BAB 3
METODE PENELITIAN

Penelitian ini menggunakan metode atau algoritma yang disebut *Natural Language Processing* (NLP) dan *Long Short-Term Memory* (LSTM). Dalam penelitian ini, kedua pendekatan tersebut digabungkan dengan NLP berfungsi sebagai komponen dari proses penyisipan kata dan LSTM berfungsi sebagai sistem penyimpanan yang dapat memproses, dan mengkategorikan informasi data berdasarkan urutan waktu.

3.1 Bahan dan Alat Penelitian

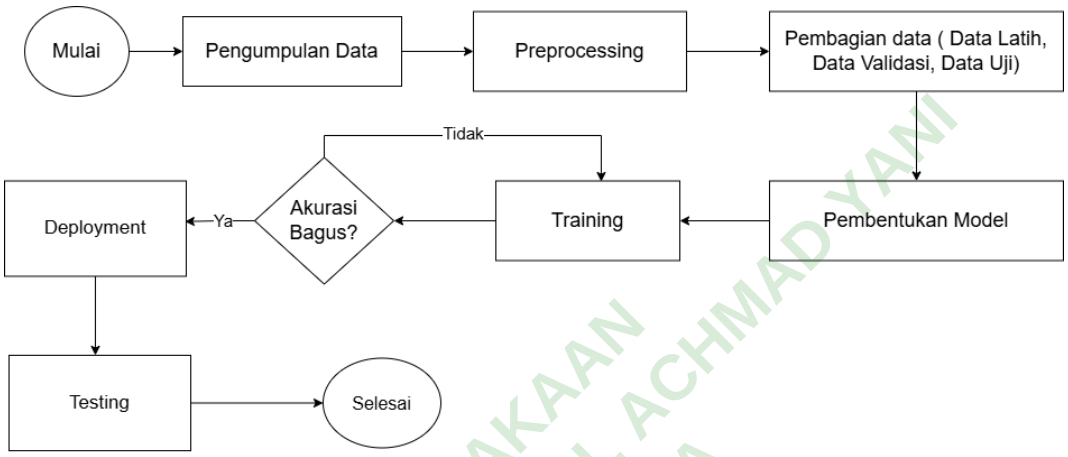
Bahan yang digunakan dalam penelitian ini adalah kumpulan data yang berkaitan dengan layanan mahasiswa di Fakultas Teknik dan Teknologi Informai UNJAYA. Data mencangkup bagian layanan seperti panduan MBKM, pembuatan transkrip nilai, pengajuan surat keterangan, prosedur pembayaran SPP, informasi mengenai tugas akhir, serta layanan-layanan lainnya yang diperlukan mahasiswa.

Alat yang digunakan dalam penelitian ini adalah laptop dengan spesifikasi yang memadai untuk menjalankan sistem operasi dan perangkat lunak pengembang, serta akses internet. Adapun sistem operasi dan *software* yang dipergunakan untuk menunjang penelitian ini sebagai berikut:

1. *Hardware*
    a. *Operating System* : Windows 11
    b. *Manufacture* : HP Laptop 14s-fq0xxx
    c. *Processor* : 8GB RAM
    d. *Memory* : AMD Athlon Gold 3150U
2. *Software*
    a. Python
    b. Google Collaboratory
    c. Visual Studio Code
    d. Streamlit

3.2 Jalan Penelitian

Penelitian ini mengikuti *Machine Learning Model Development Life Cycle* (MDLC) sebagai panduan untuk menjalankan proses penelitian. Garis besar tahapan dalam jalannya penelitian ini diperlihatkan dalam Gambar 3.1.



Gambar 3.1 *Flowchart* Jalan Penelitian

Pada Gambar 3.1 merupakan *flowchart* jalan penelitian, dimana pada diagram tersebut memiliki beberapa tahapan yang dilalui dari pengumpulan data hingga pembentukan model yang akan dijelaskan sebagai berikut:

1. Melakukan pengumpulan data dengan membuat data pertanyaan dan jawaban berdasarkan informasi dari Fakultas. Setelah data dikumpulkan, data disimpan dalam format JSON dan dikonversi menjadi *dataframe.*
2. Pada tahap *preprocessing* bertujuan untuk mengolah data yang telah konversi menjadi *dataframe* dengan melakukan *remove punctuation,* yaitu melakukan penghapusan tanda baca atau simbol khusus pada teks. Dalam proses ini, menghapus karakter seperti titik, koma, tanda tanya, dan simbol lainnya yang tidak memberikan makna tambahan pada teks.
3. Selanjutnya, setiap kata dalam teks tokenisasi diproses lebih lanjut dengan menghilangkan stopwords menggunakan pustaka NLTK. Kata-kata yang tersisa kemudian dilakukan stemming menggunakan Sastrawi untuk mengubahnya ke bentuk dasarnya.

4. Langkah berikutnya adalah encoding data menggunakan Word2Vec dari Gensim untuk menghasilkan representasi vektor dari kata-kata dalam teks dan membuat *embedding* matrix berdasarkan model Word2Vec yang telah dilatih.
5. Selanjutnya tahap *Tokenization* yang digunakan untuk memecah teks menjadi kata per kata. Pada proses ini, kalimat yang sudah melewati pembersihan teks akan masuk ke proses *tokenization* untuk mengubah kalimat teks menjadi urutan angka yang merepresentasikan per kalimat dan diisi agar memiliki panjang yang sama menggunakan pad_sequences.
6. Label data kemudian dilakukan *encode* menggunakan LabelEncoder dari scikit-learn dan diubah menjadi *one-hot encoding*. Untuk menangani ketidakseimbangan data, digunakan RandomOverSampler dari imblearn.
7. Selanjutnya *split* data. Pada tahap ini data yang sudah melalui pemrosesan teks akan dibagi menjadi data latih dan data uji dengan rasio 80:20 menggunakan *train_test_split* dari scikit-learn.
8. Setelah melalui pemrosesan teks dan pembagian data, lalu masuk ke pembentukan model. Dimana Model LSTM dibentuk dengan beberapa lapisan, termasuk lapisan *embedding* untuk memetakan kata-kata ke dalam vektor ruang, dua lapisan LSTM untuk memproses urutan input secara berurutan, lapisan *dropout* untuk mencegah *overfitting,* dan lapisan dense untuk menghasilkan *output*. Model dikompilasi menggunakan *optimizer* Adam dan *loss function categorical crossentropy*.
9. Model kemudian dilatih menggunakan data latih dengan pengaturan *epoch*, *batch size*, dan *callback* untuk *early stopping*, *model checkpoint*, dan *reduce learning rate*. Evaluasi model dilakukan dengan menggunakan *confusion matrix* untuk menghitung akurasi, presisi, dan recall dari model, serta mengukur hasil pelatihan dengan mengevaluasi model menggunakan data uji. Setelah itu, model yang telah dilatih disimpan dalam format h5.
10. Model diimplementasikan dalam aplikasi web menggunakan Streamlit. Aplikasi web ini terdiri dari satu halaman utama yang berisi bidang input

untuk pengguna mengajukan pertanyaan dalam bahasa Indonesia dan tombol "*Send*" untuk mengirim pertanyaan.

11. Pada tahap pengujian, model yang telah melalui pelatihan akan dinilai menggunakan metode Skala Likert yang terpisah untuk mengevaluasi tiga aspek yaitu fungsi, tampilan, dan hasil. Metode ini memungkinkan pengukuran yang sistematis terhadap sejauh mana model memenuhi tujuan yang ditetapkan, dengan mempertimbangkan efektivitas fungsionalitasnya, kejelasan tampilannya, serta relevansi dan kualitas hasil yang dihasilkan dari pengujian tersebut.