

Perancangan Model Chatbot Cerdas Dengan Pendekatan Deep Learning Untuk Layanan Pelanggan

Intelligent Chatbot Model Design with Deep Learning Approach for Customer Service

Daniel Hutabarat^{a,1}, Lili Tanti^{b,2}

^a Program Studi Informatika UPU, KL. Yos Sudarso Km. 6,5 No. 3A, Medan, 20241, Indonesia

^b Dosen Program Studi Informatika UPU, KL. Yos Sudarso Km. 6,5 No. 3A, Medan, 20241, Indonesia

danielhutabarat0503@gmail.com¹, lilitanti82@gmail.com²

ABSTRAK

Chatbot merupakan suatu program computer yang dirancang untuk meniru percakapan manusia dengan pengguna. Chatbot menjadi salah satu inovasi teknologi yang banyak digunakan sebagai media komunikasi dengan berbagai pengguna dalam berbagai bidang, termasuk layanan pelanggan. Adanya kendala seperti, pelanggan harus datang langsung ke kantor PLN untuk menanyakan permasalahan yang sering terjadi. Dengan adanya Chatbot diharapkan bisa membantu berbagai pelayanan yang terdapat pada PT PLN (Persero) UPT Labuhan. Chatbot juga bisa sangat membantu menjawab pertanyaan – pertanyaan dan kendala apa saja yang sering terjadi pada masyarakat. Dan dengan adanya Chatbot juga dapat mempersingkat waktu terhadap masyarakat untuk menanyakan kendala apa saja yang sering terjadi tanpa harus datang langsung ke kantor PLN.

Kata Kunci : Chatbot untuk Layanan Pelanggan, Deep Learning, Recurrent Neural Network (RNN)

ABSTRACT

A Chatbot is a computer program designed to mimic human conversations with users. Chatbot is one of the technological innovations that is widely used as a medium of communication with various users in various fields, including customer service. There are obstacles such as, customers must come directly to the PLN office to ask about problems that often occur. With the existence of Chatbot, it is hoped that it can help various services available at PT PLN (Persero) UPT Labuhan. Chatbot can also be very helpful in answering questions - what questions and obstacles often occur in the community. And with the Chatbot can also shorten the time for the community to ask what obstacles often occur without having to come directly to the PLN office.

Keywords : Chatbot for Customer Service, Deep Learning, Recurrent Neural Network

Disubmit: 11 Agustus 2024

Info Artikel :
Direview: 15 September 2024

Diterima : 05 Oktober 2024

Copyright © 2024 – senadimu. All rights reserved.

1. PENDAHULUAN

Chatbot adalah program komputer yang dirancang untuk meniru percakapan manusia dalam interaksi dengan pengguna. Chatbot menawarkan interaksi khusus dengan fungsi sistem penguasaan yang cocok serta meningkatkan batasan sistem pembelajaran saat ini. Chatbot yaitu program yang mampu memproses masukan dari pengguna dan menghasilkan tanggapan atau jawaban yang sesuai dengan masukan dari pengguna yang akan dikirim kembali ke pengguna. Tren Chatbot semakin bertambah pesat jumlahnya karena kemudahan dan lebih menguntungkan dengan digunakannya perangkat lunak [1]

Dalam era digital yang semakin berkembang seperti sekarang ini, Chatbot menjadi salah satu inovasi teknologi yang banyak digunakan sebagai media komunikasi dengan pengguna dalam berbagai bidang, termasuk layanan publik. Chatbot memiliki kemampuan untuk menjawab pertanyaan dan memberikan solusi dalam waktu yang lebih cepat dan efisien dibandingkan dengan interaksi manusia ke manusia. Penggunaan Chatbot dapat membantu pelayanan publik dengan cepat serta responsive. [2]

Recurrent Neural Network(RNN) merupakan arsitektur *Deep Learning* yang populer serta sangat menjanjikan untuk menyelesaikan berbagai persoalan yang terkait dengan NLP. *Chatbot* secara otomatis menanggapi pengguna menggunakan RNN [3]. Model RNN digunakan agar mesin dapat memahami bahasa manusia, mulai dari cara berkomunikasi, mendengarkan, mengenali percakapan, hingga memahami tata bahasa dan aksen.

Adanya juga beberapa kendala seperti, para pelanggan harus datang langsung ke kantor PLN untuk menanyakan permasalahan yang sedang terjadi. Pengembangan model pembelajaran mesin sederhana untuk *Chatbot*, Integrasi API dengan aplikasi *Chatbot*, Menangani keluhan customer dengan adanya pelayanan via *Chatbot*, Pengembangan aplikasi *Chatbot* sederhana untuk mendukung interaksi user. Pengembangan API Sederhana dan Penerapan Model Pembelajaran Mesin. berbicara dengan perangkat mereka untuk mengakses informasi dan mendapatkan bantuan terkait produk dan layanan [4].

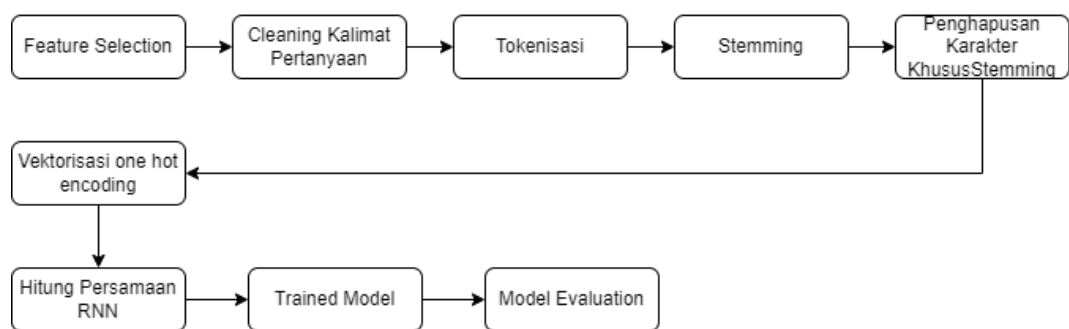
Dalam bahasa sehari-hari *Chatbot* merupakan Aplikasi atau Program komputer yang dirancang untuk meniru manusia itu sendiri, batasan yang diambil dari *Chatbot* 3 adalah mampu meniru komunikasi manusia. Jadi jika manusia sedang bercakap-cakap dengan program ini, seakan-akan ada 2 pribadi manusia yang saling berkomunikasi. Nyatanya tidak, manusia berkomunikasi dengan Bot. Bot sudah dirancang untuk merespon segala jenis pertanyaan dan pernyataan yang diinputkan oleh manusia (user) [5]. Sedangkan *chatbot* bertugas memberikan respon berupa jawaban kepada pelanggan [6].

Dengan adanya *Chatbot* diharapkan bisa membantu berbagai pelayan yang terdapat pada PT PLN (Persero) UPT Labuhan. *Chatbot* juga bisa sangat membantu terhadap pertanyaan – pertanyaan dan kendala apa saja yang sering terjadi pada masyarakat. Dengan adanya *Chatbot* juga bisa mempersingkat waktu terhadap masyarakat untuk menanyakan kendala yang sedang terjadi tanpa harus datang langsung ke kantor PLN.

2. METODE

Metode yang digunakan dalam pembuatan perancangan model chatbot dengan pendekatan deep learning untuk layanan pelanggan adalah dengan Metode *Recurrent Neural Network* Langkah – Langkah nya seperti gambar dibawah ini.

Tabel 1. Tahapan pemrosesan kalimat



1. Feature Selection

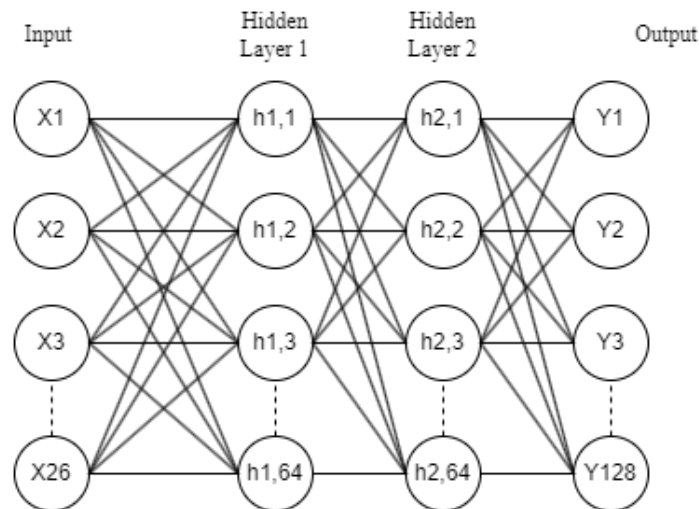
Dalam perancangan model *Chatbot* untuk layanan pelanggan, memiliki 26 dataset yang masing - masing dataset di bagi menjadi beberapa kategori yaitu : Greeting, Pembayaran, Metode, Pemadaman, Keluhan, Hemat, Gangguan, Meteran, Daya, Pindah, Korsleting, Daftar, Mengganti, Memindahkan, Batal, Ajukan, Hitung, Biaya, Penggunaan, Sanksi, Diskon, Instalansi, Pemeliharaan, Darurat, Bantuan, Good Bye.

2. Cleaning Kalimat Pertanyaan

Pembersihan (cleaning) kalimat Pertanyaan berfungsi untuk memastikan data yang diolah sesuai dengan yang diharapkan oleh model. Berikut adalah tahapan-tahapan yang biasa dilakukan dalam pembersihan kalimat dalam bahasa Indonesia

3. Tokenisasi
Memisahkan kalimat menjadi token-token kata. Dalam bahasa Indonesia, tokenisasi seringkali lebih kompleks karena ada kata-kata yang sering kali melekat bersama, seperti "bagaimana", "ke", "dari", dan lain-lain. Jadi, perlu kebijaksanaan dalam menentukan cara tokenisasi yang tepat.
4. Stemming atau Lemmatisasi
Mengubah kata-kata menjadi bentuk dasarnya. Stemming adalah proses memotong akhiran kata untuk mengembalikan kata ke bentuk dasarnya, sedangkan lemmatisasi menggunakan aturan gramatikal dan kamus untuk mengubah kata-kata ke bentuk dasar.
5. Penghapusan Karakter Khusus
Menghapus karakter khusus seperti tanda baca, simbol, atau karakter khusus lainnya yang tidak relevan untuk analisis. Ini dilakukan untuk memastikan bahwa hanya karakter-karakter yang penting dalam kalimat yang dipertahankan.
6. One-Hot Encoding

One-Hot Encoding adalah teknik representasi data yang digunakan untuk mengubah data kategori menjadi vektor biner, di mana setiap kategori direpresentasikan sebagai vektor biner yang panjangnya sama dengan jumlah kategori, dengan nilai 1 menunjukkan keberadaan kategori dan nilai 0 menunjukkan ketiadaannya. Teknik ini sering digunakan dalam pemrosesan bahasa alami dan klasifikasi teks.



Gambar 1. Arsitektur RNN Dengan 2 Hidden Layer

Input memiliki 26 yang berupa 26 kategori pertanyaan yang tersedia pada dataset, dan memiliki 2 hidden layer yang masing - masing memiliki 64 neuron secara keseluruhan, 64 neuron ini tidak terdiri dari komponen khusus atau berbeda satu sama lain. Setiap neuron hanya memiliki bobot, bias, fungsi aktivasi, dan output yang berbeda, namun mereka bekerja secara paralel untuk memproses data yang diterima dari lapisan sebelumnya. Dan memiliki output 128 yang berupa isi dari pertanyaan yang tersedia yang telah dikategorikan

7. Hitung Pada Persamaan RNN
Perhitungan RNN pada data yang digunakan dalam perhitungan yakni pada kalimat pertanyaan dari kalimat yang sudah dijadikan Key dictionary atau kata kunci, yang awalnya

kata diubah menjadi vector one hot encoding. Untuk perhitungan ini akan menggunakan vector kata “bagaimana” dari [10000].

Inisialisasi:

Bobot input to hidden (xh): [[0.1], [0.2]]

Bobot hidden to hidden (hh): [[0.3]]

Bobot hidden to output (hy): [0.4]

Bias hidden (bh): [0.5]

Bias output (by): 0.6

Bobot target (ytrue) : 0.8

Langkah 1:

Hitung hidden(h1) dari input (x1=1):

$$h1-1 = \text{sigmoid}((\text{input} * hh) + bh)$$

$$h1-1 = \text{sigmoid}((1 * 0.1) + 0.5)$$

$$h1-1 = \text{sigmoid}(0.1 + 0.5)$$

$$h1-1 \approx \text{sigmoid}(0.6)$$

$$h1-1 \approx 0.645$$

Hitung hidden(h2-1):

$$h2-1 = \text{sigmoid}((hh * h1-1) + bh)$$

$$h2-1 = \text{sigmoid}((0.3 * 0.645) + 0.5)$$

$$h2-1 = \text{sigmoid}(0.1935 + 0.5)$$

$$h2-1 \approx \text{sigmoid}(0.6935)$$

$$h2-1 \approx 0.666$$

Hitung output (y1):

$$y = \text{sigmoid}(h2-1 * hy + by)$$

$$y = \text{sigmoid}(0.666 * 0.4 + 0.6)$$

$$y = \text{sigmoid}(0.2664 + 0.6)$$

$$y \approx \text{sigmoid}(0.8664)$$

$$y \approx 0.703$$

A. Artificial Intelligence

Kecerdasan Buatan (*Artificial Intelligence*) adalah bidang ilmu komputer yang bertujuan untuk menciptakan sistem komputer atau mesin yang memiliki kemampuan untuk melakukan tugas yang memerlukan kecerdasan manusia. Ini mencakup pengembangan algoritma, perangkat lunak, dan perangkat keras yang memungkinkan komputer untuk memahami data, belajar dari pengalaman, dan melakukan tugas-tugas seperti pengambilan keputusan, pengenalan pola, pemrosesan bahasa alami, dan tugas-tugas lain yang biasanya memerlukan kecerdasan manusia [7].

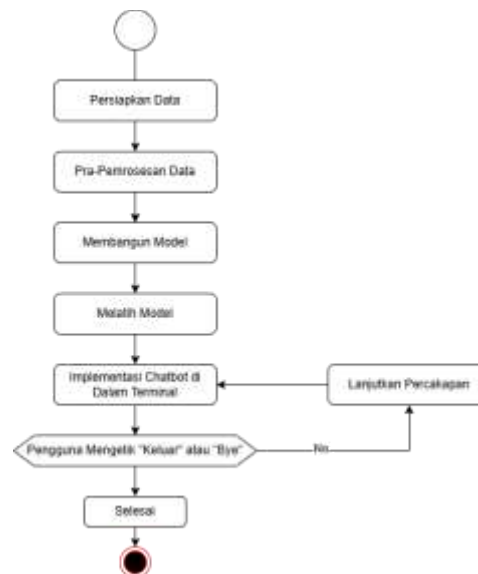
B. *Deep Learning*

Deep learning adalah bagian dari pembelajaran mesin yang berkaitan dengan algoritma dimana cara kerja dari algoritma ini meniru struktur dan fungsi otak yang disebut jaringan saraf tiruan [8]. Penelitian terkait deep learning telah banyak dilakukan dalam berbagai bidang. Seperti misalnya di bidang social, ekonomi [9] dan bahkan hingga ke bidang music [10]. *Deep Learning* juga dapat digunakan dalam beberapa penelitian di bidang kesehatan seperti tumor otak [11], deteksi covid-19 [12], klasifikasi image medis [13] dan sebagainya.

C. *Recurrent Neural Network*

RNN yang juga disebut jaringan umpan balik adalah jenis jaringan pada neural network dimana terdapat loop sebagai koneksi umpan balik dalam jaringan [14]. Jaringan RNN adalah jaringan yang mengakomodasi output jaringan untuk menjadi input pada jaringan tersebut yang kemudian digunakan untuk menghasilkan output yang baru. RNN merupakan jaringan yang dinamis karena perilaku jaringan tidak tergantung pada input saat ini saja melainkan pada operasi jaringan sebelumnya. Selain itu, RNN biasanya digunakan untuk menyelesaikan tugas terkait dengan data time series [15].

Tabel 2. Flowchart Perancangan Model



1. **Persiapkan Data**
Mengumpulkan data percakapan yang akan digunakan untuk melatih model.
2. **Pra-Pemrosesan Data**
Mengubah teks menjadi kosakata dan menyamakan urutan teks untuk memiliki panjang yang sama
3. **Membangun Model**
Membangun model untuk pemrosesan input dan untuk menghasilkan output
4. **Melatih Model**
Menggunakan data yang telah di pra-proses untuk mode
5. **Implementasi Chatbot di Terminal**
Menampilkan respon di terminal

6. *Chatbot* akan berhenti jika mengetik keluar atau bye, jika tidak akan melanjutkan percakapan.

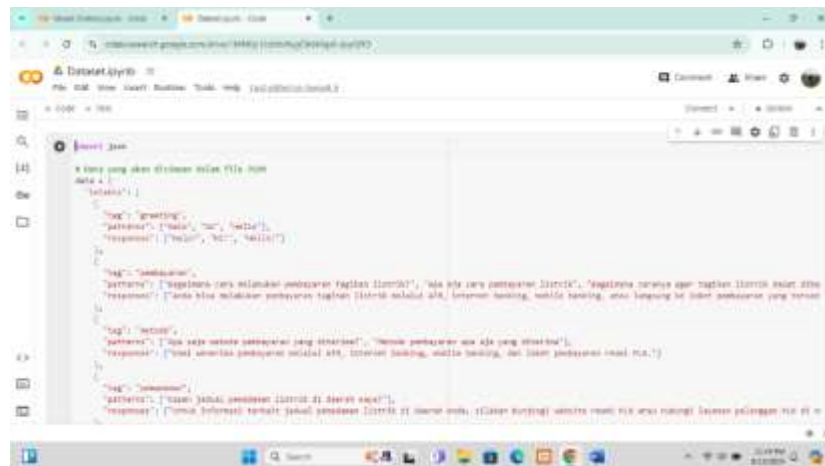
3. HASIL DAN PEMBAHASAN

Aplikasi yang digunakan untuk membuat model *chatbot* untuk layanan pelanggan menggunakan google colab dengan menggunakan bahasa python

Tahapan pertama untuk bisa menjalankan aplikasi adalah menggunakan browser apapun yang ada di device pengguna seperti, Google Chrome, Mozilla Firefox, Atau Microsoft Edge dan browser lainnya. Penulis disini menggunakan Google Chrome, bisa dilihat tahapan seperti berikut :

1. Dataset

Kode dibawah ini merupakan dataset yang sudah dibuat untuk merancang model *Chatbot*. Dataset yang berisi 26 kategori pertanyaan yang dimana masing – masing keseluruhan dari 26 katogori pertanyaan terdapat 128 pertanyaan yang ada. Lalu akan muncul data yang telah di upload berupa dataset pada *chatbot* seperti pada gambar berikut :



Gambar 2. Tampilan Dataset

2. Cleaning Kalimat Pertanyaan

Pada tahap ini model melakukan cleaning kalimat pertanyaan yang terbagi dari beberapa tahapan yaitu tokenisasi dan lematisasi. Dapat dilihat seperti gambar dibawah. :

a. Tokenisasi

Memisahkan kalimat menjadi token-token kata. Dalam bahasa Indonesia, tokenisasi seringkali lebih kompleks karena ada kata-kata yang sering kali melekat bersama, seperti "bagaimana", "ke", "dari", dan lain-lain. Jadi, perlu kebijaksanaan dalam menentukan cara tokenisasi yang tepat. Hasil dari tokenisasi dapat dilihat sebagai berikut

Tabel 3. Hasil Tokenisasi

Kalimat awal	Setelah Tokenisasi
Bagaimana cara pembayaran tagihan listrik	("Bagaimana", "cara", "pembayaran", "tagihan", "listrik")

b. Stemming (Lematisasi)

Mengubah kata-kata menjadi bentuk dasar nya. Stemming adalah proses memotong akhiran kata untuk mengembalikan kata ke bentuk dasarnya, sedangkan lemmatisasi menggunakan aturan gramatikal dan kamus untuk mengubah kata-kata ke bentuk dasar. Hasil dari stemming dapat dilihat sebagai berikut

Tabel 4. Hasil Stemming

Kalimat awal	Setelah Stemming
Bagaimana cara pembayaran tagihan listrik	“Bagaimana”, “cara”, “bayar”, “tagih”, “listrik”

3. Penghapusan Karakter Khusus.

Menghapus karakter khusus seperti tanda baca, simbol, atau karakter khusus lainnya yang tidak relevan untuk analisis. Ini dilakukan untuk memastikan bahwa hanya karakter-karakter yang penting dalam kalimat yang dipertahankan. Berikut kode pada model *chatbot* dapat dilihat sebagai berikut

Adapun hasil dari penghapusan karakter khusus dapat dilihat sebagai berikut

Tabel 5. Hasil penghapus karakter

Kalimat Awal	Setelah Dihapus
Bagaimana cara pembayaran tagihan listrik?	Bagaimana cara pembayaran tagihan listrik
Ayo bayar listrik dengan tepat waktu!	Ayo bayar listrik dengan tepat waktu
Hemat pangkal kaya.	Hemat pangkal kaya

4. Vektorisasi One Hot Encoding

One-Hot Encoding adalah teknik representasi data yang digunakan untuk mengubah data kategori menjadi vektor biner, di mana setiap kategori direpresentasikan sebagai vektor biner yang panjangnya sama dengan jumlah kategori, dengan nilai 1 menunjukkan keberadaan kategori dan nilai 0 menunjukkan ketiadaannya. Teknik ini sering digunakan dalam pemrosesan bahasa alami dan klasifikasi teks. Berikut tampilan kode untuk mencari nya dapat dilihat sebagai berikut

Adapun hasil dari vektorisasi one hot encoding dengan menggunakan contoh pertanyaan (“Bagaimana”, “cara”, “melakukan”, “pembayaran”, “tagihan”, “listrik”) dan hasil konversi ke vector one hot encoding seperti berikut

bagaimana	1	0	0	0	0	0
cara	0	1	0	0	0	0
melakukan	0	0	1	0	0	0
pembayaran	0	0	0	1	0	0
tagihan	0	0	0	0	1	0
listrik	0	0	0	0	0	1

Gambar 3. Tampilan Hasil Vektorisasi one hot encoding

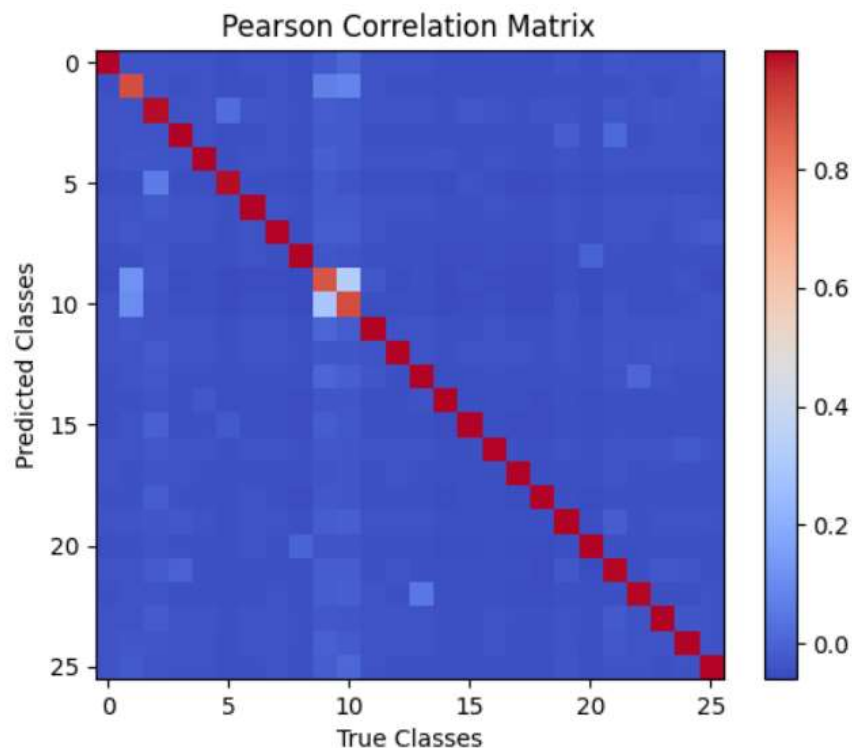
5. Corellation Matrix

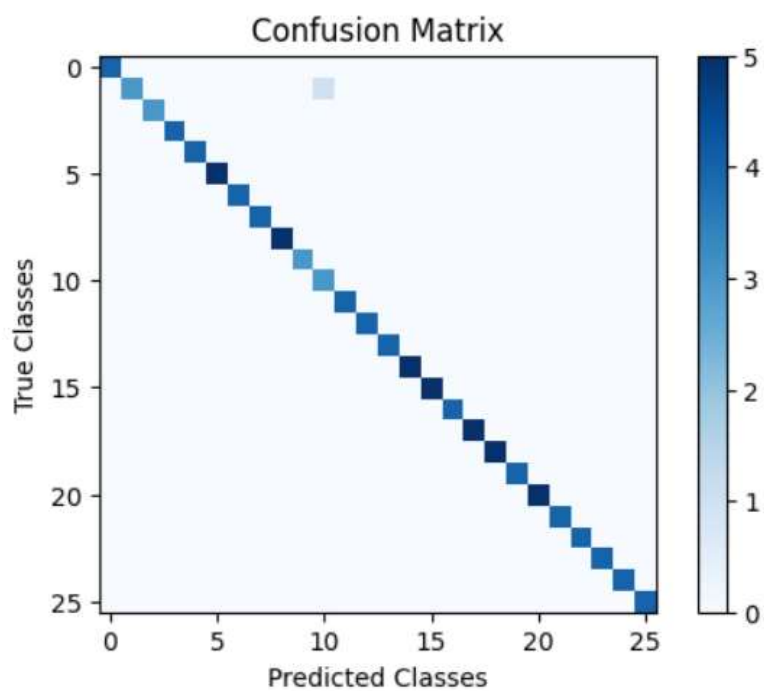
Correlation matrix adalah alat analisis yang digunakan untuk menampilkan korelasi antara berbagai variabel dalam bentuk tabel matriks. Setiap nilai dalam matriks ini menunjukkan seberapa kuat dan arah hubungan antara dua variabel, dengan nilai berkisar dari -1 hingga 1: Nilai 1 berarti korelasi positif sempurna (jika satu variabel naik, yang lainnya juga naik). Nilai -1 berarti korelasi negatif sempurna (jika satu variabel naik, yang lainnya turun). Nilai 0 berarti tidak ada korelasi (kedua variabel tidak berhubungan).

Pearson Correlation Matrix:

```
[ [ 9.98744643e-01 -3.44425651e-02 -3.55255790e-02 -4.10357285e-02
-4.21161999e-02 -4.77018386e-02 -4.15559154e-02 -3.87831501e-02
-4.77338444e-02 -2.39046627e-02 5.68743394e-03 -4.17918309e-02
-4.07761705e-02 -4.23080080e-02 -4.70657243e-02 -4.76968386e-02
-4.25000169e-02 -4.56073015e-02 -4.76358810e-02 -3.80142730e-02
-4.77481246e-02 -3.29195423e-02 -4.21985630e-02 -4.21287967e-02
-4.16479184e-02 -2.12634216e-02]
[ -4.33099064e-02 8.96801141e-01 -3.58497209e-02 -4.33909046e-02
-4.20475688e-02 -4.81672413e-02 -4.24712777e-02 -4.10222404e-02
-4.81265844e-02 7.06987974e-02 8.42589204e-02 -3.70050786e-02
-4.34749149e-02 -4.31048114e-02 -4.85058863e-02 -4.82001390e-02
-4.34684222e-02 -4.88356979e-02 -4.88097283e-02 -4.24277655e-02
-4.88213393e-02 -4.29277935e-02 -4.10428292e-02 -4.34259556e-02
-4.05926585e-02 -3.30281756e-02]
[ -3.96093388e-02 -3.57241870e-02 9.88074881e-01 -3.92363548e-02
-3.89218081e-02 2.63366674e-02 -3.15692610e-02 -3.96126321e-02
-4.28868780e-02 -1.60009017e-02 -2.48194401e-02 -3.95981657e-02
-3.74231259e-02 -3.73568831e-02 -4.42104329e-02 -2.66336781e-02
-3.87642178e-02 -4.42695023e-02 -3.54139162e-02 -3.92283339e-02
-4.30417302e-02 -2.95665391e-02 -3.94139990e-02 -3.03729342e-02
-3.44106602e-02 -3.96840375e-02]
```

Gambar 4. Correlation Matrix dalam Bentuk Angka





Gambar 7. Confusion Matrix dalam Bentuk Grafik

7. Data Training

Pada tahapan ini dilakukan 200 pelatihan terhadap model *chatbot* yang sudah dibuat dan didapatkan hasil nilai akurasi 97% dan nilai loss 70%. Dapat dilihat seperti gambar berikut

```
model.fit(train_data, train_labels, epochs=200, batch_size=5, verbose=1)

loss, accuracy = model.evaluate(train_data, train_labels)
print(f'Loss: {loss:.3f}, Accuracy: {accuracy:.3f}')
```

Epoch 1/200
8/8 ————— 3s 10ms/step - accuracy: 0.0126 - loss: 3.2743
Epoch 2/200
8/8 ————— 0s 4ms/step - accuracy: 0.0170 - loss: 3.3349
Epoch 3/200
8/8 ————— 0s 4ms/step - accuracy: 0.0000e+00 - loss: 3.3205
Epoch 4/200
8/8 ————— 0s 6ms/step - accuracy: 0.0470 - loss: 3.1803
Epoch 5/200
8/8 ————— 0s 5ms/step - accuracy: 0.0314 - loss: 3.2925
Epoch 6/200
8/8 ————— 0s 5ms/step - accuracy: 0.0537 - loss: 3.2585
Epoch 7/200
8/8 ————— 0s 4ms/step - accuracy: 0.0633 - loss: 3.2269
Epoch 8/200
8/8 ————— 0s 7ms/step - accuracy: 0.0126 - loss: 3.2923
Epoch 9/200
8/8 ————— 0s 4ms/step - accuracy: 0.0768 - loss: 3.2564
Epoch 10/200
8/8 ————— 0s 4ms/step - accuracy: 0.0126 - loss: 3.1953

Gambar 8. Tampilan Kode dari Data Training dan Hasil Training

Epoch 189/200
8/8 ————— 0s 2ms/step - accuracy: 0.5798 - loss: 1.4166
Epoch 190/200
8/8 ————— 0s 2ms/step - accuracy: 0.5968 - loss: 1.3925
Epoch 191/200
8/8 ————— 0s 3ms/step - accuracy: 0.8559 - loss: 1.0317
Epoch 192/200
8/8 ————— 0s 2ms/step - accuracy: 0.7979 - loss: 1.1686
Epoch 193/200
8/8 ————— 0s 3ms/step - accuracy: 0.7056 - loss: 1.4359
Epoch 194/200
8/8 ————— 0s 2ms/step - accuracy: 0.5713 - loss: 1.4904
Epoch 195/200
8/8 ————— 0s 2ms/step - accuracy: 0.7281 - loss: 1.3335
Epoch 196/200
8/8 ————— 0s 2ms/step - accuracy: 0.6404 - loss: 1.2623
Epoch 197/200
8/8 ————— 0s 3ms/step - accuracy: 0.6868 - loss: 1.3507
Epoch 198/200
8/8 ————— 0s 3ms/step - accuracy: 0.7700 - loss: 0.9487
Epoch 199/200
8/8 ————— 0s 4ms/step - accuracy: 0.6602 - loss: 1.3866
Epoch 200/200
8/8 ————— 0s 2ms/step - accuracy: 0.7352 - loss: 1.1638
2/2 ————— 0s 6ms/step - accuracy: 0.9725 - loss: 0.6917
Loss: 0.709, Accuracy: 0.974

Gambar 9. Tampilan Kode dari Data Training dan Hasil Training

8. Data Testing

Pada tahapan ini telah dilakuka data testing dari model *chatbot* yang sudah dibuat dan di dapatkan hasil nilai presisi 95% dan nilai *recall* 97%. Dapat dilihat tampilan berikut ini

```
[ ] # Data Testing menghitung presisi dan recall
true_classes = np.argmax(train_labels, axis=1)

# Predict classes using the trained model
predicted_classes = np.argmax(model.predict(train_data), axis=1)

precision, recall, f1_score, _ = precision_recall_fscore_support(true_classes, predicted_classes, average='weighted')

metrics = {
    'Precision': precision,
    'Recall': recall,
    'F1-Score': f1_score
}

print('Evaluation Metrics:')
for metric, value in metrics.items():
    print(f'{metric}: {value:.3F}')

2/2 ----- 0% 218ms/step
Evaluation Metrics:
Precision: 0.955
Recall: 0.974
```

Gambar 10. Tampilan Kode dari Data Testing dan Hasil Testing

9. Hasil Model Chatbot

Pada tahapan ini menampilkan hasil dari keseluruhan model *chatbot* yang sudah dirancang. Model akan merespon segala pertanyaan yang dapat dibaca oleh model yang sudah dirancang dan model akan melakukan pertanyaan jika pertanyaan yang ditanyakan tidak dapat dibaca oleh model *chatbot*. Berikut tampilan dari hasil model *chatbot*

```
Bot is running! (ketik "quit" untuk mengakhiriaya)
You: halo
1/1 ----- 0% 74ms/step
Bot: Halo!
You: apa saja metode pembayaran tagihan listrik
1/1 ----- 0% 18ms/step
Bot: Anda bisa melakukan pembayaran tagihan listrik melalui ATM, internet banking, mobile banking, atau langsung ke loket pembayaran yang tersedia.
You: apa aja cara pembayaran listrik
1/1 ----- 0% 28ms/step
Bot: Anda bisa melakukan pembayaran tagihan listrik melalui ATM, internet banking, mobile banking, atau langsung ke loket pembayaran yang tersedia.
You: apakah sudah pemadaman listrik di daerah saya?
1/1 ----- 0% 28ms/step
Bot: Untuk informasi terkait jadwal pemadaman listrik di daerah anda, silakan kunjungi website resmi PLN atau hubungi layanan pelanggan PLN di nomor 123.
You: Adakah solusi untuk menghemat energi?
1/1 ----- 0% 28ms/step
Bot: Ya, PLN memiliki berbagai program dan tips untuk penghematan energi. Anda dapat menemukan informasi lebih lanjut di website resmi PLN atau mengikuti prog
You: terima kasih
1/1 ----- 0% 28ms/step
Bot: Selamat tinggal!
You: adit
```

Gambar 11. Hasil Model Chatbot

10. Training Model dan Evaluation

a. Akurasi

Akurasi adalah persentase dari prediksi yang benar dari keseluruhan prediksi yang dibuat oleh model

b. Loss

Loss adalah ukuran seberapa jauh prediksi model dari nilai sebenarnya. Dalam supervised learning, loss mengukur ketidaksesuaian antara prediksi dan label target.

c. Recall

Recall mengukur seberapa baik model mengidentifikasi semua contoh yang sebenarnya positif.

d. Presisi

Presisi mengukur seberapa akurat model ketika memprediksi kelas positif.

Pada tahap ini Model *Chatbot* yang penulis rancang memiliki nilai akurasi 97% dan, nilai loss 70%, *recall* 97% dan nilai presisi 95%. Sedangkan Pada penelitian terkait yang dilakukan oleh (Zurayyah et al., 2019).Yang berjudul “Implementasi *Chatbot* Pada Pendaftaran Mahasiswa Baru Menggunakan *Recurrent Neural Network*” Dengan tingkat akurasi sebesar 88%, los sebesar 70%, presisi 95%, dan *recall* 92%.

Tabel 6. Training Model dan Evaluation

Penelitian	Akurasi	Loss	<i>Recall</i>	Presisi
Penulis	97%	70%	97%	95%
(Zurayyah et al., 2019)	Akurasi 88%	Loss 70%	<i>Recall</i> 92%	Presisi 95%

4. KESIMPULAN

Pada penelitian yang berjudul Perancangan Model *Chatbot* Cerdas Dengan Pendekatan *Deep Learning* Untuk Layanan Pelanggan, Penulis telah berhasil merancang model *Chatbot* menggunakan *Recurrent Neural Networks* (RNN) untuk pelatihan data yang dapat membantu menemukan jawaban yang sesuai dari pertanyaan pengguna. JSON digunakan untuk menyimpan data serta menampilkan data-data yang tersimpan menggunakan *Google Colab* dengan Bahasa Pemrograman Python untuk membantu dalam pembuatan dan pengembangan model ini .Didapatkan tingkat akurasi model sebesar 97%, presisi sebesar 95% dan *recall* sebesar 97%. Pembuatan model ini bertujuan untuk mempermudah pengguna dalam mendapatkan informasi yang cepat dan akurat secara realtime.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Universitas Potensi Utama yang telah banyak memberikan masukan dan saran dalam penyelesaian Karya Tulis Ilmiah ini. Semoga karya ilmiah yang telah penulis buat bermanfaat dan dapat menjadi sumber positif bagi perkembangan ilmu pengetahuan.

REFERENSI

- [1] M. Mustaqim, A. Gunawan, Y. Bagus Pratama, and I. Zaliman, “PENGEMBANGAN CHATBOT LAYANAN PUBLIK MENGGUNAKAN MACHINE LEARNING DAN NATURAL LANGUAGE PROCESSING,” 2023. [Online]. Available: <https://jits.unmuhbabel.ac.id/>
- [2] T. A. Zurayyah, D. K. Utami, and D. Herlambang, “Implementasi Chatbot Pada Pendaftaran Mahasiswa Baru Menggunakan Recurrent Neural Network,” *J. Ilm. Teknol. dan Rekayasa*, vol. 24, no. 2, pp. 91–101, 2019, doi: 10.35760/tr.2019.v24i2.2388.
- [3] Y. P. Yoanda, T. Hidayat, P. Studi, S. Informasi, F. T. Informasi, and U. N. Mandiri, “Rancang Bangun Chatbot Untuk Meningkatkan Pelayanan Customer Pada Aplikasi Traveloka,” vol. 3, no. 2, pp. 337–352, 2022.
- [4] D. Heryati, R. M. Fajri, D. Learning, and A. N. Network, “UNIVERSITAS INDO GLOBAL MANDIRI MENGGUNAKAN”.
- [5] A. L. Maitri and J. Sutopo, “Rancangan Bangun Chatbot Sebagai Pusat Informasi Lembaga Kursus Dan Pelatihan Menggunakan Pendekatan Natural Language Processing,” *Eprints.Uty.Ac.Id*, pp. 1–9, 2019, [Online]. Available: <http://eprints.uty.ac.id/>

- [6] R. H. Ardiansyah and A. G. Sulaksono, "Layanan pelanggan berbasis Natural Language Processing melalui chatbot pada aplikasi pesan," *J. Inf. Syst. Appl. Dev.*, vol. 1, no. 1, pp. 29–37, 2023, doi: 10.26905/jisad.v1i1.9858.
- [7] M. Solekhah, "Pemanfaatan Teknologi Artificial Intelligent CUBATBOT (Culture Balinese Chatbot) Sebagai Informasi Kebudayaan Bali," *Krepa Kreat. Pada Abdimas*, vol. 1, no. 2, pp. 90–100, 2023.
- [8] N. Rochmawati, H. B. Hidayati, Y. Yamasari, H. P. A. Tjahyaningtjas, W. Yustanti, and A. Prihanto, "Analisa Learning Rate dan Batch Size pada Klasifikasi Covid Menggunakan Deep Learning dengan Optimizer Adam," *J. Inf. Eng. Educ. Technol.*, vol. 5, no. 2, pp. 44–48, 2021, doi: 10.26740/jieet.v5n2.p44-48.
- [9] L. Barbaglia, S. Consoli, and S. Manzan, *Exploring the Predictive Power of News and Neural Machine Learning Models for Economic Forecasting*, vol. 12591 LNAI. Springer International Publishing, 2021. doi: 10.1007/978-3-030-66981-2_11.
- [10] C. Huang and Q. Zhang, "Research on Music Emotion Recognition Model of Deep Learning Based on Musical Stage Effect," *Sci. Program.*, vol. 2021, 2021, doi: 10.1155/2021/3807666.
- [11] A. Pundir and E. Rajeev Kumar, "Brain Tumor Classification in MRI Images Using Transfer Learning," *Stud. Big Data*, vol. 82, pp. 307–319, 2021, doi: 10.1007/978-981-15-9492-2_14.
- [12] W. Yustanti, N. Rahmawati, and Y. Yamasari, "Klastering Wilayah Kota/Kabupaten Berdasarkan Data Persebaran Covid-19 Di Propinsi Jawa Timur dengan Metode K-Means," *J. Inf. Eng. Educ. Technol.*, vol. 4, no. 1, pp. 1–9, 2020, doi: 10.26740/jieet.v4n1.p1-9.
- [13] D. Prevention, "Beware : Blue Light Damage On The Rise Beware : Blue Light Damage On The Rise Beware : Blue Light Damage On The Rise," *J. Digit. Imaging*, pp. 1–2, 2017.
- [14] A. A. Rizal and S. Soraya, "Multi Time Steps Prediction dengan Recurrent Neural Network Long Short Term Memory," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 18, no. 1, pp. 115–124, 2018, doi: 10.30812/matrik.v18i1.344.
- [15] D. Tarkus, S. R. U. A. Sompie, and A. Jacobus, "Implementasi Metode Recurrent Neural Network pada Pengklasifikasian Kualitas Telur Puyuh," *J. Tek. Inform.*, vol. 15, no. 2, pp. 137–144, 2020.