

# 3703-9739-1-SM

*by jiko unkhair*

---

**Submission date:** 05-Nov-2021 05:21AM (UTC-0700)

**Submission ID:** 1693842285

**File name:** 3703-9739-1-SM.docx (215.27K)

**Word count:** 3704

**Character count:** 23272

## APLIKASI CHATBOT UNTUK FAQ AKADEMIK DI IBI-K57 DENGAN LSTM DAN PENYEMATAN KATA

Astried Silvanie<sup>1</sup>, Rino Subekti<sup>2</sup>

<sup>1,2</sup>Program Studi Informatika, Fakultas Ilmu Komputer, Institut Bisnis & Informatika (IBI) Kosgoro 1957  
Email: <sup>1</sup>astried@ibi-k57.ac.id, <sup>2</sup>rino.subekti@ibik57.ac.id

(Naskah masuk: November 2021, diterima untuk diterbitkan: ...)

### Abstrak

Dalam kegiatan perkuliahan khususnya untuk mendapatkan informasi akademik, teknologi informasi dan komunikasi sudah bisa memfasilitasi hal ini untuk dilakukan secara online. Tetapi jika ada pertanyaan dari mahasiswa maka peran karyawan masih diperlukan sebagai dukungan kustomer. Jika dilihat dari kemajuan dalam kecerdasan buatan khususnya pembelajaran mesin sangat dimungkinkan untuk membuat aplikasi cerdas yang dapat menjawab pertanyaan-pertanyaan secara otomatis tanpa bantuan manusia. Penelitian ini bertujuan membuat, melatih dan mengimplementasikan aplikasi FAQ *chatterbot* cerdas yang bisa memahami maksud dari kalimat kemudian menjawabnya. Algoritma yang digunakan adalah jaringan syaraf tiruan *Long Short Term Memory* (LSTM) dengan bantuan penyisipan kata menggunakan vektor kata yang sudah terlatih dalam model *Continuous Bag Of Word* (CBOW). Metrik kinerja pembelajaran mesin yang digunakan adalah pengoptimal *Adaptive Moment* (adam), fungsi kerugian *categorical crossentropy* dan *accuracy*. Dari pelatihan didapatkan nilai *accuracy* sebesar 98.92% dan nilai *RMSE* (*Root Mean Square Error*) sebesar 2.18%. Implementasi dibangun dalam arsitektur *client-server*. Di sisi *client* dibangun aplikasi telepon gengam berbasis *Android* dengan bahasa pemrograman *kotlin*. Aplikasi ini digunakan pengguna akhir untuk bertanya mengenai masalah akademik. Di sisi *server* dibangun aplikasi dengan kerangka *web Flask* sebagai aplikasi *python* yang menjalankan input-proses-ouput menggunakan model *JST*. Aplikasi ini juga berfungsi memilih pasangan jawaban yang tepat dan kemudian mengirimkannya ke aplikasi pengguna akhir.

**Kata kunci:** *chatbot, LSTM, jaringan syaraf tiruan, Flask*

## CHATBOT APPLICATION FOR ACADEMIC FAQ IN IBI-K57 WITH LSTM AND WORD EMBEDDING

### Abstract

In lecturer activities, especially to get information about academic, the information and communication technology can facilitate this to be done online. But if there are questions from students then the role of employees is still needed as a customer support. When we see the research progress in artificial intelligence, especially machine learning, it is possible to create an intelligent application that can answer question automatically without human assistance. This research aims to create, train and then implement a smart chatter robot for FAQ that can understand the intent of the sentence and then answer it. The algorithm used is a Long Short Term Memory (LSTM) neural network with the help of word embedding using a word vector that is already trained in the Continuous Bag Of Word (CBOW) model. The machine learning performance metrics used are Adaptive Moment (adam) optimizers, categorical crossentropy loss functions and accuracy. From the training obtained an accuracy value of 98.92% and an RMSE (Root Mean Square Error) value of 2.18%. The implementations are built into a client-server architecture. On the client side we built an Android-based application with the kotlin programming language. This application is used by end users to ask questions about academic issues. On the server side we built applications with the Flask web framework a python applications that run input-process-ouput using the ANN model. The applicatian also works to choose the right answer pair and then send it back to the end user.

**Keywords:** *chatbot, LSTM, artificial neural network, Flask*

18  
1. PENDAHULUAN

Infrastruktur teknologi informasi dan komunikasi (TIK) sudah memungkinkan universitas menyediakan sistem secara online sebagai pendukung kegiatan perkuliahan dan beberapa aktivitas akademiknya [1]. Apalagi di masa pandemi ini peran TIK sangat membantu dalam kegiatan belajar mengajar dan pemrosesan informasi akademik. Tetapi sistem online ini belum tentu mempunyai fungsi *Frequently Asked Question* (FAQ) yang dapat menjawab secara otomatis. Jika ada pertanyaan baik dari mahasiswa atau dosen masih dilakukan oleh karyawan yang berperan sebagai pendukung kustomer dengan waktu kerja terbatas. Sistem FAQ secara otomatis ini dapat saja dibuat dengan bantuan komputer. Kelebihannya adalah komputer dapat mengenal lelah dan batasan waktu, sehingga bisa diakses kapan saja dan dimana saja [1]. Pertanyaannya bagaimana kita membuat sistem *Frequently Asked Question* (FAQ) yang dapat memahami maksud pertanyaan dan menjawabnya secara otomatis tanpa perlu keberadaan manusia. *Chatbot*, atau dikenal juga dengan *chat robot*, adalah agen perangkat lunak yang dapat mensimulasikan percakapan manusia melalui teks atau pesan suara [2]. Seperti layaknya asisten virtual, *chatbot* membantu menjawab beberapa pertanyaan pengguna dan kemudian merespon sesuai dengan apa maksud dari pertanyaan tersebut. Sebuah FAQ *chatbot* adalah implementasi yang membandingkan pola di kalimat, dimana urutan kalimat tersebut dikenali dan kemudian merespon dengan pola respon yang tersimpan [3].

Teknologi *chatbot* yang termasuk kedalam kategori *Natural Language Processing* (NLP). Penelitian berkaitan dengan klasifikasi bahasa dengan NLP sudah banyaksudah cukup. Ayanouz dkk membuat *chatbot* sebagai asisten kesehatan virtual [2]. Penggunaan *Natural Language Processing* (NLP) dan *Deep Learning* untuk membuat *chatbot* yang cerdas [4]. Sethi membuat *chatbot* untuk merespon *Frequently Asked Question* (FAQ) [3]. Onan dkk mengidentifikasi sarkasme di media sosial menggunakan jaringan saraf tiruan dan penyisipan kata [5]. Tuhaney dkk membandingkan metode Naive Bayes dan Support Vector Machine untuk klasifikasi bahasa indonesia, jawa dan ambon [6].

2. METODE PENELITIAN

25  
2.1. Identifikasi Masalah

Dalam penelitian ini masalah yang ingin dijawab adalah (1) bagaimana membuat aplikasi FAQ *chatbot* cerdas untuk menjawab pertanyaan-pertanyaan akademik umum menggunakan algoritme pembelajaran mesin, dan (2) bagaimana

mengimplementasikan FAQ *chatbot* tersebut dalam aplikasi telepon genggam dengan sistem operasi *Android*.

2.2. Klasifikasi Maksud

Jika diketahui kalimat  $s \in X$  dari sebuah kalimat dialog, di mana  $X$  adalah ruang ucapan dan  $K$  adalah jumlah dari kalimat  $s$ . Maka dapat kita modelkan seperti dibawah ini.

$$S = \{s_i | 1 \leq i \leq K\} \quad (1)$$

Satu set *Intent Class* sebanyak  $j$  didefinisikan sebagai berikut:

$$C = \{C_i | 1 \leq i \leq J\} \quad (2)$$

Satu set pelatihan  $S$  dengan label maksud:

$$\{s_i, c_i\}_{i=1}^N, \text{ di mana } (s, c) \in X \times C \quad (3)$$

Setiap kalimat  $s$  akan diberi label tunggal atau dengan kata lain klasifikasi label tunggal, di mana setiap berpasangan dengan satu dan hanya satu elemen dari  $C$ .

2.3. Word Embedding

*Word Embedding* atau penyematan kata adalah representasi vektor bernilai bilangan *real* untuk kosa kata berukuran tetap yang sebelumnya telah ditentukan dari kumpulan dari korpus besar yang tidak berlabel [7]. Penyematan kata digunakan dalam pemrosesan bahasa alami modern, seperti dalam analisis semantik, pencarian informasi, penguraian ketergantungan, penjawab pertanyaan, dan terjemahan mesin [8]. Salah satu model penyematan kata yaitu *Continuous Bag Of Word* (CBOW). Model CBOW mencoba memahami konteks dari kata-kata dan mengambil konteks ini sebagai masukan. Kemudian model ini akan mencoba untuk memprediksi apakah kata-kata ini secara kontekstual akurat [9].

Penelitian ini menggunakan vektor kata yang sudah dilatih oleh Kyubyong pada model *Word2vec* dari sumber teks Wikipedia. Vektor kata ini diunduh secara publik dari github Kyubyong (url: <https://github.com/Kyubyong/wordvectors>) dengan nama file *id.tsv*. Vektor ini terdiri atas 30.048 kata dalam bahasa indonesia. Setiap kata direpresentasikan sebagai vektor berukuran  $1 \times 300$ . Sebagai contoh kita ambil kata pertama yaitu "yang" direpresentasikan sebagai vektor berikut ini:

$$\text{yang} = [X_0, \dots, X_{299}] \quad (4)$$

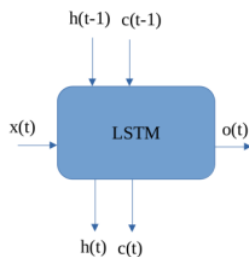
$$\text{yang} = [-0.074976, \dots, 0.0068165] \quad (5)$$

2.3. Long Short-Term Memory (LSTM)

Dalam jaringan syaraf tiruan tradisional, semua input dan output bersifat independen satu sama lain,

hal ini kurang tepat saat bekerja dengan informasi yang tersusun berurutan. Memprediksi kata berikutnya dalam sebuah kalimat membutuhkan pengetahuan urutan kata-kata dari kalimat sebelumnya. Salah satu model dari jaringan saraf *Recurrent Neural Network* (RNN), yaitu *Long Short Term Memory* (LSTM), adalah model yang paling tepat untuk memproses kalimat berurutan, dan biasa digunakan untuk proses kategorisasi teks dan terjemahan mesin [10]. *Long Short-Term Memory* (LSTM) adalah jaringan syaraf tiruan *Recurrent Neural Network* (RNN) di mana adanya gerbang yang digunakan untuk menghilangkan masalah gradien pada RNN tradisional [5]. LSTM menggunakan data historis dan data saat ini bersamaan untuk membuat prediksi. Hal ini dapat dicapai dengan menggunakan sel memori, di mana semua informasi masa lalu yang relevan disimpan. Karena sel memori dikendalikan oleh 3 gerbang - gerbang input, lupa dan keluaran. Sel memutuskan informasi mana yang disimpan dan kapan unit mengakses informasi tersebut berdasarkan apakah operasi di gerbang terbuka atau tidak. LSTM memecahkan masalah ketergantungan jangka panjang yaitu LSTM berkinerja baik dalam kasus di mana informasi jalan kembali di masa lalu diperlukan untuk membuat prediksi masa depan. Jaringan saraf berulang dua arah atau *Bidirectional Recurrent Neural Network* (BRNN) melakukan pelatihan secara maju dan mundur ke dua jaringan syaraf tiruan (JST) berulang yang terpisah. Kedua JST tersebut kemudian dihubungkan ke lapisan luaran yang sama [11].

LSTM memiliki input  $x(t)$  yang dapat berupa output dari CNN atau urutan input secara langsung.  $h(t-1)$  dan  $c(t-1)$  adalah input dari LSTM pada waktu sebelumnya. Variabel  $o(t)$  adalah output dari LSTM untuk waktu saat ini. LSTM juga membangkitkan  $c(t)$  dan  $h(t)$  untuk dimasukkan ke LSTM pada langkah waktu berikutnya.



Gambar 1. Masukan dan luaran LSTM dalam satu langkah waktu tunggal

Perhitungan untuk masukan dan luaran dalam satu langkah waktu tunggal adalah sebagai berikut:

$$f_t = \sigma_t (W_f \times X_t + U_f \times h_{t-1} + b_f) \quad (6)$$

$$i_t = \sigma_t (W_i \times X_t + U_i \times h_{t-1} + b_i) \quad (7)$$

$$o_t = \sigma_t (W_o \times X_t + U_o \times h_{t-1} + b_o) \quad (8)$$

$$c'_t = \sigma_c (W_c \times X_t + U_c \times h_{t-1} + b_c) \quad (9)$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot c'_t \quad (10)$$

$$h_t = o_t \cdot \sigma_t (c_t) \quad (11)$$

## 2.5. Data Set

Proses pengumpulan data dilakukan dengan observasi dokumen dan wawancara dengan staf Prodi di Institut Teknologi Sepuluh Nopember (ITS) dan Informatika Kosgoro 1957. Yang dicari adalah informasi mengenai hal-hal yang berkaitan dengan aktivitas akademik secara umum. Setelah dikumpulkan, kemudian data-data ini disimpan dengan format  $\{s_i, c_i\}_{i=1}^N$  dimana S adalah kalimat dan C adalah kelas maksud. Data-data ini disimpan dalam basis data MySQL terpisah dalam dua tabel yaitu tabel *intent\_class* dan tabel *intent\_test*. Kedua tabel ini mempunyai format sama dan terdiri dari tiga kolom seperti di Tabel 1.

Tabel 1. Format dataset dalam basis data

Nama	Tipe Data	Deskripsi
id	int	Kunci utama yang bernilai berurutan mulai dari satu
kalimat	text	Kalimat yang akan diberikan label maksud
maksud	char(30)	Label maksud untuk kalimat

Data latih terdiri dari 464 baris dan data uji coba terdiri atas 255 baris. Jumlah kelas maksud yang akan diidentifikasi secara unik berjumlah 26 kelas. Kelas-kelas ini yaitu *apa\_kkp*, *apa\_remedial*, *apa\_skripsi*, *cara\_berhenti\_kuliah*, *cara\_kkp*, *cara\_kuliah*, *cara\_remedial*, *cara\_skripsi*, *cara\_uas*, *cara\_ujian*, *cara\_uts*, *def\_ipk*, *def\_nilai\_bagus*, *def\_nilai\_jelek*, *def\_semester*, *def\_siakad*, *def\_uas*, *def\_ujian*, *def\_uts*, *greeting*, *jadwal\_kuliah*, *kuliah\_online*, *lama\_kuliah*, *login\_siakad*, *syarat\_lulus*.

## 2.6. Tahapan Penelitian

Tahapan proses yang dilakukan dalam penelitian ini adalah sebagai berikut:

1. Pengumpulan data dalam bentuk teks.
2. Pra-pemrosesan teks.
3. Pembangunan model jaringan syaraf tiruan (JST).

4. Proses pelatihan JST yang akan menghasilkan model JST dengan bobot lengkap.
5. Pembangunan aplikasi *Android* dengan bahasa pemrograman python.
6. Pembangunan aplikasiback-end dengan python dan diimplementasikan di komputer server.

### 18 3. HASIL DAN PEMBAHASAN

#### 3.1. Pra-pemrosesan teks

Data diambil dari basis data kemudian dilakukan beberapa pra-proses sebelum dimasukkan ke JST. Berikut ini pra-proses yang dilakukan:

1. Transformasi kata dibuat dalam huruf kecil semua.
2. Melakukan proses *stemming* proses transformasi kata menjadi kata dasarnya (*root*). Proses ini menghilangkan semua imbuhan kata (*affixes*) seperti awalan kata (*prefixes*), sisipin kata (*infixes*), akhiran kata (*suffixes*) dan menghilangkan awalan dan akhiran kata (*confixes*).
3. Data diubah menjadi larik dan disimpan dalam pengubah. Data latih disimpan dalam pengubahkalimat dan maksud. Data uji cobadisimpan dalam pengubahkalimat\_test dan maksud\_test.
4. Proses tokenisasi, yaitu proses pemisahan teks tertentu menjadi potongan-potongan lebih kecil yang disebut token.
5. Pembuatan indeks kosakata berdasarkan frekuensi kata tersebut dimulai dari kata yang paling sering muncul. Jadi jika nilai indeks lebih rendah berarti kata tersebut sering muncul.
6. Penambahan nilai 0 di awal urutan kata agar memastikan bahwa semua urutan dalam daftar memiliki panjang yang sama, proses ini dikenal dengan *padding sequences*.
7. Transformasi kata-kata ke dalam nilai numerik kategorikal.
8. Kemudian dari numerik kategorikal, ditransformasi menjadi nilai biner 1 dan 0.
9. Pembacaan kata-kata yang akan disematkan dari file id.tsvlalu disimpan dalam pengubah embedding\_words dan embedding\_index.

#### 3.2. Pembuatan JST

Jaringan syaraf tiruan LSTM yang dibentuk terdiri dari beberapa lapisan. Berikut ini dijabarkan dari lapisan paling pertama sampai lapisan akhir.

1. Lapisan Pertama adalah vektor kata untuk penyematan.  
`model.add(Embedding(num_words, 300, trainable=False, input_length=train_sequences.shape[1], weights=[embedding_matrix])).`  
num\_words : jumlah kata unik dari data pelatihan. Argumen kedua menunjukkan ukuran

e-ISSN: 2656-1948

vektor penyematan yaitu 300. Argumen *input\_length*, menentukan ukuran setiap urutan input. Setelah JST dilatih, kita bisa mendapatkan bobot lapisan penyematan dan dapat dianggap sebagai tabel yang digunakan untuk memetakan bilangan bulat ke vektor penyematan.

2. Lapisan kedua adalah *Bi-Directional LSTM*.  
`model.add(Bidirectional(LSTM(300, return_sequences=True, recurrent_dropout=0.1, dropout=0.1), 'concat'))`.  
Lapisan ini merupakan model JST LSTM dengan 300 input sesuai dengan panjang vektor kata. *Dropout* adalah teknik pemilihan neuron secara acak untuk diabaikan selama pelatihan. Tujuannya adalah untuk menghindari *over-fitting*. Disini kita akan mengabaikan 10% neuron selama pelatihan.
3. Lapisan ketiga *Drop Out*.  
`model.add(Dropout(0.3))`  
Lapisan ini mengabaikan 30% neuron secara acak di lapisan ini.
4. Lapisan keempat LSTM.  
`model.add(LSTM(300, return_sequences=False, recurrent_dropout=0.1, dropout=0.1))`  
Lapisan ini berupa mode JST LSTM kedua.
5. Lapisan kelima *Drop Out*.  
`model.add(Dropout(0.3))`  
Lapisan ini mengabaikan 30% neuron secara acak di lapisan ini.
6. Lapisan keenam adalah fungsi aktivasi.  
`model.add(Dense(150, activation='relu'))`  
Lapisan ini menunjukkan 150 unit dalam lapisan tersembunyi ini. Fungsi aktivasi yang digunakan adalah aktivasi ReLU dapat dinyatakan dengan  $f(x)=\max(0,x)$  artinya nilai kurang dari 0 akan memotong sinyal input yang memiliki [12].
7. Lapisan ketujuh luaran dengan fungsi aktivasi.  
`model.add(Dense(classes.shape[0], activation='softmax'))`.

#### 3.3. Pelatihan JST

Pada penelitian ini pelatihan data diulang sebanyak 100 *epochs*. Jumlah *epoch* adalah hyperparameter yang menentukan berapa kali algoritma pembelajaran akan bekerja melalui seluruh dataset pelatihan. Dalam satu *epoch* setiap sampel dataset pelatihan memiliki kesempatan untuk memperbarui parameter model internal. *Epoch* terdiri dari satu atau lebih *batch*, dalam pelatihan ini ditentukan 64 *batch* untuk setiap *epoch*.

Model dikompilasi dengan pengoptimal *Adaptive Moment* (adam), fungsi kerugian *categorical\_crossentropy* dan metrik validasi *accuracy*. *Accuracy* adalah selisih dari nilai

yang dibaca benar oleh sistem dengan nilai yang memang benar sesungguhnya [6]. Fungsi *categorical cross entropy loss* adalah fungsi optimasi yang digunakan dalam kasus pelatihan model klasifikasi yang mengklasifikasikan data dengan memprediksi probabilitas apakah data milik satu kelas atau kelas lain.

Tabel 2. Metrik Performa

Metrik	Nilai
Accuracy	0.9892
<i>Categorical Crossentropy</i>	0.0172
RMSE	0.02188

Luaran dari pelatihan JST ini akan menjadi masukan untuk aplikasi *FAQ chatbot* berbasis *android* kita. Luaran ini terdiri dari empat file yaitu:

1. Model pembelajaran mesin yang disimpan dengan nama file *model\_ML.h5*.
2. Semua kelas maksud yang unik disimpan dengan nama file *kelasMaksud.pkl*.
3. Hasil tokenisasi dengan nama file *tokenisasi.pkl*.
4. Hasil pemetaan urutan sumber dengan panjang variabel ke vektor dengan nama file *label\_untuk\_encoder.pkl*

#### 3.4. Perancangan Aplikasi Berbasis *Android* dan Aplikasi *Back End* di server.

Kita membutuhkan dua aplikasi, yaitu aplikasi untuk pengguna akhir dan aplikasi *back end* untuk menjalankan proses klasifikasi dengan model JST. Aplikasi pengguna akhir dibangun khusus hanya untuk telepon genggam dengan sistem operasi *Android*. Bahasa pemrograman yang digunakan untuk aplikasi ini adalah  *Kotlin*. Kebutuhan fungsional yang harus dipenuhi oleh aplikasi *FAQ chatter bot* ini adalah:

1. Login aplikasi dengan masukan teks nama dan jurusan.
2. Aplikasi dapat menerima pertanyaan dari pengguna ke *ChatBot* melalui teks pesan.
3. *Chatbot* menerima pertanyaan tersebut kemudian melakukan koneksi ke *Cloud Server* dan mengirimkannya teks pertanyaan melalui API (*Application Programming Interface*).
4. *Chatbot* menerima jawaban dari server dan menampilkannya di kotak dialog.

Aplikasi *back end* di server adalah aplikasi dengan kode *python* yang akan memasukkan input ke model JST *model\_ML.h5* dan mengembalikan luaran yaitu kelas maksud yang teridentifikasi, probabilitas kelas tersebut dan respon yang sesuai

dengan kelas maksudnya. Kebutuhan fungsional yang harus dipenuhi oleh aplikasi di sisi server adalah:

1. *Cloud Server* menerima pertanyaan dari aplikasi pengguna dan kemudian menjalankan kode *python* untuk mencoba mencari apa maksud dari pertanyaan tersebut. Kalimat pertanyaan menjadi masukan ke model *JST LSTM* kita agar bisa dicari apa maksud pertanyaan tersebut. File yang digunakan di proses adalah *model\_ML.h5*, *kelasMaksud.pkl*, *tokenisasi.pkl* dan *label\_untuk\_encoder.pkl*.
2. Server dapat menemukan kelas maksud untuk pertanyaan tersebut atau pun tidak menemukannya. Nilai ini ditentukan dari seberapa besar probabilitas kelas maksud yang teridentifikasi. Disini kita akan menetapkan nilai batasan 80% untuk nilai probabilitas.
  - a. Jika ditemukan berarti nilai probabilitas  $\geq 80\%$ , maka server akan mencari pasangan jawaban untuk kelas tersebut sebagai respon ke aplikasi pengguna akhir.
  - b. Jika tidak ditemukan berarti nilai probabilitas  $< 80\%$ , maka server akan mengembalikan pesan khusus. Pesan khusus ini sejenis dengan pernyataan berikut "*Maaf, saya tidak memahami pertanyaan anda, bisa ulangi lagi?*", "*Saya tidak menemukan jawaban untuk pertanyaan ini. Pertanyaan ini akan saya simpan*". Kasus-kasus seperti ini akan menjadikannya balik. Pertanyaan-pertanyaan yang tidak dimengerti tersebut dimasukkan ke dalam tabel *intent\_baru* di *MySQL*. Data-data pada tabel *intent\_baru* ini akan diberikan label oleh teknisi data dan dimasukkan lagi ke proses pelatihan selanjutnya. Sehingga jika semakin banyak pertanyaan maka akan memperbanyak kosa kata baru untuk mesin. Banyaknya kosa kata akan membuat mesin semakin pintar. Ini merupakan ciri dari proses *supervised learning*.
3. Server mengirimkan hasil respon terdiri dari kelas maksud yang teridentifikasi beserta pasangan jawabannya ke aplikasi pengguna akhir melalui API. Hasil respon ini disimpan dalam format *JSON*.

Di sisi server kita membutuhkan aplikasi yang dapat menjalankan model JST kami. Karena model JST kami dibangun dengan bahasa pemrograman *python* maka kami membutuhkan kerangka aplikasi yang dapat menjalankan *python*. *Flask framework* adalah salah satu *web framework* yang dapat kami jadikan sebagai lapisan aplikasi di sisi server seperti yang dilakukan dalam penelitian A.Ahmed dkk dan A. Najwa dkk [13][14]. Seperti

Tabel 3. Contoh pertanyaan dan klasifikasi maksud yang dilakukan JST

Question & Kelas Maksud	Probabilitas
<b>Question:</b> "Assalamualakum, saya mau tanya mengenai pelaksanaan UTS" <b>Kelas Maksud:</b> 'cara_uts'	89.74%
<b>Question:</b> "Saya pengen tahu masalah cara pelaksanaan ujian bu" <b>Kelas Maksud:</b> 'cara_ujian'	99.99%
<b>Question:</b> "saya mau mengisi pulsa dong" <b>Kelas Maksud:</b> 'apa_remedial' Probabilitas dibawah 80% maka akan dikembalikan pesan khusus	39.76%

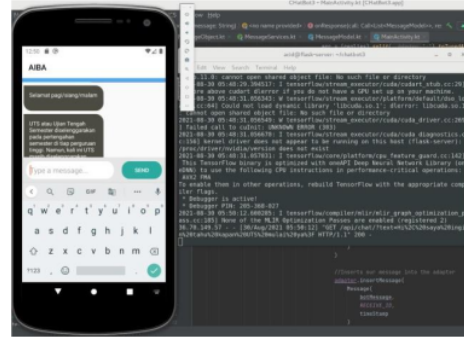
Berikut ini adalah logika untuk program aplikasi *back end* di server:

1. Load file *model\_ML.h5*, *kelasMaksud.pkl*, *tokenisasi.pkl* dan *label\_untuk\_encoder.pkl*.
2. Terima teks masuk dari aplikasi.
3. Lakukan proses *stemming*.
4. Lakukan proses tokenisasi.
5. Lakukan proses *pad-sequences*.
6. Kembalikan kelas maksud dengan nilai probabilitasnya.
7. Jika probabilitas  $\geq 80\%$  maka kembalikan pasangan jawaban untuk pertanyaan tersebut.
8. Jika probabilitas  $< 80\%$  maka kembalikan pesan khusus dan simpan pertanyaan tersebut dalam tabel *intent\_test*.

Berikut ini rancangan arsitektur antara aplikasi, internet dan server.



Gambar 2. Arsitektur Client Server yang dibangun



Gambar 3. Antarmuka aplikasi dan pengiriman teks ke server melalui API.

### 3.5. Uji Coba aplikasi

Pengujian aplikasi dilakukan dengan teknik uji coba *black-box*. Pengujian kotak hitam adalah pengujian berdasarkan spesifikasi persyaratan tanpa perlu memeriksa kode program [15]. Pengujian ini dilakukan berdasarkan sudut pandang pengguna akhir dengan kumpulan kasus uji coba yang dapat diprediksi. Tiga jenis kasus yang dicoba adalah (1) koneksi terputus atau ada kendala teknis, (2) Pertanyaan yang relevan dengan kelas maksud, (3) Pertanyaan yang tidak relevan dengan kelas maksud atau *Out Of Topic*. Setiap kasus diujikan dengan minimal 10 kali untuk kasus 1 dan 20 kali untuk kasus 2 dan 3.

Tabel 4. Kasus-kasus uji coba yang digunakan

Kasus Uji Coba	Respon
Jika koneksi terputus atau ada kendala teknis	Pesan pemberitahuan "sepertinya koneksi sedang ada kendala" di aplikasi pengguna
Pertanyaan yang relevan dengan kelas maksud	Aplikasi mengembalikan kelas maksud yang benar dengan probabilitas $\geq 80\%$ beserta jawaban yang sesuai
Pertanyaan yang tidak relevan dengan kelas maksud atau <i>Out Of Topic</i>	Aplikasi mengembalikan kelas maksud yang salah dengan probabilitas $< 80\%$ dan pesan khusus. Kemudian pertanyaan tersebut disimpan dalam basis data.

#### 4. Kesimpulan

Penelitian ini bertujuan membuat hal yaitu, (1) pembuatan *FAQ chatbot* untuk menjawab pertanyaan-pertanyaan akademik umum, (2) implementasi model ini ke dalam aplikasi yang bisa digunakan.

Pembuatan *chatbot* digunakan proses pembelajaran mesin dengan jaringan syaraf tiruan LSTM. Ditambahkan vektor kata sebagai penyematan kata dari korpus Wikipedia yang sudah dilatih dan dibagi dengan lisensi publik. Hasilnya adalah model *chatbot* yang dapat memahami maksud dari sebuah kalimat. Metrik kinerja pembelajaran mesin yang digunakan adalah pengoptimal *Adaptive Moment* (adam), fungsi kerugian *categorical crossentropy* dan *accuracy*. Dari penelitian didapatkan nilai *accuracy* sebesar 98.92% dan nilai *RMSE (Root Mean Square Error)* sebesar 2.18%.

Untuk tujuan kedua yaitu implementasi, dibangun dalam arsitektur *client-server*. Di sisi *client* dibangun aplikasi telepon gengam berbasis *Android* dengan bahasa pemrograman kotlin. Di sisi *server* dibangun aplikasi dengan kerangka web *Flask*. Ada beberapa alasan kenapa *Flask* bisa digunakan untuk aplikasi *back-end*. Yang pertama dan yang paling penting, dia bisa menjalankan kode python. Kedua dia bisa berjalan sebagai aplikasi website yang bisa diakses di jaringan komputer publik. Pengujian aplikasi menggunakan uji coba *black box* dilakukan oleh pengembang dan mahasiswa sebagai tester.

Penelitian selanjutnya akan mengembangkan *chatbot* agar dapat memperbaharui mode JST tanpa harus melakukan pelatihan dari awal agar lebih adaptif terhadap kosa kata baru.

#### 5. Daftar Pustaka

- [1] Y. Sumikawa, M. Fujiyoshi, H. Hatakeyama, and M. Nagai, "Supporting Creation of FAQ Dataset for E-Learning Chatbot," in *Intelligent Decision Technologies 2019, 2020*, pp. 3–13.
- [2] S. Ayanouz, B. A. Abdelhakim, and M. Benhmed, "A Smart Chatbot Architecture based NLP and Machine Learning for Health Care Assistance," Mar. 2020, doi: 10.1145/333123.3387897.
- [3] F. Sethi, "FAQ (Frequently Asked Questions) ChatBot for Conversation," *International Journal of Computer Sciences and Engineering Open Access Research Paper*, vol. 8, no. 10, 2020, doi: 10.26438/ijcse/v8i10.710.
- [4] T. Lalwani, S. Bhalotia, A. Pal, S. Bisen, and V. Rathod, "Implementation of a Chat Bot System using AI and NLP," *International Journal of Innovative Research in Computer Science & Technology*, vol. 6, no. 3, pp. 26–30, May 2018, doi: 10.21276/ijrcst.2018.6.3.2.

- [5] A. Onan and M. Tocoglu, "A Term Weighted Neural Language Model and Stacked Bidirectional LSTM Based Framework for Sarcasm Identification," *IEEE Access*, vol. PP, p. 1, Jan. 2021, doi: 10.1109/ACCESS.2021.3041344.
- [6] D. Tuhenay and E. Mailoa, "PERBANDINGAN KLASIFIKASI BAHASA MENGGUNAKAN METODE NAÏVE BAYES CLASSIFIER (NBC) DAN SUPPORT VECTOR MACHINE (SVM)," *Jurnal Informatika dan Komputer* Akreditasi KEMENRISTEKDIKTI, vol. 4, no. 2, 2021, doi: 10.33387/jiko.
- [7] B. Wang, A. Wang, F. Chen, Y. Wang, and C. J. Kuo, "Evaluating word embedding models: Methods and experimental results," *APSIPA Transactions on Signal and Information Processing*, vol. 8, Cambridge University Press, 2019, doi: 10.1017/ATSIP.2019.12.
- [8] J.-H. Wang, T.-W. Liu, X. Luo, and L. Wang, "An LSTM Approach to Short Text Sentiment Classification with Word Embeddings."
- [9] C. Xu, L. Xie, and X. Xiao, "A Bidirectional LSTM Approach with Word Embeddings for Sentence Boundary Detection," *Journal of Signal Processing Systems*, vol. 90, no. 7, pp. 1063–1075, Jul. 2018, doi: 10.1007/s11265-016-1289-8.
- [10] M. Aleedy, H. Shaiba, and M. Bezbradica, "Generating and Analyzing Chatbot Responses using Natural Language Processing," 2019. [online]. Available: [www.ijacsa.thesai.org](http://www.ijacsa.thesai.org)
- [11] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997, doi: 10.1109/78.650093.
- [12] H. Akbar, "KLASIFIKASI KANKER SERVIKS MENGGUNAKAN MODEL CONVOLUTIONAL NEURAL NETWORK (ALEXNET)," *Jurnal Informatika dan Komputer* Akreditasi KEMENRISTEKDIKTI, vol. 4, no. 1, 2021, doi: 10.33387/jiko.
- [13] A. A. Ahmed and G. Agunsoye, "A real-time network traffic classifier for online applications using machine learning," *Algorithms*, vol. 14, no. 8, Aug. 2021, doi: 10.3390/a14080250.
- [14] A. Najwa Arba'in, S. Letchumy, and M. Belaidan, "FAULT DETECTION AND PREDICTION IN THE SEMICONDUCTOR MANUFACTURING PROCESS," *International Journal of Management (IJM)*, vol. 11, no. 11, pp. 2023–2028, 2020, doi: 10.34218/IJM.11.11.2020.192.
- [15] S. Nidhra, "Black Box and White Box Testing Techniques - A Literature Review," *International Journal of Embedded Systems and Applications*, vol. 2, no. 2, pp. 29–50, Jun. 2012, doi: 10.5121/ijesa.2012.2204.

ORIGINALITY REPORT

16%

SIMILARITY INDEX

15%

INTERNET SOURCES

9%

PUBLICATIONS

7%

STUDENT PAPERS

PRIMARY SOURCES

1	<a href="http://ejournal.unkhair.ac.id">ejournal.unkhair.ac.id</a> Internet Source	3%
2	<a href="https://medium.com">medium.com</a> Internet Source	1%
3	Arda Sezen, Cigdem Turhan, Gokhan Sengul. "A Hybrid Approach for Semantic Image Annotation", IEEE Access, 2021 Publication	1%
4	<a href="http://www.ijeat.org">www.ijeat.org</a> Internet Source	1%
5	<a href="http://www.tandfonline.com">www.tandfonline.com</a> Internet Source	1%
6	<a href="http://aclanthology.org">aclanthology.org</a> Internet Source	1%
7	Submitted to University of Nottingham Student Paper	1%
8	<a href="http://jurnal.upnyk.ac.id">jurnal.upnyk.ac.id</a> Internet Source	1%

Submitted to British University in Egypt

9	Student Paper	1 %
10	<a href="http://www.hindawi.com">www.hindawi.com</a> Internet Source	1 %
11	<a href="http://raiith.iith.ac.in">raiith.iith.ac.in</a> Internet Source	1 %
12	Ashok M., Kumar Ramasamy, Snehitha G., Keerthi S.R.. "A Systematic Survey of Cognitive Chatbots in Personalized Learning Framework", 2021 Sixth International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), 2021 Publication	<1 %
13	<a href="http://www.sciencegate.app">www.sciencegate.app</a> Internet Source	<1 %
14	<a href="http://www.ncbi.nlm.nih.gov">www.ncbi.nlm.nih.gov</a> Internet Source	<1 %
15	<a href="http://bagiankecilmeteorologi.blogspot.com">bagiankecilmeteorologi.blogspot.com</a> Internet Source	<1 %
16	Sinarwati Mohamad Suhaili, Naomie Salim, Mohamad Nazim Jambli. "Service chatbots: A systematic review", Expert Systems with Applications, 2021 Publication	<1 %

17 Xinglin Xiao, Lei Wang, Qingchao Kong, Wenji Mao. "Social Emotion Cause Extraction from Online Texts", 2020 IEEE International Conference on Intelligence and Security Informatics (ISI), 2020  
Publication <1 %

---

18 doku.pub  
Internet Source <1 %

---

19 Ahmed Abdelmoamen Ahmed, Sheikh Ahmed. "A Real-Time Car Towing Management System Using ML-Powered Automatic Number Plate Recognition", Algorithms, 2021  
Publication <1 %

---

20 adoc.pub  
Internet Source <1 %

---

21 www.researchgate.net  
Internet Source <1 %

---

22 Yanliang Jin, Can Luo, Weisi Guo, Jinfei Xie, Dijia Wu, Rui Wang. "Text Classification Based on Conditional Reflection", IEEE Access, 2019  
Publication <1 %

---

23 bimbinganprogramskripsi.wordpress.com  
Internet Source <1 %

---

24 es.scribd.com  
Internet Source <1 %

---

25	Internet Source	<1 %
26	<a href="http://www.ijircst.org">www.ijircst.org</a> Internet Source	<1 %
27	<a href="http://www.journaltoocs.ac.uk">www.journaltoocs.ac.uk</a> Internet Source	<1 %
28	<a href="http://go.gale.com">go.gale.com</a> Internet Source	<1 %
29	<a href="http://www.scribd.com">www.scribd.com</a> Internet Source	<1 %
30	<a href="http://www.slideshare.net">www.slideshare.net</a> Internet Source	<1 %
31	"Sentimental Analysis and Deep Learning", Springer Science and Business Media LLC, 2022 Publication	<1 %

Exclude quotes Off

Exclude matches Off

Exclude bibliography Off