliputi perhitungan metrik-metrik berikut:

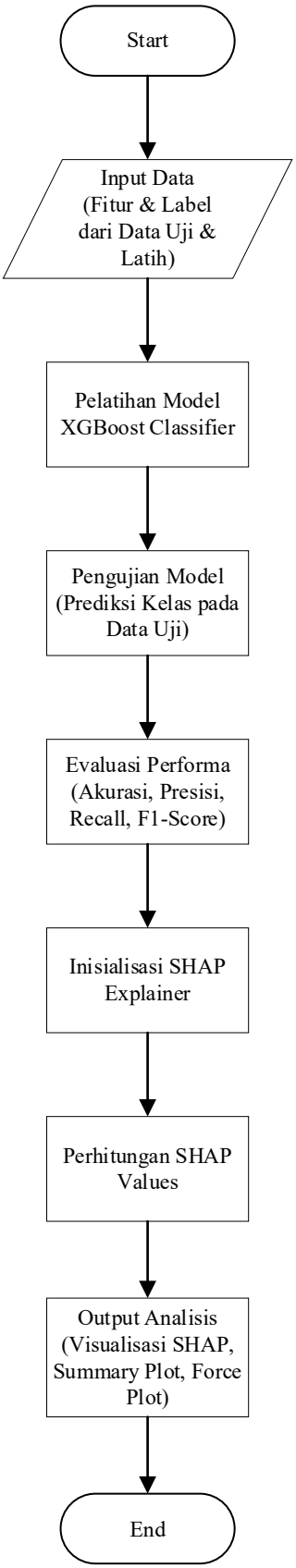
   - Panjang Rata-rata Kalimat: Menunjukkan kecenderungan penulis (AI atau manusia) dalam membangun kalimat yang ringkas atau kompleks.
   - Skor Keterbacaan *Flesch Reading Ease*: Metrik ini mengukur seberapa mudah suatu teks dipahami. Penelitian terdahulu menunjukkan bahwa teks AI seringkali memiliki skor keterbacaan yang sangat konsisten, bahkan "terlalu sempurna" karena cenderung menghindari variasi struktur kalimat yang rumit atau terlalu sederhana.

   Output dari tahap ini adalah nilai-nilai numerik yang menggambarkan karakteristik sintaksis dari setiap teks, yang kemudian akan menjadi bagian dari fitur input model.

3.4.3. Klasifikasi dengan XGBoost dan Analisis SHAP

Tahap ini merupakan puncak dari alur kerja penelitian, di mana fitur-fitur linguistik yang telah diekstraksi dari teks akan digunakan untuk membangun model klasifikasi, diikuti dengan analisis mendalam menggunakan *Explainable AI* (XAI). Tujuannya adalah tidak hanya mengklasifikasikan teks sebagai buatan manusia atau AI, tetapi juga untuk memahami mengapa model mengambil keputusan tersebut, dengan mengidentifikasi marka-marka

linguistik paling berpengaruh. Proses ini secara rinci digambarkan pada Gambar 3.2.



Gambar 3.2. Flowchart Proses Klasifikasi dan Analisis XAI

Input utama untuk tahap ini adalah data numerik yang berisi fitur-fitur linguistik hasil ekstraksi, beserta label kelasnya (Manusia atau AI). Data ini kemudian dibagi menjadi set pelatihan dan pengujian. Langkah pertama adalah Pelatihan Model XGBoost. XGBoost (*eXtreme Gradient Boosting*) dipilih sebagai model utama karena reputasinya yang tinggi dalam Pembelajaran Terarah (*Supervised Learning*) dan kemampuannya untuk menangani data tabular dengan efisien dan akurasi tinggi. Sebagai algoritma *Gradient Boosting Machines* (GBM), XGBoost membangun sebuah *ensemble* dari pohon keputusan secara sekuensial, di mana setiap pohon baru mencoba memperbaiki kesalahan prediksi dari pohon sebelumnya. Model ini dilatih menggunakan data pelatihan untuk mempelajari pola dan hubungan antara fitur-fitur linguistik dan label kelas teks.

Setelah model XGBoost terlatih, langkah berikutnya adalah Pengujian Model. Model yang telah terlatih akan digunakan untuk memprediksi label kelas pada data pengujian, yaitu data yang belum pernah dilihat sebelumnya oleh model. Hasil prediksi ini kemudian akan dievaluasi untuk mengukur kinerja model menggunakan Metrik Evaluasi Model Klasifikasi seperti Akurasi, Presisi, *Recall*, dan F1-Score. Metrik-metrik ini, yang dihitung dari *Confusion Matrix*, akan memberikan gambaran komprehensif tentang seberapa efektif model dalam mengidentifikasi teks AI sekaligus meminimalkan kesalahan klasifikasi.

Bagian krusial dari penelitian ini adalah Analisis XAI dengan SHAP. Untuk mengatasi "masalah kotak hitam" (*black box problem*) pada model kompleks seperti XGBoost, digunakan *library* SHAP (*SHapley Additive exPlanations*). Prosesnya diawali dengan Inisialisasi *SHAP Explainer*, di mana sebuah objek *explainer* dibuat berdasarkan model XGBoost yang telah dilatih. Kemudian, dilakukan Perhitungan *SHAP Values* untuk setiap fitur pada setiap sampel data pengujian. Berdasarkan Teori Permainan Kooperatif, *Shapley Values* ini secara adil mendistribusikan kontribusi setiap fitur terhadap perbedaan antara prediksi model dan *base value* (rata-rata prediksi). *Output* Analisis dari SHAP mencakup:

- Evaluasi Performa: Hasil metrik klasifikasi yang telah disebutkan sebelumnya, disajikan untuk mengkonfirmasi validitas model sebelum interpretasi.
- Visualisasi SHAP: Ini adalah hasil interpretatif utama. *SHAP Summary Plot* akan menampilkan fitur-fitur linguistik yang paling berpengaruh secara global dalam keputusan klasifikasi model, serta arah pengaruhnya (misalnya, nilai TTR rendah cenderung mengindikasikan teks AI). Selain itu, *SHAP Force Plot* akan digunakan untuk memvisualisasikan kontribusi fitur secara individual untuk beberapa sampel

teks tertentu, memberikan pemahaman mikro tentang bagaimana fitur-fitur tersebut "mendorong" prediksi ke arah kelas "Manusia" atau "AI".

Seluruh proses ini memungkinkan penelitian tidak hanya untuk membangun detektor teks AI yang akurat, tetapi yang lebih penting, untuk mendapatkan pemahaman mendalam dan transparan mengenai marka-marka linguistik spesifik yang membedakan tulisan AI dari tulisan manusia, yang sejalan dengan tujuan utama penelitian ini.