



PEMBANGUNAN CHATBOT INTERAKTIF DENGAN MENGGUNAKAN ALGORITMA NAIVE BAYES

Nimatul Mamuriyah¹, Haeruddin², Hero^{3*}

¹Program Studi Teknologi Informasi, Universitas Internasional Batam, Batam

Baloi-Sei Ladi, Jl. Gajah Mada, Tiban Indah, Kec. Sekupang, Kota Batam, Kepulauan Riau 29426,

e-mail: ¹ nimatul@uib.ac.id, ² haeruddin@uib.ac.id, ³ 2132008.hero@uib.edu

* coresspondence

ARTICLE INFO

Article history:

Received 10 Oktober 2024

Received in revised form 18 Oktober 2024

Accepted 8 Desember 2024

Available online 10 Desember 2024

ABSTRACT

Chatbots have become integral components of modern digital services, facilitating efficient and responsive interactions between users and technology. As artificial intelligence (AI) continues to shape the way businesses engage with customers, the development of interactive chatbots is increasingly essential for improving user experience. This study focuses on the implementation of the Naive Bayes algorithm to build an interactive chatbot using Python. The Naive Bayes algorithm, a simple yet powerful machine learning model, is designed to classify user input based on identified keywords or text patterns. By leveraging this algorithm, the chatbot can provide accurate and relevant responses that align with user queries. The primary goal of this study is to enhance chatbot functionality by utilizing Naive Bayes for improved classification and response generation. Python is chosen as the programming language due to its flexibility, extensive library support, and robust capabilities in handling text processing tasks. The study demonstrates that the Naive Bayes algorithm is not only efficient in terms of implementation but also highly effective in delivering personalized and contextually appropriate interactions. Additionally, it emphasizes the algorithm's speed and simplicity, making it a valuable tool for creating responsive and intelligent chatbots. Ultimately, the findings of this research suggest that Naive Bayes plays a crucial role in advancing chatbot technology by improving the accuracy and quality of responses in real-time communication.

Keywords: *Chatbot, Naive Bayes, Python, Text Classification, Responsiveness.*

Abstrak

Chatbots telah menjadi komponen integral dari layanan digital modern, memfasilitasi interaksi yang efisien dan responsif antara pengguna dan teknologi. Karena kecerdasan buatan (AI) terus membentuk cara bisnis berinteraksi dengan pelanggan, pengembangan chatbot interaktif semakin penting untuk meningkatkan pengalaman pengguna. Penelitian ini berfokus pada implementasi algoritma Naive Bayes untuk membangun chatbot interaktif menggunakan Python. Algoritma Naive Bayes, sebuah model pembelajaran mesin yang sederhana namun kuat, dirancang untuk mengklasifikasikan input pengguna berdasarkan kata kunci atau pola teks yang teridentifikasi. Dengan memanfaatkan algoritma ini, chatbot dapat memberikan respons yang akurat dan relevan yang selaras dengan pertanyaan pengguna. Tujuan utama dari penelitian ini adalah untuk meningkatkan fungsionalitas chatbot dengan memanfaatkan Naive Bayes untuk meningkatkan klasifikasi dan menghasilkan respons. Python dipilih sebagai bahasa pemrograman karena fleksibilitasnya, dukungan pustaka yang luas, dan kemampuannya yang kuat dalam menangani tugas-tugas pemrosesan teks. Penelitian ini menunjukkan bahwa algoritma Naive Bayes tidak hanya efisien dalam hal implementasi, tetapi juga sangat efektif dalam memberikan interaksi yang dipersonalisasi dan sesuai dengan konteks. Selain itu, penelitian ini menekankan kecepatan dan kesederhanaan algoritme, menjadikannya alat yang berharga untuk menciptakan chatbot yang responsif dan cerdas. Pada akhirnya, temuan penelitian ini menunjukkan bahwa Naive Bayes memainkan peran penting dalam memajukan teknologi chatbot dengan meningkatkan akurasi dan kualitas respons dalam komunikasi real-time.

Kata Kunci: Chatbot, Naive Bayes, Python, Klasifikasi Teks, Responsivitas.

1. PENDAHULUAN

Keberadaan chatbot telah menjadi bagian yang semakin umum dalam kehidupan sehari-hari. Mereka hadir dalam berbagai sektor, mulai dari aplikasi perbankan hingga layanan pelanggan online. Misalnya, dalam pemesanan online, chatbot digunakan untuk membantu pelanggan dalam menemukan produk yang mereka cari, memberikan rekomendasi bahkan menangani proses pembayaran. Keberadaan chatbot tidak hanya memudahkan interaksi antara manusia dan teknologi, tetapi juga menghadirkan solusi yang efisien dalam memenuhi kebutuhan pengguna secara cepat dan responsif. Percakapan yang terjadi antara komputer dengan manusia merupakan bentuk respons dari program yang telah di deklarasikan pada database program [1]. Chatbot menjadi penjaga pintu utama yang mempertemukan pengguna dengan layanan yang mereka butuhkan. Namun, tidak semua chatbot diciptakan sama. Ada yang mampu menciptakan interaksi yang memikat dan memuaskan pengguna, sementara yang lain terasa kaku dan tidak bermakna. Di balik chatbot yang sukses terdapat rahasia dari penerapan algoritma Naive Bayes, dimana respons yang dihasilkan merupakan hasil pemindaian kata kunci pada input user dan menghasilkan respons balasan yang di anggap paling cocok, atau pola kata yang dianggap paling mendekati [1,2]. Algoritma Naive Bayes membawa interaksi antara manusia dan mesin yang dapat memahami dan menjawab pesan. Algoritma ini telah terbukti efektif dalam menganalisis teks, sehingga memungkinkan chatbot untuk merespons dengan cepat dan akurat terhadap pertanyaan atau perintah yang diberikan oleh pengguna. [3].

Chatbot akan lebih mudah dan lebih cepat dibangun dengan menggunakan algoritma Naive Bayes. Naive Bayes adalah sebuah metode klasifikasi dalam ilmu data yang berdasarkan teorema Bayes dengan asumsi bahwa semua fitur dalam data adalah independen satu sama lain. Dalam konteks chatbot, hal ini berarti bahwa chatbot dapat menggunakan informasi dari kata-kata atau fitur lain dalam teks untuk memprediksi kategori atau label yang paling sesuai untuk pesan yang diterima [4]. Kemampuan untuk beroperasi dengan cepat dan efisien, bahkan dengan dataset yang besar merupakan salah satu keuntungan utama dari naive bayes. Namun, algoritma ini juga memiliki kelemahan terkait dengan asumsi bahwa setiap fitur adalah independen satu sama lain, yang mungkin tidak selalu benar dalam konteks data teks yang kompleks. Dengan demikian, naive bayes merupakan algoritma yang kuat dan serbaguna dalam klasifikasi teks dan dapat digunakan dalam berbagai jenis layanan termasuk pengembangan chatbot [5].

Adapun tujuan dari penelitian ini adalah untuk menyelidiki dan mengimplementasikan penggunaan algoritma Naive Bayes dalam pengembangan chatbot interaktif menggunakan bahasa pemrograman Python. Dengan mengintegrasikan Naive Bayes ke dalam struktur chatbot, kami bertujuan untuk meningkatkan responsivitas dan ketepatan chatbot dalam memahami dan merespons pesan-pesan yang diberikan oleh pengguna. Selain itu, kami juga berupaya untuk mengeksplorasi potensi algoritma Naive Bayes dalam

memperbaiki pengalaman pengguna dengan menyediakan respon yang lebih personal dan sesuai dengan konteks.

Python adalah bahasa pemrograman tingkat tinggi yang dipilih dalam penelitian ini karena kemudahan penggunaannya, dukungan pustaka yang luas untuk pemrosesan teks dan pembelajaran mesin, serta komunitas yang besar dan aktif yang menyediakan beragam sumber daya dan dukungan. Selain itu, fleksibilitas Python yang lintas-platform membuatnya cocok untuk digunakan dalam berbagai lingkungan pengembangan, memungkinkan pengembang untuk dengan mudah mengimplementasikan algoritma Naive Bayes dan membangun chatbot yang responsif tanpa hambatan [6].

2. TINJAUAN PUSTAKA

Algoritma Naive Bayes adalah metode klasifikasi yang berbasis probabilitas dan berdasarkan pada teorema Bayes. Ini digunakan untuk mengklasifikasikan data ke dalam kategori atau kelas tertentu berdasarkan fitur-fitur yang diberikan. Misalnya, dalam konteks pengembangan chatbot, algoritma Naive Bayes dapat digunakan untuk mengklasifikasikan pesan-pesan yang diterima dari pengguna ke dalam kategori topik tertentu, seperti pertanyaan tentang cuaca, pertanyaan tentang berita, atau permintaan bantuan.

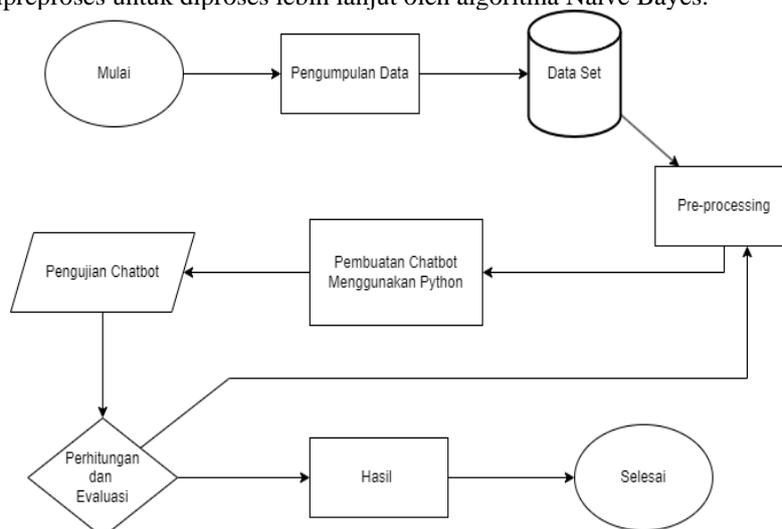
Dalam penelitian ini, pemilihan bahasa pemrograman Python dan lingkungan pengembangan Visual Studio Code untuk mengimplementasikan chatbot menggunakan algoritma Naive Bayes. Pemilihan Python karena kemudahan penggunaannya, dukungan yang luas untuk pemrosesan teks dan pembelajaran mesin, serta komunitas yang besar dan aktif. Visual Studio Code dipilih sebagai lingkungan pengembangan karena kemudahan penggunaannya, dukungan yang kuat untuk Python, dan berbagai fitur yang mendukung pengembangan perangkat lunak.

3. METODOLOGI PENELITIAN

Dalam Penelitian ini, pendekatan eksperimen untuk menguji efektivitas penggunaan algoritma Naive Bayes dalam pengembangan chatbot responsif. Pendekatan ini dipilih karena menguji kinerja chatbot dan mengumpulkan data empiris untuk mendukung kesimpulan penelitian.

3.1 Proses Pengembangan Chatbot

Proses pengembangan chatbot melibatkan beberapa langkah, termasuk persiapan data, pemrosesan teks, implementasi algoritma Naive Bayes, dan evaluasi performa chatbot. Misalnya, dalam tahap persiapan data, data latih diperlukan untuk melatih chatbot, sedangkan dalam tahap pemrosesan teks, teks-teks masukan dari pengguna diproses untuk diproses lebih lanjut oleh algoritma Naive Bayes.



Gambar 1. Proses Pengembangan Chatbot

3.1.1 Pengumpulan Dataset

Pengumpulan dataset merupakan langkah krusial dalam membangun chatbot berbasis Naive Bayes. Proses ini dimulai dengan mengumpulkan berbagai pertanyaan yang mungkin diajukan oleh pengguna serta jawaban yang sesuai. dataset harus mencakup berbagai pertanyaan yang sering diajukan terkait produk maupun stok.

Tabel 1. Pengumpulan Dataset

Pertanyaan	Jawaban
FAG 6000 ZZE	Bearing FAG 6000 ZZE Code Name : IT10003 Actual Stock: 1
FAG 6000 2NSE	Bearing FAG 6000 2NSE Code Name : IT10004 Actual Stock: 3
Nachi 6001 ZZE	Bearing Nachi 6001 ZZE Code Name : IT10005 Actual Stock: 2
Nachi 6001 2NSE	Bearing Nachi 6001 2NSE Code Name : IT10006 Actual Stock: 8
FAG 6001 ZZE	Bearing FAG 6001 ZZE Code Name : IT10007 Actual Stock: 11
FAG 6001 2NSE	Bearing FAG 6001 2NSE Code Name : IT10008 Actual Stock: 8

Setelah tabel 1 pengumpulan dataset selesai dikumpulkan dan diorganisir, data tersebut siap digunakan untuk melatih model Naive Bayes. Dataset dibagi menjadi dua subset yaitu data latih dan data uji.

a. Data Latih

Data latih adalah kumpulan pertanyaan dan jawaban yang sudah terdaftar dalam knowledge base. Data ini digunakan untuk memberikan jawaban otomatis kepada pengguna atau pembeli berdasarkan data yang telah disediakan sebelumnya.

Dengan kata lain, Ketika pengguna atau pembeli menanyakan sesuatu yang sudah terdaftar di dalam knowledge base, chatbot akan langsung memberikan jawaban yang sesuai. Contohnya, jika pembeli bertanya tentang stok barang tertentu, seperti "Nachi 6001 ZZE", chatbot akan memberikan respons seperti: "Bearing Nachi 6001 ZZE Code Name: IT10005 Actual Stock: 2." Dengan data latih ini, chatbot dirancang untuk memberikan jawaban yang cepat dan akurat kepada pembeli, meningkatkan efisiensi layanan.

b. Data Uji

Data uji digunakan khusus oleh pemilik chatbot untuk mengetahui data apa saja yang belum terdaftar di dalam knowledge base. Fungsinya adalah untuk mengidentifikasi kekurangan dalam knowledge base dan memperbaharainya agar chatbot menjadi lebih lengkap.

Ketika ada pertanyaan baru yang tidak terdaftar di dalam knowledge base, chatbot akan memberi tahu bahwa jawaban belum tersedia, misalnya dengan pesan:

"Bot: I don't know the answer. Can you teach me?". Dalam situasi ini, hanya pemilik chatbot yang memiliki akses untuk memperbarui knowledge base. Pemilik dapat memberikan jawaban baru melalui antarmuka chatbot, sehingga pertanyaan yang sebelumnya tidak terjawab akan disimpan di dalam knowledge base dan menjadi bagian dari data latih di masa mendatang.

Tujuan dari data uji ini untuk adalah memastikan bahwa chatbot terus berkembang dan menjadi lebih cerdas. Pemilik dapat menggunakannya untuk menambahkan data yang lebih relevan atau memperbaiki respons yang kurang sesuai.

3.1.2 Pembuatan Chatbot Menggunakan Python

Platform yang saya gunakan untuk membuat chatbot adalah Visual Studio Code (VS Code). VS Code merupakan editor code sumber yang dibuat oleh Microsoft, yang beroperasi pada sektor desktop dan mendukung beberapa bahasa pemrograman.

a. Import Library

```

1 import json
2 from difflib import get_close_matches
3 import tkinter as tk
4 from tkinter import scrolledtext, simpledialog

```

Gambar 2. Import Library

- 1) Import json : Mengimpor pustaka JSON untuk membaca dan menyimpan data dalam format JSON.
 - 2) From difflib import get_close_matches: Mengimpor fungsi get_close_matches dari pustaka difflib. Fungsi ini digunakan untuk menemukan string yang paling mirip dalam daftar berdasarkan algoritma pengukuran kesamaan.
 - 3) Import tkinter as tk: Mengimpor pustaka Tkinter untuk membuat antarmuka grafis (GUI).
 - 4) From tkinter import scrolledtext, simpledialog:
 - a) scrolledtext: Widget teks dengan scrollbar.
 - b) simpledialog: Digunakan untuk meminta input sederhana dari pengguna melalui dialog pop-up.
- b. Load_knowledge_base

```

6 def load_knowledge_base(file_path: str) -> dict:
7     with open(file_path, 'r') as file:
8         data: dict = json.load(file)
9     return data

```

Gambar 3. load_knowledge_base

- 1) Fungsi load_knowledge_base adalah untuk membaca file JSON yang berisi basis pengetahuan chatbot.
 - 2) file_path: str: Parameter fungsi berupa path file JSON dalam bentuk string.
 - 3) with open(file_path, 'r') as file: Membuka file dalam mode baca ('r') dan menutupnya secara otomatis setelah selesai.
 - 4) json.load(file): Membaca konten file JSON dan mengonversinya menjadi dictionary Python.
 - 5) return data: Mengembalikan dictionary yang berisi basis pengetahuan.
- c. Save_knowledge_base

```

11 def save_knowledge_base(file_path: str, data: dict):
12     with open(file_path, 'w') as file:
13         json.dump(data, file, indent=2)

```

Gambar 4. save_knowledge_base

- 1) Fungsi save_knowledge_base adalah untuk menyimpan basis pengetahuan chatbot ke file JSON.
 - 2) file_path: str: Parameter berupa lokasi file yang akan ditulis.
 - 3) data: dict: Dictionary berisi data yang akan disimpan.
 - 4) with open(file_path, 'w') as file: Membuka file dalam mode tulis ('w') dan menutupnya secara otomatis.
 - 5) json.dump(data, file, indent=2): Menyimpan dictionary ke file JSON dengan indentasi 2 spasi.
- d. Find_best_match

```

15 def find_best_match(user_question: str, questions: list[str]) -> str | None:
16     matches: list = get_close_matches(user_question, questions, n=1, cutoff=0.8)
17     return matches[0] if matches else None

```

Gambar 5. find_best_match

- 1) Fungsi find_best_match adalah untuk mencari pertanyaan di basis pengetahuan yang paling mirip dengan input pengguna.
- 2) user_question: str: Pertanyaan yang diajukan pengguna.
- 3) questions: list[str]: Daftar pertanyaan dari basis pengetahuan.
- 4) get_close_matches:
 - a) user_question : Input pengguna.
 - b) Questions : Daftar referensi.

- c) $n=1$: Mengembalikan satu hasil terbaik.
- d) $\text{cutoff}=0.8$: Ambang batas kesamaan (80%).
- 5) `return matches[0]` if matches else None: Jika ada hasil yang cocok, kembalikan elemen pertama; jika tidak, kembalikan None.

e. `Get_answer_for_question`

```

19 def get_answer_for_question(question: str, knowledge_base: dict) -> str | None:
20     for q in knowledge_base["questions"]:
21         if q["question"] == question:
22             return q["answer"]

```

Gambar 6. `get_answer_for_question`

- 1) Fungsi `get_answer_for_question` adalah untuk mendapatkan jawaban dari basis pengetahuan berdasarkan pertanyaan.
- 2) `question: str`: Pertanyaan yang ingin dicari.
- 3) `knowledge_base: dict`: Basis pengetahuan.
- 4) `for q in knowledge_base["questions"]`: Melakukan iterasi pada setiap item di `questions`.
- 5) `if q["question"] == question`: Membandingkan apakah pertanyaan cocok.
- 6) `return q["answer"]`: Mengembalikan jawaban jika cocok; jika tidak, kembalikan None secara implisit.

f. Chatbot GUI

```

24 class ChatBotGUI:
25     def __init__(self, root):
26         self.root = root
27         self.root.title("ChatBot")
28         self.root.configure(bg="#2c3e50")
29
30         self.knowledge_base = load_knowledge_base("knowledge_base.json")

```

Gambar 7. Chatbot GUI

- 1) `ChatBotGUI`: Kelas untuk mengatur antarmuka chatbot.
- 2) `__init__`: Konstruktors kelas.
- 3) `self.root`: Jendela utama GUI (objek Tkinter).
- 4) `self.root.title`: Menentukan judul jendela.
- 5) `self.root.configure(bg="#2c3e50)`: Mengatur warna latar belakang (#2c3e50 adalah warna abu-abu gelap).

g. Membuat Komponen GUI

```

self.chat_window = scrolledtext.ScrolledText(self.root, wrap=tk.WORD, state='disabled',
self.chat_window.pack(padx=10, pady=10)

self.user_input = tk.Entry(self.root, width=50, bg="#ecf0f1", fg="#2c3e50", font=('Aria
self.user_input.pack(padx=10, pady=10)
self.user_input.bind("<Return>", self.process_input)

self.send_button = tk.Button(self.root, text="Send", command=self.process_input, bg="#2
self.send_button.pack(padx=10, pady=10)

```

Gambar 8. Komponen GUI

- 1) `self.chat_window`:
 - a) Widget teks dengan scrollbar untuk menampilkan dialog.
 - b) `wrap=tk.WORD`: Membungkus teks per kata.
 - c) `state='disabled'`: Membuat widget hanya-baca.
- 2) `self.user_input`:
 - a) Widget untuk input pengguna.
 - b) `bind("<Return>", self.process_input)`: Menjalankan fungsi saat tombol Enter ditekan.
- 3) `self.send_button`:
 - a) Tombol untuk mengirim input.
 - b) `command=self.process_input`: Menjalankan fungsi `process_input` saat tombol ditekan.

h. Mencari dan Menambahkan Jawaban

```

best_match = find_best_match(user_text, [q["question"] for q in self.knowledge_base["questions"]])

if best_match:
    answer = get_answer_for_question(best_match, self.knowledge_base)
    self.chat_window.config(state='normal')
    self.chat_window.insert(tk.END, f'Bot: {answer}\n')
    self.chat_window.config(state='disabled')
else:
    self.chat_window.config(state='normal')
    self.chat_window.insert(tk.END, "Bot: I don't know the answer. Can you teach me?\n")
    self.chat_window.config(state='disabled')
    new_answer = simpledialog.askstring("Teach me", "Type the answer or 'skip' to skip:")

```

Gambar 9. Mencari dan Menambahkan jawaban

- 1) `find_best_match`: Mencocokkan pertanyaan pengguna dengan basis pengetahuan.
- 2) `simpledialog.askstring`: Meminta input jawaban dari pengguna jika tidak ada jawaban yang diketahui.

i. Main Program

```

77 if __name__ == '__main__':
78     root = tk.Tk()
79     bot_gui = ChatBotGUI(root)
80     root.mainloop()

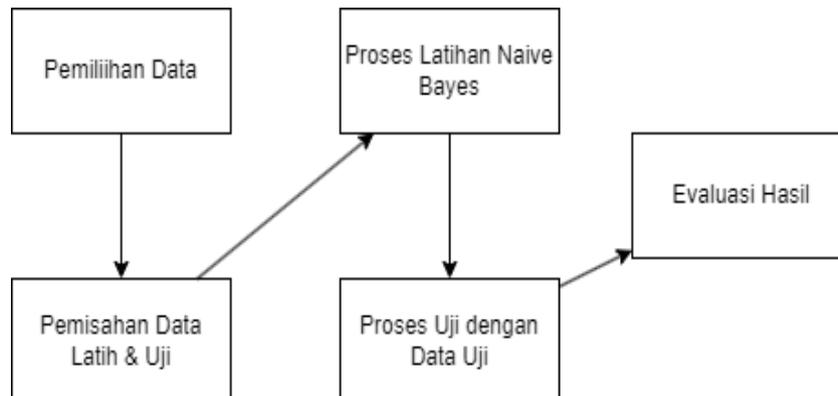
```

Gambar 10. Main Program

- 1) `if __name__ == '__main__':`: Mengeksekusi kode hanya jika file ini dijalankan langsung.
- 2) `root = tk.Tk()`: Membuat jendela utama Tkinter.
- 3) `ChatBotGUI(root)`: Menginisialisasi GUI chatbot.
- 4) `root.mainloop()`: Memulai loop utama GUI untuk menangani event.

3.1.3 Pengujian Chatbot

Dalam tahap pengujian chatbot, langkah-langkah yang diambil dalam menguji efektivitas chatbot yang dikembangkan dengan algoritma Naive Bayes akan dijelaskan secara rinci. Pengujian chatbot ini mencakup pemilihan dataset, metrik evaluasi, dan prosedur pengujian yang akan dilakukan untuk mengukur kinerja chatbot.



Gambar 11. Pengujian Chatbot

Pemilihan dataset menjadi langkah awal dalam desain eksperimen. Dataset yang digunakan harus mencakup berbagai jenis pesan yang mungkin diterima oleh chatbot, serta label atau kategori yang sesuai dengan setiap pesan. Prosedur pengujian harus dirancang untuk menguji kinerja chatbot dengan menggunakan dataset yang telah dipilih. Prosedur ini melibatkan tahap pelatihan chatbot menggunakan data latih, pengujian chatbot menggunakan data uji yang terpisah.

4. HASIL DAN PEMBAHASAN

Dalam penelitian ini, mengembangkan sebuah chatbot dengan kemampuan untuk memahami dan merespon pertanyaan pengguna berdasarkan model Naive Bayes. Algoritma Naive Bayes dipilih karena

kesederhanaannya, kecepatan dalam pelatihan, dan kemampuannya untuk memberikan hasil yang cukup baik dalam klasifikasi teks.

4.1 Deskripsi Implementasi

Dalam tahap implementasi, sistem chatbot ini dirancang dengan fokus pada kemampuan untuk menjawab pertanyaan secara interaktif dan memperbarui knowledge base berdasarkan input pengguna. Proses implementasi dimulai dengan pembuatan antarmuka pengguna menggunakan Tkinter, sebuah framework GUI yang sederhana dan ringan pada Python. Antarmuka ini dirancang agar intuitif, sehingga pengguna dapat dengan mudah berinteraksi dengan chatbot. Area input disediakan untuk mengetikkan pertanyaan, sedangkan area output menampilkan jawaban dari chatbot.

Chatbot ini memanfaatkan algoritma Naive Bayes untuk mencocokkan pertanyaan pengguna dengan data yang ada dalam knowledge base. Algoritma ini dipilih karena sifatnya yang efisien dalam menangani klasifikasi teks dan kesederhanaannya untuk diterapkan pada dataset kecil seperti knowledge base chatbot. Knowledge base disimpan dalam format JSON, yang berisi daftar pasangan pertanyaan dan jawaban. Format ini dipilih karena fleksibel, mudah dibaca, dan diubah baik oleh manusia maupun mesin.

Sistem ini juga dirancang dengan kemampuan pembelajaran berbasis input pengguna. Ketika chatbot menerima pertanyaan baru yang tidak ada dalam knowledge base, ia meminta pengguna untuk memberikan jawaban. Jawaban ini kemudian secara otomatis disimpan dalam knowledge base, memperluas kemampuan chatbot dalam menjawab pertanyaan serupa di masa depan. Proses pembelajaran ini memastikan bahwa chatbot dapat terus berkembang tanpa memerlukan pembaruan manual pada knowledge base oleh pengembang.

4.1.1 Hasil Pengumpulan Data

Pada penelitian ini, data yang berhasil dikumpulkan untuk melatih dan menguji performa chatbot berbasis algoritma Naive Bayes terdapat seribu (1.000) data. Dataset terdiri dari kumpulan pertanyaan dan jawaban yang relevan dengan domain aplikasi chatbot yang sedang dikembangkan. Dalam hal ini, dataset berisi informasi tentang stok barang, nama barang, dan kode barang.

```
C:\Users > HP > knowledge_base.json > ...
1
2 {
3   "questions": [
4     {
5       "question": "hallo",
6       "answer": "Hallo!"
7     },
8     {
9       "question": "apa kabar?",
10      "answer": "baik!"
11    },
12    {
13      "question": "Machi L500 10.00mm",
14      "answer": "no.code:12345 sisa stock : 2"
15    },
16    {
17      "question": "Machi 6000 ZZE",
18      "answer": "Bearing Machi 6000 ZZE Code Name : IT10001, Actual Stock: 12"
19    },
20    {
21      "question": "Machi 6000 2NSE",
22      "answer": "Bearing Machi 6000 2NSE Code Name : IT10002\ Actual Stock: 12"
23    },
24    {
25      "question": "FAG 6000 2NSE",
26      "answer": "Bearing FAG 6000 2NSE Code Name : IT10004\ Actual Stock: 3"
27    },
28    {
29      "question": "FAG 6000 2NSE",
30      "answer": "Bearing FAG 6000 2NSE Code Name : IT10004\ Actual Stock: 3"
31    }
32  ]
33 }
```

Gambar 12. Hasil Pengumpulan Dataset

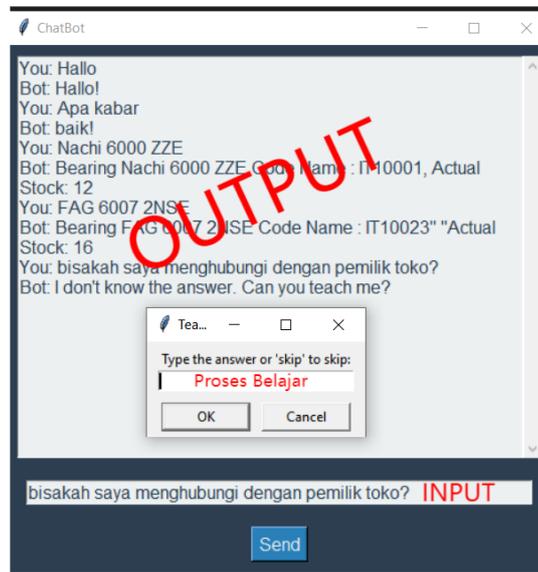
Struktur Dataset yang digunakan dalam penelitian ini diatur dalam bentuk tabel dengan dua kolom utama yaitu:

- Pertanyaan (Question) : Berisi pertanyaan yang mungkin diajukan oleh pengguna
- Jawaban (Answer) : Berisi jawaban yang sesuai dengan pertanyaan tersebut.

4.2 Hasil Implementasi

4.2.1 Antarmuka Pengguna

Antarmuka pengguna chatbot dirancang sederhana namun fungsional. Berikut adalah fitur yang tersedia:



Gambar 13. GUI Chatbot

- Area Input: Pengguna dapat mengetik pertanyaan di sini.
- Area Output: Chatbot menampilkan respons di area ini.
- Tombol "Send": Untuk mengirim pertanyaan.
- Proses Belajar: Jika chatbot tidak mengetahui jawaban, ia akan meminta pengguna untuk memberikan jawaban.

4.2.2 Proses Pembelajaran Chatbot

Terdapat mekanisme khusus untuk membedakan antara data latih dan data uji. Contoh pemisahan Data Latih dan Data Uji:

```

C:\Users\HP > HP > | knowledge_base.json > ...
  2  "questions": [
  3  |
  4  |
  5  |
  6  |
  7  |
  8  |
  9  |
 10  |
 11  |
 12  |
 13  |
 14  |
 15  |
 16  |
 17  |
 18  |
 19  |
 20  |
 21  |
 22  |
 23  |
 24  |
 25  |
 26  |
 27  |
 28  |
 29  |
 30  |
 31  |
 32  |
 33  |
 34  |
 35  |
 36  |
 37  |
 38  |
 39  |
 40  |
 41  |
 42  |
 43  |
 44  |
 45  |
 46  |
 47  |
 48  |
 49  |
 50  |
 51  |
 52  |
 53  |
 54  |
 55  |
 56  |
 57  |
 58  |
 59  |
 60  |
 61  |
 62  |
 63  |
 64  |
 65  |
 66  |
 67  |
 68  |
 69  |
 70  |
 71  |
 72  |
 73  |
 74  |
 75  |
 76  |
 77  |
 78  |
 79  |
 80  |
 81  |
 82  |
 83  |
 84  |
 85  |
 86  |
 87  |
 88  |
 89  |
 90  |
 91  |
 92  |
 93  |
 94  |
 95  |
 96  |
 97  |
 98  |
 99  |
100 |

```

Gambar 14. Data Latih

Data yang sudah terkumpul berupa pertanyaan dan jawaban yang sudah terdaftar dalam knowledge base, yang dapat langsung digunakan oleh pengguna (seperti pembeli) untuk mendapatkan jawaban.

```

C: > Users > HP > {} knowledge_base.json > [ ] questions > {} 0 > answer
2  "questions": [
8  ],
9  {
10  "question":
11  "Hi",
12  "answer":
13  "Hey :-), thanks for visiting, how can I help?"
14  },
15  {
16  "question": "Bye",
17  "answer":
18  "Thanks for visiting, Bye! Come back again soon."
19  },
20  {
21  "question": "Thanks",
22  "answer": "Happy to help!"
23  },
24  {
25  "question": "Nachi L500 10.00mm",
26  "answer": "no.code:12345 sisa stock : 2"
27  },

```

Gambar 15. Data Uji

Ketika pengguna mengajukan pertanyaan yang tidak ada dalam knowledge base, chatbot akan menampilkan pesan seperti: "Bot: I don't know the answer. Can you teach me?". Selanjutnya, pengguna dapat memberikan jawaban baru. Jawaban tersebut akan disimpan dalam knowledge_base.json.

Salah satu aspek penting dalam implementasi ini adalah integrasi antara algoritma pencocokan teks menggunakan `difflib.get_close_matches` untuk menemukan pertanyaan yang paling sesuai dengan input pengguna. Jika pertanyaan tidak cocok dengan tingkat kesesuaian tertentu, chatbot akan menampilkan pesan bahwa ia tidak mengetahui jawabannya dan meminta pengguna untuk mengajarkannya. Pendekatan ini memberikan pengalaman yang lebih dinamis dan personal kepada pengguna.

Dengan demikian, data yang dikumpulkan dalam dataset ini akan digunakan untuk melatih dan menguji chatbot berbasis Naive Bayes, memastikan bahwa chatbot dapat memberikan jawaban yang tepat dan responsif terhadap pertanyaan pengguna.

4.3 Pengujian Sistem

4.3.1 Pengujian Data Latih

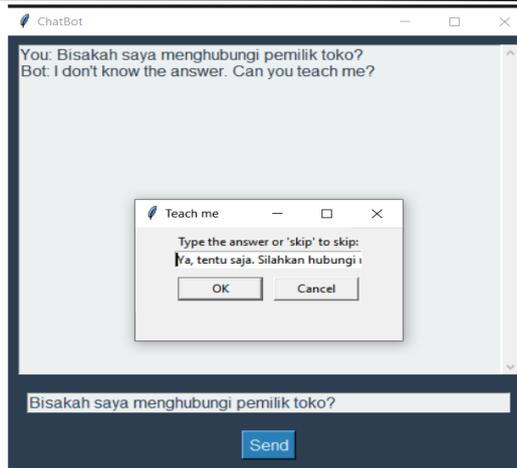
Tabel 2. Pengujian data latih

Pertanyaan	Jawaban Chatbot	Status
Hi	Hey :-), thanks for visiting, how can I help?	Benar
Bye	Thanks for visiting, Bye! Come back again soon.	Benar
Thanks	Happy to help!	Benar
FAG 6002 ZZE	Bearing FAG 6002 ZZE Code Name : IT10011 Actual Stock: 5	Benar
FAG 6002 2NSE	Bearing FAG 6002 2NSE Code Name : IT10012 Actual Stock: 4	Benar
Nachi 6026 ZZE	Bearing Nachi 6026 ZZE Code Name : IT10099 Actual Stock: 12	Benar
FAG 6028 ZZE	-	Salah
Nachi 6030 ZZE	Bearing Nachi 6030 ZZE Code Name : IT10107 Actual Stock: 31	Benar
Nachi 6152 2NSE	Bearing Nachi 6152 2NSE Code Name : IT10408 Actual Stock: 325	Benar

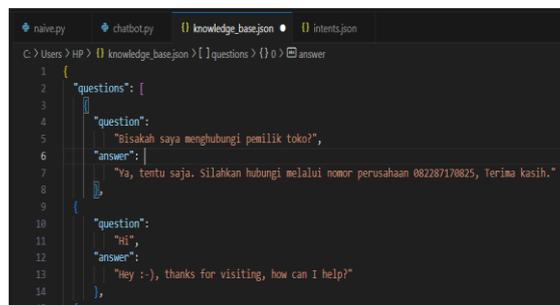
Tujuan dari pengujian data latih ini untuk menguji chatbot dengan pertanyaan yang sudah ada di knowledge base. Hasil pengujian chatbot mampu memberikan jawaban yang akurat yaitu 90% benar.

4.3.2 Pengujian Data Uji

Tujuan data uji adalah untuk menguji chatbot dengan pertanyaan baru yang belum ada di knowledge base.



Gambar 16. Input Pertanyaan baru



Gambar 17. Setelah ditambahkan

Gambar 16 menunjukkan bahwa Pemilik ingin menambahkan pertanyaan baru yaitu “Bisakah saya menghubungi pemilik toko?”, Pada gambar 17 membuktikan bahwa Chatbot merespons dan meminta Pemilik untuk memberikan jawaban untuk pertanyaan tersebut dan menyimpannya di knowledge base.

4.4 Evaluasi Kinerja Chatbot

Tahap pengujian dan evaluasi merupakan bagian penting dalam penelitian ini untuk memastikan bahwa chatbot berfungsi sesuai dengan tujuan yang telah ditetapkan. Pengujian dilakukan dengan dua pendekatan, yaitu menggunakan data latih dan data uji. Data latih adalah kumpulan pertanyaan dan jawaban yang telah tersedia di dalam knowledge base, sementara data uji adalah pertanyaan baru yang belum ada dalam knowledge base. Kedua jenis data ini membantu menilai kemampuan chatbot dalam menjawab pertanyaan serta kemampuan pembelajaran chatbot saat menghadapi pertanyaan baru.



Gambar 18. Akurasi Respons

Evaluasi kinerja sistem dilakukan berdasarkan metrik seperti tingkat akurasi jawaban dan waktu respons. Tingkat akurasi dinilai berdasarkan jumlah pertanyaan yang dijawab dengan benar dibandingkan total pertanyaan yang diajukan. Berdasarkan pengujian, tingkat akurasi chatbot pada data latih mencapai 90%, sedangkan untuk data uji 85%, chatbot menunjukkan kemampuan adaptasi yang cukup baik setelah beberapa iterasi pembelajaran. Waktu respons chatbot rata-rata berada di bawah 1 detik, menunjukkan bahwa sistem mampu memproses pertanyaan secara efisien.

4.5 Kelebihan dan Kekurangan Sistem

Kelebihan Sistem:

- a. Chatbot mampu menjawab pertanyaan-pertanyaan yang sudah ada di knowledge base dengan akurasi yang sangat baik.
- b. Waktu yang dibutuhkan untuk memproses pertanyaan dan memberikan jawaban rata-rata di bawah 1 detik.
- c. Chatbot ini dapat memperbarui knowledge base secara langsung melalui masukan pengguna.
- d. Pemisahan data latih dan data uji memungkinkan pemilik chatbot untuk melakukan analisis kualitas secara berkala.
- e. Struktur JSON pada knowledge base mempermudah pemilik dalam mengedit, menambah, atau menghapus data.
- f. Dengan menggunakan algoritma pencocokan, chatbot dapat memahami pertanyaan yang memiliki variasi kecil dalam struktur atau ejaan, sehingga lebih toleran terhadap kesalahan input.

Kekurangan Sistem:

- a. Ketika ada beberapa pertanyaan yang sangat mirip, algoritma pencocokan dapat salah memberikan jawaban, terutama jika tingkat kemiripan antara beberapa pertanyaan tinggi.
- b. Meskipun dapat menangani variasi kecil dalam ejaan, chatbot memiliki keterbatasan untuk memahami pertanyaan dengan kesalahan ejaan yang signifikan atau struktur kalimat yang tidak teratur.
- c. Meskipun user-friendly, antarmuka berbasis Tkinter terbatas pada fungsi dasar dan tidak mendukung fitur modern seperti pengenalan suara, notifikasi, atau integrasi multimedia.
- d. Ketika pertanyaan tidak ditemukan, chatbot meminta masukan pengguna setiap kali pertanyaan tersebut diajukan ulang, yang dapat menjadi kurang efisien bagi pengguna.

5. KESIMPULAN DAN SARAN

Dari hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa chatbot interaktif berbasis algoritma Naive Bayes yang telah dikembangkan mampu menjawab pertanyaan pengguna secara efektif berdasarkan data yang terdapat di dalam knowledge base. Chatbot ini dilengkapi dengan fitur pembelajaran dinamis, di mana pengguna dapat menambahkan data baru ketika pertanyaan tidak ditemukan dalam sistem, sehingga chatbot mampu terus berkembang dan menyesuaikan dengan kebutuhan operasional. Selain itu, implementasi chatbot ini memberikan solusi praktis dalam menjawab pertanyaan terkait produk, seperti ketersediaan stok dan kode barang, yang membantu meningkatkan efisiensi dan kenyamanan bagi pengguna. Pemisahan data latih dan data uji juga menjadi keunggulan utama sistem ini, memungkinkan pemilik untuk mengelola dan mengevaluasi data dengan lebih baik guna memastikan bahwa semua informasi yang relevan terintegrasi dengan sempurna dalam knowledge base. Penelitian ini menunjukkan bahwa algoritma Naive Bayes dapat menjadi pendekatan yang andal dalam membangun chatbot yang responsif, fleksibel, dan dapat dikembangkan lebih lanjut sesuai dengan kebutuhan pengguna.

DAFTAR PUSTAKA

- [1] Ajiz, M. F., Ramadan, M. F. S., Mutia, H. D., & Januari, P. D. (2023). Pengembangan Aplikasi Chatbot Informasi Akademik Berbasis Web Menggunakan Metode Artificial Intelligence Markup Language (AIML). *Media Jurnal Informatika*, 15(2), 143-148.
- [2] Putra, G. P. M., & Tenriawaru, A. (2023, November). Rancang Bangun Virtual Assistant Chatbot Menggunakan Node. Js pada Layanan Sistem Informasi Akademik. In *Prosiding Seminar Nasional Pemanfaatan Sains dan Teknologi Informasi* (Vol. 1, No. 1, pp. 345-352).
- [3] Medvedyk, A., Lohoida, M., Rybchak, Z., & Kulyna, O. (2023). IT Slang: Development of Telegram Chatbot. In *COLINS* (2) (pp. 152-162).

-
- [4] Berrar, Daniel. "Bayes' Theorem and Naive Bayes Classifier." (2019): 403-412.
- [5] Setyawan, Muhammad Yusril Helmi, Rolly Maulana Awangga, and Safif Rafi Efendi. "Comparison of multinomial naive bayes algorithm and logistic regression for intent classification in chatbot." *2018 International Conference on Applied Engineering (ICAE)*. IEEE, 2018..
- [6] Raj, Sumit, Karkal Raj, and Karkal. *Building chatbots with Python*. Apress, 2019.
- [7] Astriyani, M., Baihaqi, W. M., & Widiawati, C. R. A. (2023, November). Development of Chatbot Features for Stunting Education Using Artificial Neural Network Algorithm. In *2023 IEEE 7th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)* (pp. 1-5). IEEE.
- [8] Saputra, E. D., Harahap, N. S., Jasril, J., & Yusra, Y. (2024). Question Answering Al-Qur'an Menggunakan Generative Pre-Trained Transformer 3.5 Berbasis Chatbot Telegram. *Jutisi: Jurnal Ilmiah Teknik Informatika dan Sistem Informasi*, 13(1).
- [9] Putri, Marjuniati, Daslan Josua Valentino, and Roni Andarsyah. *CARA PRAKTIS MEMBUAT CHATBOT WHATSAPP*. Penerbit Buku Pedia, 2023.
- [10] Dewi, Riskha Dora Candra. "Early Warning System (Penggunaan Whatsapp Bot Di Bidang Kesehatan)." *Bureaucracy Journal: Indonesia Journal of Law and Social-Political Governance* 3.1 (2023): 10-19.
- [11] Akbar, Y., & Sugiharto, T. (2023). Analisis Sentimen Pengguna Twitter di Indonesia Terhadap ChatGPT Menggunakan Algoritma C4. 5 dan Naïve Bayes. *Jurnal Sains dan Teknologi*, 5(1), 115-122.