

BAB III

METODE PENELITIAN

3.1 Jenis Penelitian

Penelitian ini menggunakan pendekatan metode campuran (*mixed methods*) yang menggabungkan metode kualitatif dan kuantitatif secara sistematis. Pendekatan ini dipilih karena permasalahan yang dikaji tidak hanya membutuhkan analisis mendalam terkait kebutuhan sistem (kualitatif), tetapi juga memerlukan pengukuran performa model secara terukur melalui data numerik (kuantitatif).

Secara kualitatif, penelitian dilakukan melalui wawancara dan observasi terhadap staf administrasi Unit MBKM di Institut Informatika dan Bisnis Darmajaya. Hasil pengumpulan data ini digunakan untuk merumuskan kebutuhan pengguna, menyusun intent, menentukan struktur percakapan, serta menyusun dataset *chatbot*.

Secara kuantitatif, penelitian dilakukan melalui pengembangan dan pengujian model klasifikasi teks menggunakan algoritma Multinomial Naïve Bayes. Model yang dibangun dievaluasi menggunakan metrik accuracy, presisi, recall, dan F1-score berdasarkan confusion matrix. Selain itu, dilakukan pula User Acceptance Test (UAT) dengan melibatkan mahasiswa dan staf administrasi MBKM untuk mengukur tingkat kepuasan serta efektivitas *chatbot* dalam mendukung pelayanan administrasi.

Berdasarkan pendekatan tersebut, penelitian ini termasuk dalam kategori penelitian terapan (applied research), karena menghasilkan produk berupa *chatbot* berbasis Natural Language Processing (NLP) yang dirancang khusus untuk membantu pelayanan administrasi MBKM di IIB Darmajaya.

3.2 Sumber Data

3.2.1 Hasil Wawancara

Wawancara dilakukan dengan Bapak Dika Tondo Widakdo, S.Kom., M.T.I, selaku administrator MBKM di IIB Darmajaya. Hasil wawancara kemudian disimpulkan dalam bentuk tabel berikut:

Tabel 3. 1 Rekapitulasi Hasil Wawancara Administrator MBKM

| No | Pertanyaan | Temuan Utama | Statistik/Deskripsi |
|----|---|--|-------------------------------|
| 1 | Sejak kapan MBKM berjalan di IIB Darmajaya? | Resmi sejak 1 Juni 2020 | Data historis |
| 2 | Alur administrasi MBKM | Konsultasi PA → pilih program → konversi sesuai prodi | Proses bertahap per prodi |
| 3 | Media komunikasi utama | Instagram & WhatsApp dominan, Website untuk info satu arah | 2/3 via media sosial |
| 4 | Jumlah mahasiswa dilayani | ±160 mahasiswa/bulan | ±40/minggu → ±8/hari |
| 5 | Sistem yang digunakan | Website (info satu arah), Google Form & Drive | Belum ada aplikasi khusus |
| 6 | Beban komunikasi | Tingkat 8/10 (skala 1–10) | Tinggi |
| 7 | Kendala utama | Pertanyaan berulang (SKS, syarat, jadwal) | Puncak saat pendaftaran |
| 8 | Efisiensi layanan | Efisien dari sisi staf, kurang efektif untuk mahasiswa | Struktur organisasi sederhana |
| 9 | Potensi Chatbot | Bisa menjawab pertanyaan dasar (template) & kurangi beban | 100% mendukung integrasi |
| 10 | Kesediaan penggunaan chatbot | Bersedia menggunakan chatbot untuk pelayanan | Dukungan penuh |

Hasil wawancara menunjukkan bahwa layanan administrasi MBKM memiliki beban komunikasi yang tinggi, terutama karena adanya pertanyaan berulang mengenai SKS, syarat, dan jadwal. Media komunikasi dominan adalah Instagram dan WhatsApp, sedangkan website hanya digunakan untuk informasi satu arah. Dengan kondisi tersebut, *chatbot* dinilai berpotensi membantu meringankan beban kerja staf serta meningkatkan kecepatan dan akurasi pelayanan informasi MBKM.

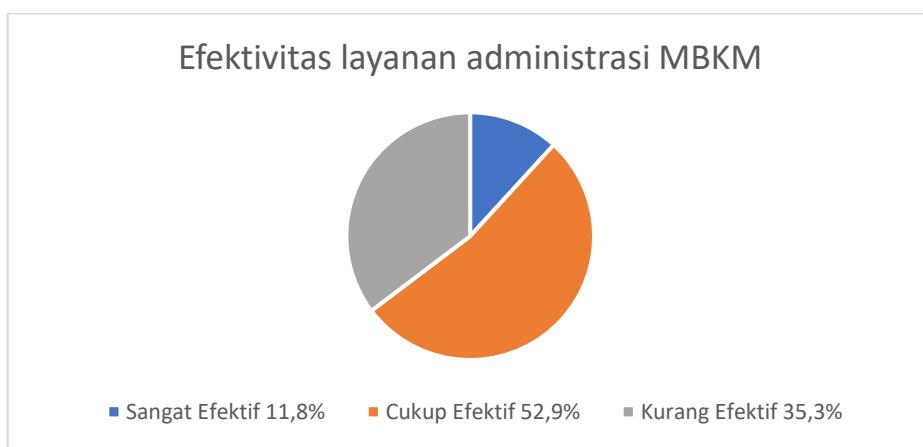
3.2.2 Hasil Kuisoner

Selain wawancara, penelitian ini juga menggunakan kuesioner untuk memperoleh data kuantitatif dari mahasiswa peserta MBKM tahun 2023 sampai 2024. Kuesioner diisi oleh 17 responden yang berasal dari berbagai program studi. Data diolah menggunakan statistik deskriptif (frekuensi dan persentase).

Pada penelitian ini, metode kuantitatif dipergunakan untuk mengetahui kecenderungan umum, pola, serta tingkat kebutuhan mahasiswa terhadap layanan administrasi MBKM melalui analisis data kuesioner. Analisis ini dilakukan dengan cara menghitung frekuensi dan persentase jawaban responden, sehingga memberikan gambaran objektif mengenai efektivitas layanan dan kebutuhan fitur *chatbot*. Sedangkan metode kualitatif yang menggunakan wawancara mendalam dipergunakan untuk menggali informasi yang lebih rinci terkait pengalaman, kendala, serta harapan mahasiswa maupun administrator MBKM. Dengan demikian, kombinasi kedua metode tersebut saling melengkapi, sehingga memberikan pemahaman yang lebih komprehensif mengenai kebutuhan sistem *chatbot* MBKM.

Tabel 3. 2 Efektivitas Layanan MBKM Saat Ini

| Kategori | Jumlah | Persentase |
|----------------|--------|------------|
| Sangat Efektif | 2 | 11,8% |
| Cukup Efektif | 9 | 52,9% |
| Kurang Efektif | 6 | 35,3% |



Gambar 3. 1 Efektifitas Layanan Administrasi MBKM

Tabel 3. 3 Kendala Mahasiswa dalam Administrasi MBKM

| Kendala | Jumlah | Persentase |
|------------------------------------|--------|------------|
| Sulit mendapatkan info pendaftaran | 10 | 58,8% |
| Proses pengajuan dokumen lama | 7 | 41,2% |
| Kurangnya respons admin | 7 | 41,2% |
| Tidak ada panduan yang jelas | 4 | 23,5% |
| Komunikasi tidak terpusat | 3 | 17,6% |

Tabel 3. 4 Fitur *Chatbot* yang Dibutuhkan Mahasiswa

| Fitur | Jumlah | Persentase |
|-----------------------------|--------|------------|
| Informasi jadwal & timeline | 14 | 82,3% |
| Tanya jawab seputar MBKM | 13 | 76,4% |
| Panduan pengisian dokumen | 11 | 64,7% |

Dengan demikian, data kuesioner mendukung hasil wawancara yang menunjukkan bahwa *chatbot* berpotensi membantu menjawab pertanyaan berulang, menyediakan panduan dokumen, serta memberikan informasi secara cepat dan terpusat.

3.3 Metode Pengembangan Life Cycle

Metode *Machine Learning Life Cycle* digunakan dalam penelitian ini sebagai pendekatan pengembangan model klasifikasi teks. *Life cycle* ini menggambarkan tahapan sistematis dalam membangun, melatih, dan mengevaluasi model *machine learning* hingga dapat diintegrasikan ke dalam sistem *chatbot* MBKM.

3.3.1 Model Requirements

Tahap *Model Requirements* merupakan langkah awal dalam *Machine Learning Life Cycle* yang bertujuan untuk mendefinisikan kebutuhan sistem sebelum proses pengembangan dimulai. Pada tahap ini dilakukan analisis terhadap jenis data, pemilihan *algoritma*, metode evaluasi kinerja, serta teknologi dan platform yang digunakan. Hasil analisis kebutuhan model ditunjukkan pada Tabel 3.1 berikut:

Tabel 3. 5 Analisis Kebutuhan Model *Chatbot* MBKM

| No | Requirements | Hasil Analisa |
|----|--------------------------|--|
| 1. | Jenis Data | Data yang digunakan berupa teks atau tipe data string dalam Bahasa Indonesia. Sumber data berasal dari SOP MBKM, buku pedoman program MBKM, FAQ, hasil wawancara dengan staf administrasi MBKM, dan hasil kuesioner dari 17 responden. Dataset disusun dalam format JSON yang memuat patterns dan responses untuk 20 intent. |
| 2. | Model Algoritma | Algoritma yang digunakan adalah Multinomial Naïve Bayes, dipilih karena efektif untuk klasifikasi teks pendek, mendukung multi-class classification, dan memiliki waktu pelatihan yang cepat |
| 3. | Pengukuran Kinerja Model | Model dievaluasi menggunakan confusion matrix dengan metrik akurasi, presisi, recall, dan F1-score untuk memastikan model bekerja sesuai tujuan yang diharapkan. |
| 4. | Platform dan Teknologi | Bahasa pemrograman yang digunakan adalah Python 3.10 dengan pustaka Scikit-learn, NLTK, dan Sastrawi. Framework API yang digunakan adalah Flask. Lingkungan pengembangan meliputi Google Colab dan laptop lokal, dengan tools pendukung seperti Visual Studio Code, dan GitHub. |

Berdasarkan hasil analisis pada Tabel 3.1, data yang digunakan dalam penelitian ini berbentuk teks dari sumber resmi dan valid, sehingga dapat merepresentasikan kebutuhan informasi layanan MBKM. Pemilihan algoritma Multinomial Naïve Bayes sesuai karena algoritma ini dirancang untuk klasifikasi teks pendek dengan banyak kelas. Evaluasi model menggunakan metrik standar machine learning (akurasi, presisi, recall, dan F1-score) agar performa dapat diukur secara objektif.

Sementara itu, teknologi dan platform yang digunakan dipilih untuk mendukung efektivitas pengembangan, pelatihan, hingga integrasi model ke dalam *chatbot* MBKM.

3.3.2 Data Collecting

Tahap Data Collection merupakan proses pengumpulan data yang akan digunakan dalam pelatihan model *chatbot*. Data yang dikumpulkan harus relevan, representatif, dan mencakup berbagai variasi pertanyaan yang mungkin diajukan oleh pengguna layanan administrasi MBKM. Dalam penelitian ini, proses pengumpulan data dilakukan melalui tiga teknik utama, yaitu:

- 1. Data Discovery**

Data discovery dilakukan dengan mengumpulkan data dari sumber resmi yang tersedia, seperti Standar Operasional Prosedur (SOP) MBKM, buku pedoman program MBKM, serta dokumen Frequently Asked Questions (FAQ) yang diterbitkan oleh Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi maupun oleh Institut Informatika dan Bisnis Darmajaya. Data ini menjadi pondasi utama untuk membangun basis pengetahuan *chatbot*.

- 2. Data Augmentation**

Data augmentation dilakukan dengan menambahkan data yang diperoleh langsung dari lapangan, yaitu melalui wawancara dengan staf administrasi MBKM dan penyebaran kuesioner kepada 17 responden yang terdiri dari mahasiswa serta pihak terkait. Tahap ini bertujuan memperluas cakupan dataset dan menyesuaikan isi *chatbot* dengan kebutuhan nyata pengguna.

- 3. Data Generation**

Data generation merupakan tahap pengolahan sekaligus penggabungan data hasil data discovery dan data augmentation sehingga terbentuk dataset yang lebih lengkap, konsisten, dan berkualitas. Pada tahap awal, dataset disusun dalam format CSV untuk memudahkan pengumpulan dan pengolahan. Setelah dataset final ditentukan, format tersebut diubah menjadi JSON dengan struktur yang memuat tag, patterns, dan responses. Format JSON dipilih karena fleksibel, mudah dibaca oleh manusia maupun mesin, serta

memiliki struktur ideal untuk merepresentasikan pasangan intent dan response pada *chatbot* berbasis NLP.

Dataset akhir terdiri dari 20 intent yang mencakup berbagai topik dan layanan terkait MBKM, seperti pendaftaran, konversi SKS, persetujuan prodi, jadwal program, hingga panduan administrasi.

3.3.3 Data Cleaning

Tahap Data Cleaning bertujuan untuk memastikan bahwa dataset yang digunakan memiliki kualitas yang baik, baik dari segi akurasi, kelengkapan, konsistensi, maupun integritas. Pada pemrosesan teks untuk Natural Language Processing (NLP), data cleaning merupakan langkah penting untuk menghilangkan noise dan menstandarkan format teks, sehingga data siap digunakan dalam proses pelatihan model machine learning.

Proses data cleaning dilakukan dengan tahapan berikut:

1. Case Folding

Seluruh teks diubah menjadi huruf kecil (*lowercase*) untuk menghindari perbedaan makna akibat penggunaan huruf kapital.

2. Penghapusan Karakter Khusus

Menghapus tanda baca, angka, dan simbol yang tidak relevan.

3. Stopword Removal

Menghapus kata-kata umum (stopwords) yang tidak memiliki bobot makna signifikan, seperti yang, dan, di, atau.

4. Stemming

Mengubah kata menjadi bentuk dasarnya (*root word*) menggunakan pustaka Sastrawi.

5. Pemeriksaan Duplikasi Data

Mengidentifikasi dan menghapus entri data yang sama atau memiliki makna yang identik untuk mencegah bias pada saat pelatihan model.

Hasil akhir dari tahap *data cleaning* adalah dataset yang bersih, terstandarisasi, dan siap digunakan pada tahap data labeling dan *feature engineering*. Dengan demikian,

kualitas data yang tinggi dapat membantu meningkatkan akurasi dan performa model *chatbot* berbasis NLP yang dikembangkan.

3.3.4 Data Labeling

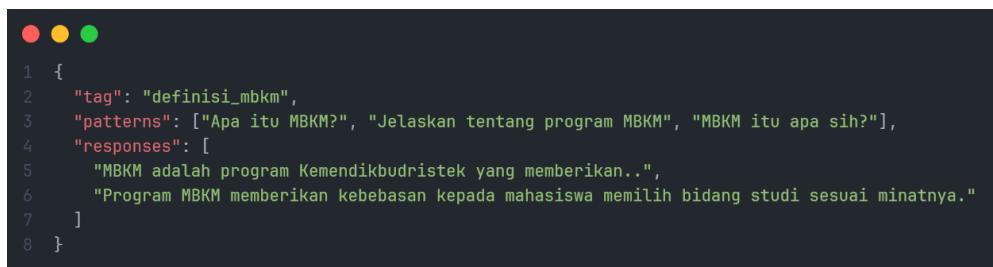
Tahap Data Labeling merupakan proses pemberian label atau kategori pada setiap data yang ada di dalam dataset, agar model *machine learning* dapat mengenali dan mengklasifikasikan pertanyaan sesuai dengan konteksnya. Label ini berfungsi sebagai penanda (*tag*) yang mengelompokkan pertanyaan-pertanyaan sejenis ke dalam satu kategori intent.

Dalam penelitian ini, proses data labeling dilakukan secara manual berdasarkan analisis terhadap isi pertanyaan dan jawaban yang telah dikumpulkan pada tahap data collection. Setiap intent diberi label yang menggambarkan topik tertentu, misalnya, definisi_mbkm, prosedur_pendaftaran, syarat_program, teknis_pelaporan, dan sebagainya.

Struktur data labeling disusun dalam format JSON dengan elemen sebagai berikut:

1. Tag label atau kategori intent yang mewakili topik pertanyaan.
2. Patterns daftar contoh pertanyaan yang kemungkinan diajukan pengguna.
3. Responses daftar jawaban yang disiapkan *chatbot* untuk merespons pertanyaan sesuai intent.

Contoh struktur data hasil pelabelan ditunjukkan pada gambar 3.2 berikut:



```
● ● ●
1 {
2   "tag": "definisi_mbkm",
3   "patterns": ["Apa itu MBKM?", "Jelaskan tentang program MBKM", "MBKM itu apa sih?"],
4   "responses": [
5     "MBKM adalah program Kemendikbudristek yang memberikan..",
6     "Program MBKM memberikan kebebasan kepada mahasiswa memilih bidang studi sesuai minatnya."
7   ]
8 }
```

Gambar 3. 2 Struktur Data JSON Tahap Labeling

Proses data labeling ini menghasilkan 20 intent yang mencakup berbagai topik seputar MBKM, mulai dari definisi program, prosedur pendaftaran, persyaratan, informasi teknis pelaksanaan, hingga panduan pelaporan. Hasil labeling inilah yang menjadi acuan utama pada proses feature engineering sehingga model dapat belajar mengklasifikasikan pertanyaan sesuai intent yang tepat.

3.3.5 Feature Engineering

Tahap Feature Engineering merupakan proses mengubah data teks yang telah dibersihkan dan diberi label menjadi representasi numerik agar dapat diproses oleh algoritma machine learning. Hal ini diperlukan karena model Multinomial Naïve Bayes tidak dapat langsung memahami teks, sehingga data harus dikonversi ke dalam bentuk vektor numerik.

Proses feature engineering dalam penelitian ini dilakukan melalui beberapa tahapan berikut:

1. Tokenization

Teks dipotong menjadi unit kata (token) untuk memudahkan analisis.

Contoh Input: “Apa syarat pendaftaran MBKM?”

Hasil Tokenisasi: [“apa”, “syarat”, “pendaftaran”, “mbkm”]

2. Normalisasi Teks

Setiap kata diubah ke bentuk baku berdasarkan hasil stemming dari tahap *data cleaning*.

Contoh: “pendaftaran” → “daftar” “mengikuti” → “ikut”

3. Stopword Removal

Kata umum yang tidak bermakna signifikan dihapus.

Contoh Input: [“apa”, “syarat”, “daftar”, “mbkm”]

Output: [“syarat”, “daftar”, “mbkm”]

4. TF-IDF Vectorization (Term Frequency – Inverse Document Frequency)

Setiap kata diubah menjadi bobot numerik berdasarkan frekuensinya dalam sebuah kalimat dibandingkan dengan seluruh dataset. Representasi TF-IDF memungkinkan model untuk memberikan bobot lebih pada kata yang unik dan relevan dalam menentukan intent.

3.3.6 Model Training

Tahap *Model Training* merupakan proses melatih model machine learning menggunakan data yang telah melalui tahap data labeling dan feature engineering. Tujuan tahap ini adalah agar model dapat mempelajari pola hubungan antara kata-kata pada pertanyaan (*patterns*) dan kategori topik (*intent*) yang sesuai, sehingga mampu melakukan klasifikasi pertanyaan baru secara akurat.

Proses model training dalam penelitian ini dilakukan melalui langkah-langkah berikut:

1. Pembagian Dataset

Dataset dibagi menjadi dua bagian, yaitu training set (80%) dan testing set (20%). Pembagian ini bertujuan memastikan bahwa model dievaluasi menggunakan data yang belum pernah dilihat sebelumnya, sehingga hasil evaluasi mencerminkan kemampuan model dalam menangani data baru (generalization).

2. Pemilihan Algoritma

Algoritma yang digunakan adalah Multinomial Naïve Bayes. Algoritma ini dipilih karena efektif untuk klasifikasi teks pendek, mendukung multi-class classification, serta efisien dari sisi waktu pelatihan.

3. Proses Pelatihan Model

Data masukan berupa matriks fitur hasil dari proses TF-IDF Vectorization digunakan sebagai input, sedangkan label intent digunakan sebagai target output. Model mempelajari distribusi kata pada setiap intent untuk menghitung probabilitas kemunculannya, sehingga dapat memprediksi intent yang sesuai untuk pertanyaan baru.

4. Pengaturan Hyperparameter

Untuk meningkatkan performa model, dilakukan penyesuaian hyperparameter seperti nilai alpha pada Laplace smoothing untuk menghindari pembagian nol pada kata yang jarang muncul, serta pengaturan ukuran n-gram pada TF-IDF untuk mempertimbangkan kombinasi kata yang lebih kompleks.

5. Report Model Classification

Hasil prediksi dibandingkan dengan data uji menggunakan beberapa metrik evaluasi, yaitu:

- a. Akurasi: Tingkat ketepatan model.
- b. Precision: Ketepatan dalam mengklasifikasikan kelas tertentu.
- c. Recall: Kemampuan model menemukan semua data yang relevan.
- d. F1-Score: Harmoni antara Precision dan Recall.

6. Classification Validation

Classification Validation proses pengecekan tambahan untuk memastikan kinerja model stabil dan tidak hanya berlaku pada subset data tertentu. Pada penelitian ini digunakan metode validasi sebagai berikut:

a. Confusion Matrix

Digunakan untuk membandingkan nilai aktual (label sebenarnya) dengan hasil prediksi model.

b. Kurva Roc (*Receiver Operating Characteristic*)

Memvisualisasikan performa model dengan memplot *True Positive Rate (TPR)* terhadap *False Positive Rate (FPR)* pada berbagai nilai ambang batas. Nilai *AUC (Area Under Curve)* yang mendekati 1 menunjukkan performa klasifikasi yang sangat baik.

c. Cross validation

Dataset dibagi menjadi k lipatan, di mana setiap lipatan digunakan sekali sebagai data validasi dan $k-1$ kali sebagai data latih. Teknik ini digunakan untuk menguji kestabilan kinerja model secara menyeluruh.

7. Penyimpanan Model

Setelah pelatihan selesai, model yang telah terlatih disimpan dalam format .pkl menggunakan pustaka joblib atau pickle. Penyimpanan ini memungkinkan model digunakan kembali pada tahap implementasi (operation) tanpa perlu dilakukan pelatihan ulang.

3.3.7 Model Evaluation

Tahap Model Evaluation dilakukan untuk memastikan bahwa model machine learning yang dibangun telah sesuai dengan tujuan penelitian, yaitu mampu mengklasifikasikan pertanyaan pengguna ke dalam intent yang tepat pada layanan *chatbot* MBKM.

Evaluasi dilakukan dengan cara membandingkan hasil prediksi model pada data uji (testing set) dengan label intent yang telah ditentukan sebelumnya dalam dataset. Misalnya, apabila sebuah pertanyaan seperti “Apa itu MBKM?” diprediksi oleh

algoritma Multinomial Naïve Bayes sebagai intent definisi_mbkm, maka hasil prediksi tersebut dibandingkan dengan label sebenarnya yang ada pada dataset.

Selain evaluasi perbandingan langsung, kinerja model juga diukur menggunakan beberapa metrik evaluasi yang dihitung melalui confusion matrix, yaitu:

- a. Akurasi → mengukur tingkat ketepatan prediksi model secara keseluruhan.
- b. Presisi → mengukur ketepatan model dalam mengklasifikasikan data pada suatu kelas tertentu.
- c. Recall → mengukur kemampuan model menemukan seluruh data yang relevan pada suatu kelas.
- d. F1-score → rata-rata harmonis antara presisi dan recall, yang memberikan penilaian lebih seimbang pada data yang tidak seimbang.

Penggunaan confusion matrix serta metrik-metrik tersebut memberikan gambaran yang lebih komprehensif mengenai kinerja model, baik secara keseluruhan maupun pada setiap kategori intent. Dengan demikian, hasil evaluasi dapat digunakan sebagai dasar untuk menilai efektivitas model dan menentukan kebutuhan perbaikan atau pengembangan lebih lanjut.

3.3.8 Operation

Tahap operation dalam siklus Machine Learning merupakan proses implementasi model yang telah selesai dilatih dan dievaluasi agar dapat digunakan secara langsung oleh pengguna. Pada penelitian ini, operation mencakup tiga aspek utama yaitu environment (lingkungan), deployment (penerapan), serta monitoring dan updating (pemantauan dan pembaruan).

- a. Environment (Lingkungan)

Environment mengacu pada infrastruktur dan tools yang digunakan untuk mengeksekusi model machine learning. Pada penelitian ini, environment meliputi bahasa pemrograman Python 3.10, pustaka utama seperti Scikit-learn, NLTK, dan Sastrawi, framework Flask sebagai API, serta hardware berupa laptop dengan spesifikasi memadai untuk menjalankan proses training dan deployment. Pengaturan environment yang tepat diperlukan agar model dapat berjalan stabil tanpa error.

b. Deployment (Penerapan)

Tahap deployment merupakan proses memindahkan model dari lingkungan pengembangan ke lingkungan produksi. Pada penelitian ini, model klasifikasi intent *chatbot* MBKM yang telah disimpan dalam format .pkl diintegrasikan dengan aplikasi *chatbot* menggunakan Flask API. *Chatbot* kemudian dihubungkan dengan antarmuka berbasis web sehingga mahasiswa maupun administrator MBKM dapat berinteraksi secara langsung. Deployment dilakukan pada server lokal, dengan kemungkinan pengembangan lebih lanjut menggunakan cloud server (misalnya Google Cloud Platform) agar dapat diakses secara luas.

c. Monitoring/Updating (Pemantauan/Pembaruan)

Setelah model di-deploy, dilakukan pemantauan kinerja *chatbot* secara berkala. Monitoring bertujuan memastikan *chatbot* tetap mampu memberikan jawaban yang relevan terhadap pertanyaan mahasiswa. Selain pemantauan teknis, dilakukan pula evaluasi berbasis feedback pengguna melalui User Acceptance Test (UAT). Apabila ditemukan adanya penurunan akurasi atau muncul pertanyaan baru yang belum dikenali, maka model perlu diperbarui (updating) dengan menambahkan dataset baru, melakukan retraining, serta mengoptimalkan kembali performa *chatbot*.