

## Implementasi *Deep Feed-Forward Neural Network* pada Perancangan *Chatbot* Berbasis *Web* di UPPIK RSUD M. YUNUS

Ruvita Faurina\*, M. Jumli Gazali, Icha Dwi Aprilia Herani

Program Studi Informatika, Fakultas Teknik, Universitas Bengkulu

Jl. WR. Supratman Kandang Limun, Bengkulu 38371 A, Indonesia

\*email: [ruvita.faurina@unib.ac.id](mailto:ruvita.faurina@unib.ac.id)

(Naskah masuk: 1 Januari 2023; diterima untuk diterbitkan: 7 Juli 2023)

**ABSTRAK** – UPPIK (Unit Pengaduan Pelanggan Informasi dan Konseling) di RSUD M. Yunus memiliki tugas penting dalam melayani pengunjung yang datang ke rumah sakit. Namun, sering kali pengunjung mengeluh tentang pelayanan UPPIK karena jam kerja yang terbatas, sehingga tidak selalu ada staff/petugas yang tersedia untuk memberikan informasi yang dibutuhkan pengunjung. Selain itu, pandemi Covid-19 yang belum mereda mengharuskan masyarakat untuk menjaga jarak dan mengurangi interaksi antar individu. Sebagai solusi atas masalah ini, sebuah chatbot otomatis telah dikembangkan untuk memberikan layanan seolah-olah pengunjung berbicara langsung dengan staff/petugas tanpa ada batasan waktu. Penelitian ini menggunakan algoritma *Deep Feed-Forward Neural Network*. Dataset yang digunakan adalah kumpulan data pertanyaan jawaban yang dikumpulkan melalui observasi langsung di UPPIK, yang terdiri dari 1464 pasangan data. Akurasi terbaik diperoleh dengan membagi dataset menjadi 80% data training (1.185 data), 10% data testing (147 data), dan 10% data validasi (132 data) dengan epoch 300, yang menghasilkan akurasi sebesar 91,98%. Evaluasi terhadap hasil ini menunjukkan nilai *precision* sebesar 0,99, *recall* sebesar 0,98, dan *f1-score* sebesar 0,99.

**Kata Kunci** – UPPIK RSUD M. Yunus Bengkulu; Kecerdasan Buatan; Chatbot; *Deep Feed-Forward Neural Network*; *Deep Learning*

## Implementation of *Deep Feed-Forward Neural Network* in the Design of *Web-Based Chatbot* at UPPIK RSUD M. YUNUS

**ABSTRACT** – The UPPIK (Customer Information and Counseling Complaint Unit) at the M. Yunus Hospital has an important role in serving visitors who come to the hospital. However, visitors often complain about the UPPIK service due to limited working hours, so there is not always staff available to provide the information needed by visitors. In addition, the ongoing Covid-19 pandemic requires people to maintain distance and reduce interaction with others. To solve this problem, an automatic chatbot has been developed to provide service as if the visitor is speaking directly to the staff without any time constraints. This research uses a *Deep Feed-Forward Neural Network* algorithm. The dataset used is a collection of question-answer data collected through direct observation at the UPPIK, consisting of 1464 pairs of data. The best accuracy was obtained by splitting the dataset into 80% training data (1,185 data), 10% testing data (147 data), and 10% validation data (132 data) with 300 epochs, which resulted in an accuracy of 91.98%. Evaluation of these results showed a *precision* value of 0.99, a *recall* value of 0.98, and an *f1-score* of 0.99.

**Keywords** - UPPIK RSUD M. Yunus Bengkulu; Artificial Intelligence; Chatbot; *Deep Feed-Forward Neural Network*; *Deep Learning*

### 1. PENDAHULUAN

Pada saat ini, kebutuhan akan ketersediaan

jaringan komunikasi dan internet sangat tinggi, karena semakin meningkatnya ketergantungan manusia pada peranan Teknologi Informasi. Orang-

orang lebih banyak menghabiskan waktu untuk berinteraksi di media sosial karena kemudahan akses dari mana saja dan kapan saja [1].

Perkembangan teknologi komunikasi dan informasi tidak hanya mempengaruhi bidang yang berkaitan dengan komputer, tetapi juga dapat mempengaruhi bidang lainnya untuk menunjang semua proses agar lebih cepat dan akurat. Salah satu bidang yang juga terpengaruh oleh perkembangan teknologi adalah customer relation, banyak rumah sakit yang menggunakan teknologi *customer relation* untuk memudahkan penyelesaian masalah yang terkumpul serta menanggapi atau permintaan layanan informasi dari konsumen [2].

Berdasarkan hasil kuesioner yang diisi oleh 41 orang pengunjung RSUD M. Yunus yaitu Kurang Baik 33%; Sangat Baik 31%; Baik 26%; Cukup Baik 10%. Dari hasil kuisisioner ini dapat disimpulkan bahwa pelayanan RSUD M. Yunus Bengkulu masih dinilai "Kurang Baik".

Selama kegiatan observasi lapangan didapatkan bahwa RSUD M. Yunus Bengkulu membutuhkan sebuah sistem informasi yang dapat mengatasi kendala dalam pelayanan pengunjung yang disebabkan oleh tidak adanya staf/petugas yang bertugas dikarenakan terbatasnya jam operasional kerja. Kurang baiknya pelayanan ini menyebabkan pengunjung tidak dapat memperoleh informasi dengan baik. Di samping itu, jauhnya jarak antar ruangan pasien dengan meja Unit Pengaduan Pelanggan Informasi dan Konseling RSUD M. Yunus Bengkulu menyebabkan pengunjung terlalu jauh mengunjungi UPPIK untuk sekedar menanyakan informasi. Tentu hal itu akan memakan cukup banyak waktu dan juga biaya. Sehingga, menyebabkan penurunan angka kepuasan pengunjung RSUD M. Yunus Bengkulu.

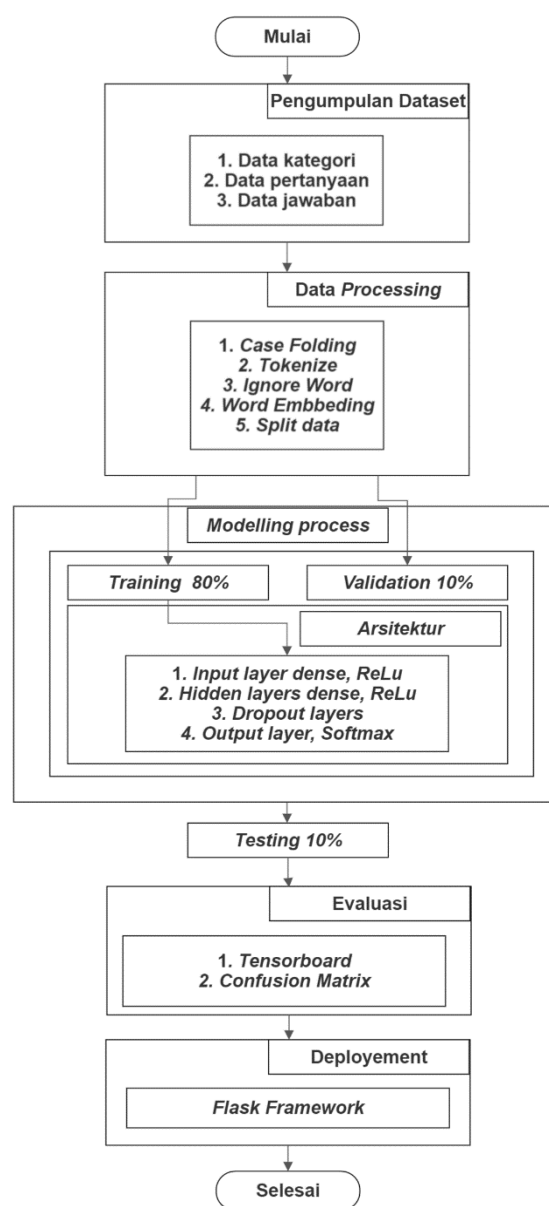
Rafly Pradana Putra, dkk. menggunakan Algoritma *Support Vector Machine* (SVM) dalam mengembangkan *Chatbot* di *Informatics Department UPN "Veteran" Yogyakarta* menunjukkan hasil akurasi sebesar 87% [3]. Kristian Adi dkk. menggunakan Algoritma *K-Nearest Neighbor* (KNN) dalam mengembangkan *Chatbot* Layanan Akademik menunjukkan hasil akurasi sebesar 53% [4]. Bhartiya dkk. menggunakan Algoritma *Feed-Forward Neural Network* (FNN) dalam mengembangkan Sistem *Chatbot* Universitas menunjukkan hasil akurasi sebesar 99% [5]. Berdasarkan hasil penelitian yang telah dilakukan sebelumnya, *Chatbot* yang menggunakan Algoritma *Feed-Forward Neural Network* menunjukkan tingkat akurasi yang paling tinggi. Oleh karena itu, peneliti memutuskan untuk membuat *Chatbot* RSUD M. Yunus dengan menggunakan Algoritma *Feed-Forward Neural*

*Network*.

Batasan topik pembahasan pada penelitian ini adalah *dataset* berupa pertanyaan mengenai pertanyaan natural yang biasanya ditanyakan oleh pengunjung pada pihak UPPIK RSUD M. Yunus seperti pertanyaan mengenai ruangan, jadwal, administrasi, dan persyaratan. Oleh karena itu, penelitian ini bertujuan untuk klasifikasi pertanyaan sesuai dengan jawaban. Sehingga pengunjung lebih mudah untuk memperoleh informasi tanpa harus mengunjungi UPPIK RSUD M. Yunus.

## 2. METODE DAN BAHAN

Metode yang dilakukan dalam melakukan penelitian *chatbot* RSUD M. Yunus ini terdapat pada Gambar 1.



Gambar 1. Metode penelitian *Chatbot* RSUD M. Yunus [1]

## Dataset

Dalam penelitian ini, data yang digunakan merupakan kumpulan data yang dimasukkan secara manual dan disimpan dalam file dengan format JSON (*JavaScript Object Notation*) [6]. Strukturnya terdiri dari *intents*, *tag*, *patterns*, dan *responses*. *Intens* merupakan nama dari kumpulan semua data pertanyaan dan jawaban yang akan digunakan untuk melatih *chatbot*. *Tag* merupakan pengelompokan dari dari banyak pola pertanyaan pengunjung dengan 1 jawaban. *Patterns* berisikan data mengenai pola-pola pertanyaan berbeda sesuai dengan cara dan bahasa pengunjung. *Responses* berisikan jawaban dari bermacam-macam pola pertanyaan pengguna.

```

1 {
2   "intents": [
3     {
4       "tag": "ICU",
5       "patterns": [
6         "Dimana ruang rawat inap ICU?",
7         "Ruang rawat inap ICU dimana?",
8         "Dimana letak ruang rawat inap ICU?",
9         "Lokasi ruang rawat inap ICU dimana?",
10        "Ruang ICU ada dimana?",
11        "Pasien ICU dimana"
12      ],
13      "responses": [
14        "Di Gedung B, Lantai 1",
15        "Lantai 1 Gedung B"
16      ]
17    },

```

Gambar 2. Dataset dalam format json [2]

Tabel 1 adalah jumlah data dari *dataset* yang telah dikumpulkan. Pada *dataset* penelitian ini, terdapat 3 label kelas yang didefinisikan berdasarkan kategori, yaitu jumlah data *documents* (*pattern* pertanyaan), jumlah data *classes* (*tag*/jenis pertanyaan), dan jumlah *lemmatizer* (kata unik).

Tabel 1. Jumlah dataset per label

Label	Jumlah
<i>documents</i>	1.464
<i>classes</i>	62
<i>lemmatizer</i>	545

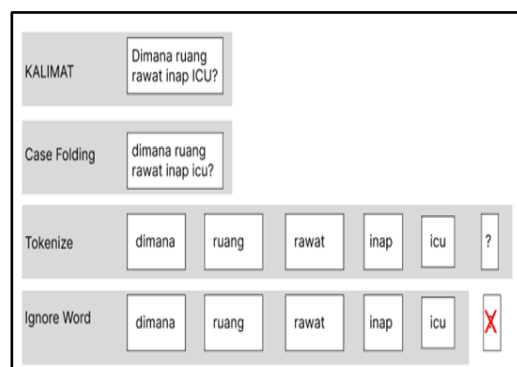
Pada tahap selanjutnya, teks akan diolah menggunakan teknik *Natural Language Processing* (NLP) dan *Natural Language Toolkit* (NLTK) sebelum diaplikasikan pada algoritma *Deep Feed-Forward Neural Network* (DFNN). Data yang berisi teks yang panjang akan melalui beberapa proses sebelum

digunakan, di antaranya: [6]

*Case Folding*, proses pengubahan semua karakter menjadi huruf kecil serta menghilangkan karakter selain huruf [7].

*Tokenize*, Dalam proses *tokenize*, setiap kata dalam sebuah kalimat akan diidentifikasi berdasarkan spasi atau tanda baca. Setiap token tersebut kemudian akan dipisahkan dan diberi label sesuai dengan tipe token yang diwakili, misalnya sebagai kata benda, kata kerja, atau tanda baca. Proses ini akan membantu komputer untuk memahami struktur dan makna dari sebuah teks dengan lebih baik, sehingga dapat digunakan untuk aplikasi seperti translasi, sumber daya manusia, atau *chatbot* [7].

*Ignore Word*, dalam proses pemrosesan bahasa alami (*natural language processing*), *ignore words* dapat dihapus dari teks untuk memperbaiki performa dari algoritma yang digunakan. Hal ini dikarenakan, *ignore words* biasanya tidak memiliki informasi yang bermanfaat bagi pemrosesan teks. Dengan menghapus *ignore words*, algoritma akan lebih mudah untuk mengidentifikasi dan memproses kata-kata yang memiliki makna atau arti yang lebih penting bagi teks tersebut [7].



Gambar 3. Ilustrasi proses Case Folding, Tokenize, dan Ignore Words [3]

*Word embedding* merubah representasi teks dari *Bag of Words* (BOW) menjadi nilai-nilai kontinu yang bernilai rendah, yang disebut vektor. Selain itu, dalam BOW, posisi kata-kata tersebut ditempatkan tidak saling terkait. Namun, dengan *word embedding*, hubungan antara kata-kata dapat diwakili dengan jarak di antara vektor-vektor tersebut. Kata yang sinonim akan dipetakan di dekat satu sama lain dalam ruang vektor, sehingga informasi yang terkandung dalam vektor kata menjadi lebih banyak, yang akan membantu komputer untuk lebih memahami hubungan antara kata-kata [8].

*Bag of word* atau kantung kata adalah proses pengumpulan data yang telah diproses sebelumnya

kemudian memodelkan setiap data dengan menghitung kemunculan dari setiap data [9].

Tabel 2. Hasil vector dari Word Embedding menggunakan Bag of Word

Kata	BoW vector				
dimana	1	0	0	0	0
ruang	0	1	0	0	0
rawat	0	0	1	0	0
inap	0	0	0	1	0
icu	0	0	0	0	1

Pada Tabel 2 merupakan contoh BoW vektor yang hasilnya BoW mendapat kata unik meliputi: dimana, ruang, rawat, inap, icu. 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*, perbandingan yang digunakan adalah perbandingan dengan rasio 80:10:10 secara berturut-turut. Tabel 3 berikut ini menunjukkan jumlah data yang telah di *split* dengan rasio tersebut.

Tabel 3. Jumlah data yang di *split* dengan rasio 80:10:10

Jenis Split	Rasio	Jumlah
Training	80%	1.185
Testing	10%	147
Validation	10%	132
<b>Total data</b>	<b>100%</b>	<b>1464</b>

Data yang di *split* merupakan data *pattern* atau pola pertanyaan yang berasal dari *dataset*. Teknik *splitting* menggunakan *train\_test\_split* yang berasal dari *library sklearn*.

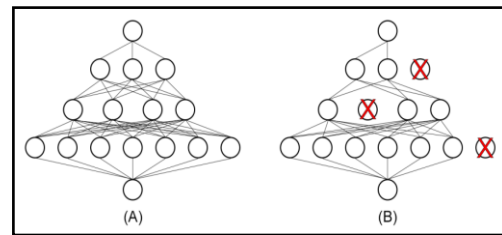
## Modeling

*Deep Feed-forward neural network* (DFNN) adalah salah satu jenis *neural network* yang digunakan untuk membuat prediksi. DFNN menggunakan algoritma *backpropagation* untuk menyesuaikan bobotnya.

Algoritma *backpropagation* terdiri dari tiga tahap: *feedforward* dari pola *input*, penghitungan dan propagasi balik *error*, dan penyesuaian bobot. Pada tahap *feedforward*, pola *input* diteruskan melalui jaringan *neural* untuk memprediksi hasil. Pada tahap penghitungan dan propagasi balik *error*, selisih antara hasil prediksi dan hasil aktual dihitung dan diteruskan kembali ke jaringan untuk memperbaiki bobotnya. Pada tahap penyesuaian bobot, bobot diperbarui berdasarkan selisih tersebut untuk membuat hasil prediksi lebih akurat di masa yang akan datang [2].

Pada Gambar 4 di bawah ini terdapat ilustrasi dari arsitektur model DFNN dengan menggunakan

*Dense (Fully Connected Layer)* dan *Dropout*.

Gambar 4. Perbedaan Arsitektur DFNN dengan *Dense* (A) dan setelah di *dropout* (B) [4]

Fungsi *aktivasi* ReLU atau *Rectified Linear Unit* telah terbukti dapat meningkatkan pelatihan jaringan saraf *deep learning*. ReLU bekerja dengan menentukan batas nilai pada 0, artinya ia mengeluarkan 0 ketika masukan kurang dari 0, dan mengeluarkan nilai masukan ketika lebih besar atau sama dengan 0. Dengan kata lain, fungsi ini didefinisikan dengan rumus 1.

$$f(x) = \max(0, x) \quad (1)$$

Ketika masukan kurang dari 0, ReLU mengeluarkan 0, dan ketika masukan lebih besar atau sama dengan 0, ia mengeluarkan fungsi linear. Fungsi aktivasi ini sederhana dan efektif, dan sering digunakan sebagai pengganti fungsi aktivasi *sigmoid* atau *tanh* karena kemampuannya menghindari masalah turunnya gradien dan mempercepat pelatihan [10].

Fungsi *aktivasi softmax* di lapisan *output* mewakili distribusi kategori atas label kelas dan mendapatkan probabilitas masing-masing masukan yang termasuk dalam sebuah label. Karena fungsi aktivasi *softmax* digunakan pada lapisan *output*, kita harus mengkode label pertanyaan ke dalam format *one-hot* untuk proses pembelajaran model [11].

*Hyperparameter Tuning* menghasilkan model optimal yang meminimalkan fungsi kerugian yang telah ditentukan pada data yang disediakan. *Epoch* menentukan berapa kali algoritma DFNN bekerja melewati seluruh *dataset* baik secara *forward*. *Batch size* menentukan berapa banyak sampel yang akan digunakan oleh algoritma untuk memperbaiki parameter model dalam setiap iterasi dari proses

pelatihan [12].

### Evaluasi

Pada penelitian ini evaluasi menggunakan *Confusion Matrix*. *Confusion matrix* menggambarkan kinerja model klasifikasi pada *dataset validation* yang 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 [13].

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). Gambar 5 adalah analogi agar klasifikasi mudah dipahami [14].

		Predicted class	
		P	N
Actual Class	P	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

Gambar 5. Ilustrasi *predict class* dan *actual class* pada *confusion matrix* [5]

*True Positive* (TP): Anda memprediksi positif dan itu benar. Anda memprediksi seorang wanita hamil dan dia benar-benar hamil [14].

*False Positive* (FP): Anda memprediksi negatif dan itu benar. Anda memprediksi seorang pria tidak hamil dan dia sebenarnya tidak hamil [14].

*False Negatif* (FN): (Kesalahan Tipe 1) Anda memprediksi positif dan itu salah. Anda memprediksi seorang pria hamil tapi dia sebenarnya tidak hamil [14].

*True Negatif* (TN): (Kesalahan Tipe 2) Anda memprediksi negatif dan itu salah. Anda memprediksi seorang wanita tidak hamil tapi dia sebenarnya hamil [14].

Rumus 2, 3, 4, 5 adalah rumus untuk menghitung *accuracy*, *precision*, *recall*, dan *f1-score*:

$$accuracy = \frac{TP+TN}{Total} * 100\% \quad (2)$$

*Accuracy* merupakan total seberapa sering model benar dalam mengklasifikasikan semua data [15].

$$precision = \frac{TP}{TP+FP} * 100\% \quad (3)$$

*Precision* adalah jumlah data yang positif benar (jumlah data positif yang dikenali dengan benar sebagai positif) dibagi dengan jumlah data yang dikenali sebagai positif [16].

$$recall = \frac{TP}{TP+FN} * 100\% \quad (4)$$

*Recall* merupakan rasio dari jumlah data yang benar-benar positif terhadap jumlah seluruh data yang sebenarnya positif (yaitu *true positive* dan *true negative*) [16].

$$f1\ score = 2 * \frac{precision * recall}{precision + recall} * 100\% \quad (5)$$

*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 [15].

### Deployment

Proses *deploy* yang digunakan pada *Chatbot* RSUD M. Yunus 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 [17].

## 3. HASIL DAN PEMBAHASAN

Pada penelitian kali ini menggunakan jumlah *epoch* sebesar 300 dan *batch size* sebesar 25. Lalu peneliti menentukan performa model dengan melakukan berbagai percobaan sebagaimana ditunjukkan pada Tabel 4.

Tabel 4. Jenis percobaan performa model *Chatbot* RSUD M. Yunus

Percobaan ke	Jenis percobaan
1	Pengaruh jumlah <i>hidden layer</i>
2	Pengaruh <i>Dropout Layer</i>
3	Pengaruh nilai <i>dropout</i>

Tabel 5. Pengaruh Jumlah *Hidden Layer* dengan *epoch* 300 dan *batch size* 25

Jumlah <i>Layer</i>	Performa Model			
	<i>loss</i>	<i>Accuracy</i>	<i>val_loss</i>	<i>val_accuacy</i>
1 <i>layer</i>	0.0143	1	0.2167	0.9394
2 <i>layer</i>	0.0151	1	0.2143	0.9242
3 <i>layer</i>	0.0073	0.9991	0.2091	0.9394
4 <i>layer</i>	0.0060	0.9992	0.2072	0.9470
5 <i>layer</i>	0.0160	0.9953	0.4139	0.9318
6 <i>layer</i>	0.0320	0.9915	0.4091	0.9394
7 <i>layer</i>	0.2389	0.9266	0.6009	0.9198
8 <i>layer</i>	0.3811	0.8709	1.1294	0.8712

### Pengaruh Jumlah *Hidden Layer*

Berdasarkan Tabel 5, pada *layer* berjumlah 1 sampai 6 menunjukkan performa model yang *overfitting* (menghafal), sedangkan pada *layer* 8 menunjukkan *underfitting* ditunjukkan dengan *val\_loss* yang tinggi. Performa model terbaik terdapat pada model dengan jumlah *Hidden Layer* sebanyak 7 antara hasil *accuracy*,

dan *val\_accuacy* stabil dan tidak menunjukkan *underfitting* atau *overfitting* menandakan model belajar dengan baik.

### Pengaruh Jumlah *Dropout*

Tabel 6. Pengaruh *Dropout Layer* dengan *epoch* 300 dan *batch size* 25 7 hidden layer

Jenis <i>Dropout</i>	Performa Model			
	<i>loss</i>	<i>accuracy</i>	<i>val_loss</i>	<i>val_accuacy</i>
Tanpa <i>dropout</i>	0.0021	0.9983	0.5816	0.8788
Per- <i>dense layer</i>	0.2389	0.9266	0.6009	0.9198
1 Sebelum <i>output layer</i>	0.0275	0.9899	0.5050	0.9470

Berdasarkan Tabel 6, pada jenis tanpa *dropout layer* dan *dropout layer* sebelum *output layer* menunjukkan performa model yang *overfitting* (menghafal), performa model terbaik terdapat pada model dengan jenis *dropout layer* yang berada disetiap

sesudah *dense layer* menunjukkan performa stabil dan tidak *underfitting* atau *overfitting* menandakan model belajar dengan baik.

### Pengaruh Nilai *Dropout*

Tabel 7. Pengaruh nilai *Dropout Layer* dengan *epoch* 300 dan *batch size* 25 7 *layer*

Nilai <i>Dropout</i>	Performa Model			
	<i>loss</i>	<i>accuracy</i>	<i>val_loss</i>	<i>val_accuacy</i>
0.1	0.0043	0.9975	0.6180	0.9091
0.2	0.0122	0.9949	0.3948	0.9318
0.3	0.0130	0.9966	0.4686	0.9242
0.4	0.0620	0.9764	0.5895	0.9167
0.5	0.2475	0.9139	0.6270	0.9015
0.6	0.8519	0.7511	0.7210	0.8182
0.7	1.7787	0.5764	1.6588	0.6136
0.8	3.2000	0.1840	3.1596	0.1894
0.9	nan	0.0380	nan	0.0379

Berdasarkan Tabel 7, pada nilai *dropout layer* sebesar 0.1 - 0.4 menunjukkan performa model yang

*overfitting* (menghafal), sedangkan pada nilai *dropout layer* sebesar 0.6 - 0.9 menunjukkan performa model

yang *underfitting* (tidak belajar dengan baik). Performa model terbaik terdapat pada model *dropout layer* dengan nilai sebesar 0.5 yang menunjukkan performa stabil dan tidak *underfitting* atau *overfitting* menandakan model belajar dengan baik.

### Modelling

Berdasarkan percobaan sebelumnya, tahap *modeling* pada penelitian ini menggunakan algoritma *Deep Feed-Forward Neural Network* dengan menggunakan 9 *layer*. Model DFNN yang akan digunakan pada *Chatbot RSUD M. Yunus layer* terdiri atas *input layer* lapisan yang membawa data masuk kedalam system untuk kemudian di proses pada *layer* selanjutnya. Jumlahnya 1 *layer* dengan 545 *neuron* (sama dengan jumlah input/kata unik) dengan fungsi aktivasi ReLu. Lalu *Hidden layer* lapisan antara *input layer* dan *output layer*, dimana *artificial neuron* yang memiliki sekumpulan *input* pembobot '*weight*' dan prosedur untuk menghasilkan *output neuron* melalui fungsi aktivasi ReLu. Jumlahnya 1 *layer* dengan 545 *neuron*, 2 *layer* dengan 300 *neuron*, 2 *layer* dengan 150 *neuron*, dan 2 *layer* dengan 62 *neuron*. Terakhir merupakan *output layer*

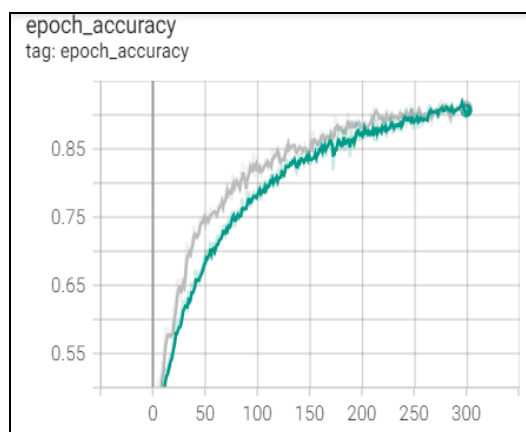
yang merupakan lapisan terakhir dari *neuron* yang menghasilkan *output*. Jumlahnya 1 *layer* dengan 62 *neuron* (sama dengan jumlah *output/tag*). Fungsi *Rectified Linear Unit* (ReLU) memiliki perhitungan yang sederhana: jika elemen bernilai negatif, nilainya diubah menjadi 0. Fungsi aktivasi *softmax* berguna untuk menghitung probabilitas dalam klasifikasi *multi kelas* dengan *output kelas* yang memiliki probabilitas paling tinggi. Fungsi aktivasi *softmax* menghasilkan *output* bernilai antara 0 hingga 1.

Dengan melihat *accuracy* dan *loss* yang terjadi saat proses klasifikasi, model dapat dikatakan baik untuk digunakan dalam pengklasifikasian pertanyaan pada *Chatbot RSUD M. Yunus* dengan *accuracy* dan *val\_accuracy* mencapai 91% dengan *epoch* 300 dan *batch size* 25. Satu *epoch* dapat diartikan bahwa *artificial neural network* telah melakukan proses belajar dari data pelatihan secara keseluruhan. Dalam *artificial neural network*, proses pembelajaran yang dilakukan secara berulang bertujuan untuk mencapai nilai *error* dan *akurasi* yang sebaik mungkin. Tabel 8 berikut ini adalah tabel ringkasan *training* per 100 *epoch*.

Tabel 8. Hasil *training* dengan menggunakan 300 *epoch* dan 25 *batch size*

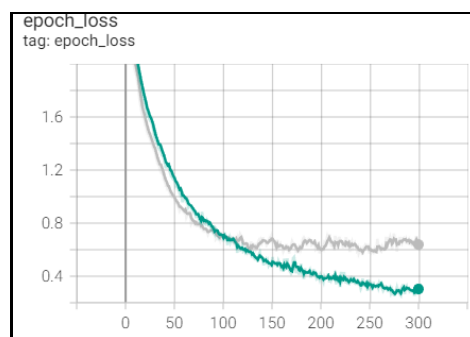
<i>epoch ke-</i>	<i>accuracy</i>	<i>loss</i>	<i>val_accuracy</i>	<i>val_loss</i>
100	0.7767	0.7069	0.8396	0.6721
200	0.8704	0.3756	0.8770	0.5733
300	0.9111	0.3108	0.9198	0.6130

Untuk mengukur performa model DFNN juga peneliti menggunakan grafik *tensorboard* dari *library tensorflow* dengan hasil sebagai pada Gambar 6 dan Gambar 7.



Gambar 6. Grafik *accuracy* [6]

Pada Gambar 6 dan Gambar 7 merupakan grafik yang menunjukkan nilai *loss*, *accuracy*, *val\_loss*, *val\_accuracy* model menggunakan *epoch* 300 dan *batch size* 25.



Gambar 7. Grafik *loss* [7]

Berdasarkan hasil tersebut *Training* dan *Validation* mendekati angka satu. Sedangkan

Training dan Validation loss mendekati angka 0. Hal ini menunjukkan performa model sangat baik dan tidak terdapat *overfitting* dan *underfitting*.

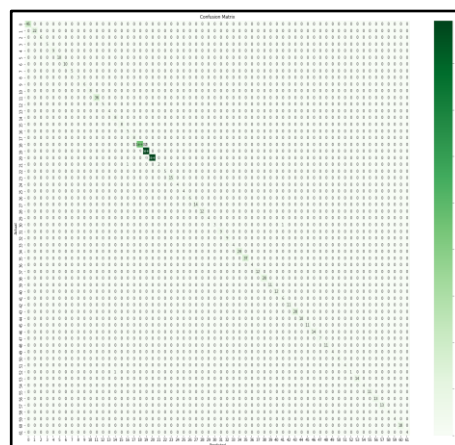
sebesar 0.99.

Tabel 9. Hasil Evaluasi

	<i>n</i>	<i>precisio</i>	<i>reca</i>	<i>f1-</i>	<i>suppo</i>
			<i>ll</i>	<i>score</i>	<i>rt</i>
0	1.00	1.0	1.0	1.	45
1	1.00	1.0	1.0	1.	22
2	1.00	1.0	1.0	1.	6
3	1.00	1.0	1.0	1.	5
4	1.00	1.0	1.0	1.	9
5	1.00	1.0	1.0	1.	10
6	1.00	1.0	1.0	1.	10
7	1.00	1.0	1.0	1.	5
8	1.00	1.0	1.0	1.	3
9	1.00	1.0	1.0	1.	6
....	...				
58	1.00	1.0	1.0	1.	4
59	1.00	1.0	1.0	1.	4
60	1.00	1.0	1.0	1.	18
61	1.00	1.0	1.0	1.	4
<i>Accura</i>				1.	1185
<i>cy</i>				00	
<i>Macro</i>	0.99	0.9	0.9	0.	1185
		8	99		
<i>Weighted</i>	1.	1.0	1.00	1.00	11
	00	0			85

**Evaluasi**

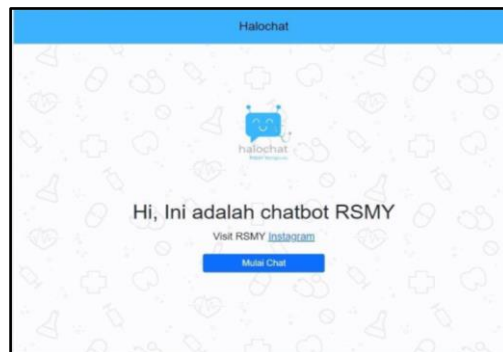
Evaluasi model penelitian kali ini menggunakan *confusion matrix* dengan hasil seperti yang ada pada Gambar 8. Evaluasi selanjutnya adalah melakukan perhitungan, *precision*, *recall*, dan *f1-score*, dapat dilihat pada tabel 9. Dari Tabel 9 didapat nilai rata-rata bobot dari 62 kelas klasifikasi yang terdiri dari *precision* sebesar 0.99, *recall* sebesar 0.98 dan *f1-score*



Gambar 8. Hasil *Confusion matrix* dengan model DFNN [8]

**Deployment**

Dengan menggunakan *Framework flask*, peneliti membuat *web* dengan 2 *user interface* yang terlihat pada Gambar 9 dan Gambar 10. Gambar 9 merupakan *user interface home* dari *Chatbot RSMY* dimana terdapat logo, tombol untuk mengunjungi *Instagram* dari RSUD M. Yunus, dan juga tombol untuk memulai *chat* dengan *chatbot*.



Gambar 9. *Interface Home* [9]



Gambar 10. *Interface Chat* [10]



Gambar 10 merupakan *user interface chat* dari *Chatbot RSMY* dimana terdapat nama Icha sebagai *bot* yang merupakan singkatan dari "*Information Chatbot*". Lalu gelembung *chat* di sebelah kiri merupakan respon dari *bot* sedangkan gelembung *chat* sebelah kanan merupakan pertanyaan atau *text input* dari *user*. Di bawahnya terdapat *box* untuk memasukkan pesan dan tombol kirim untuk mengirim pesan yang sudah dimasukkan.

#### 4. KESIMPULAN

Pada penelitian ini telah dikembangkan *chatbot* untuk membantu menjawab beberapa pertanyaan yang diajukan oleh berbagai pihak kepada UPIK RSUD M. Yunus. Karena *chatbot* belum digunakan secara operasional di rumah sakit ini, efektivitas dan kemudahan penggunaannya bagi *user* tidak dapat diukur. Jadi, kinerja *chatbot* hanya diukur dari keakuratannya dalam menghasilkan kalimat yang dapat dibaca pengguna dengan benar dan tepat. Selain itu, *chatbot* memiliki waktu respons yang cukup cepat dari model inferensi kurang dari satu detik. Selain itu, hasil performa model *chatbot* menunjukkan hasil yang cukup baik Skor performa untuk Algoritma *Deep Feed-Forward Neural Network* semuanya mencapai nilai 1, dengan *accuracy* 0,9111 dan *loss* 0,3108, serta *val\_accuracy* 0,9198 dan *val\_loss* 0,6130. Sedangkan hasil evaluasi *precision* sebesar 0.99, *recall* sebesar 0.98 dan *f1-score* sebesar 0.99.

#### UCAPAN TERIMA KASIH

Peneliti mengucapkan terima kasih kepada Ibu Arie Vatesia, S.T., M.T.I., Ph.D. selaku Ketua Program Studi Informatika Universitas Bengkulu, Bapak Mindani, Amd.Rad., S.Sos. selaku staf Rumah Sakit M. Yunus Bengkulu, dan terima kasih kepada Universitas Bengkulu yang telah menyediakan sarana prasarana yang diperlukan untuk melaksanakan pekerjaan tersebut. proyek. Peneliti juga berterima kasih kepada teman-teman peneliti yang telah membantu penelitian ini hingga selesai.

#### DAFTAR PUSTAKA

- [1] S. Sudianto, A. D. Sripamuji, I. Ramadhanti, R. R. Amalia, J. Saputra, dan B. Prihatnowo, "Penerapan Algoritma Support Vector Machine dan Multi-Layer Perceptron pada Klasifikasi Topik Berita," vol. 11, no. 2, hal. 84-91, 2022.
- [2] Mahardhika Chandra, Rizki Pratama, Fathan Azka Pradana, dan Alvita Bonita, "Chatbot Interaksi Rumah Sakit menggunakan FFNN," *Indones. J. Data Sci.*, vol. 3, no. 1, hal. 62-68, 2022, doi: 10.56705/ijodas.v3i1.36.
- [3] R. P. Putra, A. H. Pratomo, R. I. Perwira, U. Pembangunan, N. Veteran, dan I. Artikel, "Text Message Classification using Multiclass Support Vector Machine on Information Service Chatbot in the Informatics Department UPN ' Veteran ' Yogyakarta," vol. 19, no. 3, hal. 295-310, 2022, doi: 10.31515/telematika.v19i3.7418.
- [4] K. A. Nugraha dan D. Sebastian, "Chatbot Layanan Akademik Menggunakan K-Nearest Neighbor," *J. Sains dan Inform.*, vol. 7, no. 1, hal. 11-19, 2021, doi: 10.34128/jsi.v7i1.285.
- [5] N. Bhartiya, N. Jangid, S. Jannu, P. Shukla, dan R. Chapaneri, "Artificial Neural Network Based University Chatbot System," *2019 IEEE Bombay Sect. Signal. Conf. IBSSC 2019*, vol. 2019Januar, hal. 1-6, 2019, doi: 10.1109/IBSSC47189.2019.8973095.
- [6] N. A. Purwitasari dan M. Soleh, "Implementasi Algoritma Artificial Neural Network Dalam Pembuatan Chatbot Menggunakan Pendekatan Natural Language Parocessing," *J. IPTEK*, vol. 6, no. 1, hal. 14-21, 2022, doi: 10.31543/jii.v6i1.192.
- [7] D. A. C. Rachman, R. Goejantoro, dan F. D. T. Amijaya, "Implementasi Text Mining Pengelompokan Dokumen Skripsi Menggunakan Metode K-Means Clustering," *J. EKSPONENSIAL*, vol. 11, no. 2, hal. 167-174, 2020.
- [8] J. Zhang, J. Zhang, S. Ma, J. Yang, dan G. Gui, "Chatbot design method using hybrid word vector expression model based on real telemarketing data," *KSII Trans. Internet Inf. Syst.*, vol. 14, no. 4, hal. 1400-1418, 2020, doi: 10.3837/TIIS.2020.04.001.
- [9] "View of PENGGALIAN TEKS DENGAN MODEL BAG OF WORDS TERHADAP DATA TWITTER.pdf."
- [10] A. F. Agarap, "Deep Learning using Rectified Linear Units (ReLU)," no. 1, hal. 2-8, 2018, [Daring]. Tersedia pada: <http://arxiv.org/abs/1803.08375>
- [11] P. Muangkammuen, N. Intiruk, dan K. R. Saikaew, "Automated Thai-FAQ chatbot using RNN-LSTM," *2018 22nd Int. Comput. Sci. Eng. Conf. ICSEC 2018*, hal. 1-4, 2018, doi: 10.1109/ICSEC.2018.8712781.
- [12] M. Read Pablo Rivas, F. Advisor, dan M. Read, "Deployment and Hyper-Parameter Optimization of Chatbots," 2019.
- [13] P. Algoritma dan B. Moore, "Penerapan algoritma boyer moore pada chatbot ustaz abdul somad," 2021.
- [14] J. Homepage, Y. Saputra Wijaya, F. Zoromi Informatic, dan S. Amik Riau, "JAIA-Journal Of Artificial Intelligence And Applications Chatbot Designing Information Service for New Student

- Registration Based on AIML and Machine Learning," *JAlA-Journal Artif. Intell. Appl.*, vol. 1, no. 1, hal. 1-10, 2020.
- [15] I. W. Saputro dan B. W. Sari, "Uji Performa Algoritma Naïve Bayes untuk Prediksi Masa Studi Mahasiswa," *Creat. Inf. Technol. J.*, vol. 6, no. 1, hal. 1, 2020, doi: 10.24076/citec.2019v6i1.178.
- [16] Karsito dan S. Susanti, "Klasifikasi Kelayakan Peserta Pengajuan Kredit Rumah Dengan Algoritma Naïve Bayes Di Perumahan Azzura Residencia," *J. Teknol. Pelita Bangsa*, vol. 9, hal. 43-48, 2019.
- [17] S. Sultan, A. Issa, dan 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, hal. 2121-2129, 2020.