BAB 2
LANDASAN TEORI

Bab ini menyajikan landasan teori yang digunakan sebagai dasar dalam memahami permasalahan penelitian dan menyusun solusi yang diusulkan. Teori-teori yang dibahas dalam bab ini meliputi konsep ujaran kebencian, karakteristik platform X sebagai media sosial, metode *deep learning*, arsitektur IndoBERT, serta *Convolutional Neural Network* (CNN) yang menjadi dasar dalam pengembangan model klasifikasi pada studi ini. Selain itu, bab ini menguraikan konsep *Explainable Generative AI*, metrik evaluasi klasifikasi untuk membandingkan kinerja model, dan *dataset* yang digunakan dalam penelitian.

2.1 Ujaran Kebencian (*Hate Speech*)

Ujaran yang penuh kebencian atau yang biasa disebut sebagai (*hate speech*) dapat dipahami sebagai komunikasi yang memuat ungkapan kebencian, penghinaan, diskriminasi, atau dorongan untuk melakukan kekerasan terhadap orang atau kelompok tertentu yang didasarkan pada karakteristik seperti ras, etnis, agama, gender, orientasi seksual, dan pandangan politik [1, 23, 24]. Dalam konteks media sosial, ujaran kebencian dapat menyebar dengan cepat dan berdampak negatif terhadap stabilitas sosial serta kesehatan psikologis individu. Sepanjang zaman, ujaran yang penuh kebencian telah wujud, termasuk dalam aspek keagamaan, yang bisa muncul dalam bentuk tuduhan, desas-desus, dan rasa cemburu, mempengaruhi orang-orang serta kelompok [25].

Ujaran kebencian dipahami sebagai bentuk ekspresi yang merendahkan atau mendiskreditkan kelompok tertentu yang mengalami tekanan sosial secara sistematis, yang umumnya dibedakan berdasarkan karakteristik seperti jenis kelamin, orientasi seksual, maupun keyakinan agama. Ekspresi yang mengandung kebencian dapat mengurangi rasa aman dan menurunkan tingkat kepercayaan dalam masyarakat, terutama pada lingkungan yang memiliki keberagaman etnis dan budaya. Selain itu, ujaran kebencian berpotensi memicu perpecahan sosial dengan memperdalam konflik antarkelompok, baik yang bersifat etnis maupun sosial, sehingga dapat mengancam persatuan dan stabilitas suatu negara [26]. Dalam ranah politik, ujaran kebencian kerap digunakan sebagai sarana untuk memobilisasi dukungan politik sekaligus menyerang pihak lawan dengan memanfaatkan sentimen

berbasis identitas agama maupun etnis [25]. Keadaan ini mampu menguatkan penerapan politik identitas dan berpotensi mengurangi prinsip-prinsip demokrasi dalam masyarakat dan negara.

Dalam ranah *Natural Language Processing* (NLP), deteksi ujaran kebencian dipandang sebagai permasalahan klasifikasi teks, baik dalam bentuk klasifikasi biner maupun multi-kelas [27]. Tantangan utama dalam deteksi ujaran kebencian meliputi ambiguitas bahasa, sarkasme, variasi konteks budaya, serta ketidakseimbangan data [28]. Oleh karena itu, diperlukan model yang mampu memahami konteks bahasa secara mendalam dan adaptif terhadap variasi linguistik.

2.2 Platform X

Platform X, sebelumnya bernama Twitter, merupakan media sosial yang memungkinkan pengguna memublikasikan pesan singkat kepada audiens luas. Platform ini menyediakan peluang bagi penggunanya untuk mengirimkan pesan singkat disebut sebagai *tweet* kepada audiens global. Setiap *tweet* dapat memuat hingga 280 karakter dan bisa terdiri dari teks, gambar, video, maupun tautan ke sumber lainnya. Secara bawaan, konten yang dipublikasikan melalui *tweet* dapat diakses oleh publik dan tidak terbatas hanya pada pengguna yang mengikuti akun tertentu [29]. Karakteristik utama Platform X yang meliputi komunikasi secara *real-time*, keterbukaan informasi, serta kecepatan dalam penyebaran konten menjadikannya sebagai salah satu media yang berpengaruh dalam penyampaian informasi dan diskusi publik [30]. Oleh karena itu, platform ini banyak dimanfaatkan untuk penyebaran berita, membangun jejaring sosial, serta memfasilitasi interaksi dan pertukaran gagasan di masyarakat [29].

Selain digunakan untuk menyampaikan informasi dan mengikuti perkembangan isu terkini, Platform X juga menjadi ruang bagi individu, organisasi, politisi, dan tokoh publik untuk berkomunikasi secara langsung dengan masyarakat [31]. Besarnya volume data yang tersedia, sifat komunikasinya yang berlangsung secara cepat, serta aksesibilitas data yang relatif terbuka menjadikan Platform X sebagai sumber data yang penting dalam penelitian terkait ujaran kebencian. Data yang diperoleh dari platform ini sering digunakan untuk mengembangkan, pelatihan, pengujian, dan evaluasi terhadap model pembelajaran mesin untuk tugas deteksi ujaran kebencian. Selain itu, para peneliti juga memanfaatkan data tersebut untuk mempelajari pola penyebaran, perkembangan konten bermuatan kebencian, serta efektivitas berbagai strategi mitigasi, termasuk pendekatan *counter-speech*.

Berbagai penelitian telah menghasilkan dan mempublikasikan *dataset* beranotasi dalam skala besar yang bersumber dari Platform X. Salah satu contohnya adalah *COVID-HATE Dataset*, yang mencakup lebih dari 206 juta *tweet* dan digunakan untuk mengkaji fenomena kebencian terhadap masyarakat Asia selama pandemi COVID-19 [32]. Selain itu, tersedia pula sejumlah *dataset* untuk deteksi ujaran kebencian dalam berbagai bahasa, seperti bahasa Inggris, Hindi, dan Marathi [33].

2.3 *Natural Language Processing* (NLP)

Natural Language Processing (NLP) merupakan salah satu cabang dalam *Artificial Intelligence* (AI) yang berfokus pada pengembangan metode dan algoritma agar komputer mampu memahami, mengolah, serta menganalisis bahasa alami yang digunakan oleh manusia. Pada penelitian deteksi ujaran kebencian, NLP berperan dalam mengubah teks menjadi representasi yang dapat diproses komputer, mengekstraksi karakteristik linguistik, serta melakukan klasifikasi terhadap konten berbasis teks.

Pendekatan NLP tradisional umumnya memanfaatkan metode *bag-of-words* dan *TF-IDF* yang dipadukan dengan algoritma *machine learning* klasik [34]. Meskipun demikian, pendekatan tersebut memiliki keterbatasan dalam memahami konteks semantik suatu teks. Perkembangan *deep learning*, terutama melalui arsitektur IndoBERT, memungkinkan sistem memodelkan konteks bahasa secara lebih mendalam dan kontekstual [35].

2.4 Data *Preprocessing*

Data *preprocessing* Data preprocessing adalah sebuah cara untuk mengubah data mentah menjadi bentuk yang diinginkan sehingga informasi yang berguna dapat diperoleh darinya [36]. Proses ini melibatkan serangkaian langkah seperti membersihkan data, menghilangkan *noise*, dan mengorganisir data agar sesuai dengan kebutuhan analisis. Dalam penelitian ini, digunakan beberapa teknik *preprocessing* sebagai berikut:

1. *Labbeling* adalah proses untuk memberikan label pada setiap data tweet berdasarkan kategori yang telah ditentukan.

2. Data *leaning* adalah proses penting untuk mempersiapkan data teks sebelum diolah oleh model. Langkah-langkah ini melibatkan penghapusan elemen

yang tidak relevan atau mengganggu, seperti URL, tautan, atau tanda baca yang tidak dibutuhkan.

3. *Stemming* pada penelitian ini tidak diterapkan karena model yang digunakan adalah IndoBERT. IndoBERT memanfaatkan *contextual embedding* sehingga lebih efektif apabila kata dipertahankan dalam bentuk aslinya agar informasi konteks tidak hilang.

2.5 *Deep Learning*

Deep learning adalah pendekatan dalam *machine learning* yang termasuk ke dalam ranah kecerdasan buatan dan mengandalkan jaringan saraf tiruan multilapis untuk melakukan pembelajaran representasi data secara otomatis. Pendekatan ini mampu mengidentifikasi pola yang kompleks dari kumpulan data berskala besar [37]. Metode ini terbukti efektif dalam mengelola data dengan dimensi yang besar dan tingkat kompleksitas yang tinggi, yang telah mendorong kemajuan dalam sektor-sektor termasuk dalam aplikasi pengenalan citra, pengenalan suara, dan analisis bahasa alami.

Salah satu kelebihan utama *deep learning* adalah potensi untuk meningkatkan performa seiring bertambahnya jumlah dan kompleksitas data yang diproses, sehingga sangat sesuai untuk diterapkan pada lingkungan *big data*. Selain itu, model *deep learning* dapat diadaptasi untuk berbagai sektor, termasuk kesehatan, keamanan, industri, dan pemerintahan. Salah satu struktur yang paling umum diterapkan dalam *deep learning* adalah CNN, yang telah menjadi metode utama dalam berbagai tugas seperti pengenalan citra (*image recognition*), klasifikasi, dan deteksi objek [38].

2.6 IndoBERT

IndoBERT merupakan model bahasa berbasis arsitektur IndoBERT yang dikembangkan khusus untuk bahasa Indonesia. Model ini dibangun dengan memanfaatkan mekanisme *self-attention* dan dilatih menggunakan korpus teks berbahasa Indonesia dalam skala besar sehingga mampu membentuk representasi semantik yang kontekstual dengan mempertimbangkan karakteristik linguistik dan pola penggunaan bahasa Indonesia. Melalui pendekatan *bidirectional*, IndoBERT dapat memahami makna suatu kata dengan mempertimbangkan konteks sebelum dan sesudah kata tersebut dalam kalimat.

Secara umum, IndoBERT adalah arsitektur *deep learning* yang mengandalkan mekanisme *self-attention* untuk mempelajari hubungan antar token dalam suatu urutan teks [35]. Berbeda dengan *recurrent neural network* (RNN) yang memproses data secara berurutan, IndoBERT memungkinkan pemrosesan dilakukan secara paralel sehingga lebih efisien dalam menangani data berukuran besar dan mampu menangkap ketergantungan konteks jangka panjang.

Sebagai adaptasi dari *Bidirectional Encoder Representations from Transformers* (BERT), IndoBERT dirancang untuk memahami karakteristik bahasa Indonesia secara lebih baik dibandingkan model BERT umum. Kemampuan tersebut menjadikan IndoBERT efektif untuk berbagai tugas *Natural Language Processing* (NLP), seperti klasifikasi teks, analisis sentimen, deteksi emosi, serta deteksi ujaran kebencian. Berbagai penelitian menunjukkan bahwa model berbasis IndoBERT mampu menghasilkan performa yang lebih baik dibandingkan pendekatan klasifikasi konvensional pada berbagai tugas NLP [14, 15].

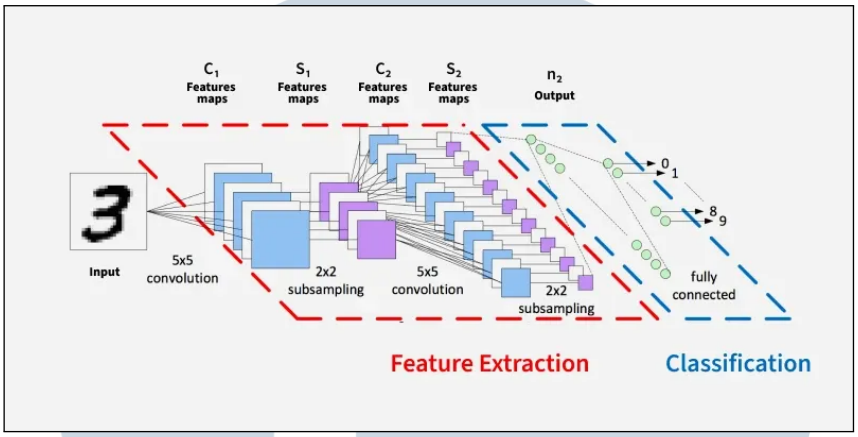
Meskipun memiliki kemampuan klasifikasi yang tinggi, model berbasis IndoBERT umumnya bersifat *black box* sehingga proses pengambilan keputusannya sulit dipahami oleh pengguna [20]. Selain itu, penggunaan satu arsitektur tunggal terkadang belum mampu menangkap seluruh variasi pola linguistik yang kompleks. Oleh sebab itu, pendekatan *hybrid* yang mengombinasikan IndoBERT dan CNN dapat dimanfaatkan untuk meningkatkan kemampuan ekstraksi fitur sekaligus menghasilkan performa klasifikasi yang lebih optimal.

2.7 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan salah satu arsitektur *deep learning* yang dikembangkan untuk pengolahan citra. Seiring perkembangan *Natural Language Processing* (NLP), CNN berhasil diadaptasi untuk berbagai tugas klasifikasi teks, seperti analisis sentimen, deteksi spam, dan deteksi ujaran kebencian [38]. Pada klasifikasi teks, CNN bekerja dengan mengekstraksi pola-pola lokal pada urutan kata sehingga mampu mengenali frasa atau kombinasi kata yang memiliki makna penting terhadap suatu kategori [39].

Berbeda dengan CNN pada pengolahan citra yang melakukan operasi konvolusi terhadap matriks dua dimensi piksel, CNN untuk klasifikasi teks melakukan operasi konvolusi terhadap representasi vektor kata (*word embedding*) atau representasi kontekstual yang dihasilkan oleh model bahasa, seperti IndoBERT. Dengan demikian, setiap filter konvolusi mempelajari pola linguistik tertentu,

seperti kombinasi kata atau *n-gram*, yang berkontribusi terhadap proses klasifikasi [40].



Gambar 2.1. *Text classification using CNN*

Sumber: [39]

Secara umum, arsitektur CNN untuk klasifikasi teks terdiri atas beberapa komponen sebagai berikut.

1. Embedding Layer

   Lapisan ini mengubah setiap kata menjadi representasi vektor berdimensi tetap. Pada pendekatan konvensional, representasi kata diperoleh menggunakan metode *word embedding* seperti Word2Vec atau GloVe. Namun, pada penelitian ini representasi kata dihasilkan oleh IndoBERT sehingga setiap token memiliki representasi kontekstual yang mempertimbangkan hubungan dengan token-token di sekitarnya.

2. Convolution Layer

   Lapisan konvolusi menerapkan sejumlah filter (*kernel*) pada urutan token untuk mendeteksi pola-pola lokal dalam teks. Setiap filter bergerak sepanjang urutan token dan menghasilkan *feature map* yang menunjukkan keberadaan pola tertentu. Pada klasifikasi teks, ukuran filter umumnya merepresentasikan panjang frasa (*n-gram*), antara lain:

   a. Ukuran filter 3 digunakan untuk menangkap pola *trigram*.

   b. Ukuran filter 4 digunakan untuk menangkap frasa yang terdiri atas empat kata.

c. Ukuran filter 5 digunakan untuk menangkap pola linguistik yang lebih panjang.

Penggunaan beberapa ukuran filter memungkinkan model mempelajari berbagai variasi pola bahasa secara lebih komprehensif sehingga mampu meningkatkan kemampuan dalam mengenali karakteristik setiap kelas.

3. Pooling Layer

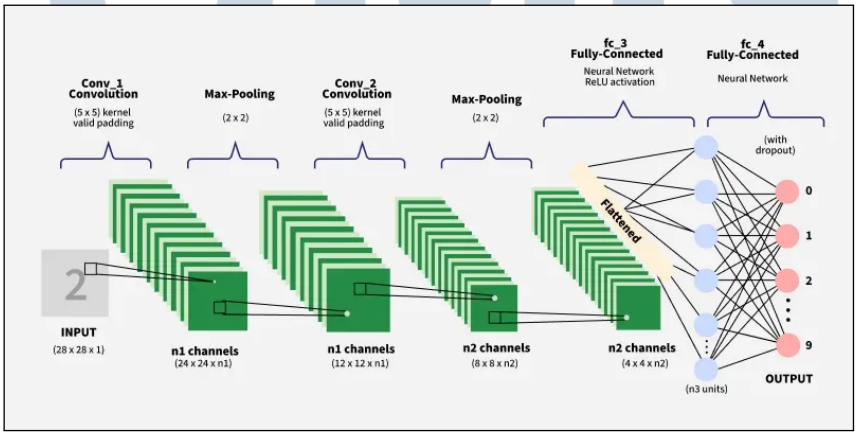
   Setelah proses konvolusi, dilakukan operasi *pooling* untuk memilih fitur-fitur yang paling penting sekaligus mengurangi dimensi data. Pada klasifikasi teks, teknik yang umum digunakan adalah *max pooling*, yaitu memilih nilai maksimum dari setiap *feature map*. Proses ini membantu mempertahankan fitur yang paling representatif sekaligus mengurangi kompleksitas komputasi.

4. Fully Connected Layer

   Fitur-fitur hasil proses *pooling* kemudian digabungkan pada lapisan *fully connected*. Lapisan ini mempelajari hubungan antarfitur sebelum menghasilkan keputusan klasifikasi.

5. Output Layer

   Lapisan keluaran menghasilkan probabilitas untuk setiap kelas menggunakan fungsi aktivasi, seperti *Softmax* pada klasifikasi multikelas atau *Sigmoid* pada klasifikasi biner. Kelas dengan probabilitas tertinggi dipilih sebagai hasil prediksi.



Gambar 2.2. *Architecture for Text Processing*

Sumber: [39]

Pada penelitian ini, CNN tidak digunakan sebagai model utama untuk menghasilkan representasi bahasa, melainkan sebagai lapisan ekstraksi fitur setelah IndoBERT. Representasi kontekstual yang dihasilkan IndoBERT menjadi masukan bagi CNN untuk mengekstraksi pola-pola lokal yang berkaitan dengan ujaran kebencian. Kombinasi tersebut memungkinkan model memanfaatkan kemampuan IndoBERT dalam memahami konteks bahasa sekaligus memanfaatkan kemampuan CNN dalam mengenali pola frasa yang penting untuk proses klasifikasi.

CNN memiliki beberapa keunggulan dalam klasifikasi teks, antara lain:

a. mampu mengekstraksi fitur secara otomatis tanpa memerlukan proses *feature engineering* secara manual;

b. efektif dalam menangkap pola lokal dan fitur *n-gram* pada teks;

c. memiliki jumlah parameter yang relatif lebih sedikit dibandingkan beberapa arsitektur *sequence model*, sehingga proses pelatihan menjadi lebih efisien;

d. memberikan performa yang baik pada berbagai tugas klasifikasi teks, termasuk deteksi ujaran kebencian.

2.8 *Hybrid* IndoBERT CNN

Pendekatan *Hybrid* IndoBERT CNN merupakan integrasi antara dua arsitektur *deep learning*, yaitu IndoBERT dan CNN, dengan tujuan untuk meningkatkan performa klasifikasi teks.

Dengan memanfaatkan mekanisme *self-attention*, model *Transformer* seperti IndoBERT dapat membangun representasi kontekstual yang kaya terhadap suatu kalimat. Kemampuan tersebut memungkinkan model membentuk pemahaman kontekstual terhadap teks dengan memanfaatkan keterkaitan semantik antarkata, sehingga makna yang terkandung dalam teks dapat dipahami secara lebih komprehensif [14]. Namun, IndoBERT memiliki keterbatasan dalam menangkap pola lokal seperti frasa atau *n-gram* tertentu yang sering muncul dalam ujaran kebencian.

Di sisi lain, CNN dikenal efektif dalam mengekstraksi fitur lokal melalui operasi *convolution* dan *pooling*. CNN mampu mengidentifikasi pola linguistik spesifik seperti kata kasar, sindiran, atau frasa diskriminatif yang menjadi ciri khas ujaran kebencian [41].

Dengan mengintegrasikan kedua pendekatan tersebut, model *Hybrid* IndoBERT CNN dapat menggabungkan keunggulan pemahaman konteks global dari IndoBERT dan kemampuan ekstraksi fitur lokal dari CNN. Kombinasi ini diharapkan mampu meningkatkan akurasi serta ketahanan model dalam mendeteksi ujaran kebencian dibandingkan dengan penggunaan model tunggal.

2.9 *Explainable Generative AI*

Explainable Generative AI merupakan pendekatan yang bertujuan meningkatkan kemampuan interpretasi terhadap model kecerdasan buatan dengan menyediakan penjelasan mengenai faktor-faktor yang mendasari proses pengambilan keputusan atau pembentukan prediksi oleh model [42]. Dalam model berbasis *deep learning*, interpretabilitas menjadi tantangan karena kompleksitas arsitektur yang tinggi. Sehingga model sering dianggap sebagai *black box*.

Explainable Generative AI dapat dimanfaatkan sebagai mekanisme untuk meningkatkan interpretabilitas sistem dengan menghasilkan penjelasan dalam bahasa alami, sehingga alasan di balik prediksi atau keputusan model dapat dipahami secara lebih jelas oleh pengguna. Dalam konteks klasifikasi teks, *Explainable Generative* AI dapat menghasilkan penjelasan naratif yang menjelaskan alasan di balik hasil prediksi model [21, 43].

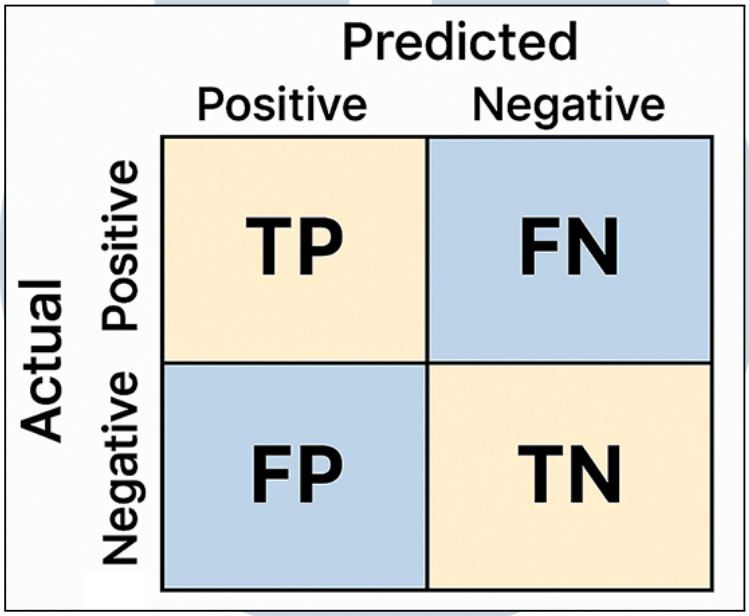
Pendekatan ini memungkinkan sistem tidak hanya memberikan label klasifikasi, seperti ujaran kebencian atau bukan, tetapi juga memberikan penjelasan yang mudah dipahami oleh manusia. Dalam penerapan moderasi konten, transparansi terhadap proses pengambilan keputusan menjadi aspek yang penting karena hasil klasifikasi yang diberikan oleh sistem harus dapat dijelaskan serta dipertanggungjawabkan kepada pengguna maupun pihak terkait.

Dalam penelitian ini, *Explainable Generative* AI digunakan sebagai mekanisme untuk mendukung interpretasi hasil klasifikasi dari model *Hybrid* IndoBERT CNN, sehingga menghasilkan sistem yang tidak hanya memberikan hasil klasifikasi yang akurat, tetapi juga mampu menjelaskan dasar pengambilan keputusan secara transparan kepada pengguna.

2.10 *Confusion Matrix*

Confusion matrix merupakan metode evaluasi performa model klasifikasi yang paling umum digunakan. *Confusion matrix* menyajikan perbandingan antara

hasil prediksi model dan label aktual dalam bentuk tabel dua dimensi. Setiap elemen pada matriks menunjukkan jumlah kasus yang diklasifikasikan ke dalam kombinasi tertentu antara prediksi dan kebenaran aktual, baik untuk kelas positif maupun negatif.



Gambar 2.3. *Confusion Matrix in Machine Learning*
Sumber: [44]

Empat komponen utama dalam *confusion matrix* dapat dijelaskan sebagai berikut:

1. TP (*True Positive*) menyatakan jumlah data pada kelas positif yang berhasil diprediksi dengan benar sebagai kelas positif.
2. TN (*True Negative*) menunjukkan jumlah data pada kelas negatif yang berhasil diklasifikasikan sesuai dengan label sebenarnya.
3. FP (*False Positive*) merupakan jumlah data pada kelas negatif yang secara keliru diprediksi sebagai kelas positif.
4. FN (*False Negative*) menyatakan jumlah data pada kelas positif yang gagal dikenali oleh model sehingga diprediksi sebagai kelas negatif.

Dari keempat nilai ini, sejumlah metrik evaluasi kinerja model dapat dihitung untuk memberikan gambaran lebih komprehensif:

2.10.1 Accuracy

Accuracy merupakan salah satu metrik evaluasi yang paling dasar digunakan untuk menilai kinerja model klasifikasi. Metrik ini menunjukkan proporsi prediksi yang berhasil diklasifikasikan dengan benar dibandingkan dengan seluruh data yang dievaluasi [44]. Nilai *accuracy* memberikan gambaran umum mengenai kemampuan model dalam menghasilkan prediksi yang sesuai dengan label aktual. Secara matematis, dapat dihitung menggunakan persamaan berikut:

Rumus 2.1 menunjukkan cara perhintungan *Precision*.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

Sebagai ilustrasi, diperoleh *confusion matrix* dari suatu model klasifikasi biner yang terdiri atas 40 *True Positive* (TP), 50 *True Negative* (TN), 10 *False Positive* (FP), dan 5 *False Negative* (FN). Nilai tersebut menunjukkan bahwa model mampu mengidentifikasi secara tepat 40 data positif dan 50 data negatif. Sebaliknya, terdapat 10 data negatif yang keliru diprediksi sebagai positif serta 5 data positif yang tidak terdeteksi dan diklasifikasikan sebagai negatif.

Berdasarkan hasil tersebut, langkah selanjutnya adalah menghitung nilai *accuracy*, yaitu rasio antara jumlah prediksi yang benar dan keseluruhan data pengujian. Metrik ini digunakan untuk mengevaluasi tingkat ketepatan model dalam menghasilkan prediksi yang sesuai dengan label sebenarnya. Dengan menyubstitusikan nilai-nilai tersebut ke dalam Persamaan **??**, diperoleh:

$$Accuracy = \frac{40 + 50}{40 + 50 + 10 + 5} = \frac{90}{105} \approx 0.857 \; (85.7\%)$$

Berdasarkan hasil yang diperoleh, model berhasil mencapai tingkat akurasi sebesar 85,7% yang menunjukkan bahwa sebagian besar data berhasil diprediksi sesuai dengan label sebenarnya. Namun, penggunaan metrik *accuracy* sebagai satu-satunya indikator evaluasi kurang tepat pada dataset yang memiliki distribusi kelas tidak seimbang. Pada kondisi tersebut, dominasi jumlah sampel pada salah satu kelas dapat menyebabkan nilai akurasi tetap tinggi meskipun kemampuan model dalam mengidentifikasi kelas minoritas masih rendah. Oleh sebab itu, diperlukan metrik evaluasi lain untuk memberikan gambaran kinerja model yang lebih komprehensif.

2.10.2 Precision

Dalam evaluasi model klasifikasi, *precision* digunakan untuk mengukur seberapa akurat prediksi positif yang dihasilkan oleh model. Metrik ini dihitung berdasarkan perbandingan antara jumlah prediksi positif yang benar dengan seluruh data yang diprediksi sebagai positif [44]. Nilai *precision* sangat relevan pada kasus yang menuntut minimisasi *False Positive*, yaitu kesalahan klasifikasi yang terjadi ketika data negatif teridentifikasi sebagai data positif. Perhitungan *precision* secara matematis dapat dinyatakan melalui persamaan berikut:

Rumus 2.2 menunjukkan cara perhitungan *Precision*.

$$Precision = \frac{TP}{TP + FP} \tag{2.2}$$

Berdasarkan *confusion matrix* yang digunakan sebagai contoh, terdapat 40 *True Positive* dan 10 *False Positive*. Nilai *precision* dihitung berdasarkan rasio antara jumlah data yang berhasil diprediksi positif dengan benar (*True Positive*) terhadap total data yang diprediksi sebagai positif oleh model, yang mencakup *True Positive* dan *False Positive*. Menggunakan Persamaan **??**, diperoleh perhitungan sebagai berikut:

$$Precision = \frac{40}{40 + 10} = \frac{40}{50} = 0.8\ (80\%)$$

Berdasarkan hasil perhitungan, diperoleh nilai *precision* sebesar 80%, yang berarti bahwa 80% dari total prediksi positif yang dihasilkan model merupakan prediksi yang benar. Nilai *precision* yang semakin tinggi menunjukkan semakin kecilnya proporsi kesalahan *False Positive*. Oleh sebab itu, metrik ini digunakan untuk mengukur tingkat ketepatan model dalam mengidentifikasi anggota kelas positif secara akurat.

2.10.3 Recall

Recall merupakan salah satu metrik evaluasi yang digunakan untuk mengukur seberapa baik model mampu mendeteksi seluruh data positif yang tersedia. Metrik ini sangat relevan pada kasus yang berfokus pada pengurangan kesalahan *False Negative*, yaitu ketika data positif tidak berhasil diidentifikasi oleh model [44]. Secara matematis, perhitungan *recall* dapat dinyatakan sebagai berikut:

Rumus 2.3 menunjukkan cara perhitungan *Recall*.

$$Recall = \frac{TP}{TP + FN} \tag{2.3}$$

Berdasarkan *confusion matrix* yang digunakan sebagai contoh, terdapat 40 *True Positive* dan 5 *False Negative*. Nilai *recall* dihitung berdasarkan rasio antara jumlah data positif yang berhasil diidentifikasi dengan benar (*True Positive*) terhadap seluruh data yang sebenarnya termasuk ke dalam kelas positif, yang terdiri atas *True Positive* dan *False Negative*. Dengan Persamaan **??**, nilai *recall* dapat dihitung sebagai berikut:

$$Recall = \frac{40}{40+5} = \frac{40}{45} \approx 0.889\ (88.9\%)$$

Berdasarkan hasil yang diperoleh, model berhasil mendeteksi 88,9% dari keseluruhan data positif yang tersedia. Nilai *recall* yang tinggi menunjukkan bahwa model memiliki kemampuan yang baik dalam mengenali anggota kelas positif serta efektif dalam menekan jumlah kesalahan *False Negative*. Meskipun demikian, peningkatan *recall* pada kondisi tertentu dapat menyebabkan penurunan *precision*, sehingga diperlukan keseimbangan antara kedua metrik tersebut.

2.10.4 F1-Score

F1-score adalah metrik evaluasi yang mengombinasikan *precision* dan *recall* dalam bentuk rata-rata harmonis. Oleh karena itu, *F1-score* sering digunakan pada kasus klasifikasi dengan distribusi kelas yang tidak seimbang maupun ketika kesalahan *False Positive* dan *False Negative* sama-sama perlu diperhatikan [44]. Secara matematis, *F1-score* dapat dihitung menggunakan persamaan berikut:

$$F1Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{2.4}$$

Berdasarkan perhitungan sebelumnya, diperoleh nilai *precision* sebesar 0,8 (80%) dan *recall* sebesar 0,889 (88,9%), yang mengindikasikan bahwa model mampu mengklasifikasikan data positif dengan cukup baik serta memiliki tingkat keberhasilan yang tinggi dalam menangkap sebagian besar data positif yang sebenarnya. Dengan memasukkan nilai tersebut ke dalam Persamaan **??**, diperoleh:

$$F1\ Score = 2 * \frac{0.8 * 0.889}{0.8 + 0.889} = 2 * \frac{0.7112}{1.689} \approx 0.841\ (84.1\%)$$

Hasil perhitungan menunjukkan nilai *F1-score* sebesar 84,1%, yang menggambarkan adanya keseimbangan antara kemampuan model dalam memberikan prediksi positif yang tepat serta kemampuannya dalam mengidentifikasi seluruh data positif yang sebenarnya. Oleh karena itu, *F1-score* sering dijadikan sebagai metrik utama dalam evaluasi dan perbandingan model klasifikasi.

2.11 Dataset

Dataset yang dipakai dalam penelitian ini bersumber dari repositori *GitHub* yang dikembangkan oleh M. O. Ibrohim dan I. Budi [22]. Dataset tersebut disusun untuk mendukung penelitian mengenai deteksi ujaran kebencian dan penggunaan bahasa yang kasar pada platform media sosial berbahasa Indonesia, seiring meningkatnya penyebaran konten yang berpotensi menimbulkan konflik sosial, diskriminasi, maupun tindakan kekerasan di ruang digital.

Dataset ini menerapkan skema anotasi *multi-label* yang mencakup data tentang pernyataan permusuhan dan kata-kata kasar. Tahapan anotasi dilakukan secara bertingkat. Pada tahap pertama, setiap *tweet* dikategorikan ke dalam kelas ujaran kebencian, kata-kata kasar, atau tidak termasuk dalam kedua kategori tersebut. Selanjutnya, pada langkah kedua dilakukan identifikasi lebih lanjut terhadap target, kategori, serta tingkat kebencian pada data yang telah ditandai sebagai ujaran kebencian.

2.12 Penelitian Sebelumnya

Beberapa penelitian terdahulu telah menerapkan berbagai algoritma untuk mendeteksi ujaran kebencian dan bahasa kasar pada media sosial. Validitas *dataset* yang digunakan dalam penelitian ini didukung oleh hasil eksperimen dari berbagai studi terdahulu yang memanfaatkan *dataset* yang sama dalam konteks deteksi ujaran kebencian dan bahasa kasar pada media sosial. Salah satu studi melaporkan bahwa penggunaan model *Support Vector Machine* (SVM) dengan parameter yang telah dioptimalkan menghasilkan akurasi rata-rata sebesar 82,65%, dengan rincian akurasi per kelas yaitu 84,92% untuk kelas ujaran kebencian, 86,60% untuk

kelas bahasa kasar, dan 76,43% untuk tingkat kebencian [45]. Angka tersebut menunjukkan bahwa dataset mampu mendukung proses klasifikasi secara efektif, meskipun performa terbaik masih diperoleh melalui pendekatan berbasis *deep learning*.

Selain itu, eksperimen menggunakan algoritma *Logistic Regression* yang memanfaatkan fitur *word embedding* juga menunjukkan akurasi yang kompetitif. Dari beberapa kali pengujian untuk memperoleh model terbaik, diperoleh akurasi rata-rata sebesar 75,59% dengan komposisi data pelatihan dan pengujian 90:10. Akurasi per kelas menunjukkan nilai yang cukup stabil, yakni 75,86% untuk ujaran kebencian, 80,05% untuk bahasa kasar, dan 70,86% untuk tingkat kebencian [46]. Penelitian lainnya menunjukkan bahwa penggunaan algoritma *Recurrent Neural Network* (RNN) pada dataset ini memberikan hasil yang paling optimal dengan akurasi mencapai 91%, mengungguli model-model konvensional berbasis *machine learning* [47]. Tingginya tingkat akurasi ini menegaskan bahwa data yang digunakan memiliki kualitas anotasi yang baik, distribusi label yang representatif, serta struktur data yang mendukung penerapan berbagai pendekatan algoritma secara fleksibel.

Berdasarkan kondisi tersebut, penelitian ini mengusulkan integrasi model *Hybrid* IndoBERT CNN dengan pendekatan *Explainable Generative AI*. Model *Hybrid* IndoBERT CNN digunakan untuk melakukan klasifikasi ujaran kebencian, sedangkan LIME dimanfaatkan untuk mengidentifikasi kata-kata yang paling berpengaruh terhadap keputusan model. Selanjutnya, hasil interpretasi LIME digunakan sebagai masukan bagi model *Generative Explainable AI* untuk menghasilkan penjelasan dalam bentuk bahasa alami. Dengan demikian, penelitian ini tidak hanya berfokus pada performa klasifikasi, tetapi juga meningkatkan transparansi dan interpretabilitas model melalui pendekatan *Explainable Generative AI*.