

BAB 3 METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Dalam penelitian ini, terdapat beberapa tahapan dalam proses perancangannya. Berikut merupakan tahap-tahap yang dilakukan:

1. Studi Literatur

Tahapan awal dalam pembuatan sistem pakar ini adalah mempelajari dan mendalami beberapa literatur, yaitu *Zero-Shot*, *Cross-Lingual*, BLOOM, serta beberapa penelitian sebelumnya dengan topik yang berkaitan.

2. Analisa Kebutuhan

Pada tahap ini kebutuhan yang diperlukan dalam penelitian adalah menggunakan *Google Colaboratory* untuk melatih dan menjalankan model.

3. Pengumpulan Data

Dataset yang dibutuhkan berasal dari Crawled Wikipedia, Medical Dialog dataset, WikiQA dataset, tydiqa dan *prompt* ChatGPT. Dataset yang diambil merupakan tipe *question answering* dengan format CSV. Kemudian dataset tersebut dimodifikasi sesuai dengan model yang ingin dibuat. *Prompting* dilakukan pada tanggal 15 April 2023 dan menggunakan input khusus seperti "Jawab pertanyaan di bawah ini dengan lengkap, jelas, dan panjang minimal 4 kalimat serta gunakan format *pipe seperated value*".

Tabel 3.1. Sumber dataset bahasa Inggris

Dataset	Sumber
<i>wiki_csai</i> , <i>medicine</i> , <i>finance</i> , dan <i>open_qa</i>	<i>How Close is ChatGPT to Human Experts? Comparison Corpus, Evaluation, and Detection</i> [12]

Tabel 3.2. Sumber dataset bahasa Indonesia

Dataset	Sumber
<i>tydiqa</i>	Huggingface[19]
<i>Prompting ChatGPT</i> pada tanggal 15 April 2023	ChatGPT[3]

4. Perancangan Sistem

Tahapan ini dilakukan dengan merancang *flowchart* dari implementasi metode *zero-shot cross-lingual transfer learning* untuk klasifikasi teks buatan mesin dengan model bahasa BLOOM.

5. Implementasi

Tahap ini melakukan implementasi dari model yang sudah dirancang sebelumnya. Bahasa pemrograman yang digunakan untuk memproses data, melatih model, dan melakukan evaluasi model adalah Python. Semua proses implementasi akan menggunakan *Google Colaboratory*.

6. Evaluasi

Tahap ini melakukan evaluasi dari model yang sudah dibuat dalam melakukan klasifikasi teks yang dibuat oleh mesin atau manusia dengan metrik evaluasi yang digunakan diantaranya, *accuracy*, *precision*, *recall*, *F1-score* untuk mengetahui hasil model yang sudah dilatih agar model tersebut bisa diperbaiki atau dikembangkan lagi.

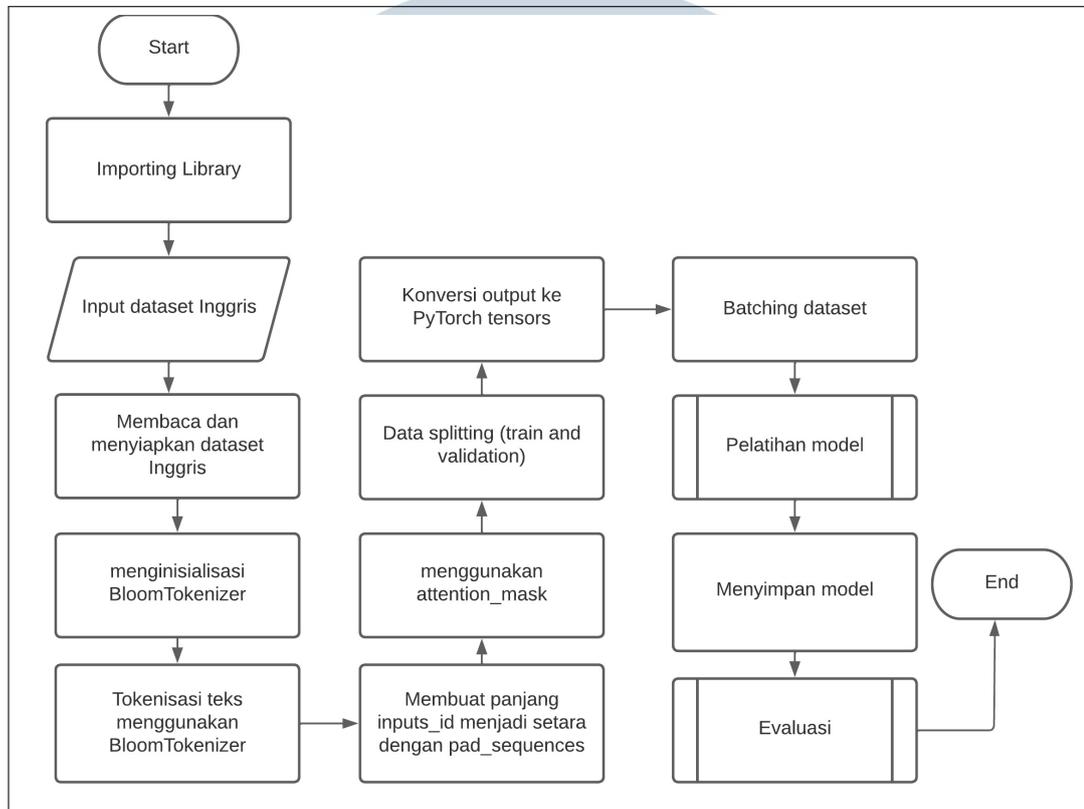
7. Konsultasi dan Penulisan Laporan

Di dalam tahap ini akan melakukan konsultasi dengan para dosen pembimbing terkait penulisan laporan penelitian yang akan dibuat.

3.2 Perancangan Model

Perancangan model yang dilakukan dengan menggunakan *flowchart*. Dalam penelitian ini, terdapat 4 *flowchart* yang dibuat, yaitu *main flowchart*, *model training*, *model validation* dan *evaluation*.

3.2.1 Main Flowchart

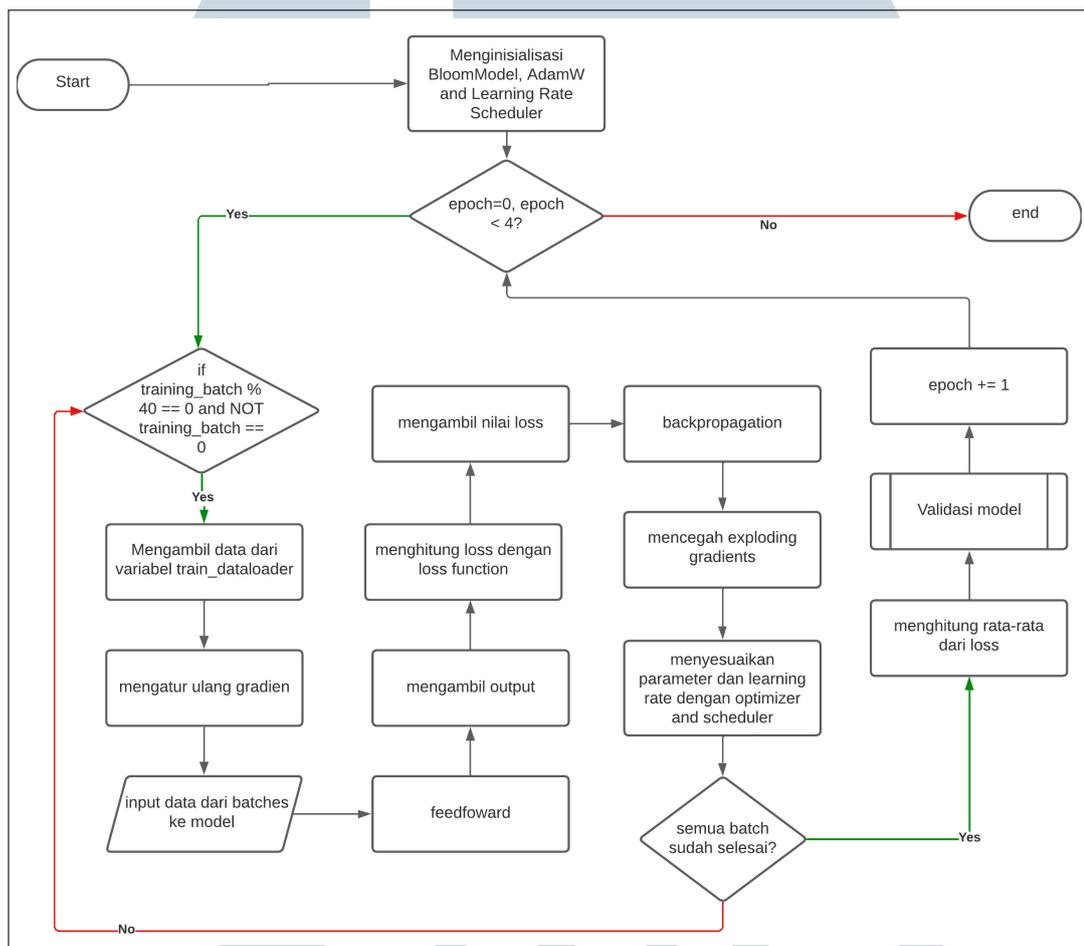


Gambar 3.1. Main flowchart

Pada Gambar 3.1, menampilkan *flowchart* utama dari pembentukan model untuk klasifikasi teks AI dan manusia dengan BLOOM. Langkah pertama yang dilakukan adalah mengimport *library-library* yang dibutuhkan untuk memudahkan proses *training* dan *evaluation*. Langkah ke-2, membaca dan melakukan *preprocessing* pada *dataset* Inggris. Kemudian, menggunakan *tokenizer* dari BLOOM supaya teks dapat dibaca oleh mesin. Setelah teks sudah menjadi *token*, langkah selanjutnya membuat panjang dari setiap token menjadi sama. Hal ini perlu dilakukan supaya model dapat memproses data dalam *batch*. Setelah itu, menggunakan *attention mask* agar model dapat membedakan nilai token sebenarnya dan nilai *padding*. Selanjutnya, *dataset* akan *displit* menjadi *training* dan *validation*. Lalu, data akan dikonversi ke dalam Pytorch tensor agar dapat menggunakan fitur *library* Pytorch, yaitu *batching dataset*, *model training*, dan *model validation*. Langkah selanjutnya adalah menyimpan model agar dapat memanggil modelnya tanpa harus melakukan *training* lagi. Setelah model sudah

dilatih, maka akan *disave* agar dapat dipanggil kembali tanpa perlu *training* lagi. Tahap terakhir adalah evaluasi model, pada tahap ini model akan diuji seberapa bagus performa dari prediksi yang diberikan.

3.2.2 Model Training

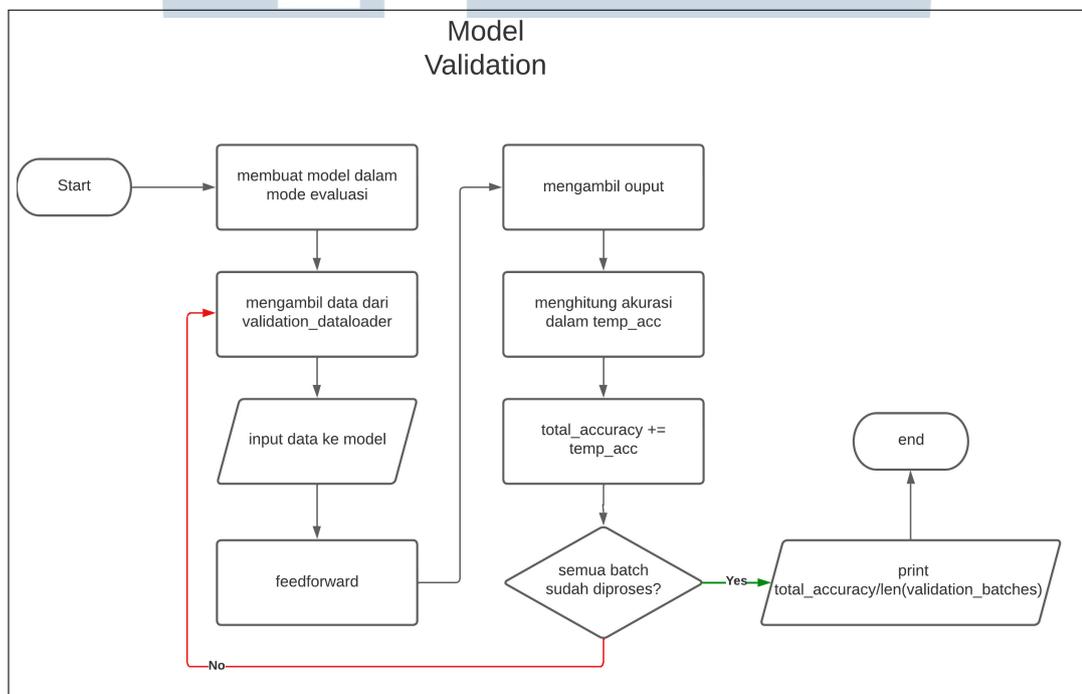


Gambar 3.2. Flowchart Model Training

Pada Gambar 3.2, merupakan *flowchart* dari *model training*. Langkah pertama, menginisialisasi model Bloom, *optimizer*, dan *scheduler*. Lalu ada *for loop* ketika *epoch* belum diproses sebanyak 4 kali, maka akan lanjut ke tahap selanjutnya. *Epoch* merupakan satu kali penyelesaian penggunaan seluruh data pelatihan untuk melatih model. Pemilihan jumlah *epoch* sebesar 4 untuk *model benchmark*. Tahap berikutnya data akan diambil setiap 40 *batches* dari *train dataloader*. Selanjutnya, *gradient* akan *direset* supaya mencegah *gradient* terakumulasi di seluruh *batch*.

Kemudian, memasukkan *input* ke dalam model. Lalu model akan menebak *weight* serta mengkalkulasi *loss*nya. Setelah itu *weight* akan diupdate oleh *optimizer* dan *learning rate* dari *optimizer* akan dirubah oleh *scheduler*. Proses ini akan berulang sampai semua *batch* sudah selesai. Setelah semua batch sudah selesai dilakukan, total dari *loss* akan dirata-ratakan. Lalu, model akan masuk tahap validasi model yang akan ditampilkan pada Gambar 3.3. Lalu, *epoch* ke-1 sudah selesai dilakukan, proses ini akan berulang sampai *epoch* terakhir sudah diproses.

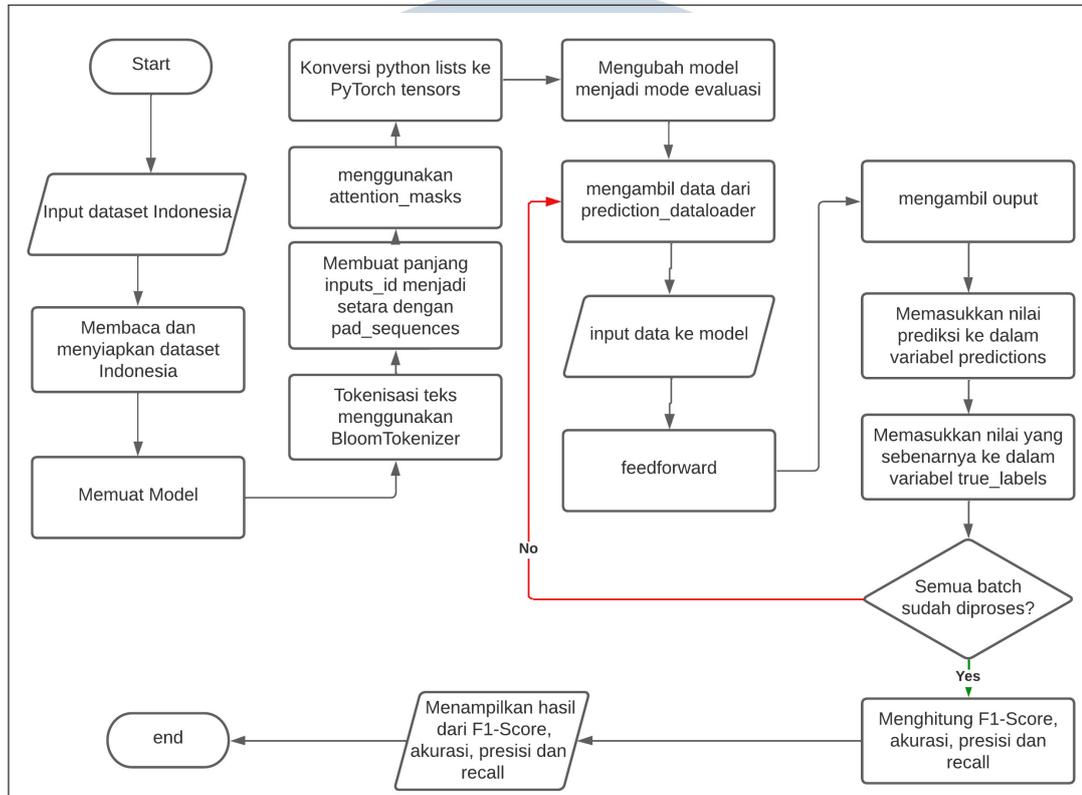
3.2.3 Model Validation



Gambar 3.3. Flowchart Model Validation

Pada Gambar 3.3, merupakan *flowchart* dari *model validation* serta lanjutan dari *flowchart model training*. Langkah pertama *flowchart* dari *model validation* adalah membuat model jadi dalam *evaluation mode*, gunanya untuk menghentikan perhitungan untuk *gradient*. Langkah selanjutnya, mengambil data dari *validation batches* dan data yang diambil akan dimasukan ke dalam model. Kemudian model akan melakukan *feedforward* untuk mendapatkan *output*. Selanjutnya, menampung hasil akurasi tiap *batches* ke dalam *temporary variable*. Langkah terakhir, setelah semua *batch* sudah diproses oleh model, maka total akurasi dari semua *batch* akan diperlihatkan.

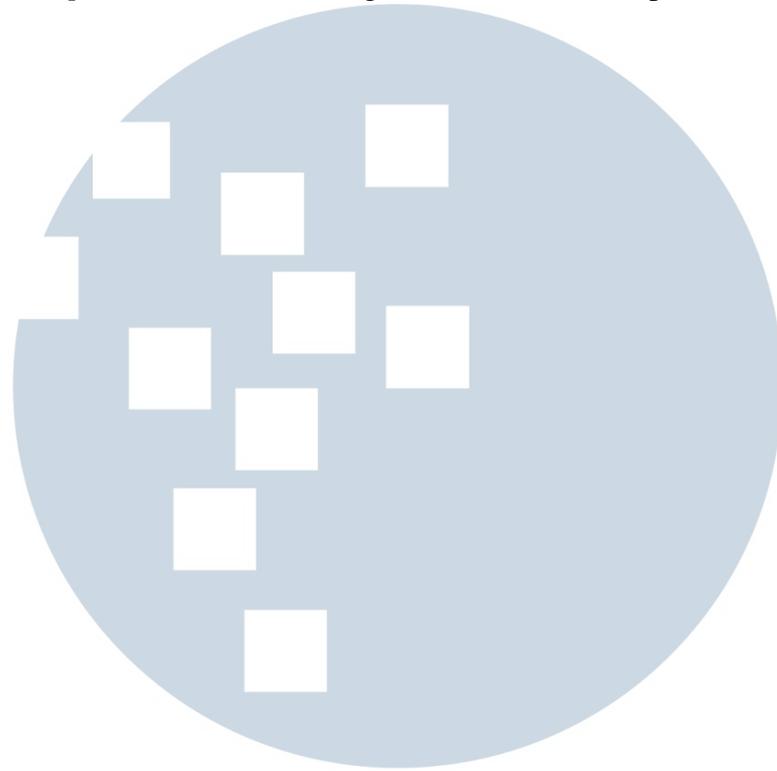
3.2.4 Evaluation



Gambar 3.4. Flowchart Evaluation

Pada Gambar 3.4, memperlihatkan *flowchart* dari evaluasi model. Karena model dilatih dengan *dataset* Inggris dan diuji dengan dataset Indonesia, maka sebelum *dataset* diuji perlu dilakukan tahap membaca dan menyiapkan untuk *dataset* Indonesia. Langkah kedua, memanggil model yang sudah dilakukan *fine tuning*. Selanjutnya *dataset* Indonesia akan ditokenisasi agar dapat dibaca oleh mesin. Kemudian, panjang token akan dibuat sama agar dapat dikonversi ke PyTorch tensor. Setelah itu, akan mengaplikasikan *attention mask* agar model dapat membedakan nilai token sebenarnya dan nilai dari *padding*. Lalu, nilai dari *attention mask*, token, dan label akan diubah menjadi PyTorch tensor supaya dapat disatukan dalam *prediction dataloader*. Langkah selanjutnya, data akan diambil dari *prediction dataloader*. Data tersebut nantinya akan dimasukkan ke dalam model sebagai input. Ketika model sudah mendapatkan input, proses *feedforward* akan dimulai untuk mendapatkan *outputnya* atau hasil dari prediksi mesin. Hasil prediksi dan *true value* akan ditampung ke dalam *variable* berbeda agar dapat dibandingkan. Proses ini akan berulang sampai semua *batch* sudah diproses. Jika semua *batch*

sudah diproses, maka akan dihitung hasil dari *F1 score* dan akurasi menggunakan *true value* dan *predicted value*. Langkah terakhir menampilkan *F1 score* dan akurasi.



UMMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA