BAB 3
METODOLOGI PENELITIAN

3.1 Metode Penelitian

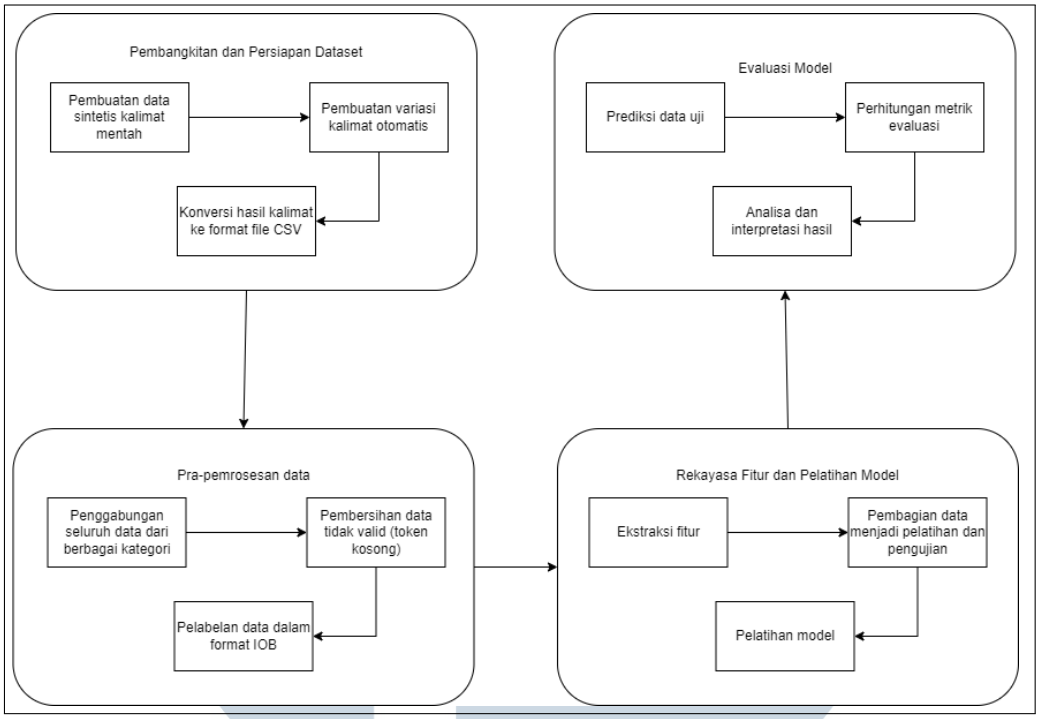
Penelitian ini menggunakan pendekatan eksperimental dengan menerapkan metode *supervised machine learning* (pembelajaran mesin terarah) untuk membangun model deteksi kesalahan penggunaan huruf miring pada teks berbahasa Indonesia. Pendekatan eksperimental dipilih karena penelitian ini berfokus pada pengujian model yang dilakukan melalui serangkaian tahapan terstruktur, meliputi *data generating*, *pre-processing*, *feature extraction*,dan *model evaluation*.

Metode *supervised machine learning* digunakan karena memungkinkan sistem untuk belajar dari data berlabel, sehingga model dapat mengenali pola-pola linguistik yang membedakan antara kata atau frasa yang seharusnya ditulis miring dan yang tidak. Dalam penelitian ini, data pelatihan berupa kalimat-kalimat berbahasa Indonesia yang dihasilkan secara otomatis melalui proses *dataset generation*. Setiap token dalam kalimat diberi label yang menunjukkan apakah token tersebut termasuk kategori "miring" atau "tegak", sesuai dengan kaidah Pedoman Umum Ejaan Bahasa Indonesia (PUEBI).

Secara keseluruhan, terdapat 12 kategori huruf miring (wadah), meliputi judul buku, film, album, acara televisi, audio, video, lakon, nama majalah, surat kabar, bahasa asing (belum terserap ke KBBI), bahasa daerah, dan nama ilmiah. Selain itu, terdapat lima kategori isi wadah (huruf tegak) yang harus ditulis di dalam tanda petik, yaitu lagu, episode, judul artikel, opini, dan judul puisi.

3.2 Tahapan Penelitian

Penelitian ini dilakukan melalui beberapa tahapan yang tersusun secara sistematis untuk memastikan proses pengembangan model deteksi kesalahan huruf miring berjalan terarah dan dapat direplikasi. Secara umum, alur penelitian ini terdiri atas empat tahap utama, yaitu pembangkitan dan persiapan dataset, pra-pemrosesan data, rekayasa fitur dan pelatihan model, serta evaluasi model. Setiap tahap saling terhubung dan memiliki peran penting dalam membentuk sistem deteksi yang akurat dan andal. *Pipeline* penelitian secara umum dapat dilihat pada Gambar 3.1.



Gambar 3.1. *Pipeline* penelitian

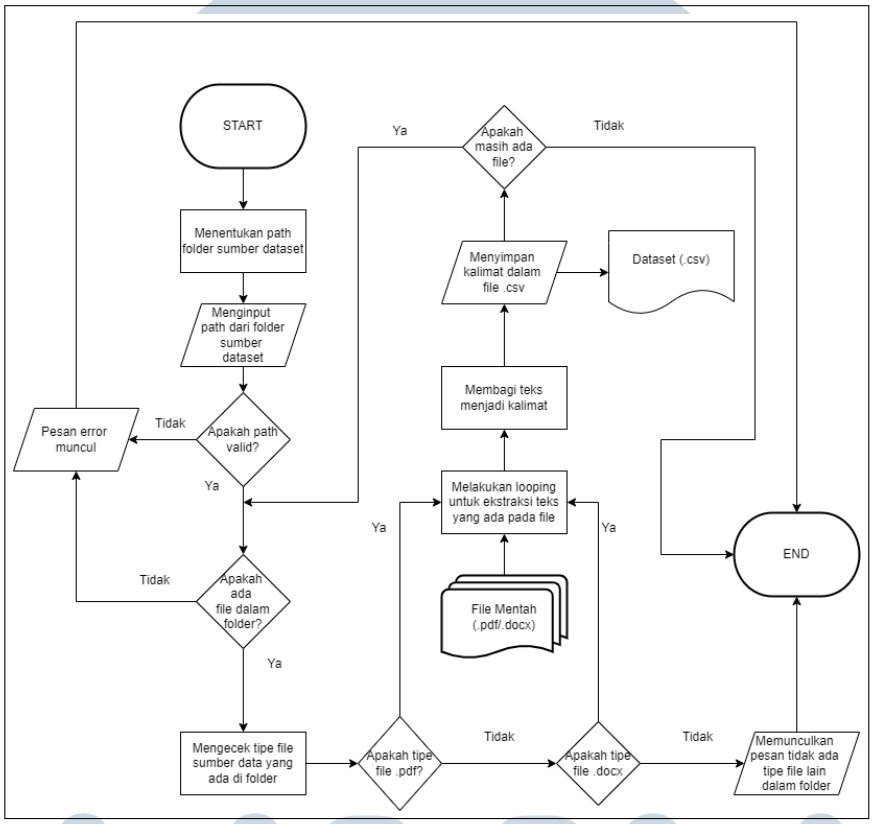
3.3 Pengumpulan dan Persiapan Dataset

Tahap pengumpulan dan persiapan dataset merupakan langkah penting dalam membangun sistem deteksi kesalahan penggunaan huruf miring. Pada penelitian ini, data utama diperoleh dari kumpulan artikel berita berbahasa Indonesia yang dikumpulkan dari berbagai sumber daring. Total artikel yang digunakan sebanyak 10.000 artikel, yang mencakup beragam topik.

Untuk melengkapi dan memperluas variasi pola linguistik, dataset diperkuat dengan data tambahan hasil pembangkitan otomatis menggunakan pendekatan *synthetic dataset generation*. Proses ini dilakukan untuk menambahkan contoh-contoh kalimat dengan struktur yang bervariasi, terutama yang melibatkan entitas berhuruf miring dan huruf tegak sesuai dengan kaidah Pedoman Umum Ejaan Bahasa Indonesia (PUEBI).

Secara teknis, proses transformasi data mentah dari dokumen sumber menjadi format dataset yang siap digunakan digambarkan melalui *flowchart* pada Gambar 3.2. Proses ini diawali dengan validasi direktori sumber data untuk memastikan keberadaan dan validitas *path*. Sistem kemudian melakukan pemindaian terhadap file dokumen dengan ekstensi *.pdf* dan *.docx*. Setiap dokumen yang valid akan melalui proses iterasi ekstraksi teks, di mana konten teks diambil dan selanjutnya

disegmentasi menjadi unit-unit kalimat terpisah. Kalimat-kalimat hasil segmentasi tersebut kemudian disimpan secara terstruktur ke dalam file *.csv*. Proses ini berulang (*looping*) hingga seluruh file dalam direktori sumber selesai diproses.



Gambar 3.2. *Flowchart* persiapan dan ekstraksi dataset

Dalam penelitian ini, entitas diklasifikasikan ke dalam dua kelompok utama, yaitu kategori wadah dan kategori isi. Kategori pertama adalah kategori isi, yang merujuk pada bagian dari suatu karya atau entitas yang lebih besar. Dalam penulisan, entitas ini ditandai dengan penggunaan tanda petik tegak. Kategori isi terdiri dari lima jenis entitas. Statistik jumlah data untuk masing-masing jenis entitas pada kategori ini disajikan pada Tabel 3.1.

Tabel 3.1. Distribusi data entitas kategori isi

Jenis Entitas (Isi)	**Jumlah Data**
Judul Lagu	2916
Episode	2879
Judul Artikel	3054

Tabel 3.1. Lanjutan 3.1

Jenis Entitas (Isi)	Jumlah Data
Opini	4180
Judul Puisi	2645

Kategori kedua adalah kategori wadah, yaitu entitas yang secara aturan penulisan ditandai dengan penggunaan huruf miring (*italics*). Kategori ini mencakup dua belas jenis entitas yang berbeda. Rincian distribusi jumlah data untuk setiap jenis entitas dalam kategori wadah dapat dilihat pada Tabel 3.2.

Tabel 3.2. Distribusi Data Entitas Kategori Wadah

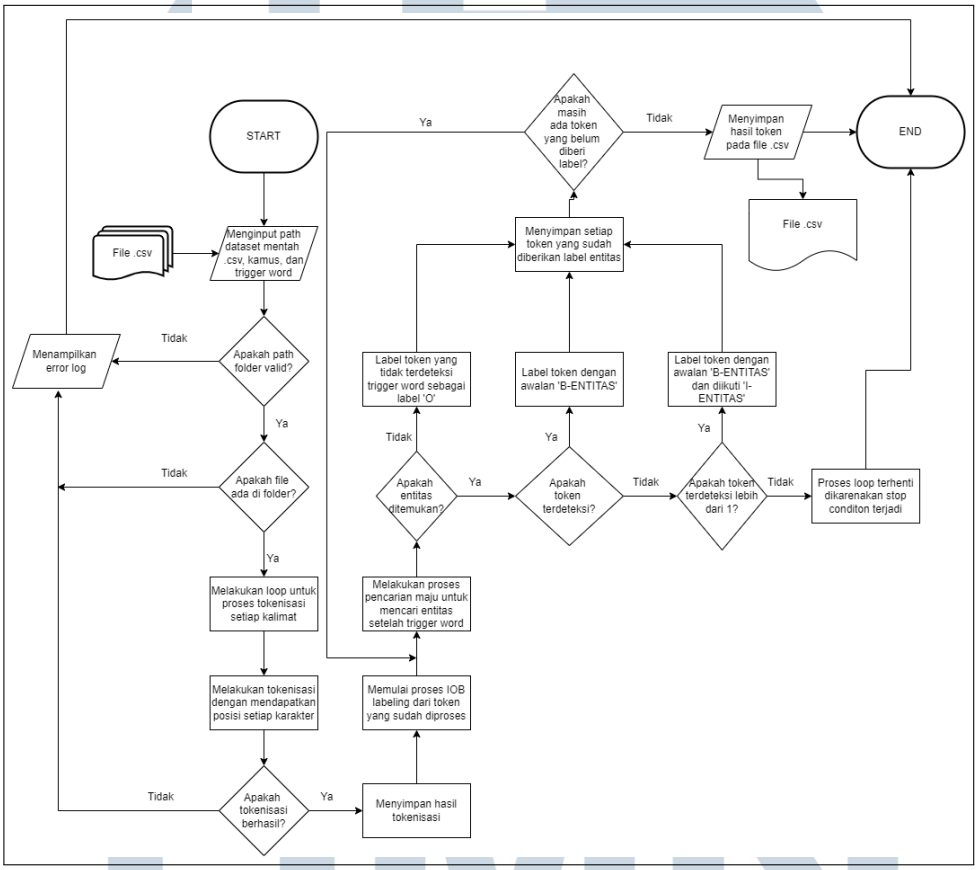
Jenis Entitas (Wadah)	Jumlah Data
Judul Buku	3279
Judul Film	2515
Judul Album	2637
Acara Televisi	2870
Audio	2525
Video	4329
Lakon (Drama)	2595
Nama Majalah	2434
Surat Kabar	2584
Bahasa Asing	3894
Bahasa Daerah	2459
Nama Ilmiah	1032

3.4 Pra-pemrosesan Data

Setelah dataset terkumpul, tahapan selanjutnya adalah pra-pemrosesan data yang memiliki tujuan untuk menyiapkan dataset agar dapat digunakan secara optimal oleh model *Conditional Random Fields* (CRF). Data yang diperoleh dari proses pengumpulan dan pembangkitan memiliki variasi format dan struktur, sehingga diperlukan serangkaian langkah normalisasi dan transformasi agar representasinya seragam.

Pada bagian ini, yang dilakukan adalah proses tokenisasi, yaitu pemisahan teks (kalimat) menjadi satuan berupa token (kata). Tokenisasi ini dilakukan untuk mengubah kalimat mentah menjadi format baris per token yang siap digunakan untuk pelabelan.

Selanjutnya dilakukan proses label IOB (*Inside, Outside, Beginning*) untuk setiap token. Label ini menandai posisi token dalam suatu entitas, apakah berada di awal entitas (B), di dalam entitas(I), atau di luar entitas (O). Skema pelabelan ini digunakan untuk membedakan kategori wadah (huruf miring) dan isi wadah (huruf tegak dengan tanda kutip), sebagaimana didefinisikan dalam kaidah PUEBI.



Gambar 3.3. Flowchart tahapan pra-pemrosesan dan pelabelan token

Secara rinci, alur implementasi teknis untuk tahapan pra-pemrosesan ini divisualisasikan pada Gambar 3.3. Proses diawali dengan inisiasi dan validasi jalur direktori untuk dataset mentah, kamus data, dan daftar kata pemicu (*trigger word*). Setelah validasi berhasil, sistem menjalankan prosedur tokenisasi untuk memecah kalimat sekaligus merekam posisi karakter setiap token. Tahapan inti dari alur ini adalah mekanisme pelabelan otomatis yang memanfaatkan pencarian maju (*forward search*) berbasis kata pemicu. Algoritma akan mendeteksi keberadaan entitas setelah kata pemicu; jika entitas ditemukan, token akan dilabeli sebagai 'B-ENTITAS' untuk kata pertama dan 'I-ENTITAS' jika terdiri dari beberapa kata. Sebaliknya, token yang tidak terindikasi sebagai entitas akan diberi label 'O'.

Iterasi ini terus berlangsung hingga seluruh token dalam dataset selesai dilabeli dan disimpan. Setelah proses pelabelan selesai, data disimpan dalam format Comma-Separated Values (CSV).

3.5 Rekayasa Fitur dan Pembagian Data

Setelah tahapan pra-pemrosesan selesai, selanjutnya dilakukan tahap rekayasa fitur yang bertujuan menyiapkan data untuk pelatihan model CRF. Tahapan ini sangat penting karena model CRF tidak bekerja langsung pada kata mentah, melainkan pada serangkaian fitur yang diekstraksi dari setiap token.

Fitur pertama yang digunakan pada model ini yakni fitur kontekstual, yang mempertimbangkan posisi token dalam kalimat serta hubungannya dengan token di sekitarnya. Fitur ini bertugas untuk menangkap pola linguistik yang bersifat kontekstual, seperti posisi awal atau akhir kalimat, keberadaan kata pemicu tertentu, serta bentuk token tetangga. Daftar fitur kontekstual dan posisi ditunjukkan pada Tabel 3.3.

Tabel 3.3. Daftar fitur pemicu, posisi, dan kontekstual model CRF

Kategori	**Nama Fitur**	**Deskripsi**
Kontekstual	`-1:token.lower().`	Kumpulan fitur (i, *shape*, leksikal) untuk 1 token sebelumnya ($idx-1$).
	`-2:token.lower().`	Kumpulan fitur (*lower*, *shape*, trigger) untuk 2 token sebelumnya ($idx-2$).
	`-3:token.lower().`	Kumpulan fitur (*lower*, *shape*, trigger) untuk 3 token sebelumnya ($idx-3$).
	`+1:token.lower().`	Kumpulan fitur (*lower*, *shape*, leksikal) untuk 1 token setelahnya ($idx+1$).
	`+2:token.lower().`	Kumpulan fitur (*lower*, *shape*, *quote*) untuk 2 token setelahnya ($idx+2$).
	`-1shp+tshp...`	Kombinasi *shape* antara token saat ini (idx) dengan token tetangganya ($idx-1$ dan $idx+1$).

Tabel 3.4. Lanjutan Tabel 3.3 (fitur pemicu dan posisil)

Kategori	Nama Fitur	Deskripsi
Pemicu	`trigger_nearby`	Mencari kata pemicu (misal: 'buku', 'film') dalam jendela 7 token sebelumnya.
Posisi	`BOS`	Biner: *Beginning of Sentence* (token pertama).
	`EOS`	Biner: *End of Sentence* (token terakhir).

Fitur kedua yaitu fitur yang digunakan bersifat lokal dan leksikal, yaitu fitur-fitur yang melekat langsung pada token itu sendiri tanpa mempertimbangkan konteks kalimat di sekitarnya. Fitur lokal mencakup bentuk huruf dan pola penulisan (seperti awalan dan akhiran), sedangkan fitur leksikal merujuk pada informasi yang diperoleh dari kamus atau daftar kata tertentu, seperti daftar kata bahasa asing, bahasa daerah, dan nama ilmiah. Daftar lengkap fitur tersebut ditunjukkan pada Tabel 3.5.

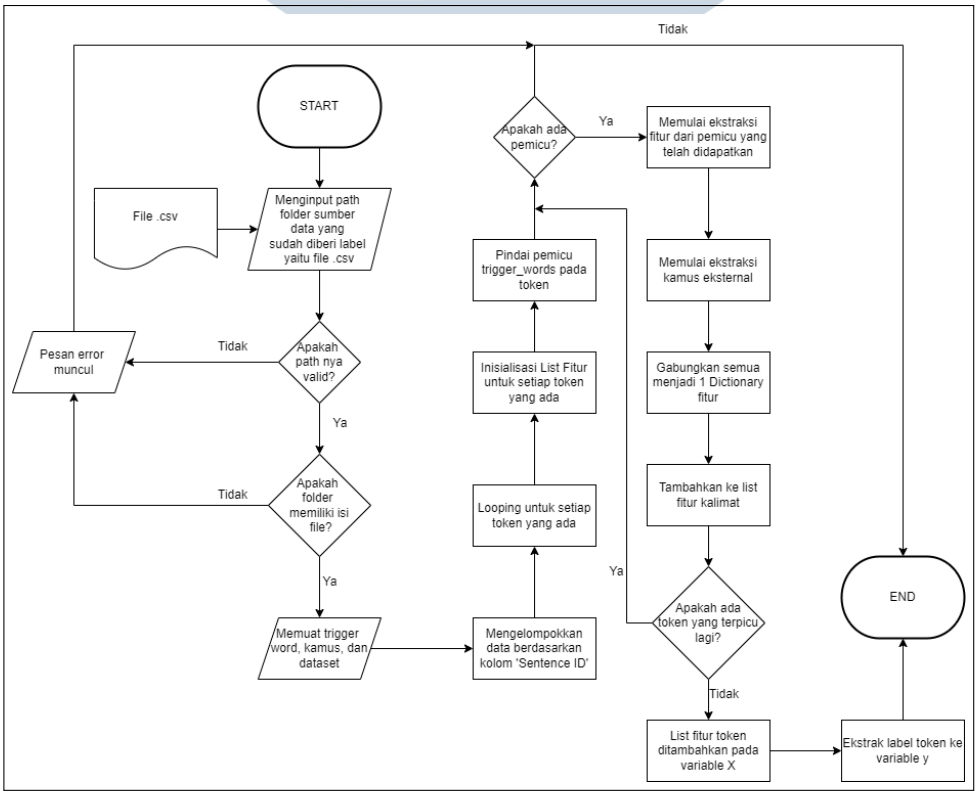
Tabel 3.5. Daftar fitur lokal dan leksikal model CRF

Kategori	Nama Fitur	Deskripsi
Lokal	`bias`	Nilai bias konstan (1.0).
	`token.lower()`	Teks token dalam huruf kecil.
	`token.isupper()`	Biner: Apakah token seluruhnya huruf besar.
	`token.istitle()`	Biner: Apakah token diawali huruf besar.
	`token.isdigit()`	Biner: Apakah token seluruhnya angka.
	`prefix-1`, `prefix-2`	1 dan 2 karakter awalan token.
	`suffix-1`, `suffix-2`	1 dan 2 karakter akhiran token.
	`token.shape`	Representasi abstrak bentuk token (misal: 'Xxxx-Xxxx').
	`token.isquote`	Biner: Apakah token adalah karakter kutip (''', '"', '*').

Tabel 3.5. Lanjutan Tabel 3.5 (Fitur tanda baca dan leksikal)

Kategori	**Nama Fitur**	**Deskripsi**
Tanda Baca	`token.ispunct_std`	Biner: Apakah token tanda baca standar (misal: '.', ',', '?').
	`token.has_internal_punct`	Biner: Apakah token punya tanda baca internal (misal: "o'clock").
Leksikal	`token.is_foreign`	Biner: Apakah token (huruf kecil) ada di kamus Bahasa Asing.
	`token.is_daerah`	Biner: Apakah token (huruf kecil) ada di kamus Bahasa Daerah.
	`token.is_ilmiah`	Biner: Apakah token (huruf kecil) ada di kamus Nama Ilmiah.

Implementasi komputasi untuk ekstraksi fitur-fitur yang telah dijabarkan di atas digambarkan melalui *flowchart* pada Gambar 3.4.



Gambar 3.4. *Flowchart* rekayasa fitur

Proses dimulai dengan memuat dataset yang telah dilabeli beserta sumber

daya eksternal seperti kamus dan daftar kata pemicu. Sistem kemudian melakukan iterasi, mengelompokkan data berdasarkan kalimat, dan memindai setiap token untuk mengekstraksi atribut lokal, kontekstual, dan leksikalnya. Seluruh informasi fitur yang terkumpul kemudian dikompilasi menjadi variabel fitur (X), sementara label entitas yang berkorespondensi diekstraksi secara terpisah ke dalam variabel target (y). Hasil akhir dari alur ini adalah struktur data numerik yang siap untuk tahap pembagian dataset.

Setelah semua kalimat dari file CSV diproses dan diubah menjadi sekuens fitur, tahapan pembagian data dilakukan. Keseluruhan sekuens data dibagi menjadi tiga set yang berbeda secara acak (*shuffle*) yaitu, data latih, data validasi, dan data uji. Data Latih (80%) digunakan untuk mempelajari pola, data validasi (10%) digunakan untuk proses *tuning* (meskipun tidak digunakan dalam pelatihan akhir), dan sisa data uji (10%) disimpan terpisah untuk evaluasi akhir.

Ketiga set data ini (fitur `X` dan label `y` untuk tiap set) kemudian disimpan dalam format *pickle* (`.joblib`) agar siap digunakan untuk tahapan pelatihan model.

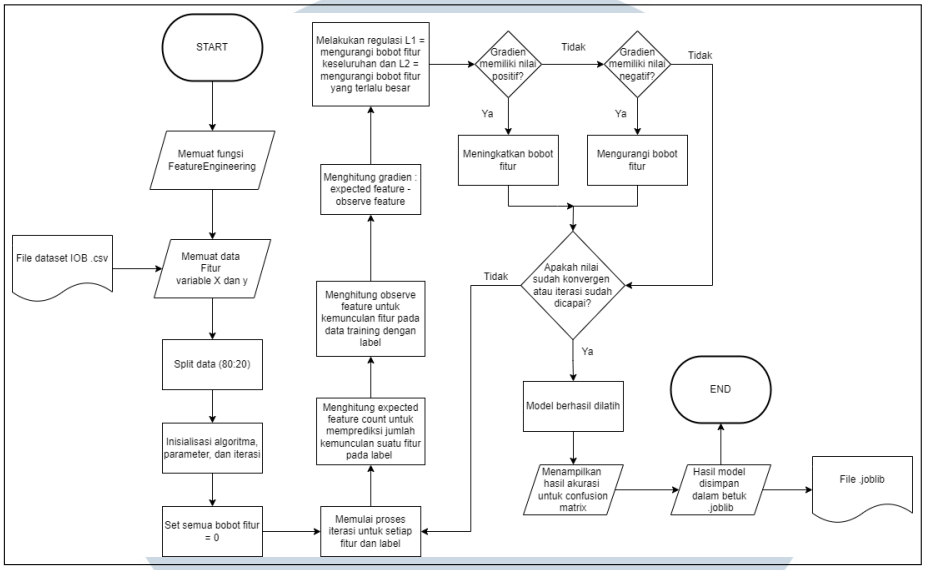
3.6 Pelatihan dan Evaluasi Model

Tahap terakhir dalam metodologi penelitian ini adalah pelatihan dan evaluasi model. Proses pelatihan dimulai dengan memuat data latih (`X_train`, `y_train`) yang telah disimpan dalam format *joblib*. Model *Conditional Random Field* (CRF) diinisialisasi menggunakan pustaka `sklearn_crfsuite` dengan serangkaian *hyperparameter* yang telah ditentukan. Parameter yang dikonfigurasi meliputi algoritma `lbfgs` sebagai metode optimisasi, koefisien regularisasi L1 (`c1`) dan L2 (`c2`) sebagai penalti bobot fitur, serta parameter `max_iterations` yang menetapkan batas atas jumlah iterasi pelatihan.

Model dilatih menggunakan fungsi `crf.fit(X_train, y_train)`. Setelah proses pelatihan selesai, model yang telah terlatih disimpan ke dalam satu file berformat `.joblib` agar dapat digunakan kembali untuk proses prediksi.

Secara algoritmis, mekanisme internal pelatihan model CRF untuk mencapai konvergensi digambarkan melalui *flowchart* pada Gambar 3.5. Proses ini dimulai dengan inisialisasi seluruh bobot fitur ke nilai nol, dilanjutkan dengan perhitungan iteratif untuk membandingkan nilai fitur yang diamati (*observed feature*) pada data latih dengan nilai ekspektasi model (*expected feature*). Selisih antara kedua nilai tersebut menentukan arah gradien, yang kemudian digunakan untuk memperbarui bobot fitur. Proses pembaruan ini turut memperhitungkan penalti regularisasi L1

dan L2. Siklus optimasi berlanjut hingga tercapai konvergensi (norma gradien mendekati nol) atau batas iterasi maksimum terpenuhi.



Gambar 3.5. *Flowchart* algoritma pelatihan model CRF

Selanjutnya, evaluasi kuantitatif dilakukan untuk mengukur performa model secara objektif. Model yang telah dilatih digunakan untuk memprediksi label pada data uji (`X_test`), yaitu data yang belum pernah dilihat oleh model selama proses pelatihan. Hasil prediksi model (`y_pred`) kemudian dibandingkan dengan label sebenarnya (`y_test`). Performa model dievaluasi menggunakan metrik standar yang dihitung melalui laporan klasifikasi (`metrics.flat_classification_report`). Metrik utama yang digunakan adalah *Accuracy*, *Precision*, *Recall*, dan *F1-Score*. Evaluasi difokuskan pada label entitas (selain label 'O') untuk memberikan gambaran yang lebih jelas mengenai efektivitas model dalam mendeteksi dua belas kategori huruf miring dan lima kategori isi wadah yang ditargetkan.