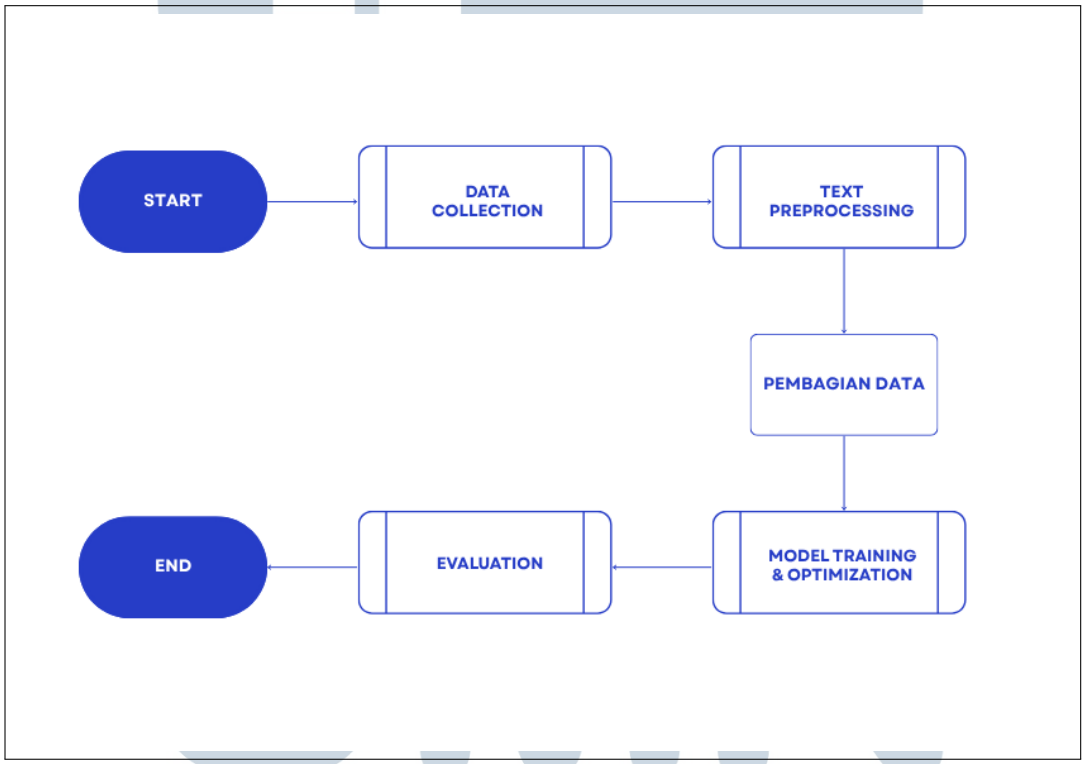
BAB 3
METODOLOGI PENELITIAN

Bab ini menjelaskan secara rinci metodologi yang digunakan dalam penelitian untuk mendeteksi konten seksis pada komentar TikTok berbahasa Indonesia. Setiap tahapan dijelaskan secara sistematis, dimulai dari proses pengumpulan data hingga pelatihan dan evaluasi model. Untuk memberikan gambaran menyeluruh mengenai alur kerja penelitian, berikut disajikan diagram metodologi penelitian yang mencakup seluruh proses yang dilakukan.



Gambar 3.1. Diagram Alur Metodologi Penelitian

Diagram pada Gambar 3.1 memperlihatkan lima tahapan utama dalam metodologi penelitian ini. Penelitian diawali dengan tahap *Data Collection*, yaitu proses pengumpulan komentar dari platform TikTok yang ditulis dalam Bahasa Indonesia. Setelah data terkumpul, dilakukan proses *labeling* untuk menentukan apakah masing-masing komentar tergolong seksis atau tidak, berdasarkan kriteria tertentu yang telah ditetapkan.

Tahap selanjutnya adalah *Text Preprocessing*, yang mencakup serangkaian proses pembersihan dan normalisasi teks, seperti konversi huruf menjadi huruf

kecil, penghapusan simbol atau karakter tidak penting, serta normalisasi kata-kata tidak baku. Tujuan dari tahap ini adalah untuk memastikan bahwa data siap digunakan dalam proses pelatihan model.

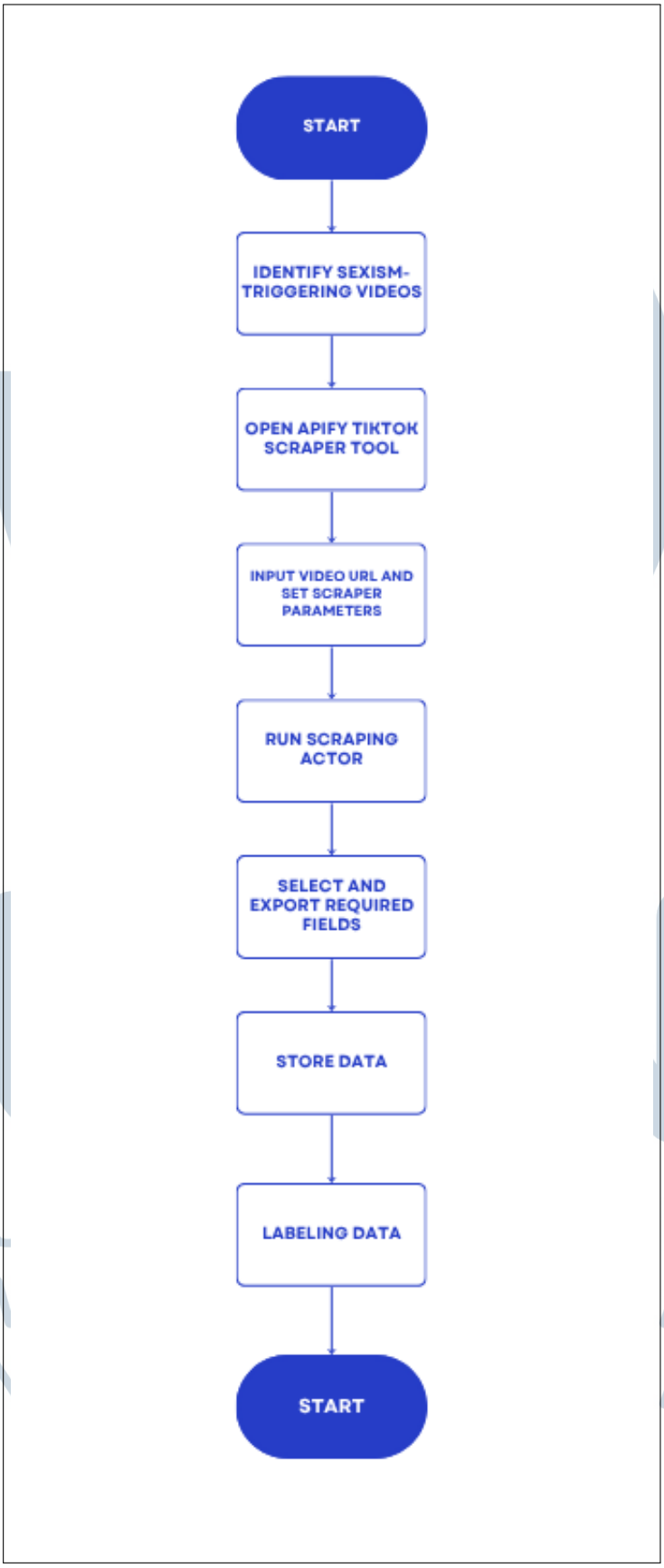
Setelah data dibersihkan, tahap berikutnya adalah Pembagian Data. Pada tahap ini, data dibagi ke dalam subset pelatihan dan pengujian, serta dikonversi ke format yang dapat dibaca oleh model berbasis transformer, seperti melalui proses tokenisasi dan pengubahan menjadi input IDs.

Tahap keempat adalah *Model Training and Optimization*. Pada tahap ini dilakukan dua proses utama. Pertama, dilakukan *cross-validation* dan *hyperparameter tuning* untuk mengevaluasi performa berbagai konfigurasi model dan memilih parameter terbaik. Kedua, model kemudian dilatih ulang menggunakan seluruh data pelatihan (final training) dengan parameter terbaik yang telah diperoleh sebelumnya.

Tahap terakhir adalah *Evaluation*, yaitu pengujian performa model pada data uji menggunakan metrik evaluasi seperti akurasi, presisi, recall, dan F1-score untuk mengetahui seberapa baik model mampu mengklasifikasikan komentar sebagai seksis atau tidak.

3.1 Data Collection

Tahap awal dalam penelitian ini adalah proses *data collection*, yaitu pengumpulan komentar dari platform TikTok dalam Bahasa Indonesia. Komentar-komentar ini diambil dari video yang mengandung potensi memicu perdebatan atau diskusi yang berkaitan dengan stereotip gender, relasi sosial, representasi perempuan, atau isu-isu yang memungkinkan terjadinya komentar bernuansa seksis.



Gambar 3.2. Flowchart Pengumpulan Data

Berdasarkan Gambar 3.2, proses pengumpulan data dilakukan melalui beberapa langkah sebagai berikut:

1. *Identify sexism-triggering videos*
   Langkah pertama adalah mengidentifikasi video TikTok yang berpotensi memicu komentar seksis. Video-video tersebut umumnya memiliki konten yang menyentuh isu gender secara langsung atau secara tidak langsung menyinggung peran sosial, stereotip perempuan, relasi romantis, atau respons terhadap narasi umum tentang perempuan.

2. *Open Apify TikTok scraper tool*
   Setelah video ditentukan, platform *Apify* digunakan sebagai alat bantu untuk melakukan *web scraping*. Apify menyediakan *actor* khusus untuk mengambil data komentar dari TikTok secara otomatis.

3. *Input video URL and set scraper parameters*
   URL video TikTok dimasukkan ke dalam form konfigurasi Apify, kemudian dilakukan pengaturan parameter seperti jumlah maksimum komentar per video dan jumlah maksimum balasan per komentar.

4. *Run scraping actor*
   Actor dijalankan untuk memulai proses *scraping*. Proses ini akan secara otomatis mengambil data komentar dari video sesuai konfigurasi yang telah ditetapkan.

5. *Select and export required fields*
   Setelah proses *scraping* selesai, dilakukan seleksi terhadap variabel yang ingin disimpan, seperti `text`, `id`, `webVideoUrl`, dan `createTimeIso`.

6. *Store data*
   Data komentar yang telah dipilih kemudian diekspor dan disimpan dalam format Excel (`.xlsx`). Dataset ini menjadi korpus utama untuk proses pelabelan dan analisis selanjutnya.

7. *Labeling data*
   Komentar-komentar yang telah berhasil dikumpulkan kemudian dilabeli secara manual untuk menentukan apakah termasuk dalam kategori *seksis* atau *tidak seksis*. Proses pelabelan ini dilakukan dengan mempertimbangkan konteks komentar serta keterkaitannya dengan isi video yang dikomentari.

Pelabelan data dilakukan oleh dua anotator yang sebelumnya telah diberikan briefing mengenai teori seksisme untuk memastikan kesamaan pemahaman terhadap kriteria pelabelan. Hasil pelabelan dari kedua anotator tersebut kemudian divalidasi menggunakan Cohen's Kappa guna mengukur tingkat kesepakatan antar anotator secara statistik, sehingga memastikan reliabilitas proses anotasi.

Namun, untuk keperluan klasifikasi dalam penelitian ini, label akhir tetap dibagi secara biner sebagai berikut:

(a) `SEKSIS`: komentar yang mengandung unsur objektifikasi, pelecehan, stereotip gender, dominasi maskulin, atau bentuk-bentuk misogini lainnya.

(b) `TIDAK SEKSIS`: komentar yang tidak mengandung unsur-unsur tersebut dan bersifat netral atau tidak berkaitan dengan isu gender.

Label akhir disimpan dalam kolom `label` dalam format file `.xlsx`, menggunakan format teks `SEKSIS` dan `TIDAK SEKSIS`.

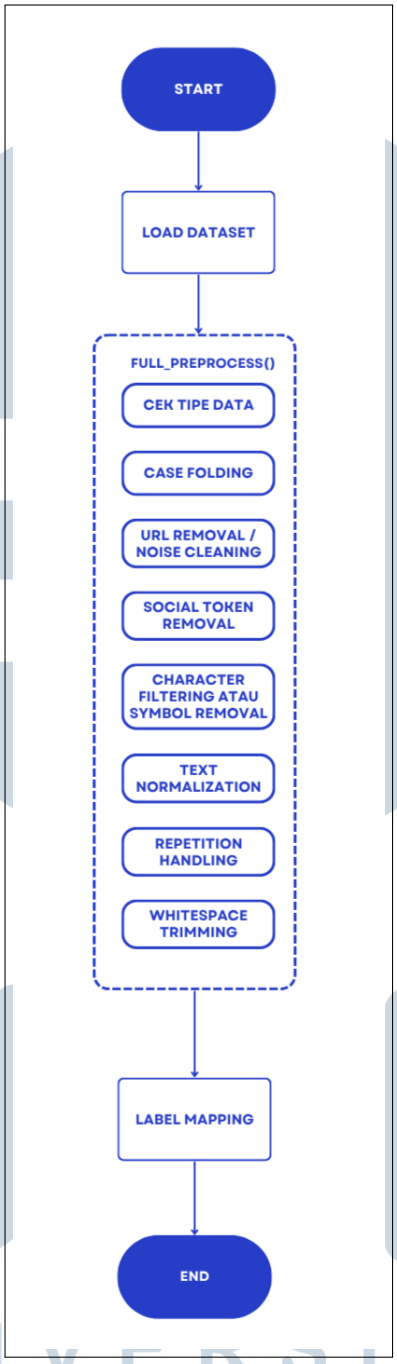
Dataset akhir yang diperoleh terdiri atas 1019 komentar unik yang telah disimpan secara sistematis dan siap digunakan dalam tahap *text preprocessing* dan pelatihan model.

.

3.2 Text Preprocessing

Sebelum data dapat digunakan dalam proses pelatihan model, dilakukan tahapan *text preprocessing* untuk membersihkan dan menormalisasi data teks agar sesuai dengan kebutuhan pemrosesan bahasa alami. Teks dari media sosial seperti TikTok sering kali mengandung bentuk penulisan informal, simbol-simbol tidak relevan, serta variasi dalam penggunaan bahasa, sehingga proses ini penting untuk meningkatkan kualitas input model.

Gambar 3.3 memperlihatkan alur tahapan *preprocessing* yang digunakan dalam penelitian ini.



Gambar 3.3. Diagram Alur Text Preprocessing

Proses diawali dengan memuat dataset komentar dalam Bahasa Indonesia yang telah melalui tahap pelabelan sebelumnya. Setelah dataset dimuat, dilakukan serangkaian transformasi teks melalui fungsi `full_preprocess()` yang mencakup delapan langkah utama berikut:

1. *Type checking* Sebelum diproses, setiap masukan diperiksa untuk

memastikan bahwa data berupa string. Jika bukan string (misalnya `NaN` atau nilai numerik), data tersebut akan diabaikan atau dikembalikan sebagai string kosong.

2. *Case folding*
   Semua huruf dalam teks diubah menjadi huruf kecil (lowercase) untuk menghindari redundansi makna akibat perbedaan kapitalisasi. Contohnya, kata `\Perempuan"` dan `\perempuan"` akan dianggap sama.

3. *Noise removal and social token cleaning*
   Langkah ini menghapus elemen-elemen yang tidak memiliki nilai semantik dalam konteks klasifikasi, seperti URL, mention (`@username`), dan hashtag (`#tag`) yang umum ditemukan pada platform media sosial.

4. *Character filtering*
   Proses ini menghapus karakter non-alfabet seperti angka, tanda baca, emoji, dan simbol khusus, dengan hanya menyisakan huruf dan spasi. Tujuannya adalah menyederhanakan masukan dan mencegah gangguan dalam pemrosesan token.

5. *Text normalization*
   Teks dinormalisasi dengan mengubah kata-kata tidak baku, singkatan, atau kata slang menjadi bentuk formal dalam Bahasa Indonesia menggunakan kamus normalisasi. Contohnya: `\gk"` menjadi `\tidak"`, `\cewe"` menjadi `\perempuan"`.

6. *Repetition handling*
   Huruf yang diulang lebih dari dua kali secara berurutan disederhanakan menjadi dua huruf saja, untuk menghindari kata-kata seperti `\cantiiikkkkk"` yang tidak dikenali oleh model. Contohnya: `\huuuuftttt"` menjadi `\huuftt"`.

7. *Whitespace trimming and normalization*
   Spasi berlebih dan baris kosong dihapus dengan cara menggabungkan spasi berulang menjadi satu spasi, serta menghapus spasi di awal dan akhir teks. Ini bertujuan untuk memastikan teks memiliki format yang bersih dan konsisten.

Setelah proses pembersihan dan normalisasi selesai, dilakukan pemetaan label (*label mapping*). Label awal yang berupa string, yaitu `SEKSIS` dan `TIDAK`

`SEKSIS`, dikonversi ke dalam bentuk numerik untuk keperluan pelatihan model. Nilai `SEKSIS` dipetakan menjadi `1`, dan `TIDAK SEKSIS` menjadi `0`. Hasil dari proses ini disimpan dalam kolom baru bernama `label_numeric`.

Baris-baris data yang tidak memiliki label yang valid akan dibuang dari dataset, dan kolom `label_numeric` dikonversi ke tipe data numerik `integer`. Seluruh proses ini membentuk tahapan *text preprocessing* yang siap untuk digunakan pada tahap berikutnya, yaitu Pembagian Data.

3.3 Pembagian Data

Tahap Pembagian Data merupakan proses lanjutan setelah seluruh data komentar melalui tahapan *text preprocessing*. Pada tahap ini, dataset yang telah dibersihkan dipersiapkan untuk keperluan pelatihan dan evaluasi model.

Langkah utama dalam proses ini adalah pembagian dataset menjadi dua bagian, yaitu *data train* dan *data testing*. *Data train* digunakan selama proses pelatihan model, termasuk *cross-validation* dan *hyperparameter tuning*, sementara *data testing* dialokasikan secara terpisah untuk mengukur performa akhir model secara objektif terhadap data yang tidak pernah dilihat sebelumnya.

Untuk menjaga keseimbangan label pada masing-masing subset data, pembagian dilakukan menggunakan teknik *stratified splitting*, yaitu metode pembagian data yang mempertahankan proporsi distribusi kelas pada keseluruhan dataset. Hal ini penting untuk mencegah bias model terhadap kelas mayoritas dan memastikan bahwa performa model yang dievaluasi mencerminkan kemampuan generalisasi yang sebenarnya.

Dengan demikian, tahap Pembagian Data bertujuan untuk menyediakan struktur data yang siap digunakan dalam proses pelatihan dan evaluasi model secara terpisah dan adil.

3.4 Model Training and Optimization

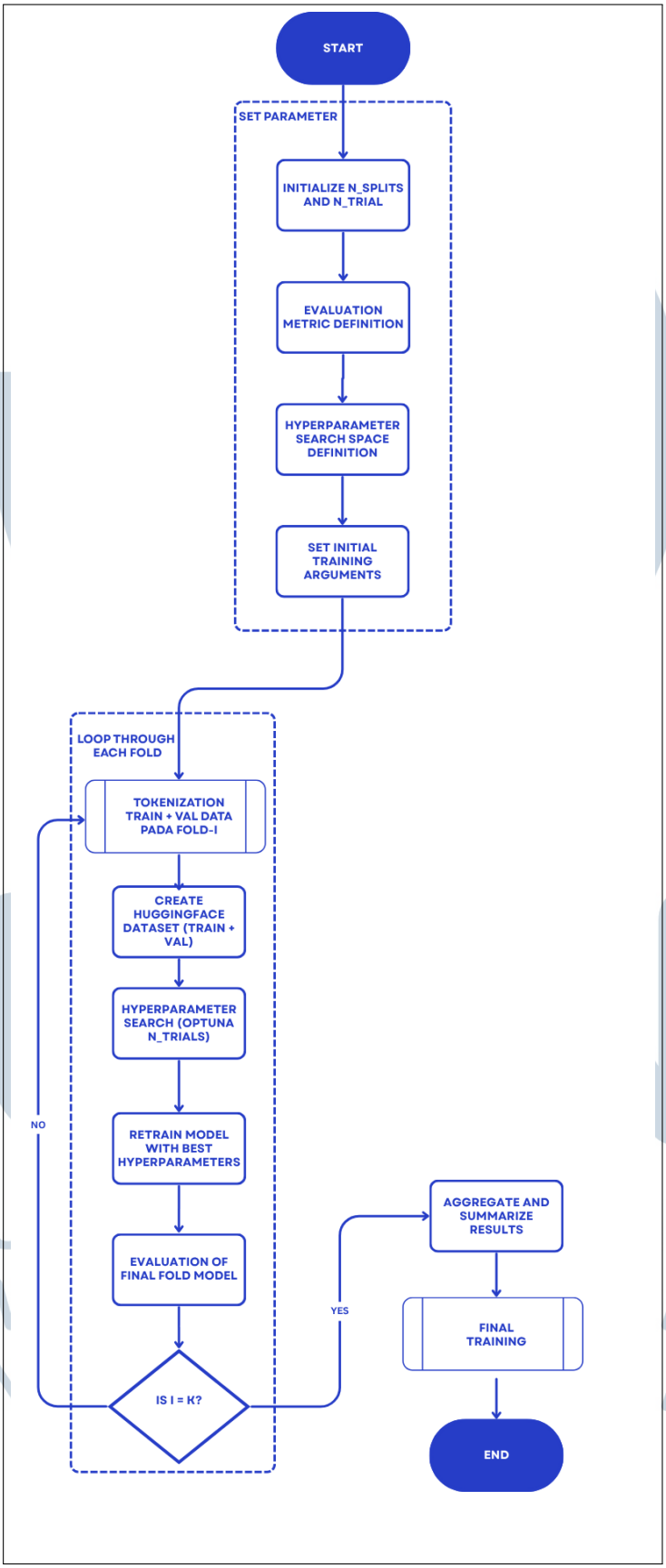
Tahap *Model Training dan Optimization* merupakan inti dari pengembangan sistem deteksi seksisme berbasis model bahasa *IndoBERT*. Pada tahap ini, terdapat dua proses utama yang dilakukan secara bertahap, yaitu: (1) **Cross-Validation dan Hyperparameter Tuning (CV+HPT)**: bertujuan untuk mengeksplorasi konfigurasi pelatihan terbaik melalui kombinasi pencarian *hyperparameter* dan validasi silang untuk menghindari overfitting serta mengevaluasi generalisasi

model, dan (2) **Final Training**: merupakan tahap pelatihan ulang model dengan konfigurasi *hyperparameter* terbaik yang diperoleh dari proses sebelumnya, menggunakan seluruh data *development* untuk memaksimalkan pembelajaran model.

3.4.1 Cross-Validation dan Hyperparameter Tuning

Cross-validation dilakukan menggunakan teknik *Stratified K-Fold* dengan $k = 5$ untuk memastikan distribusi kelas tetap seimbang di setiap fold. Proses ini melibatkan pembagian data *development* ke dalam lima bagian, lalu secara bergiliran menggunakan satu bagian sebagai data validasi dan empat bagian lainnya sebagai data pelatihan.

Proses lengkap dari CV+HPT dapat dilihat pada Gambar 3.4.



Gambar 3.4. Diagram Alur Model Training dan Optimization

Proses cross-validation dan hyperparameter tuning (CV + HPT) seperti yang ditampilkan pada Gambar 3.4 dimulai dengan proses *Stratified K-Fold Data Splitting* terhadap data *development* (90% dari total data), yang memastikan distribusi label antara data pelatihan dan validasi tetap seimbang di setiap fold. Misalnya, jika digunakan 5-fold cross-validation, maka pada setiap iterasi, data train dibagi menjadi 80% data pelatihan dan 20% data validasi, dan proses ini diulang sebanyak lima kali, bergantian menggunakan setiap fold sebagai validasi. Perlu dicatat bahwa data uji (10% awal) tidak digunakan dalam proses ini dan hanya digunakan untuk pengujian akhir setelah final tuning.

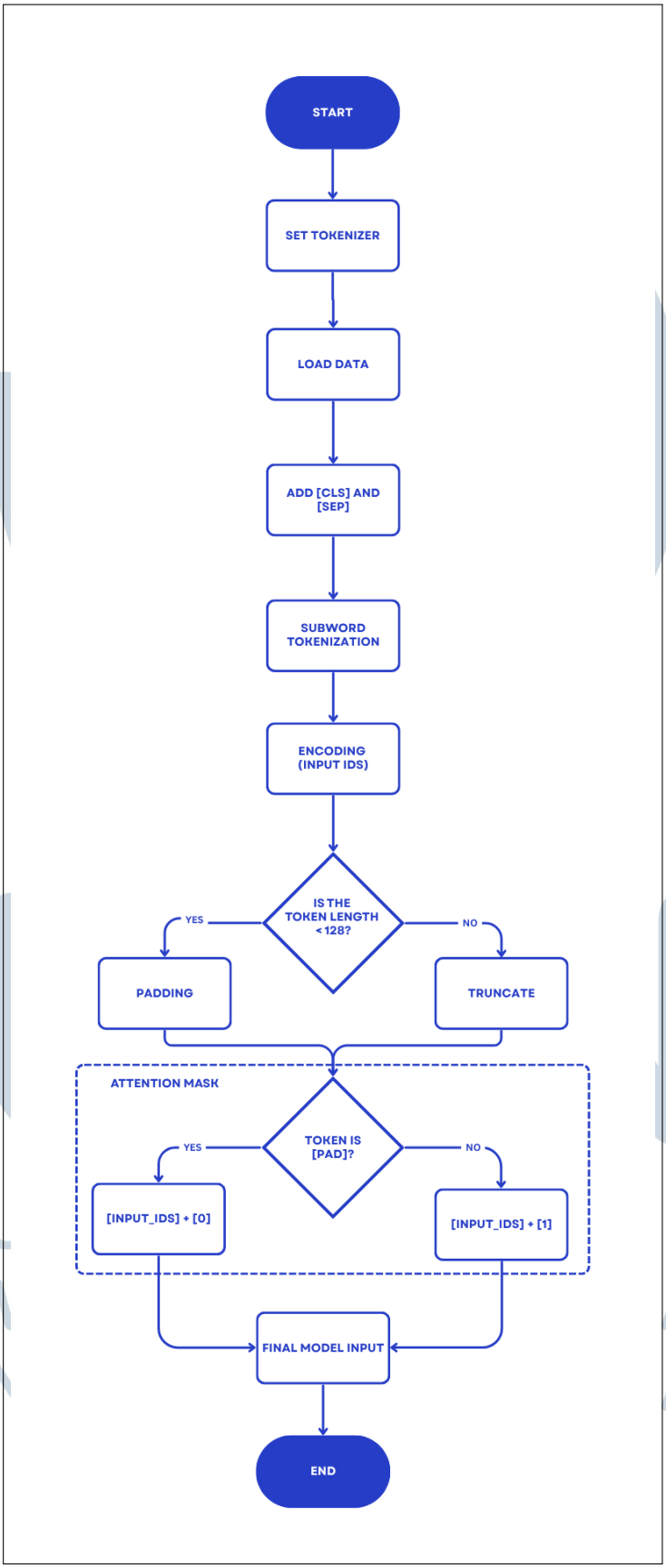
Selanjutnya, metrik evaluasi utama didefinisikan, dalam hal ini adalah *macro F1-score*, karena dataset bersifat tidak seimbang dan metrik ini memberikan penilaian yang lebih adil terhadap masing-masing kelas. Setelah itu, ruang pencarian hyperparameter (*search space*) ditentukan, yang mencakup parameter seperti `learning rate`, `batch size`, `number of epochs`, dan `weight decay`.

Konfigurasi pelatihan awal (*training arguments*) kemudian disiapkan sebelum masuk ke tahap iterasi per fold. Dalam setiap fold, data terlebih dahulu mengalami proses tokenisasi menggunakan tokenizer IndoBERT. Tokenisasi ini mencakup segmentasi subword, penambahan token khusus `[CLS]` dan `[SEP]`, encoding menjadi `input_ids`, serta pembuatan `attention_mask`. Hasil tokenisasi ini kemudian diubah menjadi format dataset HuggingFace yang kompatibel untuk pelatihan menggunakan `Trainer`.

Setelah data siap, pencarian kombinasi hyperparameter terbaik dilakukan menggunakan `Optuna`, sebuah framework otomatisasi tuning yang secara efisien mengeksplorasi kombinasi parameter berdasarkan hasil evaluasi sebelumnya. Untuk setiap kombinasi, model dilatih dan divalidasi menggunakan satu fold, dan hasil metriknya dicatat. Proses ini berulang untuk setiap fold dan setiap kombinasi parameter.

Setelah seluruh fold selesai dilatih dan divalidasi, dilakukan perhitungan rata-rata nilai F1-score dari semua fold untuk setiap kombinasi hyperparameter. Kombinasi dengan nilai rata-rata F1-score tertinggi dipilih sebagai konfigurasi terbaik. Namun, penting untuk dicatat bahwa model tidak langsung dilatih ulang menggunakan seluruh data train secara utuh. Sebaliknya, yang dimaksud dengan ”retrain” pada diagram adalah proses pelatihan model pada setiap fold menggunakan konfigurasi hyperparameter terbaik yang telah ditemukan, dan kemudian hasil metrik dari tiap fold kembali dihitung untuk mengukur performa akhir dari konfigurasi tersebut.

Akhir dari proses ini adalah agregasi dan penyimpulan performa model terbaik yang nantinya akan digunakan pada tahap *final tuning*, yaitu pelatihan ulang pada seluruh data train (90%) dengan konfigurasi terbaik dan pengujian akhir pada data uji (10%) yang belum pernah digunakan sebelumnya.



Gambar 3.5. Diagram Alur Tokenisasi IndoBERT

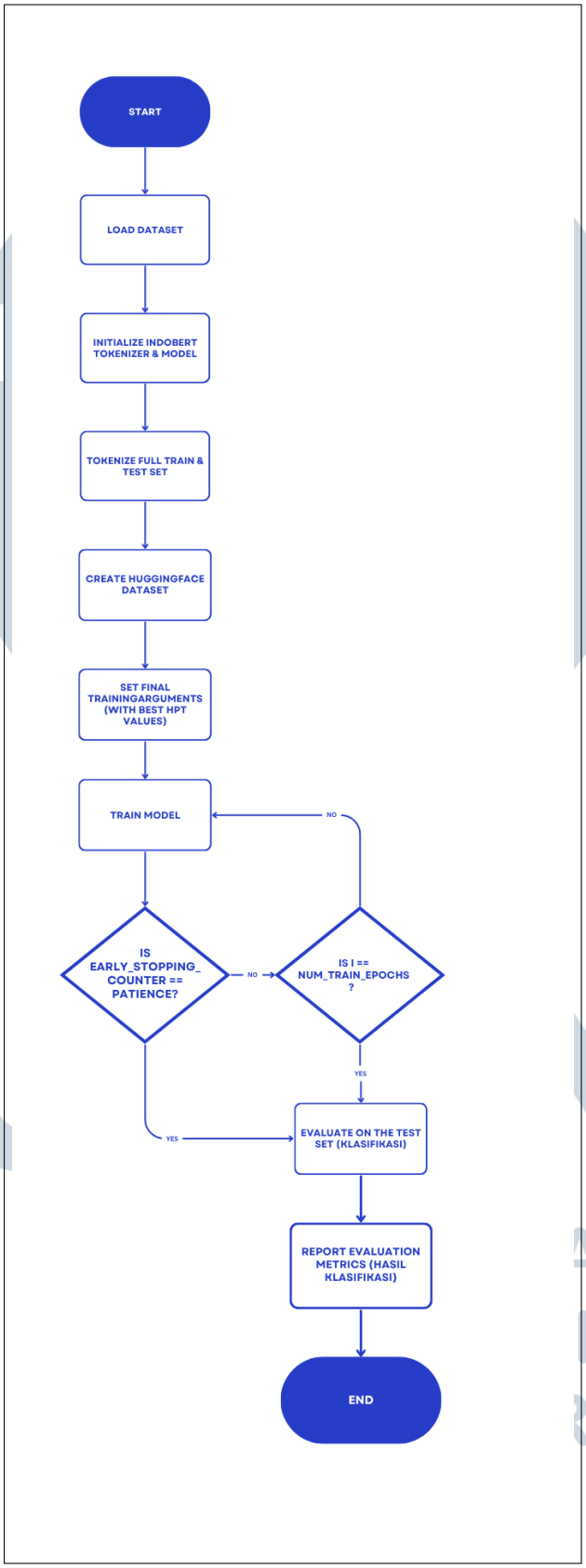
Seperti yang digambarkan pada Gambar 3.5, proses tokenisasi dimulai dari inisialisasi tokenizer IndoBERT, diikuti dengan proses pemuatan data teks mentah dari dataset. Token khusus `[CLS]` dan `[SEP]` ditambahkan di awal dan akhir setiap kalimat sebagai penanda klasifikasi dan batas akhir teks. Setelah itu, dilakukan tokenisasi subword untuk memecah kata-kata menjadi unit-unit subword menggunakan algoritma WordPiece yang telah disesuaikan dalam IndoBERT.

Setiap token kemudian diubah menjadi ID numerik (Input IDs) yang dapat diproses oleh model. Jika jumlah token kurang dari panjang maksimal (dalam kasus ini ¡128 token), maka proses *padding* dilakukan untuk menambahkan token kosong agar panjangnya seragam. Sebaliknya, jika jumlah token melebihi batas maksimal, maka dilakukan proses *truncation* untuk memotong kelebihan token.

Selanjutnya, dibuat *attention mask* untuk menandai posisi mana yang merupakan token asli (bernilai 1) dan mana yang merupakan hasil padding (bernilai 0). Attention mask ini penting dalam proses atensi pada model Transformer agar model hanya fokus pada token-token yang relevan. Akhirnya, seluruh hasil encoding, padding, dan attention mask digabungkan menjadi input final yang siap diberikan ke model IndoBERT untuk pelatihan atau inferensi.

3.4.2 Final Training

Setelah diperoleh konfigurasi hyperparameter terbaik melalui proses *cross-validation* dan *hyperparameter tuning*, tahap selanjutnya adalah melakukan pelatihan akhir (*final training*).



Gambar 3.6. Diagram Alur Final Tuning

Seperti yang ditunjukkan pada Gambar **??**, proses *Final Training* dimulai setelah kombinasi hyperparameter terbaik diperoleh dari tahap Cross-Validation dan Hyperparameter Tuning (CV + HPT). Data yang digunakan pada tahap ini adalah seluruh data *development* (90% dari dataset), yang sebelumnya telah digunakan dalam proses training dan validasi pada setiap fold. Tahap awal mencakup proses persiapan data pelatihan penuh, di mana data disusun kembali untuk digunakan secara menyeluruh dalam proses pelatihan akhir.

Selanjutnya, *tokenizer* dan model IndoBERT diinisialisasi kembali untuk memastikan bahwa proses pelatihan akhir tidak dipengaruhi oleh parameter sebelumnya. Kemudian, proses tokenisasi dilakukan terhadap seluruh data pelatihan (train) dan data uji (test), mencakup segmentasi teks, encoding menjadi `input_ids`, dan pembuatan `attention_mask`. Setelah tokenisasi selesai, dataset dalam bentuk tensor dibuat untuk kedua subset tersebut menggunakan struktur `PyTorch Dataset` agar dapat diproses dalam `Trainer` dari HuggingFace.

Parameter pelatihan terbaik dari proses HPT (misalnya learning rate, batch size, dan epoch) kemudian digunakan sebagai konfigurasi akhir dalam `training arguments`. Model dilatih menggunakan seluruh data pelatihan tersebut tanpa pembagian lagi, karena tidak ada lagi kebutuhan untuk validasi antar-fold. Setelah pelatihan selesai, model langsung dievaluasi menggunakan data uji (10% dari dataset awal) yang sejak awal tidak digunakan sama sekali dalam proses pelatihan ataupun validasi.

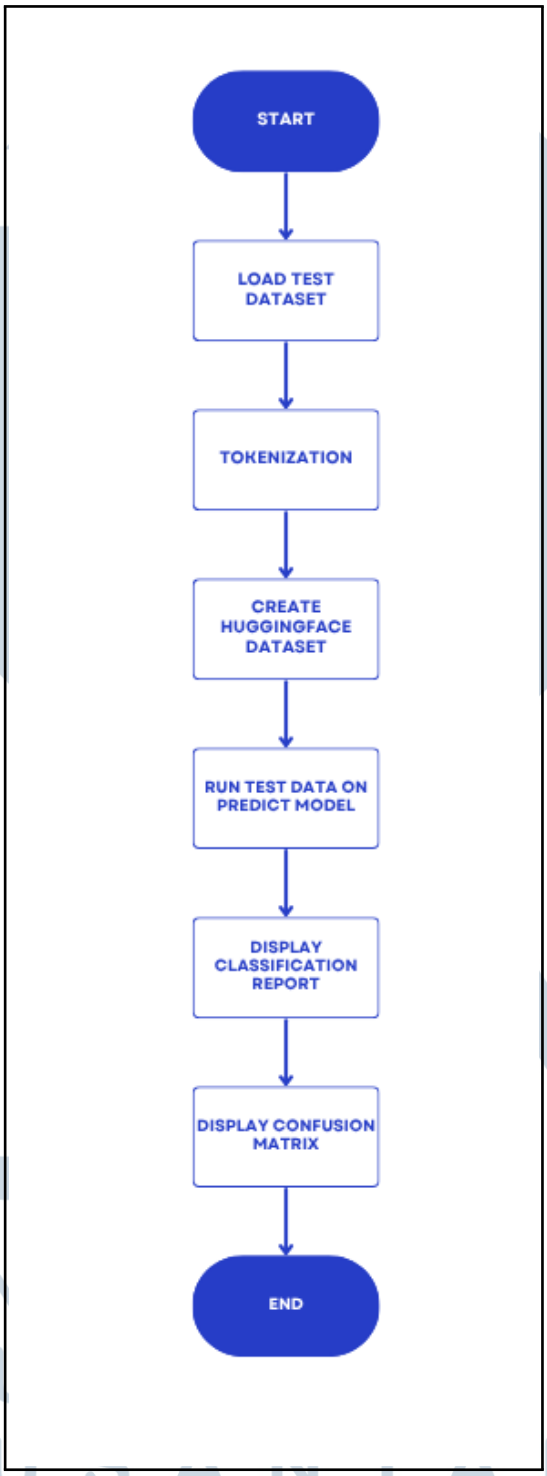
Hasil klasifikasi pada data uji ini dievaluasi menggunakan metrik yang sama seperti pada tahap sebelumnya, yakni *F1-score*, serta dapat dilengkapi dengan nilai akurasi, precision, dan recall. Tahap ini menjadi penilaian akhir terhadap performa model dan menjadi dasar pelaporan hasil klasifikasi sebagai kontribusi utama dari penelitian.

3.5 Final Evaluation

Evaluasi model bertujuan untuk mengukur sejauh mana performa model dalam mengklasifikasikan komentar ke dalam kategori `SEKSIS` dan `TIDAK SEKSIS` secara akurat. Tahap ini dilakukan setelah model selesai dilatih ulang menggunakan seluruh data *development* dan diuji pada data *testing* (10% dari total dataset) yang telah disisihkan sejak awal dan tidak pernah digunakan dalam proses pelatihan maupun validasi.

Gambar 3.7 menunjukkan alur evaluasi akhir yang dilakukan terhadap

model.



Gambar 3.7. Diagram Alur Evaluasi Akhir Model

Langkah pertama adalah memuat data *testing*, diikuti dengan proses tokenisasi menggunakan tokenizer IndoBERT untuk menghasilkan *input IDs* dan

attention mask. Data yang telah ditokenisasi kemudian dikonversi ke dalam format *HuggingFace Dataset* agar kompatibel dengan modul `Trainer`.

Model yang telah dilatih kemudian digunakan untuk melakukan prediksi terhadap data *testing*. Hasil prediksi selanjutnya dianalisis menggunakan *classification report* yang mencakup metrik-metrik seperti *F1-score*, *accuracy*, *precision*, dan *recall*.

Metrik utama yang digunakan adalah *F1-score*, karena metrik ini mempertimbangkan keseimbangan antara *precision* dan *recall* dalam satu nilai gabungan. Selain itu, nilai *accuracy*, *precision*, dan *recall* juga dilaporkan untuk memberikan evaluasi yang lebih lengkap. Akurasi mengukur proporsi prediksi yang benar secara keseluruhan, precision menunjukkan proporsi prediksi `SEKSIS` yang benar, dan recall mengukur kemampuan model dalam menemukan seluruh data yang benar-benar `SEKSIS`.

Sebagai pelengkap, ditampilkan pula *confusion matrix* yang menggambarkan kinerja klasifikasi model dalam empat kategori: True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN). Visualisasi ini membantu untuk mengidentifikasi kecenderungan kesalahan model, seperti apakah model lebih sering gagal mengenali komentar `SEKSIS` atau cenderung terlalu sering menandai komentar netral sebagai `SEKSIS`.

Evaluasi akhir ini menjadi dasar dalam menyimpulkan efektivitas model serta memberikan gambaran umum tentang kemampuan generalisasi model terhadap data baru.