BAB III
METODE PENELITIAN

3.1 Subjek dan Objek Penelitian

Penelitian memiliki tujuan menganalisis model IndoBERT untuk membangun chatbot yang mempermudah para petani dalam mengakses informasi mengenai hama dan penyakit tanaman secara cepat dan efisien. Subjek yang diambil sebagai fokus penelitian ini adalah buku berjudul "Hama dan Tanaman" yang diterbitkan oleh Yasayan Kita Menulis dan ditulis oleh Cheppy Wati, Arsi, Tili Karenina, Riyanto, dkk pada tahun 2021. Dipakainya buku tersebut karena mencakup banyak tentang penyakit dan hama yang terbaru dan sesuai dengan kondisi saat ini. Melalui penelitian ini, diharapkan informasi mengenai hama dan penyakit tanaman dapat lebih mudah diakses dan bermanfaat bagi para petani.

3.2 Alat dan Bahan

Untuk Penelitian ini digunakan alat dan bahan untuk membantu keberhasilan penelitian. Beberapa alat dan bahan yang dipakai:

3.2.1 Alat

Penelitian ini mengkategorikan alat yang digunakan menjadi dua bagian, yaitu perangkat keras (hardware) dan perangkat lunak (software). Berikut adalah spesifikasi dari perangkat keras yang digunakan:

1. Perangkat Keras *(Hardware)*
   a. Device : HP Pro-book G5
   b. Processor : Intel(R) Core(TM) i7-855U CPU @ 1.80GHz 1.99 GHz
   c. RAM : 16GB

Perangkat lunak yang digunakan adalah sebagai berikut :

2. Perangkat Lunak *(Software)*
   a. Sistem Operasi : *Windows 10 Enterprise 64-bit (10.0, Build 2261)*

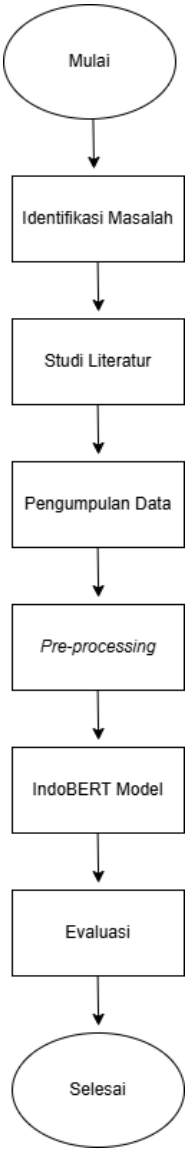
b. Bahasa Pemrograman : *Python*
c. Aplikasi : *Google Collab*

3.2.2 Bahan

Bahan yang digunakan dalam penelitian ini adalah dataset yang berasal dari buku berjudul "Hama dan Tanaman". Buku tersebut ditulis oleh Cheppy Wati, Arsi, Tili Karenina, Riyanto, Yogi Nirwanto Intan Nurcahya, Dewi Melani, Dwi Astuti, Dewi Septiarini, Sri Rezeki, Fransiska Purba, Evan Purnama Ramdan, dan Dwiwiyati Nurul pada tahun 2021. Dataset ini kemungkinan berisi informasi terkait hama dan penyakit tanaman yang digunakan dalam analisis dan penelitian yang dilakukan..

3.3 Diagram Alir Penelitian

Pada penyusunan laporan penelitian, terdapat tahapan-tahapan dalam melakukan penelitian. Penelitian diawali dengan identifikasi dan perumusan masalah, studi literatur, pengumpulan data, *preprocessing*, perancangan model, evaluasi model, dan di akhiri dengan prediksi jawaban ataupun respon dari chatbot dengan model yang memiliki performa paling baik dari skenario riset yang dilakukan. Diagram alir penelitian ini sebagai berikut :



Gambar 3. 1 Diagram alir penelitian

3.3.1 Identifikasi dan Perumusan Masalah

Penelitian ini diawali dengan proses identifikasi dan perumusan masalah yang diselesaikan. Fokus utama penelitian ini adalah pada pengembangan chatbot menggunakan model IndoBERT. Proses ini mencakup tahap identifikasi kebutuhan dan perumusan tujuan chatbot, serta merinci masalah-masalah yang diharapkan dapat diatasi oleh implementasi chatbot dengan menggunakan model IndoBERT. Dengan pendekatan ini, penelitian lebih terarah dan dapat memberikan solusi yang relevan dalam konteks penggunaan chatbot dengan teknologi IndoBERT.

3.3.2 Studi Literatur

Tahap selanjutnya yaitu studi literatur berkaitan dengan perumusun masalah yang ada. Studi literatur ini di dapatkan dari beragam sumber seperti, jurnal, buku, website, skripsi, atau sumber-sumber lain yang memiliki keterkaitan dengan permasalahan yang dibahas pada penelitian ini.

3.3.3 Pengumpulan Data

Tahap yang ketiga yaitu pengumpulan data. Data yang dipakai merupakan data public atau *open source* yang diambil dari sumber buku berjudul “Hama dan Tanaman Penyakit” terbitan Yayasan Kita Menulis pada tahun 2021. Data yang diambil merupakan sebagian dari 12 bab yang ada di buku dibagi menjadi beberapa bagian untuk mempermudah pembuatan dataset dalam format *JSON.*

Gambar 3. 2 hasil pengumpulan data

3.3.4 *Preprocessing*

Text preprocessing merupakan tahap penting dalam pengembangan model, alasannya karena data teks yang diambil melalui proses pengumpulan data tidak selalu berada dalam kondisi yang ideal atau terstruktur secara baik untuk diproses. Untuk beberapa algoritma, khususnya pada algoritma pembelajaran statistik dan probabilistik, keberadaan noise dan fitur yang tidak relevan dapat berdampak negatif kepada kinerja sistem. Oleh karena itu, diperlukan suatu proses preprocessing yang dapat mengubah data menjadi

lebih terstruktur melalui beberapa metode tertentu. Tujuan utama dari text preprocessing adalah mempersiapkan data teks agar dapat diolah dengan lebih efektif dan akurat dalam tahap selanjutnya, seperti tahap pembuatan model atau analisis. Metode-metode dalam *text preprocessing* mencakup tokenisasi, penghapusan *stopwords*, normalisasi teks, dan langkah-langkah lainnya yang bertujuan untuk meningkatkan kualitas dan kesesuaian data teks sebelum diaplikasikan ke dalam model. [10].

Pada Tahap *preprocessing* penelitian ini, dataset yang sudah dikumpulkan diolah terlebih dahulu untuk mempermudah pengolahan data pada model. Data yang diperoleh mencakup informasi sebagai berikut : Buku tersebut memiliki bagian yang terdiri dari 12 BAB dengan masing-masingnya terdapat penjelasan tentang hama dan penyakit tertentu. Data tersebut masih dalam bentuk data mentah dan harus diolah terlebih dahulu sebelum digunakan untuk melatih model. Berdasarkan data yang diperoleh, maka *preprocessing* yang dilakukan sebagai berikut :

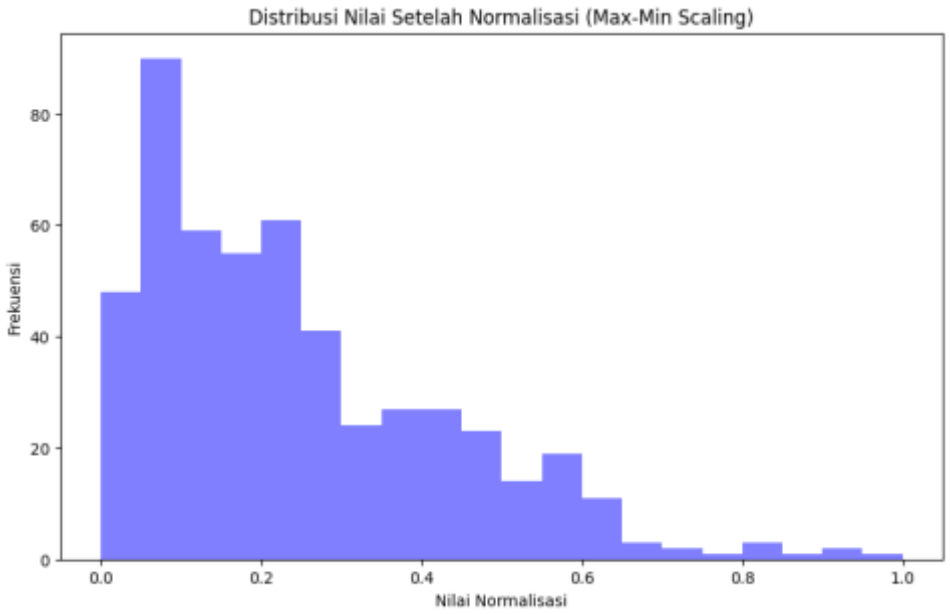
1. Mengubah dataset menjadi format json

   Proses ini ini dilakukan untuk membuat dataset yang sesuai dengan format yang dipakai untuk modeling, yaitu format json.

   Dalam setiap satu sampai dua paragraf yang terdapat di buku, menjadikan satu *context* dan lebih dari dua *questions* dan *answers.*

2. Normalisasi Dataset

   Normalisasi data dengan metode Max-Min Scaling adalah teknik yang digunakan untuk mengubah nilai-nilai data menjadi rentang tertentu, biasanya antara 0 dan 1. Cara kerjanya adalah dengan mengurangi nilai minimum dari setiap nilai data, kemudian membaginya dengan selisih antara nilai maksimum dan nilai minimum tersebut. Dengan demikian, setiap nilai data tertransformasi ke dalam rentang yang sama, yaitu antara 0 dan 1.

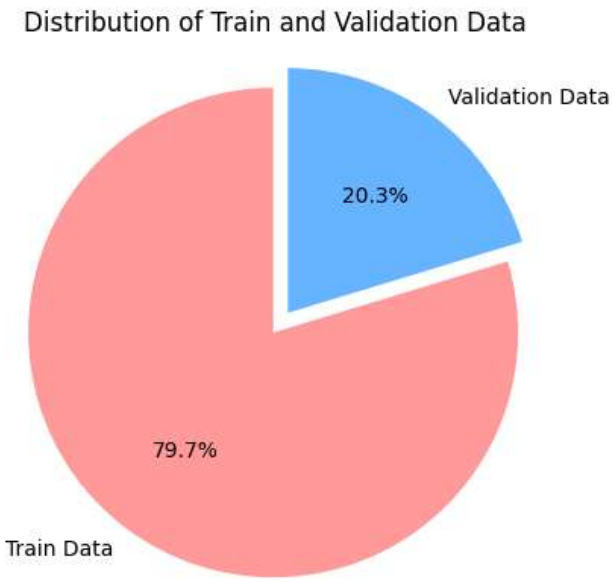


Gambar 3. 3 Distribusi Nilai Normalisasi

Misalkan kita memiliki sekumpulan data dengan nilai minimum minmin dan nilai maksimum maxmax. Manfaat dari normalisasi Max-Min Scaling adalah mengubah skala data sehingga perbedaan skala antara atribut-atribut tidak mengganggu proses pembelajaran pada model.

3. Split *Data Train* dan *Data Test*

   Dalam implementasi kode tersebut, proses pembagian data dilakukan untuk memisahkan dataset menjadi dua bagian utama: data pelatihan (train) dan data validasi (val). Pertama, dataset dimuat dan diorganisir dalam struktur data yang sesuai. Selanjutnya, konteks, pertanyaan, dan jawaban diekstrak dari struktur data tersebut dan disimpan dalam list terpisah, yaitu `texts`, `queries`, dan `answers`.



Gambar 3. 4 Diagram Pembagian Data

Dengan demikian jumlah data train 204 dan data validasi 52, setelah pembagian, dataset tersebut memiliki dua subset data yang independen untuk pelatihan dan evaluasi model, masing-masing mencakup konteks, pertanyaan, dan jawaban yang dibutuhkan.

4. Penentuan End Index

Dalam konteks pengembangan model untuk tugas identifikasi dan ekstraksi jawaban pada teks, penentuan end index merupakan langkah krusial dalam mencari posisi akhir dari jawaban yang benar. Proses ini dilakukan dengan memanfaatkan indeks awal jawaban (`start_idx`) yang telah diberikan dan teks lengkapnya (`text`).

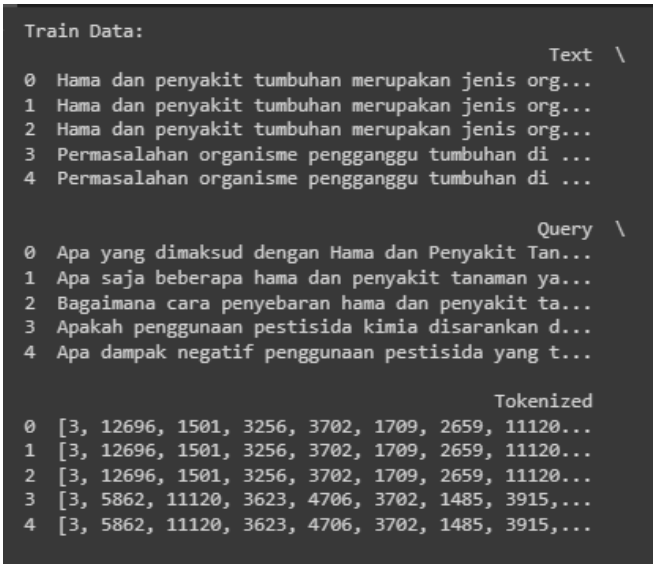
```
Contoh 1:
Teks: Hama dan penyakit tumbuhan merupakan jenis organisme pengganggu tumbuhan (
Jawaban sebenarnya: Hama dan penyakit tumbuhan merupakan jenis organisme penggan
Start index: 0
End index: 93

Contoh 2:
Teks: Hama dan penyakit tumbuhan merupakan jenis organisme pengganggu tumbuhan (
Jawaban sebenarnya: Belalang, lalat buah, ulat grayak, penyakit antaknose, fuso,
Start index: -383
```

Gambar 3. 5 Hasil Penentuan End Index

5. Tokenisasi

Tokenisasi merupakan proses mengubah kalimat input menjadi token berdasarkan kosa kata *(vocabulary)* yang telah ditentukan. Jika terdapat kata yang tidak ada dalam *vocabulary*, kata tersebut digantikan dengan kata dasar yang terdapat dalam *vocabulary*, dan bagian yang tidak dikenali diberi tanda pagar (#). Proses ini membantu mempersiapkan data teks untuk analisis atau pengolahan lebih lanjut dengan memecah kalimat menjadi unit-unit kecil yang dapat diolah oleh model atau algoritma secara lebih efektif. [11].



```
Train Data:
                                                Text  \
0  Hama dan penyakit tumbuhan merupakan jenis org...
1  Hama dan penyakit tumbuhan merupakan jenis org...
2  Hama dan penyakit tumbuhan merupakan jenis org...
3  Permasalahan organisme pengganggu tumbuhan di ...
4  Permasalahan organisme pengganggu tumbuhan di ...

                                               Query  \
0  Apa yang dimaksud dengan Hama dan Penyakit Tan...
1  Apa saja beberapa hama dan penyakit tanaman ya...
2  Bagaimana cara penyebaran hama dan penyakit ta...
3  Apakah penggunaan pestisida kimia disarankan d...
4  Apa dampak negatif penggunaan pestisida yang t...

                                           Tokenized
0  [3, 12696, 1501, 3256, 3702, 1709, 2659, 11120...
1  [3, 12696, 1501, 3256, 3702, 1709, 2659, 11120...
2  [3, 12696, 1501, 3256, 3702, 1709, 2659, 11120...
3  [3, 5862, 11120, 3623, 4706, 3702, 1485, 3915,...
4  [3, 5862, 11120, 3623, 4706, 3702, 1485, 3915,...
```

Gambar 3. 6 Hasil Tokenisasi

Hasil ataupun *Output* tokenisasi selanjutnya dirubah menjadi token ID dan *attention mask,* dan perlu diubah menjadi tensor. Pada penelitian ini, transformasi data menjadi tensor dilakukan menggunakan model RifkyIndoBERT-QA yang diperoleh dari perpustakaan Hugging Face *Transformer*s. Proses ini melibatkan konversi token ID dan *attention mask* menjadi struktur data tensor, yang biasanya didukung oleh *framework deep learning* seperti PyTorch atau TensorFlow. Dengan mengubah data ke dalam bentuk tensor, menjadikan data tersebut untuk lebih efisien

dikarenakan model dapat maksimal saat data sudah melalui proses tokenisasi.

6. Membuat Kelas Dataset

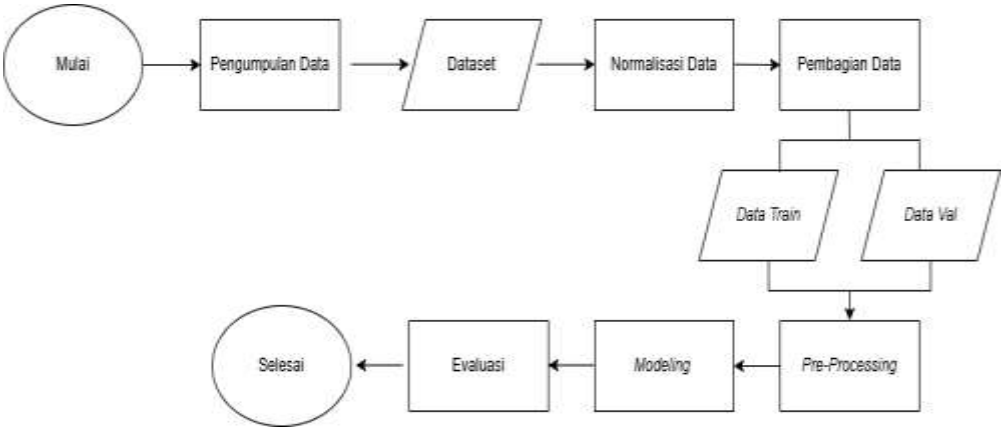
   Kelas `*SquadDataset*` dirancang sebagai objek dataset untuk mendukung pemrosesan dan pelatihan model *Question Answering* pada dataset *SQuAD (Stanford Question Answering Dataset).* Konstruktor menerima representasi tokenisasi dataset, seperti 'input_ids' dan 'attention_mask'. Metode `__getitem__` digunakan untuk mengambil item dari dataset pada indeks tertentu, dan penanganan khusus dilakukan untuk nilai-nilai `None` dalam representasi tokenisasi, diubah menjadi nilai '-1' agar dapat diatasi potensi masalah dalam pemrosesan. Metode `__len__` mengembalikan panjang total dataset, diukur berdasarkan panjang 'input_ids'. Keseluruhan implementasi ini dirancang untuk memudahkan pemrosesan dan pelatihan model Question Answering dengan memastikan integritas data pada tingkat dataset.

3.3.5 Implementasi IndoBERT Model

Pada tahap ini dimulai dengan memasukkan data yang dipakai, kemudian dilakukan pembagian data, selanjutnya dilakukan beberapa tahap pada *preprocessing* data seperti yang sudah dijelaskan di tahap sebelumnya. Semua data yang telah melalui tahap *preprocessing* disiapkan untuk melatih model pada tahap ini. Setelah data sudah siap digunakan untuk melatih model, tahap selanjutnya yaitu inisialisasi model IndoBERT.

Implemetasi dan model arsitektur merupakan tahapan untuk mentransformasikan rumusan konsep penelitian menjadi implementasi sistem yang dibangun oleh penulis. Setelah tahapan praproses data, dataset diubah menjadi input yang diterima oleh Rifky/Indobert-QA dalam bentuk vector representasi kata menggunakan Tokenizer. Kemudian dilakukan *fine-tuning*

saat kondisi ini *pre-train* model IndoBERT diadaptasi untuk melakukan prediksi jawaban.



Gambar 3. 1 Model Flowchart

Berdasarkan Gambar 3.7, terlihat bahwa proses pemodelan dilakukan setelah beberapa tahap pra-pemrosesan yang telah dilakukan sebelumnya. Setelah model berhasil dibangun dan dievaluasi, selanjutnya *dilakukan fine-tuning* atau penyesuaian untuk mencapai nilai parameter yang optimal. Skenario *fine-tuning* melibatkan beberapa perubahan parameter utama pelatihan dan penambahan *train data* . Parameter utama pembelajaran melibatkan *learning-rate* (tingkat pembelajaran yang menunjukkan seberapa banyak informasi yang diserap), *max length* (panjang maksimum teks yang disesuaikan), ukuran *batch* (jumlah data yang diproses dalam satu langkah), dan jumlah epoch (iterasi) yang disesuaikan melalui beberapa iterasi sampai konfigurasi parameter yang menghasilkan model terbaik ditemukan. Proses ini memungkinkan model untuk disesuaikan dan diperbaiki secara iteratif hingga mencapai performa yang optimal.

3.3.6 Evaluasi

Evaluasi model dilakukan untuk menghitung performa dari nilai prediksi yang dihasilkan oleh model IndoBERT dalam melakukan eksekusi. Dari semua eksperimen fine-tuning dan parameter yang sudah dilakukan, model terbaik

dihasilkan dengan pengaturan learning rate, batch size, max length, dan epoch yang optimal. Evaluasi kinerja model ini kemudian dilakukan menggunakan metrik F1-score, precision, dan recall. Precision mengukur akurasi model dalam mengidentifikasi contoh positif dari semua contoh yang diprediksi sebagai positif, sedangkan recall mengukur kemampuan model dalam menemukan semua contoh positif yang seharusnya diprediksi. F1-score, sebagai harmonic mean dari precision dan recall, memberikan ukuran yang seimbang antara keduanya, sehingga memberikan gambaran yang komprehensif tentang kinerja keseluruhan model dalam mendeteksi dan mengidentifikasi penyakit tanaman secara akurat. Dengan menggunakan metrik evaluasi ini, kita dapat memastikan bahwa model IndoBERT yang dihasilkan tidak hanya mampu memberikan prediksi yang akurat tetapi juga konsisten dalam performanya di berbagai situasi.