

BAB 2

LANDASAN TEORI

Landasan teori ini berfungsi sebagai landasan dalam melakukan analisis permasalahan dan merancang solusi dalam penelitian ini. Secara umum, landasan teori ini mencakup ujaran kebencian (*hate speech*), media sosial X, *Natural Language Processing*, *transformer*, BERT, *indoBERT*, Ensemble Learning, XGBoost, serta penelitian terdahulu mengenai teori tersebut.

2.1 Ujaran Kebencian

Ujaran Kebencian atau *hate speech* merujuk pada ekspresi berbasis teks, gambar, atau simbol yang bertujuan menyebarkan kebencian, diskriminasi, atau permusuhan terhadap individu maupun kelompok berdasarkan atribut seperti ras, agama, etnis, jenis kelamin, atau orientasi seksual [1]. Ekspresi ini kerap memanfaatkan anonimitas platform digital untuk memperluas jangkauan serta intensitas dampaknya [2]. Karakteristik utama mencakup bahasa ofensif, stereotip negatif, dan niat untuk merendahkan atau memicu konflik terhadap targetnya.

Paparan ujaran kebencian menyebabkan gangguan emosional seperti kecemasan dan penurunan harga diri, terutama pada remaja dan kelompok rentan. Dampak lebih luas meliputi polarisasi sosial, perpecahan antar kelompok, hingga potensi eskalasi kekerasan fisik [2]. Oleh karena itu, deteksi ujaran kebencian menjadi krusial untuk mitigasi risiko tersebut, terutama melalui pendekatan berbasis *machine learning* yang memanfaatkan analisis teks otomatis. Teknologi ini mendukung identifikasi konten berbahaya secara efisien di lingkungan digital.

2.2 Media Sosial X

Media Sosial X atau biasa dikenal sebagai Twitter, merupakan platform daring yang memungkinkan pengguna berbagi pesan singkat berjumlah maksimum 280 karakter per postingan [20]. Fitur utama, seperti *retweet*, *hashtag*, dan *reply*, mendukung interaksi cepat antar pengguna, menjadikan platform ini pusat komunikasi global pengguna aktif secara *real-time* [21]. Struktur terbuka memfasilitasi penyebaran informasi secara instan, termasuk konten berupa ujaran kebencian, akibat kemampuan amplifikasi melalui mekanisme jaringan sosial [8].

Media sosial X telah menjadi wadah bagi masyarakat untuk menyuarakan berbagai opini, menjadikannya sebagai ruang digital yang terbuka untuk diskusi publik. Namun, di sisi lain, platform ini juga dimanfaatkan sebagai sarana penyebaran ujaran kebencian, khususnya melalui penggunaan *hashtag* yang memungkinkan konten bermuatan ofensif tersebar luas dalam waktu singkat [22]. Penelitian mencatat rasio ujaran kebencian secara daring di Indonesia meningkat sepuluh kali lipat dalam dua tahun terakhir, dengan kelompok minoritas agama dan etnis menjadi sasaran utama [9].

Tingginya volume interaksi dan akses secara simultan oleh pengguna mempercepat laju penyebaran konten negatif, menjadikannya sulit untuk dikendalikan secara *real time*. Kondisi ini semakin diperparah oleh tingkat anonimitas yang tinggi di platform, yang memungkinkan individu mengekspresikan kebencian tanpa konsekuensi langsung atau hambatan yang berarti [23].

Paparan ujaran kebencian di Media Sosial X memengaruhi individu dan kelompok secara luas. Konten tersebut kerap menggunakan bahasa ofensif serta stereotip negatif untuk menargetkan korban berdasarkan ras, agama, atau etnis [24]. Dampaknya mencakup gangguan emosional seperti kecemasan pada pengguna rentan, polarisasi sosial antar komunitas, hingga potensi eskalasi konflik di dunia nyata [23].

2.3 TF-IDF

Term Frequency-Inverse Document Frequency (TF-IDF) adalah metode statistik yang digunakan untuk mengevaluasi seberapa penting suatu kata dalam sebuah dokumen relatif terhadap kumpulan dokumen lainnya (corpus). Metode ini banyak digunakan dalam pencarian informasi dan penambangan teks untuk merepresentasikan teks ke dalam bentuk numerik yang dapat diproses oleh algoritma pembelajaran mesin [25].

Term Frequency (TF) mengukur seberapa sering sebuah kata muncul dalam sebuah dokumen. Semakin sering kata tersebut muncul, semakin besar nilainya. Dinyatakan sebagai:

$$TF(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \quad (2.1)$$

di mana $f_{t,d}$ adalah jumlah kemunculan kata t dalam dokumen d .

Inverse Document Frequency (IDF) mengukur seberapa unik atau jarang kata tersebut di seluruh dokumen dalam korpus. Dinyatakan sebagai:

$$IDF(t) = \log \left(\frac{N}{df_t} \right) \quad (2.2)$$

dengan N adalah jumlah total dokumen, dan df_t adalah jumlah dokumen yang mengandung kata t .

Nilai akhir dari TF-IDF diperoleh dengan mengalikan nilai TF dan IDF:

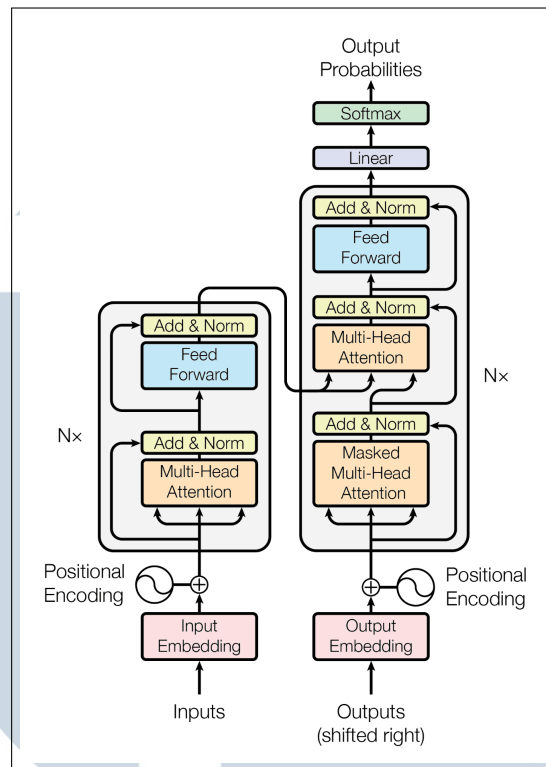
$$TFIDF(t, d) = TF(t, d) \times IDF(t) \quad (2.3)$$

Dengan pendekatan ini, kata-kata umum yang sering muncul di banyak dokumen akan memiliki bobot rendah, sementara kata-kata yang jarang namun relevan dalam konteks dokumen tertentu akan memiliki bobot lebih tinggi.

2.4 Transformer

Transformer merupakan model *neural network* berbasis *sequence-to-sequence* yang sepenuhnya mengandalkan mekanisme *self attention* untuk memproses data urutan, tanpa menggunakan jaringan berulang (*recurrent neural networks*) atau konvolusi (*convolutional neural networks*). Berbeda dengan pendekatan sebelumnya yang memproses teks secara berurutan, *self-attention* memungkinkan *Transformer* menangkap hubungan antar elemen dalam data secara simultan. Model ini menjadi dasar bagi arsitektur modern seperti BERT dan *IndoBERT*, atau model BERT lainnya yang relevan untuk tugas klasifikasi teks seperti deteksi tweet ujaran kebencian pada Platform X.

Arsitektur *Transformer* terdiri dari encoder-decoder dan setiap layer memiliki mekanisme *self-attention* bertumpuk yang dilengkapi dengan *multi head attention* untuk menangkap berbagai aspek hubungan dalam teks. Encoder mengolah data masukan menjadi representasi kontekstual, sedangkan decoder menghasilkan keluaran berdasarkan representasi tersebut, seperti terlihat pada Gambar 2.1.



Gambar 2.1. Arsitektur *transformer*

Sumber: [26]

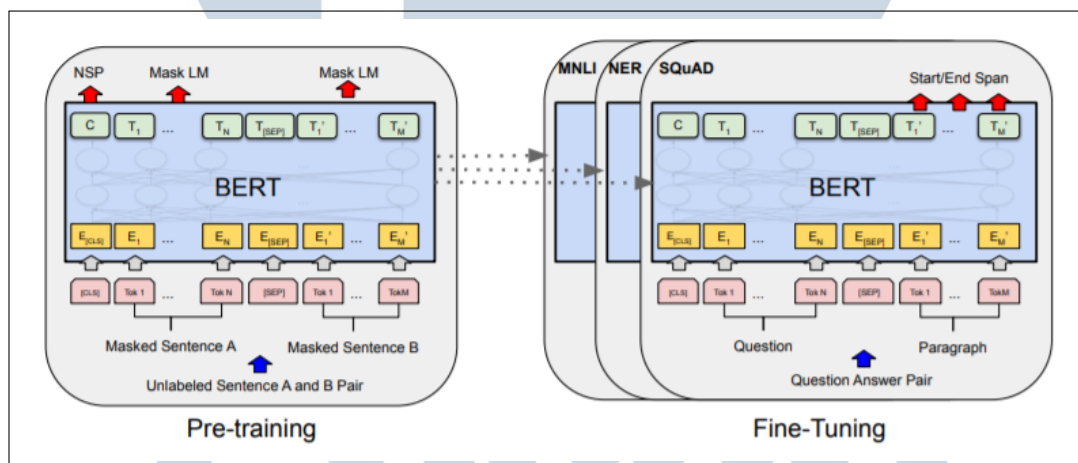
2.4.1 Bidirectional Encoder Representations from Transformers (BERT)

Bidirectional Encoder Representations from Transformers (BERT) adalah model *natural language processing* (NLP) yang dikembangkan untuk memahami konteks kata dalam kalimat secara mendalam. Model ini pertama kali diperkenalkan oleh Devlin et al. (2019) pada Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. BERT unik karena menggunakan pendekatan bidirectional, yang memungkinkan model mempertimbangkan konteks dari arah kiri dan kanan kata dalam kalimat secara bersamaan [11]. Pendekatan ini berbeda dengan model sebelumnya seperti Word2Vec yang bersifat unidirectional [27].

BERT dibangun berdasarkan arsitektur *Transformer*, yang pertama kali diperkenalkan oleh Vaswani et al. (2017) dalam *Advances in Neural Information Processing Systems*. *Transformer* terdiri dari encoder dan decoder, tetapi BERT hanya menggunakan bagian encoder yang terdiri dari lapisan *self attention* dan *feed-forward neural network* [28]. Mekanisme *self attention* memungkinkan BERT untuk menimbang pentingnya setiap kata dalam kalimat terhadap kata lainnya,

menghasilkan representasi vektor yang kontekstual.

Proses kerja BERT melibatkan dua tahap utama: *pretraining* dan *finetuning*, sebagaimana divisualisasikan pada Gambar 2.1. Pada tahap *pre-training*, BERT dilatih dengan dua tugas utama: *Masked Language Model* (MLM) dan *Next Sentence Prediction* (NSP). Dalam MLM, sekitar 15% kata dalam kalimat disembunyikan secara acak, dan model dilatih untuk memprediksi kata-kata tersebut berdasarkan konteks. Sementara itu, NSP melatih model untuk memprediksi apakah dua kalimat berurutan memiliki hubungan logis, seperti ditunjukkan pada bagian kiri Gambar 2.2 dengan "Masked Sentence A" dan "Masked Sentence B". Tahap *finetuning* memungkinkan BERT disesuaikan untuk tugas spesifik, seperti klasifikasi ujaran kebencian dalam penelitian ini, dengan menambahkan lapisan output pada vektor [CLS], sebagaimana digambarkan pada bagian kanan Gambar 2.2.



Gambar 2.2. Arsitektur proses *pretraining* dan *finetuning* pada *bidirectional encoder representations from transformers* (bert)

Sumber: [29]

Keunggulan BERT terletak pada kemampuan *finetuning* untuk berbagai tugas NLP, seperti klasifikasi teks dan deteksi ujaran kebencian. Menurut Mozafari et al. (2020) dalam jurnal PLOS ONE, BERT efektif menangkap nuansa bahasa yang kompleks seperti sarkasme dalam deteksi ujaran kebencian [12]. Representasi *bidirectional* yang dihasilkan oleh BERT, sebagaimana divisualisasikan pada Gambar 2.2, memungkinkan model untuk memahami konteks yang lebih kaya dibandingkan pendekatan tradisional, menjadikannya dasar yang kuat untuk pengembangan varian seperti BERT lainnya.

2.4.2 indoBERT

IndoBERT merupakan model bahasa berbasis arsitektur BERT yang dikembangkan secara khusus untuk pemrosesan bahasa alami (NLP) dalam bahasa Indonesia [30]. Model ini dilatih menggunakan lebih dari 220 juta kata yang dikumpulkan dari berbagai sumber teks berbahasa Indonesia, termasuk artikel Wikipedia, berita daring, dan media sosial. IndoBERT dirancang untuk memberikan representasi linguistik yang lebih akurat dan kontekstual dalam bahasa Indonesia. Kemampuannya yang unggul dalam berbagai tugas NLP, seperti klasifikasi teks, analisis sentimen, dan peringkasan, menjadikan IndoBERT sebagai *base line* baru dalam pengembangan aplikasi NLP berbahasa Indonesia. Secara arsitektural, IndoBERT mengadopsi struktur *BERT-base* dengan 12 lapisan *transformer*, dimensi *hidden state* sebesar 768, serta 12 *attention heads* [30]. Dengan jumlah parameter mencapai sekitar 110 juta, IndoBERT menawarkan keseimbangan antara performa dan efisiensi komputasi.

Sejumlah penelitian telah menunjukkan efektivitas IndoBERT dalam berbagai aplikasi NLP berbahasa Indonesia. Koto et al. [30] menguji performa IndoBERT pada benchmark dataset IndoLEM, yang mencakup berbagai tugas seperti analisis sentimen dan peringkasan teks. Hasilnya menunjukkan bahwa IndoBERT mampu mengungguli model BERT multilingual, dengan peningkatan akurasi hingga 5% pada beberapa tugas. Selanjutnya, Cahyawijaya et al. [31] menunjukkan bahwa IndoBERT dapat diadaptasi untuk tugas-tugas generatif seperti teks otomatisasi, dengan hasil yang kompetitif dibandingkan model sejenis. Selain itu, studi oleh [32] mengeksplorasi penerapan IndoBERT dalam analisis sentimen di media sosial, dan mencatatkan skor F1 diatas 0,85, yang menunjukkan kemampuan model ini dalam menangani teks informal berbahasa Indonesia secara efektif.

2.5 Ensemble Learning

Ensemble learning adalah pendekatan dalam pembelajaran mesin yang menggabungkan beberapa model pembelajaran (*base learners*) untuk menghasilkan prediksi yang lebih akurat, stabil, dan robust dibandingkan model tunggal. Teknik ini bertujuan meminimalkan bias, variance, serta kesalahan prediksi dengan memanfaatkan keunggulan masing-masing model secara bersamaan[33]. Secara umum, terdapat tiga strategi utama dalam ensemble learning, yaitu bagging, boosting, dan stacking. Bagging (*bootstrap aggregating*) bekerja dengan melatih

beberapa model secara paralel pada subset data acak untuk mengurangi variance, seperti yang diterapkan pada algoritma Random Forest [34]. Sebaliknya, boosting melatih model secara berurutan dengan menitikberatkan pada perbaikan kesalahan prediksi model sebelumnya sehingga dapat mengurangi bias dan meningkatkan akurasi, contohnya pada AdaBoost dan XGBoost [35]. Sementara itu, stacking menggabungkan output dari beberapa model dasar menggunakan meta-learner untuk menghasilkan prediksi akhir yang optimal dengan memanfaatkan pola kesalahan antar model [36].

2.5.1 Gradient Boosting

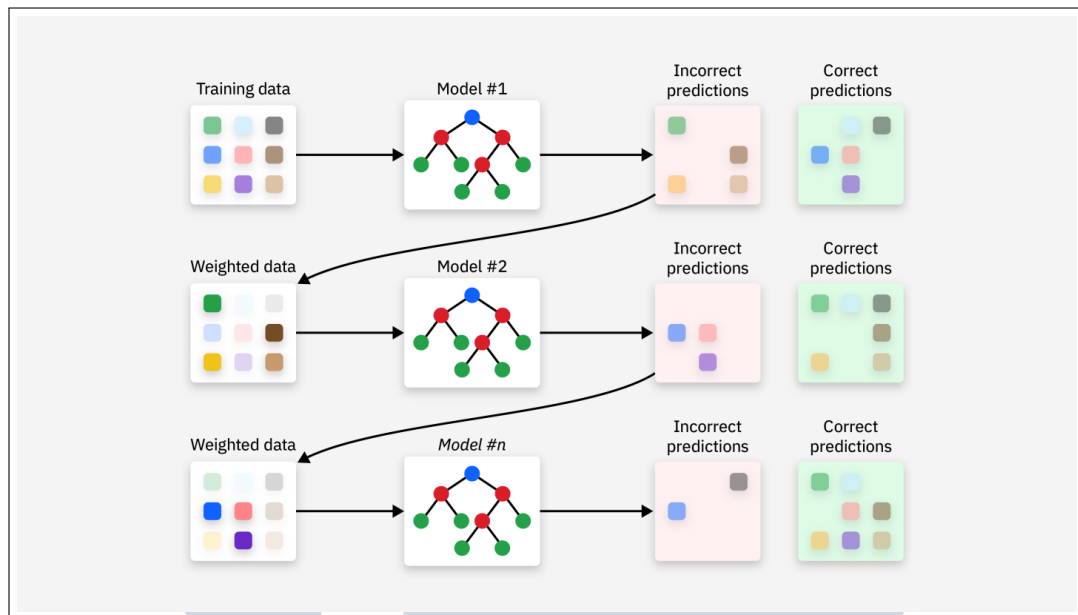
Gradient Boosting adalah metode *ensemble learning* yang digunakan untuk meningkatkan akurasi prediksi dengan menggabungkan sejumlah model sederhana (biasanya berupa pohon keputusan) secara bertahap. Konsep utamanya adalah setiap model baru dibangun untuk memperbaiki kesalahan prediksi dari model sebelumnya dengan menggunakan pendekatan *gradient descent* terhadap fungsi kerugian [37].

Secara sederhana, proses kerja Gradient Boosting dapat diringkas dalam formula berikut:

$$\hat{y}_i = \sum_{m=1}^M h_m(x_i) \quad (2.4)$$

dengan \hat{y}_i adalah hasil prediksi akhir untuk data ke- i , dan $h_m(x)$ adalah pohon keputusan ke- m yang dilatih untuk meminimalkan kesalahan dari prediksi sebelumnya. Model ini berfokus pada data yang sulit diprediksi dengan memberikan bobot lebih besar terhadap kesalahan sebelumnya, sehingga proses pembelajaran menjadi lebih adaptif.

Visualisasi alur kerja Gradient Boosting dapat dilihat pada Gambar 2.3, yang menggambarkan proses pembentukan pohon keputusan secara bertahap, di mana setiap pohon bertugas mengoreksi prediksi dari pohon sebelumnya.



Gambar 2.3. Arsitektur Gradient Boosting

Sumber: [38]

Visualisasi pada Gambar 2.3 menggambarkan alur kerja Gradient Boosting secara bertahap. Proses dimulai dari pelatihan model pertama (*Model 1*) menggunakan data awal. Setelah prediksi dilakukan, data yang salah diklasifikasikan akan dikenali dan diberi bobot lebih besar. Data berbobot ini kemudian digunakan untuk melatih model kedua (*Model 2*), yang fokus memperbaiki kesalahan dari model sebelumnya. Proses ini berlanjut hingga model ke- n , di mana setiap model baru dilatih untuk mengurangi error (*residual*) dari hasil prediksi kumulatif sebelumnya. Akhirnya, semua model digabungkan untuk menghasilkan prediksi akhir yang lebih akurat.

2.5.2 XGBoost

XGBoost (*Extreme Gradient Boosting*) adalah algoritma *machine learning* berbasis *gradient boosting* yang dikembangkan oleh Chen dan Guestrin (2016). Algoritma ini merupakan penyempurnaan dari metode *boosting* tradisional dengan menggabungkan serangkaian pohon keputusan (*decision trees*) secara berurutan, di mana setiap pohon baru dibangun untuk mengoreksi kesalahan prediksi pohon sebelumnya [35]. XGBoost menggunakan pendekatan optimasi *gradient* untuk meminimalkan fungsi kerugian, yang memungkinkannya mencapai performa tinggi pada tugas klasifikasi dan regresi, termasuk dalam konteks deteksi hate speech pada penelitian ini [39].

XGBoost membentuk model prediksi secara aditif, di mana hasil akhir merupakan penjumlahan dari kontribusi seluruh pohon keputusan yang dibangun secara berurutan. Fungsi prediksinya dituliskan sebagai berikut:

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i) \quad (2.5)$$

dengan K adalah jumlah total pohon, $f_k(x_i)$ merupakan prediksi dari pohon ke- k terhadap input x_i , dan \hat{y}_i adalah hasil akhir prediksi untuk data ke- i . Setiap f_k merupakan fungsi dari ruang pohon regresi \mathcal{F} .

Model ini kemudian dioptimalkan dengan meminimalkan fungsi objektif, yang mencakup dua komponen utama, yaitu fungsi kerugian dan fungsi regularisasi:

$$obj(\theta) = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k) \quad (2.6)$$

Komponen $l(y_i, \hat{y}_i)$ berfungsi mengukur selisih antara nilai sebenarnya dan prediksi model, sementara $\Omega(f_k)$ adalah fungsi regulasi yang mengontrol kompleksitas pohon, untuk mencegah overfitting.

Fungsi regulasi tersebut didefinisikan sebagai berikut:

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \quad (2.7)$$

dengan T sebagai jumlah daun (terminal nodes) dalam pohon keputusan, w_j sebagai nilai output dari daun ke- j , γ adalah parameter penalti terhadap jumlah daun, dan λ adalah koefisien regularisasi L2 terhadap besar bobot w_j .

Agar proses optimasi lebih efisien, XGBoost mendekati fungsi kerugian menggunakan ekspansi Taylor orde kedua. Dengan pendekatan ini, fungsi objektif pada iterasi ke- t dapat dihamperi sebagai:

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^n \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t(x_i)^2 \right] + \Omega(f_t) \quad (2.8)$$

di mana g_i dan h_i masing-masing adalah turunan pertama (gradien) dan kedua (Hessian) dari fungsi kerugian terhadap prediksi sebelumnya.

Lebih lanjut, dalam proses pembangunan pohon, XGBoost menentukan

pemisahan (split) terbaik berdasarkan nilai *gain*, yaitu seberapa besar peningkatan akurasi yang diperoleh dari membagi satu simpul menjadi dua. Rumus *gain* dirumuskan sebagai:

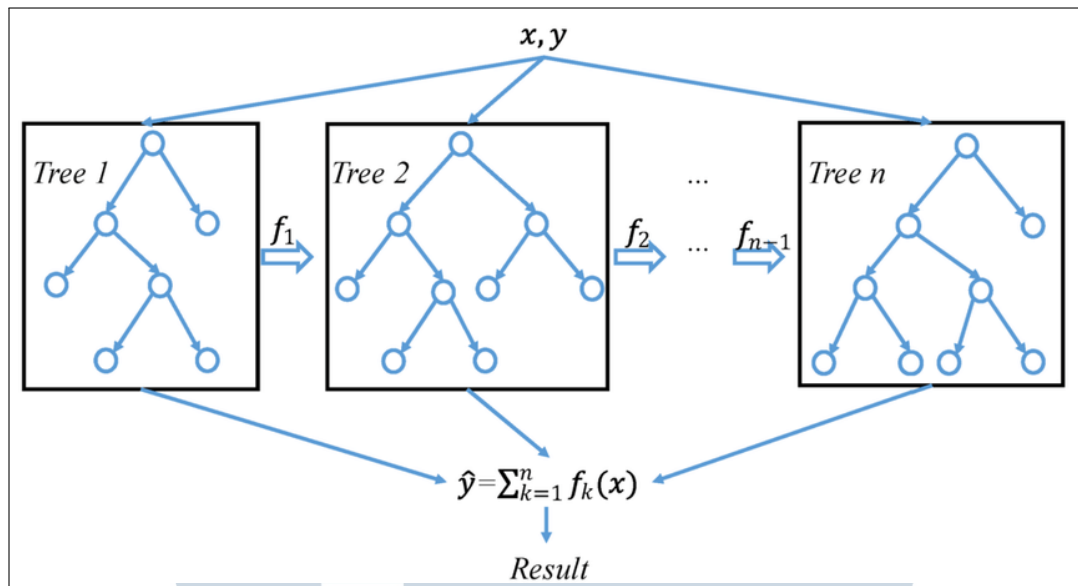
$$Gain = \frac{1}{2} \left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \gamma \quad (2.9)$$

dengan G_L dan G_R adalah jumlah gradien pada simpul anak kiri dan kanan, H_L dan H_R adalah jumlah Hessian-nya, serta λ dan γ sebagai parameter regularisasi. Nilai *gain* yang tidak melebihi γ akan menyebabkan split dibatalkan, guna menghindari pembentukan model yang terlalu kompleks.

Arsitektur XGBoost berbasis pada konsep *ensemble learning*, di mana model akhir merupakan agregasi dari banyak pohon keputusan lemah. Setiap pohon dilatih dengan mempertimbangkan gradien error dari pohon sebelumnya, yang ditingkatkan dengan fitur seperti regularisasi L1 dan L2 untuk mencegah overfitting, serta dukungan untuk data yang hilang [35]. Algoritma ini juga mendukung pelatihan paralel dan skalabilitas pada dataset besar, menjadikannya dapat mengolah fitur teks yang dihasilkan oleh model bahasa lainnya [40].

Proses kerja XGBoost divisualisasikan pada Gambar 2.4, yang menggambarkan alur pembangunan pohon keputusan secara berurutan. Gambar ini menunjukkan bagaimana input data (x, y) diproses melalui beberapa pohon (*Tree 1*, *Tree 2*, ..., *Tree n*), dengan setiap pohon menghasilkan fungsi prediksi (f_1, f_2, \dots, f_n) yang dikoreksi secara bertahap. Prediksi akhir dihitung sebagai penjumlahan skor dari semua pohon ($\hat{y} = \sum f(x)$), yang digunakan untuk klasifikasi teks hate speech berdasarkan fitur dari BERT.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 2.4. Arsitektur XGBoost

Sumber: [41]

Keunggulan XGBoost terletak pada kemampuannya untuk menangani dataset kompleks dan memberikan interpretasi model melalui analisis pentingnya fitur. Selain itu, dalam *Journal of Machine Learning Research* disampaikan bahwa XGBoost unggul dalam kecepatan dan skalabilitas dibandingkan algoritma boosting lain, menjadikannya pilihan yang ideal untuk pengolahan data besar seperti teks media sosial [42].

2.6 Confussion Matrix

Confusion matrix merupakan metode evaluasi performa model klasifikasi yang paling umum digunakan. Confussion 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. Empat komponen utama dalam confusion matrix dapat dijelaskan sebagai berikut:

- **True Positive (TP):** jumlah data positif yang diprediksi benar sebagai positif oleh model.
- **True Negative (TN):** jumlah data negatif yang diprediksi benar sebagai negatif oleh model.

- **False Positive (FP)**: jumlah data negatif yang secara keliru diprediksi sebagai positif.
- **False Negative (FN)**: jumlah data positif yang secara keliru diprediksi sebagai negatif.

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

- **Accuracy**, yaitu proporsi prediksi yang benar dari seluruh jumlah prediksi:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (2.10)$$

- **Precision**, yang mengukur seberapa banyak dari prediksi positif yang benar:

$$Precision = \frac{TP}{TP + FP} \quad (2.11)$$

- **Recall** (atau *Sensitivity*), yang menunjukkan seberapa banyak data positif yang berhasil dikenali oleh model:

$$Recall = \frac{TP}{TP + FN} \quad (2.12)$$

- **F1-Score**, yaitu harmonic mean dari precision dan recall, memberikan keseimbangan antara keduanya:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2.13)$$

Evaluasi berbasis confusion matrix sangat krusial terutama dalam kasus klasifikasi teks yang tidak seimbang, seperti pada deteksi ujaran kebencian. Dengan memahami distribusi prediksi model terhadap masing-masing kelas, confusion matrix dapat menjadi dasar analisis yang kuat untuk perbaikan model lebih lanjut [43].