

Aplikasi Layanan dan Informasi Akademik Berbasis *Chatbot* Telegram Menggunakan *Natural Language Processing*

Maulida Qalimaturrahmah ^{1*}, Dwi Budi Santoso ²

^{1,2} Program Studi Sistem Informasi, Fakultas Teknologi Informasi dan Industri, Universitas Stikubank, Kota Semarang, Provinsi Jawa Tengah, Indonesia.

article info

Article history:

Received 4 December 2023

Received in revised form

7 February 2024

Accepted 25 March 2024

Available online April 2024.

DOI:

<https://doi.org/10.35870/jti.k.v8i2.1887>

Keywords:

Chatbot; Natural Language Processing; Rasa; Telegram.

Kata Kunci:

Chatbot; Natural Language Processing; Rasa; Telegram.

abstract

In the digital era, technological advances change various aspects of life. One of the significant technological advances is Chatbot technology. In the world of higher education, a problem often experienced by students today is a lack of literacy regarding important information available on faculty websites, because not all students always access the website. With Chatbot technology based on Natural Language Processing using the Rasa framework which is integrated into the Telegram platform, it will make it easier for students to get information more quickly and efficiently. The method for creating this Chatbot is literature study, data collection, data pre-processing, flavor framework, implementation, system testing, and results. The data used in creating the Chatbot was taken from information contained on the faculty website. Modification of the Rasa Framework file with a total of 47 intents and 394 examples, training and testing processes carried out via command prompt, and integration of the Chatbot into Telegram via Visual Studio Code. From the test results, the Chatbot that has been integrated into Telegram can be used well, the response given by the Chatbot is correct.

abstract

Dalam era digital, kemajuan teknologi merubah berbagai aspek kehidupan. Salah satu kemajuan teknologi yang signifikan adalah teknologi Chatbot. Pada dunia Pendidikan tinggi, permasalahan yang sering dialami mahasiswa saat ini adalah kurangnya literasi terhadap informasi penting yang tersedia di website fakultas, karena tidak semua mahasiswa selalu mengakses website. Dengan teknologi Chatbot berbasis Natural Language Processing menggunakan framework Rasa yang diintegrasikan dalam platform telegram, akan memudahkan mahasiswa mendapatkan informasi lebih cepat dan efisien. Metode pembuatan Chatbot ini dengan studi literatur, pengumpulan data, data pre-processing, framework Rasa, implementasi, pengujian system, dan hasil. Data yang dipakai dalam pembuatan Chatbot diambil dari informasi yang terdapat didalam website fakultas. Modifikasi file Framework Rasa dengan total 47 intent dan 394 exemples, proses training dan testing dilakukan melalui command prompt, dan integrasi Chatbot ke telegram melalui visual studio code. Dari hasil pengujian, Chatbot yang telah diintegrasikan ke telegram dapat digunakan dengan baik, response yang diberikan Chatbot tepat.

Corresponding Author. Email: maulidaqalimaturrahmah@mhs.unisbank.ac.id ^{1}.

© E-ISSN: 2580-1643.

Copyright © 2024 by the authors of this article. Published by Lembaga Otonom Lembaga Informasi dan Riset Indonesia (KITA INFO dan Riset). This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License. 



ACM Computing Classification System (CCS)

 EBSCOhost

Communication and Mass Media Complete (CMMC)

1. Latar Belakang

Pendidikan tinggi merupakan salah satu pilar utama dalam pengembangan sumber daya manusia yang berkualitas dan berdaya saing tinggi. Dalam era digital saat ini, teknologi informasi dan komunikasi telah mengubah berbagai aspek kehidupan, termasuk dunia pendidikan. Salah satu perkembangan teknologi yang signifikan adalah teknologi *Chatbot*. Teknologi ini telah membuka peluang baru dalam penyediaan layanan dan informasi di berbagai bidang, termasuk pendidikan. Dalam dunia pendidikan tinggi, mahasiswa sering menghadapi tantangan dalam mengakses informasi akademik melalui website, karena tidak semua mahasiswa selalu mengakses website kampus atau fakultas, seperti halnya dalam mencari informasi pengumuman penting terkait kuliah, atau berita kampus. Proses mencari informasi ini terkadang memakan waktu dan bisa menjadi rumit, terutama jika mahasiswa harus mengakses berbagai platform atau menghubungi banyak pihak terkait. Oleh karena itu, pengembangan aplikasi layanan dan informasi akademik berbasis *Chatbot* dapat menjadi alternatif yang menarik untuk memberikan akses informasi yang lebih efisien dan cepat bagi mahasiswa.

Chatbot merupakan sebuah program software yang dibangun untuk dapat berinteraksi melalui kata-kata tertulis atau lisan dengan pengguna. *Chatbot* telah digunakan di banyak bidang, namun penggunaan tersebut masih sedikit penerapannya pada lingkungan akademik. Dalam penelitian yang dilakukan oleh Mutiara Zaafira, dimana membuat *Chatbot* dengan menggunakan development RASA, dapat dihasilkan hasil pengujian alfa dengan akurasi sebesar 100% dan pengujian beta 93% dari 30 responden memilih setuju dan sangat setuju, pengujian akurasi system 90% dari 5 pengguna. Hasil tersebut dilakukan untuk menguji apakah *Chatbot* dapat memberikan jawaban yang sesuai dengan input pengguna. Selain itu dengan uji reliabilitas kuesioner dengan hasil 0.955 dan uji validitas kuesioner pada tugas akhir ini dinyatakan valid. Dari hasil pengujian system dapat berjalan sesuai dengan tujuannya [1].

Dalam penelitian Ariyan Zubaidi (2019), yang memanfaatkan *Chatbot* pada platform telegram untuk layanan dan informasi akademik di program studi Teknik informatika universitas mataram. Bot yang

dibangun menggunakan Bahasa pemrograman python, SQLite dan React JS. Prototipe bot yang dibuat mampu menyiarkan informasi terbaru kepada user dan memberikan layanan akademik seperti layanan tesis dan magang. Dari hasil pengujian yang dilakukan juga didapat masing-masing command yang dibuat mampu dieksekusi dengan baik dan respon yang diberikan sesuai dengan yang diinginkan oleh pengguna [2].

Teddy Surya Gunawan *dkk* (2021), melakukan penelitian dengan mengembangkan *Chatbot* telegram menggunakan Bahasa alami dimana *Chatbot* dapat mengenali emosi dengan algoritma deep learning untuk mengekstrak bentuk emosi dari inputan teks. Dengan menggunakan python ver. 3.9.0, dan python telegram bot ver 13.7. Framework yang digunakan adalah google dialogflow, dan metode yang digunakan adalah analisis sentimen. Hasil yang terlihat *Chatbot* dapat mengekstrak emosi dari teks yang diberikan user, yaitu user sedang merasa sedih, kemudian *Chatbot* merespon dengan menyarankan cara untuk membantu user. Dan jika *Chatbot* tidak paham dengan pertanyaan yang diberikan user, maka *Chatbot* akan merespon bahwa ia tidak memahami pertanyaannya [3]. Mhd. Furqan *dkk* (2023), dalam penelitiannya membuat *Chatbot* dengan platform telegram menggunakan *Natural Language Processing* untuk meningkatkan keefisienan dalam *Frequently Ask Question* (FAQ) yang terdapat dalam website Poltekpar. Metode yang digunakan adalah *System Development Life Cycle* (SDLC). Dalam penelitian ini melakukan proses pembuatan system dengan menggunakan metode *Natural Language Processing* (NLP) yang diterapkan pada sistem yang dibangun dengan Telegram Messenger dan Bahasa pemrograman Python. Hasil *Chatbot* yang diuji mendapat nilai persentase sebesar 94%. Hasil ini menunjukkan sistem *Chatbot* sangat layak dan efektif dalam membantu mahasiswa memperoleh informasi yang dibutuhkan [4].

Dalam penelitian yang dilakukan oleh Alifya Hikmah *dkk* (2023), dengan mengimplementasikan *Natural Language Processing* (NLP) pada *Chatbot* untuk layanan akademik. Pengujian pada sistem *Chatbot* ini menggunakan metode pengujian beta dengan Usability Testing yang mendapatkan hasil sebesar 93% dari total 42 responden dan 6 pertanyaan. Untuk pengujian akurasi pada *Chatbot* mendapatkan hasil

100% dari total 54 pertanyaan yang diajukan secara acak. Dari hasil pengujian yang telah dilakukan, dinyatakan bahwa *Chatbot* ini mampu memberikan response dengan menilai jawaban dari pertanyaan yang diberikan oleh pengguna berdasarkan kata kunci pada *Chatbot*, sehingga *Chatbot* dapat memberikan jawaban yang tepat [5]. Laksmi Anindyati juga melakukan penelitian dengan merancang aplikasi *Chatbot* penerimaan mahasiswa baru politeknik astra menggunakan framework RASA. Hasil dari penelitian ini adalah kebutuhan sistem yang direpresentasikan pada use case diagram dan flowchart lalu pemilihan pipeline NLU untuk *Chatbot*, arsitektur sistem, perancangan database dalam bentuk physical data model, dan perancangan desain antarmuka (*mockup*) sistem penunjang *Chatbot* framework RASA [6].

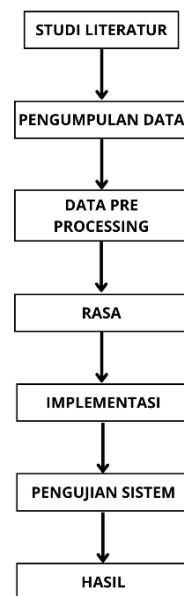
Berdasarkan dari referensi yang didapat dari hasil penelitian terdahulu yang telah disebutkan, peneliti menggunakan referensi tersebut sebagai bahan acuan dalam penelitian yang akan dilakukan. Pada penelitian ini akan menggunakan platform telegram dalam pembuatan *Chatbot* untuk layanan informasi akademik fakultas teknologi informasi dan industry, dengan metode *Natural Language Processing* (NLP). Pemilihan metode NLP sendiri karena NLP dapat membuat *Chatbot* berkomunikasi dengan user menggunakan bahasa sehari-hari tanpa perlu mematuhi format atau bahasa yang kaku. Dengan mengimplementasikan teknologi NLP ke dalam Telegram, dapat diciptakan sebuah *Chatbot* yang mampu memberikan layanan dan informasi akademik kepada mahasiswa dengan cepat dan efisien.

Dalam pembuatan *Chatbot* telegram ini akan berfokus menggunakan framework RASA. Dipilihnya framework RASA karena memiliki performa yang baik, sebab adanya Rasa NLU dan Rasa Core. RASA bersifat open source, dan memungkinkan pengembang untuk dapat mengontrol sepenuhnya alur percakapan serta memanfaatkan kemampuan NLP untuk memahami niat user. Selain itu framework RASA memiliki pemahaman Bahasa alami yang sangat kuat. Dataset yang digunakan bersumber dari website fakultas yang akan diubah dalam file *domain.yml*. Adapun tujuan dari penelitian ini adalah dapat membantu mahasiswa dalam mencari informasi tentang berita di lingkup

Fakultas, pengumuman kuliah, syarat kelulusan, pengumuman kelulusan dan berbagai informasi penting lainnya dengan hanya mengakses *Chatbot* melalui *platform* telegram saja.

2. Metode Penelitian

Dalam penelitian yang akan peneliti lakukan terdapat objek penelitian yang akan digunakan yaitu pada Fakultas Teknologi Informasi dan Industri yang ada di Universitas Stikubank Semarang. Adapun Teknik pengambilan data dilakukan secara manual dengan bersumber dari website Fakultas Teknologi Informasi dan Industri Universitas Stikubank Semarang, yang kemudian data tersebut diubah ke dalam file *domain.yml*.



Gambar 1. Alur Penelitian

Dari gambar 1 merupakan alur dari penelitian yang dilakukan oleh peneliti, dimulai dari proses mencari studi literatur, pengumpulan data, melakukan data *preprocessing*, kemudian masuk pada framework RASA, lalu tahap implementasi, setelah itu melakukan pengujian system, dan hasil. Berikut ini penjelasan dari masing-masing tahap pada alur penelitian yang dilakukan.

Studi Literatur

Pada tahap studi literatur ini peneliti melakukan pengumpulan berbagai sumber literatur atau sumber informasi atau Pustaka lainnya yang berkaitan dengan

penelitian yang dilakukan. Studi literatur ini dapat bersumber dari jurnal ilmiah dari penelitian terdahulu yang selaras dengan tema penelitian yang kemudian dapat dijadikan bahan acuan.

Pengumpulan Data

Dalam pengumpulan data pada penelitian ini, peneliti mengambil data yang bersumber dari website fakultas teknologi informasi dan industri serta wawancara langsung dengan pihak Kepala Biro Administrasi Akademik (BAAK) Fakultas. Data yang diperoleh dapat berupa pertanyaan atau permasalahan yang sering diajukan oleh mahasiswa fakultas ke pihak BAAK.

Data PreProcessing

Pada metode *Natural Language Processing*, terdapat proses preprocessing, antara lain:

1) *Case Folding*

Proses dimana mengubah seluruh teks menjadi huruf besar atau huruf kecil, serta menghilangkan karakter yang tidak perlu seperti tanda baca, symbol atau angka.

2) *Tokenisasi*

Proses memisahkan teks menjadi kata per kata individu. Dimana suatu kata ini tidak terkait dengan kata lainnya.

3) *Stop Words*

Proses dimana kata-kata penghubung seperti “dan”, “atau” yang tidak memiliki makna penting akan di hapus.

4) *Stemming*

Proses mengubah kata-kata menjadi kata dasar dengan menghilangkan unsur sufiks atau kata imbuhan pada kata, misalnya: “mendaftar” menjadi “daftar”.

Framework RASA

Setelah melalui data *preprocessing*, selanjutnya adalah masuk pada *framework* RASA. RASA merupakan sebuah *framework* yang bersifat open source, RASA adalah kombinasi dari dual modul yaitu RASA NLU (*Natural Language Understanding*) pemrosesan Bahasa alami dan RASA Core yang dapat memprediksi dan memberikan respon kepada *user*. Data yang telah dikumpulkan oleh peneliti akan diolah dalam *framework* RASA, dimana data informasi akan dimasukkan dalam file *domain.yml*, kemudian contoh pertanyaan yang mungkin akan ditanyakan

dimasukkan dalam file *nlu.yml* pada bagian *intent* dan *examples*.

Implementasi

Setelah semua data telah diolah dalam *framework* RASA, selanjutnya yang dilakukan adalah mengintegrasikan *Chatbot* RASA ke bot telegram yang telah dibuat dengan menggunakan Telegram API.

Pengujian Sistem

Proses pengujian sistem dilakukan dengan menguji *Chatbot* melalui telegram bot yang telah dibuat, apakah *Chatbot* dapat memberikan respon sesuai dengan apa yang diinginkan oleh user atau tidak.

Hasil

Hasil akhir adalah terciptanya sebuah *Chatbot* layanan informasi akademik fakultas teknologi informasi dan industri menggunakan platform telegram.

Metode pengembangan sistem

Instrument Penelitian

Selain metode penelitian, adapun instrument penelitian yang perlu dilakukan oleh peneliti. Instrument ini diperlukan agar mengetahui apa saja yang diperlukan oleh peneliti dalam merealisasikan pembuatan *Chatbot* telegram. Instrument ini dibagi dua, yaitu perangkat lunak (*software*), dan perangkat keras (*hardware*).

1) *Perangkat Lunak (Software)*

Adapun perangkat lunak yang digunakan oleh peneliti sebagai berikut:

- a) System operasi PC: Windows 10
- b) Bahasa pemrograman: python
- c) Framework: rasa
- d) Tools editor: Visual Studio Code, Command Prompt

2) *Perangkat Keras (Hardware)*

Perangkat keras yang digunakan oleh peneliti sebagai berikut:

- a) Perangkat: Laptop Asus X441UB
- b) Ram: Minimum 1 GB
- c) Smartphone: Android 10

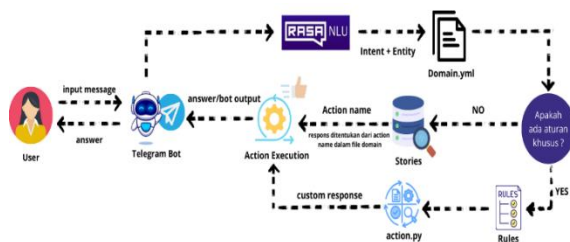
Desain Sistem Chatbot

Chatbot telegram yang dirancang oleh peneliti ini menggunakan *framework* Rasa. Dalam *framework* Rasa terdapat beberapa komponen yaitu:

- 1) NLU (*Natural Language Understanding*)
NLU adalah bagian dari sistem *Chatbot* yang bertanggung jawab untuk memahami niat (*intent*) dan entitas (*entities*) dari input teks yang diberikan oleh pengguna. Dalam NLU terdapat 2 komponen utama yaitu *intent* dan *examples* (contoh atau sampel input pertanyaan dari pengguna) atau sering disebut sebagai “training data”.
- 2) Domain
Domain adalah tempat dimana mendefinisikan semua elemen yang dibutuhkan untuk memahami dan merespons percakapan pengguna dengan *Chatbot*. Ini mencakup *intent*, *respons*, *template respons (utterances)*, *pengaturan sesi (session settings)*.
- 3) Rules
Rules adalah sebuah aturan-aturan sederhana atau sebuah kebijakan untuk memberikan sebuah respons secara langsung. Ini berarti jika dalam kondisi tertentu terpenuhi, bot akan memberikan respons tertentu tanpa melibatkan model machine learning atau tanpa perlu mengacu pada Riwayat dialog (*stories*). Dalam rules terdapat beberapa komponen antara lain, *rule*, *steps*, *intent*, dan *action*.
- 4) Story
Stories adalah urutan aksi dan respon yang sudah terdefinisi sebelumnya. Dalam stories juga mendefinisikan dan melatih pola dialog atau interaksi antar pengguna dan *Chatbot*. Setiap story akan mewakili skenario atau percakapan tertentu yang mungkin terjadi selama interaksi dengan *Chatbot*. Dalam stories terdapat beberapa komponen antara lain, *rule*, *steps*, *intent*, dan *action*.

Desain Alur Proses Chatbot

Berdasarkan dari desain system *Chatbot*, maka diperoleh alur proses *Chatbot* sebagai berikut.



Gambar 2. Alur Proses Chatbot

Dari gambar 2 dapat dijelaskan alur dari proses *Chatbot* seperti dibawah ini:

- 1) User memberikan input berupa pesan *text* kepada telegram bot dengan *framework* rasa.
- 2) Kemudian telegram bot melihat ke dalam file *rasa* NLU untuk mencari apakah ada maksud (*intent*) dan entitas (jika ada) yang ditemukan dalam user input.
- 3) Telegram bot mencari respons dalam file domain berdasarkan maksud (*intent*) yang diinputkan *user*.
- 4) Lalu telegram bot akan memeriksa apakah ada aturan khusus dalam memberikan respons.
- 5) Jika ada aturan khusus yang ditulis, maka akan merujuk pada rules yang telah dibuat, *Chatbot* akan mempertimbangkannya disini, kemudian akan merujuk pada file action, setelah itu akan mengambil tindakan/aksi (*action execution*) dan mengirimkan respons pada telegram bot, kemudian akan memberikan jawaban kepada *user*.
- 6) Jika tidak ada aturan khusus yang ditulis, maka system akan memeriksa stories yang telah dilatih untuk menemukan pola yang sesuai dengan interaksi user input.
- 7) Lalu dari stories akan memberikan action name, dimana respons ditentukan dari action name yang ada dalam file domain.
- 8) Kemudian bot akan melakukan action execution (mengambil Tindakan/aksi) berdasarkan hasil dari tahap domain dan stories.
- 9) Setelah itu Telegram Bot akan memberikan respons atau jawaban atau bot output kepada user.

3. Hasil dan Pembahasan

Pada hasil dan pembahasan pembuatan *Chatbot* telegram yang dirancang oleh peneliti, terdapat beberapa langkah untuk merealisasikan rancangan *Chatbot*. Berikut ini langkah-langkah dari proses training dan testing menggunakan *framework* rasa.

Install Virtual Environment

Langkah pertama yang harus dilakukan yaitu dengan install virtual environment terlebih dahulu. Peneliti melakukan langkah ini menggunakan *command prompt* dengan perintah, *pip install virtualenv*. Kemudian membuat nama *virtualenv* dengan perintah, *virtualenv rasaenv*.

```

C:\Users\MAULIDA>pip install virtualenv
Collecting virtualenv
  Downloading virtualenv-20.24.6-py3-none-any.whl (3.8 MB)
    Collecting distlib<=0.3.7
      Downloading distlib-0.3.7-py3-none-any.whl (468 kB)
    Collecting platformdirs<=3.9.1
      Downloading platformdirs-3.11.0-py3-none-any.whl (17 kB)
    Collecting filelock<=3.12.2
      Downloading filelock-3.13.1-py3-none-any.whl (11 kB)
Installing collected packages: distlib, platformdirs, filelock, virtualenv
Successfully installed distlib-0.3.7 filelock-3.13.1 platformdirs-3.11.0 virtualenv-20.24.6
WARNING: You are using pip version 20.2.3; however, version 23.1.1 is available.
You should consider upgrading via the 'C:\Users\MAULIDA\AppData\Local\Programs\Python\Python38\python.exe -m pip install --upgrade pip' command.

C:\Users\MAULIDA>virtualenv venv
created virtual environment C:\Users\MAULIDA\venv\python3.8.10\python.exe in 2409ms
creator Python3.8.10\python.exe, clear=False, no_venv_ignore=False, global=False
added seed packages: pip=20.2.3, setuptools=57.5.0, wheel=0.37.0
activators: BatchActivator, BatchActivator, FileActivator, PowerShellActivator, PythonActivator

```

Gambar 3. Install Virtual Environment

Install Framework Rasa

Framework RASA merupakan sebuah kerangka kerja yang bersifat open source untuk pemrosesan Bahasa alami. Umumnya terdapat dua tahap pemrosesan teks pada sebuah system *Chatbot* atau system percakapan berbasis AI. Tahap pertama adalah bagaimana system dapat menginterpretasikan teks dari user. Tahap kedua adalah bagaimana system dapat menghasilkan output teks respon. Framework Rasa ini Dapat terintegrasi dengan platform komunikasi lain melalui API. Sebelum install rasa, peneliti mengaktifkan terlebih dahulu virtualenv yang telah dibuat, dengan perintah : `.\rasaenv\Scripts\activate` pada command prompt. Setelah itu install rasa pada virtualenv yang telah aktif, dengan perintah: `pip install rasa`.

```

C:\Users\MAULIDA>.\rasaenv\Scripts\activate
(rasaenv) C:\Users\MAULIDA>pip install rasa
Collecting rasa
  Downloading rasa-3.6.13-py3-none-any.whl.metadata (28 kB)
Collecting CacheControl<0.13.0,>=0.12.9 (from rasa)
  Downloading CacheControl-0.12.14-py3-none-any.whl.metadata (2.2 kB)
Collecting PyJWT<3.0.0,>=2.0.0 (from PyJWT[crypto]<3.0.0,>=2.0.0->rasa)
  Downloading PyJWT-2.8.0-py3-none-any.whl.metadata (4.2 kB)
Collecting SQLAlchemy<1.5.0,>=1.4.0 (from rasa)
  Downloading SQLAlchemy-1.4.50.tar.gz (8.5 MB)
    Preparing metadata (setup.py) ... done
Collecting aio-pika<8.2.4,>=6.7.1 (from rasa)
  Downloading aio-pika-8.2.3-py3-none-any.whl (49 kB)

```

Gambar 4. Install Framework RASA

Menginisialisasi Framework rasa

Pada tahap ini, peneliti melakukan inisialisasi pada proyek rasa setelah dilakukan instalasi. Inisialisasi proyek menggunakan perintah: `rasa init` pada command prompt. Tujuan dilakukannya inisialisasi ini adalah membuat struktur dasar pada proyek rasa, termasuk file-file konfigurasi dan data dasar.

Modifikasi Dataset

Setelah melakukan inisialisasi, selanjutnya memodifikasi dataset sesuai dengan hasil data yang

telah diperoleh peneliti. Adapun beberapa file yang harus dimodifikasi seperti: `data/nlu.yml`, `data/rules.yml`, `data/stories.yml`, `domain.yml`.

1) Data/nlu.yml

Pada data `nlu.yml` ini berisi contoh-contoh dari teks pertanyaan yang mungkin akan ditanyakan oleh user. File ini digunakan untuk melatih model pemahaman Bahasa alami (NLU). Peneliti memodifikasi file ini dengan total 47 intent atau topik, dan 394 examples atau contoh pertanyaan yang akan digunakan untuk membandingkan dengan pertanyaan yang diinputkan oleh user, apakah pertanyaan ada atau tidak.

```

! nlu.yml
1 version: "3.1"
2
3 nlu:
4 - intent: greet
5   examples: |
6     - hey
7     - hello
8     - hi
9     - halo
10    - halo semuanya
11    - selamat pagi
12    - good
13    - selamat siang
14    - selamat sore
15    - sore
16    - selamat malam
17    - malam
18    - good morning
19    - good evening
20    - good night
21    - good afternoon
22    - ayo
23    - start
24    - malah
25
26 - intent: pertanyaan
27   examples: |
28     - mau tanya
29     - tanya dong
30     - tanya
31

```

Gambar 5. Data/nlu.yml

2) Data/rules.yml

Dalam file `rules.yml` ini berisi aturan yang menentukan perilaku *Chatbot* berdasarkan dengan kondisi tertentu. Aturan ini berguna untuk menetapkan Tindakan tertentu Ketika suatu kondisi yang ditentukan terpenuhi. Pada file `rules.yml` yang peneliti buat ini berisikan suatu tindakan dimana *Chatbot* akan memberikan respons dengan action yang ada dalam `utter_goodbye`, setiap user menginputkan kata-kata goodbye.

```

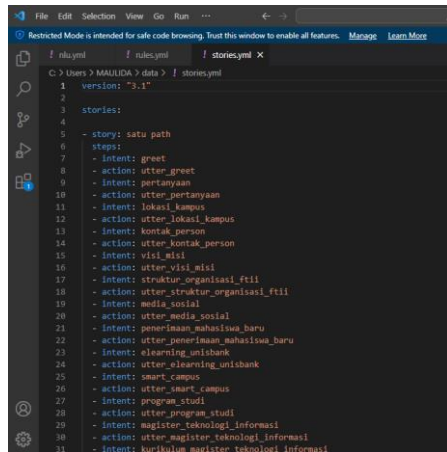
! rules.yml
1 version: "3.1"
2
3 rules:
4
5 - rule: Say goodbye anytime the user says goodbye
6   steps:
7     - intent: goodbye
8     - action: utter_goodbye
9
10 - rule: Say 'I am a bot' anytime the user challenges
11   steps:
12     - intent: bot_challenge
13     - action: utter_iambat
14

```

Gambar 6. Data/rules.yml

3) Data/stories.yml

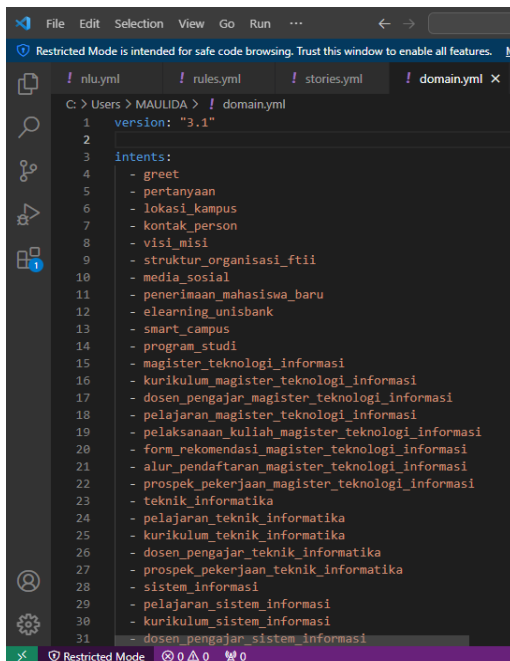
File ini berisi urutan percakapan atau riwayat percakapan “stories” yang digunakan untuk melatih model dialog *Chatbot*. Dimana setiap story terdiri dari serangkaian langkah-langkah yang terdiri dari intent user dan tindakan atau action yang diberikan oleh *Chatbot*.



Gambar 7. Data/stories.yml

4) Domain.yml

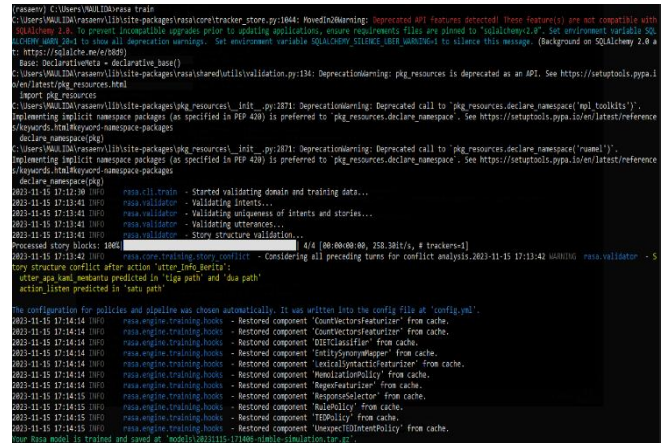
Dalam file ini menggambarkan sebuah domain atau konteks dari *Chatbot*, dimana di dalam file terdapat intent, action (Tindakan), respons, entitas dan konfigurasi lainnya. Pada file domain.yml ini yang akan menjawab dari file yang ada pada nlu.yml.



Gambar 8. File Domain.yml

5) Training Dataset

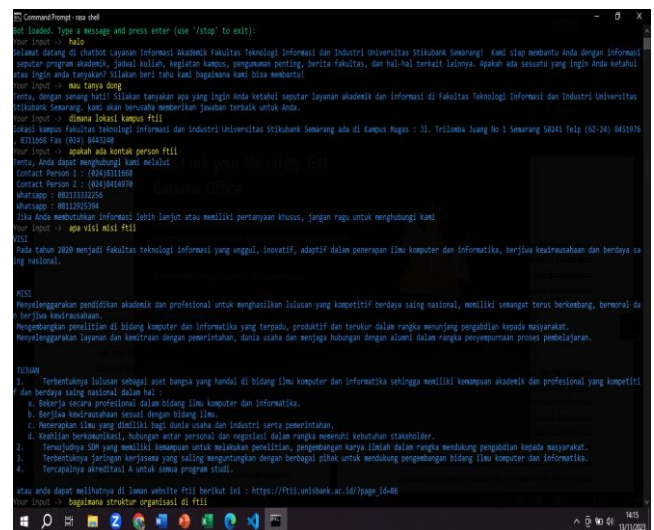
Setelah melakukan modifikasi pada beberapa data, langkah yang selanjutnya harus dilakukan adalah melatih data atau proses training model. Peneliti melakukan proses training data pada *command prompt* dengan perintah: *rasa train*.



Gambar 9. Proses Training Dataset

6) Testing Model

Langkah selanjutnya setelah melakukan training dataset adalah melakukan testing model. Testing model ini dilakukan pada command prompt dengan perintah: *rasa shell*. Berikut ini hasil testing model yang dilakukan pada *command prompt*.



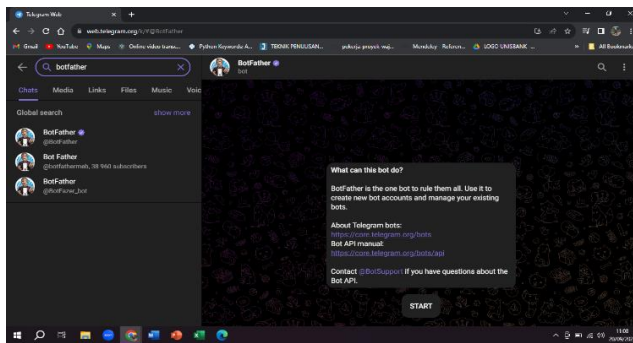
Gambar 10. Proses Training Dataset Integrasi Rasa ke Platform Telegram

Setelah proses instalasi sampai pada proses testing model atau dataset. Langkah selanjutnya yang harus dilakukan adalah mengintegrasikan atau menghubungkan program rasa ke *platform telegram*.

Berikut ini langkah-langkah yang harus dilakukan untuk menghubungkan program rasa ke *platform* telegram:

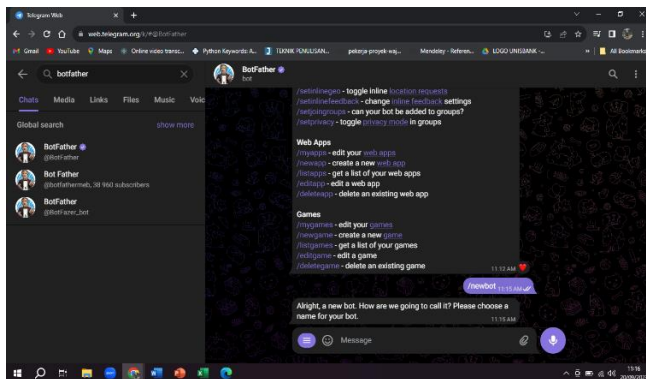
1) Membuat akun bot baru

Langkah pertama yang harus dilakukan adalah dengan membuat akun bot baru melalui telegram dengan ketik *BotFather* pada kolom pencarian, kemudian klik start.



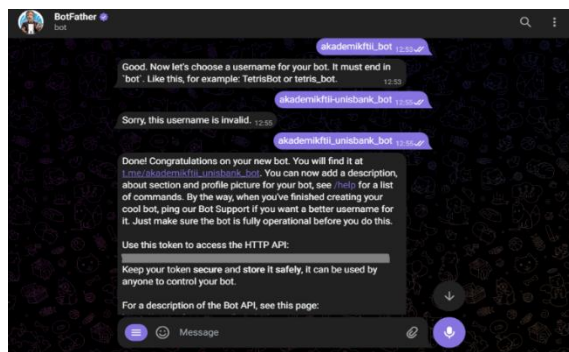
Gambar 11. Langkah pertama membuat akun bot

Setelah klik start, kemudian ketik */newbot* untuk membuat akun baru.



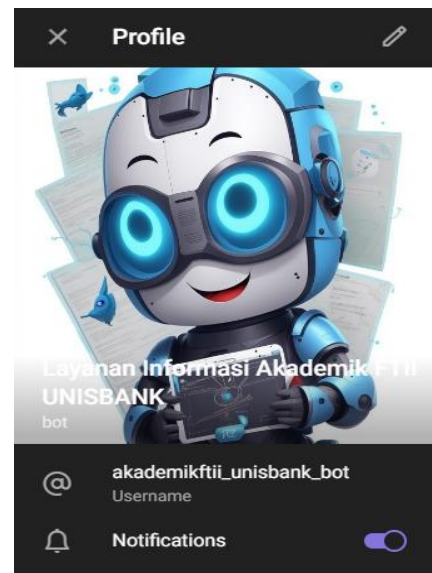
Gambar 12. Langkah kedua membuat akun bot

Kemudian ketik nama akun bot baru yang akan dibuat, dan username untuk akun *Chatbot*.



Gambar 13. Langkah ketiga membuat akun bot

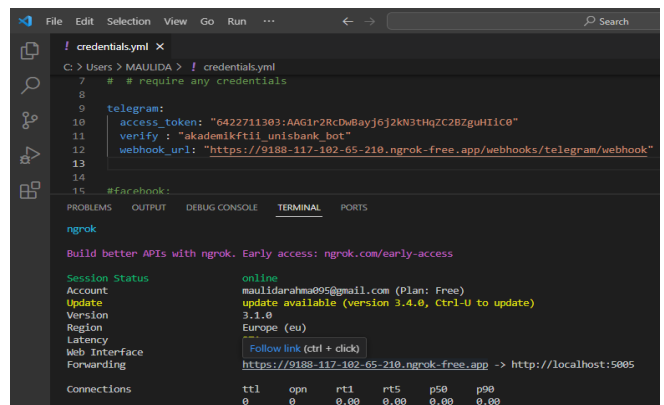
Berikut tampilan akun bot yang telah berhasil dibuat.



Gambar 14. Tampilan Profil Akun *Chatbot* Telegram

2) Input kode API, *Username* dan url *webhook* pada file *Credentials.yml*.

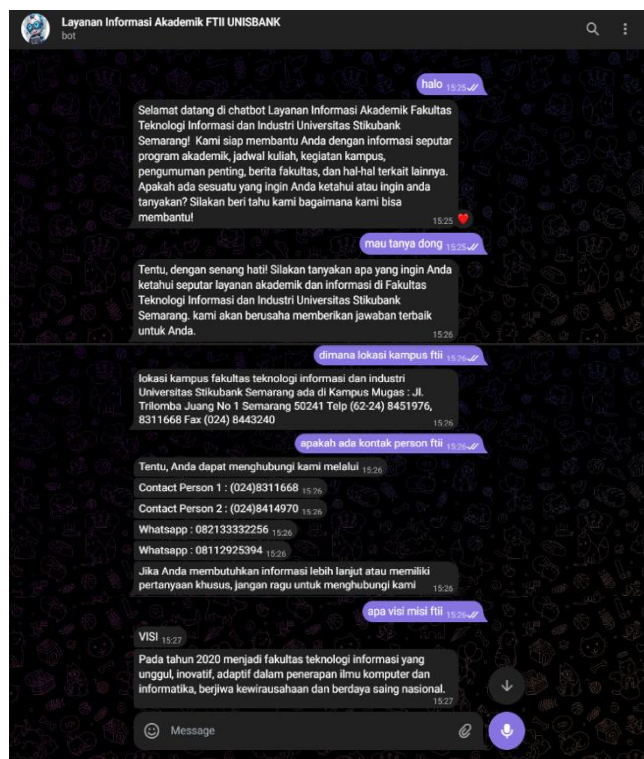
Langkah selanjutnya adalah memasukkan *Token* atau kode API yang didapat dari *BotFather* kemudian diinputkan dalam file *credentials.yml*. Setelah itu input url *webhook* dengan perintah *ngrok http 5005*.



Gambar 15. Tampilan Hasil Input

3) Testing program pada platform Telegram

Jika serangkaian langkah sebelumnya telah berhasil dilakukan. Maka program sudah dapat di uji coba langsung pada platform telegram, dengan membuka akun bot yang telah dibuat. Berikut ini tampilan hasil testing pada platform telegram:

Gambar 16. Tampilan *Chatbot* Telegram

4. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan oleh peneliti. Dataset di peroleh dari informasi yang terdapat dalam website fakultas. Peneliti menggunakan command prompt sebagai uji coba proses training dan testing, dan visual studio code sebagai tools editor. Hasil modifikasi dari beberapa file yang terdapat dalam framework rasa, dimana pada data/nlu.yml dengan total 47 intent atau topik, dan 394 examples atau contoh pertanyaan dapat dilakukan training dan testing dengan baik. Hasil yang didapat dari pembuatan *Chatbot* dengan menggunakan framework rasa dengan integrasi ke dalam platform telegram dapat digunakan dengan baik. *Chatbot* telegram ini dapat membantu memudahkan mahasiswa dalam mencari informasi akademik pada lingkup fakultas. Mahasiswa akan lebih cepat mendapatkan informasi dari response *Chatbot* yang tidak perlu menunggu lama.

5. Daftar Pustaka

- [1] Itonsaputri, M. Z., Saputra, R. E., & Nugrahaeni, R. A. (2023). Implementasi Rasa Pada Chatbot Layanan Akademik. *eProceedings of Engineering*, 10(1).
- [2] Zubaidi, A., & Ramdani, R. (2019). Layanan Dan Informasi Akademik Berbasis Bot Telegram Di Program Studi Teknik Informatika Universitas Mataram. *Jurnal Teknologi Informasi, Komputer, Dan Aplikasinya (JTika)*, 1(1), 103-110.
- [3] Gunawan, T. S., Babiker, A. B. F., Ismail, N., & Effendi, M. R. (2021, August). Development of intelligent telegram chatbot using *Natural Language Processing*. In *2021 7th International Conference on Wireless and Telematics (ICWT)* (pp. 1-5). IEEE. DOI: 10.1109/ICWT52862.2021.9678471.
- [4] Furqan, M., Sriani, S., & Shidqi, M. N. (2023). Chatbot Telegram Menggunakan *Natural Language Processing*. *Walisongo Journal of Information Technology*, 5(1), 15-26. DOI: 10.21580/wjit.2023.5.1.14793.
- [5] Hikmah, A., Azmi, F., & Nugrahaeni, R. A. (2023). Implementasi *Natural Language Processing* Pada Chatbot Untuk Layanan Akademik. *eProceedings of Engineering*, 10(1).
- [6] Anindyati, L. (2023). Analisis dan Perancangan Aplikasi Chatbot Menggunakan Framework Rasa dan Sistem Informasi Pemeliharaan Aplikasi (Studi Kasus: Chatbot Penerimaan Mahasiswa Baru Politeknik Astra). *Jurnal Teknologi Informasi Dan Ilmu Komputer (Jtiik)*, 10(2), 291-300.
- [7] Ferdian, A. D., & Anwar, S. N. (2023). Pengembangan Chatbot untuk Informasi Wisata Interaktif di Tangerang Selatan menggunakan Framework Rasa. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, 5(4), 476-483. DOI: <https://doi.org/10.47233/jteksis.v5i4.953>.

- [8] Wulandari, D., & Wibowo, J. S. (2023). Implementasi Chatbot Menggunakan Framework Rasa Untuk Layanan Informasi Wisata Di Kota Pati. *INTECOMS: Journal of Information Technology and Computer Science*, 6(2), 794-801. DOI: <https://doi.org/10.31539/intecom.v6i2.710>.
- [9] Afifa, N., Saputra, R. E., & Nugrahaeni, R. A. (2023). Implementasi NLP Pada Chatbot Layanan Akademik Dengan Algoritma Bert. *eProceedings of Engineering*, 10(1).
- [10] Cannavaro, N. (2023). Aplikasi Chatbot untuk Layanan Akademik Menggunakan Platform RASA Open Source dengan Fitur Two Stage Fallback. *Jurnal Ilmu Komputer dan Informatika*, 3(1), 53-64. DOI: <https://doi.org/10.54082/jiki.73>.
- [11] Mustakim, F., Fauziah, F., & Hayati, N. (2021). Algoritma Artificial Neural Network pada Text-based Chatbot Frequently Asked Question (FAQ) Web Kuliah Universitas Nasional. *Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)*, 5(4), 438-446. DOI: <https://doi.org/10.35870/jtik.v5i4.261>.