

**PENERAPAN NATURAL LANGUAGE UNDERSTANDING PADA
SISTEM CHATBOT INFORMASI PALSU (HOAX) MENGGUNAKAN
DUAL INTENT AND ENTITY TRANSFORMER (DIET)
CLASSIFIER DAN FUZZY STRING MATCHING**

SKRIPSI

WILLI NARDO

181402126



PROGRAM STUDI S1 TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2023

**PENERAPAN NATURAL LANGUAGE UNDERSTANDING PADA
SISTEM CHATBOT INFORMASI PALSU (HOAX) MENGGUNAKAN
DUAL INTENT AND ENTITY TRANSFORMER (DIET)
CLASSIFIER DAN FUZZY STRING MATCHING**

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat ijazah

Sarjana Teknologi Informasi

WILLI NARDO

181402126



**PROGRAM STUDI S1 TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA**

MEDAN

2023

PERSETUJUAN

Judul : PENERAPAN NATURAL LANGUAGE UNDERSTANDING PADA SISTEM CHATBOT INFORMASI PALSU (HOAX) MENGGUNAKAN DUAL INTENT AND ENTITY TRANSFORMER (DIET) CLASSIFIER DAN FUZZY STRING MATCHING

Kategori : Skripsi

Nama Mahasiswa : Willi Nardo

Nomor Induk Mahasiswa : 181402126

Program Studi : Sarjana (S-1) Teknologi Informasi

Fakultas : Ilmu Komputer dan Teknologi Informasi

Universitas Sumatera Utara

Medan, 13 Juni 2023

Komisi Pembimbing:

Pembimbing 2,



Dr. Erna Budhiarti Nababan M.IT
NIP. 196210262017042001

Pembimbing 1,



Prof. Dr. Drs. Opim Salim Sitompul M.Sc
NIP. 196108171987011001

Diketahui/disetujui oleh

Program Studi S-1 Teknologi Informasi

Ketua,



Dedy Arisandi S.T., M.Kom.
NIP. 197908312009121002

PERNYATAAN

**PENERAPAN NATURAL LANGUAGE UNDERSTANDING PADA
SISTEM CHATBOT INFORMASI PALSU (HOAX) MENGGUNAKAN
DUAL INTENT AND ENTITY TRANSFORMER (DIET)
CLASSIFIER DAN FUZZY STRING MATCHING**

SKRIPSI

Saya mengakui bahwa skripsi ini merupakan hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sumbernya.

Medan, 28 Juli 2023

WILLI NARDO

181402126

UCAPAN TERIMA KASIH

Puji dan syukur penulis sampaikan kepada Tuhan yang Maha Esa, karena atas kasih dan berkat-Nya penulis dapat menyelesaikan penyusunan skripsi ini yang merupakan salah satu syarat untuk memperoleh gelar Sarjana pada Program Studi S1 Teknologi Informasi Universitas Sumatera Utara. Penulis tidak dapat menyelesaikan penyusunan skripsi ini tanpa bimbingan, doa, dan dukungan dari berbagai pihak yang terlibat dalam masa perkuliahan hingga penyusunan skripsi ini selesai. Adapun dalam kesempatan ini penulis ingin mengucapkan terima kasih kepada:

1. Keluarga penulis, Ayah Hendra Gaho dan Ibu Vera Suryani yang tiada henti memberikan kasih sayang, doa, dukungan, dan semangat kepada penulis sehingga dapat menyelesaikan skripsi ini.
2. Bapak Dr. Muryanto Amin, S.Sos., M.Si., selaku Rektor Universitas Sumatera Utara.
3. Ibu Dr. Maya Silvi Lydia B.Sc., M.Sc., selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
4. Bapak Prof. Dr. Drs. Opim Salim Sitompul M.Sc., selaku Dosen Pembimbing I dan Ibu Dr. Erna Budhiarti Nababan M.IT., selaku Dosen Pembimbing II yang telah meluangkan waktu untuk membimbing, memberikan masukan, kritik dan saran kepada penulis selama proses pengerjaan dari awal hingga akhir.
5. Bapak Ivan Jaya, S.Si., M.Kom. sebagai Dosen Pembimbing Pertama dan Ibu Rossy Nurhasanah S.Kom., M.Kom. sebagai Dosen Pembimbing Kedua yang banyak memberikan masukan dalam menyelesaikan skripsi ini.
6. Bapak Dedy Arisandi S.T., M.Kom., selaku Ketua Program Studi S1 Teknologi Informasi Universitas Sumatera Utara.
7. Seluruh Dosen Program Studi S1 Teknologi Informasi yang telah memberikan ilmu dan mengajari penulis selama perkuliahan.
8. Para pegawai dan staf yang telah membantu dalam kegiatan administrasi ataupun akademik dari Prodi S1 Teknologi Informasi USU dan Fasilkom-TI USU.

9. Kepada penulis sendiri yang selalu berusaha dan bertahan untuk tidak menyerah dalam menyelesaikan tugas akhir ini meskipun banyak melewati rintangan hidup lainnya selama proses pengerjaan tugas akhir ini.
10. Terima kasih kepada teman-teman seperjuangan penulis semasa kuliah Leonardo Wijaya, Jimmy Widiyanto, Aldrich William Choales, Xixillia Sunaryo, Ammar Rafi Afandi Hasibuan, Alvin Daeli, Rasyid Hafiz, dan Sintya Veronica Rotua Munthe yang telah membantu dan Bersama-sama berjuang juga saling memberikan dukungan, doa, dan semangat di masa perkuliahan sampai selesai penyusunan skripsi ini.
11. Teman-teman Angkatan 2018 Teknologi Informasi USU yang telah Bersama-sama berjuang bersama penulis dari semasa perkuliahan hingga selesai penyusunan skripsi.
12. Semua pihak yang terlibat langsung maupun tidak langsung yang tidak dapat penulis tuliskan satu persatu yang telah membantu dalam penyelesaian skripsi ini.

Medan, 28 Juli 2023

Penulis

Abstrak

Informasi palsu (hoax) merupakan informasi yang tidak benar-benar terjadi dan memiliki dampak negatif yang mampu mendistorsi interpretasi, persepsi, dan anggapan pembaca. Penyebaran informasi palsu (hoax) saat ini di Indonesia terus mengalami peningkatan yang sangat luar biasa. Kondisi ini dapat dikatakan berada di kondisi yang mengkhawatirkan. Apabila kondisi ini tidak ditangani dengan cepat dan tepat, maka tentu akan meningkatkan kesalahpahaman masyarakat yang tinggi akan suatu peristiwa. Maka dari itu, diperlukan suatu pendekatan untuk menjelaskan kebenaran sebuah informasi. Penelitian ini memiliki tujuan untuk mengembangkan sistem chatbot tanya jawab seputar informasi palsu (hoax) dengan menerapkan metode *Dual Intent and Entity Transformer (DIET) Classifier* dan *Fuzzy String Matching*. Penelitian ini menggunakan data tanya jawab yang berasal dari website <https://www.mafindo.or.id/> dan channel youtube Mafindo. Penelitian ini menggunakan *search engine* yang dapat mencari informasi yang berasal dari website <https://www.liputan6.com>, <https://cekfakta.tempo.co>, <http://turnbackhoax.id/>, dan <https://www.kompas.com/>. Pengujian yang dilakukan menggunakan pendekatan metode *Dual Intent and Entity Transformer (DIET) Classifier* menghasilkan akurasi sebesar 0.9.

Kata Kunci: Chatbot, Informasi palsu (Hoax), *Dual Intent and Entity Transformer (DIET) Classifier* dan *Fuzzy String Matching*, *Search Engine*

**IMPLEMENTATION OF NATURAL LANGUAGE UNDERSTANDING IN
THE DEVELOPMENT OF A CHATBOT ABOUT FAKE INFORMATION
(HOAX) USING DUAL INTENT AND ENTITY TRANSFORMER (DIET)
CLASSIFIERS AND FUZZY STRING MATCHING**

ABSTRACT

Fake information (hoaxes) is information that does not really happen and has a negative impact that can distort the interpretation, perception, and assumption of the reader. The spread of false information (hoaxes) in Indonesia continues to increase tremendously. This condition can be said to be in an alarming condition. If this condition is not handled quickly and appropriately, it will certainly increase the high public misunderstanding of an event. Therefore, an approach is needed to explain the truth of information. This research aims to develop a question and answer chatbot system about false information (hoax) by applying the Dual Intent and Entity Transformer (DIET) Classifier and Fuzzy String Matching methods. This research uses question and answer data from the website <https://www.mafindo.or.id/> and Mafindo's youtube channel. This research uses search engines that can find information from the websites <http://turnbackhoax.id/>, <https://www.liputan6.com>, <https://cekfakta.tempo.co>, and <https://www.kompas.com/>. Tests conducted using the Dual Intent and Entity Transformer (DIET) Classifier method approach resulted in an accuracy of 0.9.

Keyword: Chatbot, Fake information (hoaxes), Dual Intent and Entity Transformer (DIET) Classifier dan Fuzzy String Matching, Search Engine

DAFTAR ISI

BAB I	PENDAHULUAN	1
	1.1. Latar Belakang	1
	1.2. Rumusan Masalah	3
	1.3. Tujuan Penelitian	3
	1.4. Batasan Masalah	3
	1.5. Manfaat Penelitian	3
	1.6. Sistematika Penulisan	4
BAB II	LANDASAN TEORI	5
	2.1. Chatbot	5
	2.2. NLP (<i>Natural Language Processing</i>)	5
	2.3. NLU (<i>Natural Language Understanding</i>)	6
	2.4. <i>Intent Classification</i>	7
	2.5. <i>Fuzzy string matching</i>	15
	2.6. <i>CountVectorsFeaturizer</i>	16
	2.7. Penelitian Terdahulu	17
BAB III	ANALISIS DAN PERANCANGAN SISTEM	20
	3.1. Blok Diagram Keseluruhan sistem	20
	3.2. Data	23
	3.3. Arsitektur Umum	28
	3.3.1. Split Data	30
	3.3.2. <i>Preprocessing</i>	30
	3.3.2.1. <i>Tokenizer</i>	30
	3.3.2.2. <i>CountVectorsFeaturizer</i>	31
	3.3.3. <i>Model Building</i>	33
	3.3.4. <i>Testing Model</i>	33
	3.3.5. Implementasi <i>Fuzzy String Matching</i>	34
	3.3.6. Implementasi <i>Search Engine</i>	35
	3.3.7. Dialog Management	36

3.3.8. Respon	36
3.4. Diagram Alur Sistem Chatbot	37
3.5. Perancangan Antarmuka Aplikasi	38
3.5.1. Rancangan Tampilan Login pada Web Admin	38
3.5.2. Rancangan Tampilan Dashboard pada Web Admin	38
3.5.3. Rancangan Tampilan List Data FAQ pada Web Admin	39
3.5.4. Rancangan Tampilan Create pada Web Admin	40
3.5.5. Rancangan Tampilan Edit pada Web Admin	41
3.5.6. Rancangan Tampilan Profile pada Sistem Chatbot	42
3.5.7. Rancangan Tampilan Halaman Chat pada Sistem Chatbot	43
3.6. Evaluasi Model	43
 BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM	 45
4.1. Implementasi Sistem	45
4.1.1. Spesifikasi Hardware dan Software	45
4.1.2. Implementasi Perancangan Antarmuka	46
4.2. Implementasi Model	50
4.3. Evaluasi Model	51
4.4. Pengujian <i>Fuzzy String Matching</i>	53
4.5. Pengujian sistem	57
 BAB V KESIMPULAN DAN SARAN	 64
5.1. Kesimpulan	64
5.2. Saran	64
 DAFTAR PUSTAKA	 65

DAFTAR TABEL

Tabel 2.1. Penelitian Terdahulu	18
Tabel 3.1. Contoh Data API	24
Tabel 3.2. Contoh Penerapan <i>Tokenizer</i>	31
Tabel 3.3. Contoh Penerapan <i>CountVectorsFeaturizer</i>	31
Tabel 4.1. Hasil Pengujian Pencocokan <i>String</i>	55
Tabel 4.2. Hasil Pengujian Sistem	57

DAFTAR GAMBAR

Gambar 2.1. Proses Token Kata “apa”	8
Gambar 2.2. Arsitektur Proses Token Kata “apa”, “itu”, hoaks”	9
Gambar 2.3. Proses __CLS__	9
Gambar 2.4. Arsitektur Proses Token Kata “__CLS__”	10
Gambar 2.5. Arsitektur Proses Transformer (2 layers)	11
Gambar 2.6. Tahap Similarity Mask Language Model	12
Gambar 2.7 Tahap Intent Classification	13
Gambar 2.8. Tahap Entity Recognition	13
Gambar 2.9. Tahap Total Loss	14
Gambar 3.1. Diagram Blok Sistem	22
Gambar 3.2. Arsitektur Umum	29
Gambar 3.3. Diagram Alur Sistem Chatbot	37
Gambar 3.4. Rancangan Tampilan Login pada Web Admin	38
Gambar 3.5. Rancangan Tampilan Dashboard pada Web Admin	39
Gambar 3.6. Rancangan Tampilan List Data FAQ pada Web Admin	39
Gambar 3.7. Rancangan Tampilan Create pada Web Admin	40
Gambar 3.8. Rancangan Tampilan Edit pada Web Admin	41
Gambar 3.9. Rancangan Tampilan Profile Sistem Chatbot	42
Gambar 3.10. Rancangan Tampilan Halaman Chat pada Sistem Chatbot	43
Gambar 4.1. Tampilan Halaman Login pada Web Admin	46
Gambar 4.2. Tampilan Halaman Dashboard pada Web Admin	47
Gambar 4.3. Tampilan Halaman List Data FAQ pada Web Admin	47

Gambar 4.4. Tampilan Halaman Create Data API pada Web Admin	48
Gambar 4.5. Tampilan Halaman Edit Data API pada Web Admin	48
Gambar 4.6. Tampilan Halaman Profile pada Sistem Chatbot	49
Gambar 4.7. Tampilan Halaman Chat pada Sistem Chatbot	50
Gambar 4.8. Hasil Proses Training	50
Gambar 4.9. Confusion Matrix Akurasi Dari Data Uji	51

BAB I

PENDAHULUAN

1.1. Latar Belakang

Kemajuan teknologi informasi tumbuh begitu sangat cepat dengan hadirnya berbagai media sosial dan media daring sebagai sarana pemberi informasi. Kecepatan dan jumlah beredarnya informasi melalui kedua media tersebut telah mampu mengalahkan kecepatan dan jumlah media cetak. Namun, informasi yang tersebar di media sosial dan media daring tersebut bukan hanya memiliki nilai kualitas informasi yang positif akan tetapi juga yang negatif seperti informasi palsu (hoax).

Informasi palsu (hoax) merupakan informasi yang tidak benar-benar terjadi dan memiliki dampak negatif yang mampu mendistorsi interpretasi, persepsi, dan anggapan pembaca. Informasi palsu (hoax) ini tentunya dapat mempengaruhi pembaca dengan mencemarkan citra dan kredibilitas. Informasi palsu (hoax) bertujuan untuk mengelabui pembaca sehingga pembaca mempercayai informasi palsu (hoax) tersebut benar-benar terjadi. (Rahadi, 2017)

Penyebaran informasi palsu (hoax) saat ini di Indonesia terus mengalami peningkatan yang sangat luar biasa. Kondisi ini dapat dikatakan berada di kondisi yang mengkhawatirkan. Apabila kondisi ini tidak ditangani dengan cepat dan tepat, maka tentu akan meningkatkan kesalahpahaman masyarakat yang tinggi akan suatu peristiwa. Menurut hasil survey (MASTEL, 2019), pada survey 2017 menunjukkan bahwa dari 1.146 responden, ada sebanyak 44,3% dari seluruh responden menerima informasi palsu (hoax) setiap hari dan ada sebanyak 17,2% dari seluruh responden menerima informasi palsu (hoax) lebih dari satu kali dalam sehari. Sedangkan pada survey 2019 mengungkapkan bahwa dari 941 responden, ada sebanyak 34,6% dari seluruh responden menerima informasi palsu (hoax) setiap hari dan ada sebanyak 14,7% dari seluruh responden menerima informasi palsu (hoax) lebih dari satu kali dalam sehari.

Tampaknya ada penurunan frekuensi penyebaran informasi palsu (hoax) namun bentuk hoax semakin bervariasi dan samar. Media berita yang dianggap sebagai sumber terpercaya terkadang ikut tercemar oleh informasi palsu (hoax). Selain itu, penyebaran hoax juga banyak terjadi melalui sosial media. Informasi palsu (hoax) yang sering disebar terus-menerus melalui media sosial dapat membentuk opini publik bahwa informasi tersebut benar-benar terjadi. (Juditha, 2018)

Dalam penelitian ini, penulis akan membuat suatu pendekatan berupa chatbot untuk menjelaskan kebenaran sebuah informasi. *Chatbot* tersebut menggunakan DIET Classifier dalam mencocokkan intent dan menggunakan *fuzzy string matching* pada saat pencarian pola pertanyaan yang diajukan untuk menampilkan response yang tepat.

Penelitian terdahulu yang berkaitan dengan chatbot dalam menangani informasi virus corona pernah dilakukan oleh (Gunson et al., 2021). Penelitian ini membuat sistem AI percakapan yang dapat menangani informasi yang salah menggunakan pendekatan dua arah, yaitu dari teks ke informasi ringkas dan dari mitos yang beredar mengenai virus corona, pengujian menggunakan Natural Language Understanding yang F1-score mencapai sebesar 0,906

Penerapan algoritma *Dual Intent and Entity Transformer* (DIET) Classifier juga diterapkan pada penelitian terdahulu. Penelitian tersebut dilakukan oleh (Astuti et al., 2021). Di penelitian ini bertujuan dalam menciptakan *chatbot* sebagai percakapan antara mesin dan manusia. *Chatbot* yang dikembangkan berfungsi untuk memprediksi jawaban untuk pertanyaan tentang COVID-19 dengan menggunakan kerangka kerja pipeline DIET Classifier untuk 300 data training. Hasil uji dengan model DIET Classifier memberikan persentase sekitar 85%.

Penelitian yang menjelaskan keunggulan NLU juga dilakukan oleh (Jiao, 2020). Penelitian ini mengembangkan chatbot menggunakan prinsip Rasa NLU yang kemudian mengintegrasikan Rasa NLU dengan metode *Neural Network* (NN) dengan implementasi berdasarkan sistem entitas yang didasarkan pada pengenalan intent. Hasil dari analisis 2 metode di penelitian ini memvalidasi bahwa Rasa NLU mengungguli NN dalam akurasi dengan akurasi Rasa NLU sebesar 100% sedangkan NN sebesar 95%, tetapi NN memiliki integritas yang lebih baik dalam mengklasifikasikan entitas dari kata-kata yang tersegmentasi.

Berdasarkan latar belakang yang ada, penelitian yang hendak dilakukan oleh penulis adalah berjudul **PENERAPAN NATURAL LANGUAGE UNDERSTANDING PADA SISTEM CHATBOT INFORMASI PALSU (HOAX) MENGGUNAKAN DUAL INTENT AND ENTITY TRANSFORMER (DIET) CLASSIFIER DAN FUZZY STRING MATCHING**

1.2. Rumusan Masalah

Informasi palsu (hoax) merupakan informasi yang tidak benar-benar terjadi dan memiliki dampak negatif bagi masyarakat karena dapat menimbulkan kesalahpahaman akan suatu peristiwa yang terjadi. Diperlukan suatu pendekatan untuk menjelaskan kebenaran sebuah informasi.

1.3. Tujuan Penelitian

Penelitian ini bertujuan untuk mengembangkan sistem chatbot tanya jawab seputar informasi palsu (hoax) dan untuk menjelaskan kebenaran sebuah informasi dengan menerapkan metode *Dual Intent and Entity Transformer* (DIET) Classifier dan Fuzzy String Matching.

1.4. Batasan Masalah

Penelitian ini memiliki beberapa batasan masalah, sebagai berikut:

- Data yang dikumpulkan berupa data tanya jawab informasi palsu (hoax) yang berasal dari website <https://www.mafindo.or.id/> dan channel youtube Mafindo.
- *Search engine* yang dapat mencari informasi yang berasal dari website <https://www.liputan6.com>, <https://cekfakta.tempo.co>, <http://turnbackhoax.id/>, dan <https://www.kompas.com/>
- Bot hanya dapat berinteraksi menggunakan bahasa Indonesia.
- Bot dibuat untuk menjawab pertanyaan dari *user* mengenai informasi palsu (hoax) dan untuk mencari kebenaran suatu informasi yang sudah diklasifikasikan oleh pihak Mafindo, liputan6, Kompas, dan cekfakta.

1.5. Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut:

- Meminimalisir penyebaran informasi palsu (hoax) dari berbagai media penyebar informasi.
- Memberikan kemudahan kepada masyarakat dalam mencari informasi sehingga terhindar dari hal-hal negatif.
- Dapat menghemat waktu dalam melakukan tanya jawab seputar informasi palsu (hoax).

1.6. Sistematika Penulisan

Sistematika penulisan dari skripsi ini terdiri dari lima bagian, meliputi:

Bab I: Pendahuluan

Pada bab pendahuluan berisi latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, dan sistematika penulisan.

Bab II: Landasan Teori

Pada bab landasan teori berisi penjelasan teori mengenai pemrosesan bahasa alami menggunakan *intent classification*, dan *Fuzzy String Matching* dalam pembuatan *chatbot*.

Bab III: Metodologi Penelitian

Pada bab ini berisi penjelasan mengenai analisis dan penerapan metode *Dual Intent and Entity Transformer* (DIET) dan *Fuzzy String Matching* untuk pembuatan *chatbot*. Bab ini juga membahas arsitektur umum, dan perancangan antarmuka sistem.

Bab IV: Implementasi dan Pengujian Sistem

Pada bab implementasi dan pengujian sistem berisi pembahasan mengenai implementasi dan perancangan dari hasil metodologi penelitian yang sebelumnya telah disusun pada Bab III dan hasil yang diperoleh dari tahap pengujian akan dijelaskan pada bab ini.

Bab V: Kesimpulan dan Saran

Pada bab kesimpulan dan saran berisi tentang kesimpulan dari seluruh pembahasan bab yang telah dilakukan pada penelitian ini, bab ini juga memberikan saran dari penulis untuk perbaikan di penelitian terkait selanjutnya.

BAB II

LANDASAN TEORI

2.1. Chatbot

Chatbot merupakan suatu aplikasi sistem cerdas yang menerapkan Natural Language Processing dalam prosesnya dan dapat mensimulasikan percakapan layaknya manusia melalui teks ataupun interaksi suara. *Chatbot* adalah robot virtual berbasis *artificial intelligence* yang dapat menjawab input yang diberikan, input ini berupa pertanyaan. *Bot* akan melakukan pemindaan kata kunci yang ada pada input tersebut. Kemudian, *bot* akan memberikan respon jawaban berdasarkan kata kunci yang paling sesuai dari data yang tersedia sesuai dengan pertanyaan dari *user*.

Penerapan chatbot biasanya diarahkan untuk menjadi asisten dari *user*, misalnya *chatbot* yang digunakan sebagai layanan *customer service* selama 24 jam. Selain itu, banyak perusahaan dan organisasi yang menggunakan chatbot untuk berkomunikasi dengan user mereka (Ngai et al., 2021). Chatbot juga dapat digunakan sebagai pendukung layanan *Frequently Ask Question* (FAQ). Dalam penelitian ini, chatbot akan digunakan sebagai sarana rekomendasi informasi palsu (hoax) dan untuk menjawab pertanyaan *user*.

2.2. NLP (Natural Language Processing)

Natural Language Processing (NLP) adalah bidang kecerdasan buatan yang bertujuan dalam memberikan komputer kemampuan untuk dapat memahami teks ataupun kata-kata yang diucapkan manusia. NLP digunakan untuk menganalisa, memahami, dan menghasilkan bahasa manusia dengan cara yang dapat diproses dan dimengerti oleh komputer.

Dengan menggunakan NLP, komputer dapat memproses input bahasa manusia dalam bentuk teks ataupun suara dalam memahami makna dari input tersebut. (Pustejovsky & Stubbs, 2013) menjelaskan ada beberapa area utama dalam penelitian NLP, yaitu:

- **Question Answering System** merupakan kemampuan komputer dalam memberi respon pada pertanyaan yang diajukan oleh user. Tanpa mengetikkan kata kunci pada mesin pencarian, user dapat mengajukan pertanyaan dalam bahasa alami yang mereka gunakan.
- **Summarization** merupakan pembuatan rangkuman pada beberapa kumpulan berkas dokumen yang memungkinkan user untuk mengubah teks dari dokumen yang berukuran besar menjadi ringkasan teks yang berukuran kecil.
- **Machine Translation** merupakan aplikasi yang dapat mengartikan bahasa yang satu ke bahasa yang lain, contoh aplikasi tersebut yaitu Google Terjemahan.
- **Speech Recognition** merupakan suatu area NLP dalam pembuatan model yang digunakan oleh mesin untuk pengenalan bahasa manusia
- **Document Classification** merupakan bidang NLP yang bertujuan dalam menentukan tempat yang paling baik untuk memasukkan dokumen yang baru ke dalam sistem.

2.3. NLU (Natural Language Understanding)

Natural Language Understanding (NLU) adalah sub bidang dari NLP yang berfokus pada machine reading comprehension melalui informasi dari suatu pesan yang memungkinkan untuk menentukan makna yang dimaksud dari suatu kalimat. NLU pada dialog sistem memiliki tugas sebagai asisten pribadi dalam memahami masukan *user* untuk membuat interaksi manusia dengan mesin menjadi lebih efektif (Zheng et al., 2020). NLU menggunakan Teknik pembelajaran mesin dan NLP untuk mengekstrak struktur informasi (intent dari pertanyaan user dan entity terkait) dari *user input* (Abdellatif et al., 2021). Dalam segi penerapan NLU, terdapat beberapa NLU yang dapat digunakan secara luas yang mudah diintegrasikan dengan aplikasi pihak ketiga yaitu:

- **IBM Watson** : NLU yang dikembangkan oleh IBM dan memiliki model prebuild untuk domain yang berbeda (misalnya perbankan) dan editor dialog visual untuk menyederhanakan pembuatan dialog bagi orang non-programmer.

- **Dialogflow:** NLU yang dikembangkan oleh Google dan mendukung lebih dari 20 bahasa serta dapat diintegrasikan ke banyak platform chatting seperti slack.
- **Rasa:** NLU open-source yang dimiliki oleh Rasa Teknologi. Rasa mengizinkan pihak pengembang untuk menkonfigurasi, deploy, dan menjalankan NLU pada server local. Serta dapat meningkatkan kecepatan pemrosesan dengan menghemat waktu jaringan dibandingkan dengan platform berbasis cloud.
- **Language Understanding Intelligent Service (LUIS):** NLU yang dikembangkan oleh Microsoft dan mendukung 5 bahasa pemrograman yaitu C#, Go, Java, NodeJS, dan Python.

2.4. Intent Classification

Tahap selanjutnya dari NLP dalam pengembangan chatbot adalah intent classification yang membantu dalam mengidentifikasi dan mendeteksi makna dari pesan *user*. (Rustamov et al., 2021). Untuk *intent classification* pada penelitian ini menggunakan *DIET Classifier*.

Dual Intent and Entity Transformer (DIET) adalah metode yang digunakan untuk klasifikasi intent yang dalam menyesuaikan dengan jawaban yang akan diberikan. Keuntungan utama dari arsitektur DIET adalah kemampuan pemrosesan kata, dan pre-trained word embedding dalam waktu yang sama. Masalah *intent classification* dan *entity recognition* dapat diatasi menggunakan DIET Classifier. Arsitektur ini memiliki 3 component: intent classification, named entity recognition, dan mask language model.

Penjelasan cara kerja algoritma *Dual Intent and Entity Transformer* (DIET) pada contoh kalimat “apa itu hoaks”.

Proses algoritma dimulai dari Kata “apa”, “itu”, “hoaks” yang sudah ditokenizer dan sudah dibuat kedalam karakter vector akan proses ke dalam 2 tahap berbeda yaitu:

1. *Sparse Features* dan *Feed-Forward*

Pada tahap *sparse Features* akan dilakukan pemisahan perhuruf menjadi karakter endgram lalu pada tahap *feed-forward* dilakukan penggabungan pada karakter endgram dan disatukan dengan rumus $= g(W_x + b)$.

$$\text{rumus} = g(W_x + b) \quad (2.1.)$$

Keterangan:

g = fungsi transfer pada neural network

W = nilai weight pada vector

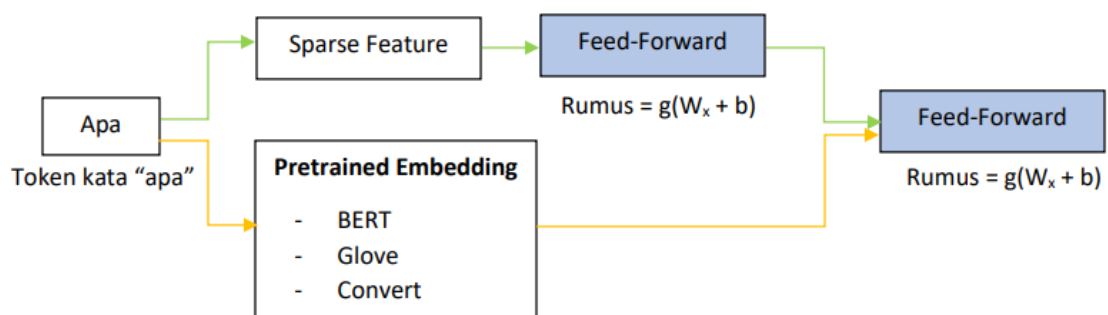
x = parameter

b = nilai bias pada vector

2. Pretrained Embedding

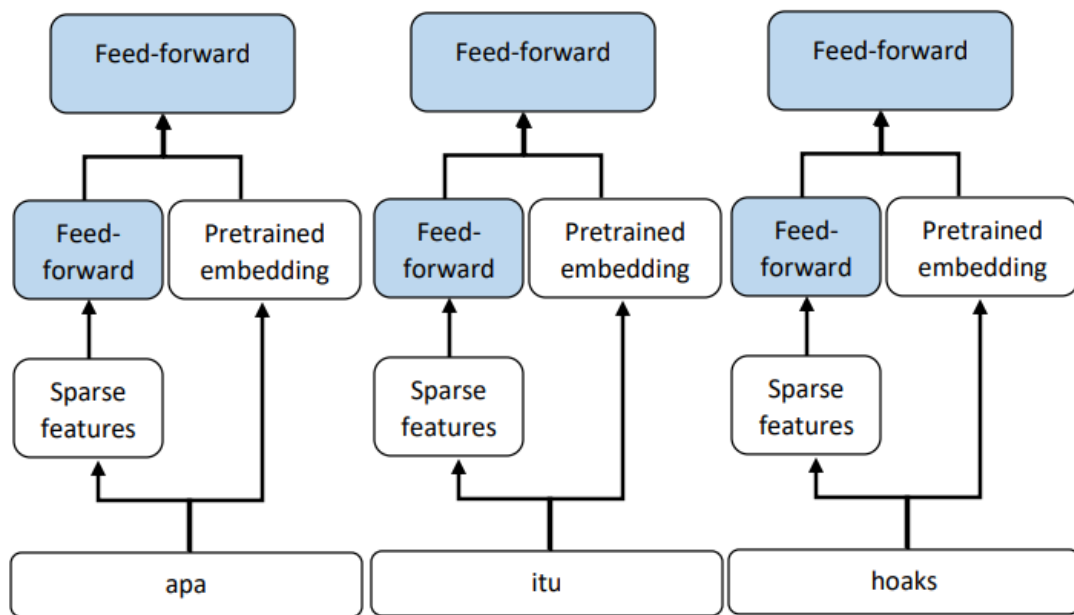
Bertujuan untuk merepresentasi kata yang telah dihitung sebelumnya dan dilatih pada korpora teks besar. Lalu menangkap informasi semantik dan sintaksis tentang kata-kata dan yang sering digunakan sebagai fitur dalam tugas pemrosesan bahasa alami (NLP). Pretrained word embeddings biasanya berasal dari metode unsupervised learning, seperti Word2Vec, GloVe, atau fastText. Model-model ini bertujuan untuk memetakan kata-kata ke dalam representasi vektor berdasarkan pola kemunculan dalam data latih. Pada proses ini kata “apa” akan diproses kedalam *Pretrained Embedding* yang menggunakan *BERT* (Devlin et al., 2019), *Convert* (Henderson et al., 2020), atau *Glove* (Pennington et al., 2014).

Proses Selanjutnya, ada penggabungan 2 tahap berbeda kedalam bentuk Feed-Forward menggunakan rumus $= g(W_x + b)$. Proses dapat dilihat pada Gambar 2.1.



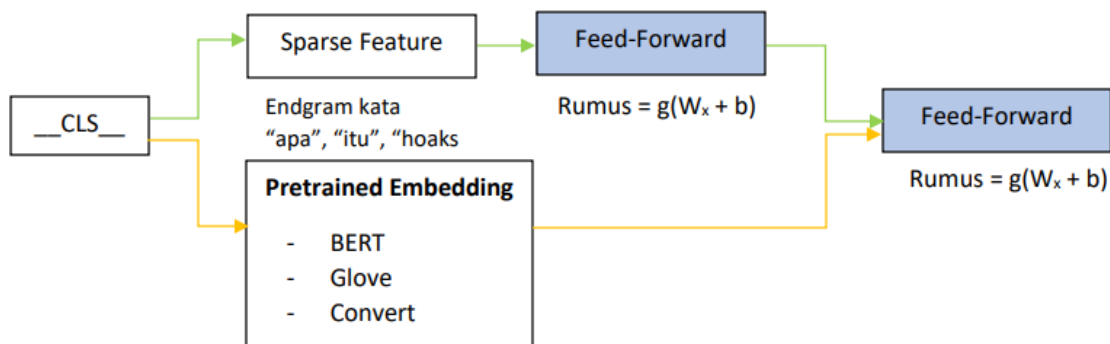
Gambar 2.1. Proses Token Kata “apa”

Sehingga arsitektur proses keseluruhan dari kalimat “apa itu hoaks” dapat dilihat pada Gambar 2.2.



Gambar 2.2. Arsitektur Proses Token Kata “apa”, “itu”, “hoaks”

Adapun proses untuk merespon kalimat “apa itu hoaks” dapat dilihat pada Gambar 2.3.



Gambar 2.3. Proses `__CLS__`

Tahap ini merupakan proses utter (respon) dalam kalimat “apa itu hoaks” yang sudah ditokenizer dan sudah dibuat kedalam karakter vector akan proses ke dalam 2 tahap berbeda yaitu:

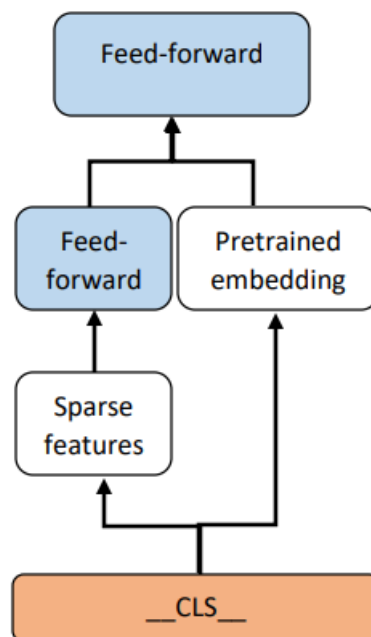
1. *Sparse Features* dan *Feed-Forward*

Pada tahap *sparse Features* akan dilakukan pemisahan perhuruf menjadi karakter endgram lalu pada tahap *feed-forward* dilakukan penggabungan pada karakter endgram dan disatukan dengan rumus $= g(Wx + b)$.

2. Pretrained Embedding

Pada proses ini kata “apa” akan diproses kedalam *Pretrained Embedding* yang menggunakan *BERT*, *Convert*, atau *Glove*.

Sehingga arsitektur proses ini dapat dilihat pada Gambar 2.4.



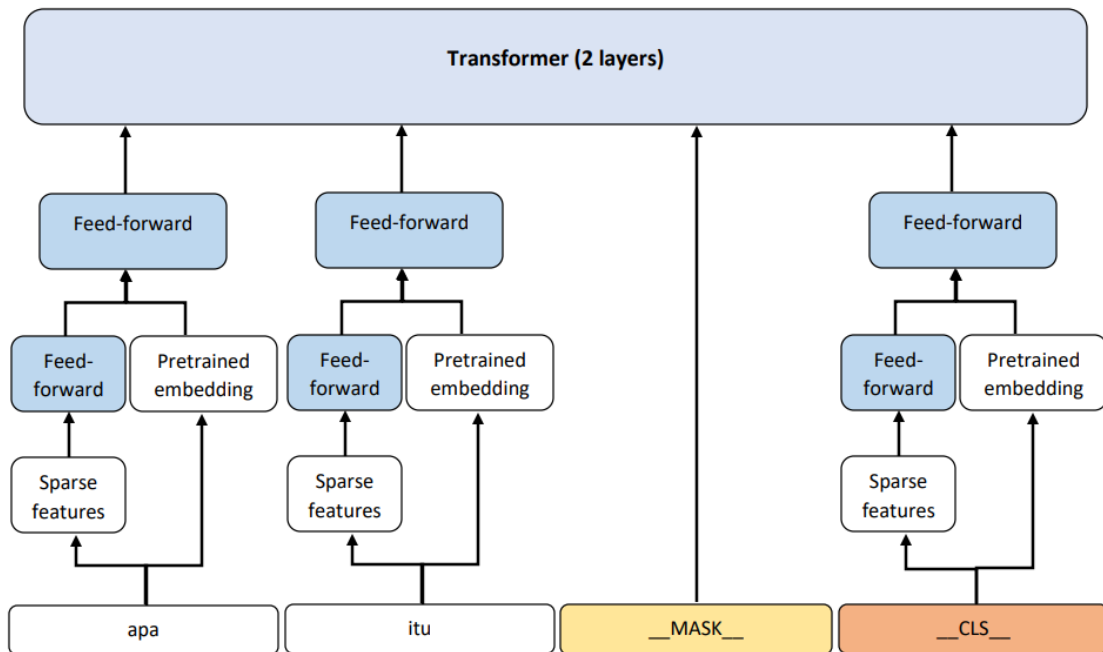
Gambar 2.4. Arsitektur Proses Token Kata “___CLS__”

Pada saat algoritma *Dual Intent and Entity Transformer* (DIET) memproses kalimat dari user, algoritma tersebut menghilangkan 1 kata random dan mengubahnya kedalam bentuk “___MASK__” yang mana tujuan nya untuk memprediksi kata yang dihilangkan tersebut pada respon ataupun jawaban untuk kalimat dari *user*.

Pada contoh kalimat “apa itu hoaks” kata “hoaks” akan dihilangkan dan diubah kedalam bentuk “___MASK__” dan akan dilanjut pada tahap “Transformer (2 layers)”

Pada hasil dari proses token “apa”, “itu”, “___MASK__”, “___CLS__” akan dilanjut ke tahap transformer (2 layers)

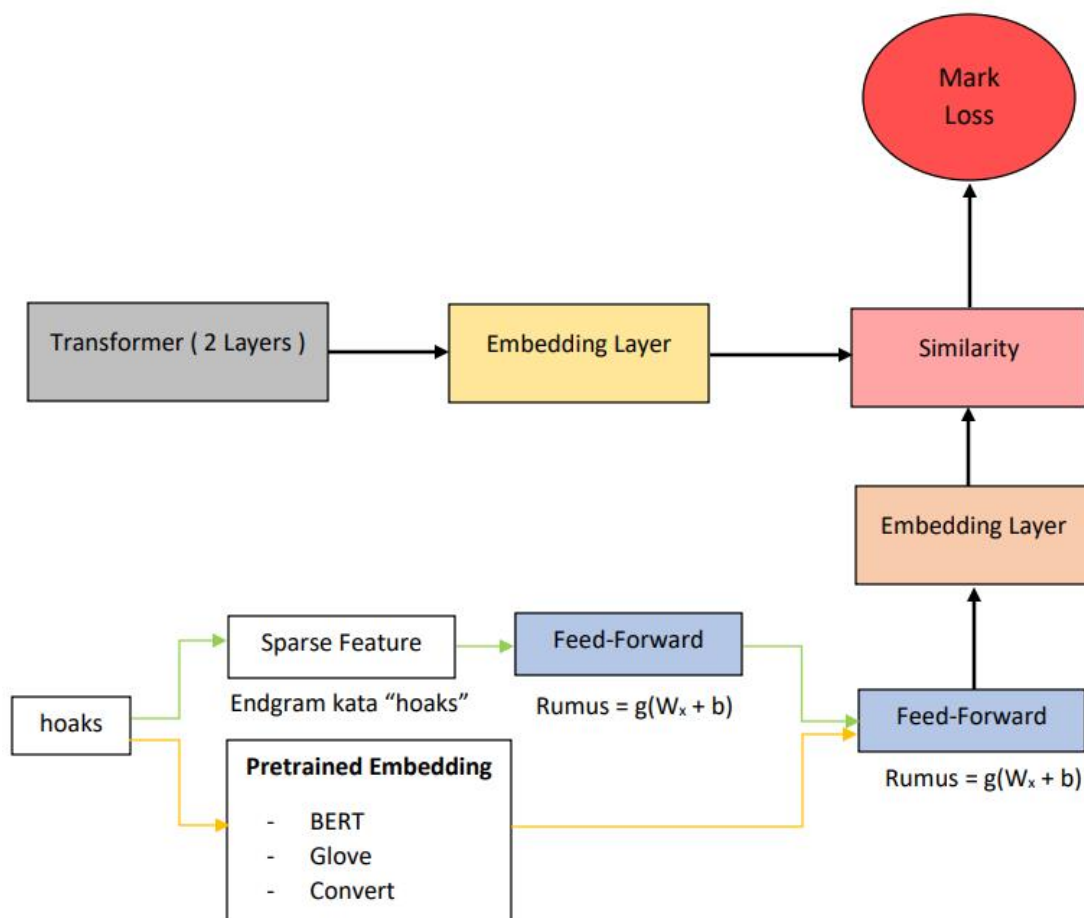
Sehingga arsitektur proses dapat dilihat pada Gambar 2.5.



Gambar 2.5. Arsitektur Proses *Transformer* (2 layers)

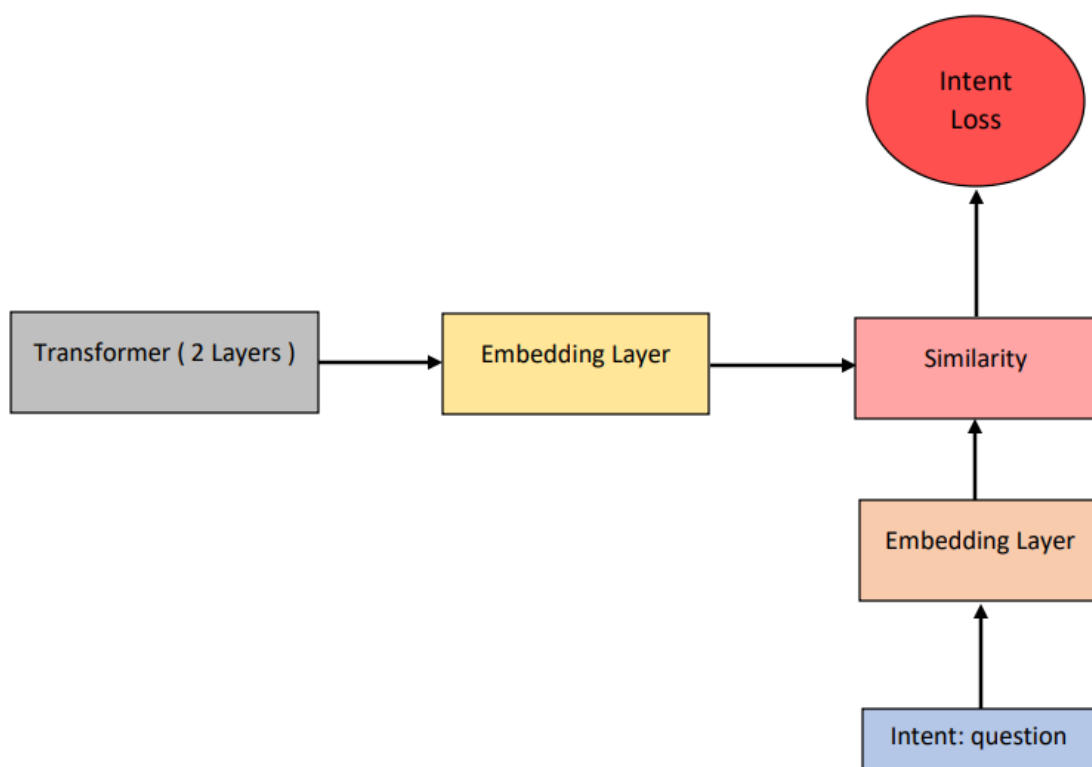
Setelah token “apa”, “itu”, “__MASK__” , “__CLS__” diproses pada tahap *transformer* (2 layers), hasil dari tahap *transformer* (2 layers) akan dilanjut pada 3 tahap similarity yang berbeda yaitu:

1. tahap similarity *mask language model*, pada contoh arsitektur ini menggunakan token “hoaks” yaitu dimana token “hoaks” akan melalui tahap embedding layer dan akan dilakukan pencocokan pada tahap *similarity* dari hasil embedding layer pada tahap *transformer* (2 layers) dan apabila tidak memiliki kemiripan maka akan menghasilkan “*mark loss*” Sehingga proses tahap ini dapat dilihat pada Gambar 2.6.



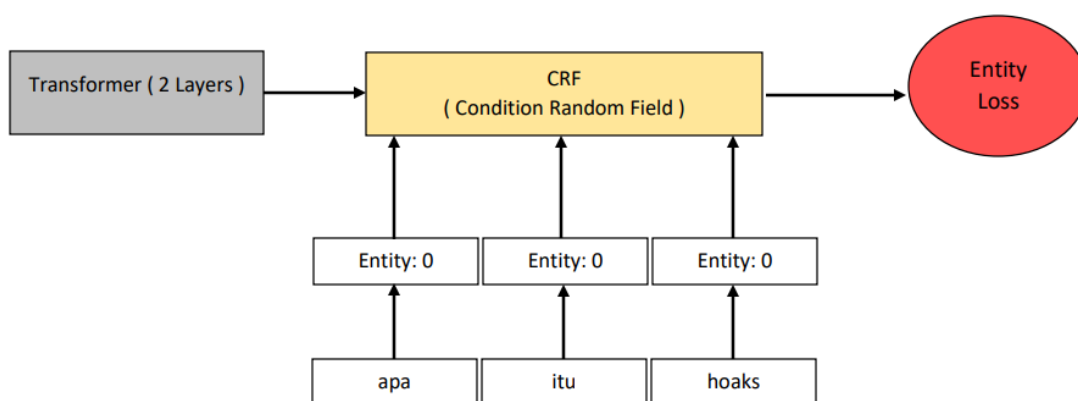
Gambar 2.6. Tahap *Similarity Mask Language Model*

2. tahap *similarity* dengan “*intent: question*”, disini intent dianggap sebagai label dan dilakukan proses pencocokan dengan token yang ada pada *transformer (2 layers)* apabila tidak ada kemiripan maka akan menghasilkan “*intent loss*”. Sehingga proses tahap ini dapat dilihat pada Gambar 2.7.



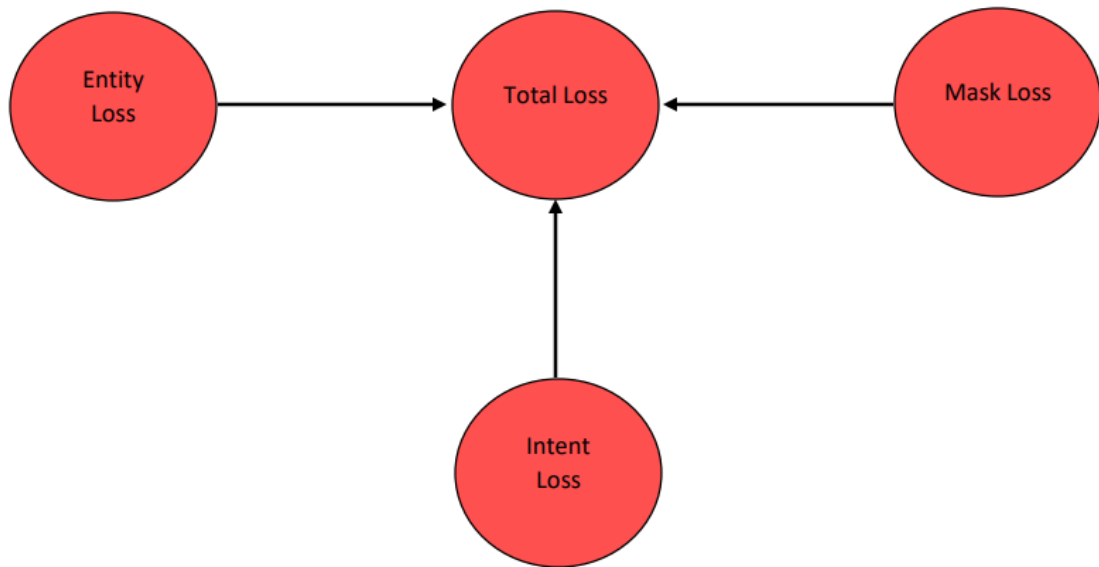
Gambar 2.7. Tahap *Intent Classification*

3. tahap similarity menggunakan *entity* diproses pada *CRF* (*Condition Random Field*) yang mana apabila tidak memiliki kemiripan maka akan menghasilkan “*entity loss*”. Sehingga proses tahap ini dapat dilihat pada Gambar 2.8.



Gambar 2.8. Tahap *Entity Recognition*

Hasil dari *entity loss*, *total loss*, dan *mask loss* akan disatukan menjadi *total loss*. Sehingga proses tahap ini dapat dilihat pada Gambar 2.9.



Gambar 2.9. Tahap *Total Loss*

Sehingga di dapat dirumuskan menjadi:

$$L_{\text{total}} = L_I + L_E + L_M \quad (2.2)$$

Keterangan:

L_I = intent classification loss

L_E = entity recognition loss

L_M = masked language loss

2.5. Fuzzy String Matching

Fuzzy string matching merupakan metode yang digunakan untuk mencocokkan *string* yang mirip, tapi tidak sama persis. *String* yang disamakan memiliki kemiripan dalam penyusunan karakter yang dimiliki *string* dan mempunyai kemungkinan perbedaan dari segi urutan ataupun jumlah karakter, *string* akan tetap memiliki kemiripan baik secara penulisan ataupun secara ucapan. Nilai tersebut dapat berupa *'true'* atau *'false'* pada penggunaan metode *fuzzy string matching*. Nilai *'true'* atau *'false'* pada metode *fuzzy string matching* berkaitan dengan kedudukan yang dimiliki. Metode *fuzzy string matching* memiliki kedudukan nilai range antara 0 hingga 1. Metode *fuzzy string matching* digunakan untuk menyatakan nilai dari terjemahan linguistik.

Metode *fuzzy string matching* dapat digunakan pada saat melakukan proses mencari kata yang dikerjakan *chatbot*, *fuzzy string matching* akan melakukan pengolahan data input untuk menghasilkan output yang *valid*.

Fuzzy String Matching dapat menggunakan penyusunan karakter *string* dalam menentukan kemiripan *string* yang ingin ditemukan pada *string* yang ada di dalam data walaupun *string* tersebut tidak begitu sama persis dengan *string* yang ada pada data. *Similarity function* digunakan untuk menentukan kemiripan dari suatu *string* pada saat melakukan pencarian pola pada *string* dan akan didapatkan kemiripan pola pada *string* dengan pendekatan aproksimasi. *Fuzzy String Matching* yang beroperasi menggunakan pendekatan aproksimasi dalam mencari pola kemiripan biasanya menggunakan banyak sumber daya komputasi. (Sebastian & George, 2016).

Beberapa Jenis dari *Fuzzy String Matching*, yaitu *Levenshtein Distance*, *Hamming*, *Sellers*, dan *Damerau-Levenshtein*,. Algoritma *Fuzzy String Matching* ini melakukan pencarian *string* dengan cara yang berbeda.

Pada algoritma *Levenshtein Distance* menggunakan proses dalam menghitung ukuran berdasarkan jumlah kemiripan antara *string* dengan *string* lainnya. Algoritma ini sering disebut dengan algoritma *Edit Distance*. Hal ini disebabkan karena algoritma ini menghitung jumlah pada operasi penukaran, penghapusan, dan penyisipan dalam *string* sehingga sama dengan *string* lain yang menjadi pembanding. Sebagai contoh dari algoritma *Edit Distance*, *string* "computer" dengan "komputer" memiliki distance 1 sebab hanya memerlukan pengerjaan proses pengubahan satu *string* ke *string* yang lain.

Pada string “computer” dengan “komputer” melakukan proses pertukaran karakter “c” ke karakter “k”.

2.6. CountVectorsFeaturizer

CountVectorsFeaturizer adalah salah satu komponen penting dalam library Rasa NLU (*Natural Language Understanding*). Komponen ini digunakan untuk mengubah data teks menjadi fitur numerik yang dapat digunakan oleh model machine learning dalam memprediksi intent atau tindakan yang diinginkan oleh *user*.

CountVectorsFeaturizer bekerja dengan cara menghitung frekuensi kemunculan setiap kata dalam dokumen dan menghasilkan representasi vektor untuk setiap kata. Representasi vektor ini kemudian digunakan sebagai fitur dalam model NLU (*Natural Language Understanding*) untuk melakukan klasifikasi intent.

CountVectorsFeaturizer dapat disesuaikan dengan beberapa parameter, termasuk jumlah kata yang akan diambil sebagai fitur, tipe tokenisasi, dan penanganan kata-kata yang tidak dikenal dalam data teks. Komponen ini juga dapat digunakan dengan teknik regularisasi, seperti L1 atau L2, untuk mencegah overfitting pada model NLU (*Natural Language Understanding*).

CountVectorsFeaturizer memiliki beberapa keuntungan dalam pengembangan chatbot menggunakan Rasa NLU (*Natural Language Understanding*), di antaranya:

1. Meningkatkan akurasi model NLU (*Natural Language Understanding*): Dengan menggunakan fitur numerik yang dihasilkan oleh CountVectorsFeaturizer, model NLU dapat mempelajari pola-pola yang ada pada data teks dengan lebih baik, sehingga meningkatkan akurasi model.
2. Mengurangi dimensi fitur: CountVectorsFeaturizer menghasilkan fitur numerik yang lebih efisien dalam memori dibandingkan dengan representasi teks asli, karena hanya mempertahankan kata-kata penting dalam data teks.
3. Mempercepat waktu training: Dengan menggunakan CountVectorsFeaturizer, waktu training model NLU (*Natural Language Understanding*) dapat dipercepat, karena komponen ini mengurangi jumlah fitur yang harus dipelajari oleh model.

2.7. Penelitian Terdahulu

Penelitian terdahulu yang berkaitan dengan chatbot dalam menangani informasi virus corona pernah dilakukan oleh (Gunson et al., 2021). Penelitian ini membuat sistem AI percakapan yang dapat menangani informasi yang salah menggunakan pendekatan dua arah, yaitu dari teks ke informasi ringkas dan dari mitos yang beredar mengenai virus corona, pengujian menggunakan *Natural Language Understanding* yang F1-score mencapai sebesar 0,906

Penerapan algoritma *Dual Intent and Entity Transformer (DIET) Classifier* juga diterapkan pada penelitian terdahulu. Penelitian tersebut dilakukan oleh (Astuti et al., 2021). Di penelitian ini bertujuan dalam menciptakan *chatbot* sebagai percakapan antara mesin dan manusia. *Chatbot* yang dikembangkan berfungsi untuk memprediksi jawaban untuk pertanyaan tentang COVID-19 dengan menggunakan kerangka kerja pipeline *Dual Intent and Entity Transformer (DIET) Classifier* untuk 300 data traning. Hasil uji dengan model *Dual Intent and Entity Transformer (DIET) Classifier* memberikan persentase sekitar 85%

Penelitian lain mengenai penerapan NLU (*Natural Language Understanding*) dilakukan oleh (Abro et al., 2022). Penelitian ini memperkenalkan kerangka NLU (*Natural Language Understanding*) dengan menggunakan model kesamaan argumen BERT dan BiLSTM untuk mengidentifikasi sistem argument yang dirujuk pengguna pada sistem dialog argumentative pada data ucapan pengguna untuk interaksi model pengklasifikasi tersebut mengungguli model *Dual Intent and Entity Transformer (DIET)*, DistilBERT dan BERT dengan akurasi BERT 93,2 dan BiLSTM 94,0

Penelitian lainnya mengenai penerapan *chatbot* tanya jawab dilakukan oleh (Muangkammuen et al., 2018). Penelitian ini membuat chatbot yang dapat memberikan tanggapan terhadap masalah laporan dari *customer* dan membalas solusi yang sama untuk permasalahan yang sama secara otomatis berdasarkan *Frequently Asked Questions* (FAQ) dengan huruf Thailand menggunakan LSTM dan menghasilkan akurasi sebesar 93,2%

Penelitian yang menjelaskan keunggulan NLU (*Natural Language Understanding*) juga dilakukan oleh (Jiao, 2020). Penelitian ini mengembangkan chatbot menggunakan prinsip Rasa NLU (*Natural Language Understanding*) yang kemudian

mengintegrasikan Rasa NLU (*Natural Language Understanding*) dengan metode *Neural Network* (NN) dengan implementasi berdasarkan sistem entitas yang didasarkan pada pengenalan intent. Hasil dari analisis 2 metode di penelitian ini memvalidasi bahwa Rasa NLU (*Natural Language Understanding*) mengungguli *Neural Network* (NN) dalam akurasi dengan akurasi Rasa NLU (*Natural Language Understanding*) sebesar 100% sedangkan *Neural Network* (NN) sebesar 95%, tetapi *Neural Network* (NN) memiliki integritas yang lebih baik dalam mengklasifikasikan *entitas* dari kata-kata yang tersegmentasi. Rangkuman dari penelitian terdahulu dapat dilihat pada Table 2.1

Tabel 2.1. Penelitian Terdahulu

No.	Penulis	Metode	Keterangan
1.	Abro et al., (2022)	BERT dan BiLSTM	Penelitian ini memperkenalkan kerangka NLU dengan menggunakan model kesamaan argumen BERT dan BiLSTM untuk mengidentifikasi sistem argument yang dirujuk pengguna pada sistem dialog argumentative pada data ucapan pengguna untuk interaksi model pengklasifikasi tersebut mengungguli model DIET, DistilBERT dan BERT dengan akurasi BERT 93,2 dan BiLSTM 94,0
2.	Astuti et al., (2021)	DIET Classifier	Pada penelitian ini chatbot yang dikembangkan berfungsi untuk memprediksi jawaban untuk pertanyaan tentang COVID-19 dengan menggunakan kerangka kerja pipeline DIET Classifier untuk 300 data training. Hasil uji dengan model DIET Classifier memberikan persentase sekitar 85%

Tabel 2.1. Penelitian Terdahulu (Lanjutan)

No.	Penulis	Metode	Keterangan
3.	Gunson et al., (2021)	Natural Language Understanding	Penelitian ini membuat sistem AI percakapan yang dapat menangani informasi yang salah menggunakan pendekatan dua arah, yaitu dari teks ke informasi ringkas dan dari mitos yang beredar mengenai virus corona, pengujian menggunakan <i>Natural Language Understanding</i> yang F1-score mencapai sebesar 0,906
4.	Muangkammue n et al., (2018)	Recurrent Neural Network-Long Short Term Memory	Penelitian ini membuat chatbot yang dapat memberikan tanggapan terhadap masalah laporan dari customer dan membalas solusi yang sama untuk permasalahan yang sama secara otomatis berdasarkan <i>Frequently Asked Questions</i> (FAQ) dengan huruf Thailand menggunakan LSTM dan menghasilkan akurasi sebesar 93,2%
5.	Jiao., (2020)	RASA NLU dan Neural Network	Penelitian ini mengembangkan chatbot menggunakan prinsip Rasa NLU yang kemudian mengintegrasikan Rasa NLU dengan metode <i>Neural Network</i> (NN) dengan implementasi berdasarkan sistem entitas yang didasarkan pada pengenalan <i>intent</i> . Hasil dari analisis 2 metode di penelitian ini memvalidasi bahwa Rasa NLU mengungguli NN dalam akurasi, tetapi NN memiliki integritas yang lebih baik dalam mengklasifikasikan <i>entitas</i> dari kata-kata yang tersegmentasi.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Blok Diagram Keseluruhan Sistem

Penelitian ini menggunakan Express JS dan Mongoose sebagai *Back-end Development*, React JS sebagai *Front-end Development*, MongoDB, dan Rasa Framework sebagai *Chatbot Development*.

React JS merupakan framework JavaScript yang digunakan pada saat membangun tampilan *User Interface* suatu website. React JS memiliki dua fitur yang menjadi keunggulan yaitu,

- JavaScript *Extension* (JSX) merupakan extension syntax javascript yang memungkinkan untuk memodifikasi *Document Object Model* (DOM) dengan kode bergaya HTML (*HyperText Markup Language*). DOM adalah *application programming interface* (API) yang berfungsi untuk mengatur struktur halaman web.
- virtual *Document Object Model* (DOM) merupakan Salinan dari DOM asli yang bertujuan untuk melihat bagian dari DOM asli yang berubah. Tanpa Virtual DOM, website akan menggunakan HTML untuk mengupdate DOM.

Express JS merupakan framework *website application* yang ditulis menggunakan bahasa pemrograman JavaScript. Express JS adalah framework *back-end* yang bertanggung jawab dalam mengatur fungsionalitas suatu website, seperti melakukan permintaan *Hypertext Transfer-Transfer Protocol* (HTTP), penanganan error, melakukan pengelolaan *routing* dan *session*, dan pertukaran data di server.

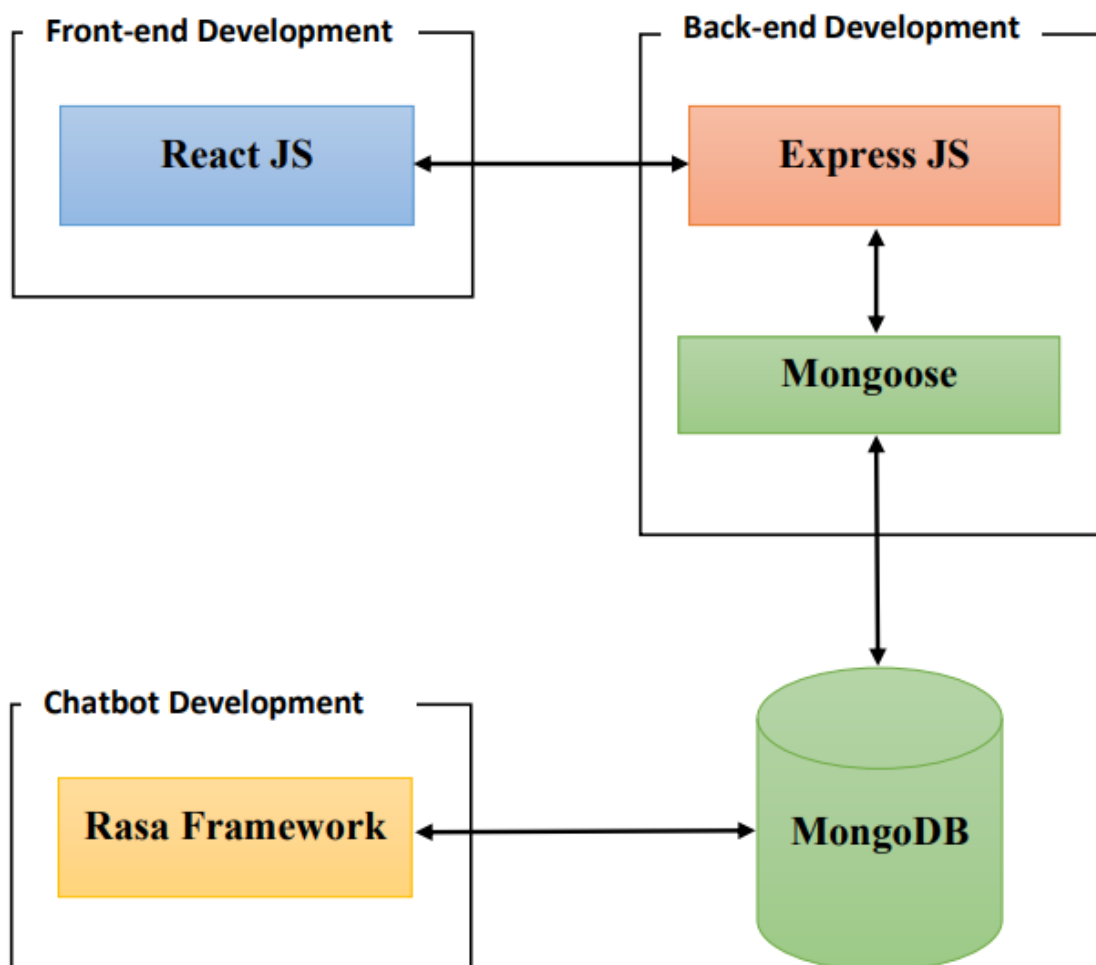
Mongoose adalah salah satu jenis database NoSQL (*Not Only SQL*) yang bersifat tanpa relasi (*non-relational*) dan merupakan sebuah *Object Document Mapper* (ODM) yang dapat mendefinisikan objek menggunakan skema dan dipetakan ke sebuah dokumen MongoDB. Mongoose menyediakan fungsionalitas yang berkaitan dengan pembuatan dan pengerjaan skema.

MongoDB merupakan sebuah database yang menyimpan data sebagai dokumen dalam struktur JSON (*JavaScript Object Notation*). Dokumen tersebut disimpan dalam *collection*. MongoDB memiliki fleksibilitas dalam masalah struktur pada database.

Rasa Framework adalah kerangka pembelajaran mesin sebagai percakapan cerdas berbasis teks yang dapat memahami masukan *user*, dan dapat mengadakan percakapan dengan *user*. Rasa Framework memiliki dua komponen utama yaitu:

- Rasa NLU merupakan *natural language processing tools* yang digunakan dalam mengklasifikasikan maksud (*intent classification*) dan ekstraksi entitas (*entity extraction*) dalam suatu percakapan, kemudian menggunakan *machine learning* ketika mengambil pola pertanyaan *user* dan menggeneralisasi kalimat yang berada diluar dari topik percakapan. (Astuti et al., 2021)
- Rasa Core adalah sebuah kerangka obrolan yang menangani percakapan kontekstual dan digunakan untuk mengelolah percakapan berbasis *machine learning*. (Bocklisch et al., 2017)

Penelitian ini membangun sistem yang terdiri dari *Front-end*, *Back-end*, MongoDB, dan *Chatbot*. Pada bagian *Front-end*, akan dibangun sebuah tampilan web admin menggunakan React JS dalam mengelola data melalui pengekseskuan *Hypertext Transfer–Transfer Protocol* (HTTP) dari web server pada bagian sistem *Back-end*. Sedangkan pada bagian *Back-end*, akan dibangun sebuah *web server* sebagai *Hypertext Transfer–Transfer Protocol* (HTTP) menggunakan Express JS dan menggunakan Mongoose yang berfungsi dalam mengelola data dalam mengirim data melalui *request* dan *web server* sebagai penerima data yang akan bertanggung jawab dalam memodifikasi database dan mengirimkan semua *request* data kembali ke MongoDB sebagai data API (*Application Programming Interface*) dalam format JSON. Selanjutnya, data API yang tersimpan pada MongoDB akan digunakan pada sistem *Chatbot* menggunakan metode *Dual Intent and Entity Transformer (DIET) Classifier* dan *fuzzy string matching* melalui Rasa Framework. Diagram Blok yang menggambarkan keseluruhan sistem pada penelitian ini dapat dilihat pada Gambar 3.1.



Gambar 3.1. Diagram Blok Sistem

3.2. Data

Data yang digunakan untuk penelitian ini adalah data tanya jawab yang dikumpulkan dari Website <https://www.mafindo.or.id/> dan channel youtube Mafindo. Dan menggunakan *Search engine* yang dapat mencari informasi yang berasal dari website <https://www.liputan6.com>, <https://cekfakta.tempo.co>, <http://turnbackhoax.id/>, dan <https://www.kompas.com/>.

Website <http://turnbackhoax.id/> merupakan website yang berisi tentang kumpulan informasi palsu (hoax) yang sudah diklasifikasikan sesuai dengan fakta yang beredar oleh pihak Mafindo. Website <https://www.liputan6.com> merupakan website yang berisi artikel berita terbaru yang ada di Indonesia, <https://www.liputan6.com> memiliki saluran sumber cek fakta yang selalu menyajikan artikel verifikasi untuk informasi palsu (hoax) berdasarkan klaim-klaim tertentu secara lebih sistematis dan terstruktur. Website <https://cekfakta.tempo.co> merupakan website yang dikelola oleh tempo.co dengan kolaborasi dengan komunitas Mafindo dalam memeriksa informasi palsu (hoax). Website <https://www.kompas.com/> merupakan website yang berisi artikel berita terbaru yang ada di Indonesia.

Website <https://www.mafindo.or.id/> merupakan website komunitas anti-hoax resmi yang telah melakukan berbagai kegiatan untuk melawan informasi palsu (hoax), seperti hoax busting, edukasi publik, seminar, advokasi ke berbagai pihak. Sedangkan, channel youtube Mafindo merupakan suatu channel video anti-hoax yang dibuat oleh Mafindo untuk berkomunikasi dan berdiskusi dengan masyarakat mengenai penyebaran informasi palsu (hoax).

Data yang dikumpulkan dari website <https://www.mafindo.or.id/>, dan channel youtube Mafindo merupakan data tanya jawab yang dijadikan sebagai data API. Sedangkan *search engine* dapat mencari informasi secara *real-time* yang berasal dari website <https://www.liputan6.com>, <https://cekfakta.tempo.co>, <http://turnbackhoax.id/>, dan <https://www.kompas.com/> yang akan digunakan *chatbot* dalam memberikan respon jawaban kepada *user*. Contoh data API dapat dilihat pada Tabel 3.1

Tabel 3.1. Contoh Data API

Pertanyaan	Response
Kerugian apa yang dialami oleh media yang diserang oleh hoaks?	kerugian yang dialami itu yaitu kepercayaan public sehingga media tersebut akan dicap sebagai media hoaks, lalu pihak media akan konsisten dalam melakukan klasifikasi atau penghimpunan data sebab tugas dari media yaitu menyampaikan informasi dan melakukan verifikasi isu-isu dan tanggapan-tanggapan yang beredar di masyarakat.
ada situs-situs yang menyebarkan berita hoaks, kira-kira apa pendapat jurnalis mengenai situs-situs tersebut?	situs-situs itu sengaja dibuat oleh sekelompok orang di era WFH karena masyarakat tidak bisa keluar dan situs-situs tersebut bersifat negatif bagi masyarakat karena kemungkinan besar tidak dibuat oleh orang yang berkegiatan dalam bidang jurnalistik sebab jurnalistik media memiliki pedoman undang-undang Pers dan diawasi oleh Dewan Pers yang harus dilakukan oleh instant Pers, sehingga jika ada informasi yang menyebabkan hoaks tanpa mencantumkan sumber-sumber yang jelas itu sudah dipastikan bukan dari media-media massa dan hanya merupakan situs-situs yang berusaha untuk mendapatkan sorotan publik ataupun netizen untuk mendapatkan keuntungan.

Tabel 3.1. Contoh Data API (Lanjutan)

Pertanyaan	Response
apa penyebab sering munculnya situs-situs yang menyebarkan informasi hoaks?	penyebab munculnya situs-situs tersebut, upaya dalam memberikan pemahaman yang salah kepada masyarakat, karena hoaks ini bertujuan untuk memberikan image yang salah kepada masyarakat sehingga dapat menjatuhkan ataupun menyalahkan suatu kelompok ataupun pemerintahan.
cara efektif untuk mengatasi situs-situs yang menyebarkan informasi hoaks?	cara efektif yaitu memperbanyak literasi pada media, dan mencari dan membandingkan informasi yang didapatkan ke media massa yang terpercaya lainnya. Karena media massa selalu menyaring informasi yang didapatkan.
apa alasan Tirto membuat riset mandiri terkait hoax di tahun 2020 ?	alasan Tirto membuat riset hoax di tahun 2020 adalah untuk mengetahui seberapa banyak hoax tentang pandemi yang beredar di tahun 2020 dan info hoax itu diambil dari situs Mafindo
Mengapa Tirto mengambil data dari Mafindo?	karena Mafindo merupakan lembaga pemeriksa fakta yang sudah berjalan dari tahun 2015 dan kemudian diikuti oleh Tirto, tempo Liputan6 dan lembaga-lembaga pemeriksa fakta lain yang tergabung ke International Fact Checking Network (IFCN)
Bagaimana manfaat riset mandiri Tirto terkait hoax itu sendiri?	riset terkait hoax itu sendiri berusaha memberikan insight dan pemetaan yang diambil berdasarkan kesimpulan tiap agenda publik yang terjadi di Indonesia, sehingga masyarakat dapat berhati-hati terhadap berita hoax yang beredar

Tabel 3.1. Contoh Data API (Lanjutan)

Pertanyaan	Response
apa manfaat hasil riset mandiri Tirto untuk para pemangku kepentingan seperti platform media sosial?	manfaat untuk pemangku kepentingan media sosial yaitu agar para pemangku media sosial tersebut dapat lebih berhati-hati dengan agenda-agenda publik yang ada di Indonesia dan dapat menerapkan kebijakan yang inklusif dalam menghindari berita hoax
apakah banyak berita hoax pandemi yang dikaitkan dengan politik?	berita hoax pandemi memang banyak dikaitkan dengan politik sehingga berita hoax pandemi dapat didefinisikan dalam berbagai kategori
bagaimana pendapat dari sudut pandang MUI mengenai hoax bertema agama?	menurut MUI, berita hoax tidak bertema agama tidak bisa dibenarkan dikarenakan berita kebohongan itu dapat berdampak luas kepada orang lain serta bangsa dan negara
mengapa hoax yang bertema agama selalu dikaitkan dengan politik ?	sebab ada sebagian kalangan yang ingin menjadikan agama sebagai kendaraan politik dalam menyebarkan kebohongan politik menggunakan agama
bagaimana cara MUI menangkal hoax bertema agama?	MUI memiliki team dalam mengklarifikasi ataupun memberikan informasi yang lebih valid mengenai berita tersebut

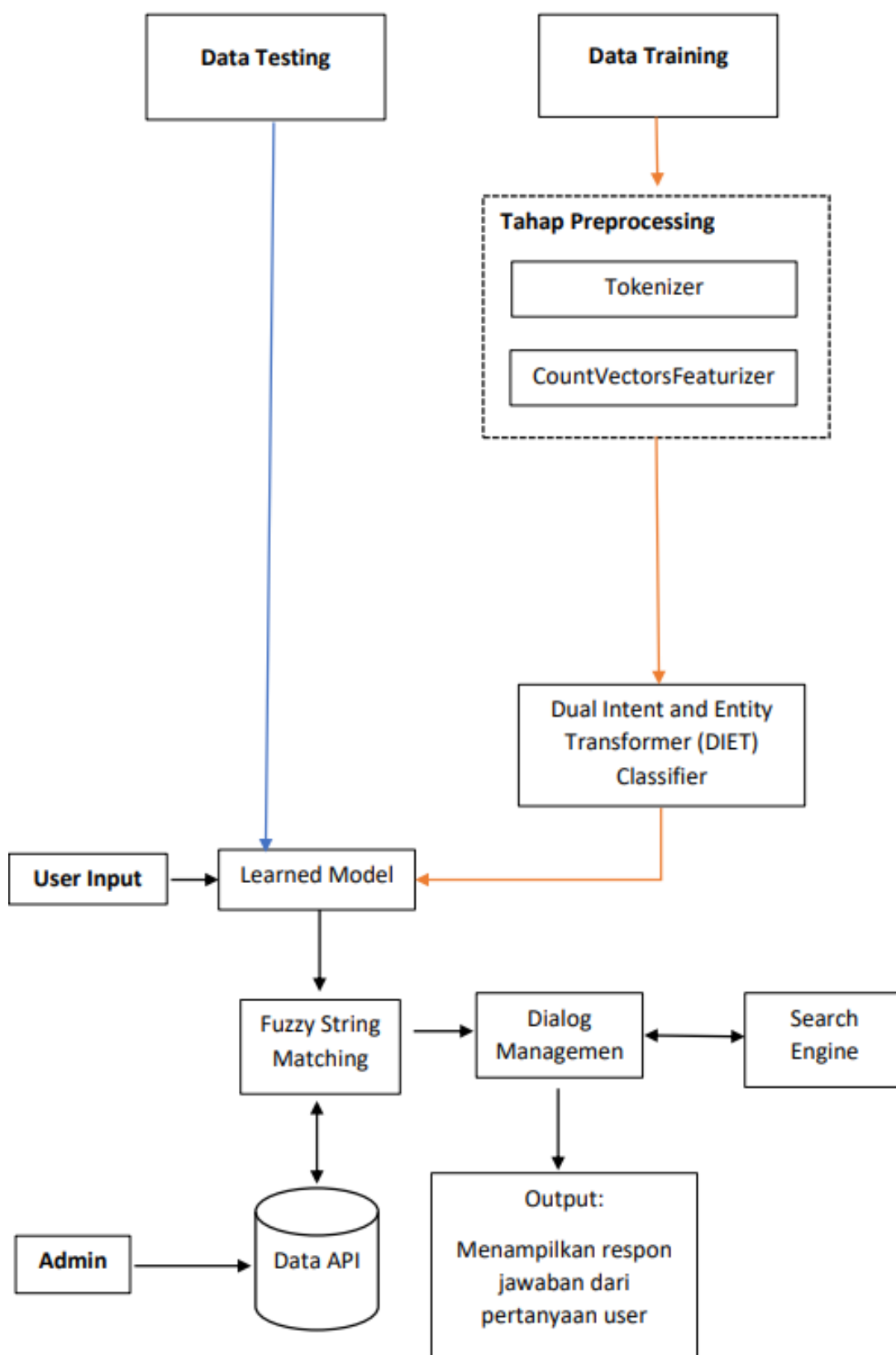
Tabel 3.1. Contoh Data API (Lanjutan)

Pertanyaan	Response
fenomena masyarakat dalam melihat hoax?	masyarakat merasa hoax perlu dibasmi untuk mencegah hal yang tidak diinginkan terjadi
seberapa efektif iklan layanan masyarakat (PSA) dalam menangkal hoax?	iklan layanan masyarakat (PSA) melakukan pendekatan cerita atau film sebagai medium yang efektif dalam memberikan informasi agar masyarakat tidak merasa digurui
apa tujuan dari Pubic Service Announcemen (PSA) ?	PSA ataupun Iklan Layanan Masyarakat berfungsi untuk menyampaikan informasi kepada masyarakat
tips masyarakat dalam membuat PSA anti-hoax	masyarakat dapat mulai dari melihat banyak film pendek inspiratif PSA dan membuat suatu cerita yang menarik serta memikirkan script yang bagus

3.3. Arsitektur Umum

Penelitian ini akan dilakukan dalam beberapa tahapan, antara lain, pengumpulan data dimana data tersebut merupakan kumpulan pertanyaan dan jawaban mengenai informasi palsu (hoax) yang diambil dari <https://www.mafindo.or.id/>, dan channel youtube Mafindo, data tersebut dibuat kedalam data API dan dapat dikelola oleh *admin*. Selanjutnya data API dibagi menjadi data uji dan data latih, kemudian data latih akan memasuki tahap pre-processing. Tahapan pre-processing terdiri dari *Tokenizer* dan *CountVectorsFeaturizer*. Setelah data latih melewati tahap *Tokenizer* maka akan memasuki tahap *CountVectorsFeaturizer*, dimana memiliki tujuan untuk mengubah data-data menjadi bentuk vector yang digunakan dalam menanggapi intent, kemudian, hasil dari tahap tersebut akan digunakan ke dalam tahap pembuatan model. Pada proses ini, model akan dilatih dengan data yang sudah di preprocessing. Setelah model di training, uji kemampuan model dilakukan menggunakan data uji untuk memprediksi intent untuk mencocokkan pola pertanyaan dan memilih respon untuk menampilkan output jawaban yang akurat.

User input berbentuk text chat yang dikirim oleh *user* ke *chatbot* dan akan masuk ke tahap learned model yang dapat memprediksi intent untuk mencocokkan pola pertanyaan yang ada pada data API menggunakan Fuzzy String Matching untuk menampilkan output jawaban yang akurat. Sedangkan, *search engine* dapat mencari informasi yang berasal dari website <https://www.liputan6.com>, <https://cekfakta.tempo.co>, <http://turnbackhoax.id/>, dan <https://www.kompas.com/>. Arsitektur umum yang menggambarkan setiap tahapan pada penelitian ini dapat dilihat pada Gambar 3.2.



Gambar 3.2. Arsitektur Umum

3.3.1. Split Data

Data API akan diubah ke dalam bentuk nlu.yml menggunakan *pandas*, lalu data yang sudah ada dalam bentuk nlu.yml akan di-*split* menggunakan command “*rasa data split nlu*”. Dengan jumlah data NLU (*Natural Language Understanding*) sebanyak 100 dan di-*split* secara acak menjadi 20% data uji dengan sebanyak 20 data tanya jawab dan 80% data latih dengan sebanyak 80 data tanya jawab.

3.3.2. Preprocessing

3.3.2.1. Tokenizer

Dalam tahapan ini, data teks yang berbentuk kalimat ataupun paragraf akan dipecah menjadi satuan kata. Pada penelitian ini proses *Tokenizer* diperlukan dalam melakukan pemisahan setiap kata yang memiliki makna tertentu. Contoh penerapan *Tokenizer* dapat dilihat pada Tabel 3.2.

Penggalan Kode *Tokenizer*

```

Class CustomTokenizer(Tokenizer):
    def __init__(self, component_config: Optional[Dict[Text, Any]] =
None) -> None:
        super().__init__(component_config)
    def tokenize(self, message: Message, attribute: Text) ->
List[Token]:
        text = message.get(attribute)
        words = re.findall(r'\b\w+\b', text.lower())
        tokens = []
        start = 0
        for word in words:
            token = Token(word, start)
            tokens.append(token)
            start += len(word) + 1
        return tokens

```

Table 3.2. Contoh Penerapan *Tokenizer*

Sebelum <i>Tokenizer</i>	Sesudah <i>Tokenizer</i>
apa penyebab sering munculnya situs-situs yang menyebarkan situs', 'yang', 'menyebarkan', 'informasi', informasi hoaks	['apa', 'penyebab', 'sering', 'munculnya', 'situs-situs', 'yang', 'menyebarkan', 'informasi', 'hoaks']

3.3.2.2 CountVectorsFeaturizer

Pada tahap ini dilakukan proses *CountVectorsFeaturizer* untuk menghitung jumlah kemunculan kata dalam data dan menghasilkan representasi vektor untuk setiap kata tersebut. Representasi ini hanya mencakup informasi tentang frekuensi kemunculan kata dalam data dan tidak memperhitungkan konteks atau hubungan antar kata. Dalam hal ini, *CountVectorsFeaturizer* dapat dianggap sebagai representasi teks yang relatif sederhana dan tidak memperhitungkan arti atau makna kata. Setelah proses *CountVectorsFeaturizer* terbentuk, selanjutnya digunakan dalam menanggapi intent pada model. Contoh penerapan *CountVectorsFeaturizer* dapat dilihat pada Tabel 3.3

Tabel 3.3. Contoh Penerapan *CountVectorsFeaturizer*

```
cek berita virus virtex
{
  "text": "cek berita virus virtex",
  "intent": {
    "id": 8537441080812072163,
    "name": "search_google",
    "confidence": 0.9799091219902039
  },
  "entities": [],
  "intent_ranking": [
    {
      "id": 8537441080812072163,
      "name": "search_google",
      "confidence": 0.9799091219902039
    }
  ]
}
```

```

    },
    {
      "id": 7884393087880158403,
      "name": "question",
      "confidence": 0.019184881821274757
    },
    {
      "id": -2599594072194022127,
      "name": "start",
      "confidence": 0.0005819965736009181
    },
    {
      "id": 1398688877044962153,
      "name": "goodbye",
      "confidence": 0.00021920898871030658
    },
    {
      "id": -1944470843256161350,
      "name": "bot",
      "confidence": 5.259932004264556e-05
    },
    {
      "id": -5709476966672481467,
      "name": "out_of_scope",
      "confidence": 5.213536860537715e-05
    }
  ]
}

```

Pada contoh penerapan *CountVectorsFeaturizer* kalimat “cek berita virus virtex” terdapat kesamaan (*confidence*) sebesar 0.9799091219902039 pada intent *search_google*.

3.3.3. Model Building

Tahapan berikutnya merupakan pembentukan model dengan menggunakan arsitektur *Dual Intent and Entity Transformer (DIET)* untuk klasifikasi *intent* dalam menyesuaikan dengan jawaban yang akan diberikan sistem. Proses algoritma dimulai dari kata yang sudah ditokenizer dan sudah dibuat kedalam karakter vector lalu diproses ke dalam 2 tahap yaitu *Sparse Features* dan *Pretrained Embedding*, hasil dari proses tersebut akan diproses ke tahap *transformer (2 layers)*. Arsitektur ini memiliki 3 component yaitu:

1. *intent classification*, pada tahap *similarity* menggunakan *intent* yang dianggap sebagai label akan dilakukan proses pencocokan dengan token yang ada pada *transformer (2 layers)* apabila tidak ada kemiripan maka akan menghasilkan “*intent loss*”.
2. *named entity recognition*, pada tahap *similarity* menggunakan *entity* yang diproses pada *CRF (Condition Random Field)* yang mana apabila tidak memiliki kemiripan maka akan menghasilkan “*entity loss*”.
3. *mask language model*, pada tahap ini akan menggunakan token kata yang diproses melalui *Sparse Features* dan *Pretrained Embedding* lalu dilakukan pencocokan dengan token yang ada pada tahap *transformer (2 layers)* dan apabila tidak memiliki kemiripan maka akan menghasilkan “*mark loss*”.

Hasil dari *entity loss*, *total loss*, dan *mask loss* akan disatukan menjadi *total loss*. Setelah didapatkan *total loss* maka model sudah terbentuk, selanjutnya akan dilakukan pengujian model.

3.3.4. Testing Model

Pada tahap ini, akan dilakukan 2 pengujian terhadap model

1. Pengujian menggunakan data uji
 Pada tahap ini dilakukan pengujian menggunakan data uji dengan tujuan untuk mengamati akurasi dari data yang telah dimodelkan menggunakan *Dual Intent and Entity Transformer (DIET) Classifier*

2. Pengujian *Testing*

Tahap testing dilakukan untuk mengamati performa dari kemampuan model dalam menjawab input berupa pertanyaan dari *user*.

3.3.5. Implementasi *Fuzzy String Matching*

Pada tahapan ini, pertanyaan *user* akan disesuaikan dengan pertanyaan yang berada pada data API menggunakan algoritma *fuzzy string matching* untuk memunculkan jawaban sesuai dengan pertanyaan *user tersebut*. Adapun Tahapan penyesuaian tersebut yaitu:

1. *string* akan dicocokkan dengan data API dan apabila memiliki kesamaan, hasil dari jawaban tersebut akan ditampilkan pada *chatbot*.
2. jika belum ada *string* yang ditemukan maka akan dilakukan pemeriksaan *string* berdasarkan persamaan huruf awal, panjang *string* yang sama, dan huruf akhir. Ketika ditemukan kemiripan dari *string* tersebut maka jawaban dari pertanyaan tersebut akan ditampilkan.

Contoh: “hoaks menyerang media” maka proses *string matching* akan dimulai dengan mencari kata pertama dengan kemiripan panjang *string* yang sama yaitu “hoaks”. Setelah itu, akan dilakukan pemeriksaan pada data API ternyata ada beberapa kata yang cocok yaitu “kenapa hoaks bisa menyerang media-media informasi?”, “apa saja bentuk hoaks yang menyerang media-media informasi?” dan seterusnya. Kemudian proses *string matching* akan melihat kemiripan antara “menyerang” dengan “menyerang” dan “media” dengan “media”. Sehingga jawaban untuk pertanyaan *user* dapat diambil dari pertanyaan yang ada di data API yaitu “kenapa hoaks bisa menyerang media-media informasi?” karena ada kecocokan antara kata awal dan kata akhir.

3. Tahapan selanjutnya apabila belum menemukan *string* yang sesuai maka akan dilakukan pemeriksaan huruf awal dan akhir dengan menghiraukan panjangnya, apabila menemukan sebuah kesamaan maka jawaban dari pertanyaan tersebut akan ditampilkan.
4. Apabila tidak memiliki kesamaan sama sekali terhadap data API maka akan menampilkan jawaban “maaf, saya tidak menemukan jawaban untuk pertanyaan tersebut”.

Penggalan Kode *Fuzzy String Matching*

```

query = tracker.latest_message['text']
questions = list(self.faq['pertanyaan'])
answer = list(self.faq['response'])
Ratios = process.extract(query, questions)
print(Ratios)
mathed_question, score = process.extractOne(query, questions,
scorer=fuzz.token_set_ratio)
if score > 50:
    matched_row = self.faq.loc[self.faq['pertanyaan'] ==
mathed_question,]
    answer = matched_row['response'].values[0]
    response = "{ } \n".format(answer)
else:
    response = "maaf, saya tidak menemukan jawaban untuk
pertanyaan tersebut"
dispatcher.utter_message(response)

```

3.3.6. Implementasi *Search Engine*

Pada tahap ini, *search engine* akan diterapkan pada *chatbot* menggunakan CSE (*Custom Search Engine*) yang merupakan layanan dari Google yang mengizinkan dalam membuat *search engine* sendiri. CSE (*Custom Search Engine*) menggunakan teknologi pencarian yang sama seperti Google, akan tetapi dapat mengatur pencarian pada website tertentu yang ingin diterapkan pada *search engine* yang dibuat. Pada penelitian ini, penulis menggunakan CSE (*Custom Search Engine*) yang mengatur pencarian pada website <https://www.liputan6.com>, <https://cekfakta.tempo.co>, <http://turnbackhoax.id/>, dan <https://www.kompas.com/>.

Penggalan Kode *Search Engine*

```

query = message.replace("cek berita", "").strip()
service = build("customsearch", "v1", developerKey="developerKey")
result = service.cse().list(q=query, cx="cxKey", num=3).execute()
results = result.get("items", [])
response = ""
for item in results:
    response += f"{item['title']}\n{item['snippet']}\n{item['link']}\n\n"
if response:
    dispatcher.utter_message(response)
else:
    dispatcher.utter_message("maaf, saya tidak menemukan hasil dari
informasi yang anda cari")

```

3.3.7. Dialog Management

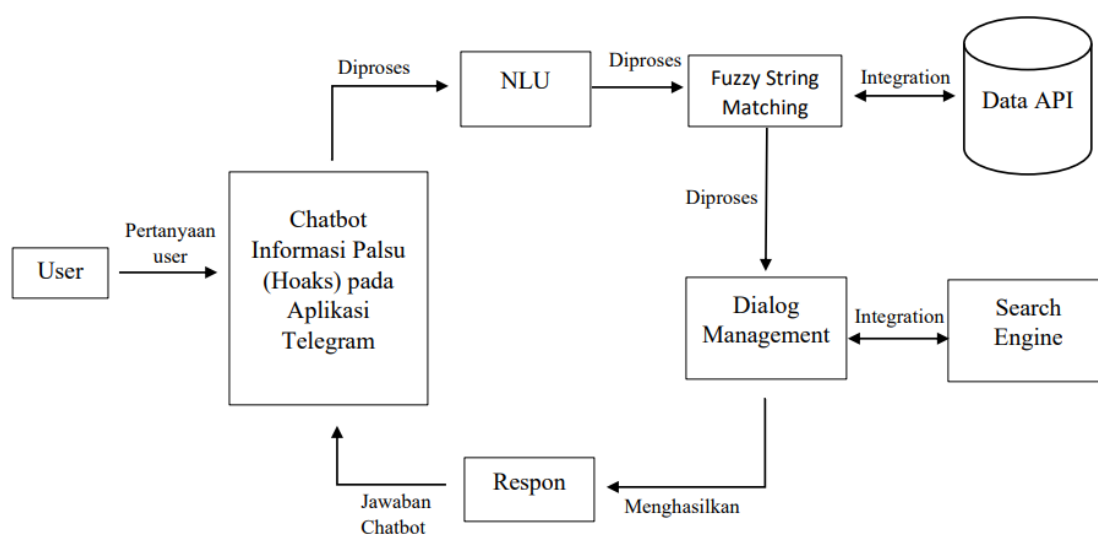
Dialog management merupakan tool dalam melakukan percakapan dengan *chatbot*. Setelah *intent classification* dari *DIET Classifier* dan *fuzzy string matching* dilakukan, dialog management akan memprediksi respon yang tepat untuk percakapan dengan user. Dialog management pada sistem chatbot dapat menggunakan *rule-based* ataupun *machine learning-based* yang bertujuan untuk menangani percakapan dan memberikan respon jawaban yang tepat. Pada penelitian ini akan menggunakan *TED policy* sebagai machine learning based sebagai metode dialog management. *TED policy* akan mengambil intent dari *user input* lalu akan memprediksi respon yang tepat.

3.3.8. Respon

Respon merupakan jawaban atas pertanyaan user yang diajukan kedalam chatbot

3.4. Diagram Alur Sistem Chatbot

Diagram alur sistem *chatbot* menjelaskan alur yang dapat dilakukan pada sistem *chatbot*. Saat *user* masuk ke channel *chatbot*, sistem akan memperlihatkan halaman tampilan *chatbot*. *User* dapat memasukkan pertanyaan seputar hoaks ataupun memeriksa berita pada channel *chatbot*. Setelah itu pertanyaan *user* diproses pada *Natural Language Understanding* (NLU) melalui preprocessing dan diidentifikasi oleh model *DIET Classifier*, hasil dari proses *Natural Language Understanding* (NLU) dilanjutkan ke proses *similarity string* oleh *fuzzy string matching* yang terhubung dengan data API dalam melakukan pencocokan pertanyaan dari *user* dengan *string* yang ada pada data API. Apabila ditemukan pola pertanyaan yang paling sesuai, jawaban dari pola pertanyaan tersebut diteruskan ke dialog manajemen, sedangkan apabila *user* ingin mencari kebenaran informasi maka dapat menggunakan kata kunci (*keyword*) ‘cek berita’ sehingga dialog manajemen mencari informasi yang sudah terhubung dengan *search engine* dan menghasilkan jawaban dari pencarian *search engine* tersebut. Dialog manajemen menggunakan TED Policy (*Transformer Embedding Dialogue Policy*) yang mengatur respon jawaban yang ditampilkan pada channel *chatbot* telegram tersebut. Respon jawaban *chatbot* dari pertanyaan *user* yang sudah diproses, ditampilkan ke channel *chatbot*. Diagram alur sistem *chatbot* dapat dilihat pada Gambar 3.3.



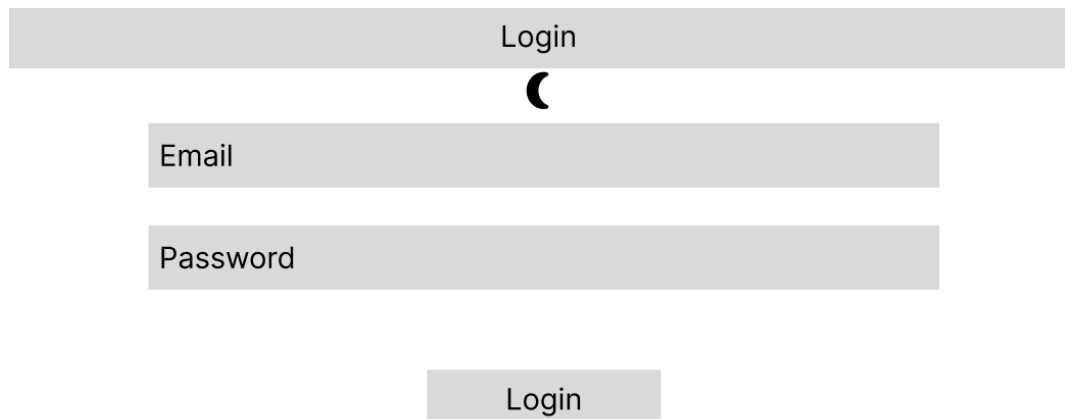
Gambar 3.3. Diagram Alur Sistem *Chatbot*

3.5. Perancangan Antarmuka Aplikasi

Pada bagian ini akan dipaparkan mengenai rancangan antarmuka aplikasi yang dibuat dalam menjalankan sistem yang dikembangkan.

3.5.1. Rancangan Tampilan Login pada Web Admin

Tampilan login adalah tampilan yang akan muncul saat *admin* pertama kali membuka sistem *web admin*. Rancangan tampilan login pada *web admin* dapat dilihat dalam Gambar 3.4.

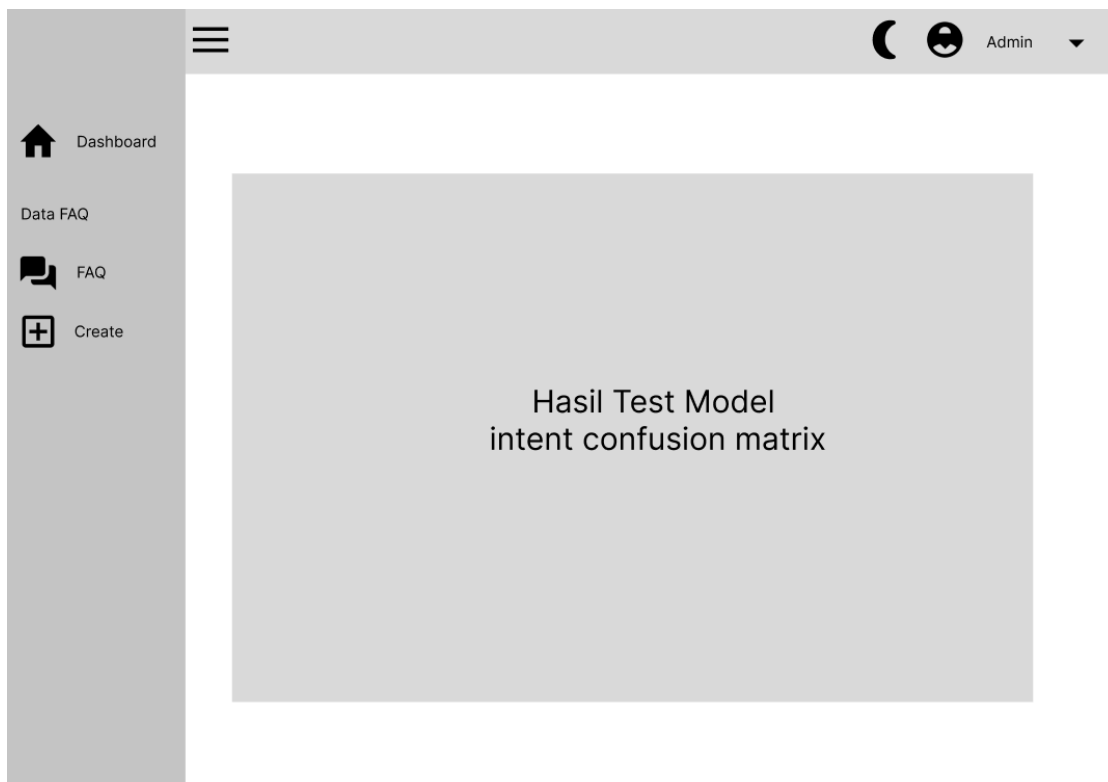


The image shows a login form layout. At the top, there is a wide gray rectangular box containing the word "Login" in the center. Below this box is a small black crescent-shaped icon. Underneath the icon are two stacked gray rectangular input fields. The top field is labeled "Email" on its left side, and the bottom field is labeled "Password" on its left side. At the bottom of the form, centered, is a gray rectangular button with the word "Login" in the center.

Gambar 3.4. Rancangan Tampilan Login pada Web Admin

3.5.2. Rancangan Tampilan Dashboard pada Web Admin

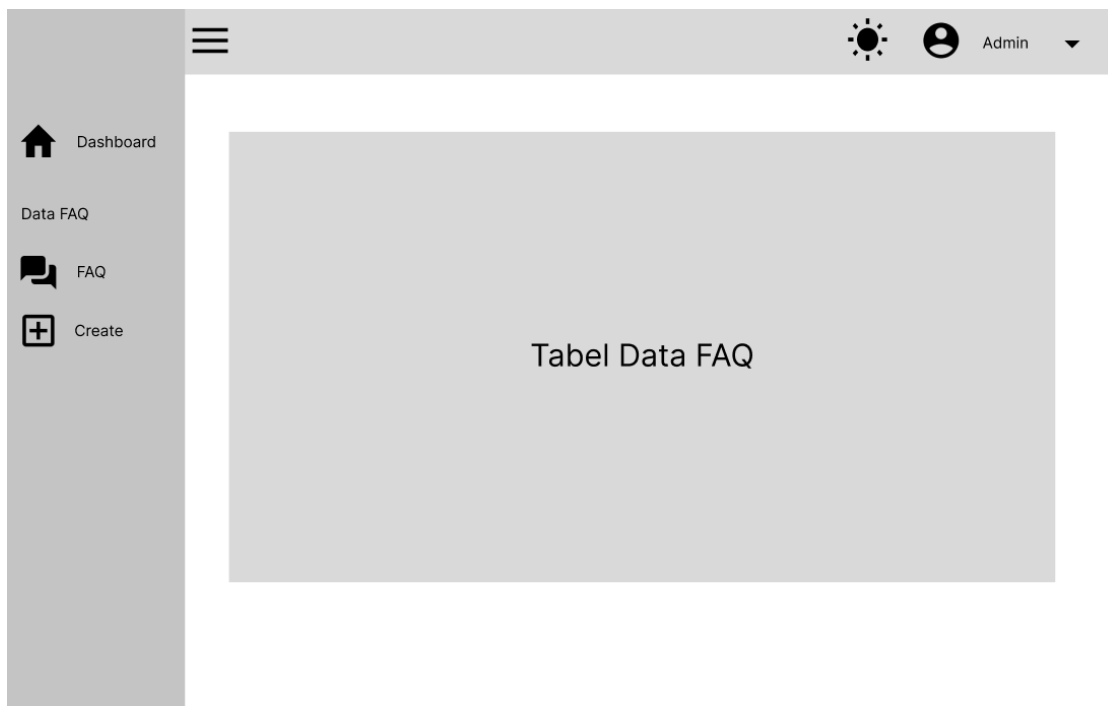
Tampilan dashboard adalah tampilan yang muncul ketika *admin* sudah berhasil masuk kedalam sistem *web admin*. Di halaman ini terdapat hasil test model algoritma DIET. Rancangan tampilan dashboard pada *web admin* dapat dilihat dalam Gambar 3.5.



Gambar 3.5. Rancangan Tampilan Dashboard pada Web Admin

3.5.3. Rancangan Tampilan List Data FAQ pada web Admin

Tampilan list data FAQ adalah tampilan yang memunculkan tabel dari data API. Rancangan tampilan list data FAQ pada *web admin* dapat dilihat dalam Gambar 3.6.



Gambar 3.6. Rancangan Tampilan List Data FAQ pada Web Admin

3.5.4. Rancangan Tampilan Create pada Web Admin

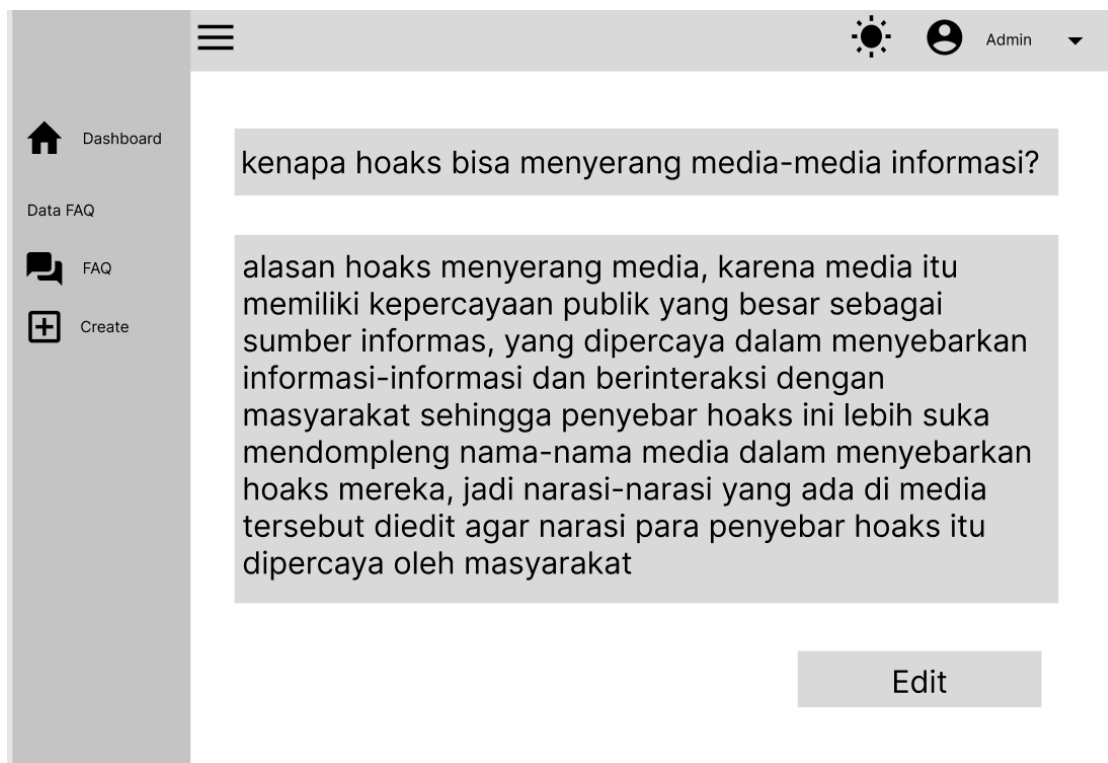
Tampilan *create* adalah tampilan yang memunculkan untuk menambahkan data pada data API. Rancangan tampilan pembuatan pertanyaan dan respon pada web admin dapat dilihat dalam Gambar 3.7.

The image shows a wireframe of a web admin interface for creating new data. It features a sidebar on the left with navigation links: 'Dashboard' (home icon), 'Data FAQ', 'FAQ' (speech bubble icon), and 'Create' (plus icon). The main content area has a top header with a hamburger menu, a sun icon, a user profile icon labeled 'Admin', and a dropdown arrow. Below the header, there are two large text input fields labeled 'Input Pertanyaan' and 'Input Response'. At the bottom right of the main area is a 'Create' button.

Gambar 3.7. Rancangan Tampilan Create pada Web Admin

3.5.5. Rancangan Tampilan Edit pada Web Admin

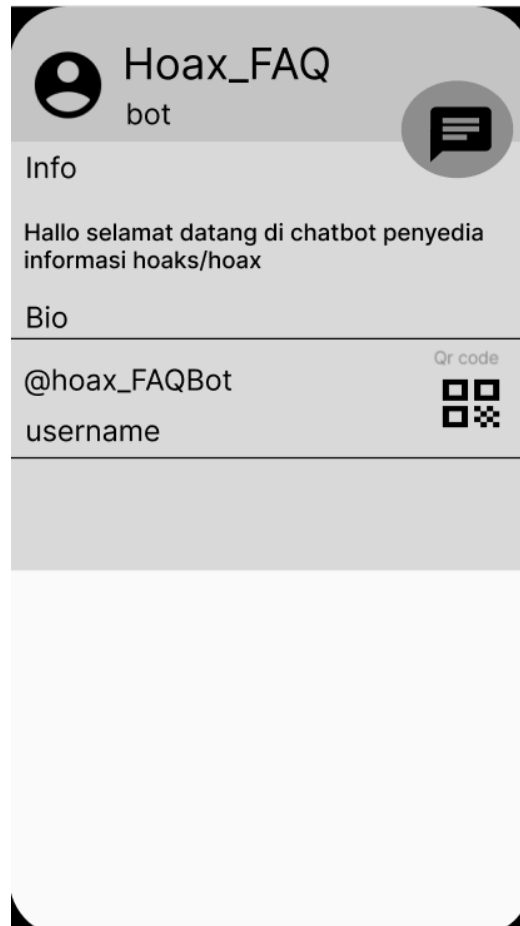
Tampilan *edit* adalah tampilan yang memunculkan untuk mengedit data pada data API. Rancangan Tampilan Edit Pertanyaan dan Respon pada web admin dapat dilihat dalam Gambar 3.8.



Gambar 3.8. Rancangan Tampilan Edit pada Web Admin

3.5.6. Rancangan Tampilan Profile Sistem Chatbot

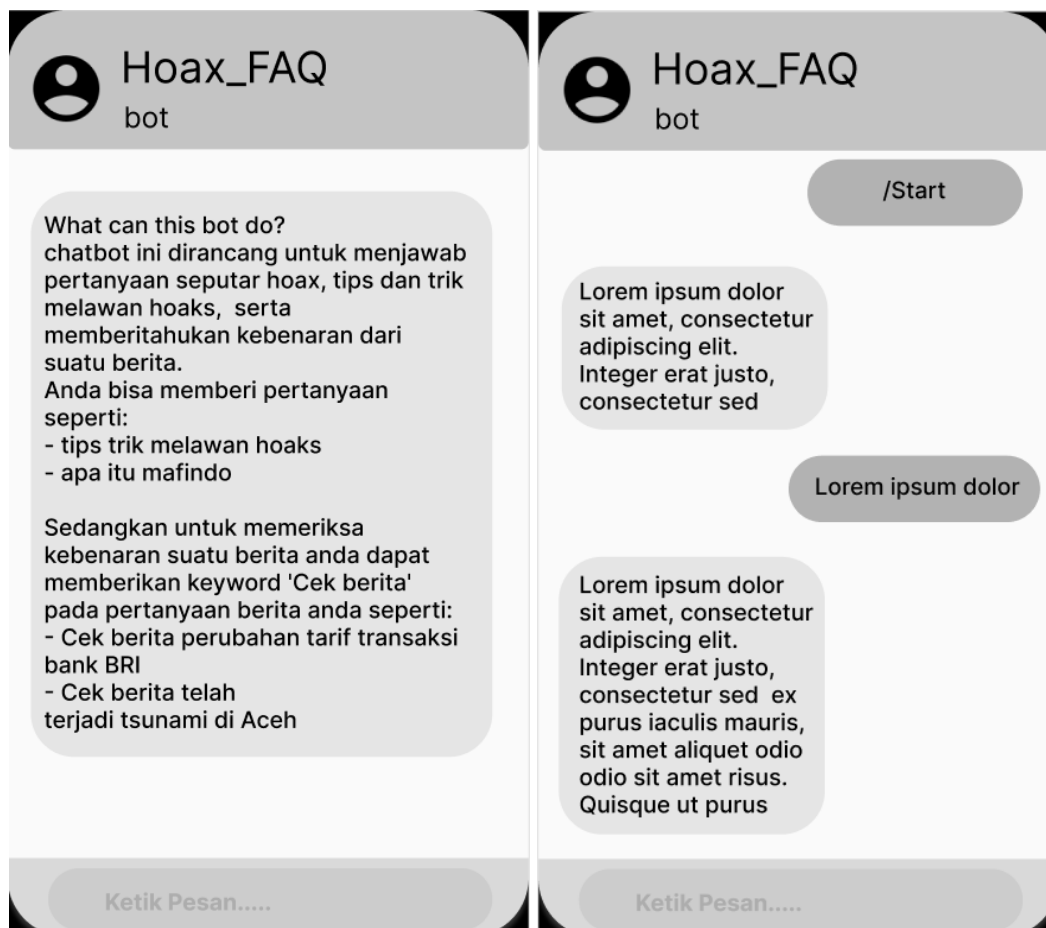
Tampilan *profile* merupakan halaman yang menjelaskan info dari sistem chatbot ketika *user* masuk dalam sistem chatbot. Rancangan Tampilan *Profile* Sistem Chatbot dapat dilihat dalam Gambar 3.9.



Gambar 3.9. Rancangan Tampilan Profile Sistem Chatbot

3.5.7. Rancangan Tampilan Halaman Chat pada Sistem Chatbot

Tampilan halaman chat merupakan halaman yang memungkinkan *user* melakukan percakapan dengan sistem *chatbot*. Rancangan Tampilan Halaman Chat pada Sistem Chatbot dapat dilihat dalam Gambar 3.10.



Gambar 3.10. Rancangan Tampilan Halaman Chat pada Sistem Chatbot

3.6 Evaluasi Model

Proses evaluasi model klasifikasi *intent* memerlukan metode-metode yang dapat mengukur kinerja model sehingga dapat diperhitungkan dalam memilih model dengan performa terbaik untuk data yang digunakan dalam melakukan perhitungan dan pengambilan keputusan.

Performance metrics yang digunakan pada penelitian ini terdiri dari:

1. *Accuracy*

metrics accuracy dihitung dengan rasio banyaknya data diprediksi sesuai dengan data aktual baik yang kelasnya positif atau negatif.

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative} \times 100\% \quad (3.1)$$

dimana:

True Positive = Banyak data yang diprediksi positif dari kelas aktual positif

False Positive = Banyak data yang diprediksi positif dari kelas aktual negatif

True Negative = Banyak data yang diprediksi negatif dari kelas aktual negatif

False Negative = Banyak data yang diprediksi negatif dari kelas aktual positif

2. *Precision*

Precision adalah rasio perbandingan True Positive (TP) dengan total banyaknya data yang diprediksi bernilai positif. Nilai precision tinggi berhubungan dengan tingkat False Positive (FP) yang rendah.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \times 100\% \quad (3.2)$$

dimana:

True Positive = Banyak data yang diprediksi positif dari kelas aktual positif

False Positive = Banyak data yang diprediksi positif dari kelas aktual negatif

3. *Recall*

Recall adalah rasio perbandingan True Positive (TP) dengan total banyaknya data positif yang sebenarnya.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \times 100\% \quad (3.3)$$

dimana:

True Positive = Banyak data yang diprediksi positif dari kelas aktual positif

False Negative = Banyak data yang diprediksi negatif dari kelas aktual positif

4. *F1-Score*

F1-Score adalah nilai harmonic mean dari precision dan recall

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \times 100\% \quad (3.4)$$

BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1. Implementasi Sistem

Pada Bab 3, penulis telah membuat rancangan sistem dan mengimplementasikan beberapa *hardware*/perangkat keras atau *software*/perangkat lunak yang berfungsi sebagai sarana pendukung, yaitu:

4.1.1. Spesifikasi Hardware dan Software

Pada penelitian ini, penulis menggunakan beberapa perangkat keras dalam melakukan implementasi sistem dengan beberapa spesifikasi seperti berikut:

1. Processor: Intel CORE i5 8th Gen
2. RAM(Memory): 8 GB DDR4
3. GPU: Nvidia GEFORCE GTX 1050
4. SSD: 500 GB M.2 NVMe 2280
5. HDD: 1 TB

Penulis juga menggunakan perangkat lunak dalam pembangunan sistem ini yaitu:

1. Operation System: Windows 10 Pro 64 Bit
2. Visual Studio Code
3. NodeJs dengan versi 18.12.1
4. NPM (Node Package Manager) dengan versi 8.19.2
5. ExpressJS dengan versi 4.18.2
6. ReactJS dengan versi 18.2.0
7. MongoDB
8. Python dengan versi 3.7.0
9. Rasa dengan versi 2.8.0
10. Rasa SDK dengan versi 2.8.0

11. Library yang berasal dari bahasa python, sebagai berikut:

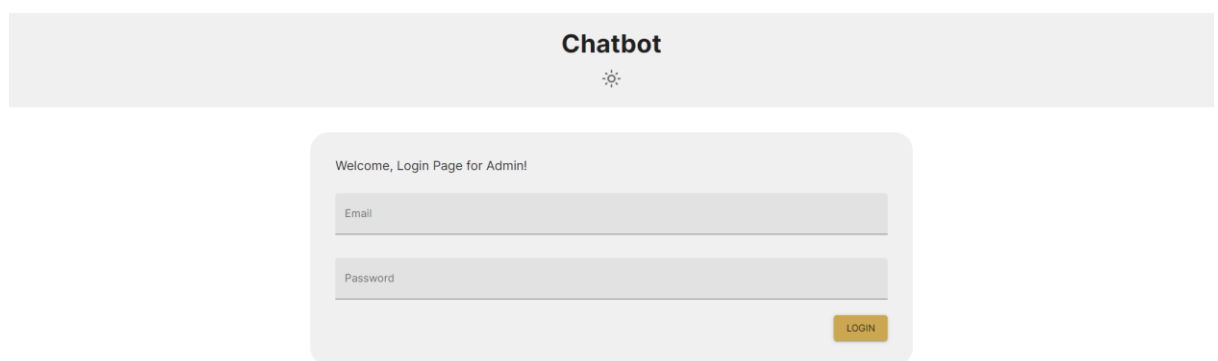
- a. Pandas
- b. Pymongo
- c. Fuzzywuzzy
- d. TensorFlow dan Keras
- e. Sklearn
- f. Numpy
- g. NLTK
- h. Google API Python Client

4.1.2. Implementasi Perancangan Antarmuka

Rancangan antarmuka yang dibahas pada Bab 3, telah memiliki hasil dan diimplementasikan dalam penelitian ini yaitu:

1. Tampilan Halaman Login pada Web Admin

Tampilan halaman *login* merupakan halaman pertama bagi *admin* dalam membuka *web admin*. Disini akan ada *field* email dan password yang diperlukan untuk melakukan proses *login*. Tampilan halaman *login* dapat dilihat pada Gambar 4.1.

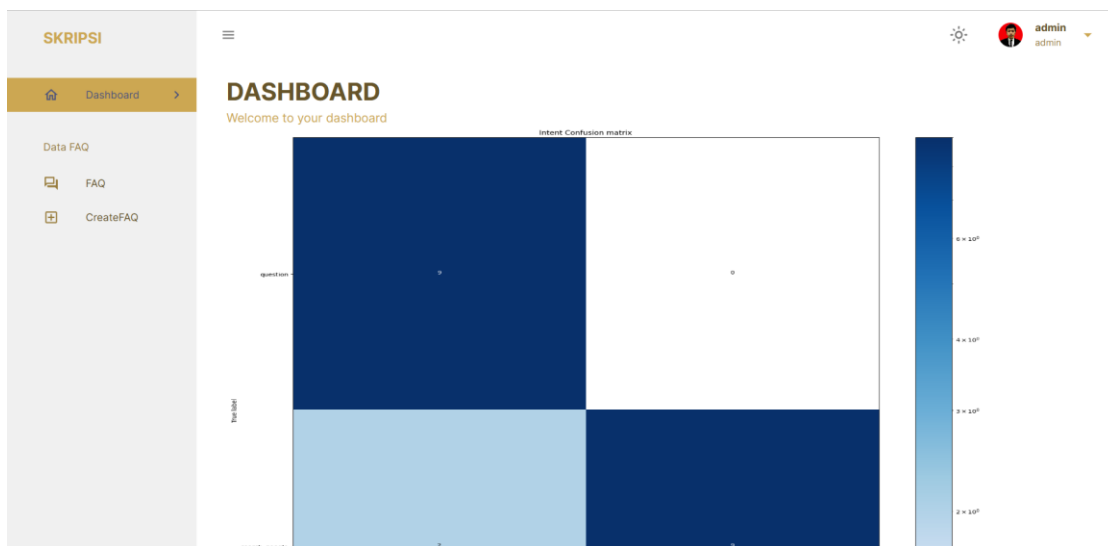


The image shows a web interface for a chatbot. At the top, there is a header bar with the text "Chatbot" and a small gear icon. Below the header, there is a login form. The form has a title "Welcome, Login Page for Admin!". It contains two input fields: "Email" and "Password". To the right of the "Password" field is a yellow button labeled "LOGIN".

Gambar 4.1. Tampilan Halaman Login pada Web Admin

2. Tampilan Halaman Dashboard pada Web Admin

Tampilan halaman *dashboard* merupakan halaman yang memberitahu hasil uji model *Dual Intent and Entity Transformer* (DIET) pada sistem chatbot. Tampilan halaman *dashboard* dapat dilihat pada Gambar 4.2.



Gambar 4.2. Tampilan Halaman Dashboard pada Web Admin

3. Tampilan Halaman List Data FAQ pada Web Admin

Tampilan halaman list data FAQ merupakan halaman yang memperlihatkan list data FAQ kepada *admin*. Disini *admin* dapat mengelola data FAQ yang ada di API. Tampilan halaman list data FAQ dapat dilihat pada Gambar 4.3.

ID	pertanyaan	response	edit	delete
63c113b0bed0b05a43579e0	kenapa hoaks bisa menyerang media-media informasi?	alasan hoaks menyerang media, karena media itu memiliki kepe...	EDIT	DELETE
63c113b0bed0b05a43579e1	apa saja bentuk hoaks yang menyerang media-media inform...	bentuk berita hoaks yang sering ditemukan dan menyebar di m...	EDIT	DELETE
63c113b0bed0b05a43579e2	Kerugian apa yang dialami oleh media yang diserang oleh hoaks?	kerugian yang dialami itu yaitu kepercayaan public sehingga m...	EDIT	DELETE
63c113b0bed0b05a43579e3	ada situs-situs yang menyebarkan berita hoaks, kira-kira apa p...	situs-situs itu sengaja dibuat oleh sekelompok orang di era WF...	EDIT	DELETE
63c113b0bed0b05a43579e4	apa penyebab sering munculnya situs-situs yang menyebarkan...	penyebab munculnya situs-situs tersebut, upaya dalam memerl...	EDIT	DELETE
63c113b0bed0b05a43579e5	cara efektif untuk mengatasi situs-situs yang menyebarkan inf...	cara efektif yaitu memperbanyak literasi pada media, dan menc...	EDIT	DELETE
63c113b0bed0b05a43579e6	Tips & trik untuk melawan hoax	Tips & trik untuk melawan hoax 📢 Hoax itu sangat berbahaya d...	EDIT	DELETE
63c113b0bed0b05a43579e7	Malindo	Tentang Malindo Dawal dari grup Facebook bernama FAFIH L...	EDIT	DELETE
63c113b0bed0b05a43579e8	apa alasan Tirto membuat riset mandiri terkait hoax di tahun 20...	alasan Tirto membuat riset hoax di tahun 2020 adalah untuk m...	EDIT	DELETE

Gambar 4.3. Tampilan Halaman List Data FAQ pada Web Admin

4. Tampilan Halaman Create Data API pada Web Admin

Tampilan halaman *create* data API merupakan halaman yang dapat digunakan oleh *admin* dalam menambahkan data pertanyaan dan jawaban ke data API. Tampilan halaman *create* data API dapat dilihat pada Gambar 4.4.

Gambar 4.4. Tampilan Halaman Create Data API pada Web Admin

5. Tampilan Halaman Edit Data API pada Web Admin

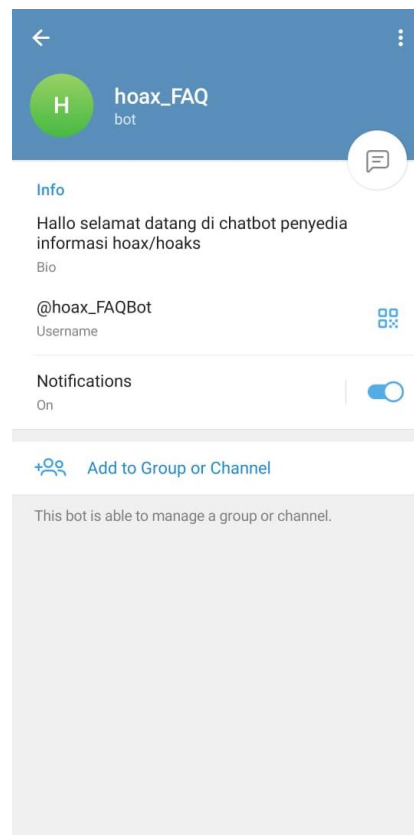
Tampilan halaman edit data API merupakan halaman yang dapat digunakan oleh admin dalam mengedit data pertanyaan dan jawaban ke data API. Tampilan halaman edit data API dapat dilihat pada Gambar 4.5.

Gambar 4.5. Tampilan Halaman Edit Data API pada Web Admin

6. Tampilan Halaman Profile pada Sistem Chatbot

Pada halaman *profile*, diberikan informasi *profile* dari akun *chatbot* telegram yang siap untuk menjawab pertanyaan *user* mengenai hoaks. Pada halaman *profile*, *user* dapat

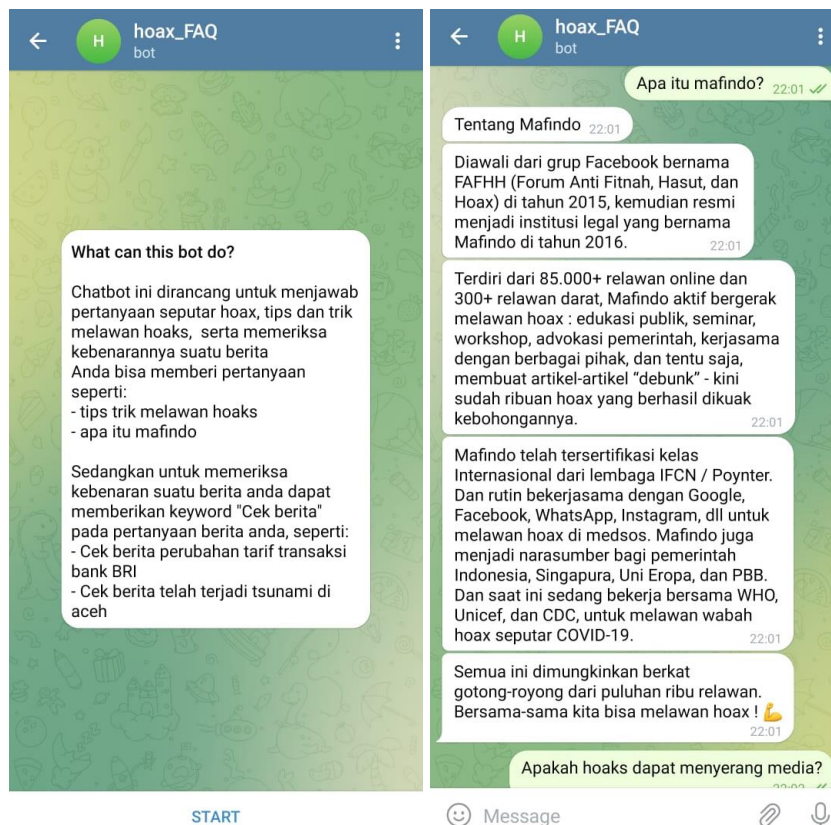
melihat avatar *chatbot*, informasi singkat mengenai *chatbot*, dan *username chatbot*. Tampilan halaman profile dapat dilihat pada Gambar 4.6.



Gambar 4.6. Tampilan Halaman Profile pada Sistem Chatbot

7. Tampilan Halaman Chat pada Sistem Chatbot

Pada halaman chat merupakan tampilan ketika *user* menjalankan *chatbot* pada Telegram. *Chatbot* akan mengirim pesan deskripsi singkat mengenai penggunaan dari *chatbot* informasi palsu (hoax). *User* dapat mengklik perintah */Start* sehingga *chatbot* dapat beroperasi. Tampilan halaman chat dapat dilihat pada Gambar 4.7.



Gambar 4.7. Tampilan Halaman Chat pada Sistem Chatbot

4.2. Implementasi Model

Saat melakukan pelatihan model, percobaan proses *training* telah dilakukan beberapa kali oleh penulis seperti mengubah kuantitas unit epoch agar menghasilkan performa terbaik dari model *Dual Intent and Entity Transformer (DIET) Classifier*. Proses model dengan performa terbaik yang didapatkan penulis pada penelitian ini adalah epoch sebesar 100 epoch dengan tingkat akurasi pada proses *training* mencapai 100%. Hasil proses *training* dapat dilihat pada Gambar 4.8.

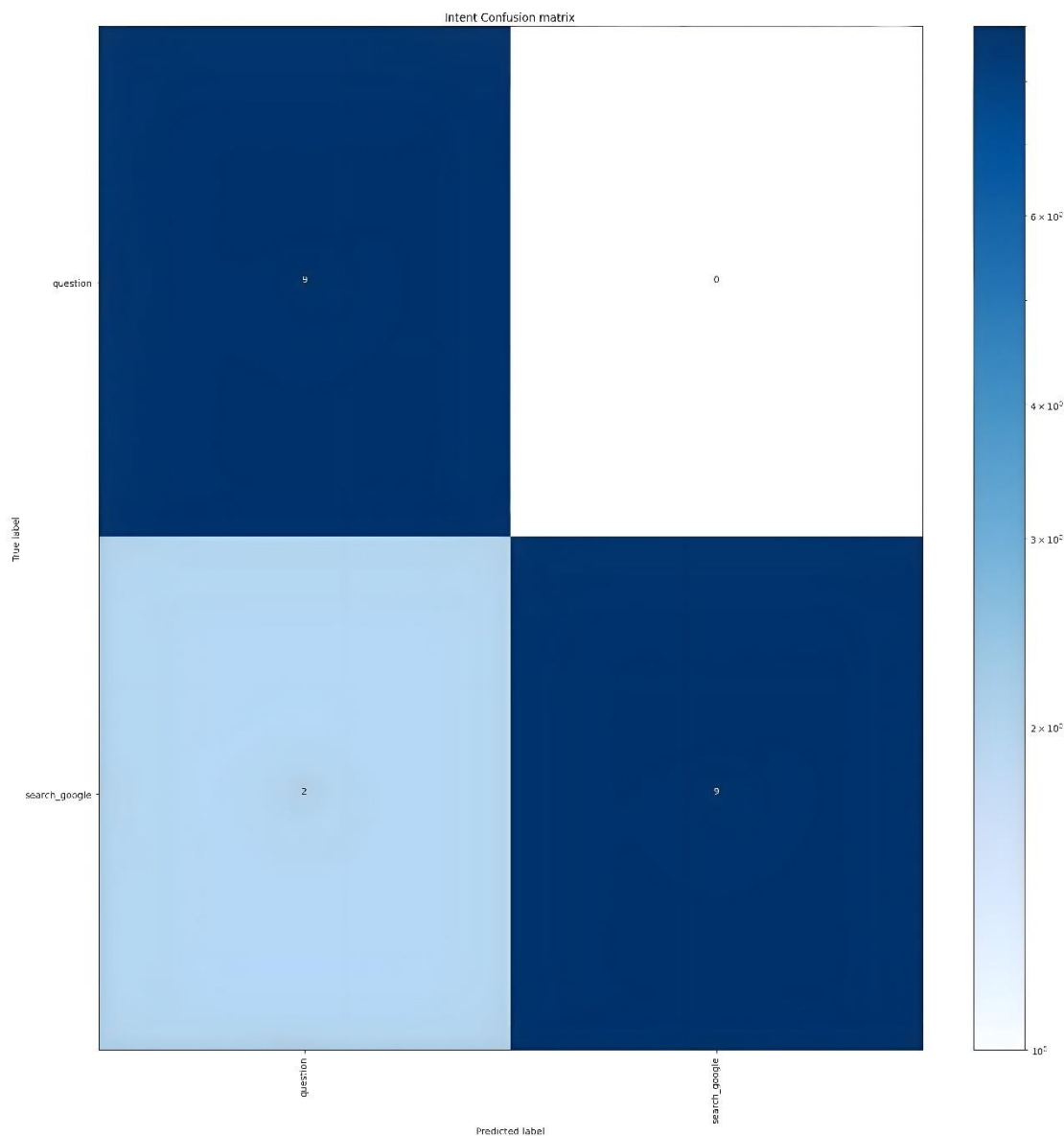
```

Training NLU model...
2023-01-13 15:44:38 INFO rasa.shared.nlu.training_data.training_data - Training data stats:
2023-01-13 15:44:38 INFO rasa.shared.nlu.training_data.training_data - Number of intent examples: 71 (6 distinct intents)
2023-01-13 15:44:38 INFO rasa.shared.nlu.training_data.training_data - Found intents: 'start', 'question', 'goodbye', 'search_google', 'out_of_scope',
2023-01-13 15:44:38 INFO rasa.shared.nlu.training_data.training_data - Number of response examples: 0 (0 distinct responses)
2023-01-13 15:44:38 INFO rasa.shared.nlu.training_data.training_data - Number of entity examples: 0 (0 distinct entities)
2023-01-13 15:44:38 INFO rasa.nlu.model - Starting to train component: WhitespaceTokenizer
2023-01-13 15:44:38 INFO rasa.nlu.model - Finished training component.
2023-01-13 15:44:38 INFO rasa.nlu.featureizers.sparse_featurizer.count_vectors_featurizer - 1544 vocabulary items were created for text attribute.
2023-01-13 15:44:38 INFO rasa.nlu.model - Finished training component.
2023-01-13 15:44:38 INFO rasa.nlu.model - Starting to train component: DIETClassifier
2023-01-13 15:45:13 INFO rasa.nlu.model - Finished training component.
2023-01-13 15:45:13 INFO rasa.nlu.model - Starting to train component: FallbackClassifier
2023-01-13 15:45:13 INFO rasa.nlu.model - Finished training component.
2023-01-13 15:45:13 INFO rasa.nlu.model - Successfully saved model into 'C:\Users\Wind\AppData\Local\Temp\tmp01qtkSwi\nlu'
NLU model training completed.
  
```

Gambar 4.8. Hasil proses training

4.3. Evaluasi Model

Evaluasi model bertujuan untuk mengukur kemampuan model *Dual Intent and Entity Transformer (DIET)* dalam melakukan klasifikasi *intent*. Pada evaluasi model ini digunakan data uji sebanyak 20 data. Hasil pengujian model ini menggunakan tabel *confussion matrix* untuk melihat banyaknya data yang bernilai *True Positive*, *True Negative*, *False Positive*, dan *False Negative* yang dapat dilihat pada Gambar 4.9.



Gambar 4.9. Confusion Matrix Akurasi dari Data Uji

Pada Gambar 4.9 dapat dilihat bahwa sebanyak 9 data *question* yang diprediksi benar (*True Negative*), dan sebanyak 0 data *question* yang diprediksi salah (*False Positive*). Data *search_google* yang diprediksi benar sebanyak 9 data (*True Positive*) dan 2 data *search_google* yang diprediksi salah (*False Negative*). Perhitungan akurasi dari model *Dual Intent and Entity Transformer (DIET)* menggunakan persamaan 3.1.

$$\begin{aligned}
 Accuracy &= \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative}} \times 100\% \\
 &= \frac{9 + 9}{9 + 9 + 0 + 2} \times 100\% \\
 &= \frac{18}{20} \times 100\% \\
 &= 90\%
 \end{aligned}$$

Untuk perhitungan nilai precision, recall, dan f1-score digunakan table confusion matrix sebagai visualisasi yang mempermudah dalam proses mencari banyaknya nilai True Positive, True Negative, False Positive, dan False Negative. Berikut Perhitungan nilai precision, recall, dan f1-score dari model *Dual Intent and Entity Transformer (DIET)*

$$\begin{aligned}
 Precision &= \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \times 100\% \\
 &= \frac{9}{9 + 0} \times 100\% \\
 &= \frac{9}{9} \times 100\% \\
 &= 100\%
 \end{aligned}$$

$$\begin{aligned}
 Recall &= \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \times 100\% \\
 &= \frac{9}{9 + 2} \times 100\% \\
 &= \frac{9}{11} \times 100\% \\
 &= 82\%
 \end{aligned}$$

$$\begin{aligned}
 F1 - Score &= \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \times 100\% \\
 &= \frac{2 \times 100\% \times 82\%}{100\% + 82\%} \times 100\% \\
 &= 90\%
 \end{aligned}$$

Melalui nilai performance metrics yang berupa accuracy, precision, recall dan f1-score yang telah dihitung di atas, maka dapat membuktikan bahwa *chatbot* dari *Dual Intent and Entity Transformer (DIET)* mampu melakukan klasifikasi *intent* dengan cukup baik.

4.4. Pengujian Fuzzy String Matching

Penulis menerapkan *fuzzy string matching* yang tersedia sebagai *library* dalam *python* pada penelitian ini. *fuzzy string matching* merupakan *library python* yang berupa *FuzzyWuzzy* dan digunakan dalam menemukan *string* yang sama dengan pola pertanyaan yang diajukan oleh *user* pada *chatbot*. *Library FuzzyWuzzy* menggunakan *edit distance* atau disebut juga *levenshtein distance* dalam menghitung perbedaan antar urutan *string*.

Setelah melalui tahap identifikasi intent dari pertanyaan *user* menggunakan model *Dual Intent and Entity Transformer (DIET)*. Selanjutnya pertanyaan *user* akan dicocokkan dengan pertanyaan yang ada pada data API menggunakan metode *fuzzy string matching*. Dalam *library FuzzyWuzzy* terdapat 4 fungsi yang dapat diterapkan dalam menghitung kesamaan *string*, yaitu ratio, partial ratio, token sort ratio, token set ratio.

1. Ratio

Fungsi ini bertujuan dalam mencocokkan dan menghitung rasio kesamaan jarak dengan algoritma *Levenshtein* standar antara dua urutan kata.

2. Partial Ratio

Fungsi ini bertujuan dalam membantu pencocokkan *string* dalam situasi yang lebih kompleks dalam pencocokkan *substring* ketika pengambilan *string* yang paling pendek dan melakukan pencocokkan *substring* yang mempunyai panjang yang sama.

3. Token Sort Ratio

FuzzyWuzzy memiliki fungsi token yang mentokenisasi *string*, yang berfungsi dalam mengubah huruf besar menjadi huruf kecil, dan melakukan penghapusan tanda baca. Token sort ratio bertujuan untuk mengurutkan *string* berdasarkan dengan abjad dan menggabungkan *string* tersebut. Pada saat *string* yang disamakan terdapat kesamaan berdasarkan ejaan tetapi tidak memiliki kesamaan pada urutan maka fuzz.ratio akan dihitung.

4. Token Set Ratio

Fungsi ini menyerupai token sort ratio, akan tetapi fungsi token set ratio mengeluarkan token umum sebelum melakukan perhitungan fuzz.ratio pada *string* baru. Fungsi ini berguna saat diterapkan pada sekumpulan *string* yang memiliki perbedaan panjang yang signifikan.

Penulis menggunakan fungsi token set ratio dalam melakukan pencocokan *string*. Hasil pengujian pencocokan pertanyaan *user* dengan pertanyaan yang ada di data API dapat dilihat pada Tabel 4.1

Tabel 4.1. Hasil Pengujian Pencocokan *String*

No	Pertanyaan <i>user</i>	Pertanyaan dalam Data API	Persentase Kecocokan
1	Bagaimana cara efektif dalam mengatasi situs yang menyebarkan informasi hoaks?	cara efektif untuk mengatasi situs-situs yang menyebarkan informasi hoaks?	90%
2	Penyebab munculnya situs-situs menyebarkan hoaks	apa penyebab sering munculnya situs-situs yang menyebarkan informasi hoaks?	86%
3	Mengapa hoaks bertema agama dikaitkan dengan politii?	mengapa hoax yang bertema agama selalu dikaitkan dengan politik ?	85%
4	Fenomena masyarakat lihat hoaks	fenomena masyarakat dalam melihat hoax?	100%
5	Siapa oknum yang menyebarkan hoaks?	siapa oknum atau orang-orang yang suka menyebarkan hoax?	100%
6	Apa itu mafindo?	mafindo	90%
7	Trik melawan hoaks	Tips & trik untuk melawan hoaks	100%
8	karakter dari sebuah hoax pada twitter	bagaimana karakter dari sebuah hoax yang disebarakan di twitter?	100%

Tabel 4.1. Hasil Pengujian Pencocokan *String* (lanjutan)

No	Pertanyaan <i>user</i>	Pertanyaan dalam Data API	Persentase Kecocokan
9	apa itu literasi	apa itu literasi digital?	90%
10	tingkat literasi digital di Indonesia	bagaimana tingkat literasi digital di masyarakat Indonesia?	100%
11	misinformasi, disinformasi / hoax	apakah misinformasi, disinformasi / hoax itu sama?	95%
12	penyebab facebook banyak berita hoax	apa penyebab facebook memiliki banyak penyebaran hoax?	100%
13	perbedaan hoax yang tersebar	adakah perbedaan hoax yang tersebar di berbagai media sosial?	90%
14	cara MUI menangkal hoax	bagaimana cara MUI menangkal hoax bertema agama?	90%
15	apa hoaks bisa menyerang media	kenapa hoaks bisa menyerang media-media informasi?	90%
16	apakah banyak berita hoax pandemi yang dikaitkan dengan politik	apakah banyak berita hoax pandemi yang dikaitkan dengan politik?	100%
17	pendapat jurnalis mengenai situs penyebar berita hoaks	ada situs-situs yang menyebarkan berita hoaks, kira-kira apa pendapat jurnalis mengenai situs-situs tersebut?	86%

Tabel 4.1. Hasil Pengujian Pencocokan *String* (lanjutan)

No	Pertanyaan <i>user</i>	Pertanyaan dalam Data API	Persentase Kecocokan
18	bentuk hoaks yang menyerang media	apa saja bentuk hoaks yang menyerang media-media informasi ?	86%
19	cara untuk menangani hoax yang tersebar di media sosial	bagaimana cara untuk menangani hoax yang tersebar di media sosial?	95%
20	Kerugian apa yang dialami oleh media yang diserang hoaks?	Kerugian apa yang dialami oleh media yang diserang oleh hoaks?	96%

4.5. Pengujian Sistem

Hasil pengujian sistem adalah pertanyaan *user* dan respon jawaban *chatbot* menggunakan model *Dual Intent and Entity Transformer (DIET)* dalam mengidentifikasi intent yang sudah dibuat dan *fuzzy string matching* untuk mencocokkan pertanyaan dan jawaban. Pengujian sistem *chatbot* dengan menggunakan *search engine*. Hasil pengujian sistem dapat dilihat pada Tabel 4.2.

Tabel 4.2. Hasil Pengujian Sistem

No	Pertanyaan <i>user</i>	Respon Chatbot	Hasil
1	Bagaimana cara efektif dalam mengatasi situs yang menyebarkan informasi hoaks?	cara efektif yaitu memperbanyak literasi pada media, dan mencari dan membandingkan informasi yang didapatkan ke media massa yang terpecaya lainnya. Karena media massa selalu menyaring informasi yang didapatkan.	Sesuai

Tabel 4.2. Hasil Pengujian Sistem (lanjutan)

No	Pertanyaan <i>user</i>	Respon Chatbot	Hasil
2	Penyebab munculnya situs-situs menyebarkan hoaks	penyebab munculnya situs-situs tersebut, upaya dalam memerikan pemahaman yang salah kepada masyarakat, karena hoaks ini bertujuan untuk memberikan image yang salah kepada masyarakat sehingga dapat menjatuhkan ataupun menyalahkan suatu kelompok ataupun pemerintahan.	Sesuai
3	Mengapa hoaks bertema agama dikaitkan dengan politii?	sebab ada sebagian kalangan yang ingin menjadikan agama sebagai kendaraan politik dalam menyebarkan kebohongan politik menggunakan agama	Sesuai
4	Fenomena masyarakat lihat hoaks	masyarakat merasa hoax perlu dibasmi untuk mencegah hal yang tidak diinginkan terjadi	Sesuai
5	Siapa oknum yang menyebarkan hoaks?	oknum ataupun orang-orang yang menyebarkan hoax adalah orang yang mempercayai informasi tanpa memeriksa terlebih dahulu informasi tersebut	Sesuai
6	karakter dari sebuah hoax pada twitter	karakter dari sebuah hoax yang disebarkan di twitter tidak memiliki perbedaan dengan disinformasi karena kebenaran informasi tersebut salah	Sesuai

Tabel 4.2. Hasil Pengujian Sistem (lanjutan)

No	Pertanyaan <i>user</i>	Respon Chatbot	Hasil
7	apa itu literasi	Literasi digital adalah kemampuan untuk memahami dan menggunakan informasi dalam banyak format dari berbagai sumber	Sesuai
8	tingkat literasi digital di Indonesia	tingkat literasi di masyarakat Indonesia masih sangat mengkhawatirkan sehingga masih diperlukan peningkatan literasi melalui program-program dalam memajukan budaya baca di Indonesia	Sesuai
9	misinformasi, disinformasi / hoax	Disinformasi adalah informasi yang keliru, dan orang yang menyebarkan tahu bahwa itu salah, tetapi tetap menyebarkan. Sedangkan, misinformasi adalah informasi yang keliru, tetapi orang yang menyebarkan percaya bahwa itu benar.	Sesuai
10	penyebab facebook banyak berita hoax	facebook merupakan suatu platform dalam merilis opini-opini pengguna sehingga opini-opini tersebut dapat menimbulkan penyebaran hoax	Sesuai

Tabel 4.2. Hasil Pengujian Sistem (lanjutan)

No	Pertanyaan <i>user</i>	Respon Chatbot	Hasil
11	perbedaan hoax yang tersebar	perbedaan hoax diberbagai media sosial memiliki perbedaan dalam metode penyampaian informasi hoax tersebut	Sesuai
12	cara MUI menangkal hoax	MUI memiliki team dalam mengklarifikasi ataupun memberikan informasi yang lebih valid mengenai berita tersebut	Sesuai
13	apa hoaks bisa menyerang media	alasan hoaks menyerang media, karena media itu memiliki kepercayaan publik yang besar sebagai sumber informas, yang dipercaya dalam menyebarkan informasi-informasi dan berinteraksi dengan masyarakat sehingga penyebar hoaks ini lebih suka mendompleng nama-nama media dalam menyebarkan hoaks mereka, jadi narasi-narasi yang ada di media tersebut diedit agar narasi para penyebar hoaks itu dipercaya oleh masyarakat	Sesuai

Tabel 4.2. Hasil Pengujian Sistem (lanjutan)

No	Pertanyaan <i>user</i>	Respon Chatbot	Hasil
14	pendapat jurnalis mengenai situs penyebar berita hoaks	situs-situs itu sengaja dibuat oleh sekelompok orang di era WFH karena masyarakat tidak bisa keluar dan situs-situs tersebut bersifat negatif bagi masyarakat karena kemungkinan besar tidak dibuat oleh orang yang berkegiatan dalam bidang jurnalistik sebab jurnalistik media memiliki pedoman undang-undang Pers dan diawasi oleh Dewan Pers yang harus dilakukan oleh instant Pers, sehingga jika ada informasi yang menyebabkan hoaks tanpa mencantumkan sumber-sumber yang jelas itu sudah dipastikan bukan dari media-media massa dan hanya merupakan situs-situs yang berusaha untuk mendapatkan sorotan publik ataupun netizen untuk mendapatkan keuntungan.	Sesuai

Tabel 4.2. Hasil Pengujian Sistem (lanjutan)

No	Pertanyaan <i>user</i>	Respon Chatbot	Hasil
15	bentuk hoaks yang menyerang media	bentuk berita hoaks yang sering ditemukan dan menyebar di media sosial itu yaitu perubahan judul, perubahan teks, dan perubahan foto. Judul biasanya menjadi headline yang menjadi fokus ataupun sumber perhatian dari pembaca, penyebar hoaks merubah judul-judul tersebut sesuai dengan keinginan penyebar hoaks. Ada juga yang melakukan editing foto seperti foto-foto yang didapatkan di lapangan tersebut diedit ataupun dicrop dan diubah sesuai dengan narasi penyebar hoaks, biasanya foto di media diberikan caption untuk menekankan peristiwa apa yang terjadi akan tetapi penyebar hoaks merubah narasi tersebut. Sedangkan dalam perubahan naskah berita, penyebar hoaks merubah isi informasi dari naskah berita tersebut seperti memotong informasi ataupun mengambil kutipan-kutipan untuk mendukung narasi dari penyebar hoaks tersebut.	Sesuai

Tabel 4.2. Hasil Pengujian Sistem (lanjutan)

No	Pertanyaan <i>user</i>	Respon Chatbot	Hasil
16	Cek berita perubahan tarif transaksi bank BRI	[HOAKS] Perubahan Biaya Transaksi BRI Rp 150.000 Halaman all ... 14 Mei 2022 ...	Sesuai
17	Cek berita terjadi tsunami di Aceh	Berita Tsunami Aceh Hari Ini - Kabar Terbaru Terkini Liputan6.com Berita Tsunami Aceh - Pada 26 Desember 2004, ...	Sesuai
18	Cek berita gadis ular menggemparkan dunia	[SALAH] Gadis Ular Menggemparkan Dunia – TurnBackHoax.ID 23 Mar 2022 ...	Sesuai
19	Cek berita Perubahan Tarif Transaksi Bank Aceh Kepada Bank Lain	Hati-Hati Penipuan Perubahan Tarif Mengatasnamakan Bank Aceh ... 23 Nov 2022 ...	Sesuai
20	Cek berita Angin Puting beliung dan banjir terjang Cianjur setelah Gempa	[HOAKS] Angin Puting Beliung Pascagempa Cianjur Halaman all ... 1 Des 2022 ...	Sesuai

BAB V

KESIMPULAN DAN SARAN

Bab ini membahas kesimpulan dari penelitian *chatbot* tanya jawab tentang informasi palsu (hoax) menggunakan pendekatan metode *Dual Intent and Entity Transformer (DIET)* dan *fuzzy string matching*.

5.1 Kesimpulan

Dari pengujian sistem *chatbot* yang telah dilakukan, didapatkan kesimpulan sebagai berikut:

1. Tujuan penelitian ini tercapai melalui model *Dual Intent and Entity Transformer (DIET) Classifier* dan *Fuzzy String Matching* yang berhasil melakukan tanya jawab dengan *user* seputar informasi palsu (hoax) dan menjelaskan kebenaran sebuah informasi dengan cukup baik sesuai dengan hasil pengujian sistem yang didapatkan.
2. Pada penelitian model *Dual Intent and Entity Transformer (DIET)* menghasilkan tingkat *accuracy* 0.9 dengan menggunakan 100 *epoch*.
3. Dengan *accuracy* 0.9, model mampu mengidentifikasi pertanyaan dari *user* dengan baik.
4. Hasil pengujian sistem yang dilakukan menunjukkan hasil yang cukup baik dengan total pengujian 25 pertanyaan, 24 jawaban benar, dan 1 diluar topik pembahasan.

5.2 Saran

Saran yang dapat penulis berikan untuk pengembangan penelitian berikutnya yaitu sebagai berikut:

1. Diharapkan pada penelitian selanjutnya dapat melakukan analisis lebih dalam terhadap metode dan melakukan pengolahan data terutama pada data tanya jawab yang memiliki data pertanyaan yang hampir sama sehingga identifikasi yang dilakukan oleh model menjadi lebih baik.

Daftar Pustaka

- Abdellatif, A., Badran, K., Costa, D., & Shihab, E. (2021). A Comparison of Natural Language Understanding Platforms for Chatbots in Software Engineering. *IEEE Transactions on Software Engineering*, 5589(FEBRUARY), 1–18.
<https://doi.org/10.1109/TSE.2021.3078384>
- Abro, W. A., Aicher, A., Rach, N., Ultes, S., Minker, W., & Qi, G. (2022). Natural language understanding for argumentative dialogue systems in the opinion building domain. *Knowledge-Based Systems*, 242, 108318.
<https://doi.org/10.1016/j.knosys.2022.108318>
- Astuti, W., Putri, D. P. I., Wibawa, A. P., Salim, Y., Purnawansyah, & Ghosh, A. (2021). Predicting Frequently Asked Questions (FAQs) on the COVID-19 Chatbot using the DIET Classifier. *2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT)*, 25–29.
<https://doi.org/10.1109/EIConCIT50028.2021.9431913>
- Bocklisch, T., Faulkner, J., Pawlowski, N., & Nichol, A. (2017). *Rasa: Open Source Language Understanding and Dialogue Management*. 1–9.
<http://arxiv.org/abs/1712.05181>
- Bunk, T., Varshneya, D., Vlasov, V., & Nichol, A. (2020). *DIET: Lightweight Language Understanding for Dialogue Systems*. <http://arxiv.org/abs/2004.09936>
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, 1(Mlm)*, 4171–4186.
- Gunson, N., Sieińska, W., Yu, Y., Garcia, D. H., Part, J. L., Dondrup, C., & Lemon, O. (2021). Coronabot: A conversational AI system for tackling misinformation. *GoodIT 2021 - Proceedings of the 2021 Conference on Information Technology for Social Good*, 265–270. <https://doi.org/10.1145/3462203.3475874>
- Henderson, M., Casanueva, I., Mrkšić, N., Su, P. H., Wen, T. H., & Vulić, I. (2020).

- ConveRT: Efficient and accurate conversational representations from transformers. *Findings of the Association for Computational Linguistics Findings of ACL: EMNLP 2020*, 2161–2174. <https://doi.org/10.18653/v1/2020.findings-emnlp.196>
- Jiao, A. (2020). An Intelligent Chatbot System Based on Entity Extraction Using RASA NLU and Neural Network. *Journal of Physics: Conference Series*, 1487(1). <https://doi.org/10.1088/1742-6596/1487/1/012014>
- Juditha, C. (2018). Hoax Communication Interactivity in Social Media and Anticipation (Interaksi Komunikasi Hoax di Media Sosial serta Antisipasinya). *Journal Pekommas*, 3(1), 31. <https://doi.org/10.30818/jpkm.2018.2030104>
- MASTEL. (2019). Hasil Survey Wabah Hoax Nasional 2019. *Website Masyarakat Telematika Indonesia*, 35. <https://mastel.id/hasil-survey-wabah-hoax-nasional-2019/>
- Muangkammuen, P., Intiruk, N., & Saikaew, K. R. (2018). Automated Thai-FAQ chatbot using RNN-LSTM. *2018 22nd International Computer Science and Engineering Conference, ICSEC 2018*, 1–4. <https://doi.org/10.1109/ICSEC.2018.8712781>
- Ngai, E. W. T., Lee, M. C. M., Luo, M., Chan, P. S. L., & Liang, T. (2021). An intelligent knowledge-based chatbot for customer service. *Electronic Commerce Research and Applications*, 50(October), 101098. <https://doi.org/10.1016/j.elerap.2021.101098>
- Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, June 2018*, 1532–1543. <https://doi.org/10.3115/v1/d14-1162>
- Pustejovsky, J., & Stubbs, A. (2013). *Natural Language Annotation for Machine Learning -- A guide to Corpus-building for applications*.
- Rahadi, D. R. (2017). PERILAKU PENGGUNA DAN INFORMASI HOAX DI MEDIA SOSIAL. *JURNAL MANAJEMEN DAN KEWIRAUSAHAAN*, 5(1), 58–70. <https://doi.org/10.26905/jmdk.v5i1.1342>

- Rustamov, S., Bayramova, A., & Alasgarov, E. (2021). Development of Dialogue Management System for Banking Services. *Applied Sciences*, 11(22), 10995. <https://doi.org/10.3390/app112210995>
- Sebastian, A., & George, J. (2016). *Fuzzy Pattern Matching Algorithm for Location Based Approximate Strings*. 7(7), 583–587.
- Zheng, Y., Chen, G., & Huang, M. (2020). Out-of-Domain Detection for Natural Language Understanding in Dialog Systems. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 28(c), 1198–1209. <https://doi.org/10.1109/TASLP.2020.2983593>