

BAB III

METODE PENELITIAN

Penelitian ini berfokus pada perancangan *Chatbot* yang dapat digunakan untuk tanya jawab antara manusia dan robot, perancangan ini hanya sebagai dasar inisiasi untuk *Based-Line Model* pada Website <https://sistel.upi.edu> dan akan dikembangkan lebih lanjut. Dalam perancangan *Chatbot* ini diperlukan skema model atau prototye apa saja yang dibangun nantinya. Serta metodologi penelitian ini berisikan *Research and Development* dan dilakukannya percobaan dengan memomodifikasi fitur-fitur *chatbot* yang ada dan mengukur pengaruhnya terhadap tingkat *user experience*. Berikut pembangunan *Chatbot* antara lain:

3.1 Skema perancangan

Pada penelitian ini, selain melakukan perancangan dalam pengembangan *website* program studi, juga dilakukan perancangan *chatbot* RNN untuk berinteraksi dengan pengguna atau mahasiswa melalui antarmuka pengguna. Dalam perancangan *chatbot* algoritma yang digunakan RNN, langkah-langkah yang akan diambil meliputi pengumpulan dataset percakapan, *preprocessing* data untuk membersihkan dan mempersiapkan teks, pembuatan kamus kata unik, penerapan model RNN untuk pemodelan teks, dan akhirnya, interaksi dengan pengguna melalui antarmuka pengguna yang responsif dan informatif. Proses interaksi dengan pengguna dimulai dengan penggunaan pemahaman bahasa alami (NLU) untuk mengidentifikasi maksud dan tujuan dari percakapan. Setelah itu, manajemen percakapan akan menentukan respons yang tepat berdasarkan analisis hasil NLU/NLP. Respons tersebut dihasilkan melalui penggunaan model RNN atau model bahasa alami lainnya, dan dipresentasikan kembali dalam bahasa alami yang mudah dipahami oleh pengguna melalui antarmuka pengguna.

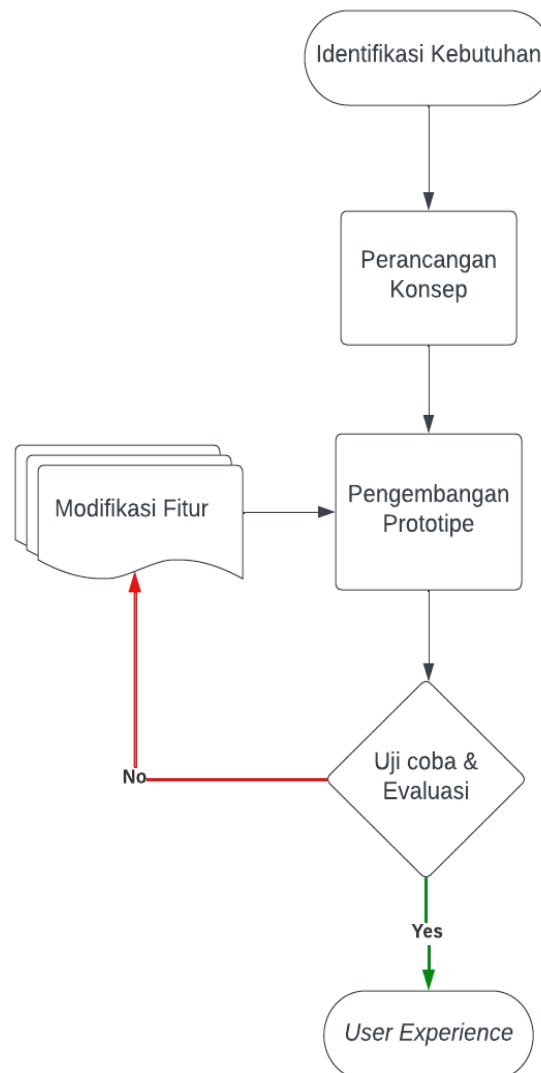
Penelitian ini bertujuan untuk mengimplementasikan algoritma RNN untuk merancang suatu konsep yang lebih mudah dan pemahaman bahasa alami yang lebih baik, sehingga respons yang dihasilkan dapat lebih berguna dalam interaksi dengan pengguna.

3.1.1 Jenis Penelitian

Dalam penelitian ini, pendekatan yang digunakan adalah Research and Development (R&D) untuk mengembangkan *chatbot*. Metode penelitian R&D ini melibatkan serangkaian langkah yang bertujuan untuk merancang, mengembangkan, dan menguji prototipe *chatbot* yang dapat digunakan dalam interaksi tanya jawab antara manusia dan robot. Selain itu, penelitian ini juga melibatkan modifikasi fitur-fitur *chatbot* yang ada dan pengukuran pengaruhnya terhadap tingkat user experience. *Chatbot* yang dikembangkan juga akan menjadi dasar bagi pengembangan lebih lanjut dalam konteks *Website* <https://sistel.upi.edu>.

3.1.2 Tahapan Penelitian

Berikut adalah langkah-langkah yang dilakukan dalam penelitian ini:



Gambar 3. 1 Tahapan penelitian

Mohammad Luthfan Faohan, 2023

PENERAPAN ALGORITMA RECURRENT NEURAL NETWORK DAN TERM FREQUENCY INVERSE DOCUMENT FREQUENCY SEBAGAI PENGEKSTRAKSI FITUR RANCANG BANGUN CHATBOT UNTUK PENGEMBANGAN WEBSITE PRODI SISTEM TELEKOMUNIKASI UPI

Universitas Pendidikan Indonesia | repository.upi.edu | [Perpustakaan.upi.edu](https://perpustakaan.upi.edu)

Pada gambar diagram alir diatas dapat digambarkan dari jenis penelitian dalam skema chatbot dengan penjeleasan:

1. Identifikasi Kebutuhan: mengidentifikasi kebutuhan pengguna dan masalah yang ingin dipecahkan oleh *chatbot*. Hal ini dilakukan melalui survei, wawancara, dan analisis kebutuhan pengguna potensial.
2. Perancangan Konsep: merancang konsep *chatbot* yang sesuai dengan kebutuhan pengguna. Konsep ini meliputi fitur-fitur utama, antarmuka pengguna, dan algoritma yang akan digunakan.
3. Pengembangan Prototipe: mengembangkan prototipe *chatbot* berdasarkan konsep yang telah dirancang. Prototipe ini mencakup implementasi model RNN dan penerapan metode TF-IDF untuk meningkatkan kemampuan *chatbot* dalam memahami dan merespons pertanyaan pengguna.
4. Uji Coba dan Evaluasi: melakukan uji coba terhadap prototipe *chatbot* dengan melibatkan pengguna potensial. Selama uji coba, mengumpulkan umpan balik dari pengguna terkait pengalaman penggunaan *chatbot* dan kualitas respons yang diberikan. Evaluasi dilakukan berdasarkan metrik user experience, seperti kepuasan pengguna, efisiensi interaksi, dan kemudahan penggunaan.
5. Modifikasi Fitur-fitur: Berdasarkan hasil uji coba dan evaluasi, melakukan modifikasi terhadap fitur-fitur *chatbot* yang ada. Modifikasi ini bertujuan untuk meningkatkan kinerja *chatbot* dan pengalaman pengguna.
6. Pengukuran Pengaruh: mengukur pengaruh modifikasi fitur-fitur *chatbot* terhadap tingkat user experience. Hal ini dilakukan dengan membandingkan hasil evaluasi sebelum dan sesudah modifikasi.

Melalui pendekatan R&D dan percobaan yang dilakukan, penelitian ini bertujuan untuk menghasilkan *chatbot* yang lebih efektif dan memenuhi kebutuhan pengguna dalam interaksi tanya jawab dengan robot. Prototipe chatbot

3.2 Perancangan sistem *chatbot*

Sistem *chatbot* ini memiliki arsitektur yang terdiri dari beberapa komponen utama. Pertama, terdapat lapisan input yang menerima masukan teks dari *user*. Selanjutnya, masukan teks tersebut akan melalui tahap pra-pemrosesan, seperti tokenisasi dan normalisasi teks, untuk mempersiapkannya sebelum diproses lebih

lanjut. Setelah itu, masukan teks akan diberikan ke lapisan embedding. Lapisan ini bertanggung jawab untuk mengonversi kata-kata dalam teks menjadi representasi vektor yang lebih padat dan bermakna. Representasi vektor ini membantu *chatbot* dalam memahami konteks dan makna dari kata-kata dalam percakapan. Selanjutnya, terdapat lapisan RNN (*Recurrent Neural Network*) yang memodelkan urutan kata-kata dalam teks dan memperhitungkan konteks sebelumnya dalam percakapan. RNN memungkinkan *chatbot* untuk mempertahankan informasi tentang percakapan sebelumnya dan menggunakan konteks tersebut dalam menghasilkan respons yang lebih baik. Setelah melalui lapisan RNN, hasilnya akan diberikan ke lapisan output.

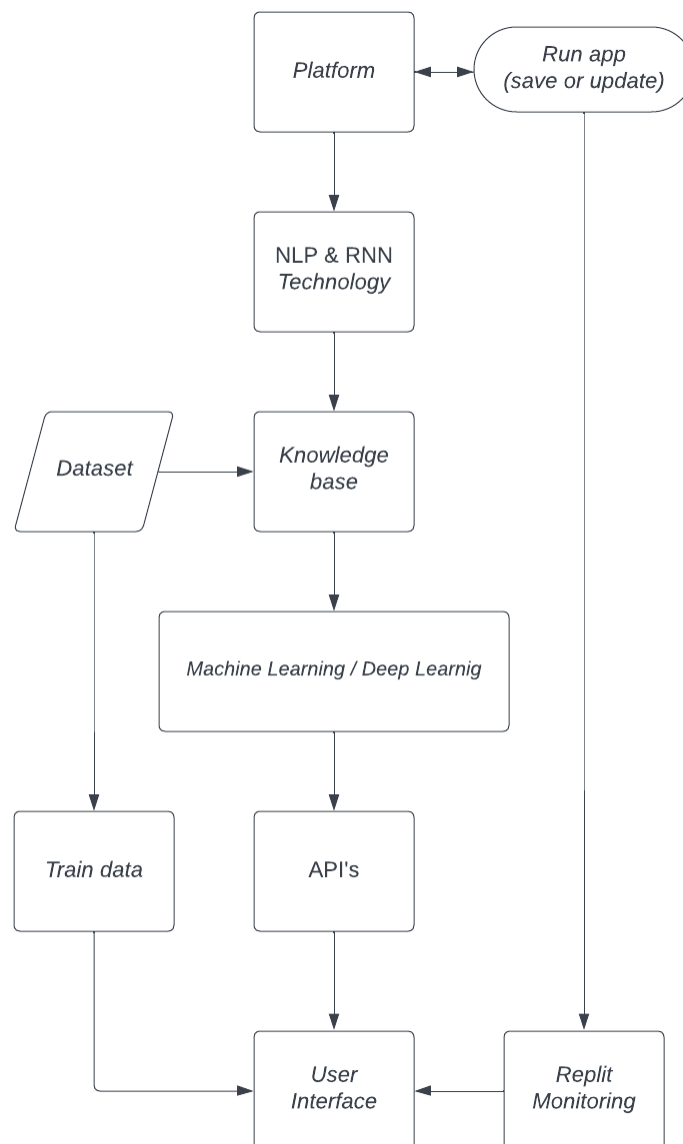
Lapisan ini akan menginterpretasikan hasil dari lapisan RNN untuk menghasilkan respons yang tepat untuk diberikan kepada pengguna. Respons tersebut dapat berupa teks yang memberikan informasi, menjawab pertanyaan, atau memberikan panduan kepada mahasiswa baru. Selain itu, sistem *chatbot* ini juga dapat dianalisis dengan metode TF-IDF (*Term Frequency-Inverse Document Frequency*). Metode ini digunakan untuk menganalisis kata-kata dalam pertanyaan pengguna dan mengidentifikasi kata-kata kunci yang paling relevan dalam menghasilkan respons yang lebih kontekstual dan relevan. Seluruh sistem *chatbot* ini akan dihosting pada platform *website* Prodi, yang dimana mahasiswa baru untuk mengaksesnya dengan mudah melalui *website* resmi Prodi. *Hosting* ini akan memastikan ketersediaan *chatbot* secara *online* sehingga mahasiswa baru dapat mengajukan pertanyaan dan mendapatkan bantuan kapan pun mereka membutuhkannya. *Hosting* ini tidak hanya memberikan aksesibilitas yang luas, tetapi juga memastikan ketersediaan *chatbot* secara online sepanjang waktu. Dengan demikian, mahasiswa baru dapat mengajukan pertanyaan dan mendapatkan bantuan kapan pun mereka membutuhkannya. Hal ini tidak hanya meningkatkan pengalaman pengguna, tetapi juga mendukung mahasiswa baru dalam menjalani transisi mereka ke lingkungan perguruan tinggi dengan dukungan yang lebih baik.

Chatbot adalah program komputer yang dirancang untuk berinteraksi dengan pengguna dan memberikan respons berbasis teks. Berikut perancangan sistem yang diperlukan untuk membuat *chatbot* yang efektif antara lain:

Mohammad Luthfan Faohan, 2023

PENERAPAN ALGORITMA RECURRENT NEURAL NETWORK DAN TERM FREQUENCY INVERSE DOCUMENT FREQUENCY SEBAGAI PENGEKSTRAKSI FITUR RANCANG BANGUN CHATBOT UNTUK PENGEMBANGAN WEBSITE PRODI SISTEM TELEKOMUNIKASI UPI

Universitas Pendidikan Indonesia | repository.upi.edu | Perpustakaan.upi.edu



Gambar 3. 2 Perancangan sistem

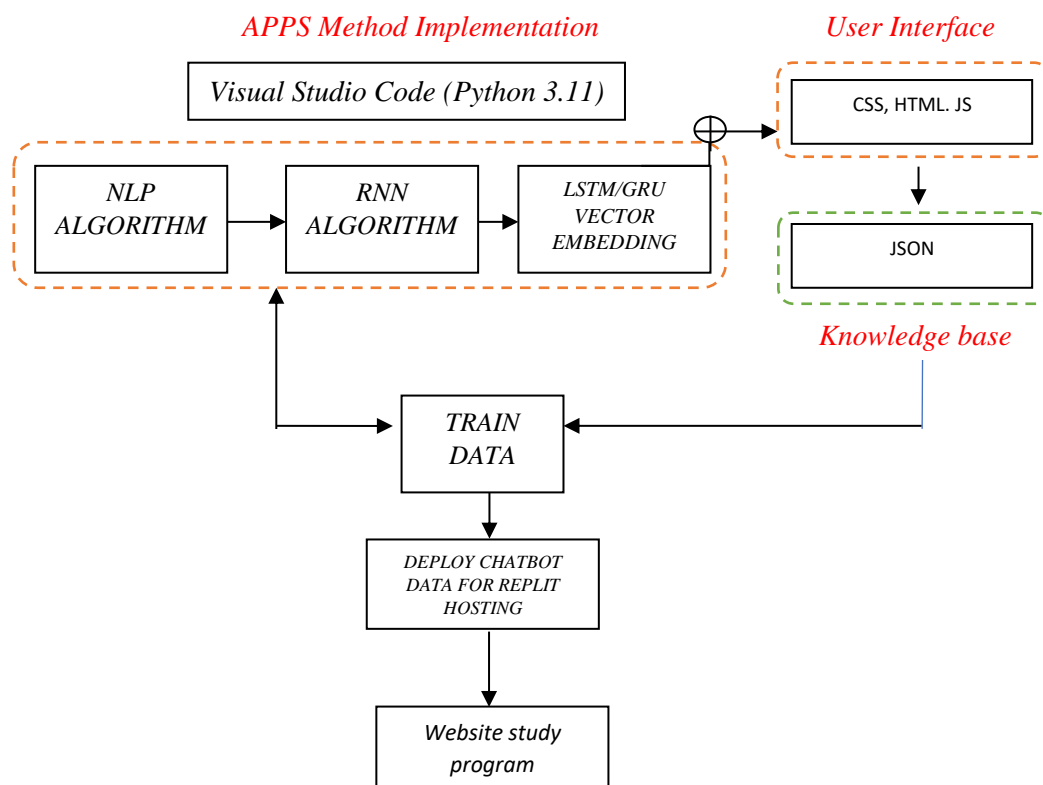
Dapat digambarkan dari perancangan sistem *chatbot* diagram alir diatas adalah:

1. *Platform* atau *Framework*: Memilih platform atau kerangka kerja (*framework*) yang sesuai untuk mengembangkan *chatbot*, seperti Dialogflow, Microsoft Bot *Framework*, atau Rasa. Dan untuk penelitian ini memakai *flask*.
2. *Natural Language Processing* (NLP): Menggunakan teknologi NLP untuk memahami dan memproses bahasa alami pengguna. Ini melibatkan pemrosesan teks, pemecahan kalimat, identifikasi entitas, dan ekstraksi informasi.

3. *Database* atau *Knowledge Base*: *Chatbot* memerlukan basis data yang berisi informasi yang relevan dan diperlukan untuk memberikan jawaban yang tepat. Basis data ini bisa berisi FAQ (pertanyaan yang sering diajukan), pengetahuan domain, dan data lainnya.
4. *Machine Learning* atau *Deep Learning*: Teknik-teknik pembelajaran mesin atau pembelajaran mendalam (deep learning) dapat membantu meningkatkan kemampuan *chatbot* dalam memahami konteks dan memberikan respon yang lebih akurat.
5. APIs atau Integrasi Layanan: Mengintegrasikan *chatbot* dengan API pihak ketiga untuk mengambil informasi dari sistem eksternal atau mengakses layanan tertentu. Misalnya, API penerbangan untuk memberikan informasi penerbangan terbaru.
6. *User Interface* (UI): Memberikan antarmuka yang ramah pengguna untuk berinteraksi dengan *chatbot*, seperti tampilan teks atau suara.
7. Data Pelatihan: *Chatbot* perlu dilatih dengan data pelatihan yang mencakup contoh interaksi pengguna dan respon yang benar. Data ini digunakan untuk mengajari *chatbot* bagaimana merespons berbagai pertanyaan dan masukan.
8. Monitoring dan Analitik: Alat untuk memantau kinerja *chatbot* dan menganalisis interaksi pengguna. Ini membantu dalam memperbaiki dan meningkatkan kualitas *chatbot* seiring waktu.
9. Keamanan: Memastikan keamanan informasi pribadi pengguna dan melindungi *chatbot* dari serangan atau penyalahgunaan.
10. Pemeliharaan dan Pembaruan: Menerapkan strategi pemeliharaan dan pembaruan untuk menjaga kinerja *chatbot* yang optimal seiring berjalannya waktu dan berubahnya kebutuhan pengguna.

Dengan menggunakan peralatan di atas, Sehingga dapat membuat *chatbot* yang cerdas dan responsif, memberikan pengalaman interaktif yang lebih baik kepada pengguna. Setiap poin memberikan pandangan penting tentang aspek-aspek yang harus diperhatikan dalam proses pengembangan *chatbot* yang sukses. Dengan memperhatikan semua elemen ini, Anda dapat merancang dan mengembangkan *chatbot* yang memberikan pengalaman interaktif yang lebih baik kepada pengguna.

3.2.1 Arsitektur *Chatbot*



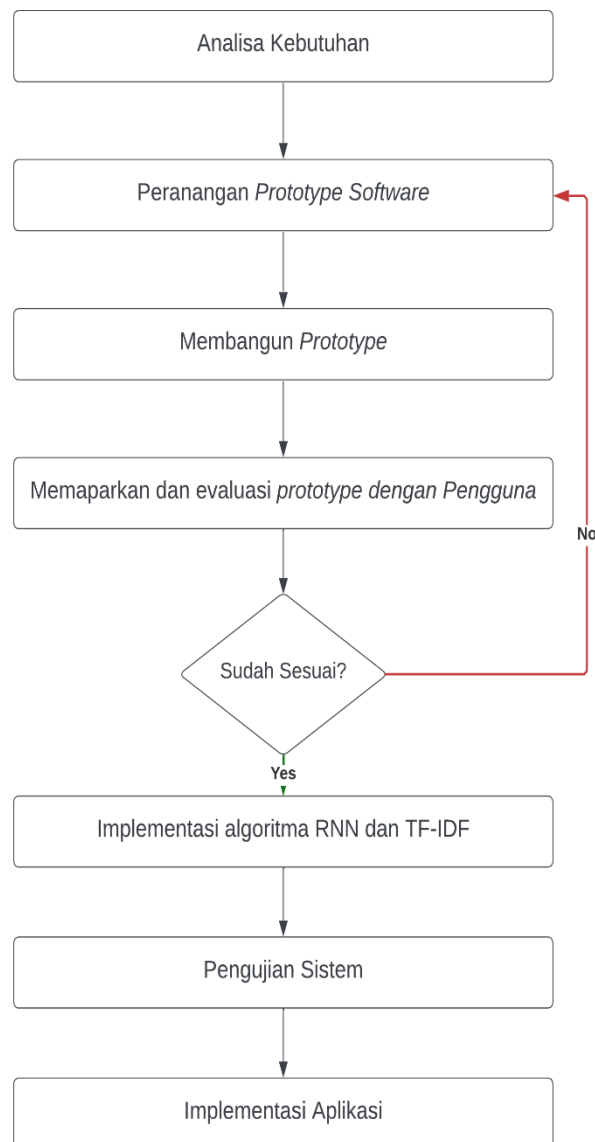
Gambar 3. 3 Arsitektur *chatbot*

Arsitektur *chatbot* pada gambar 3.3 merupakan Arsitektur *chatbot* mengacu pada struktur dan rancangan sistem yang digunakan untuk mengimplementasikan dan menjalankan *chatbot*. Arsitektur *chatbot* mencakup komponen-komponen dan alur kerja yang memungkinkan *chatbot* berfungsi dengan baik untuk di implementasikan pada *website study program* atau *website* <https://sistel.upi.edu>. Dimulai dari *apps methods implementation* mencakup 3 faktor penting *chatbot* yang akan dibuat yaitu NLP, RNN Algoritma, serta pemecahan kata serta model RNN nantinya berupa LSTM/GRU *Vector Embedding*. Masuk ke *user interface* untuk pemuatan *design* nantinya seperti apa, lalu beralih kepada *dataset* atau *knowledge base chatbot* itu sendiri, Latih data dalam *VSCODE* atau *google colab*, *deploy* ke *replit.com* dan implementasikan *chatbot* ke *website* <https://sistel.upi.edu>.

3.2.2 *Prototype Software*

Prototype perangkat lunak adalah versi awal atau rancangan sederhana dari perangkat lunak yang akan dikembangkan. Tujuannya adalah untuk memperlihatkan secara kasar dan cepat bagaimana perangkat lunak tersebut akan

berfungsi dan berinteraksi dengan pengguna. *Prototype* perangkat lunak ini membantu dalam proses pengembangan karena dapat memberikan gambaran lebih jelas tentang fitur-fitur yang akan ada, antarmuka pengguna, dan alur kerja dari perangkat lunak yang akan dibuat. Berikut adalah gambar diagram alir *prototyping software* pada *chatbot*:



Gambar 3. 4 Diagram alur sistem

Deskripsi tentang langkah-langkah dalam merancang dan mengimplementasikan *chatbot* menggunakan RNN (*Recurrent Neural Network*) dan TF-IDF (*Term Frequency-Inverse Document Frequency*).

Berikut adalah langkah-langkah untuk menggambarkan proses tersebut:

Mohammad Luthfan Faohan, 2023

PENERAPAN ALGORITMA RECURRENT NEURAL NETWORK DAN TERM FREQUENCY INVERSE DOCUMENT FREQUENCY SEBAGAI PENGEKSTRAKSI FITUR RANCANG BANGUN CHATBOT UNTUK PENGEMBANGAN WEBSITE PRODI SISTEM TELEKOMUNIKASI UPI

Universitas Pendidikan Indonesia | repository.upi.edu | Perpustakaan.upi.edu

1. **Analisa Kebutuhan:** Lakukan analisis mendalam tentang kebutuhan sistem *chatbot*. Identifikasi fungsi-fungsi dan fitur-fitur yang diharapkan dari *chatbot*, serta tujuan interaksi dengan pengguna.
2. **Perancangan Prototype System:**
Buatlah perancangan sistem *chatbot* secara keseluruhan, termasuk antarmuka pengguna, aliran percakapan, dan penggunaan RNN dan TF-IDF untuk memproses teks.
3. **Membangun Prototype:**
Implementasikan perancangan sistem menjadi prototype *chatbot*. Buatlah model RNN untuk memahami dan menghasilkan respons teks. Gunakan juga TF-IDF untuk melakukan analisis teks dan mendapatkan bobot kata-kata.
4. **Memaparkan dan Evaluasi Prototype dengan Pengguna:**
Lakukan presentasi prototype *chatbot* kepada pengguna atau kelompok pengguna potensial. Dapatkan umpan balik dan evaluasi performa chatbot untuk memahami kekuatan dan kelemahan sistem.
5. **Implementasi Algoritma RNN dan TF-IDF:**
Implementasikan algoritma RNN dan TF-IDF ke dalam sistem *chatbot*. Pastikan integrasi berjalan dengan baik dan mampu memahami maksud pengguna serta memberikan respons yang relevan.
6. **Pengujian Sistem:** Lakukan pengujian sistem secara menyeluruh untuk memastikan *chatbot* berfungsi dengan baik dalam berbagai skenario dan situasi. Uji coba kemampuan *chatbot* dalam memahami bahasa alami dan memberikan respons yang akurat.
7. **Implementasi Aplikasi:** Setelah sistem *chatbot* berhasil diuji dan dievaluasi, lakukan implementasi aplikasi *chatbot* secara keseluruhan. Siapkan *chatbot* untuk digunakan oleh pengguna sesuai kebutuhan dan platform yang telah ditentukan.

3.3 Pengembangan model *chatbot*

Pengembangan model *chatbot* adalah proses merancang, membangun, dan melatih model yang dapat memungkinkan *chatbot* untuk berinteraksi dengan pengguna secara otomatis melalui teks. Proses ini melibatkan beberapa tahap yang

mencakup pemahaman bahasa alami, pembuatan model berbasis machine learning, dan pengujian serta peningkatan performa *chatbot*.

3.3.1 Diagram alur pengembangan model *chatbot*

Dalam pengembangan model *chatbot*, peneliti menggunakan bantuan *software* Python 3.11 dengan langkah yang dilakukan dalam penelitian pengguna menjadi huruf kecil. Dalam hal ini dilakukan agar sesuai bentuk kata kunci yang dibuat seperti:

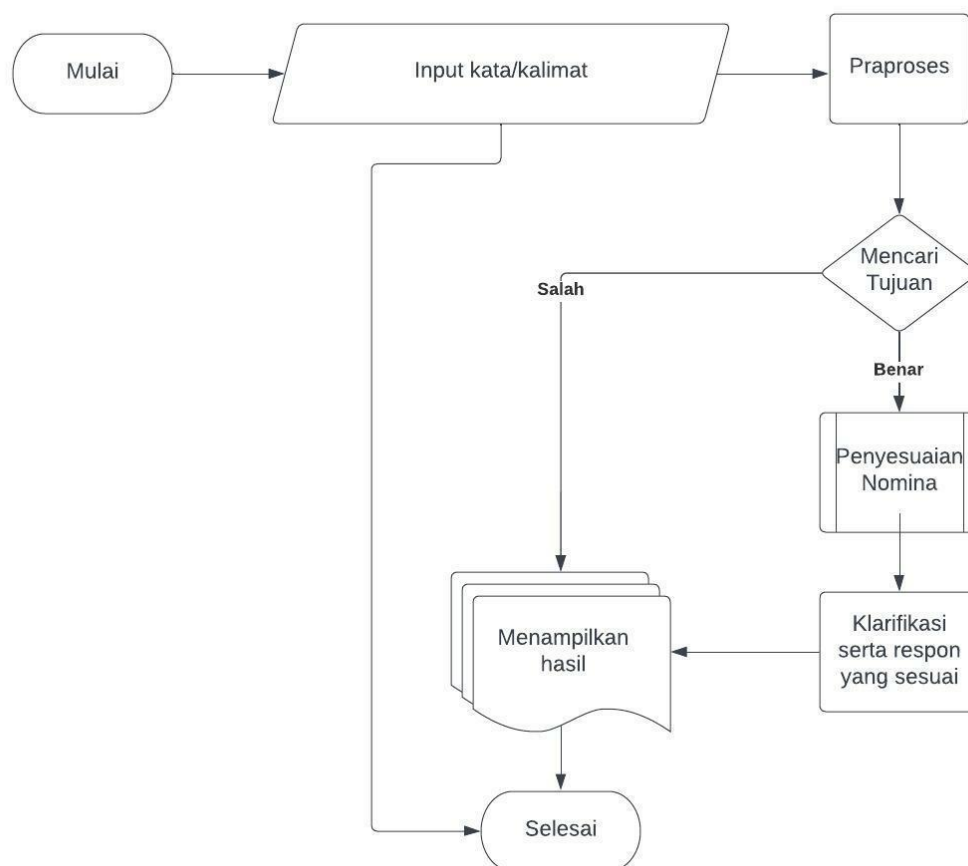
- a. Input kata/kalimat yang akan ditanyakan dari *user*.
- b. Pembuatan skenario interaksi di praproses terlebih dahulu yang akan dilakukan oleh *chatbot*, termasuk pertanyaan dan jawaban yang akan diberikan oleh *chatbot*.
- c. Implementasikan skenario interaksi yang telah dibuat ke dalam *chatbot* dengan menggunakan bahasa pemrograman atau *framework* yang sesuai dengan algoritma *Recurrent Neural Network*.
- d. Uji coba *chatbot* dengan menggunakan *data* test untuk memastikan bahwa *chatbot* dapat berinteraksi dengan benar sesuai dengan skenario yang telah dibuat.
- e. Setelah melalui uji coba dan perbaikan yang diperlukan, lakukan peluncuran *chatbot* ke pasar atau ke pengguna yang telah ditentukan.

Lalu, evaluasi kinerja *chatbot* setelah peluncuran, termasuk tingkat kepuasan *user* dan tingkat akurasi jawaban yang diberikan oleh *chatbot*. *Chatbot* membutuhkan runtutan yang pasti dalam pembuatan agar tidak salah berikut merupakan diagram *chatbot* dalam keseluruhan yang dibuat pada penelitian ini sehingga memberikan respon yang sesuai dengan “penyesuaian nomina”. Selain evaluasi kuantitatif, visualisasi dalam bentuk diagram dapat membantu dalam memahami secara lebih intuitif bagaimana keseluruhan sistem *chatbot* bekerja. Diagram ini dapat menggambarkan alur interaksi antara pengguna dan *chatbot*, melibatkan komponen-komponen seperti input teks, lapisan RNN, lapisan Softmax, integrasi TF-IDF, dan respons yang dihasilkan. Diagram ini akan memberikan gambaran yang jelas tentang bagaimana data mengalir melalui sistem dan bagaimana respons dihasilkan seperti dibawah ini:

Mohammad Luthfan Faohan, 2023

PENERAPAN ALGORITMA RECURRENT NEURAL NETWORK DAN TERM FREQUENCY INVERSE DOCUMENT FREQUENCY SEBAGAI PENGEKSTRAKSI FITUR RANCANG BANGUN CHATBOT UNTUK PENGEMBANGAN WEBSITE PRODI SISTEM TELEKOMUNIKASI UPI

Universitas Pendidikan Indonesia | repository.upi.edu | Perpustakaan.upi.edu



Gambar 3. 5 Flowchart chatbot

Pada diagram diatas dapat dideksripsikan dan diklarifikasi dari *chatbot model* tersebut dengan:

a. Input Kalimat

Kalimat yang digunakan untuk melakukan interaksi *chatbot* ini menggunakan kalimat yang EYD. Dengan setiap kalimat akan diproses lebih lanjut untuk mendapatkan hasil secara maksimal dengan melakukan *preprocessing* dengan mendapatkan jawaban yang sesuai dari pertanyaan yang telah diajukan.

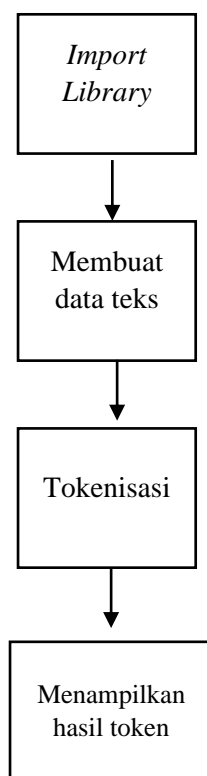
b. Preprocessing

Preprocessing adalah tahap pengolahan data yang dilakukan sebelum data dianalisis atau digunakan dalam suatu proses. Ini biasanya dilakukan untuk mempersiapkan data sehingga sesuai dengan kebutuhan proses yang akan datang. Pada pembuatan *chatbot*, preprocessing data dapat dilakukan untuk menyiapkan data yang akan digunakan dalam pelatihan model jaringan saraf tiruan (neural network). *Preprocessing data* mungkin termasuk membersihkan data, menyajikan

ulang data ke dalam format yang lebih mudah diproses, normalisasi data, dan mengurangi dimensi data. Ini penting untuk memproses data dengan benar karena data yang tidak terproses dengan benar dapat menyebabkan hasil yang tidak akurat dari model neural network yang digunakan.

c. *Tokenizing*

Tokenizing adalah proses memecah teks menjadi bagian-bagian yang lebih kecil, biasanya kata-kata atau karakter-karakter. *Tokenizing* biasanya dilakukan sebagai langkah awal dalam *Preprocessing* pada *chatbot* menggunakan algoritma *Recurrent Neural Network (RNN)*, karena dengan memecah teks menjadi token-token yang lebih kecil, dapat memudahkan proses pemodelan data teks.



Gambar 3. 6 *Tokenizing Chatbot* menggunakan RNN

Dalam contoh sederhana *tokenizing* pada *Chatbot* menggunakan RNN dengan contoh tersebut adalah data teks di-load ke dalam variabel “*teks*”, kemudian di-tokenisasi menjadi kata-kata menggunakan “*tokenizer*” dari library “*keras*”. Hasil tokenisasi kemudian ditampilkan untuk dapat diamati. Dan dari penyesuaian *syntax* tersebut sesuai dengan kebutuhan dan tipe data yang akan ditokenisasi.

d. Lemmatizing

Teknik *preprocessing* pada *Chatbot* menggunakan algoritma *Recurrent Neural Network* dengan bertujuan untuk mengurangi variasi dari kata yang sama dengan mengubah kata ke dalam bentuk dasar (*Lemma*) nya. Dengan kata “bermain” akan diubah menjadi “main”, kata “membuat” akan diubah menjadi kata “buat”, dan sebagainya. *Lemmatizing* dapat membantu meningkatkan efisiensi dan akurasi dari model RNN dengan mengolah teks. *Lemmatizing* dilakukan setelah proses *tokenizing* yaitu dengan memecah teks menjadi kata-kata atau karakter yang memerlukan proses model *Natural Language Processing* yang biasa mengenali *Part of speech* dari kata-kata dalam teks. Beberapa library NLP digunakan untuk melakukan *Lemmatizing* , seperti NLTK, Spacy, Gensim. Berikut adalah contoh *Lemmatizing* :

Teks asli: saya sedang berjalan ke pasar.

Teks lemmatized: saya sedang jalan ke pasar.

Teks asli: Anak-anak sedang bermain di taman.

Teks lemmatized: anak-anak sedang main di taman

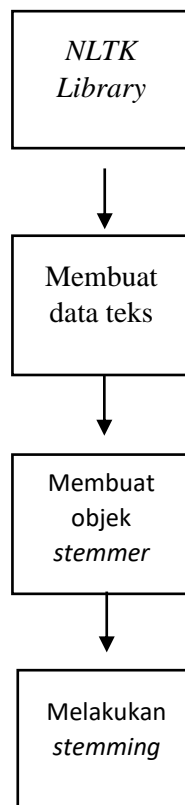
Teks asli: dia memakan ayam goreng di warung.

Teks lemmatized: dia makan ayam goreng di warung.

Diatas adalah contoh hasil *Lemmatizing* pada *Chatbot* menggunakan RNN. Pada contoh kata-kata yang terlihat sebagai bentuk jamat atau terikat konjugasi diubah menjadi bentuk dasarnya. Sebagai contoh, “berjalan” diubah menjadi “jalan”, “memakan” diubah menjadi “makan”, dan sebagainya.

e. Stemming

Stemming merupakan teknik *preprocessing* pada *chatbot* menggunakan algoritma *Recurrent Neural Network* yang bertujuan untuk mengurangi variasi dari kata yang sama dengan cara untuk mengubah kata ke dalam bentuk dasar yaitu “*stem*”. *Stemming* dilakukan untuk menghapus akhiran kata, seperti -s, -ed, atau -ing. *Stemming* dapat membantu meningkatkan efisiensi dan akurasi pada model RNN dari pengolahan teks. *Stemming* merupakan proses *tokenizing*, yaitu memecah *teks* menjadi kata -kata dengan beberapa library seperti NLTK, PorterStemmer, dll.



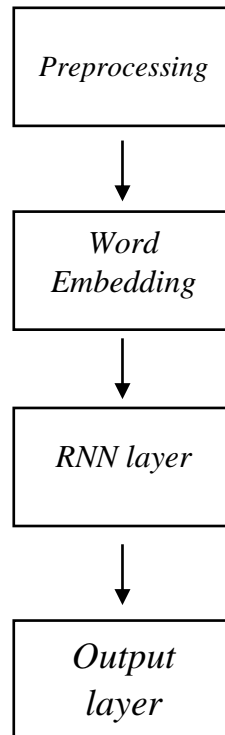
Gambar 3. 7 *Stemming*

Dengan menggunakan pustaka NLTK (*Natural Language Toolkit*) yang mencakup alat-alat untuk pemrosesan bahasa alami. Dengan mengimpor *PorterStemmer*, yang merupakan salah satu algoritma *stemming* yang cukup umum digunakan. Anda menerapkan *stemming* pada kumpulan data kata dalam variabel *text_data*. Hasil keluaran dari kode di atas akan menunjukkan kata-kata setelah diproses dengan algoritma stemming.

Contohnya, kata-kata seperti "*running*", "*runner*", "*runs*", dan "*ran*" akan disederhanakan menjadi "*run*", dan kata-kata seperti "*easily*" akan tetap sama karena bukan bentuk kata benda atau kata kerja yang diakhiri dengan akar kata umum. Namun, penting untuk dicatat bahwa algoritma stemming mungkin tidak selalu menghasilkan hasil yang sempurna. Terkadang, kata-kata yang secara morfologis terkait tetapi memiliki makna yang berbeda bisa diubah menjadi bentuk dasar yang sama, atau sebaliknya.

3.3.2 Diagram alur implementasi algoritma RNN

Berikut adalah contoh diagram alur pembuatan implementasi RNN pada *chatbot*:



Gambar 3. 8 Implementasi algoritma RNN

Diagram ini menggambarkan alur pembuatan implementasi RNN pada *chatbot*. Tahap-tahap yang terlibat dalam implementasi tersebut adalah sebagai berikut:

1. *Preprocessing*:

Tahap ini melibatkan pra-pemrosesan teks input seperti tokenisasi, normalisasi teks, dan penghapusan tanda baca. *Preprocessing* membantu mempersiapkan teks input sebelum diolah lebih lanjut oleh model RNN.

2. *Word Embedding*:

Setelah *preprocessing*, teks input akan diubah menjadi representasi vektor menggunakan teknik *word embedding*. *Word embedding* membantu mewakili kata-kata dalam teks input menjadi representasi numerik yang bermakna.

3. RNN Layer:

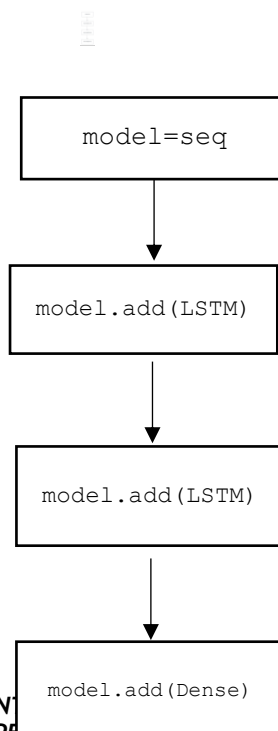
Representasi vektor hasil *word embedding* akan diberikan ke lapisan RNN (*Recurrent Neural Network*). Lapisan RNN akan memodelkan urutan kata-kata dalam teks input dan mempertahankan informasi kontekstual dari percakapan sebelumnya.

4. Output Layer:

Hasil dari lapisan RNN akan diinterpretasikan oleh lapisan *output* untuk menghasilkan respons yang sesuai. Lapisan *output* mengambil hasil dari lapisan RNN dan menghasilkan respons teks yang akan dikirimkan kembali kepada pengguna.

Diagram tersebut memberikan gambaran visual tentang alur pembuatan implementasi RNN pada *chatbot*, dengan setiap tahap mengambil input dari tahap sebelumnya dan memberikan output ke tahap selanjutnya.

Berikut *output* yang dapat dibuat dari RNN algoritma pada *chatbot* yang dibuat:



Gambar 3. 9 Model RNN

Model source code di atas merupakan implementasi dari sebuah model *Chatbot* menggunakan RNN (*Recurrent Neural Network*) dengan menggunakan Keras API.

Penjelasan *output* di atas:

1. Pertama, mendefinisikan model *Chatbot* menggunakan `Sequential()`, yang adalah model linear stack untuk rangkaian layer.
2. Selanjutnya, menambahkan *layer Embedding*, yang digunakan untuk mengubah representasi kata menjadi vektor numerik dengan panjang 10. *vocabulary* adalah ukuran kosakata atau jumlah kata yang unik dalam dataset, dan *input_length* adalah panjang dari input teks atau sekuens kata.
3. Kemudian, menambahkan *layer LSTM (Long Short-Term Memory)*, yang merupakan tipe dari RNN. *Layer LSTM* memiliki 10 *unit* atau *neuron*, dan *return_sequences=True* menunjukkan bahwa ingin mendapatkan *output* dari setiap langkah waktu (*time step*) dalam sekuens *input*.
4. Setelah itu, menambahkan *layer Flatten*, yang digunakan untuk meratakan *output* dari layer LSTM menjadi satu dimensi. Hal ini diperlukan agar *output* dapat digunakan sebagai input untuk *layer dense* berikutnya.
5. Terakhir, menambahkan *layer dense* dengan fungsi aktivasi *softmax*. Layer ini digunakan untuk menghasilkan *output* akhir, dengan panjang sebanding dengan jumlah kata yang ingin diprediksi dalam keluaran *chatbot*. Fungsi aktivasi *softmax* digunakan untuk menghasilkan distribusi probabilitas untuk setiap kata dalam keluaran *chatbot*, sehingga dapat memilih kata dengan probabilitas tertinggi sebagai respons dari *chatbot*.

Ini adalah pengaturan umum untuk arsitektur *chatbot* yang menggunakan RNN dan LSTM. Namun, dalam implementasi nyata, perlu mempertimbangkan hal-hal seperti ukuran lapisan, jumlah unit LSTM, fungsi aktivasi yang sesuai, penggunaan *dropout* untuk mencegah *overfitting*, dan lain-lain. *Flatten Layer* sebelumnya tidak

Mohammad Luthfan Faohan, 2023

PENERAPAN ALGORITMA RECURRENT NEURAL NETWORK DAN TERM FREQUENCY INVERSE DOCUMENT FREQUENCY SEBAGAI PENGEKSTRAKSI FITUR RANCANG BANGUN CHATBOT UNTUK PENGEMBANGAN WEBSITE PRODI SISTEM TELEKOMUNIKASI UPI

Universitas Pendidikan Indonesia | repository.upi.edu | Perpustakaan.upi.edu

diperlukan dalam arsitektur *chatbot* karena berurusan dengan data berurutan (seperti teks) daripada data spasial yang perlu dilipat.

Dengan demikian, model *Chatbot* menggunakan RNN tersebut memiliki beberapa layer, termasuk *Embedding*, LSTM, *flatten*, dan *dense* dengan fungsi aktivasi *softmax*, yang secara bersama-sama membentuk arsitektur yang dapat digunakan untuk menghasilkan respons dari teks input yang diberikan oleh pengguna. Arsitektur yang lebih umum untuk model *chatbot* menggunakan RNN (*Recurrent Neural Network*) atau variasinya seperti LSTM atau GRU (*Gated Recurrent Unit*). *Embedding Layer*: Mengonversi kata-kata menjadi vektor yang lebih padat. LSTM (atau GRU) Layer: Memproses urutan kata-kata dan memahami konteks teks. *Dense Layer* dengan *Softmax Activation*: Menghasilkan distribusi probabilitas untuk kata-kata atau tindakan respons yang mungkin.

Biasanya, model *chatbot* menggunakan arsitektur berbasis RNN (*Recurrent Neural Network*) atau variasinya seperti LSTM (*Long Short-Term Memory*) atau GRU (*Gated Recurrent Unit*). Arsitektur ini dirancang untuk memproses urutan kata-kata dalam teks input dan memahami konteksnya.

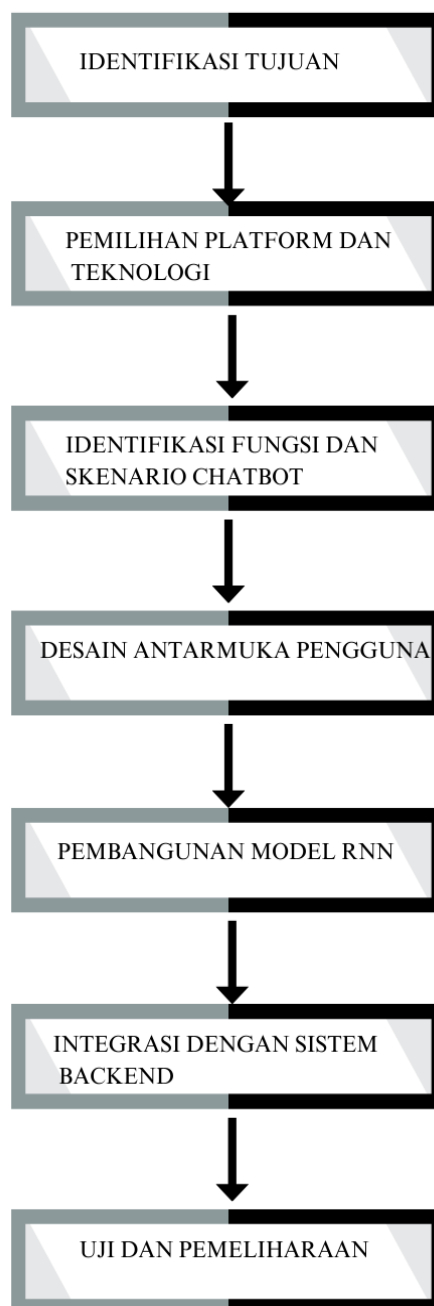
Proses dimulai dengan lapisan *Embedding* yang bertugas mengonversi kata-kata dalam teks menjadi vektor yang lebih padat dan kontinu. Ini membantu dalam merepresentasikan makna kata secara lebih baik dalam ruang vektor. Selanjutnya, lapisan LSTM (atau GRU) berperan penting dalam memproses urutan kata-kata. Lapisan ini memungkinkan model untuk memahami ketergantungan konteks antara kata-kata dan mengenali pola dalam urutan tersebut. Hal ini membantu model untuk merespons dengan lebih kontekstual dan sesuai.

Secara keseluruhan, arsitektur ini menggabungkan komponen-komponen yang berbeda untuk menghasilkan respons yang lebih cerdas dan kontekstual dalam interaksi antara pengguna dan *chatbot*. Terakhir, lapisan *dense* dengan aktivasi *softmax* digunakan untuk menghasilkan distribusi probabilitas. Dalam konteks *chatbot*, lapisan ini menghasilkan distribusi probabilitas untuk kata-kata atau tindakan yang mungkin sebagai respons. Dengan demikian, model dapat memilih respons yang paling sesuai berdasarkan probabilitas yang dihasilkan.

Penting untuk diingat bahwa pengembangan model *chatbot* melibatkan banyak eksperimen dan penyesuaian berdasarkan jenis tugas dan data yang

digunakan. Konfigurasi lapisan dan parameter dapat bervariasi tergantung pada kebutuhan. Di bawah ini adalah contoh arsitektur yang lebih sesuai untuk model *chatbot*:

3.3.3 Diagram alur pembuatan *design web chatbot*



Gambar 3. 10 *Flowchart design chatbot*

Mohammad Luthfan Faohan, 2023

PENERAPAN ALGORITMA RECURRENT NEURAL NETWORK DAN TERM FREQUENCY INVERSE DOCUMENT FREQUENCY SEBAGAI PENGEKSTRAKSI FITUR RANCANG BANGUN CHATBOT UNTUK PENGEMBANGAN WEBSITE PRODI SISTEM TELEKOMUNIKASI UPI

Universitas Pendidikan Indonesia | repository.upi.edu | Perpustakaan.upi.edu

Berikut adalah gambar 3.10 yaitu diagram alur pembuatan desain *chatbot*:

1. Identifikasi tujuan *Chatbot*:

Tentukan tujuan utama dari *chatbot*, seperti memberikan informasi, menjawab pertanyaan, atau memberikan bantuan kepada pengguna baru. Definisikan dengan jelas apa yang ingin dicapai oleh *chatbot*.

2. Pemilihan *Platform* dan Teknologi:

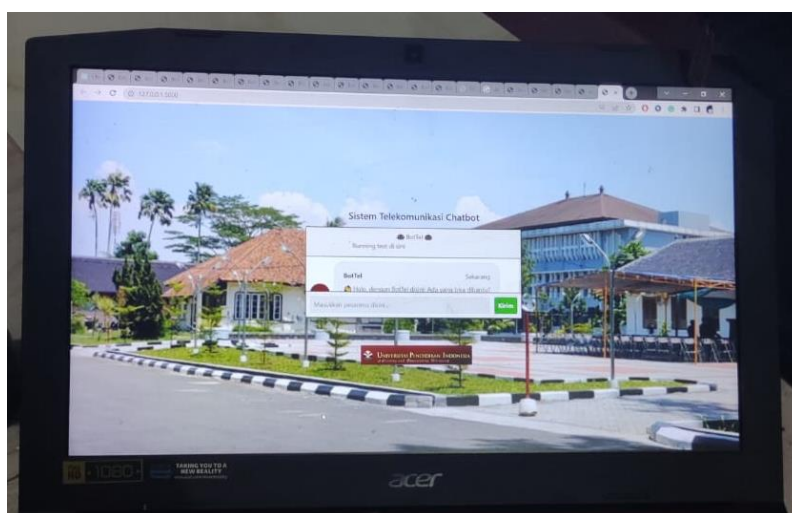
Pilih platform yang sesuai untuk mengembangkan *chatbot*, misalnya menggunakan Python dan *framework* seperti *TensorFlow* atau *PyTorch* untuk implementasi RNN. Pertimbangkan pilihan teknologi seperti *Natural Language Processing* (NLP) dan *Machine Learning* untuk meningkatkan kemampuan *chatbot* dalam memahami dan merespons teks pengguna.

3. Identifikasi fungsi dan skenario *Chatbot*:

Tentukan fungsi-fungsi utama yang akan diberikan oleh *chatbot*, misalnya menyediakan informasi tentang program studi, jadwal kuliah, atau proses pendaftaran. Identifikasi skenario atau percakapan umum yang dapat terjadi antara pengguna dan *chatbot* untuk setiap fungsi.

4. Desain antarmuka pengguna:

Buat desain antarmuka pengguna yang intuitif dan mudah digunakan. Pertimbangkan penggunaan elemen-elemen seperti input teks, tombol, atau menu *dropdown* untuk interaksi dengan *chatbot*.



Gambar 3. 11 *Design web chatbot* awal

Mohammad Luthfan Faohan, 2023

PENERAPAN ALGORITMA RECURRENT NEURAL NETWORK DAN TERM FREQUENCY INVERSE DOCUMENT FREQUENCY SEBAGAI PENGEKSTRAKSI FITUR RANCANG BANGUN CHATBOT UNTUK PENGEMBANGAN WEBSITE PRODI SISTEM TELEKOMUNIKASI UPI

Universitas Pendidikan Indonesia | repository.upi.edu | Perpustakaan.upi.edu

5. Pembangunan model RNN:

Lakukan pembangunan model RNN menggunakan library atau *framework* yang telah dipilih. Latih model menggunakan dataset yang relevan, termasuk pertanyaan dan respons yang telah dipreproses sebelumnya.

6. Integrasi dengan sistem *backend*:

Buat koneksi antara *chatbot* dengan sistem backend yang menyediakan informasi atau layanan yang dibutuhkan. Pastikan *chatbot* dapat mengambil data dari sistem backend secara *real-time* untuk memberikan respons yang akurat.

7. Uji dan pemeliharaan:

Uji *chatbot* secara menyeluruh untuk memastikan kinerja yang baik dalam merespons pertanyaan dan memberikan layanan yang diinginkan. Perbarui dan perbaiki *chatbot* secara berkala berdasarkan umpan balik dan evaluasi pengguna. Perlu dicatat bahwa desain *chatbot* dapat disesuaikan dengan kebutuhan dan preferensi spesifik yang dimiliki.

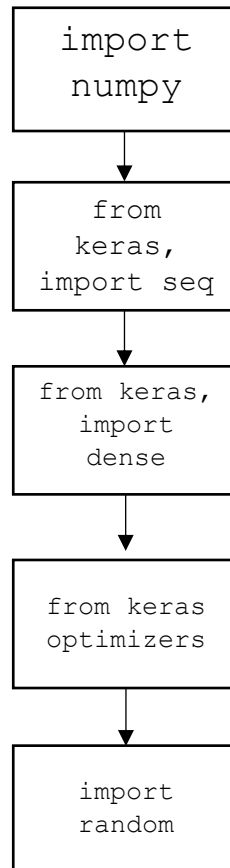
3.3.4 Pengembangan RNN

Recurrent Neural Network (RNN) sebagai bagian dari pendekatan kuantitatif dalam R&D *chatbot* akan dilakukan dengan mempertimbangkan metode pengambilan sampel data teks. Dalam penelitian ini, pengambilan sampel data teks dilakukan dengan cermat untuk mencakup subjek-subjek yang relevan dan representatif. Populasi seluruh objek yang akan diteliti, dalam hal ini data teks yang relevan dengan konteks *chatbot*, akan diidentifikasi dan digunakan sebagai dataset untuk melatih model RNN. Kemudian, sampel data yang representatif akan dipilih dari dataset tersebut untuk proses pelatihan model RNN. Dengan demikian, pendekatan kuantitatif R&D dalam penelitian ini akan memastikan bahwa *Chatbot* yang dikembangkan menggunakan algoritma RNN memiliki dasar data yang kuat dan mewakili untuk dapat memberikan respons yang akurat dan relevan terhadap masukan teks dari pengguna.

Pada penelitian mengenai *Chatbot* menggunakan algoritma *Recurrent Neural Network* dengan populasi dari seluruh objek yang akan diteliti dalam penelitian dengan sampelnya menunjukkan subjek yang dimana akan digunakan untuk penelitian. *Chatbot* dengan menggunakan RNN, dapat di populasikan menggunakan seluruh data yang diidentifikasi sebagai seluruh data yang akan

digunakan dalam pelatihan model RNN. Dari sampel yang digunakan dalam pelatihan model RNN.

Contoh pemilihan sampel data teks pada *Chatbot* menggunakan algoritma *Recurrent Neural Network* menggunakan Python:



Gambar 3. 12 *Sample library chatbot* menggunakan RNN

Library tersebut merupakan contoh dari pustaka Keras yang digunakan untuk membangun model *neural network* untuk tugas-tugas pembelajaran mesin. Berikut adalah penjelasan singkat mengenai setiap baris kode:

1. ``import numpy as np``: Mengimpor pustaka NumPy dan memberikan alias ``np``. NumPy adalah pustaka yang digunakan untuk manipulasi data dalam bentuk array dan matriks.
2. ``from keras.models import Sequential``: Mengimpor class ``Sequential`` dari pustaka Keras. ``Sequential`` adalah model *neural network* yang terdiri dari serangkaian lapisan (*layer*) yang tersusun secara berurutan.

Mohammad Luthfan Faohan, 2023

PENERAPAN ALGORITMA RECURRENT NEURAL NETWORK DAN TERM FREQUENCY INVERSE DOCUMENT FREQUENCY SEBAGAI PENGEKSTRAKSI FITUR RANCANG BANGUN CHATBOT UNTUK PENGEMBANGAN WEBSITE PRODI SISTEM TELEKOMUNIKASI UPI

Universitas Pendidikan Indonesia | repository.upi.edu | Perpustakaan.upi.edu

3. `from keras.layers import dense, Activation, Embedding, LSTM, Flatten``: Mengimpor beberapa jenis lapisan (*layer*) yang sering digunakan dalam pembuatan model *neural network*, yaitu: ``Dense`` (lapisan *fully connected*), ``Activation`` (fungsi aktivasi), ``Embedding`` (lapisan *embedding*), ``LSTM`` (lapisan *Long Short Term Memory*), dan ``Flatten`` (lapisan *flatten*).
4. `from keras.optimizers import SGD``: Mengimpor *optimizer Stochastic Gradient Descent* (SGD) dari pustaka *Keras. Optimizer* digunakan untuk mengatur proses pembelajaran (*training*) model dan memperbarui bobot dan bias agar model dapat melakukan prediksi dengan akurat.
5. `import random``: Mengimpor pustaka ``random`` untuk menghasilkan angka acak.

Setelah semua pustaka yang diperlukan diimpor, Menggunakan kelas dan fungsi dari Keras untuk membangun dan melatih model *neural network* sesuai kebutuhan. Contoh kelas dan fungsi yang dapat digunakan adalah ``Sequential``, ``Dense``, ``Activation``, ``Embedding``, ``LSTM``, ``Flatten``, dan ``SGD``.

3.3.5 Teknik pengumpulan data

Aspek terpenting dalam penelitian *Chatbot* menggunakan algoritma *Recurrent Neural Network* dengan sampai saat ini peneliti masih menggunakan teknik pengumpulan data yang akan dipengaruhi oleh hasil penelitian dengan:

- a. Observasi – Yaitu teknik pengumpulan data yang dilakukan dengan mengamati sesuatu secara langsung terhadap teks. Seperti obrolan dengan *Chatbot*, atau terhadap data non-teks, seperti tingkah laku *Chatbot user*.
- b. Eksperimen terhadap teknik pengumpulan data yang dilakukan untuk mengubah satu atau lebih variabel dan mengamati suatu perubahan yang terjadi. Eksperimen dapat dilakukan terhadap data teks, seperti obrolan dengan pengguna *Chatbot*, atau terhadap data non-teks.
- c. Kuesioner yang diberikan untuk mahasiswa sistel terkait perkembangan lebih lanjut mengenai *Chatbot* yang diperlukan untuk program studi.

3.4. Implementasi sistem *chatbot*

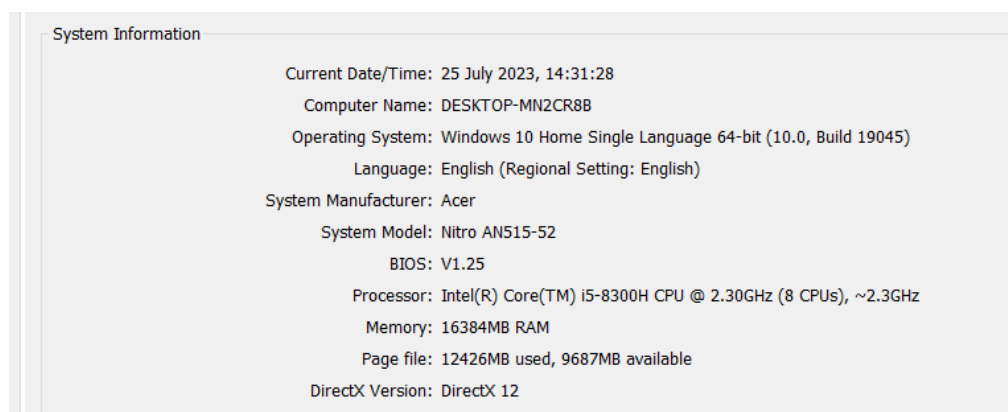
Implementasi *chatbot* dapat melibatkan berbagai kombinasi antara *hardware* dan *software*, tergantung pada kebutuhan dan tujuan penggunaan *chatbot*

tersebut. Berikut adalah penjelasan mengenai implementasi *chatbot* dengan hardware dan software:

3.4.1 Perangkat Keras (*Hardware*)

Penting untuk memperhatikan spesifikasi laptop yang memadai saat melatih data *chatbot*, terutama jika menggunakan model *deep learning* atau *machine learning* yang kompleks. Proses pelatihan data dapat sangat intensif secara komputasional, dan spesifikasi laptop yang cukup kuat dapat mempercepat waktu pelatihan dan meningkatkan kinerja *chatbot* yang dihasilkan. Beberapa faktor penting dalam spesifikasi laptop yang perlu dipertimbangkan termasuk jumlah *core* dan kecepatan prosesor, kapasitas RAM, dan kartu grafis yang mumpuni. Memiliki laptop yang memenuhi syarat akan memberikan pengalaman pengembangan *chatbot* yang lebih efisien dan produktif. Berikut spesifikasi laptop yang digunakan oleh peneliti:

a. Laptop:



Gambar 3. 13 Spesifikasi perangkat keras

3.4.2 Perangkat lunak (*Software*)

Software *chatbot* adalah inti dari sistem *chatbot* dan bertanggung jawab untuk memahami input pengguna, memproses informasi, dan memberikan respons yang sesuai. Berbagai teknologi dan bahasa pemrograman dapat digunakan untuk mengembangkan software *chatbot*, seperti Python, JavaScript, atau Java. Beberapa komponen penting dalam *software chatbot* adalah:

1. *Natural Language Processing* (NLP): NLP merupakan teknologi utama yang digunakan chatbot untuk memahami dan memproses bahasa alami yang

Mohammad Luthfan Faohan, 2023

PENERAPAN ALGORITMA RECURRENT NEURAL NETWORK DAN TERM FREQUENCY INVERSE DOCUMENT FREQUENCY SEBAGAI PENGEKSTRAKSI FITUR RANCANG BANGUN CHATBOT UNTUK PENGEMBANGAN WEBSITE PRODI SISTEM TELEKOMUNIKASI UPI

Universitas Pendidikan Indonesia | repository.upi.edu | Perpustakaan.upi.edu

diucapkan oleh pengguna. Dengan NLP, *chatbot* dapat mengurai kalimat, mengenali entitas, dan memahami maksud dari input pengguna.

2. *Machine Learning*: Beberapa *chatbot* menggunakan teknik *machine learning* untuk meningkatkan kemampuan dalam memahami dan merespons bahasa alami secara lebih baik dari waktu ke waktu. Dengan *machine learning*, *chatbot* dapat belajar dari data pelatihan dan menyesuaikan responsnya berdasarkan pengalaman sebelumnya.
3. *Backend Server*: *Software chatbot* berjalan pada server yang dapat menerima permintaan dari pengguna, memprosesnya, dan mengirimkan respons kembali. *Backend server* juga bertanggung jawab untuk mengintegrasikan *chatbot* dengan sumber daya atau layanan lain yang diperlukan, seperti basis data atau API eksternal.

Tambahan pada software *Colab* dan *VSCode* berfokus pada bagaimana *chatbot* dapat diimplementasikan dan dijalankan menggunakan kedua *platform* tersebut. Berikut adalah langkah-langkah untuk mengimplementasikan *chatbot* pada *Colab* dan *VSCode*:

1. Implementasi *Chatbot* pada *Google Colab*:

Google Colab adalah *platform* yang memungkinkan untuk menjalankan kode Python secara interaktif pada lingkungan *cloud*. Berikut adalah langkah-langkah untuk mengimplementasikan *chatbot* pada *Google Colab*:

- Pastikan memiliki kode *chatbot* yang telah dikembangkan menggunakan Python dan *library* seperti *TensorFlow*, *PyTorch*, atau *Hugging Face Transformers*, tergantung dari teknologi NLP yang digunakan.
- Unggah kode *chatbot* ke *Google Drive* atau buat *notebook* baru pada *Google Colab*.
- Pastikan *notebook Colab* memiliki akses internet untuk dapat mengunduh atau mengakses model NLP yang diperlukan.
- Instal dan siapkan dependensi yang diperlukan dengan menggunakan perintah seperti ``pip install``.

- Jalankan kode *chatbot* dan interaksikan dengan *chatbot* melalui *notebook Colab* tersebut.
- Menyimpan model *chatbot* yang telah dilatih pada *Google Drive* atau *cloud storage* lainnya untuk diakses kembali pada sesi *Colab* berikutnya.

2. Implementasi *Chatbot* pada *Visual Studio Code (VSCode)*:

Visual Studio Code adalah *editor teks* dan kode untuk mengembangkan dan menjalankan aplikasi dengan berbagai bahasa pemrograman. Berikut adalah langkah – langkah untuk mengimplementasikan *chatbot* pada *VSCODE*:

- Pastikan telah menginstal dan mengonfigurasi lingkungan Python pada *VSCode* dengan menggunakan ekstensi seperti Python atau *Anaconda*.
- Buat proyek baru untuk *chatbot* dan pastikan folder proyek yang memiliki struktur sesuai, termasuk kode chatbot dan model NLP yang diperlukan.
- Gunakan terminal atau *integrated terminal* pada *VSCode* untuk menginstal dependensi dengan menggunakan perintah ``pip install``.
- Jalankan kode *chatbot* pada terminal dengan perintah seperti ``python your_chatbot_script.py``.
- Menggunakan Debugger pada *VSCode* untuk melakukan *debugging* kode *chatbot* dan melihat interaksi serta respons yang dihasilkan oleh *chatbot*.
- Pastikan memiliki akses internet untuk dapat mengunduh atau mengakses model NLP yang diperlukan.
- Dapat menyimpan model *chatbot* yang telah dilatih pada folder proyek atau direktori lain untuk diakses kembali pada sesi *VSCode* berikutnya. Perlu diingat bahwa langkah-langkah di atas memberikan panduan umum untuk mengimplementasikan *chatbot* pada kedua platform tersebut. Spesifikasinya dapat bervariasi tergantung pada teknologi *chatbot* yang digunakan dan kebutuhan proyek. Pastikan untuk mengikuti petunjuk dan dokumentasi yang relevan untuk mengintegrasikan chatbot dengan benar pada platform *Colab* dan *VSCode*.

3.4.3 Objek Penelitian

Objek penelitian dalam penelitian ini adalah pengembangan *chatbot* untuk tanya jawab sesama antara manusia dan robot. Penelitian ini bertujuan untuk

Mohammad Luthfan Faohan, 2023

PENERAPAN ALGORITMA RECURRENT NEURAL NETWORK DAN TERM FREQUENCY INVERSE DOCUMENT FREQUENCY SEBAGAI PENGEKSTRAKSI FITUR RANCANG BANGUN CHATBOT UNTUK PENGEMBANGAN WEBSITE PRODI SISTEM TELEKOMUNIKASI UPI

Universitas Pendidikan Indonesia | repository.upi.edu | Perpustakaan.upi.edu

merancang, mengembangkan, dan menguji prototipe *chatbot* yang dapat digunakan dalam interaksi antara pengguna manusia dan robot. Objek penelitian ini meliputi perancangan konsep *chatbot*, implementasi model RNN, penerapan metode TF-IDF, modifikasi fitur-fitur *chatbot*, dan pengukuran pengaruhnya terhadap tingkat *user experience*. Selain itu, penelitian ini juga fokus pada pengembangan *chatbot* sebagai dasar inisiasi untuk *Based-Line Model* pada Website <https://sistel.upi.edu>. Pengembangan *chatbot* ini akan menjadi objek penelitian yang melibatkan identifikasi kebutuhan pengguna, perancangan konsep, pengembangan prototipe, uji coba dan evaluasi, modifikasi fitur-fitur, serta pengukuran pengaruh modifikasi terhadap *user experience*. Dengan demikian, objek penelitian ini meliputi keseluruhan proses pengembangan *chatbot*, mulai dari perancangan konsep hingga pengujian dan evaluasi, serta pengembangan lebih lanjut dalam konteks penggunaan chatbot pada Website <https://sistel.upi.edu>. Objek penelitian ini juga meliputi pengembangan *chatbot* untuk tanya jawab sesama antara manusia dan robot, menggunakan bahasa pemrograman Python, dengan pengembangan dilakukan menggunakan aplikasi *Visual Studio Code (VSCode)*, serta proses *hosting chatbot* pada website Prodi:

1. Merancang dan mengembangkan *chatbot* menggunakan bahasa pemrograman Python: Penelitian ini akan fokus pada penggunaan bahasa pemrograman Python sebagai bahasa utama dalam mengembangkan *chatbot*. Python merupakan bahasa pemrograman yang populer dan memiliki berbagai pustaka dan *framework* yang dapat digunakan dalam pengembangan aplikasi AI seperti *chatbot*.
2. Menggunakan aplikasi *Visual Studio Code (VSCode)* sebagai lingkungan pengembangan: Penelitian ini akan menggunakan aplikasi *VSCode* sebagai lingkungan pengembangan utama. *VSCode* adalah salah satu aplikasi pengembangan terkemuka yang mendukung berbagai bahasa pemrograman dan menyediakan fitur-fitur yang memudahkan pengembangan, seperti *sintaks highlighting*, *debugging*, dan pengelolaan proyek.
3. *Hosting chatbot* pada website Prodi: Tujuan *hosting chatbot* pada website Prodi adalah agar chatbot dapat diakses dan digunakan oleh pengguna secara *online*.

Hosting chatbot pada *website Prodi* memungkinkan akses yang mudah dan cepat bagi pengguna untuk berinteraksi dengan *chatbot*.

4. Pada *website SISTEL UPI*, memiliki *chatbot SISTEL UPI* yang bertujuan untuk memberikan layanan dan bantuan terkait sistem telekomunikasi di Universitas Pendidikan Indonesia (UPI). Dengan menggunakan *Chatbot SISTEL UPI*, Anda dapat dengan mudah bertanya tentang informasi panduan jurusan sistem telekomunikasi, serta mengetahui laporan yang sedang berlangsung. *Chatbot* adalah sebuah program komputer yang menggunakan kecerdasan buatan (AI) untuk berinteraksi dengan manusia melalui chat atau percakapan teks. *Chatbot* dirancang untuk memahami bahasa manusia dan memberikan respons yang relevan dan sesuai dengan pertanyaan atau permintaan yang diajukan oleh pengguna.

Dengan menggabungkan pengembangan *chatbot* menggunakan bahasa pemrograman python, penggunaan aplikasi VSCode, dan *hosting chatbot* pada *website Prodi*, penelitian ini bertujuan untuk menghasilkan sebuah *chatbot* yang dapat digunakan secara efektif dalam konteks tanya jawab antara manusia dan robot, dengan kemampuan pengembangan dan aksesibilitas yang optimal melalui penggunaan teknologi dan alat yang relevan.

3.5 Analisis

Analisis keterkaitan antara RNN dan TF-IDF dalam sistem pemrosesan teks atau data berurutan menunjukkan bahwa RNN berfungsi untuk memproses data secara berurutan, sementara TF-IDF digunakan untuk mengukur pentingnya kata atau frasa dalam dokumen. Dalam hal ini, RNN digunakan untuk memproses teks yang nantinya akan diolah, dan TF-IDF digunakan untuk menilai pentingnya kata atau frasa dalam teks tersebut.