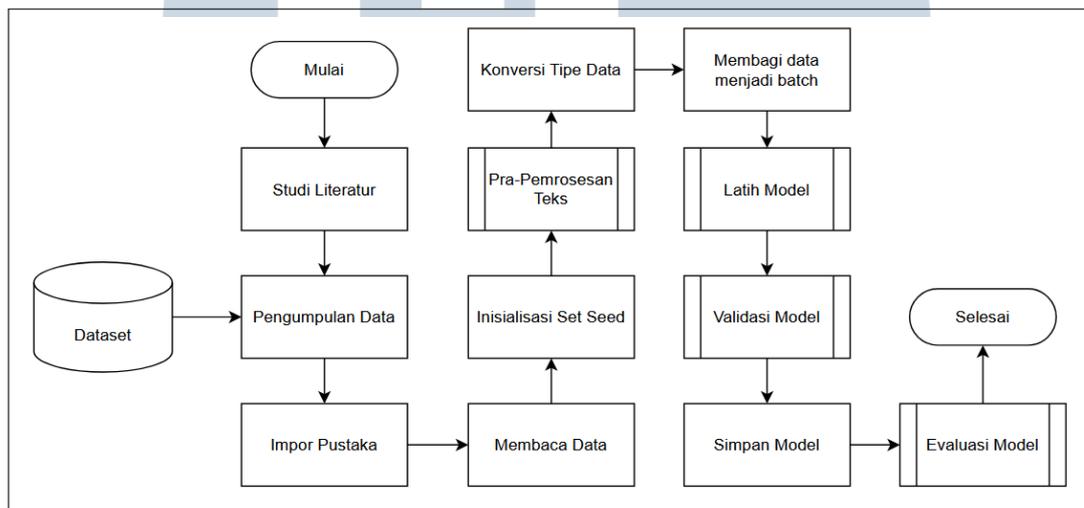


BAB 3 METODOLOGI PENELITIAN

Gambar 3.1 menggambarkan secara menyeluruh urutan tahapan yang dilakukan dalam penelitian ini, mulai dari tahap awal hingga ke proses akhir. Diagram tersebut disusun untuk memberikan gambaran yang jelas mengenai metode yang diterapkan, sekaligus menunjukkan keterkaitan antar setiap langkah dalam rangka mencapai tujuan utama dari penelitian ini.



Gambar 3.1. Flowchart Metodologi Penelitian

3.1 Studi Literatur

Pada tahap studi literatur, dilakukan kajian terhadap penelitian-penelitian yang relevan dengan deteksi teks buatan. Berbagai pendekatan telah dipelajari untuk mengidentifikasi perbedaan antara teks yang dihasilkan oleh manusia dan teks yang dihasilkan oleh model bahasa otomatis. Dalam hal ini, literatur mencakup teknik-teknik berbasis pembelajaran mendalam (deep learning) yang memanfaatkan model-model berbasis transformer, seperti RoBERTa.

3.2 Pengumpulan Data

Dataset yang digunakan dalam penelitian ini adalah *OpenGPTText-Final*, yang diperoleh dari artikel ilmiah berjudul “GPT-Sentinel: Distinguishing Human and ChatGPT Generated Content” [9]. Pemilihan dataset ini didasarkan pada

beberapa pertimbangan penting. Pertama, korpus teks manusia dikompilasi pada tahun 2019, sebelum munculnya model bahasa berskala besar, sehingga kemungkinan kontaminasi oleh teks sintetis dapat diabaikan. Ketiga, jumlah sampel untuk masing-masing kelas (manusia dan AI) disusun secara seimbang, guna mencegah bias selama proses pelatihan dan evaluasi model. Keempat, teks yang digunakan memiliki kesamaan dalam domain dan gaya bahasa, karena teks buatan AI merupakan hasil parafrase dari korpus yang sama. Terakhir, lisensi dataset yang terbuka memungkinkan replikasi eksperimen secara bebas. Kombinasi karakteristik tersebut menjadikan dataset ini sebagai landasan data yang valid dan andal dalam mengevaluasi kinerja model deteksi teks buatan.

3.3 Impor Pustaka

Pada tahap ini, sejumlah pustaka diimpor untuk mendukung seluruh proses pelaksanaan penelitian, mulai dari pemrosesan data, tokenisasi, pelatihan model, hingga evaluasi hasil. Pustaka-pustaka ini menyediakan fungsi-fungsi yang dibutuhkan untuk menjalankan proses klasifikasi teks berbasis model *transformer* secara efisien dan terstruktur.

3.4 Membaca Data

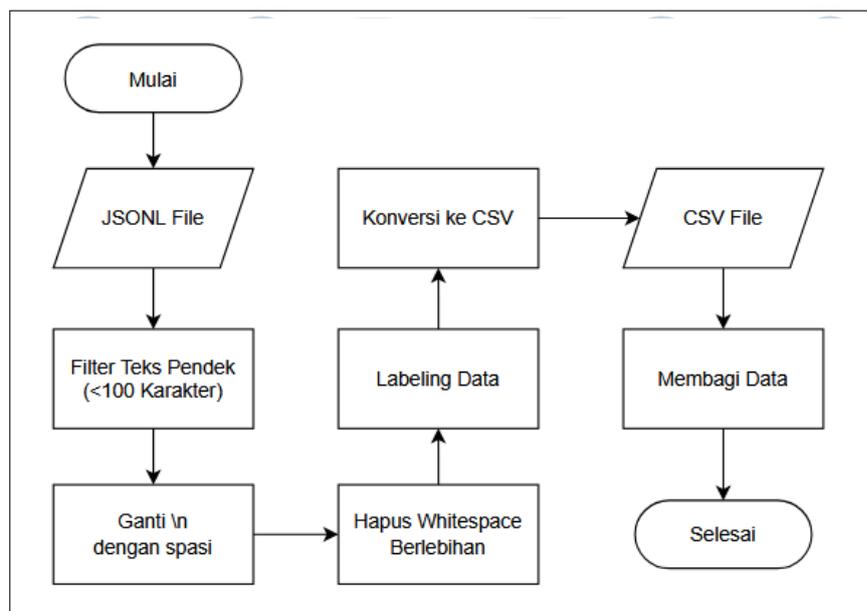
Pada tahap ini, dataset *OpenGPTText-Final* yang telah dikumpulkan dimuat ke dalam program untuk selanjutnya diproses dan digunakan dalam pelatihan model. Proses ini dilakukan dengan membaca file berformat *.csv* yang memuat teks serta label klasifikasinya menggunakan pustaka pemrosesan data. Tahap ini merupakan langkah awal dalam menyiapkan data agar dapat diolah lebih lanjut dalam *pipeline* pelatihan model deteksi teks buatan.

3.5 Inisialisasi Set Seed

Agar hasil eksperimen yang diperoleh bersifat konsisten dan dapat direproduksi, dilakukan inisialisasi nilai *set seed* untuk memastikan semua proses acak menghasilkan keluaran yang sama setiap kali dijalankan. Dalam penelitian ini, nilai *seed* yang digunakan adalah 42, yang diterapkan pada pustaka-pustaka yang relevan seperti *PyTorch* dan *NumPy*.

3.6 Pra-pemrosesan Teks

Gambar 3.2 menggambarkan secara menyeluruh alur pra-pemrosesan teks yang dilakukan dalam penelitian ini. Proses dimulai dengan pembacaan data dari file JSONL. Selanjutnya, dilakukan pemfilteran teks pendek dengan menghapus teks yang memiliki panjang kurang dari 100 karakter untuk memastikan kualitas data yang diproses. Tahap berikutnya adalah penggantian karakter newline ($\backslash n$) dengan spasi untuk menjaga kontinuitas teks. Setelah itu, dilakukan penghapusan whitespace berlebihan untuk menormalkan format teks. Proses dilanjutkan dengan pelabelan data, di mana teks yang berasal dari ChatGPT diberi label `is_ai_generated = 1`, sedangkan teks dari OpenWeb diberi label `is_ai_generated = 0`. Kemudian, data yang telah diproses dikonversi ke dalam format CSV agar siap digunakan dalam tahap pelatihan model. Dalam penelitian ini, digunakan total 10.000 data latih dan validasi, yang terdiri dari 5.000 sampel teks buatan (ChatGPT) dan 5.000 sampel teks manusia (OpenWeb). Dataset ini dibagi dengan rasio 90:10, yaitu 9.000 data (4.500 AI dan 4.500 manusia) untuk pelatihan, dan 1.000 data (500 AI dan 500 manusia) untuk validasi. Penggunaan rasio 90:10 dapat meningkatkan tingkat akurasi dari sebuah model [41, 42]. Selain itu, disiapkan pula dataset uji terpisah sebanyak 2.000 sampel, yang terdiri dari 1.000 teks buatan dan 1.000 teks manusia, yang digunakan untuk mengevaluasi performa akhir dari model secara objektif.



Gambar 3.2. Flowchart Pra-pemrosesan teks

3.7 Konversi Tipe Data

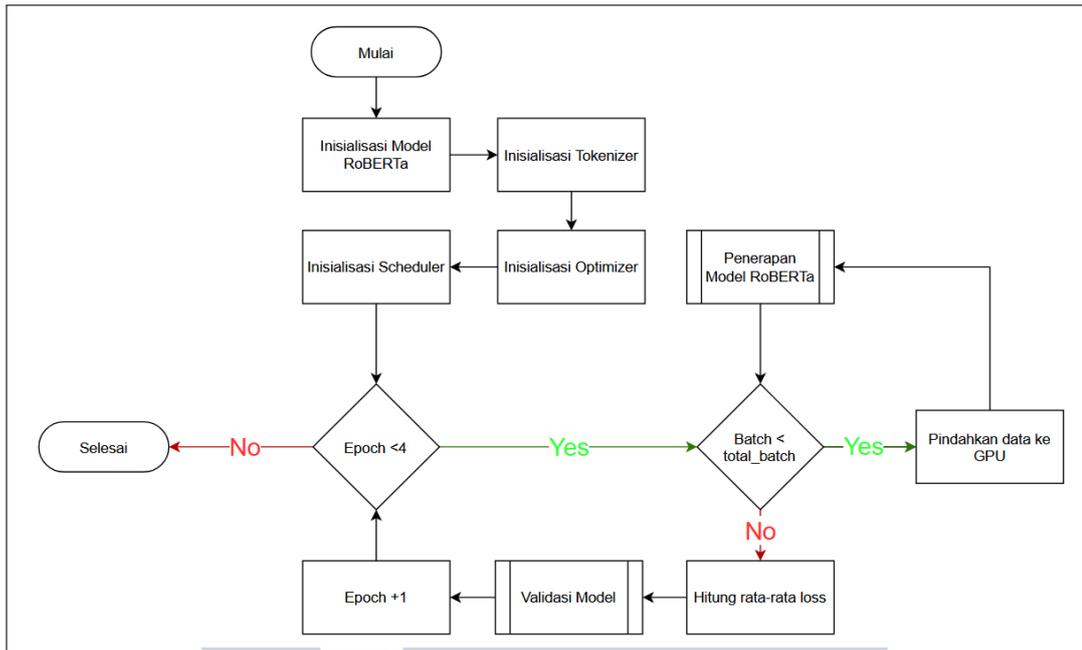
Konversi tipe data merupakan tahapan penting dalam persiapan data untuk pemrosesan oleh model RoBERTa. Pada tahap ini, data teks dan label yang telah melalui proses pra-pemrosesan diubah ke dalam format tensor yang dapat diproses oleh model pembelajaran mendalam. Proses konversi ini melibatkan tokenisasi teks menggunakan tokenizer khusus RoBERTa yang mengubah teks menjadi representasi numerik. Setiap teks diproses dengan menambahkan token khusus, melakukan padding hingga panjang maksimum yang ditentukan (512 token), dan menghasilkan mask perhatian yang menandai posisi token yang valid. Hasil akhir dari proses ini adalah dataset yang berisi input ID, attention mask, dan label dalam format tensor yang siap digunakan untuk pelatihan model.

3.8 Membagi Data Menjadi Batch

Pembagian data menjadi batch merupakan strategi penting untuk mengoptimalkan proses pelatihan model pembelajaran mendalam. Dalam penelitian ini, dataset yang telah dikonversi ke format tensor dibagi menjadi kelompok-kelompok kecil (batch) dengan ukuran yang konsisten. Pendekatan ini memungkinkan model untuk memproses sejumlah kecil data pada satu waktu, sehingga meningkatkan efisiensi komputasi dan penggunaan memori. Dengan ukuran batch yang tepat, proses pelatihan dapat berjalan lebih cepat dan stabil. Selain itu, pembagian data secara acak ke dalam batch-batch pelatihan membantu model untuk generalisasi yang lebih baik dengan menghindari bias urutan data. Untuk data validasi, batch dibuat tanpa pengacakan untuk memastikan konsistensi dalam evaluasi performa model.

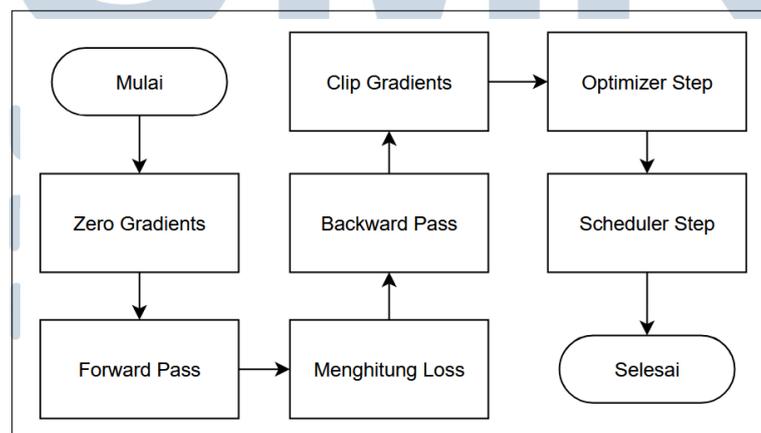
3.9 Latih Model

Gambar 3.3 menggambarkan alur metodologi pelatihan model yang diimplementasikan dalam penelitian ini. Proses dimulai dengan inisialisasi model *RoBERTa*, *tokenizer*, *optimizer*, dan *scheduler*. Setelah inisialisasi, proses pelatihan berlangsung dalam *loop epoch* dengan batasan maksimum 4 *epoch*. Untuk setiap *epoch*, data diproses dalam *batch*, dan setelah semua *batch* selesai diproses, dihitung rata-rata *training loss* serta dievaluasi menggunakan data validasi untuk memantau performa model. Proses ini berulang hingga seluruh *epoch* selesai.



Gambar 3.3. Flowchart Latih Model

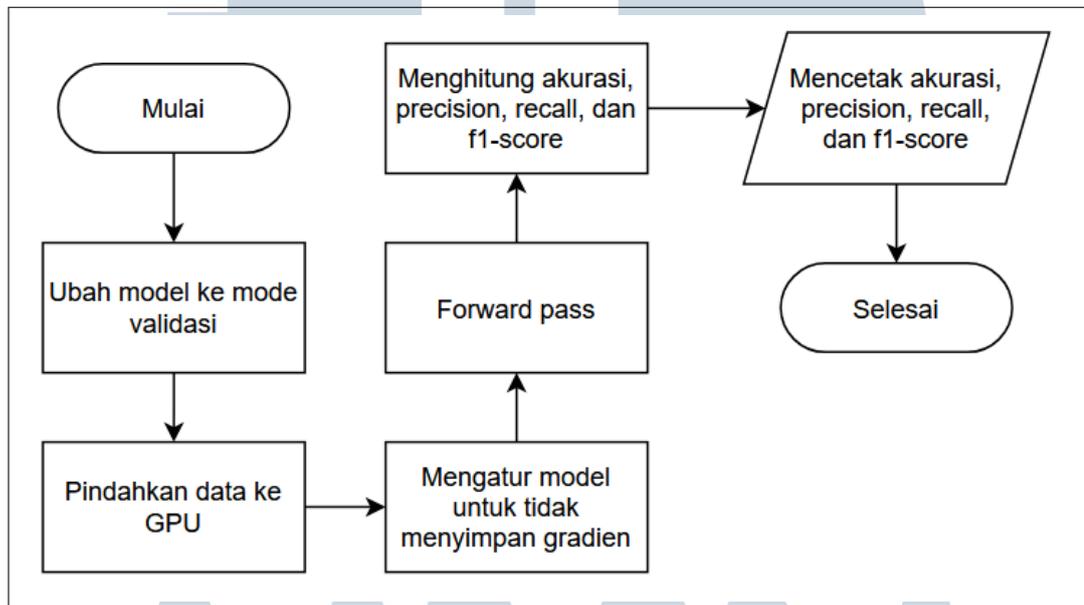
Pada setiap iterasi *batch*, dilakukan serangkaian tahapan pelatihan yang lebih mendetail sebagaimana ditunjukkan pada Gambar 3.4. Proses ini dimulai dengan *zero gradients* untuk menghapus akumulasi gradien dari iterasi sebelumnya. Kemudian dilakukan *forward pass* untuk memperoleh prediksi dari model, diikuti oleh penghitungan *loss* antara prediksi dan label sebenarnya. Langkah selanjutnya adalah *backward pass* untuk menghitung gradien. Untuk mencegah *exploding gradients*, diterapkan teknik *gradient clipping* guna membatasi nilai gradien maksimum. Setelah itu, parameter model diperbarui melalui *optimizer step* dan *learning rate* disesuaikan dengan *scheduler* yang telah ditentukan.



Gambar 3.4. Flowchart Penerapan Model RoBERTa

3.10 Validasi Model

Gambar 3.5 menggambarkan alur validasi model yang dilakukan setelah setiap epoch pelatihan. Pada tahap ini, model diubah ke dalam mode evaluasi dan menerima data validasi yang telah diproses sebelumnya. Proses validasi serupa dengan pelatihan, namun tanpa perhitungan gradien serta tanpa menjalankan proses optimisasi dan penyesuaian melalui *scheduler*, karena model hanya melakukan prediksi untuk mengukur performanya pada data yang tidak dilatih.



Gambar 3.5. Flowchart Validasi Model

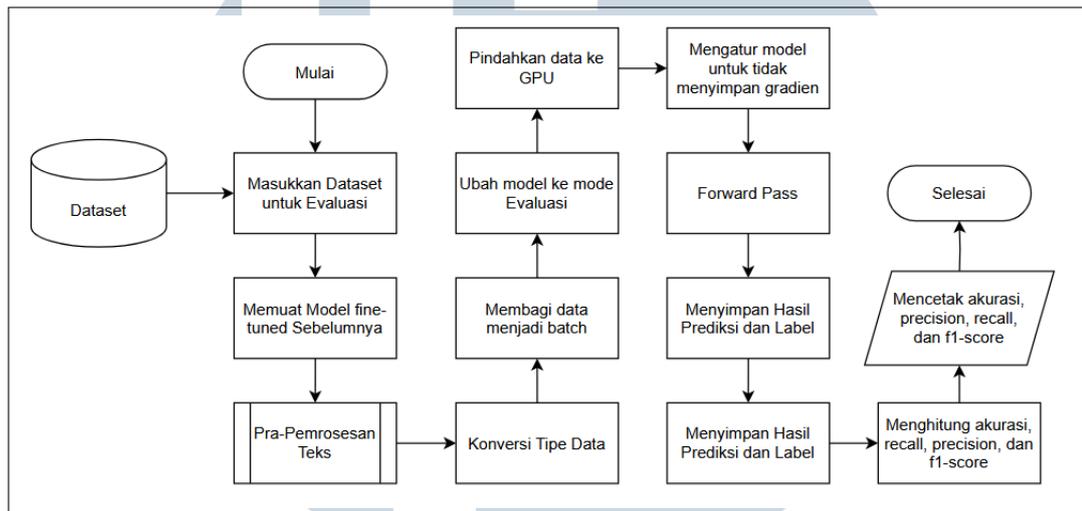
3.11 Simpan Model

Pada tahap ini, model yang telah selesai dilatih dan diverifikasi performanya akan disimpan ke dalam format file agar dapat digunakan kembali tanpa perlu melalui proses pelatihan ulang. Proses penyimpanan ini mencakup penyimpanan parameter dari model serta *tokenizer* yang digunakan, sehingga pada saat proses inferensi atau evaluasi lanjutan, model dapat langsung dimuat dengan konfigurasi yang sama seperti saat pelatihan.

3.12 Evaluasi Model

Gambar 3.6 menunjukkan alur evaluasi model yang dilakukan setelah proses pelatihan selesai. Proses ini serupa dengan tahap validasi, namun menggunakan

dataset uji yang terpisah. Dataset tersebut terlebih dahulu melalui tahapan pra-pemrosesan teks sebelum dievaluasi oleh model yang telah dilatih. Hasil prediksi kemudian dianalisis menggunakan empat metrik evaluasi, yaitu akurasi, presisi, *recall*, dan *f1-score*, untuk mengukur kinerja model secara menyeluruh dengan bantuan pustaka *scikit-learn*.



Gambar 3.6. Flowchart Evaluasi Model

3.13 Dokumentasi

Dokumentasi penelitian ini dilakukan melalui proses penulisan laporan akhir menggunakan $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, yaitu sistem typesetting yang banyak digunakan dalam penulisan karya ilmiah dan akademik. Penggunaan $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ memberikan keunggulan dalam pengaturan tata letak dokumen, manajemen referensi, penulisan rumus matematis, serta konsistensi format secara keseluruhan. Seluruh struktur laporan mulai dari pendahuluan, kajian pustaka, metodologi, hingga hasil dan kesimpulan disusun dan dirapikan dengan bantuan editor $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ seperti Overleaf.

UNIVERSITAS
MULTIMEDIA
NUSANTARA