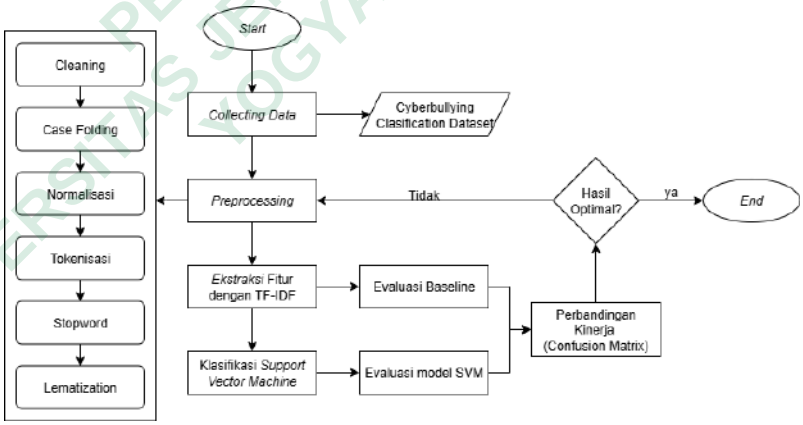
BAB 3
METODE PENELITIAN

Penelitian ini menerapkan pendekatan kuantitatif eksperimental dalam pengembangan serta evaluasi model deteksi *cyberbullying* berbasis metode TF-IDF yang disempurnakan dengan algoritma *Support Vector Machine* (SVM). Tujuan utama penelitian ini yaitu meningkatkan akurasi deteksi *cyberbullying* pada data teks multiclass yang bersumber dari media sosial. Tahapan penelitian dimulai dengan analisis permasalahan *cyberbullying*, dilanjutkan dengan proses pengumpulan dataset teks (komentar atau pesan) yang mengandung indikasi *cyberbullying*. Metode TF-IDF digunakan untuk mengekstrak fitur penting dari teks, sementara algoritma SVM digunakan untuk mengklasifikasikan teks. Hasil analisis ini menjadi dasar dalam mengevaluasi performa model dan mengoptimalkan akurasi deteksi. Penelitian ini diharapkan dapat menghasilkan sistem yang efektif dalam mendeteksi *cyberbullying*. Proses penelitian dijabarkan dalam gambar 3.1 berikut.



Gambar 3.1 Alur Penelitian

Perbandingan kinerja model dilakukan untuk melihat pengaruh penggunaan algoritma klasifikasi terhadap hasil deteksi *cyberbullying*. Perbandingan dilakukan antara model *baseline* yang menggunakan TF-IDF tanpa algoritma klasifikasi dan menggunakan kombinasi TF-IDF dan algoritma *Support Vector Machine* (SVM). Setiap model akan di evaluasi menggunakan *confusion matrix*, dengan fokus pada pengurangan nilai *false positive*. Model yang menunjukkan nilai *precision* dan *F1-score* tertinggi serta *false*

positive rate (FPR) terendah akan dianggap memiliki kinerja terbaik dalam mendeteksi *cyberbullying*.

3.1 Bahan dan Alat Penelitian

Data yang digunakan dalam penelitian ini adalah data sekunder yang diperoleh dari *Cyberbullying Classification Dataset* yang tedapat di *Kaggle.com*. Dataset ini berisi 47.692 *tweet* berbahasa inggris yang telah diberi label berdasarkan kategori *cyberbullying*. Dataset ini dikembangkan sebagai respons terhadap peningkatan kasus *cyberbullying* akibat meningkatnya penggunaan media sosial, terutama selama pandemi COVID-19. Dataset ini digunakan dengan tujuan untuk membantu membangun model deteksi otomatis yang dapat mengidentifikasi konten berbahaya di media sosial serta menganalisis pola bahasa yang digunakan dalam *cyberbullying*.

Dalam penelitian ini, dataset dibagi menjadi dua subset, yaitu 80% data latih dan 20% data uji. Pembagian dilakukan secara acak (*random split*) untuk menjaga distribusi label agar tetap seimbang di kedua subset data. Data latih digunakan untuk membangun model klasifikasi dan data uji untuk menilai performa model dalam mengenali kata-kata yang mengandung unsur *cyberbullying* dengan menggunakan metrik evaluasi.

Untuk mendukung proses analisis data dan pemodelan, penelitian ini memerlukan perangkat dengan spesifikasi yang memadai agar mampu menangani pemrosesan data berdimensi besar secara optimal. Berikut adalah rincian spesifikasinya:

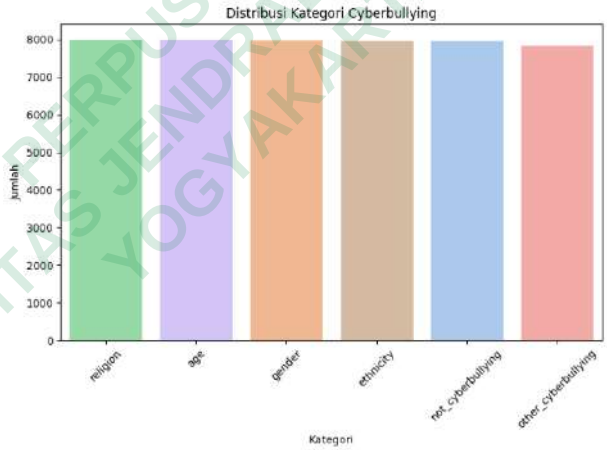
1. Sistem Operasi: Windows 10 pro
2. Prosessor : Intel Core i7 generasi ke-6 (6600)
3. RAM : 16 GB
4. Penyimpanan : SSD 256GB
5. Software : *Python* dan *Google Colaboratory.*

3.2 Jalan Penelitian

Penelitian ini melalui beberapa tahapan sistematis agar proses perancangan, implementasi, dan evaluasi model deteksi *cyberbullying* berjalan terukur. Berikut adalah penjelasan alur penelitian berdasarkan Gambar 3.1 diatas.

3.2.1 Pengumpulan Data

Tahap awal dalam penelitian ini adalah pengumpulan data menggunakan data sekunder berupa komentar berbahasa inggris. Dataset yang digunakan bersumber dari *repository* terbuka *Kaggle* yang diperoleh melalui tautan berikut: https://www.kaggle.com/datasets/andrewmvd/cyberbullying-classification.
Dataset ini mencakup komentar-komentar yang sudah diklasifikasikan ke dalam enam kategori, yaitu age, ethnicity, gender, religion, other *cyberbullying*, dan not *cyberbullying*. Total data yang tersedia berjumlah 47.692 komentar, dengan distribusi data yang relatif seimbang.

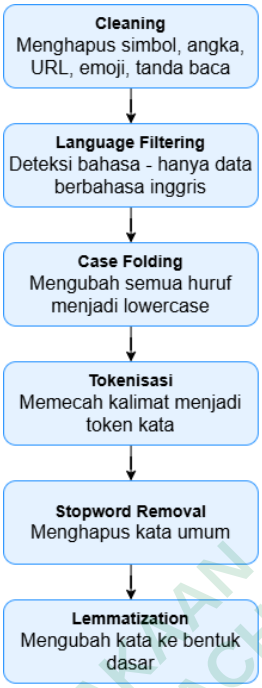


Gambar 3.2 Distribusi Data

Berdasarkan gambar 3.2, dapat dilihat bahwa data telah terdistribusi secara merata pada setiap kelas.

3.2.2 *Pre-processing*

Pre-processing merupakan tahapan krusial untuk memastikan data yang digunakan dalam keadaan bersih, terstruktur, dan konsisten. Tahapan ini sangat penting untuk memperoleh hasil klasifikasi yang optimal serta meningkatkan efisiensi proses pengolahan data.



Gambar 3.3 Tahapan *Pre-processing*

Berdasarkan gambar 3.3 berikut beberapa tahapan *pre-processing* yang diterapkan dalam penelitian ini:

1. *Cleaning*

   Pada tahap ini dilakukan pembersihan data untuk membuang elemen yang tidak diperlukan, misalnya tanda baca, simbol khusus, angka, URL, emoji, atau karakter lain yang tidak relevan. Proses ini bertujuan mengurangi gangguan (noise) dalam data sehingga kata-kata yang penting dapat diolah secara lebih fokus oleh model.

2. *Language Filtering*

   Setelah tahap *cleaning*, data disaring dan hanya menyisakan komentar berbahasa inggris, karena model klasifikasi yang dibangun hanya untuk teks bahasa Inggris, sehingga data berbahasa lain justru akan menurunkan performa dan menimbulkan kesalahan klasifikasi. *Language filtering* memanfaatkan algoritma deteksi bahasa berbasis *n-gram* atau pustaka *langdetect* untuk memilih hanya data berbahasa inggris.

3. *Case Folding*

Tahapan ini bertujuan untuk mengonversi seluruh huruf pada teks menjadi huruf kecil *(lowercase)* untuk menghindari perbedaan representasi kata akibat kapitalisasi. Contohnya, kata *"Hate"* dan *"hate"* diubah menjadi sama agar dianggap satu kata yang seragam.

4. Normalisasi

   Tahapan normalisasi berfungsi menstandarkan kata-kata tidak baku atau bentuk slang menjadi bentuk kata standar. Hal ini penting terutama untuk data media sosial yang sering mengandung penulisan tidak resmi, singkatan, atau istilah gaul. Dengan normalisasi, kata-kata semacam ini diubah ke bentuk yang lebih lazim, sehingga model lebih mudah memahami maknanya.

5. Tokenisasi

   Tokenisasi memecah kalimat menjadi kata-kata *(token)* individu. Tahapan ini memungkinkan setiap kata dianalisis secara terpisah, sekaligus menjadi dasar perhitungan frekuensi kata dalam proses ekstraksi fitur nantinya.

6. *Remove Stopword*

   Stopword merupakan kata-kata yang sering muncul dalam teks, namun tidak memiliki makna signifikan dalam proses klasifikasi, misalnya *the, and, is, at,* dan sebagainya. Menghapus *stopword* dapat mengurangi dimensi data dan membuat model lebih berfokus pada kata yang memiliki nilai informasi yang lebih tinggi.

7. Lematisasi

   Merupakan proses mengembalikan kata ke bentuk lema (bentuk dasar secara gramatikal) menggunakan kaidah bahasa. Contohnya, kata *running, runs*, dan *ran* akan dikembalikan menjadi *run*. Proses ini lebih akurat daripada stemming karena mempertimbangkan konteks linguistik kata, sehingga meningkatkan konsistensi makna pada data.

3.2.3 Ekstraksi Fitur dengan TF-IDF

Setelah teks diproses, langkah selanjutnya adalah *ekstraksi* fitur menggunakan pendekatan TF-IDF. Metode ini berfungsi untuk menentukan tingkat

kepentingan suatu kata dalam sebuah dokumen berdasarkan frekuensinya. Selain itu, metode ini juga membantu memperkuat bobot kata-kata yang lebih relevan dalam mendeteksi *cyberbullying*, sehingga model yang dihasilkan dapat lebih akurat dalam melakukan klasifikasi.

3.2.4 Metode *Baseline Centeroid-Based Classifier*

Metode *baseline* ini digunakan sebagai pembanding kinerja klasifikasi berbasis *machine learning*. Pendekatan ini memanfaatkan representasi fitur TF-IDF, kemudian menghitung kemiripan dokumen uji terhadap rata-rata vektor *(centroid)* setiap kelas menggunakan *cosine similarity*. Langkah-langkah baseline secara umum adalah sebagai berikut:

1. Pembuatan *Centroid* Tiap Kelas

   Setelah vektor TF-IDF diperoleh, dilakukan perhitungan rata-rata vektor dokumen pada setiap kelas sebagai representasi *centroid*. *Centroid* ini merepresentasikan "ciri khas" kata-kata di masing-masing kelas.

2. Pengukuran Kemiripan

   Dokumen uji akan dibandingkan dengan centroid seluruh kelas menggunakan *cosine similarity* yang dihitung dengan persamaan 10 berikut:

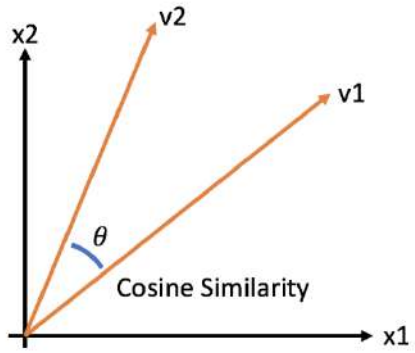
   $$similarity(x, c) = \frac{x.c}{\|x\|.\|c\| + \varepsilon} \quad (10)$$

   dengan $\varepsilon = 10^{-10}$ untuk mencegah pembagian nol.

3. Prediksi

   Dokumen uji diklasifikasikan ke kelas dengan nilai *cosine similarity* tertinggi terhadap *centroidnya*.

Cosine similarity bekerja dengan menghitung sudut antara antar-vektor dalam ruang fitur. Nilai yang mendekati 1 mengindikasikan bahwa dua vektor memiliki tingkat kemiripan yang tinggi, sedangkan nilai yang mendekati 0 menunjukkan kemiripannya rendah atau hampir tidak ada.



Gambar 3.4 Konsep *Cosine similarity*

Secara visual, hubungan antara kedua vektor dapat digambarkan seperti Gambar 3.4 berikut, di mana:

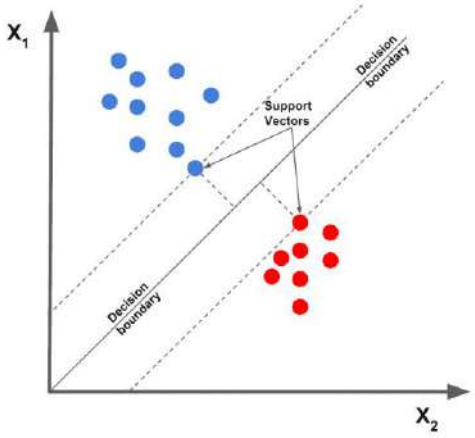
1) sumbu $x1$ dan $x2$ adalah dimensi vektor fitur
2) $v1$ menunjukkan arah vektor komentar
3) $v2$ menunjukkan arah vektor kategori *cyberbullying*
4) sudut θ menunjukkan derajat kesamaan antar-vektor

Proses klasifikasi *baseline* dilakukan dengan menentukan nilai ambang batas *(threshold) cosine similarity*. Jika skor *cosine similarity* melebihi *threshold*, maka komentar diklasifikasikan sebagai *cyberbullying*, sedangkan jika di bawah *threshold* diklasifikasikan sebagai non-*cyberbullying*.

3.2.5 *Support Vector Machine*

Support Vector Machine (SVM) adalah algoritma *supervised learning* yang digunakan untuk menemukan *hyperplane* optimal dalam memisahkan data ke dalam beberapa kelas. *Hyperplane* tersebut dirancang agar memiliki margin maksimal, yaitu jarak terbesar terhadap data terdekat dari titik-titik data terdekat di setiap kelas, yang dikenal sebagai *support vectors*.

Support vectors berperan penting karena menjadi titik acuan pembentukan batas keputusan *(decision boundary)*. Dengan pendekatan ini, SVM mampu mengklasifikasikan komentar ke dalam kategori *cyberbullying* atau *non-cyberbullying*, serta kategori lainnya dalam kategori *multiclass*, secara lebih presisi dibanding *baseline*.



Gambar 3.5 *Visualisasi* SVM

Gambar 3.5 memperlihatkan ilustrasi prinsip kerja SVM, di mana data dua kelas dipisahkan oleh *hyperplane* dengan margin optimal, meminimalkan kesalahan klasifikasi pada data baru.

3.2.6 Evaluasi Model

Evaluasi model dilakukan untuk menilai performa sistem klasifikasi dalam mengenali konten *cyberbullying*. Penilaian dilakukan dengan memanfaatkan metrik berbasis *confusion matrix*, yang mencakup:

1) Akurasi, yaitu proporsi prediksi yang benar terhadap keseluruhan data pengujian.
2) *Precision*, yaitu menunjukkan seberapa tepat model dalam mengidentifikasi data positif secara akurat.
3) *Recall*, yaitu kemampuan model dalam menangkap seluruh data positif yang sebenarnya.
4) *F1-Score*, yaitu rata-rata dari *precision* dan *recall* untuk menyeimbangkan keduanya.

Proses evaluasi dilakukan baik pada *baseline* berbasis *cosine similarity*, maupun pada model klasifikasi SVM, agar dapat dibandingkan secara objektif. Model dengan *precision* dan *F1-score* tertinggi serta *false positive rate* (FPR), yaitu proporsi data negatif yang salah diklasifikasikan sebagai positif, paling rendah akan dipilih sebagai model terbaik untuk mendeteksi komentar *cyberbullying*.