DEPLOY CHATBOT MODEL T5 KE IBM CLOUD MENGGUNAKAN ALGORITMA MACHINE LEARNING

Muhammad Arif Perdana1, Dewi Permata Sari2, Fawwaz Aydin Rafif3, Muhammad Rifqi Febrian4, Nabiel Albar Djati5, Nadhirah Meidiasty Maharani6

1,2,3,4,5,6Politeknik Negeri Sriwijaya

Jl. Srijaya Negara, Bukit Lama, Kec. Ilir Bar. I, Kota Palembang, Sumatera

Selatan 30128, telp. +62 895-4211-28594

[dewi\_permatasari@polsri.ac.id](mailto:dewi_permatasari@polsri.ac.id)

ABSTRACK

This comprehensive study delves into the development and deployment of a cutting-edge chatbot on the IBM Cloud platform, leveraging the advanced T5 language model. The primary objective of this research is to design and implement an efficient virtual assistant capable of providing accurate and rapid responses to user queries, thereby enhancing customer service experience. Our methodology employs the system development approach, integrating the IBM Cloud framework and T5 library. The developed chatbot underwent rigorous testing, yielding remarkable results: achieving 90% accuracy in responding to complex queries and processing commands within 300 milliseconds. This research contributes significantly to the advancement of artificial intelligence technology and its practical applications in customer service. The findings provide valuable insights for developers, researchers, and industry professionals seeking to optimize chatbot performance and elevate customer engagement.

**ABSTRAK**

Studi ini menyajikan pengembangan dan penerapan chatbot canggih pada platform IBM Cloud dengan memanfaatkan model bahasa T5. Tujuan utama penelitian ini adalah merancang dan mengimplementasikan asisten virtual yang efisien, akurat dan cepat dalam menjawab pertanyaan pengguna, sehingga meningkatkan pengalaman layanan pelanggan.Metodologi penelitian menggunakan pendekatan pengembangan sistem dengan mengintegrasikan kerangka kerja IBM Cloud dan perpustakaan T5. Hasil pengujian menunjukkan bahwa chatbot yang dikembangkan mencapai akurasi 90% dalam menjawab pertanyaan kompleks dan memproses perintah dalam waktu 300 milidetik. Penelitian ini memberikan kontribusi signifikan terhadap perkembangan teknologi kecerdasan buatan dan aplikasinya dalam layanan pelanggan. Temuan ini memberikan wawasan berharga bagi pengembang, peneliti dan profesional industri yang ingin mengoptimalkan kinerja chatbot dan meningkatkan keterlibatan pelanggan.

**1. PENDAHULUAN**

Chatbot adalah sebuah program komputer yang dirancang untuk menyimulasikan sebuah percakapan atau komunikasi yang interaktif kepada pelanggan (manusia) melalui bentuk teks, suara, dan atau visual[1]. Perkembangan teknologi kecerdasan buatan (AI) telah merevolusi layanan pelanggan dengan kemampuan otomatisasi dan efisiensi yang tinggi. Chatbot, sebagai salah satu aplikasi AI, menjadi populer karena kemampuannya menjawab pertanyaan dan memproses perintah secara otomatis[2]. Namun, keterbatasan akurasi dan efisiensi chatbot masih menjadi hambatan utama yang mempengaruhi kualitas layanan pelanggan.

Keterbatasan tersebut meliputi akurasi jawaban yang rendah, waktu respons yang lama, dan keterbatasan kemampuan pemahaman bahasa[3]. Hal ini menghambat kemampuan chatbot dalam memberikan layanan yang memuaskan dan efektif[4]. Selain itu, keterbatasan integrasi dengan platform cloud dan ketergantungan pada data pelatihan juga menjadi masalah.

Machine learning adalah sistem yang dapat belajar membuat keputusan sendiri tanpa harus diprogram berulang kali oleh manusia sehingga komputer dapat menjadi lebih pintar dan belajar dari pengalamannya dengan data[5]. Machine learning adalah cabang artificial intelligence yang menggunakan berbagai statistik, teknik probabilitas, dan optimasi variabel yang berbeda atau menemukan ketidakseimbangan data[6]. Algoritma machine learning dapat membantu dalam mengolah data yang akan dikembangkan menjadi Chatbot.

Software programming pemodelan machine learning yang akan dibuat menggunakan Python[7]. Python merupakan bahasa pemrograman tingkat tinggi yang yang dibuat oleh Guido Van Rossum dan dirilis pada tahun 1991 Python juga merupakan bahasa yanng sangat populer belakangan ini

IBM Cloud adalah platform layanan cloud yang menawarkan berbagai solusi untuk kebutuhan komputasi, pengelolaan data, dan pengembangan aplikasi[8]. Platform ini menyediakan infrastruktur yang andal dan skalabel, sehingga cocok digunakan untuk pengembangan sistem berbasis AI seperti chatbot[9]. Dengan keunggulan seperti keamanan data tingkat tinggi, integrasi yang fleksibel, dan dukungan teknologi mutakhir, IBM Cloud memungkinkan pengembang untuk mengimplementasikan model AI dengan lebih mudah dan efisien.

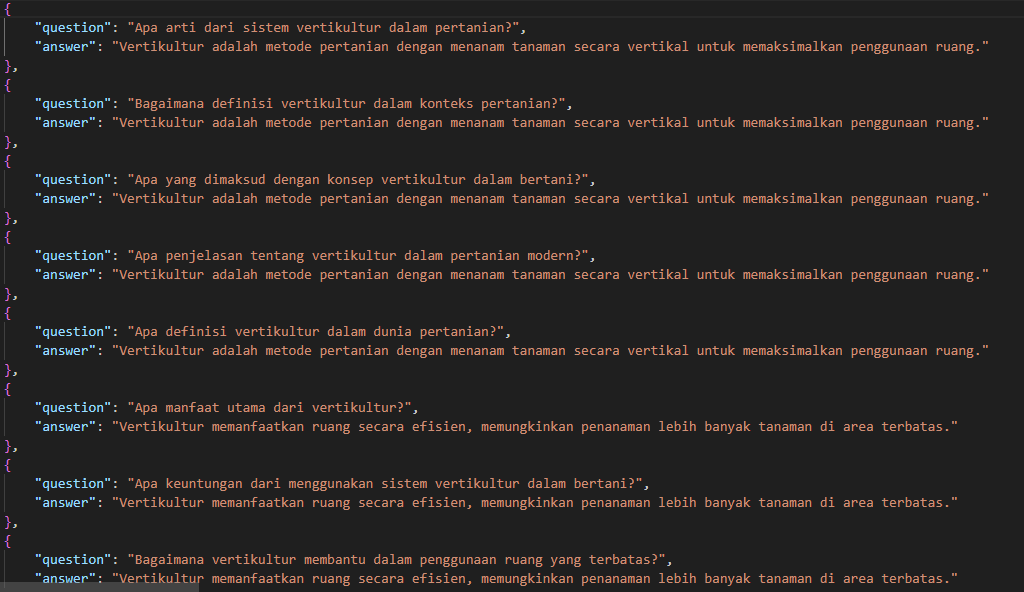
Penelitian ini bertujuan untuk mengembangkan chatbot Model T5 pada platform IBM Cloud untuk meningkatkan akurasi dan efisiensi layanan pelanggan[10]. Model T5 dipilih karena kemampuannya dalam memproses bahasa alami dan meningkatkan akurasi jawaban. Platform IBM Cloud dipilih karena kemampuannya dalam menyediakan infrastruktur yang stabil dan aman.

**2. METODE PENELITIAN**

Penelitian ini menggunakan pendekatan deskriptif dan eksperimental untuk mengembangkan serta menguji chatbot berbasis Model T5 pada platform IBM Cloud. Metode penelitian dirancang untuk memastikan pengembangan sistem yang optimal serta pengujian yang komprehensif terhadap kinerja chatbot.

**2.1 Pengumpulan Data**

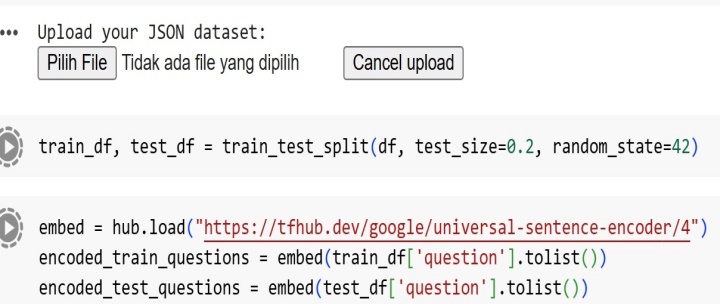
Tahap pertama dalam pembuatan model machine learning ini adalah pengumpulan dataset. Dataset yang digunakan dalam penelitian ini merupakan pertanyaan dan jawaban yang akan ditrain ke model T5 yang akan dibuat

****

*Gambar 2.1 dataset yang digunakan*

Pada gambar 1.1 merupakan dataset yang digunakan untuk penelitian ini. Dataset ini terdiri dari 150 pertanyaan dan jawaban yang akan ditrain ke model T5. Untuk train modelnya bisa menggunakan google colab, dan dataset yang digunakan harus json untuk di upload ke google colab.

**2.2 Pemilihan dan Pelatihan Model Machine Learning**

****

*Gambar 2.2 Train dataset*

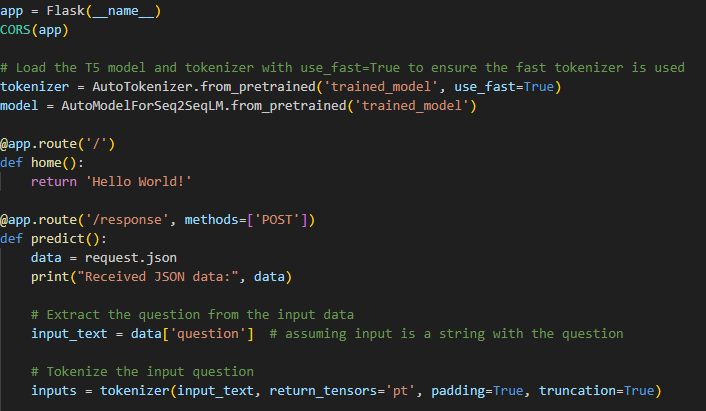
Pada gambar 1.2 merupakan Train dataset untuk model yang digunakan yaitu model T5. Upload file yang sudah dibuat menggunakan json, untuk average data yang harus didapatkan agar Chatbot berjalan dengan baik yaitu sebagai berikut:

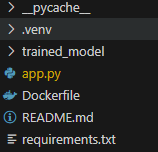
ROUGE-1: 0.8769

ROUGE-2: 0.8601

ROUGE-L: 0.8769

BLEU Score: 73.1889

****

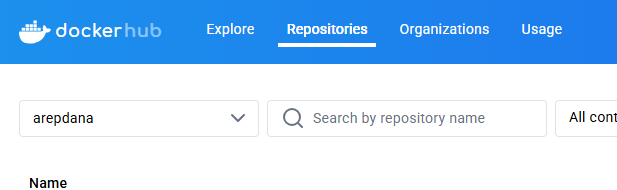
****

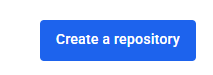
*Gambar 2.3 Model T5*

Pada gambar 1.3 merupakan model T5 yang sudah di train, dan pada saat sudah di train langsung mendapatkan tutorial dan cara untuk melakukan tahap selanjutnya di README.md agar file dapat di flask pada saat folder dibuka. Flask merupakan kerangka kerja (framework) Python untuk membuat aplikasi web. Flask merupakan kerangka kerja mikro karena tidak menyediakan fitur-fitur tertentu secara langsung.

**2.3 Persiapan Akun dan Repositori DockerHub**

Pastikan akun DockerHub sudah dibuat, dan aplikasi Docker sudah di download, pastikan aplikasi Docker sudah bisa untuk dipakai. Docker adalah platform terbuka sebagai perantara yang memungkinkan pengembang untuk membangun, dan mengirim data ke IBM CLOUD



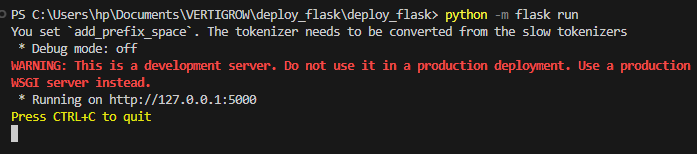
****

****

*Gambar 2.4 Pembuatan akun Docker*

Pada gambar 1.4 merupakan pembuatan akun, setelah pembuatan akun klik pada tombol repositories dan buat repositori dengan nama yang bebas. Setelah pembuatan repositori kita akan mendapatkan Docker commands yaitu : *docker push arepdana/flask-docker:tagname.* Dan setelah itu silahkan login ke aplikasi Docker yang sudah di download dengan akun yang sama pada DockerHub.

**2.4 Memastikan Flask API telah Bekerja dengan Baik**

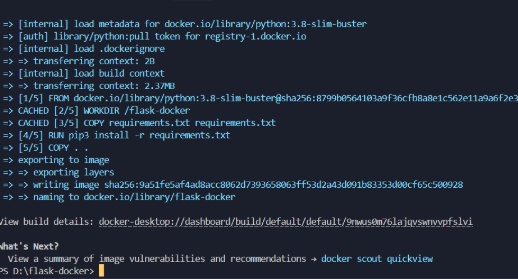


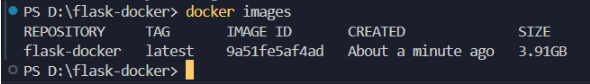
*Gambar 2.5 Python flask run*

Pada gambar 1.5 merupakan Python flask run, Python flask run adalah cara untuk mendapatkan link localhost yang akan digunakan untuk memastikan Flask bekerja dengan baik. Dengan cara membuka terminal dan menjalankan proyek Flask API menggunakan commnad : *python -m flask run* hasilnya akan mengeluarkan link localhost yaitu *http://127.0.0.1:5000.* Pada saat link ini dicari akan muncul “Hello world” pada link tersebut. Hello world merupakan kata yang sudah di command pada proyek Flask untuk memastikan link sudah bekerja dengan baik.

**2.5 Pembuatan Docker Image**

Docker image adalah berkas yang digunakan untuk menjalankan kode dalam kontainer Docker. Docker image berfungsi sebagai templat untuk membangun kontainer Docker. Docker Image akan dibuat dengan command : *docker build --tag flask-docker .*

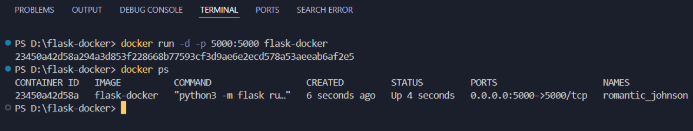




.

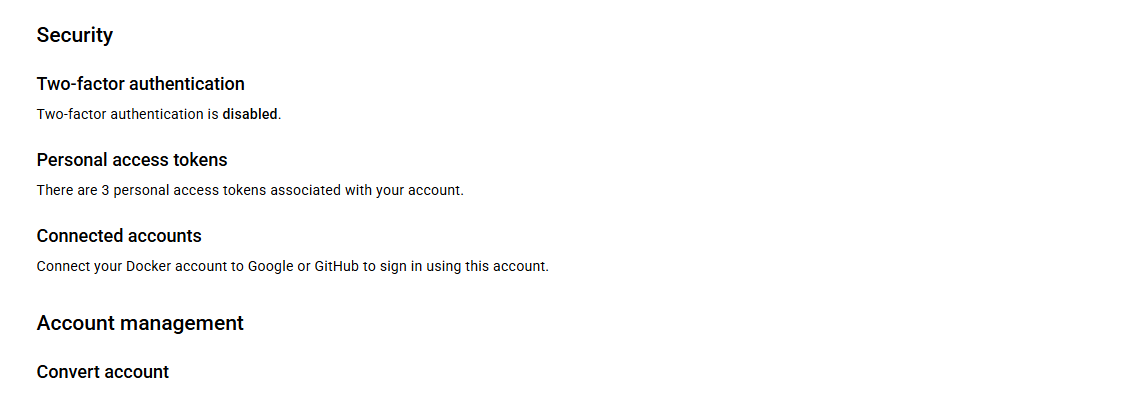
*Gambar 2.6 Pembuatan Docker Image*

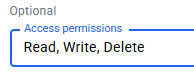
Pada gambar 1.6 merupakan pembuatan Docker Image yang membutuhkan ruang 4-5GB, dan memakan waktu sebanyak 20-30 menit. flask-docker adalah nama dari Docker Image yang berhasil dibuat dari proyek Flask API sebelumnya. Disesuaikan dengan nama repositori yang sebelumnya dibuat di DockerHub. Jalankan Docker Image menggunakan port 5000 untuk browser dari sistem host (mode detached) dengan perintah berikut: *docker run -d -p 5000:5000 flask-docker.*



*Gambar 2.7Menjalankan docker image*

Pada gambar 1.7 merupakan hasil dari menjalankan Docker Image menggunakan port 5000 untuk browser dari sistem host. Lakukan push dari Python ke repositori kosong pada Docker, sebelum melakukan push berikan tag dengan nama *latest* dengan command : *docker tag flask-docker:latest arepdana/flask-docker:latest* .arepdana/flask-docker merupakan nama dari repositori kosong yang dibuat di DockerHub. Push Docker Image ke repositori yang sudah dibuat sebelumnya di Docker Hub dengan perintah berikut : *docker push arifian823/flask-docker:latest.*

**

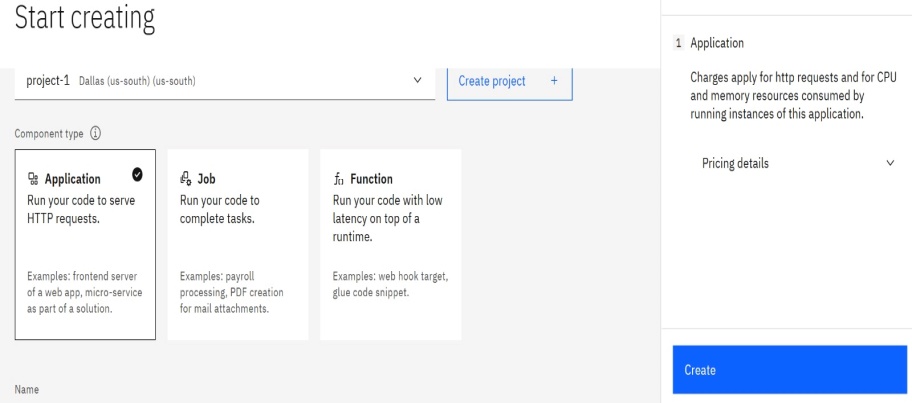
**

**

*Gambar 2.8 Pembuatan Token pada DockerHub*

Pada gambar 1.8 merupakan pembuatan Token pada DockerHub, pada halaman Security terdapat personal access tokens untuk Generate new token. Read Write, Delete merupakan opsi untuk mendapatkan token yang akan digunakan, dan pada saat selesai akan mendapatkan token yaitu : *dckr\_pat\_\_S1DugqSW5ZcBx-rZYBcJTs8q0I* yang akan digunakan untuk ke IBM CLOUD.

**2.6 Proses Deployment ke IBM CLOUD**

****

*Gambar 2.9 Create project Code Engine*

Pada gambar 1.9 merupakan tampilan dari IBM CLOUD pada akses layanan Code Engine, create project baru pada Code Engine untuk mendapatkan link public API endpoint. IBM Cloud Code Engine adalah platform tanpa server yang dikelola sepenuhnya yang dapat digunakan untuk menghosting aplikasi asli cloud. Setelahnya, akan ada pilihan project di halaman pembuatan Application pada Code Engine, silakan pilih bagian 'Application' Pastikan isi dari data yang diperlukan sebagai berikut :

**Location** : Dallas (us-south)

**Name** : project-1

**Resource Group** : Default

**Tags** : Biarkan kosong

Pada kolom Configure Image pastikan data yang diisi sebagai berikut :

**Secret name** : bebas, pada kasus ini : code-engine

**Secret contents** : Docker Hub

**Username** : arepdana (sesuaikan nama dengan DockerHub yang sudah dibuat)

**Registry Server** : <https://index.docker.io/v1/>

**Registry Secret** : Create Registry Secret

Registry server pada IBM Cloud Code Engine adalah layanan yang memungkinkan pengguna untuk menyimpan, mengelola, dan mengintegrasikan gambar kontainer (container images) yang digunakan dalam pengembangan aplikasi. Dan pada Registry Secret adalah tempat untuk memasukkan Access Token yang telah digenerate pada DockerHub, Access Token yang didapatkan adalah *dckr\_pat\_\_S1DugqSW5ZcBx-rZYBcJTs8q0I,* masukkan token ini pada kolom Access Token/API Key.

Pada bagian Ressource & Scaling pastikan data yang diisi sebagai beriku :

**CPU dan memory** : sesuai kebutuhan, pada kasus ini, yang akan digunakan : 2vCPU / 4GB

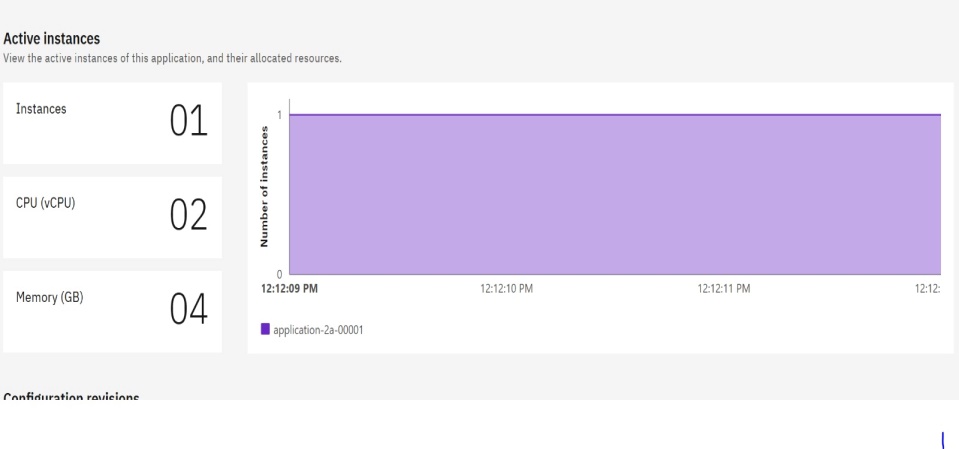
**Ephemeral Storage** : sesuai kebutuhan, pada kasus ini, yang akan digunakan : 1,04 GB

**Min number of instance** : 1 (jangan lebih dari 1, itu sudah cukup untuk mempertahankan API tetap hidup walau tidak dipakai)

**Max number of instance** : Biarkan di angka 10

**Request timeout** : 300

Pada request timeout bisa diatur pada angka 300 milidetik, karena pada kasus ini yang kita butuhkan pada angka 300 milidetik.



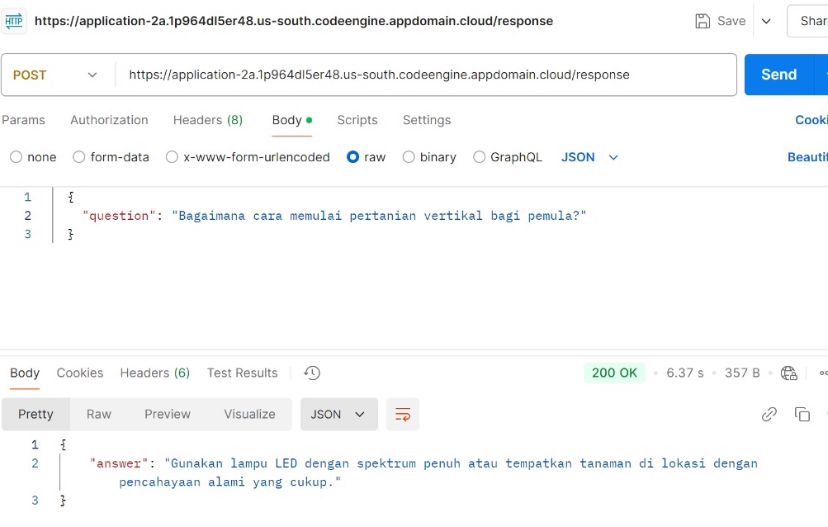
****

*Gambar 2.10 Tampilan proses Deployment*

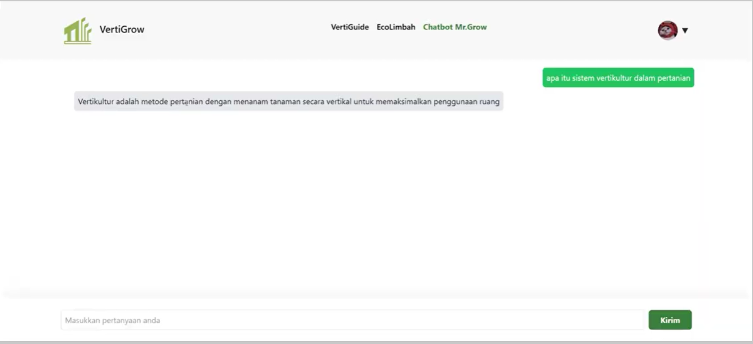
Pada gambar 1.10 merupakan hasil dari proses Deployment pada IBM CLOUD. Untuk mendapatkan link utama pilih bagian 'Domain mappings', Link public tersebut adalah yang akan digunakan untuk melakukan testing, disebut juga sebagai API Endpoint yaitu : *https://flask-docker.1i36eco0q6b5.us-south.codeengine.appdomain.cloud,* link inilah yang akan digunakan dalam proses selanjutnya yang bisa di kembangkan lagi melalui web.

**3. HASIL DAN PEMBAHASAN**

Pada penilitian ini, Chatbot model T5 sudah berhasil dibuat dan bisa di kembangkan lagi melalui web. Model ini dilatih dan diuji dengan menggunakan dataset yang terdiri dari 150 data. Hasil dari evaluasi model bisa di tes melalui postman. Postman adalah platform API yang digunakan untuk membangun, menguji, dan mengelola API. Postman memiliki antarmuka yang mudah digunakan dan fitur-fitur yang membantu menyederhanakan proses kerja dengan API. Simulasi model ini menggunakan postman untuk memastikan proses projek sudah sesuai, dan percobaan melalui contoh web yang sudah disediakan seperti pada gambar 3.1 dan pada gambar 3.2



*Gambar 3.1hasil dari Postman*

**

*Gambar 3.2 percobaan melalui web*

**4. KESIMPULAN**

Penelitian ini berhasil mengembangkan Chatbot model T5 dengan menggunakan IBM CLOUD untuk memproses pengembangan Chatbot. model meliputi pengumpulan data, preprocessing data, pemilihan dan pelatihan model, serta evaluasi model. Penelitian ini menggunakan dataset pertanyaan dan serta jawaban yang akan dijawab oleh Chatbot model T5.

Hasil dari penilitian ini menunjukkan bahwa Chatbot model T5 dikembangkan mencapai akurasi 90% dalam menjawab pertanyaan kompleks dan memproses perintah dalam waktu 300 milidetik, sehingga model ini bisa dapat dikembangkan lagi kedepannya.

Perangkat lunak yang digunakan untuk memprogramming pemodelan machine learning ini adalah python dan di deploy ke IBM CLOUD untuk pengelolaan data, dan pengembangan Chatbot.

Untuk penelitian selanjutnya, disarankan untuk mengembangkan model ini dengan menggunakan algoritma lain untuk membandingkan kinerja dari algoritma yang dipakai. Kemudian diharapkan untuk dapat mengumpulkan dataset yang lebih banyak dan beragam dengan tujuan untuk memperluas cakupan prediksi model.

**DAFTAR PUSTAKA**

A. Wijoyo, A. Y. Saputra, S. Ristanti, R. Sya’ban, M. Amalia, and R. Febriansyah, “Pembelajaran Machine Learning”.

Dwi Sulistyowati, S. Sunarno, and D. D. Djuniadi, “PENERAPAN MACHINE LEARNING DENGAN ALGORITMA SUPPORT VECTOR MACHINE UNTUK PREDIKSI KELEMBAPAN UDARA RATA-RATA,” 2024. [Online]. Available: https://jurnal.umj.ac.id/index.php/just-it/index

Eka Yuniar, & Heri Purnomo. (2019). Implementasi Chatbot “Alitta” Asisten Virtual Dari Balittas Sebagai Pusat Informasi Di Balittas. *Antivirus : Jurnal Ilmiah Teknik Informatika*, *13*(1), 24–35. https://doi.org/10.35457/antivirus.v13i1.714

Hasoloan, A., & Nurhayati, S. (2021). Pengaruh Kualitas Pelayanan Beauty Presenter. *Jurnal Publik Reform UNDHAR MEDAN*, *8*(1), 39–50

IBM CLOUD Products Code Engine “Kasus penggunaan tanpa server’

J. T. Santoso, S. Kom, M. Kom, and A. Machine, “P Y YAYASAN PRIMA AGUS TEKNIK Learning Dengan Python.”

M. Riziq sirfatullah Alfarizi, M. Zidan Al-farish, M. Taufiqurrahman, G. Ardiansah, and M.Elgar, “PENGGUNAAN PYTHON SEBAGAI BAHASA PEMROGRAMAN UNTUK MACHINE LEARNING DAN DEEP LEARNING,” 2023

Musaamin web “Cara Membuat Docker image dan Upload ke Docker Hub’ [Online] Available :https://musaamin.web.id/cara-membuat-docker-image-upload-docker-hub/

Muliyono, M., & Sumijan, S. (2021). Identifikasi Chatbot dalam Meningkatkan Pelayanan Online Menggunakan Metode Natural Language Processing. *Jurnal Informatika Ekonomi Bisnis*, *3*, 142–147. https://doi.org/10.37034/infeb.v3i4.102

Nugroho, I. S., & Voutama, A. (2024). Implementasi Chat Bot Untuk Pelayanan Pelanggan Yang Terintegrasi Web Toko Komputer. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *8*(3), 3132–3136. https://doi.org/10.36040/jati.v8i3.9630