

Pengembangan Chatbot Menggunakan Deep Feed-Forward Neural Network sebagai Pusat Layanan Informasi Akademik

Ruvita Faurina¹, Dede Revanza², Ahmad Sopran³

Program Studi Informatika

Universitas Bengkulu

Bengkulu, Indonesia

e-mail: ¹ruvita.faurina@unib.ac.id, ²dederevanza@mail.com, ³ahmadsopran011602@gmail.com

Diajukan: 27 Desember 2022; Direvisi: 23 Mei 2023; Diterima: 24 Agustus 2023

Abstrak

Program studi informatika merupakan salah satu program studi unggulan di Universitas Bengkulu. Sebagai program studi unggulan, tentunya pelayanan terbaik untuk setiap elemen civitas akademika yang ada di Program Studi Informatika Universitas Bengkulu harus diperhatikan. Dalam hal pelayanan ini, adanya pusat layanan informasi akademik bagi civitas akademika di informatika sangat dibutuhkan. Namun, belum adanya pusat layanan informasi akademik yang bisa diakses dari mana dan kapan saja menjadi salah satu hambatan terlaksananya layanan informasi akademik. Pembatasan aktivitas yang memungkinkan terjadinya interaksi akibat pandemi covid-19 juga menjadi kendala. Sebagai upaya menindaklanjuti keterbatasan tersebut dikembangkanlah chatbot layanan informasi akademik program studi informatika untuk mengatasi kendala yang dihadapi. Dengan chatbot layanan informasi akademik ini pengguna dapat bertanya mengenai informasi layanan akademik kepada bot yang akan menjawab informasi yang dibutuhkan. Algoritma yang digunakan dalam penelitian ini adalah deep feed-forward neural network. Adapun knowledge dari chatbot ini berupa informasi mata kuliah, informasi dosen, dan informasi administrasi di Program Studi Informatika Universitas Bengkulu. Pada proses train model, data sebanyak 2059 dibagi menjadi 80% sebagai data train, 10% data validation, dan 10% data test pada epoch 450 dan batch size 100 didapat akurasi 94%, evaluasi Precision 0.88, recall 0.89, dan f1-score 0.88.

Kata kunci: Chatbot, Deep feed-forward neural network, Informatika.

Abstract

University of Bengkulu's informatics study program is one of its leading study programs. Informatics study program most provided the best service. In terms of this service, the existence of an academic information service center for the academic community in informatics is needed. However, the absence of an academic information service center that can be accessed from anywhere and at any time is one of the obstacles to the implementation of academic information services. As an effort to follow up on these limitations, an academic information service chatbot for the informatics study program was developed. With this academic information service chatbot, users can ask questions about academic service information to the bot which will answer the information needed. The algorithm used in this study is a deep feed-forward neural network. The knowledge from this chatbot is in the form of course information, lecturer information, and administrative information at the Informatics Study Program at the University of Bengkulu. In the train model process, the 2059 data is divided into 80% as data train, 10% data validation, and 10% test data on epoch 450 and batch size 100 to get 94% accuracy, evaluation Precision 0.88, recall 0.89, and f1-score 0.88.

Keywords: Chatbot, Deep Feed-Forward Neural Network, Informatika.

1. Pendahuluan

Pelayanan akademik merupakan suatu hal yang harus diperhatikan dan diberikan oleh stakeholder yang ada di universitas kepada seluruh civitas akademika yang ada di suatu universitas[1]. Beberapa pelayanan terkait informasi layanan akademik masih dilakukan secara konvensional yang menyebabkan efektivitas dan efisiensi penerimaan informasi layanan akademik masih kurang. Kendala berupa keterbatasan jumlah petugas mengurangi kepuasan terhadap tersampainya informasi kepada civitas akademika Program Studi Informatika Universitas Bengkulu. Sehingga ketika terjadi kendala berupa

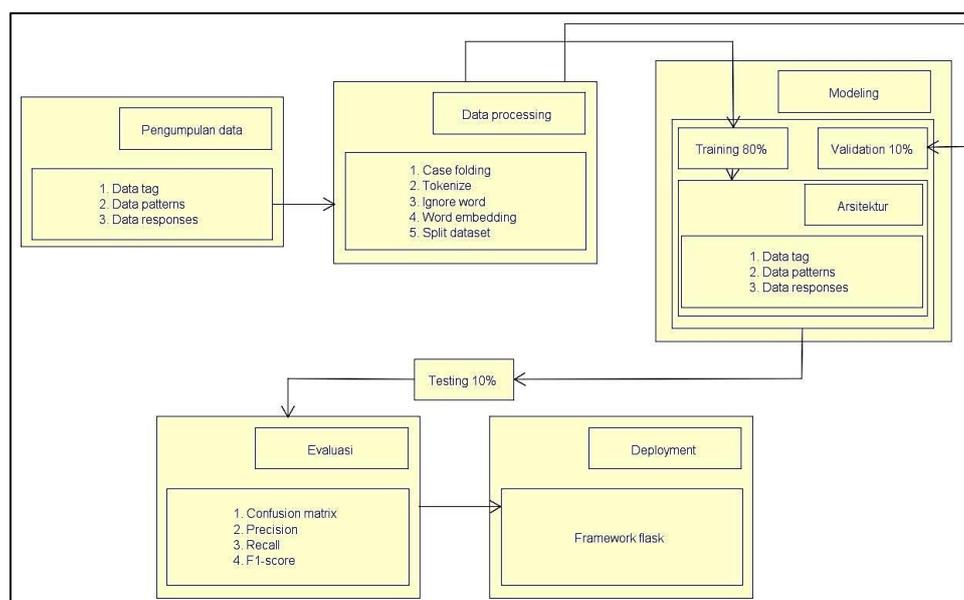
petugas yang sakit, cuti, atau tidak datang ke kantor masih menjadi kendala. Selain itu, kekurangan tersebut juga diakibatkan karena dengan metode yang masih konvensional memiliki keterbatasan ruang dan waktu. Keterbatasan tersebut mengakibatkan pihak yang membutuhkan informasi hanya bisa mendapatkannya dengan datang langsung ke kantor pada waktu jam kerja. Dengan adanya pusat informasi yang bisa diakses secara daring memungkinkan orang yang membutuhkan informasi untuk mendapatkan informasi yang dibutuhkan dengan lebih mudah. Sebagai upaya penyediaan layanan informasi, ada banyak media yang bisa digunakan seperti *chat*, telepon, *website*, dan masih banyak lagi. Upaya pengadaan media layanan layanan tersebut dilakukan untuk menjadi wadah komunikasi internal maupun eksternal kampus[2]. Program Studi Informatika Universitas Bengkulu sudah menggunakan *website* program studi sebagai salah satu media penyampaian informasi. Namun, konten *website* yang tidak diperbarui secara berkala dan tidak interaktif masih menjadi keterbatasan dalam penyampaian informasi.

Akibat dari pesatnya perkembangan teknologi, banyak hal yang ikut mengalami perkembangan seperti halnya pada penggunaan dan pengembangan *website*[3]. Salah satu media yang dapat digunakan untuk penyampaian informasi ini adalah dengan menggunakan *chatbot*. Namun, pelayanan informasi akademik di Program Studi Informatika Universitas Bengkulu dengan menggunakan *chatbot* masih belum diterapkan. Saat ini pengembangan *chatbot* sebagai media pusat pelayanan informasi di berbagai sektor kehidupan semakin banyak dilakukan. Hal ini terbukti dari banyaknya *chatbot* yang sudah diimplementasikan di berbagai aplikasi di dunia industri seperti Veronica milik Telkomsel, Indira milik indosat, dan masih banyak lagi. Selain itu di dunia pendidikan pun juga sudah ada yang menjadikan *chatbot* sebagai layanan informasi mereka seperti yang dikembangkan di Universitas Negeri Yogyakarta[4]. Oleh karena itu, sebagai solusi untuk mengatasi kendala terjadi dan untuk menambah efektivitas dan efisiensi layanan informasi akademik di program studi informatika dikembangkanlah sebuah *chatbot* layanan informasi akademik Program Studi Informatika Universitas Bengkulu.

Penelitian yang dilakukan untuk mengembangkan *chatbot* menggunakan berbagai metode sudah sering dilakukan. G. Krishna Vamsi, dkk menggunakan algoritma *deep feed-forward neural network* untuk membuat *chatbot* mendapatkan nilai akurasi 100% [5]. B. Tamizharasi, dkk menggunakan algoritma *SVM* untuk membuat *chatbot* mendapatkan nilai akurasi 92,33% [6]. Ming-Hsiang Su, dkk menggunakan algoritma *LSTM* untuk membuat *chatbot* mendapatkan nilai akurasi 79,96% [7]. Berdasarkan penelitian-penelitian tersebut, pengembangan *chatbot* menggunakan algoritma *deep feed-forward neural network* memiliki nilai akurasi tertinggi. Untuk itu, algoritma *deep feed-forward neural network* akan digunakan dalam pengembangan *chatbot* layanan informasi akademik di Program Studi Informatika Universitas Bengkulu.

2. Metode Penelitian

Metode yang digunakan untuk melakukan penelitian ini adalah CRISP-DM. Adapun alur pelaksanaan penelitian digambarkan melalui *flowchart* berikut.



Gambar 1. Flowchart Penelitian Chatbot

Penelitian dimulai dengan proses pengumpulan data yang terdiri dari *data tag*, *data patters*, dan *data responses*. Setelah dikumpulkan data akan memasuki tahap *data processing* yang di dalam proses ini terjadi juga proses *case folding*, *tokenize*, *ignore word*, *word embedding*, dan *split dataset*. Data yang sudah diproses akan memasuki tahap *modeling*. Pada tahap *modeling*, data akan dibagi mejadi 80% *data training*, 10% *data validation*, dan 10% *data test*. Setelah melalui tahap *modeling*, akan dilakukan evaluasi dengan menggunakan *confusion matrix*, *Precision*, *recall*, dan *F1-Score*. Selanjutnya model *chatbot* akan dilakukan *deployment* menggunakan *framework Flask*.

2.1. Dataset

Penelitian ini menggunakan data yang di-*input* secara manual yang kemudian disimpan ke dalam *file* dengan format JSON. Data yang tersimpan tersebut tersusun dengan struktur sebagai berikut:

Tabel 1. Struktur Data Json

No	Nama struktur	Keterangan
1	Intens	Nama dari kumpulan pasangan pertanyaan dan jawaban.
2	Tags	Pengkategorian jenis pertanyaan
3	Patterns	Berisi kumpulan pertanyaan
4	Respons	Berisi satu atau lebih jawaban

Pengumpulan data dilakukan dengan cara melakukan observasi dan pencatatan langsung oleh peneliti di Program Studi Informatika Universitas Bengkulu.

```

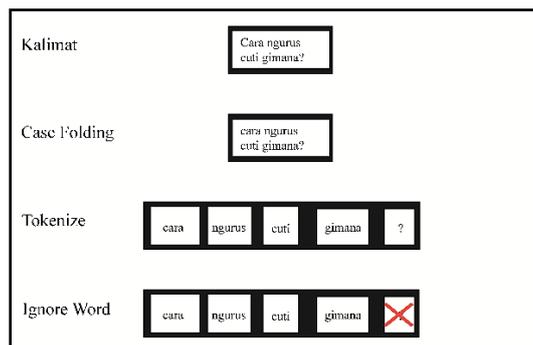
{"intents": [
  {"tag": "Sapaan",
   "patterns": ["Hi", "Hy", "Hai", "Halo", "Helo", "Hmm", "Oy", "Oi", "Hoy", "Hoi", "Tes", "Test", "P"],
   "responses": ["Hai, apa yang bisa saya bantu? ChatBot ini didesain untuk menjawab beber
  ]},
  {"tag": "Terimakasih",
   "patterns": ["Thanks", "Thank you", "Terimakasih", "Terimakasih banyak", "Makasih", "Thankyo"],
   "responses": ["Senang bisa membantumu!:", "Sama-sama :)!"]
  },
  {"tag": "Apa",
   "patterns": ["Apa yang bisa kamu lakukan", "Kamu bisa apa", "bisa apa", "Kamu siapa?"],
   "responses": ["Hai, apa yang bisa saya bantu? ChatBot ini didesain untuk menjawab beber
  ]},

```

Gambar 2. Contoh Dataset

Gambar di atas menampilkan tampilan dari *dataset* yang dikumpulkan dan disimpan ke dalam *file json*. Jumlah data yang dikumpulkan adalah 2059 *documents*, 91 *classes*, dan 296 *lematized words*. Data yang telah dikumpulkan selanjutnya akan diolah dengan menggunakan teknik *natural language processing* dan *natural language toolkit* sebelum diaplikasikan pada algoritma *feed-forward neural network*. Ada 4 proses yang akan dilalui oleh data teks tersebut sebelum nantinya digunakan, di antaranya:

1. *Case folding*
 Pada tahap ini data teks akan dikonverikan ke dalam bentuk teks standar atau menjadikan semua teks menjadi huruf kecil.
2. *Tokenize*
 Setiap kata dari teks akan dipecah menjadi unit-unit kecil yang dinamakan token.
3. *Ignore word*
 Pada tahap ini unit kata akan difilter untuk memilah kata yang tidak memiliki pengaruh signifikan dalam pemaknaan teks.



Gambar 3. Proses *case folding*, *tokenize*, dan *ignore words*

4. *Word embedding*

Word embedding adalah jaan dalam *natural language processing* untuk mengubah teks dari *Bag of Word* kedalam angka vektor.

Tabel 2. Konversi *Bow* Kedalam Vektor

Kata	BoW			
Cara	1	0	0	0
Ngurus	0	1	0	0
Cuti	0	0	1	0
Gimana	0	0	0	1

Dari tabel tersebut didapat kata unik yaitu: cara, ngurus, cuti, dan gimana. Setelah dilakukan pemrosesan *dataset* menggunakan *Case folding*, *Tokenize*, *Ignore word*, dan *bag of word*, selanjutnya adalah membagi atau split *dataset* untuk dilakukan *training*, *testing*, dan *validation*, dimana perbandingan yang digunakan adalah perbandingan dengan rasio 80:10:10 secara berturut-turut. *Dataset* tersebut di split dengan menggunakan *train_test_split* dari *library sklearn*. Dengan menggunakan rasio tersebut maka didapat data yang akan digunakan untuk melakukan *training* berjumlah 1647,2 dari 2059 data, 205,9 data akan digunakan untuk melakukan *testing*, dan 205,9 data lainnya akan digunakan sebagai *data validation*.

2.2. **Modeling**

Feed-forward neural network merupakan jenis *artificial neural network* yang digunakan untuk memetakan data *input* ke data *output*. *Feed-forward neural network* ini terdiri dari *input layer*, satu atau lebih *Hidden layer*, dan *output layer*[8]. Algoritma yang digunakan dalam model *feed-forward neural network* ini adalah algoritma *backpropagation*. Ada tiga tahapan yang terjadi pada algoritma *backpropagation*. Di antara ketiga tahapan tersebut yaitu *forward*, *error calculation*, dan *weight updates*[9]:

1. *Forward*
 Pada tahap *forward* data *input*-an akan melewati jaringan dan proses prediksi data *output* dilakukan dengan menghitung bobot koneksi antar neuron[10].
2. *Error calculating*
 Pada tahap ini *error* antara *output* yang diprediksi dengan *output* sebenarnya dihitung menggunakan *loss function*. *Loss function* mengukur perbedaan antara *output* yang diprediksi dan *output* sebenarnya dan memberikan nilai skalar yang dapat digunakan untuk memandu *weight updates* pada tahap selanjutnya[10].
3. *Weight updates*
 Pada tahap ini bobot koneksi antar neuron akan disesuaikan untuk meminimalkan kesalahan antara *output* yang diprediksi dengan *output* sebenarnya. Hal ini biasanya dilakukan dengan menggunakan algoritma pengoptimalan. Bobot akan disesuaikan sedemikian rupa sehingga mengurangi kesalahan dalam *output* yang diprediksi[10].

2.3. **Evaluasi**

Model dievaluasi evaluasi menggunakan *confusion matrix*. *Confusion matrix* menggambarkan kinerja model klasifikasi pada *dataset validation* dimana nilai sebenarnya diketahui. *Confusion matrix*

terdiri dari empat perhitungan: *recall*, *Precision*, *accuracy*, dan *f1-score*. Perhitungan-perhitungan ini digunakan untuk mengevaluasi kemampuan model untuk memprediksi kelas-kelas dari contoh-contoh uji dengan benar. *Confusion matrix* berguna untuk mengidentifikasi pola-pola klasifikasi yang salah, serta untuk membandingkan kinerja model yang berbeda[11].

Terdapat 4 kombinasi yang berbeda dari nilai yang diprediksi dan nilai yang sebenarnya, yaitu *True Positive (TP)*, *False Positive (FP)*, *False Negatif (FN)*, dan *True Negatif (TN)*[12].

1. *True Positive (TP)*: Anda memprediksi positif dan itu benar. Anda memprediksi seorang pasien sakit dan dia benar-benar sakit[12].
2. *False Negative (FN)*: Anda memprediksi negatif dan itu benar. Anda memprediksi seorang pasien tidak sakit dan dia sebenarnya tidak sakit[12].
3. *False Positive (FP)*: (Kesalahan Tipe 1) Anda memprediksi positif dan itu salah. Anda memprediksi seorang Pasien sakit tapi dia sebenarnya tidak sakit[12].
4. *True Negative (TN)*: (Kesalahan Tipe 2) Anda memprediksi negatif dan itu salah. Anda memprediksi seorang pasien tidak sakit tapi dia sebenarnya sakit[12].

Untuk menghitung *accuracy*, *Precision*, *recall*, dan *f1-score* menggunakan persamaan-persamaan berikut ini:

$$accuracy = \frac{TP + TN}{Total} \times 100\% \quad (1)$$

Accuracy merupakan total seberapa sering model benar dalam mengklasifikasikan semua data[13]. *Accuracy* sering digunakan untuk melihat seberapa baik kinerja dari *neural network*.

$$precision = \frac{TP}{TP + FP} \quad (2)$$

Precision adalah ukuran keakuratan pengklasifikasi saat melakukan prediksi kelas positif[14]. *Precision* digunakan untuk mengevaluasi kinerja dari model *machine learning* termasuk *deep feed-forward neural network*.

$$recall = \frac{TP}{TP + FN} \times 100\% \quad (3)$$

Recall merupakan rasio dari jumlah data yang benar-benar positif terhadap jumlah seluruh data yang sebenarnya positif (yaitu *true positive* dan *true negative*)[14]. *Recall* digunakan untuk mengevaluasi kinerja model pada task tertentu.

$$f1 = 2 \times \frac{precision \times recall}{precision + recall} \times 100\% \quad (4)$$

F1-score merupakan rata-rata dari *Precision* dan *Recall*, sebuah ukuran yang digunakan untuk mengukur kualitas suatu model pembelajaran mesin atau sistem yang digunakan untuk memprediksi suatu kelas atau label[13].

2.4. Deployment

Proses *deployment* yang digunakan pada *chatbot* layanan informasi akademik Program Studi Informatika adalah web menggunakan *framework Flask* dari python. *Flask* dikategorikan sebagai *micro-framework* karena tidak membutuhkan alat atau library khusus. *Flask* tidak memiliki lapisan abstraksi basis data, validasi formulir, atau komponen lain yang sudah diberikan oleh *library* pihak ketiga. Namun, *Flask* mendukung ekstensi yang dapat menambahkan fitur pada aplikasi seolah-olah fitur tersebut telah diimplementasikan dalam *Flask* sendiri[15].

3. Hasil dan Pembahasan

3.1. Modeling

Teknik pemodelan yang digunakan dalam pengembangan *chatbot* ini adalah algoritma *deep feed-forward neural network*. Untuk menemukan komposisi terbaik dari hasil *training* model *chatbot* ini

dilakukan modifikasi nilai di beberapa komponen *training*. Di antara komponen-komponen yang dilakukan modifikasi yaitu jumlah *batch size*, nilai momentum, jumlah *Hidden layer*, dan nilai *epoch* yang digunakan untuk melakukan *training model*.

3.1.1. Batch size

Dalam modifikasi jumlah *batch size*, dilakukan percobaan *training* model menggunakan 3 nilai *batch size* yang berbeda. Nilai *batch size* yang digunakan pada penelitian ini di antaranya 25, 50, dan 100. Nilai-nilai tersebut di *training* dengan kondisi algoritma *deep feed-forward neural network* memiliki 296 neuron pada *input layer* dengan *dropout* 0,89. *Hidden layer* sebanyak 2 yang masing-masing memiliki 250 dan 245 neuron. Pada *output layer* terdapat 91neuron dengan *dropout* 0,1. Nilai momentum 0.95. Untuk *training* dilakukan pada epoch 450. Tabel III di bawah ini menampilkan hasil *training* dengan 3 nilai *batch size* tersebut.

Tabel 3. *Training batch size*

<i>Batch size</i>	<i>accuracy</i>	<i>loss</i>	<i>Val_accuracy</i>	<i>Val_loss</i>
25	0,8388	0,6440	0,9247	0,3857
50	0,8936	0,3703	0,9301	0,2956
100	0,9208	0,2340	0,9462	0,2307

Modifikasi jumlah *batch size* dari 25 hingga 100 menyebabkan kenaikan nilai *accuracy* dan penurunan nilai *loss*. Dari tabel III didapat nilai terbaik berada pada *batch size* 100.

3.1.2. Momentum

Dalam modifikasi nilai momentum, dilakukan percobaan *training* model menggunakan 4 nilai momentum yang berbeda. Nilai momentum yang digunakan pada penelitian ini di antaranya 0.80, 0.85, 0.90, dan 0.95. Nilai-nilai tersebut di *training* dengan kondisi algoritma *deep feed-forward neural network* memiliki 296 neuron pada *input layer* dengan *dropout* 0,89. *Hidden layer* sebanyak 2 yang masing-masing memiliki 250 dan 245 neuron. Pada *output layer* terdapat 91neuron dengan *dropout* 0,1. Nilai *batch size* 100. Untuk *training* dilakukan pada epoch 450. Tabel IV di bawah ini menampilkan hasil *training* dengan 4 nilai momentum tersebut.

Tabel 4. *Training momentum*

<i>Momentum</i>	<i>Accuracy</i>	<i>Loss</i>	<i>Val_accuracy</i>	<i>Val_loss</i>
0,80	0,9097	0,3014	0,9946	0,0187
0,85	0,9077	0,2984	0,9785	0,0551
0,90	0,9097	0,2713	0,9785	0,0469
0,95	0,9189	0,2519	0,9301	0,2206

Dari Tabel IV perubahan yang terjadi di saat nilai momentum dinaikan terlihat pada *val_accuracy*. Ketika nilai momentum dinaikkan, maka nilai *val_accuracy* akan mengecil.

3.1.3. Hidden layer

Dalam modifikasi jumlah lapisan *Hidden layer*, dilakukan percobaan *training* model menggunakan 3 jumlah lapisan *Hidden layer* yang berbeda. Jumlah lapisan *Hidden layer* yang digunakan pada penelitian ini di antaranya 1 *Hidden layer* untuk variasi pertama yang memiliki 250 neuron tanpa *dropout*. Untuk variasi yang kedua menggunakan 2 *hidden layer* tanpa *dropout* dengan masing-masing *Hidden layer* memiliki 250 dan 245 neuron. Untuk variasi ketiga menggunakan 3 *Hidden layer* tanpa *dropout* dengan masing-masing *layer* memiliki 250, 245, dan 240 neuron. Nilai-nilai tersebut di *training* dengan kondisi algoritma *deep feed-forward neural network* dengan *input layer* yang memiliki 296 neuron dengan *dropout* bernilai 0.89, *output layer* yang memiliki 91 neuron dengan *dropout* bernilai 0.1, nilai momentum 0.95, dan nilai *batch size* 100. Untuk *training* dilakukan pada epoch 450. Tabel V di bawah ini menampilkan hasil *training* dengan 3 nilai *batch size* tersebut.

Tabel 5. *training hidden layer*

<i>Hidden layer</i>	<i>Accuracy</i>	<i>Loss</i>	<i>Val_accuracy</i>	<i>Val_loss</i>
1	0,9252	0,2303	1,000	0,0091
2	0,9121	0,2667	0,9462	0,2611
3	0,9111	0,2827	0,8441	1,0573

Dari tabel V diketahui bahwa dengan bertambahnya jumlah *Hidden layer* akan menyebabkan penurunan nilai *accuracy* dan *val_accuracy*. Sedangkan untuk nilai *loss* dan *val_loss* mengalami peningkatan.

3.1.4. Epoch

Setelah dilakukan modifikasi pada komponen *batch size*, momentum, dan jumlah *Hidden layer* didapatkan hasil terbaik pada *batch size* 100, momentum 0,95, dan 2 lapis *Hidden layer*. Untuk itu pada tahap ini nilai-nilai tersebut akan dilakukan *training* dengan menggunakan 3 variasi jumlah epoch yang berbeda. Jumlah epoch yang akan dilakukan percobaan adalah 250, 350, dan 450. Tabel VI di bawah ini menampilkan hasil *training* menggunakan ketiga nilai epoch tersebut.

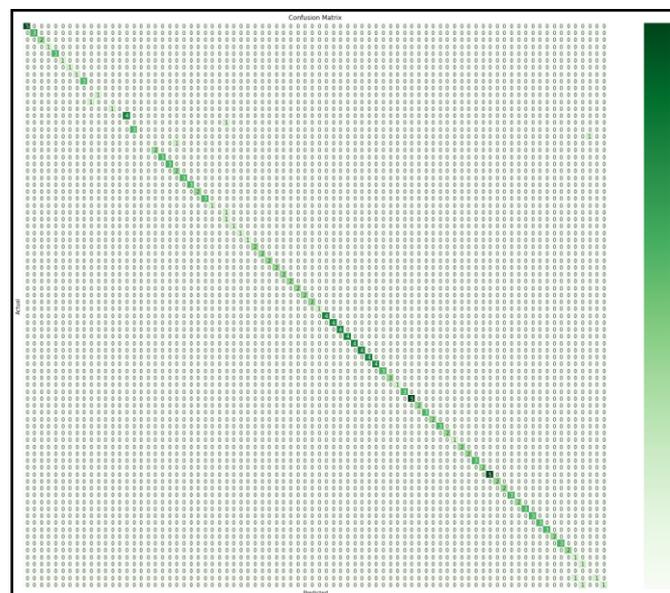
Tabel 6. *training* epoch

<i>Epoch</i>	<i>Accuracy</i>	<i>Loss</i>	<i>Val_accuracy</i>	<i>Val_loss</i>
250	0,8878	0,3591	0,9731	0,0049
350	0,9058	0,2699	0,9624	0,1054
450	0,9247	0,2430	0,9516	0,2049

Dari tabel VI didapatkan perubahan terjadi ketika model di *training* pada 3 nilai epoch berbeda. Ketika nilai epoch dinaikan nilai *accuracy* mengalami kenaikan sedangkan nilai *val_accuracy* mengalami penurunan.

3.2. Evaluasi

Untuk melakukan evaluasi dari model yang di-*training* pada penelitian ini digunakan *confusion matrix*. Setelah dilakukan evaluasi menggunakan *confusion matrix* selanjutnya model akan kembali dievaluasi menggunakan 4 jenis perhitungan lainnya. Adapaun keempat perhitungan lainnya yang digunakan untuk mengevaluasi model yang di *training* ini diantaranya *accuracy*, *Precision*, *recall*, dan *f1-score*.



Gambar 4. Hasil evaluasi *confusion matrix*

Tabel VII di bawah ini menampilkan nilai dari *Precision*, *recall*, dan *f1-score*.

Tabel 7. *Precision*, *recall*, *f1-score*

	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>	<i>Support</i>
0	1.00	1.00	1.00	5
2	1.00	1.00	1.00	3
3	1.00	1.00	1.00	2

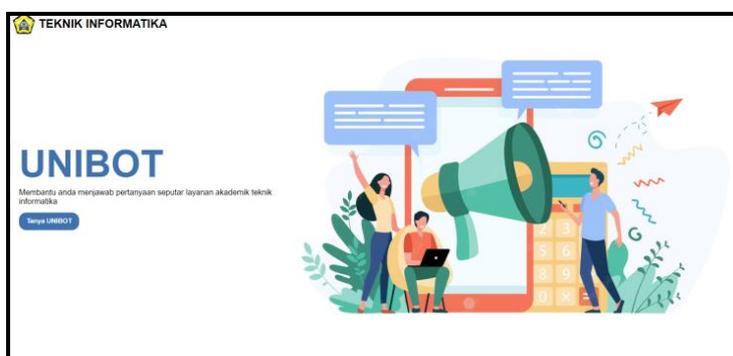
4	1.00	1.00	1.00	1
5	1.00	1.00	1.00	3
6	1.00	1.00	1.00	1
7	1.00	1.00	1.00	1
8	1.00	1.00	1.00	1
9	1.00	1.00	1.00	3
10	0.00	0.00	0.00	0
11	1.00	1.00	1.00	1
12	0.00	0.00	0.00	1
15	1.00	1.00	1.00	1
17	0.00	0.00	0.00	4
18	0.00	0.00	0.00	1
20	1.00	1.00	1.00	3
21	0.00	0.00	0.00	1
22	0.00	0.00	0.00	1
23	1.00	1.00	1.00	2
24	1.00	1.00	1.00	3
25	1.00	1.00	1.00	3
26	0.67	1.00	0.80	2
27	1.00	1.00	1.00	3
28	1.00	1.00	1.00	3
29	1.00	1.00	1.00	2
30	1.00	1.00	1.00	3
31	1.00	1.00	1.00	1
32	0.00	0.00	0.00	1
33	0.33	1.00	0.50	1
34	1.00	1.00	1.00	1
35	1.00	1.00	1.00	1
37	1.00	1.00	1.00	1
38	1.00	1.00	1.00	2
39	1.00	1.00	1.00	2
40	1.00	1.00	1.00	2
41	1.00	1.00	1.00	2
42	1.00	1.00	1.00	2
43	1.00	1.00	1.00	2
44	1.00	1.00	1.00	2
45	1.00	1.00	1.00	2
46	1.00	1.00	1.00	2
47	1.00	1.00	1.00	1
48	1.00	1.00	1.00	4
49	1.00	1.00	1.00	4
50	1.00	1.00	1.00	4
51	1.00	1.00	1.00	4
52	1.00	1.00	1.00	4
53	1.00	1.00	1.00	4
54	1.00	1.00	1.00	4
55	1.00	1.00	1.00	4
56	1.00	1.00	1.00	3
57	1.00	1.00	1.00	2
58	1.00	1.00	1.00	1
59	1.00	1.00	1.00	3
60	1.00	1.00	1.00	5
61	1.00	1.00	1.00	2
62	1.00	1.00	1.00	3
63	1.00	1.00	1.00	2
64	1.00	1.00	1.00	3
65	1.00	1.00	1.00	2
66	1.00	1.00	1.00	1
67	1.00	1.00	1.00	2
68	1.00	1.00	1.00	2
69	1.00	1.00	1.00	3
70	1.00	1.00	1.00	2
71	1.00	1.00	1.00	5
72	1.00	1.00	1.00	2
73	1.00	1.00	1.00	2
74	1.00	1.00	1.00	3
75	1.00	1.00	1.00	2
76	1.00	1.00	1.00	3
77	1.00	1.00	1.00	3

78	1.00	1.00	1.00	3
79	1.00	1.00	1.00	3
80	1.00	1.00	1.00	2
81	1.00	1.00	1.00	3
82	1.00	1.00	1.00	2
83	0.50	1.00	0.67	1
84	0.50	1.00	0.67	1
87	0.00	0.00	0.00	0
88	1.00	0.50	0.67	2
89	1.00	0.50	0.67	2
Rerata	0.88	0.89	0.88	

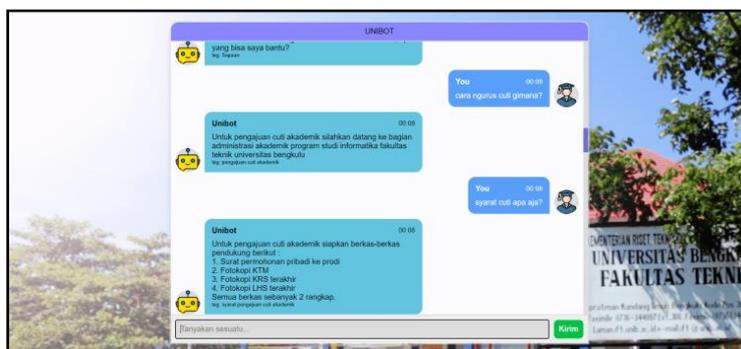
Dari tabel VII didapat rerata *Precision* sebesar 0.88, *recall* 0.89, dan *f1-score* 0.88. selain itu didapat juga nilai *accuracy* sebesar 0.94 pada tahap evaluasi ini.

3.3. Deployment

Setelah proses *modeling* dan evaluasi selesai, model *chatbot* layanan informasi akademik program studi informatika universitas Bengkulu ini kemudian di *deploy* menjadi *chatbot* berbasis *web* menggunakan *framework Flask*. *Chatbot* layanan informasi akademik ini memiliki tampilan *home page* dan *chatting page*.



Gambar 5. Home page



Gambar 6. Chatting page

Pada gambar 5 ditampilkan *home page chatbot* dimana disana terdapat *button Tanya unibot* untuk memulai *chat* dengan bot. kemudian pada gambar 6 merupakan tampilan dari *chatting page* dari *chatbot* layanan informasi akademik Program Studi Informatika Universitas Bengkulu.

4. Kesimpulan

Dalam penelitian ini, *chatbot* dikembangkan untuk mengatasi keterbatasan ruang dan waktu untuk mendapatkan informasi akademik di Program Studi Informatika Universitas Bengkulu. Dengan adanya *chatbot* ini diharapkan bisa meningkatkan efektivitas dan efisiensi pelayanan akademik di Program Studi Informatika Universitas Bengkulu. Pada tahap pengembangannya, *chatbot* menjalani proses pelatihan dengan menggunakan data yang ada. Setelah melalui proses tersebut, *chatbot* berhasil mencapai tingkat akurasi sebesar 0.9247, yang berarti sekitar 92.47% dari prediksi yang dibuat oleh *chatbot* sesuai dengan

jawaban yang diharapkan. Selain itu, dalam proses pelatihan tersebut, *chatbot* juga mengalami tingkat kehilangan (*loss*) sebesar 0.2430. Semakin rendah nilai *loss*, semakin baik pula performa *chatbot* dalam meminimalkan kesalahan prediksi. Untuk memastikan kinerja *chatbot* pada data yang belum pernah dilihat sebelumnya, dilakukan evaluasi menggunakan data validasi. Hasilnya menunjukkan bahwa *chatbot* memiliki akurasi validasi sebesar 0.9516, yang berarti sekitar 95.16% dari prediksi yang dilakukan pada data validasi sesuai dengan kenyataan. Tidak hanya itu, evaluasi juga dilakukan menggunakan *confusion matrix*, yang menghasilkan beberapa metrik evaluasi tambahan. Hasil evaluasi tersebut menunjukkan bahwa akurasi setelah dievaluasi adalah sebesar 0.94, yang berarti sekitar 94% dari prediksi yang dilakukan oleh *chatbot* adalah benar. Selanjutnya, *Precision* (presisi) *chatbot* adalah 0.88, yang menggambarkan sejauh mana *chatbot* dapat memberikan hasil yang tepat dalam prediksinya. *Recall* (sensitivitas) *chatbot* adalah 0.89, yang menunjukkan sejauh mana *chatbot* dapat mengenali dan memprediksi dengan benar kasus positif. *F1-score chatbot* adalah 0.88, yang merupakan rata-rata harmonik antara presisi dan *recall*, dan memberikan gambaran keseluruhan tentang performa *chatbot* dalam memprediksi dengan baik.

Daftar Pustaka

- [1] B. R. Ranoliya, N. Raghuwanshi, and S. Singh, "Chatbot for university related FAQs," *2017 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2017*, vol. 2017-January, pp. 1525–1530, 2017, doi: 10.1109/ICACCI.2017.8126057.
- [2] H. A. Santoso *et al.*, "Chatbot for University Admission Services," *2018 Int. Semin. Appl. Technol. Inf. Commun.*, pp. 417–423, 2018.
- [3] D. Toniuc and A. Groza, "Climebot: An argumentative agent for climate change," *Proc. - 2017 IEEE 13th Int. Conf. Intell. Comput. Commun. Process. ICCP 2017*, vol. 2, pp. 63–70, 2017, doi: 10.1109/ICCP.2017.8116984.
- [4] G. Guntoro, Loneli Costaner, and L. Lisnawita, "Aplikasi Chatbot untuk Layanan Informasi dan Akademik Kampus Berbasis Artificial Intelligence Markup Language (AIML)," *Digit. Zo. J. Teknol. Inf. dan Komun.*, vol. 11, no. 2, pp. 291–300, 2020, doi: 10.31849/digitalzone.v11i2.5049.
- [5] G. Hajela, "Chatbot," 2020.
- [6] B. Tamizharasi, L. M. Jenila Livingston, and S. Rajkumar, "Building a medical chatbot using support vector machine learning algorithm," *J. Phys. Conf. Ser.*, vol. 1716, no. 1, 2021, doi: 10.1088/1742-6596/1716/1/012059.
- [7] M. H. Su, C. H. Wu, K. Y. Huang, Q. B. Hong, and H. M. Wang, "A chatbot using LSTM-based multi-layer embedding for elderly care," *Proc. 2017 Int. Conf. Orange Technol. ICOT 2017*, vol. 2018-January, pp. 70–74, 2018, doi: 10.1109/ICOT.2017.8336091.
- [8] S. Roller *et al.*, "Recipes for building an open-domain chatbot," *EACL 2021 - 16th Conf. Eur. Chapter Assoc. Comput. Linguist. Proc. Conf.*, pp. 300–325, 2021, doi: 10.18653/v1/2021.eacl-main.24.
- [9] A. P. Chaves and M. A. Gerosa, "How Should My Chatbot Interact? A Survey on Social Characteristics in Human–Chatbot Interaction Design," *Int. J. Hum. Comput. Interact.*, vol. 37, no. 8, pp. 729–758, 2021, doi: 10.1080/10447318.2020.1841438.
- [10] Mahardhika Chandra, Rizki Pratama, Fathan Azka Pradana, and Alvita Bonita, "Chatbot Interaksi Rumah Sakit menggunakan FFNN," *Indones. J. Data Sci.*, vol. 3, no. 1, pp. 62–68, 2022, doi: 10.56705/ijodas.v3i1.36.
- [11] P. Algoritma and B. Moore, "Penerapan algoritma boyer moore pada chatbot ustaz abdul somad," 2021.
- [12] J. Homepage, Y. Saputra Wijaya, F. Zoromi Informatic, and S. Amik Riau, "JAIA-Journal Of Artificial Intelligence And Applications Chatbot Designing Information Service for New Student Registration Based on AIML and Machine Learning," *JAIA-Journal Artif. Intell. Appl.*, vol. 1, no. 1, pp. 1–10, 2020.
- [13] I. W. Saputro and B. W. Sari, "Uji Performa Algoritma Naïve Bayes untuk Prediksi Masa Studi Mahasiswa," *Creat. Inf. Technol. J.*, vol. 6, no. 1, p. 1, 2020, doi: 10.24076/citec.2019v6i1.178.
- [14] Karsito and S. Susanti, "Klasifikasi Kelayakan Peserta Pengajuan Kredit Rumah Dengan Algoritma Naïve Bayes Di Perumahan Azzura Residencia," *J. Teknol. Pelita Bangsa*, vol. 9, pp. 43–48, 2019.
- [15] S. Sultan, A. Issa, and R. Bayu, "Pengembangan Chatbot Yanies Cookies Untuk Pemesanan Kue Kering Berbasis Dialogflow," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 4, no. 7, pp. 2121–2129, 2020.