

b(Artikel Penelitian/Ulasan)

Perancangan Aplikasi Web Chatbot Multi-Bahasa Berbasis NPL Translator API Dengan Multibahasa Terjemahan

Robet¹, Johannes Terang Kita Perangin Angin², Randy Wilson³

¹ Sekolah Tinggi Manajemen Informatika dan Komputer (TIME) ; email : robet@stmik-time.ac.id

² Sekolah Tinggi Manajemen Informatika dan Komputer (TIME) ; email : timejohanes@gmail.com

³ Sekolah Tinggi Manajemen Informatika dan Komputer (TIME) ; email : randywilson2808@gmail.com

Abstract: *Computer software that can mimic and process spoken or written human interactions is called a multilingual chatbot, or chatterbot. With a chatbot, people can communicate with a digital device as if they were talking to a real person. The goal of this development research is to use machine learning techniques to build and create a multilingual chatbot application with translation capabilities. The system development process requires a number of phases. One approach to software development is the Software Development Life Cycle. The Transformer method was chosen as the system development approach for this research. The anticipated goal of this research is: to create a website-based multilingual chatbot application using the NPL Translator API. Users who experience language barriers may find it easier to use the chat application's built-in automatic translator. Because of its web-based nature, this chat program facilitates user communication from any location.*

Keywords: *Chatbot Multibahasa , NPL Translator API , Algoritma Machine Learning*

Abstrak: Perangkat lunak komputer yang dapat meniru dan memproses interaksi manusia yang diucapkan atau ditulis disebut chatbot multibahasa, atau chatterbot. Dengan chatbot, orang dapat berkomunikasi dengan perangkat digital seolah-olah mereka berbicara dengan orang sungguhan. Tujuan dari penelitian pengembangan ini adalah untuk menggunakan teknik pembelajaran mesin untuk membangun dan membuat aplikasi chatbot multibahasa dengan kemampuan penerjemahan. Proses pengembangan sistem memerlukan sejumlah fase. Salah satu pendekatan untuk pengembangan perangkat lunak adalah Siklus Hidup Pengembangan Perangkat Lunak. Metode Transformer dipilih sebagai pendekatan pengembangan sistem untuk penelitian ini. Tujuan yang diantisipasi dari penelitian ini adalah: untuk membuat aplikasi chatbot multibahasa berbasis situs web menggunakan NPL Translator API. Pengguna yang mengalami kendala bahasa mungkin merasa lebih mudah untuk menggunakan penerjemah otomatis bawaan aplikasi obrolan. Karena sifatnya yang berbasis web, program obrolan ini memfasilitasi komunikasi pengguna dari lokasi mana pun.

Kata kunci: Chatbot Multibahasa , NPL Translator API , Algoritma Machine Learning

Diterima: 02 Februari 2025
Direvisi: 12 Februari 2025
Diterima: 28 Februari 2025
Diterbitkan: 12 Maret 2025
Versi sekarang: 28 Maret 2025



Hak cipta: © 2025 oleh penulis.
Diserahkan untuk kemungkinan publikasi akses terbuka berdasarkan syarat dan ketentuan lisensi Creative Commons Attribution (CC BY SA) (<https://creativecommons.org/licenses/by-sa/4.0/>)

1. Pendahuluan

Banyak aspek kehidupan manusia, termasuk pendidikan, telah mengalami perubahan signifikan sebagai hasil dari kemajuan teknologi informasi dan komunikasi. Salah satu perkembangan signifikan adalah munculnya teknologi kecerdasan buatan seperti Chatbot. Chatbot adalah teknologi yang memungkinkan orang dan komputer berinteraksi melalui bahasa alami dan memberikan respons yang cepat berdasarkan apa yang dimaksudkan pengguna. Meskipun awalnya digunakan untuk meniru percakapan manusia dan menghibur

pengguna, aplikasinya telah berkembang ke berbagai bidang seperti pendidikan, pencarian informasi, bisnis, dan e-commerce. yang mengklaim bahwa chatbot adalah sebuah teknologi yang dapat membaca masukan teks dan kemudian menghasilkan kata kunci untuk menyampaikan respons.

Pengembangkan aplikasi chatbot multibahasa yang mampu berkomunikasi dengan pengguna dalam berbagai bahasa dan dilengkapi dengan fitur terjemahan untuk membantu pengguna dalam berkomunikasi dengan orang lain yang menggunakan bahasa yang berbeda. Komunikasi antar individu dari berbagai negara menjadi semakin sering terjadi. Hal ini menimbulkan kebutuhan alat bantu komunikasi yang mampu menerjemahkan bahasa secara real-time.

Oleh karena itu, pengguna harus menggunakan program penerjemahan terpisah agar dapat berkomunikasi dengan mereka yang berbicara berbagai bahasa, yang dapat menyulitkan dan memakan waktu. Selain itu, chatbot yang ada saat ini juga tidak selalu memberikan hasil terjemahan yang akurat dan natural. Hal ini dapat menyebabkan miskomunikasi dan kebingungan bagi pengguna.

Aplikasi ini akan mampu berkomunikasi dengan pengguna dalam berbagai bahasa dan memberikan hasil terjemahan yang akurat dan natural. Salah satu teknologi yang paling menjanjikan untuk menciptakan chatbot yang dapat berbicara banyak bahasa adalah pembelajaran mesin. Machine learning dapat digunakan untuk melatih chatbot agar mampu memahami dan menghasilkan bahasa secara natural. Penelitian sebelumnya telah menunjukkan bahwa machine learning dapat digunakan untuk mengembangkan chatbot yang bisa berkomunikasi dalam berbagai Bahasa dengan Tingkat akurasi yang tinggi untuk hasil pengujian Black Box menunjukkan performa chatbot mampu merespon 8 pertanyaan yang diajukan secara acak dengan benar dan mendapatkan presentase sebesar 97% yang dimana ini sudah cukup baik untuk model chatbot.

Untuk mengatasi permasalahan tersebut, penelitian ini mengusulkan pengembangan sebuah aplikasi chatbot multi-bahasa dengan fitur terjemahan menggunakan NPL Translator API. NPL Translator API ini di pilih karena NPL Translator API telah menjadi bidang lanjutan dari perkembangan teknologi komputer untuk meniru otak manusia dan telah membawa bidang disiplin statistik yang luas dan menghasilkan teori komputasi statistik dasar dari proses pembelajaran. Hasil akhir dari penelitian ini diharapkan aplikasi chatbot multibahasa berbasis machine learning dengan fitur terjemahan yang mampu berkomunikasi dengan pengguna dalam berbagai Bahasa, Memberikan hasil terjemahan yang akurat dan natural, Mudah digunakan dan dipahami oleh pengguna.

2. Tinjauan Literatur

2.1. Chatbot Multi-bahasa

Chatbot atau chatterbot adalah program komputer yang mampu menyimulasikan dan memproses percakapan manusia (tertulis atau lisan). Chatbot memungkinkan manusia untuk berinteraksi dengan perangkat digital seperti layaknya mereka berbicara dengan manusia nyata. Bentuk chatbot dapat berupa program yang mampu merespon satu baris kalimat sederhana pada sebuah pertanyaan, atau bisa juga berbentuk asisten digital yang belajar dan berevolusi untuk memberikan tingkat personalisasi tingkat tinggi seiring chatbot mengumpulkan dan memproses informasi. Dengan kata lain, chatbot adalah komputer atau robot yang dibuat khusus untuk berkomunikasi dengan orang-orang.

2.2. AI Model (Google Gemini Chat Model)

Google AI, melalui proyek Gemini, telah mengembangkan model pemrosesan bahasa alami yang canggih, yang dapat digunakan untuk meningkatkan kemampuan chatbot dalam memahami dan merespons pertanyaan pengguna. Gemini AI memanfaatkan teknik pembelajaran mendalam (deep learning) untuk menganalisis dan memahami konteks percakapan, sehingga dapat memberikan jawaban yang lebih akurat dan relevan

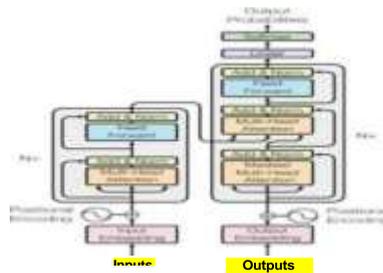
2.3. Natural Language Processing (NLP)

Menurut Eisenstein, J. (2018), Kumpulan metode yang digunakan untuk merancang bahasa manusia agar dapat diolah dan diakses oleh komputer disebut dengan Natural Language Processing (NLP). Penggunaan dari natural language processing banyak diaplikasikan pada kehidupan manusia seperti mesin penerjemah otomatis; spam email

filter dalam melakukan klasifikasi teks email masuk dari pesan sampah; search engine yang bergerak melampaui pencocokan kalimat dan analisis jaringan ke tingkat kecanggihan linguistik yang tinggi; dialog systems yang menyediakan cara yang semakin umum dan efektif untuk mendapatkan dan berbagi informasi.

2.4. Metode Transformer

Dengan dua sublapis per lapisan jaringan umpan maju yang terhubung secara posisional dan mekanisme perhatian mandiri multihead encoder terdiri dari tumpukan N lapisan identik. Mekanisme perhatian mandiri multihead yang tertopeng, mekanisme perhatian mandiri multihead, dan jaringan umpan maju yang terhubung secara posisional dan posisional adalah tiga sublapis yang membentuk masing-masing N level identik dekoder.

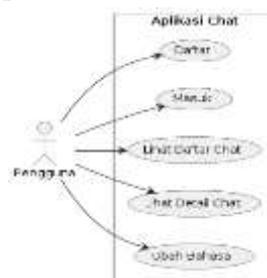


Gambar 2 1 Arsitektur Transformers

Arsitektur transformator mengubah urutan menggunakan dua encoder penting dan decoder. Fungsi encoder adalah menangkap urutan masukan dan mengolah informasi dengan panjang dan urutan yang tetap, setelah itu decoder menampilkan keluaran yang telah diproses sebelumnya pada encoder, seperti terlihat pada Gambar.

2.5. Use Case Diagram

Salah satu alat untuk pemodelan sistem yang merupakan bagian dari UML yang digunakan untuk menjelaskan bagaimana pengguna dan sistem berinteraksi adalah diagram use case. Diagram Use Case dapat menjadi alat yang berguna untuk menggambarkan batasan sistem secara jelas dan memberikan penjelasan tentang konteksnya. Tujuan dari tampilan use case adalah untuk mencantumkan aktor dan use case serta menampilkan aktor mana yang terlibat dalam setiap use case.



Gambar 2 2 Use Case Diagram

Pada gambar 22. merupakan contoh dari use case juga yang mendeskripsikan chatbot multi-bahasa dengan fitur terjemahan. Terdapat tingkat detail yang menentukan bentuk faktor yang lebih jelas dalam hal kasus penggunaan lain yang lebih sederhana. Sebuah use case diimplementasikan sebagai kolaborasi dalam tampilan interaksi [9]. Berikut simbol use case diagram yang ditunjukkan pada tabel 2.1.

2.6. Black Box Testing

Salah satu jenis pengujian perangkat lunak yang digunakan untuk memastikan fungsionalitas aplikasi disebut pengujian "kotak hitam". Pengujian kotak hitam terutama berkaitan dengan apakah masukan aplikasi tersedia atau mematuhi keluaran yang diharapkan untuk setiap nilai masukan. Metode pengujian ini didasarkan pada kebutuhan

dan spesifikasi perangkat lunak dari interface aplikasi sehingga cara kerja internal dari item yang diuji tidak diketahui oleh penguji. Jadi dapat dikatakan bahwa teknik pengujian ini juga disebut based testing and behaviour testing. Metode ini mendapatkan namanya dari fakta bahwa penguji tidak perlu memahami bagaimana kode internal aplikasi diimplementasikan.

Seperti yang terlihat pada Gambar 2.3, pengujian ini menanggapi masukan yang valid dan salah berdasarkan kebutuhan pengguna.

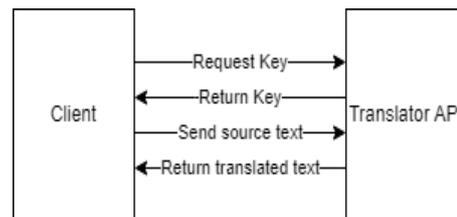
3. Metode

3.1. Design Penelitian

Tujuan dari penelitian pengembangan ini adalah untuk menggunakan teknik pembelajaran mesin guna membangun dan membuat aplikasi chatbot multibahasa dengan kemampuan penerjemahan. Suatu sistem harus melalui sejumlah fase dalam proses pengembangannya. Salah satu pendekatan untuk pengembangan perangkat lunak adalah Siklus Hidup Pengembangan Perangkat Lunak. Metode Transformer dipilih sebagai pendekatan pengembangan sistem untuk penelitian ini.

3.1.1. Perancangan

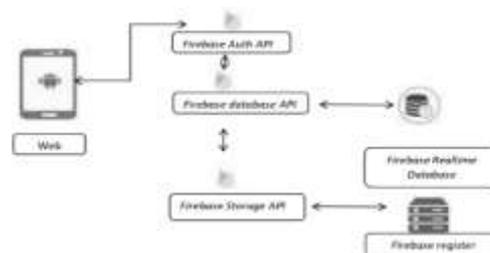
Perancangan membahas mengenai proses perencanaan dan desain sistem yang menjadi dasar dalam pengembangan aplikasi atau penelitian yang sedang dilakukan. Perancangan merupakan tahap penting untuk memastikan sistem yang dirancang mampu memenuhi kebutuhan pengguna dan tujuan yang telah ditetapkan. Alur dan deskripsi aplikasi yang akan dikembangkan dijelaskan dalam desain ini.



Gambar 3.1 rancangan chatbot multi-bahasa

Alur data yang disebutkan di atas menjelaskan cara kerja program, dimulai dari layar login. Menu registrasi akan muncul jika pengguna belum memiliki akun. Setelah register, pengguna diarahkan kembali ke halaman login untuk menginputkan data pengguna untuk login. Jika login berhasil maka pengguna akan diarahkan ke halaman utama. Pengguna dapat memulai percakapan dengan orang yang mereka ajak bicara di halaman Chatting setelah menambahkan teman di tab semua pengguna jika mereka belum punya teman.

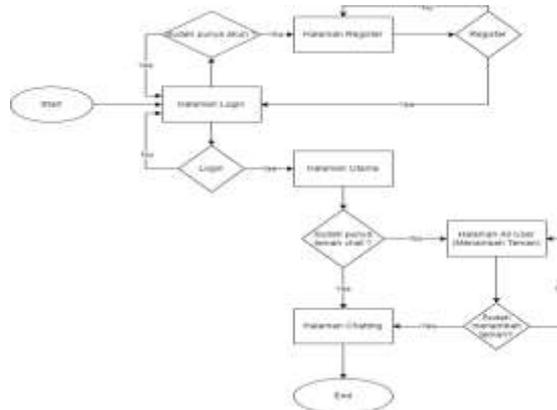
3.1.2. Rancangan Kebutuhan Database



Gambar 3.2 Skema Kebutuhan Database

Arsitektur penggunaan basis data dalam studi ini digambarkan dalam grafik di atas. Program ini memungkinkan pengguna untuk terhubung ke layanan Firebase seperti Firebase Auth, yang mendaftarkan akun email ke dalam basis data; Firebase Storage, yang menyimpan file seperti foto avatar saat pengguna membuat akun; dan Firebase Realtime Database, yang menyimpan semua pesan, kontak, dan data pendaftar.

3.1.3. Penggunaan NLP Translator API (Gemini AI API)



Gambar 3.3 Alur Penggunaan NLP Translator API

Berikut adalah langkah-langkah yang digunakan untuk menerjemahkan pesan menggunakan NLP Translation API:

- a. Client melakukan request Key
- b. Sebagai tanggapan, server memberikan kunci kepada klien untuk digunakan.
- c. Pelanggan memiliki opsi untuk mengirim pesan untuk diterjemahkan.

Tabel 3.1 di bawah ini menunjukkan parameter yang diperlukan untuk mengirimkan permintaan.

Tabel 3 1 Request API

<i>Params</i>	Deskripsi
<i>text</i>	<i>Required.</i> Sebuah <i>string</i> yang akan diterjemahkan
<i>to</i>	<i>Required.</i> <i>String</i> kode bahasa yang digunakan untuk menterjemahkan teks ke bahasa sasaran.
<i>from</i>	<i>Optional.</i> Sebuah <i>string</i> kode bahasa untuk awal mula teks yang akan diterjemahkan

Sebagai respons terhadap permintaan pengguna, server menyediakan hasil terjemahan. Tabel 3.2 di bawah ini menampilkan informasi yang diambil dari server.

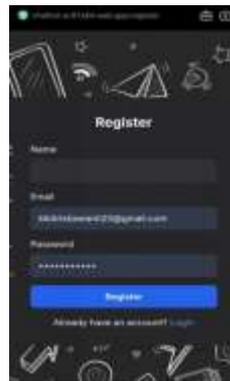
Tabel 3 2 Response API

<i>Body</i>	Deskripsi
<i>status</i>	Sebuah kembalian dari <i>server</i> yaitu <i>http status code</i>
<i>original_text</i>	Sebuah kembalian dari <i>server</i> berupa teks awal sebelum diterjemahkan
<i>translated_text</i>	Sebuah kembalian dari <i>server</i> berupa teks awal sesudah diterjemahkan

4. Hasil dan Pembahasan

Pada bab IV akan berfokus pada pembuatan model untuk melakukan perancangan terhadap model *Translator* dalam chatbot multi-bahasa dengan fitur terjemahan melalui metode pengembangan *NLP Translator API* pada kerangka penelitian dan kemudian diimplementasikan pada tahap metode pengembangan aplikasi dalam bentuk rancangan *Application Programming Interface (API)* yang disertai dengan tampilan sistem dan bagaimana proses bentuk dari *output system*. Implementasi Desain Antarmuka.

4.1. Halaman Register



Gambar 4.1 Halaman Register

Pengguna dapat mengisi formulir data di halaman ini dengan login, alamat email, kata sandi, bio, dan bahasa pilihan mereka.

4.2 Halaman Login

Pengguna dapat memasukkan alamat email terdaftar dan informasi kata sandi di halaman ini.



Gambar 4.2 Halaman Login

4.3 Halaman Utama

Ini adalah halaman utama yang memuat informasi tentang pengguna yang telah chatting dengan mereka.



Gambar 4.3 Halaman Utama

4.4 Halaman Room Chat



Gambar 4.4 Room Chat Pengguna Bahasa Korea Ke Bahasa Indonesia

Pengguna dapat berbincang satu sama lain di halaman ini. Saat pengguna membuat akun, pesan akan secara otomatis diterjemahkan ke dalam bahasa yang mereka gunakan.

4.5. Halaman Profile

Pada halaman ini pengguna dapat melakukan *Sign Out* pada aplikasi.



Gambar 4.5 Halaman Profile

4.6. Kode Program Implementasi Menggunakan Javascrip

4.6.1. Login

```
1 const [username, password] = ['root'];
2 const [email, first, /lastName/];
3 const [username, password] = ['root', 'root'];
4 const [username, password] = ['root', 'root'];
5 const [username, password] = ['root', 'root'];
6 const [username, password] = ['root', 'root'];
7
8 const defaultFunction = () => {
9   const message = 'Welcome to the application!';
10  const {username, password} = {username: 'root', password: 'root'};
11  const {username, password} = {username: 'root', password: 'root'};
12  const {username, password} = {username: 'root', password: 'root'};
13  const {username, password} = {username: 'root', password: 'root'};
14  const {username, password} = {username: 'root', password: 'root'};
15  const {username, password} = {username: 'root', password: 'root'};
16  const {username, password} = {username: 'root', password: 'root'};
17  const {username, password} = {username: 'root', password: 'root'};
18  const {username, password} = {username: 'root', password: 'root'};
19  const {username, password} = {username: 'root', password: 'root'};
20  const {username, password} = {username: 'root', password: 'root'};
21  const {username, password} = {username: 'root', password: 'root'};
22  const {username, password} = {username: 'root', password: 'root'};
23  const {username, password} = {username: 'root', password: 'root'};
24  const {username, password} = {username: 'root', password: 'root'};
25  const {username, password} = {username: 'root', password: 'root'};
26  const {username, password} = {username: 'root', password: 'root'};
27  const {username, password} = {username: 'root', password: 'root'};
28  const {username, password} = {username: 'root', password: 'root'};
29  const {username, password} = {username: 'root', password: 'root'};
30  const {username, password} = {username: 'root', password: 'root'};
31  const {username, password} = {username: 'root', password: 'root'};
32  const {username, password} = {username: 'root', password: 'root'};
33  const {username, password} = {username: 'root', password: 'root'};
34  const {username, password} = {username: 'root', password: 'root'};
35  const {username, password} = {username: 'root', password: 'root'};
36  const {username, password} = {username: 'root', password: 'root'};
37  const {username, password} = {username: 'root', password: 'root'};
38  const {username, password} = {username: 'root', password: 'root'};
39  const {username, password} = {username: 'root', password: 'root'};
40  const {username, password} = {username: 'root', password: 'root'};
41  const {username, password} = {username: 'root', password: 'root'};
42  const {username, password} = {username: 'root', password: 'root'};
43  const {username, password} = {username: 'root', password: 'root'};
44  const {username, password} = {username: 'root', password: 'root'};
45  const {username, password} = {username: 'root', password: 'root'};
46  const {username, password} = {username: 'root', password: 'root'};
47  const {username, password} = {username: 'root', password: 'root'};
48  const {username, password} = {username: 'root', password: 'root'};
49  const {username, password} = {username: 'root', password: 'root'};
50  const {username, password} = {username: 'root', password: 'root'};
51  const {username, password} = {username: 'root', password: 'root'};
52  const {username, password} = {username: 'root', password: 'root'};
53  const {username, password} = {username: 'root', password: 'root'};
54  const {username, password} = {username: 'root', password: 'root'};
55  const {username, password} = {username: 'root', password: 'root'};
56  const {username, password} = {username: 'root', password: 'root'};
57  const {username, password} = {username: 'root', password: 'root'};
58  const {username, password} = {username: 'root', password: 'root'};
59  const {username, password} = {username: 'root', password: 'root'};
60  const {username, password} = {username: 'root', password: 'root'};
61  const {username, password} = {username: 'root', password: 'root'};
62  const {username, password} = {username: 'root', password: 'root'};
63  const {username, password} = {username: 'root', password: 'root'};
64  const {username, password} = {username: 'root', password: 'root'};
65  const {username, password} = {username: 'root', password: 'root'};
66  const {username, password} = {username: 'root', password: 'root'};
67  const {username, password} = {username: 'root', password: 'root'};
68  const {username, password} = {username: 'root', password: 'root'};
69  const {username, password} = {username: 'root', password: 'root'};
70  const {username, password} = {username: 'root', password: 'root'};
71  const {username, password} = {username: 'root', password: 'root'};
72  const {username, password} = {username: 'root', password: 'root'};
73  const {username, password} = {username: 'root', password: 'root'};
74  const {username, password} = {username: 'root', password: 'root'};
75  const {username, password} = {username: 'root', password: 'root'};
76  const {username, password} = {username: 'root', password: 'root'};
77  const {username, password} = {username: 'root', password: 'root'};
78  const {username, password} = {username: 'root', password: 'root'};
79  const {username, password} = {username: 'root', password: 'root'};
80  const {username, password} = {username: 'root', password: 'root'};
81  const {username, password} = {username: 'root', password: 'root'};
82  const {username, password} = {username: 'root', password: 'root'};
83  const {username, password} = {username: 'root', password: 'root'};
84  const {username, password} = {username: 'root', password: 'root'};
85  const {username, password} = {username: 'root', password: 'root'};
86  const {username, password} = {username: 'root', password: 'root'};
87  const {username, password} = {username: 'root', password: 'root'};
88  const {username, password} = {username: 'root', password: 'root'};
89  const {username, password} = {username: 'root', password: 'root'};
90  const {username, password} = {username: 'root', password: 'root'};
91  const {username, password} = {username: 'root', password: 'root'};
92  const {username, password} = {username: 'root', password: 'root'};
93  const {username, password} = {username: 'root', password: 'root'};
94  const {username, password} = {username: 'root', password: 'root'};
95  const {username, password} = {username: 'root', password: 'root'};
96  const {username, password} = {username: 'root', password: 'root'};
97  const {username, password} = {username: 'root', password: 'root'};
98  const {username, password} = {username: 'root', password: 'root'};
99  const {username, password} = {username: 'root', password: 'root'};
100 const {username, password} = {username: 'root', password: 'root'};
```

Gambar 4.6 Kode Halaman Login

Gambar 4.6 adalah gambaran yang menunjukkan implementasi halaman login menggunakan React.Js dengan Fibrasi Authentication termasuk validasi kesalahan pada saat proses autentikasi.

4.6.2. Register

```

import defaultFunctionRegister from './defaultFunctionRegister';
import { useState } from 'react';
import { createUserWithEmailAndPassword } from 'firebase/auth';
import { doc, setDoc } from 'firebase/firestore';

const Register = () => {
  const [username, setUsername] = useState('');
  const [password, setPassword] = useState('');
  const [error, setError] = useState('');

  const register = async () => {
    try {
      const userCredential = await createUserWithEmailAndPassword(
        auth,
        username,
        password
      );

      const user = userCredential.user;

      await setDoc(doc(db, 'users', user.uid), {
        name: username,
        email: user.email,
        password: password,
      });

      navigate('/home');
    } catch (error) {
      setError(error.message);
    }
  };

  return (
    <div>
      <input type="text" value={username} onChange={setUsername} />
      <input type="password" value={password} onChange={setPassword} />
      <button onClick={register}>Register</button>
      {error ? <p>{error}</p> : null}
    </div>
  );
};

export default Register;

```

Gambar 4.7 Kode Halaman Register

Pada gambar 4.7 menunjukkan kode halaman registrasi yang memverifikasi panjang kata sandi sebelumnya dan membuat akun baru menggunakan firebase authentication dan memperbaharui profil pengguna.

4.6.3. Koneksi

```

import { initializeApp } from 'firebase/app';
import { getFirestore } from 'firebase/firestore';
import { getAuth } from 'firebase/auth';

const firebaseConfig = {
  apiKey: 'AIzaSyA...',
  authDomain: '...',
  projectId: '...',
  storageBucket: '...',
  messagingSenderId: '...',
  appId: '1:...'
};

const app = initializeApp(firebaseConfig);
const db = getFirestore(app);
const auth = getAuth(app);

export { db, auth };

```

Gambar 4.8 Kode Halaman Login

Gambar 4.8 menjelaskan tentang kode ini mengatur koneksi ke Firebase Firestore menggunakan Firebase Admin SDK untuk mengelola database di aplikasi backend.

4.6.4. Fungsi Pengiriman Pesan

```

import { useState } from 'react';
import { doc, updateDoc } from 'firebase/firestore';
import { auth } from './firebase';

const SendMessage = () => {
  const [message, setMessage] = useState('');
  const [recipient, setRecipient] = useState('');
  const [error, setError] = useState('');

  const sendMessage = async () => {
    try {
      const docRef = doc(db, 'messages', Date.now());
      await setDoc(docRef, {
        sender: auth.currentUser.uid,
        recipient: recipient,
        message: message,
        timestamp: Date.now(),
      });
    } catch (error) {
      setError(error.message);
    }
  };

  return (
    <div>
      <input type="text" value={message} onChange={setMessage} />
      <input type="text" value={recipient} onChange={setRecipient} />
      <button onClick={sendMessage}>Send</button>
      {error ? <p>{error}</p> : null}
    </div>
  );
};

export default SendMessage;

```

Gambar 4.9 Kode Fungsi Pengiriman Pesan

Gambar 4.9 Kode ini menunjukkan fungsi pengiriman pesan dalam React yang berinteraksi dengan API berbasis machine learning, memungkinkan pengguna mengirim teks yang kemudian diproses oleh model AI untuk memberikan respons yang relevan.

4.6.5. Implementasi Fungsi Getchatresponse Dalam Node.Js



Gambar 4.10 Kode Implementasi Fungsi Getchatresponse Dalam Node.Js

Gambar 4.10 menunjukkan kode implementasi fungsi getChatResponse dalam Node.js, yang memanfaatkan model Google Generative AI (Gemini 1.5 Pro) sebagai bagian dari sistem machine learning, di mana model tersebut digunakan untuk memahami dan menerjemahkan teks secara otomatis berdasarkan perintah yang diberikan, serta dilakukan pemrosesan teks tambahan untuk membersihkan format output agar lebih sesuai dengan kebutuhan pengguna.

4.7. Pengujian Black Box

Aplikasi yang telah diimplementasikan kemudian masuk ke tahap pengujian, di mana tabel pengujian bahasa dan pengujian kotak hitam digunakan untuk mengetahui tingkat keberhasilan dan kualitas [25]. Pendekatan ini mengevaluasi karakteristik aplikasi, khususnya untuk input dan output aplikasi. Tabel 4.1 di bawah ini menunjukkan hasil pengujian black box.

Tabel 4 1 Pengujian Black Box

1. Test Case: Pendaftaran (Daftar)

ID	Test Case	Input	Expected Output	Status
TC-01	Daftar dengan data valid	Nama, Email, Password yang valid	Akun berhasil dibuat, notifikasi sukses	✓
TC-02	Daftar dengan email yang sudah digunakan	Email yang sudah terdaftar	Pesan error "Email sudah digunakan"	✓
TC-03	Daftar tanpa mengisi semua data	Kosongkan salah satu input	Pesan error "Semua field wajib diisi"	✓

2. Test Case: Login (Masuk)

ID	Test Case	Input	Expected Output	Status
TC-04	Login dengan akun yang valid	Email & Password yang benar	Berhasil masuk ke halaman utama	✓

TC-05	Login dengan password salah	Email benar, Password salah	Pesan error "Password salah"	✓
TC-06	Login dengan email tidak terdaftar	Email yang belum terdaftar	Pesan error "Email tidak ditemukan"	✓

3. Test Case: Lihat Daftar Chat

ID	Test Case	Input	Expected Output	Status
TC-07	Melihat daftar chat setelah login	User masuk ke halaman chat	Tampil daftar chat yang dimiliki user	✓
TC-08	Daftar chat kosong	User tanpa chat sebelumnya	Tampilkan pesan "Belum ada chat"	✓

4. Test Case: Lihat Detail Chat

ID	Test Case	Input	Expected Output	Status
TC-09	Membuka detail chat	Klik salah satu chat	Tampilan detail chat dengan pesan-pesan di dalamnya	✓
TC-10	Membuka chat yang sudah dihapus	Klik chat yang sudah dihapus	Pesan error "Chat tidak ditemukan"	✓

5. Test Case: Ubah Bahasa

ID	Test Case	Input	Expected Output	Status
TC-11	Mengubah bahasa ke Inggris	Pilih "English" di dropdown	Semua teks diterjemahkan ke bahasa Inggris	✓
TC-12	Mengubah bahasa ke bahasa yang tidak tersedia	Pilih bahasa yang tidak ada	Pesan error "Bahasa tidak tersedia"	✓

Keterangan:

✓ = Lulus

✗ = Gagal (Perlu perbaikan)

4.8. Hasil Pengujian Black Box

4.8.1. Hasil Pengujian Terjemahan Pesan

Tabel 4 2 Pengujian Bahasa Indonesia – Bahasa Korea. Dua pengguna yang berbicara dalam dua bahasa berbeda akan menguji aplikasi dalam pengujian ini. Untuk melakukan pengujian, pengguna A mengirim pesan ke pengguna B, dengan harapan NLP Translation API akan menerjemahkan pesan tersebut secara otomatis.

Tabel 4.2 menampilkan tabel pengujian.

No	Pengguna A (Korea)		Pengguna B (Indonesia)		Hasil Pengujian
	<i>Username</i>	Pesan	<i>Username</i>	Pesan	
1	Nam Do San	안녕하세요 (annyeonghaseyo)		Halo	Sesuai
2		안녕 (annyeong)	Iqbalnova	Selamat jalan	Sesuai
3	Nam Do San	방갑습니다 (banggapseupnida)		Senang bertemu denganmu	Sesuai
4		용서 해줘 (yongseo haejyo)	Iqbalnova	Maafkan aku	Sesuai

Berdasarkan hasil pengujian yang disebutkan, pesan yang dikirim dan diterima memenuhi harapan, khususnya pesan tersebut sesuai dengan bahasa pengguna.

5. Kesimpulan

Berdasarkan pembuatan aplikasi dan penerapan NLP Translator API yang telah dilakukan pada penelitian ini, maka dapat disimpulkan bahwa: chatbot multibahasa dirancang menggunakan metode pengembangan Translator API dan dapat menerjemahkan ke dalam tiga bahasa: Inggris, Korea, dan Indonesia. Aplikasi ini mampu berkomunikasi dengan pengguna dan memberikan hasil terjemahan yang akurat serta natural. Selain itu, aplikasi ini telah berkembang untuk digunakan dalam berbagai bidang seperti pendidikan, bisnis, dan e-commerce.

Adapun saran pada penelitian ini yaitu:

1. Meskipun chatbot telah berhasil menangani pertanyaan umum, pengembangan lebih lanjut dalam pemrosesan bahasa alami bisa difokuskan pada pemahaman konteks yang lebih mendalam dan pengelolaan dialog yang lebih kompleks. Ini dapat memungkinkan chatbot untuk menangani percakapan yang lebih dinamis dan mendetail.
2. Peneliti ini dapat dilanjutkan dengan mempertimbangkan pengembangan lebih lanjut dengan menggunakan teknologi AI yang lebih canggih, seperti machine learning atau penyimpanan data percakapan, untuk memungkinkan chatbot belajar dari interaksi sebelumnya dan memberikan layanan yang semakin personal dan relevan. chatbot dan memberikan pengalaman yang lebih baik kepada pengguna, seiring dengan perkembangan teknologi.

Referensi

Anindyati,Laksmi.2023.ANALISIS DAN PERANCANGAN APLIKASI CHATBOT MENGGUNAKAN FRAMEWORK RASA DAN SISTEM INFORMASI PEMELIHARAAN APLIKASI (STUDI KASUS: CHATBOT PENERIMAAN MAHASISWA BARU POLITEKNIK ASTRA). Vol. 10, No. 2, April 2022, hlm. 291-300 p-ISSN: 2355-7699 Akreditasi KEMENRISTEKDIKTI, No. 36/E/KPT/2019e-ISSN: 2528-6579.https://www.researchgate.net/publication/373529407_Analisis_dan_Perancangan_Aplikasi_Chatbot_Menggunakan_Framework_Rasa_dan_Sistem_Informasi_Pemeliharaan_Aplikasi_Studi_Kasus_Chatbot_Penerimaan_Mahasiswa_Baru_Politeknik_Astra

Auslander, P. (2002). Live from Cyberspace: Or, I Was Sitting at My Computer This Guy Appeared He Thought I Was a Bot. PAJ: A Journal of Performance and Art, 24(1), 16–21. <http://www.jstor.org/stable/3246456>

Eisenstein, J. (2018). Natural Language Processing. In MIT Press. <https://doi.org/10.4324/9780203103517-5>

- Muhammad Izzuddin Mubarak et al.2024.IMPLEMENTASI NATURAL LANGUAGE PROCESSING DALAM PERANCANGAN APLIKASI CHATBOT PADA FIKTI UMSU. JATI (Jurnal Mahasiswa Teknik Informatika) Vol. 8 No. 6, Desember 2024. ejournal.itn.ac.id
- Nia Agustina Purwitasari et al.2024. Implementasi Algoritma Artificial Neural Network Dalam Pembuatan Chatbot Menggunakan Pendekatan Natural Language Processing.Program Studi Teknik Informatika, Institut Teknologi Indonesia. <https://www.scribd.com/document/571417512/Implementasi-Algoritma-Artificial-Neural-NetworkDalam-Pembuatan-ChatbotMenggunakan-Pendekatan-Natural-Language-Processing>
- Rahman, Muhammad Arief.2023.IMPLEMENTASI MACHINE LEARNING PADA APPLICATION PROGRAMMING INTERFACE UNTUK KLASIFIKASI KOMENTAR CYBERBULLYING.skripsi.<https://repository.uinjkt.ac.id/dspace/bitstream/123456789/67839/1/MUHAMMAD%20ARIEF%20RAHMAN-FST.pdf>
- Wijaya & Wicaksana.2019. JACOB Voice Chatbot Application Using Wit.ai for Providing Information in UMN.International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249-8958 (Online), Volume-8 Issue-6S3, September 2019.<https://www.ijeat.org/wp-content/uploads/papers/v8i6S3/F10170986S319.pdf>