

# **Pengembangan Aplikasi *Chatbot* Dinas Pariwisata Kota Semarang Berbasis *Neural Network***

*Development of a Chatbot Application for the Semarang City Tourism Office Based on a Neural Network*

Yohana Tri Widayati<sup>1</sup>, Stephanus Widjaja<sup>2</sup>, Adityo Putro Wicaksono<sup>3</sup>, Satrio Agung Prakoso<sup>4</sup>, Christina Priscilla Putri<sup>5</sup>

<sup>1</sup>Sistem Informasi, Fakultas Teknik dan Informatika, Universitas AKI Semarang

<sup>2</sup>Teknik Informatika, STMIK AKI Pati

<sup>3,4,5</sup>Teknik Informatika, Fakultas Teknik dan Informatika, Universitas AKI Semarang

E-mail: <sup>1</sup>yohana.tri@unaki.ac.id, <sup>2</sup>stephanuswidjaja@gmail.com, <sup>3</sup>adityo.putro@unaki.ac.id,

<sup>4</sup>satrio.agung@unaki.ac.id, <sup>5</sup>223190004@student.unaki.ac.id

## **Abstrak**

Jawa Tengah memiliki beragam tempat wisata yang menarik untuk dijelajahi, namun keterbatasan waktu dan terbatasnya sumber informasi membuat sulit untuk memberikan informasi yang tepat. Selain itu, pengelola objek wisata kesulitan memberikan informasi yang tepat kepada pengunjung. Tujuan dari penelitian ini adalah memasukkan neural network ke dalam aplikasi chatbot objek wisata Jawa Tengah untuk meningkatkan kemampuannya dalam memahami dan bereaksi secara tepat terhadap pertanyaan pengguna. Teknik AI Project Cycle yang dipadukan dengan pendekatan Natural Language Processing (NLP) berfungsi sebagai tahap preprocessing pada pengolahan kata. Penelitian ini menghasilkan akurasi sebesar 99,92% dan loss sebesar 0%. Program chatbot ini bertujuan untuk memberikan kemudahan akses informasi tempat wisata di Jawa Tengah kepada pengguna.

Kata kunci: chatbot, pariwisata, neural network

## **Abstract**

*Central Java has a variety of interesting tourist attractions to explore, but limited time and limited sources of information make it difficult to provide accurate information. In addition, tourist attraction managers have difficulty providing appropriate information to visitors. This research aims to incorporate a neural network into a Central Java tourist attraction chatbot application to improve its ability to understand and react appropriately to user questions. The AI Project Cycle technique combined with the Natural Language Processing (NLP) approach is a preprocessing stage in word processing. This research resulted in an accuracy of 99.92% and a loss of 0%. This chatbot program aims to give users easy access to information on tourist attractions in Central Java.*

*Keywords: chatbot, tourism, neural network*

## **1. PENDAHULUAN**

Menurut data yang diperoleh dari Badan Pusat Statistik Provinsi Jawa Tengah pada tahun 2021, dari berbagai provinsi di Pulau Jawa, Jawa Tengah memiliki luas wilayah sekitar 28,94%. Selain itu, Jawa Tengah juga memiliki 28 kabupaten dan 6 kota. Dari semua sektor bidang, bidang pariwisata di Jawa Tengah memiliki daya tarik tersendiri bagi pengunjung domestik dan mancanegara dikarenakan kekayaan alam, warisan budaya, dan destinasi tempat wisata yang beragam. Bidang Pariwisata Jawa Tengah mencatat bahwa tingkat pengunjung wisatawan yang berkunjung ke Indonesia, terutama Jawa Tengah sebanyak 7.304.731 wisatawan, atau sekitar 25,5% dari jumlah wisatawan yang berkunjung ke Indonesia[1].

Wisatawan di dalam berkunjung ke berbagai pusat destinasi wisata, memiliki beberapa tantangan di dalam mencari informasi mengenai objek wisata, salah satunya adalah informasi objek wisata di Jawa Tengah yang tersebar di berbagai faktor, di antaranya terdapat banyak informasi tentang objek wisata di Jawa Tengah yang tersebar di berbagai platform sehingga wisatawan kesulitan untuk menemukan sumber informasi yang dapat diandalkan dan akurat, pihak pengelola objek wisata menghadapi tantangan dalam memberikan informasi yang memadai kepada pengunjung karena keterbatasan tenaga, serta chatbot di situs web resmi pariwisata Jawa Tengah memiliki kekurangan karena adanya jam operasional pada chatbot sehingga tidak dapat selalu menanggapi pertanyaan pengguna.

Menurut Lubis, *chatbot* dengan peningkatan teknologi yang canggih berbasis kecerdasan buatan telah mendorong penggunaan aplikasi untuk bisa melakukan tugas-tugas yang biasanya membutuhkan tenaga manusia[2]. *Chatbot* juga memungkinkan pengguna bisa berinteraksi layaknya layanan informasi yang bisa memenuhi permintaan pengguna. Dalam konteks pariwisata, aplikasi *chatbot* dapat memberikan informasi tentang lokasi, fasilitas, acara, jam operasional, atau layanan yang tersedia di objek wisata[3], [4][5].

Dalam pengembangan aplikasi *chatbot*, metode yang digunakan adalah *neural network* (jaringan syaraf tiruan). *Neural network* adalah model algoritma yang mengambil inspirasi dari bagaimana fungsi *neuron* otak manusia dan mentransmisikan informasi melalui antar *neuron*. Selain itu juga, penerapan *learning rate* dan *epoch* pada *Neural Network* juga dapat mempengaruhi kinerja dari ketepatan hasil prediksi untuk penerapan chatbot ini.[2][6][7]. Pengembangan algoritma ini akan diterapkan untuk chatbot di dalam proses prediksi hasil *responses* terhadap *patterns* yang diberikan. Selain itu, dengan adanya aplikasi *chatbot* ini, diharapkan para wisatawan dapat dengan mudah dan cepat mendapatkan informasi objek pariwisata di Jawa Tengah dan juga dapat mengurangi beban kerja pengelola tempat wisata untuk memberikan layanan informasi secara *up to date* kepada wisatawan.

## 2. METODE PENELITIAN

### 2.1 Populasi dan Sampel

Populasi dari penelitian ini adalah masyarakat umum yang berwisata ke Jawa Tengah. Dikarenakan jumlah dari wisatawan yang berkunjung ke Jawa Tengah tidak diketahui, maka populasi pada penelitian ini dapat dikategorikan tidak terhingga atau tidak diketahui[8]. Jumlah sampel yang diambil dalam penelitian ini menggunakan rumus Lemeshow yang ditulis pada rumus 1, hal ini dikarenakan jumlah populasi tidak diketahui atau tidak terhingga. Berdasarkan rumus tersebut, maka besar sampel dalam penelitian ini adalah sebagai berikut[9]:

$$n = \frac{(z_1 - a/2)^2 p(1-p)}{d^2} \quad (1)$$

Keterangan:

n = jumlah sampel

z = skor z pada kepercayaan 95% = 1,96

p = maksimal estimasi

d = alpha (0,10) atau *error sampling* = 10%

### 2.2 Metode Pengumpulan Data

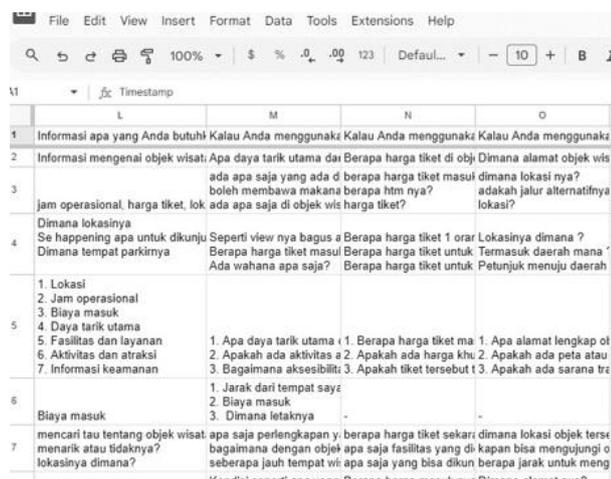
Pada penelitian ini pengumpulan data menggunakan metode observasi, yaitu dengan menyebarkan angket ke responden. Jawaban responden dalam angket digunakan dalam dataset pada bagian *patterns* yang merupakan daftar pertanyaan, sedangkan untuk informasi objek wisata didapat berdasarkan situs resmi dari pariwisata Provinsi Jawa Tengah yaitu

<https://visitjawatengah.jatengprov.go.id/id/destinasi-wisata>. Berikut adalah pertanyaan yang penulis ajukan dalam angket pada tabel 1 dan jawaban dari pertanyaan yang sudah dikumpulkan dalam bentuk excel format ditunjukkan pada gambar 1.

Tabel 1 Daftar Pertanyaan Kuisisioner

No	Pertanyaan	Jawaban
1	Apakah mudah untuk mendapatkan informasi mengenai objek wisata yang ada di Jawa Tengah?	<ul style="list-style-type: none"> <li>● Sangat Tidak Mudah</li> <li>● Tidak Mudah</li> <li>● Mudah</li> <li>● Sangat Mudah</li> </ul>
2	Apakah media/teknologi untuk mendapatkan informasi mengenai objekwisata di Jawa Tengah sudah akurat?	<ul style="list-style-type: none"> <li>● Sangat Tidak Akurat</li> <li>● Tidak Akurat</li> <li>● Akurat</li> <li>● Sangat Akurat</li> </ul>
3	Informasi apa yang Anda butuhkan jika Anda akan pergi ke suatu objek wisata yang baru Anda datangi?	
4	Kalau Anda menggunakan <i>chatbot</i> , apayang akan Anda tanyakan ketika ingin mengetahui <b>informasi/deskripsi</b> objek wisata yang akan dituju? Tuliskan minimal 3 pertanyaan	
5	Kalau Anda menggunakan <i>chatbot</i> , apayang akan Anda tanyakan ketika ingin mengetahui <b>harga tiket</b> objek wisata yang akan dituju? Tuliskan minimal 3 pertanyaan	
6	Kalau Anda menggunakan <i>chatbot</i> , apayang akan Anda tanyakan ketika ingin mengetahui <b>alamat</b> objek wisata yang akan dituju? Tuliskan minimal 3 pertanyaan	
7	Kalau Anda menggunakan <i>chatbot</i> , apayang akan Anda tanyakan ketika ingin mengetahui <b>jam operasional</b> objek wisata yang akan dituju? Tuliskan minimal 3 pertanyaan	

Hasil dari sebaran kuisisioner berdasarkan dari tabel 1 kemudian dibuat ke dalam formulir dan hasil dari jawaban kuisisioner dalam bentuk excel. Pada Gambar 1 ditunjukkan hasil dari pengumpulan kuisisioner yang dijadikan acuan untuk pembuatan dataset chatbot. Setelah dilakukan pengumpulan data pada gambar 1, dilakukan proses pembuatan data JSON untuk dapat dilakukan proses modeling pada tahapan *AI Project Life Cycle*.

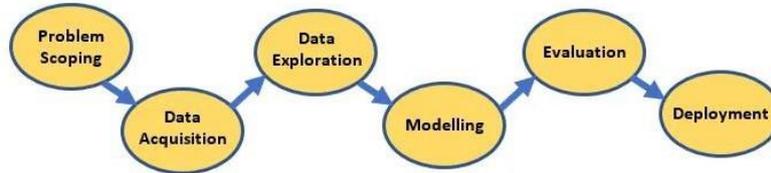


Gambar 1 Hasil pengumpulan data kuisisioner

### 2.3 AI Project Life Cycle

Siklus proyek kecerdasan buatan (AI) adalah metode bertahap yang harus digunakan oleh

organisasi atau individu guna membuat dan menerapkan proyek kecerdasan buatan (AI) sambil memecahkan masalah[10][11]. Siklus ini juga dikenal sebagai siklus hidup pengembangan AI. Siklus ini memberi kita landasan yang memadai untuk mencapai tujuan yang ingin dicapai. Langkah-langkah dari siklus proyek AI terdapat pada gambar 2[12][13].



Gambar 2 Siklus AI Project Life Cycle

### 1. Problem Scoping

Pada tahap awal sebuah proyek, *problem scoping* digunakan untuk memahami dan mengidentifikasi masalah atau tantangan yang perlu diatasi. Hal ini diperlukan untuk memastikan bahwa waktu, orang, dan sumber daya yang tersedia dapat digunakan secara efektif dan efisien untuk mencapai tujuan yang diinginkan. Peluang keberhasilan proyek akan meningkat dan risiko proyek akan berkurang dengan melakukan *problem scoping*. Berikut beberapa masalah yang diidentifikasi dan menjadi *problem scoping* penelitian ini, antara lain:

1. Para wisatawan menghadapi kendala dalam mencari dan memperoleh informasi yang lengkap tentang objek wisata di Jawa Tengah, dikarenakan keterbatasan waktu kunjungan dan harga tiket masuk objek wisata tersebut.
2. Terdapat banyak sumber informasi tentang objek wisata di Jawa Tengah yang tersebar di berbagai platform dan situs web, sehingga kesulitan untuk menemukan sumber informasi yang dapat diandalkan dan akurat.
3. Pihak pengelola objek wisata menghadapi tantangan dalam memberikan informasi yang memadai kepada pengunjung karena keterbatasan tenaga.

Beberapa masalah di atas yang membuat penulis menggunakan kecerdasan buatan dan *natural language processing* untuk membuat *chatbot* yang diharapkan mampu memberikan informasi secara jelas mengenai objek-objek wisata yang ada di Jawa Tengah.

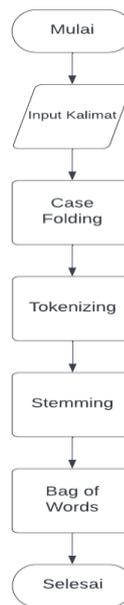
### 2. Data Acquisition

*Data acquisition* adalah proses pengumpulan data untuk dianalisis atau diproses, yang dapat dilakukan secara manual atau otomatis tergantung pada jenis data yang akan dikumpulkan dan sumber data yang dapat diakses. Penelitian ini menggunakan data primer dari kuesioner untuk mendapatkan pertanyaan (*patterns*) dari responden, sedangkan *responses chatbot* didapat berdasarkan *website* resmi dari pariwisata Provinsi Jawa Tengah. *Dataset* ini mencakup 2072 *tag*, 114506 *patterns*, dan 2072 *responses*, yang mencakup 507 objek wisata di 34 kabupaten atau kota. *Dataset* ini dibuat secara manual menggunakan perangkat lunak *Microsoft Visual Studio Code*.

### 3. Data Exploration

*Data exploration* adalah proses penyelidikan dan penjelahan data secara sistematis dengan tujuan untuk memahami struktur, pola dan hubungan dalam data. EDA

(*Exploratory Data Analysis*), yang diperkenalkan oleh John Tukey seorang ahli statistik, adalah sebuah pendekatan analisis data yang sistematis mengeksplorasi data untuk memahami struktur dan karakteristiknya. Pada tahap ini penulis melakukan visualisasi dan *preprocessing* data. Visualisasi data digunakan untuk mengetahui jumlah *patterns*, *tags*, dan *responses*. Ukuran dan kompleksitas *dataset* dapat ditentukan dengan mengetahui jumlah *patterns*, *tags*, dan *responses*. Jika terdapat banyak pola dan jawaban (*responses*), maka *dataset* akan menjadi kaya dan *chatbot* akan menjadi lebih responsif. Pemrograman *Python* digunakan di Google Colab untuk visualisasi data. Langkah selanjutnya adalah membersihkan data teks dari karakter tertentu atau item lain yang dapat mengganggu proses pemodelan. *Case folding*, *tokenizing*, *stemming*, *Bag of Words* digunakan pada tahap *preprocessing*, yang merupakan metode *Natural Language Processing* (NLP). Tahapan dalam *preprocessing* digambarkan pada gambar 3.



Gambar 3 Flowchart *Preprocessing Data* dalam EDA

#### 4. Modelling

Tahap ini merupakan tahap inti dimana proses pembuatan algoritma dalam bahasa pemrograman tertentu sebagai metode machine learning dibuat. Pemilihan model bergantung pada jenis masalah, ketersediaan data, dan hasil yang diinginkan. Setelah pelatihan, model mempelajari pola, memprediksi, atau memecahkan masalah yang ditargetkan menggunakan data pelatihan. Berikut adalah beberapa *hyperparameter* yang digunakan dalam membuat sebuah model.

1. *Epoch* adalah siklus pengulangan dalam pembelajaran *neural network* yang bertujuan untuk mencapai nilai *error* dan akurasi yang optimal dengan cara belajar secara berulang-ulang dari data pelatihan[7].
2. *Batch Size* adalah jumlah sampel pelatihan yang digunakan dalam satu kali iterasi[14].
3. *Learning Rate* adalah parameter pelatihan, digunakan untuk menghitung nilai koreksi bobot. Nilai yang terlalu kecil menyebabkan waktu pelatihan yang lebih

lama, sementara nilai yang terlalu besar menyebabkan pembelajaran menjadi kurang optimal karena terlalu cepat dan proses *training* menjadi tidak stabil.

4. *Hidden Size* adalah jumlah unit atau neuron di lapisan tersembunyi model *neural network*. *Hidden size* adalah bagian penting dalam model *neural network* yang membantu model “belajar” dan “mengingat” informasi dari data yang diberikan.
5. *Accuracy* digunakan untuk mengukur keakuratan model dalam memprediksi kelas atau label secara akurat. Semakin besar nilai akurasi maka semakin baik.
6. *Loss* berfungsi untuk mengukur kinerja model, yang bertujuan untuk mengurangi nilai *loss* sehingga model dapat memberikan prediksi yang lebih baik dan akurat[15].

Pada penelitian ini, modeling algoritma yang digunakan pada penelitian ini adalah *neural network*. Fungsi algoritma dalam proyek ini adalah melakukan klasifikasi berdasarkan kata- kata yang dimasukkan oleh pengguna, mencocokkan dengan kalimat-kalimat pada pola yang sudah dibentuk, serta melakukan prediksi dengan jawaban yang sesuai dengan pola yang cocok. Pola-pola dalam *chatbot* dibentuk dengan manual, serta disimpan ke dalam sebuah file dengan format json. Pada tahap ini ditentukan hyperparameternya seperti jumlah *epoch*, *batch size*, *learning rate*, *hidden size*, dan lain-lain.

#### 5. *Evaluation*

*Evaluation* adalah proses untuk menilai kualitas atau keakuratan model atau algoritma. Evaluasi dilakukan dengan menguji model atau algoritma untuk mengetahui seberapa baik model atau algoritma tersebut. Setelah model dibuat, evaluasi model AI dilakukan untuk memastikan bahwa model AI yang telah dibangun dapat bekerja dengan baik pada data yang digunakan. Sebuah model *chatbot* membutuhkan *training* atau pelatihan agar dapat mendapatkan hasil yang sesuai atau akurat.

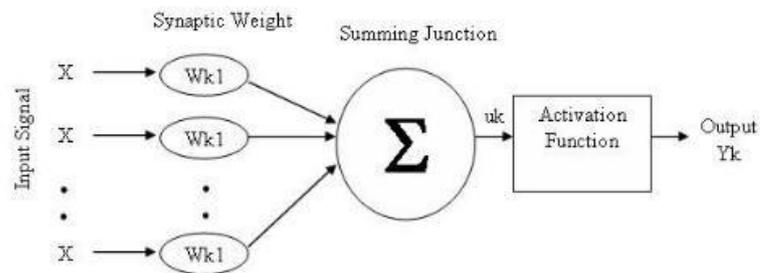
#### 6. *Deployment*

Jika model bekerja dengan baik dan merespon *chatbot* dengan benar, maka tahap *deployment* bisa dimulai. *Deployment* adalah proses implementasi AI pada sebuah aplikasi atau sistem sesuai dengan tujuan pembuatan produk sehingga diharapkan dapat memudahkan pengguna untuk mengakses aplikasi tersebut. Jika model bekerja dengan baik dan dapat merespon *chatbot* dengan benar, maka tahap *deployment* bisa dilakukan. Tujuan dari *deployment* adalah untuk menyebarkan aplikasi yang telah dikerjakan. Penulis melakukan *deployment* dalam bentuk sebuah website dan dapat dijalankan di *localhost*.

### 2.4 *Neural Network*

*Neural Network* adalah mendefinisikan jaringan saraf sebagai prosesor paralel yang menyimpan dan menghasilkan pengetahuan melalui pengalaman. Jaringan saraf tiruan memungkinkan komputer untuk melakukan tugas-tugas yang setara dengan manusia.

Mekanisme jaringan saraf melibatkan tiga bagian, yaitu *summing junction*, *activation function*, dan *output*, seperti yang terlihat pada gambar 4.



Gambar 4. Bentuk Model Dasar Neuron

1. Aliran sinyal digambarkan dalam gambar oleh tanda panah.
2. Sinyal input  $X_1, X_2, \dots, X_n$  berfungsi sebagai input untuk pembelajaran dan pengenalan objek.
3.  $W_1, W_2, \dots, W_n$  adalah bobot (*weight*) untuk sinyal yang sesuai. Bobot (*weight*) merupakan hubungan antar neuron-neuron.
4. *Summing junction* (fungsi penjumlahan) adalah simpul dalam jaringan saraf tiruan yang menjumlahkan sinyal input dari neuron sebelumnya.
5. *Activation function* (fungsi aktivasi) adalah sebuah proses pemetaan sinyal *output* oleh suatu neuron untuk menghasilkan *output*.
6.  $y$  (*output*) merupakan keluaran dari hasil pengenalan suatu objek.

Setiap *neuron* jaringan saraf melakukan beberapa operasi matematika untuk menghasilkan output.

### 3. HASIL DAN PEMBAHASAN

Pada hasil pembahasan penelitian ini, akan dibahas setiap proses pada *AI Project Life Cycle* untuk aplikasi *Chatbot* yang dikembangkan saat ini.

#### 1. Pengumpulan Data

Pada tahapan pengumpulan data, data dikumpulkan dalam penelitian ini dengan menggunakan metode observasi, yaitu melalui penyebaran kuesioner. Bagian *patterns dataset* menggunakan jawaban kuesioner dari responden, sedangkan informasi objek wisata diambil dari situs resmi pariwisata Provinsi Jawa Tengah. Berikut format penulisan dataset yang dibutuhkan oleh chatbot yang ditunjukkan pada gambar 5.

```

Welcome  () intents.json  X
E: > ina > materi dan tugas > skripsi > {} intents.json > [ ] intents > {} 247
1  {
2  }
3  {
4  }
5  {
6  }
7  {
8  }
9  {
10 }
11 {
12 }
13 {
14 }
15 {
16 }
17 {
18 }
19 {
20 }
21 {
22 }
23 {
24 }
25 {
26 }
27 {
28 }
29 {
30 }
31 {
32 }

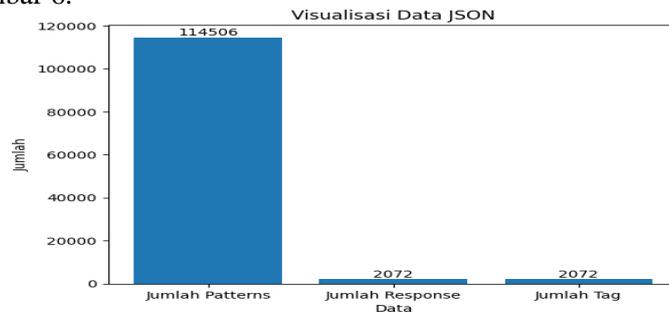
```

Gambar 5 Query dataset chatbot

Seperti yang ditunjukkan pada Gambar 5, data ini dibuat di Visual Studio Code dan disimpan dalam format JSON. Terdapat 2072 tags, 114506 patterns, dan 2072 responses yang digunakan pada dataset ini.

## 2. Data Exploration

Pada tahapan data Exploration, proses visualisasi data sederhana dan preprocessing data dilakukan. Proses visualisasi data yang dilakukan adalah untuk melihat banyaknya patterns yang bisa diambil berdasarkan tags dan responses yang akan menjadi dasar database dari aplikasi chatbot ini. Patterns menggambarkan pola pertanyaan yang akan diberikan chatbot kepada response, sedangkan response merupakan hasil jawaban dari patterns yang diberikan [16]. Hasil dari visualisasi dengan google colab disajikan pada gambar 6.



Gambar 6 Visualisasi Data patterns, response, dan tag

Setelah dilakukan proses visualisasi data guna melihat ukuran dan kompleksitas dataset secara keseluruhan, langkah berikutnya adalah dilakukan proses preprocessing data. Pada bagian ini, Preprocessing dibagi menjadi 3 tahap, yaitu proses case folding, tokenizing, dan stemming. Case folding merupakan tahapan pembersihan data teks yang mengubah format penulisan teks dari uppercase menjadi lowercase. Kemudian pada tahapan kedua dilakukan proses stemming, yaitu proses merubah seluruh stuktur kata pada data teks menjadi kata dasar. Pada tahapan ini diperlukan library untuk mendukung proses stemming dengan bantuan PorterStemmer. Kemudian langkah berikutnya adalah mengubah data teks menjadi pecahan token per kata yang disebut sebagai tokenization. Proses tokenization ini akan digunakan untuk proses modelling chatbot. Hasil dari preprocessing data ini ditampilkan pada gambar 7.

```
114506 patterns
2072 tags: ['Alamat_Agrowisata_Kaligua', 'Alamat_Agrowisata_Salib_Pt
1230 unique stemmed words: ['', '1912', '6', 'abadi', 'abdul', 'ad
```

Gambar 7 Hasil proses Cleaning Data

### 3. Modelling

Setelah dilakukan pembersihan *dataset*, tahap selanjutnya membuat data pelatihan dan *modelling* data dengan *neural network*. Terdapat beberapa tahapan untuk proses *modeling* ini. Pada tahap 1, dilakukan pembagian *data training* dan *data testing* dari *dataset chatbot* yang sudah dibersihkan. Melalui *X\_train* dan *y\_train* dilakukan proses pembagian data berdasarkan jumlah data dari tag dan *pattern\_sentence* untuk semua dataset yang sudah dibuat.

```
# create training data
X_train = []
y_train = []
for (pattern_sentence, tag) in xy:
    # X: bag of words for each pattern_sentence
    bag = bag_of_words(pattern_sentence, all_words)
    X_train.append(bag)
    # y: PyTorch CrossEntropyLoss needs only class labels, not one-hot
    label = tags.index(tag)
    y_train.append(label)

X_train = np.array(X_train)
y_train = np.array(y_train)
```

Gambar 8 Proses Data Training

Kemudian setelah dilakukan proses *training* data, dilakukan *setup* model untuk *neural network* dengan menggunakan library *Pytorch*. Pada gambar 9, dibuat menjadi 3 lapisan layer *hidden* dengan menggunakan aktivasi *RELU*.

```
class NeuralNet(nn.Module):
    def __init__(self, input_size, hidden_size, num_classes):
        super(NeuralNet, self).__init__()
        self.l1 = nn.Linear(input_size, hidden_size)
        self.l2 = nn.Linear(hidden_size, hidden_size)
        self.l3 = nn.Linear(hidden_size, num_classes)
        self.relu = nn.ReLU()

    def forward(self, x):
        out = self.l1(x)
        out = self.relu(out)
        out = self.l2(out)
        out = self.relu(out)
        out = self.l3(out)
        # no activation and no softmax at the end
        return out
```

Gambar 9 Proses Modeling Neural Network

Setelah dilakukan proses *modeling* dengan *neural network*, maka dari hasil *training* ini akan dilakukan *training* dan *testing* untuk melihat hasil dari *chatbot* ini apakah sudah berhasil atau belum. Dan pada eksperimen ini, juga ada beberapa penyesuaian terhadap nilai parameter yang digunakan pada proses *neural network*. Penyesuaian ini digunakan untuk menentukan proses *training* dan *testing* sebanyak *n* kali melalui *epoch* dan *learning rate*. Untuk penyesuaian parameter diberikan pada Gambar 10.

```
# Hyper-parameters
num_epochs = 200
batch_size = 8
learning_rate = 0.001
input_size = len(X_train[0])
hidden_size = 1024
output_size = len(tags)
print(input_size, output_size)
```

Gambar 10 Setting Hyperparameter untuk Modeling.

Untuk proses pelatihan *dataset chatbot*, dilakukan sebanyak 200 kali percobaan dengan nilai *learning rate* 0,001. Setiap *epoch*, model akan memproses data latihan dalam *batch-batch* kecil menggunakan *DataLoader*. Selama setiap *batch*, langkah-langkah berikut dilakukan:

- a. Data latihan (*all\_words* dan *tags*) dimuat dan dipindahkan ke perangkat (GPU jika tersedia, jika tidak, CPU).
- b. Model diberikan data latihan dan menghitung prediksi untuk setiap data.
- c. Akurasi model dihitung berdasarkan perbandingan antara prediksi dan label sebenarnya.
- d. Model diperbarui menggunakan *optimizer* Adam untuk mengurangi *error* atau *loss*.

Setelah satu *epoch* selesai, akurasi keseluruhan model dihitung berdasarkan semua prediksi dan label selama pelatihan.

#### 4. Evaluation

Setelah melakukan proses *training* langkah selanjutnya adalah menguji performa dari model yang digunakan, seperti terlihat dalam gambar 11 di bawah ini.

No	Epoch	Batch Size	Learning Rate	Hidden Size	Accuracy Data Latih	Loss Data Latih
1	100	8	0.01	8	46.4	3.75
2	100	8	0.001	8	99.14	0
3	200	8	0.001	8	97.03	0
4	200	8	0.001	1024	99.92	0
5	100	8	0.001	1024	98.14	3.51

Gambar 11 Nilai Akurasi dan *Loss*

Gambar 11 menunjukkan bahwa hasil data *training* memiliki nilai akurasi sebesar 99.92% dengan *loss* 0%. Sedangkan pada *learning\_rate* yang digunakan, untuk mencapai nilai akurasi yang lebih baik, menggunakan *learning\_rate* paling kecil yaitu 0,001 dan berdasarkan perbandingan *learning\_rate*, jika *learning\_rate* memiliki nilai yang lebih kecil, maka hasil ketepatan di dalam proses pembelajaran akan semakin baik tetapi di dalam proses belajar dan waktu proses akan membutuhkan waktu yang lebih. Sedangkan untuk *hidden\_size*, diperoleh hasil maksimal jika menggunakan *hidden\_size* sebesar 1024. Dengan nilai *hidden\_size* yang lebih besar, maka hasil dari prediksi akan memiliki kinerja yang baik, tetapi akan bergantung dengan waktu proses dan modeling yang lebih kompleks. Dari hasil pengujian ini, maka pengaturan untuk modeling Neural Network menggunakan nilai nomor 4 agar menghasilkan prediksi chatbot untuk jawaban yang lebih baik.

#### 5. Deployment and Implementation

Tahap *deployment* dapat dimulai jika model berkinerja dengan baik dan menjawab *chatbot* dengan benar. Tujuan dari *deployment* adalah untuk menyebarkan aplikasi yang telah dikerjakan agar mudah diakses oleh pengguna. Penulis melakukan *deployment* dan dijalankan di *localhost*.

##### 1. Analyze

###### 1. Analisis Kebutuhan Isi/Konten

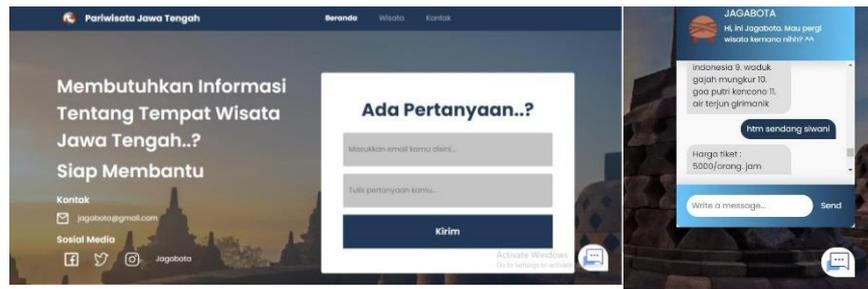
Pada tahap ini, melakukan analisis tentang kebutuhan isi atau konten yang harus disajikan dalam aplikasi *chatbot*. Isi atau konten berfokus pada apa yang pengguna butuhkan dalam *chatbot*. Berikut ini analisisnya:

1. Data dan informasi yang diberikan pengguna terkait pertanyaan atau permintaan dalam menggunakan aplikasi *chatbot* wisata.

2. Temuan analisis kebutuhan membantu dalam menentukan fungsi situs web *chatbot*. Situs web ini memiliki fitur-fitur yaitu menu beranda, menu wisata, menu kontak, dan *chatbot*.
2. Analisis Kebutuhan *Software*  
Melakukan identifikasi dan memahami kebutuhan yang harus dipenuhi oleh *chatbot* berbasis web. Beberapa analisis yang dilakukan antara lain:
  1. Menentukan bagaimana *chatbot* akan berkomunikasi dengan pengguna.
  2. Mempertimbangkan bahasa yang digunakan oleh *chatbot*.
  3. Mempertimbangkan tata letak dan tampilan antarmuka *chatbot* pada situs web.
  4. Membuat daftar kebutuhan teknis *chatbot*, seperti bahasa pemrograman, *platform* pengembangan, dan teknologi implementasi.

## 2. Design

Pada tahapan ini, dilakukan proses desain aplikasi web untuk penerapan *chatbot* pada portal pariwisata Jawa Tengah melalui *localhost*. Pada bagian ini juga, dibagi menjadi 2 jenis desain, yaitu desain *wireframe* dari aplikasi yang akan dibuat dan desain aplikasi yang sudah jadi. Untuk desain dari aplikasi yang sudah *dideploy* ditampilkan pada gambar 12.



Gambar 12 Desain aplikasi chatbot pariwisata Jawa Tengah

## 4. KESIMPULAN DAN SARAN

Kesimpulan yang dapat diambil dari penerapan algoritma *neural network* dalam pengembangan aplikasi *chatbot* objek wisata Jawa Tengah, antara lain:

1. Pengembangan *chatbot* dilakukan dengan cara melatih model menggunakan algoritma *neural network* dengan *input size* sebesar 1230, *hidden size* sebesar 1024, dan *output size* sebesar 2072. Menghasilkan akurasi sebesar 99.92% dan *loss* sebesar 0.
2. Pengembangan *chatbot* yang mengintegrasikan algoritma *neural network* dapat merespon pengguna dalam waktu 2 detik dibandingkan *chatbot* dari situs resmi pariwisata di Jawa Tengah yang merespon pengguna dalam waktu 56 detik.

Berdasarkan penelitian ini, terdapat beberapa saran untuk pengembangan aplikasi *Chatbot* Objek Wisata Jawa Tengah menggunakan *neural network*:

1. Aplikasi *chatbot* dapat diintegrasikan dengan teknologi *Augmented Reality* (AR), memungkinkan pengguna untuk melihat panduan visual atau informasi yang ditampilkan dunia nyata melalui kamera ponsel.
2. Aplikasi *chatbot* dapat diintegrasikan dengan aplikasi lain yang relevan dalam industri pariwisata, seperti aplikasi pemesanan tiket atau aplikasi navigasi.

DAFTAR PUSTAKA

- [1] B. P. S. P. J. Tengah, “Badan Pusat Statistik Provinsi Jawa Tengah.” [Online]. Available: <https://jateng.bps.go.id/indicator/153/613/1/luas-wilayah-menurutkabupaten-kota.html>
- [2] R. Ridwan, H. Lubis, and P. Kustanto, “Implementasi Algoritma Neural Network dalam Memprediksi Tingkat Kelulusan Mahasiswa,” *J. Media Inform. Budidarma*, vol. 4, no. 2, pp. 286–293, 2020, doi: 10.30865/mib.v4i2.2035.
- [3] A. K. Sari, F. Febryansyah, M. Y. Febriansyah, A. Wahyu, A. Saifudin, and I. Kusyadi, “Pengembangan Aplikasi Chat Bot untuk Rekomendasi Wisata,” *Inform. Univ. Pamulung*, vol. 6, no. 3, pp. 550–554, 2021, [Online]. Available: <http://openjournal.unpam.ac.id/index.php/informatika>
- [4] D. Rahayu, M. Mukrodin, and R. C. S. Hariyono, “Penerapan Artificial Intelligence Dalam Aplikasi Chatbot Sebagai Helpdesk Objek Wisata Dengan Permodelan Simple Reflex-Agent (Studi Kasus : Desa Karangbenda),” *Smart Comp Jurnalnya Orang Pint. Komput.*, vol. 9, no. 1, pp. 7–21, 2020, doi: 10.30591/smartcomp.v9i1.1813.
- [5] R. C. Utama, F. Fauziah, and R. T. Komalasari, “Aplikasi Chatbot Berbasis Teks Menggunakan Algoritma Naive Bayes Classifier FAQ GrabAds,” *STRING (Satuan Tulisan Ris. dan Inov. Teknol.*, vol. 6, no. 1, pp. 90–97, 2021, doi: 10.30998/string.v6i1.9919.
- [6] N. A. Purwitasari and M. Soleh, “Implementasi Algoritma Artificial Neural Network Dalam Pembuatan Chatbot Menggunakan Pendekatan Natural Language Parocessing,” *J. IPTEK*, vol. 6, no. 1, pp. 14–21, 2022, doi: 10.31543/jii.v6i1.192.
- [7] F. Mustakim, F. Fauziah, and N. Hayati, “Algoritma Artificial Neural Network pada Text-based Chatbot Frequently Asked Question (FAQ) Web Kuliah Universitas Nasional,” *J. JTik (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 5, no. 4, pp. 438–447, 2021, doi: 10.35870/jtik.v5i4.261.
- [8] E. Yunitasari, A. Triningsih, and R. Pradanie, “Analysis of Mother Behavior Factor in Following Program of Breastfeeding Support Group in the Region of Asemrowo Health Center Surabaya,” *NurseLine J.*, vol. 4, no. 2, pp. 94–102, 2019, doi: 10.19184/nlj.v4i2.11515.
- [9] A. Muklis, “Pengaruh Harga dan Kualitas Produk Terhadap Keputusan Pembelian Sepeda Motor Yamaha Tipe MX King (Studi pada Pengguna Sepeda Motor Yamaha Tipe MX King Di Sangatta),” *J. Adm. Bisnis Fisipol Unmul*, vol. 9, no. 4, pp. 314–321, 2021, doi: 10.54144/jadbis.v9i4.6005.
- [10] W. Ertel, *Introduction to Artificial Intelligence*, 2nd ed., vol. 51, no. 1. Weingarten: Springer International Publishing AG, 2017. doi: 10.4324/9781003218913-2.
- [11] C. C. Aggarwal, *Neural Networks and Deep Learning*. New York: Springer, 2022. doi: 10.1007/978-3-031-03758-0\_5.
- [12] S. Raj, *Building Chatbots with Python*, 1st ed. Bangalore: Apress, 2019. doi: 10.1007/978-1-4842-4096-0.
- [13] P. Goyal, S. Pandey, and K. Jain, *Deep Learning for Natural Language Processing*, vol. 190. Bangalore: Apress, 2021. doi: 10.1007/978-981-16-0882-7\_45.
- [14] 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.
- [15] P. Kim, *MATLAB Deep Learning*. Seoul: Apress, 2017. doi: 10.1007/978-1-4842-2845-6.
- [16] M. Bowles, *Machine Learning in Python*, vol. 6, no. 1. Indianapolis: Wiley, 2017. [Online]. Available: <http://repositorio.unan.edu.ni/2986/1/5624.pdf%0Ahttp://fiskal.kemenkeu.go.id/ejournal%0Ahttp://dx.doi.org/10.1016/j.cirp.2016.06.001%0Ahttp://dx.doi.org/10.1016/j.powtec.2016.12.055%0Ahttps://doi.org/10.1016/j.ijfatigue.2019.02.006%0Ahttps://doi.org/10.1>