Implementasi Algoritma *Artificial Neural Network* Dalam Pembuatan *Chatbot* Menggunakan Pendekatan *Natural Language Processing*

(Implementation Of Artificial Neural Network Algorithm In Chatbot Development Using Natural Language Processing Approach)

Nia Agustina Purwitasari*, Muhamad Soleh

Program Studi Teknik Informatika, Institut Teknologi Indonesia
Jl. Raya Puspiptek, Serpong, Kota Tangerang Selatan, 15413

Abstrak

Perkembangan kecerdasan buatan telah dimanfaatkan dalam berbagai bidang, salah satunya adalah bidang hukum. Hukum di Indonesia memiliki banyak peraturan perundang-undangan yang membuat masyarakat kesulitan dalam mencari dan mengetahui peraturan yang ada. Oleh sebab itu, dibutuhkan sebuah aplikasi bernama "Peraturanku" yang dapat memudahkan masyarakat dalam mencari informasi mengenai peraturan perundang-undangan. Aplikasi dilengkapi dengan fitur chatbot yang merupakan penerapan dari kecerdasan buatan. Chatbot merupakan teknologi yang mengolah masukan berupa teks dan kemudian menghasilkan kata kunci untuk memberikan respon atau jawaban. Penggunaan chatbot dapat memudahkan pengguna dalam berkomunikasi dan mendapatkan informasi mengenai pertanyaan seputar peraturan yang ada pada aplikasi. Dalam penelitian ini akan dilakukan analisis terhadap pembangunan chatbot menggunakan algoritma Machine Learning yaitu Artificial Neural Network dan pendekatan Natural Language Processing. Tahapan yang akan dilakukan adalah pengumpulan dataset, text preprocessing dan pengklasifikasian data. Tahapan text preprocessing akan diolah menggunakan pendekatan Natural Language Processing, dimana hasilnya akan digunakan dalam proses pelatihan data menggunakan algoritma Artificial Neural Network untuk menghasilkan model chatbot. Model kemudian diuji dengan menggunakan 35 percakapan yang berbeda dan berhasil menjawab tepat sebanyak 33 percakapan dan mendapatkan tingkat akurasi tinggi sebesar 94,28% dan kesalahan 5,71%. Model chatbot tersebut kemudian disimpan ke dalam file dengan format TensorFlow Lite untuk digunakan dalam aplikasi "Peraturanku".

Kata Kunci : Algoritma Artificial Neural Network, Chatbot, Natural Language Processing

Abstract

The development of artificial intelligence has been used in various fields, including the law field. Indonesia is having an enormous number of laws and regulations that makes the people having the difficulty looking up the article they intended. Therefore, it needs an application called "Peraturanku" to make the people finding out the existing regulations effortlessly. This application will be equipped with one of the artificial intelligence applications, namely chatbot. Chatbot is a technology that processes text input and obtains the keywords to provide particular answers or responses. This chatbot facilitates the public as a user to communicate easier and get information on questions about the regulations contained in the application. This research will analyze the development of chatbots using Artificial Neural Network based on Natural Language Processing. The general process consist of dataset collection, text preprocessing and data classification. The text preprocessing processed using Natural Language Processing, where the results of text preprocessing used for training data using Artificial Neural Network algorithm to generate a chatbot model. The model was then tested using 35 different conversations and succeed to answer 33 conversations, and perform 94,28% accuracy rate

and 5,71% error. The chatbot models will be saved in TensorFlow Lite format for use in "Peraturanku" application.

Keyword : Artificial Neural Network Algorithm, Chatbot, Natural Language Processing

*Penulis Korespondensi. Telp : +62 81321301463
Alamat E-mail : niaagustina335@gmail.com (Nia Agustina Purwitasari)

1. Pendahuluan

Artificial Intelligence (AI) merupakan sebuah teknik yang penggunaannya adalah untuk meniru kecerdasan yang dimiliki oleh mahluk hidup maupun benda mati yang dimodelkan di dalam mesin dan diprogram agar bisa berpikir seperti halnya manusia dan dapat menyelesaikan masalah [1]. Kecerdasan buatan atau *artificial intelligence* (AI) dikembangkan dalam berbagai bidang yang ada di dunia, salah satunya adalah bidang hukum.

Indonesia adalah negara hukum yang senantiasa mengutamakan hukum sebagai landasan semua aktivitas negara dan masyarakat [2]. Hukum di Indonesia memiliki banyak sekali aturan perundang-undangan yang bertujuan untuk mengatur jalannya pemerintahan, sehingga untuk mengetahui setiap undang-undang pun menjadi hal yang rumit. Selain itu, adanya berbagai macam aturan perundang-undangan yang tumpang tindih atau tidak tepat sehingga membuat masyarakat sulit memahaminya.

Oleh sebab itu, dibutuhkan sebuah sistem yang dapat memudahkan masyarakat untuk mendapatkan informasi mengenai hukum yang ada di Indonesia dan menghindari adanya kebingungan mengenai peraturan perundang-undangan yang tumpang tindih. Salah satunya adalah dengan membangun sebuah aplikasi bernama "Peraturanku" yang memuat berbagai aturan perundang-undangan yang ada di Indonesia, sehingga memudahkan masyarakat untuk mengetahui hukum dan peraturan apa saja yang ada di Indonesia. Selain itu, aplikasi tersebut akan dilengkapi dengan fitur *chatting bot* atau *chatbot* yang merupakan contoh penerapan dari kecerdasan buatan atau *artificial intelligence* (AI).

Chatbot adalah sebuah teknologi yang dapat mengolah masukan atau *input* berupa teks dan kemudian menghasilkan sebuah kata kunci untuk memberikan jawaban atau respon [3]. Respon yang didapatkan adalah hasil memindai kata kunci dari masukan *user* atau pengguna sehingga menghasilkan sebuah respon atau balasan yang paling sesuai atau paling mendekati, pada umumnya untuk melakukannya menggunakan pendekatan *Natural Language Processing* (NLP). Dalam pembuatan *chatbot* terdapat beberapa proses penting yaitu *training* dan *testing* menggunakan salah satu algoritma *Machine Learning* yaitu *Artificial Neural Network* (ANN) serta menggunakan peranan TensorFlow [4].

Berdasarkan uraian permasalahan diatas, pada penelitian ini akan dibuat sebuah *chatbot* menggunakan salah satu algoritma *Machine Learning* yaitu algoritma (ANN) serta menggunakan modul TensorFlow untuk melakukan *training* model. *Chatbot* yang telah dibuat akan dibuat ke dalam model TensorFlow Lite untuk dapat diaplikasikan ke dalam aplikasi *mobile* "Peraturanku".

2. Metodologi

Data yang digunakan

Data yang digunakan dalam penelitian ini merupakan kumpulan data (*dataset*) yang di masukkan secara manual dan disimpan dalam sebuah *file* dengan format JSON (*JavaScript Object Notation*)[1]. *Dataset* memiliki struktur diantaranya:

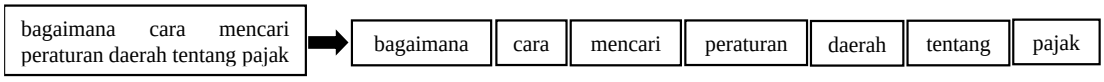
1. *Intents*, kumpulan semua data *input* dan *output* yang digunakan untuk melatih *chatbot.*
2. *Tags*, mengelompokkan data teks yang serupa dan menggunakan yang sama sebagai keluaran yang ditargetkan untuk melatih jaringan neural.
3. *Patterns*, bagian yang berisi data pola *input* yang diinginkan pengguna.
4. *Responses*, bagian yang berisi data pola *output* yang akan dikirimkan *chatbot* pada pengguna.

Pengolahan data *text preprocessing*

Pada tahapan ini teks akan diolah menggunakan pendekatan *Natural Language Processing* (NLP) dan *Natural Language Toolkit* (NLTK), sebelum digunakan pada algoritma

Artificial Neural Network (ANN). Data dengan teks yang panjang akan diolah melalui beberapa proses/tahapan, diantaranya:

1. *Case Folding*, proses pengubahan semua karakter menjadi huruf kecil serta menghilangkan karakter selain huruf [5].
2. *Tokenizing*, proses memecahkan kalimat menjadi beberapa kata yang disebut juga token. Setiap kata pada proses *tokenizing* akan di identifikasi berdasarkan spasi [6]. Proses *tokenizing* ditunjukkan pada Gambar 1.
3. *Filtering*, proses pemilihan kata penting dari hasil token dengan menghilangkan kata yang tidak memiliki arti atau disebut juga *stopwords* [5].
4. *Stemming*, proses untuk melakukan pemotongan awalan atau akhiran kata dan diubah ke bentuk akarnya yang bertujuan untuk menghilangkan imbuhan [6].
5. *Bag of Words*, atau kantung kata adalah proses pengumpulan data yang telah diproses sebelumnya kemudian memodelkan setiap data dengan menghitung kemunculan dari setiap data [7]. Penerapan *bag of words* dapat dilihat pada Tabel 1**.** dimana data dalam *bag of words vector* hanya diambil 10 data dari 132 data.



Gambar 1. Proses pemecahan kalimat (*tokenizing*)

Tabel 1. Proses *bag of words*

Kata	*Bag of Words Vector*									
	bagaimana	cari	simpan	cara	peraturan	gubernur	daerah	tentang	covid	pajak
bagaimana	1	0	0	0	0	0	0	0	0	0
cara	0	0	0	1	0	0	0	0	0	0
cari	0	1	0	0	0	0	0	0	0	0
peraturan	0	0	0	0	1	0	0	0	0	0
daerah	0	0	0	0	0	0	1	0	0	0
tentang	0	0	0	0	0	0	0	1	0	0
pajak	0	0	0	0	0	0	0	0	0	1

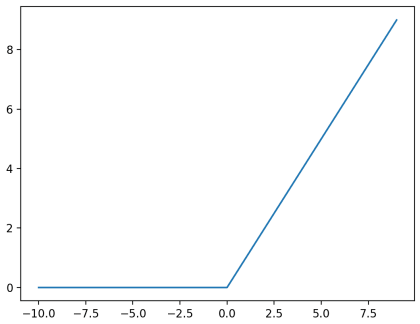
Algoritma *Artificial Neural Network* (ANN)

Artificial Neural Network (ANN) atau jaringan syaraf tiruan merupakan algoritma komputasi yang dapat memecahkan masalah kompleks yang meniru proses otak dalam cara yang disederhanakan dan terdiri dari *neuron* atau *node* buatan yang merupakan unit pemroses informasi yang tersusun berlapis-lapis dan saling berhubungan dengan bobot sinaptik/koneksi. Model arsitektur ANN adalah jaringan 3 (tiga) lapis dari *node* yang saling berhubungan, yaitu lapisan *input* (*input layer*), lapisan tersembunyi (*hidden layer*), dan lapisan *output* (*output layer*) [8]. Dalam setiap *layer* ANN terdapat fungsi aktivasi. Fungsi aktivasi adalah fungsi yang akan menentukan keluaran suatu *neuron* yang berbentuk *linear* atau *nonlinear*. Terdapat beberapa fungsi aktivasi yaitu ReLU (*Rectified Linear Unit*), *Softmax*, *Sigmoid*, *Linear*, dan sebagainya [9]. Fungsi aktivasi yang akan digunakan pada penelitian ini diantaranya:

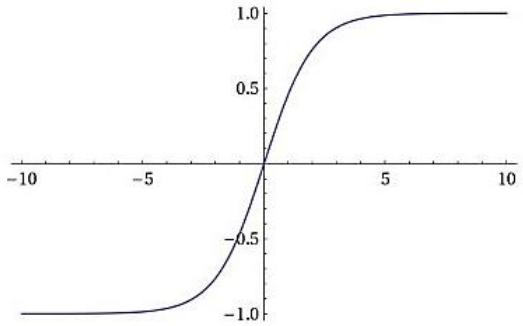
1. Fungsi aktivasi ReLU

Fungsi ReLu (*Rectified Linear Unit*) adalah fungsi aktivasi yang mempunyai perhitungan sederhana. Jika terdapat elemen dengan nilai negatif maka nilai akan diubah menjadi 0, serta tidak ada operasi eksponensial, perkalian, atau pembagian. Fungsi aktivasi ReLu di rumuskan pada persamaan (1) dan divisualisasikan pada Gambar 2. berikut [9]:

$$f(x) = ma\,x(0, x)\; atau$$
$$f(x) = \begin{cases} 0\; untuk\; x \leq 0 \\ x\; untuk\; x > 0 \end{cases} \qquad (1)$$



Gambar 2. Grafik Fungsi Aktivasi ReLU



Gambar 3. Grafik Fungsi Aktivasi *Softmax*

2. Fungsi Aktivasi *Softmax*

Fungsi aktivasi *softmax* merupakan fungsi yang digunakan dalam perhitungan probabilitas untuk menentukan klasifikasi multi kelas dengan output kelas yang mempunyai nilai probabilitas yang paling tinggi. *Output* yang dihasilkan fungsi aktivasi softmax mempunyai nilai probabilitas antara 0 hingga 1. Fungsi aktivasi *softmax* dirumuskan pada persamaan (2) dan divisualisasikan pada Gambar 3. berikut [9]:

$$f(Xi) = \frac{Exp(Xi)}{\sum_{j=0}^{k} Exp(Xi)}, nilai\ i\ =\ 0, 1, 2, \ldots, k \quad (2)$$

3. Hasil dan Pembahasan

Dataset

Hasil akhir dari pembuatan *dataset* menghasilkan sebuah *dataset* yang terdiri dari 50 *tags*/kategori, *patterns* sebanyak 128 kalimat, serta *reponses* sebanyak 70 kalimat. Pada Gambar 4**.** ditampilkan potongan 7 *tags*/kategori pertama pada *dataset* dalam format JSON.

Text Preprocessing

Sebelum melakukan pengolahan data *text preprocessing*, *dataset* harus di*import* terlebih dahulu seperti yang ditampilkan pada Gambar 5**.** *Dataset* JSON yang telah dibuat sebelumnya akan di*import* agar dapat digunakan pada pengolahan data *text preprocessing*.


```
{"intents":[
    {"tag": "noanswer",
        "patterns": [""],
        "responses": ["Mohon Maaf, saat ini aku hanya dapat menjawab mengenai Peraturan Daerah, Peraturan Gubernur dan Keputusan Gubernur saja.",
                      "Maaf, aku tidak mengerti. Bisa kamu ulangi?", "Aku tidak yakin apa maksudmu"]
    },
    {"tag": "greetings",
        "patterns": ["hai", "halo", "permisi", "hei", "test", "hi"],
        "responses": ["Halo!, apa yang bisa aku bantu?", "Hai, senang bertemu denganmu! ketik tanya jika kamu ingin bertanya.",
                      "Hai, ada yang bisa aku bantu?", "Selamat datang di Peraturanku Bot! Apa yang bisa aku bantu?"]
    },
    {"tag": "farewell",
        "patterns": ["sampai nanti", "selamat tinggal", "sampai jumpa", "bye"],
        "responses": ["Sampai bertemu lain kali!", "Sampai jumpa!", "Selamat tinggal :)", "Sedihnya harus berpisah:("]
    },
    {"tag": "thanks",
        "patterns": ["trims", "makasih", "terimakasih", "terima kasih", "terima kasih sudah membantu", "thanks", "thank you"],
        "responses": ["Senang bisa membantu:)", "Tentu, tidak masalah", "Terimakasih kembali!"]
    },
    {"tag": "sorry",
        "patterns": ["maafkan aku", "maaf mengganggu", "sorry"],
        "responses": ["Tentu, tidak masalah", "Tidak apa-apa"]
    },
    {"tag": "agree",
        "patterns": ["oke", "baiklah", "iya", "ok"],
        "responses": ["Oke, apa ada lagi yang dapat aku bantu?"]
    },
    {"tag": "disagree",
        "patterns": ["tidak", "cukup", "tidak ada"],
        "responses": ["Baiklah, untuk menyudahi sesi dengan bot silahkan ketik (stop) jika masih ingin bertanya ketik (tanya)"]
    },
```


Gambar 4. Potongan *dataset* dalam bentuk .JSON

```
import json
intents=json.loads(open('intents.json').read())
```

Gambar 5. *Import dataset*

Setelah melakukan *import dataset*, data akan diproses dan diolah dalam *text preprocessing* menggunakan *library Natural Language Toolkit* (NLTK) yang merupakan penerapan dari *Natural Language Processing* (NLP). Hasil dari proses *text preprocessing* didapatkan sebanyak 132 token atau kata unik dan 50 label atau kategori. Hasil proses *text preprocessing* dapat dilihat pada **Gambar 6.** Kemudian data akan diubah menjadi matriks *bag of words* berbentuk *array*, dimana hasil dari konversi tersebut berisi angka 0 dan 1. Angka pada *array* akan bernilai 1 apabila sebuah kata dari pola saat ini berada di *pattern* yang ditetapkan, dan akan bernilai 0 apabila tidak berada pada *pattern* yang ditetapkan. Data akan dibagi menjadi 2 (dua) variabel yaitu *train_x* dan *train_y*. Dimana *train_x* akan menampung isi matriks dari token atau kata unik, dan *train_y* akan menampung isi matriks dari label atau kategori. Proses pengisian matriks *bag of words* dapat dilihat pada Gambar 7.

```
Kata Unik/Token : 132
['ada', 'aku', 'anak', 'anggaran', 'apa', 'apbd', 'assalamualaikum', 'bagaimana',
'baiklah', 'bangunan', 'bansos', 'bantu', 'bantuan', 'belanja', 'bencana', 'berapa',
'bermotor', 'bertanya', 'bisa', 'bpjs', 'buka', 'bumi', 'butuh', 'bye', 'cara', 'cari',
'corona', 'covid', 'covid19', 'cukup', 'daerah', 'dan', 'dapatkah', 'dari', 'dimana',
'disini', 'download', 'gimana', 'gubernur', 'hai', 'halo', 'hei', 'help', 'hi', 'hiburan',
'industri', 'ingin', 'ini', 'itu', 'iya', 'jaminan', 'jumpa', 'kamu', 'kapan', 'kasih',
'kekerasan', 'kendaraan', 'kepemudaan', 'kepgub', 'keputusan', 'kerja', 'kesehatan',
'kesejahteraan', 'ketenagakerjaan', 'ktp', 'lain', 'lakukan', 'lalu', 'lintas', 'maaf',
'maafkan', 'makasih', 'mana', 'mau', 'membantu', 'membantuku', 'mengenai', 'mengganggu',
'motor', 'mrt', 'nanti', 'nomor', 'ok', 'oke', 'olahraga', 'pajak', 'pariwisata',
'parkir', 'pasar', 'pegawai', 'pemuda', 'pendapatan', 'peraturan', 'peraturanku', 'perda',
'perempuan', 'pergub', 'perindustrian', 'perlindungan', 'permisi', 'perparkiran',
'reklame', 'restoran', 'retribusi', 'rokok', 'saja', 'sampah', 'sampai', 'selain',
'selamat', 'siapa', 'sorry', 'sosial', 'sudah', 'tanya', 'tenaga', 'tentang', 'terbaru',
'terima', 'terimakasih', 'test', 'thank', 'thanks', 'tidak', 'tinggal', 'tolong',
'transportasi', 'trim', 'unduh', 'virus', 'yang', 'you']

Kategori : 50
['agree', 'ask', 'ask_peraturan', 'disagree', 'download', 'farewell', 'greetings',
'greetings2', 'help', 'kep_gub', 'kepgub_covid', 'lainnya', 'noanswer', 'options',
'peraturanku', 'perda', 'perda_apbd', 'perda_bencana', 'perda_covid', 'perda_kepemudaan',
'perda_kesejahteraan_sosial', 'perda_ketenagakerjaan', 'perda_ktp', 'perda_lalulintas',
'perda_mrt', 'perda_olahraga', 'perda_pajak_bumi_bangunan', 'perda_pajak_hiburan',
'perda_pajak_motor', 'perda_pajak_reklame', 'perda_pajak_restoran', 'perda_pajak_rokok',
'perda_pariwisata', 'perda_parkir', 'perda_pasar', 'perda_perempuan_anak',
'perda_perindustrian', 'perda_reklame', 'perda_retribusi_daerah', 'perda_sampah',
'perda_transportasi', 'pergub', 'pergub_bansos', 'pergub_bpjs', 'pergub_covid',
'ruang_lingkup', 'sorry', 'thanks', 'when', 'who']
```

Gambar 6. Hasil *text preprocessing*

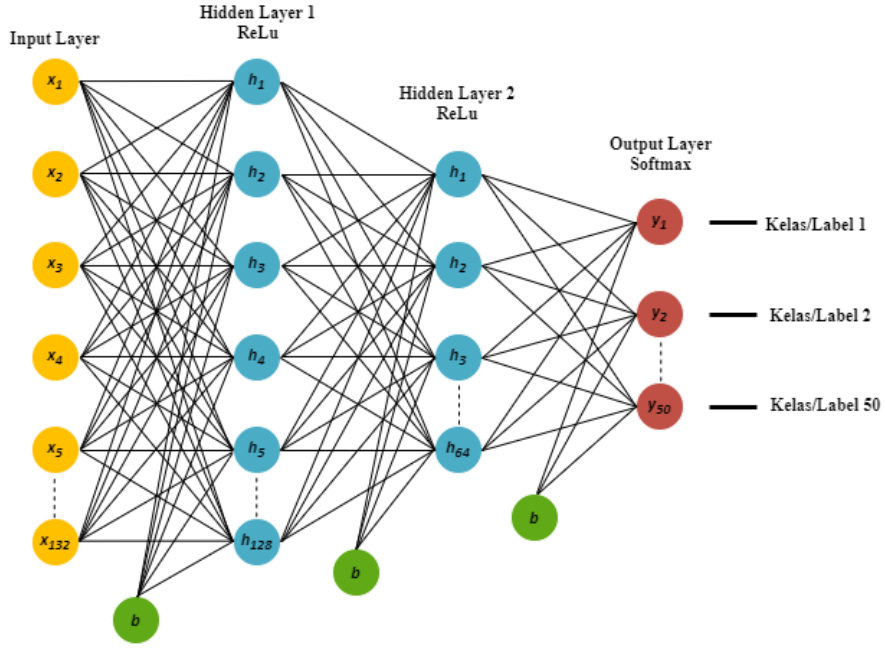
```
train_x = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
           0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
           0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
           0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0]

train_y = [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
           0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

Gambar 7. Hasil matriks *bag of words*

Model *Artificial Neural Network* (ANN)

Dalam penelitian ini terdapat 4 (empat) *layer* yang akan digunakan dalam proses *training* data untuk pengembangan *chatbot*. *Layer* pertama merupakan *input layer*, *input layer* yang digunakan berasal dari data yang telah melewati tahapan *text preprocessing*. Berdasarkan *dataset* JSON yang telah diproses sebelumnya, didapatkan 132 *neuron* sebagai *input*an untuk pelatihan data. Data tersebut merupakan hasil proses *text preprocessing* yang memiliki 132 token dan 50 kelas atau kategori sebagai *output*. *Layer* kedua merupakan *hidden layer* yang memiliki 128 *neuron* dengan fungsi aktivasi ReLu. Kemudian, *layer* ketiga juga merupakan *hidden layer* dengan 64 *neuron* dengan fungsi aktivasi ReLu. Dan *layer* yang keempat merupakan *output layer* yang memiliki 50 *neuron* dan menggunakan fungsi aktivasi *Softmax*. Pada proses pelatihan, input yang memiliki 132 *neuron* akan diproses oleh *Dense Layer* yang berada di *hidden layer*. *Dense* merupakan fungsi untuk menambahkan *Fully-Connected Layer*, sebuah lapisan dimana semua aktivitas *neuron* sebelumnya terhubung dengan *neuron* di lapisan

selanjutnya. Arsitektur ANN dari pembangunan *chatbot* tersebut dapat dilihat pada Gambar 8.

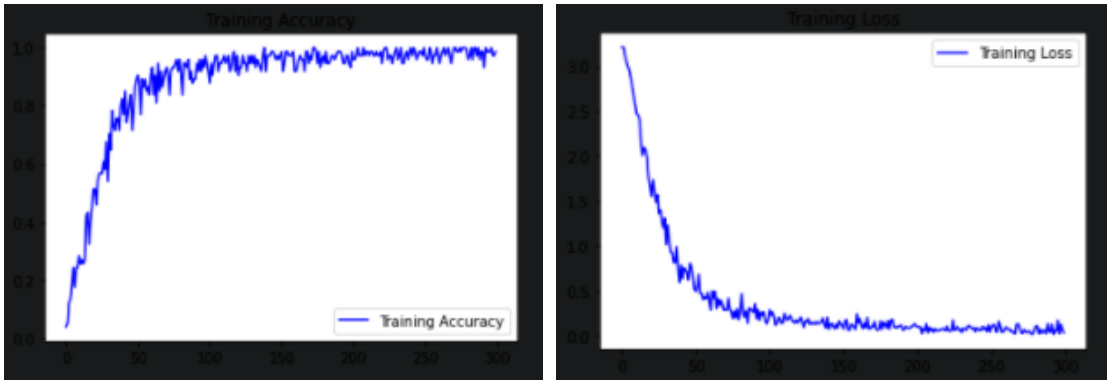


Gambar 8. Arsitektur ANN dengan 4 *layer*

Proses *training* data akan dilakukan secara berulang dan bertahap hingga mendapatkan hasil nilai akurasi yang paling tinggi dan nilai kesalahan yang semakin kecil. Informasi mengenai akurasi dan *loss* pada proses *training* data per *epoch* akan di visualisasikan dalam bentuk grafik. Dalam pelatihan data, model yang telah dibangun akan dikompilasi (*compile*) menggunakan *optimizer* SGD (*Stochastic Gradient Descent*) dan *loss function* "*Categorical Crossentropy*". Proses pelatihan data dilakukan dengan *stoping* kriteria berdasarkan *epoch* sebanyak 300 dengan *batch size* sebanyak 5. Pada **Gambar 9.** ditampilkan data yang merupakan data 10 (sepuluh) *epoch* terakhir dan **Gambar 10.** Menampilkan grafik hasil akurasi dan *loss* dari pelatihan data.

```
26/26 [==============================] - 0s 2ms/step - loss: 0.1774 - accuracy: 0.9453
Epoch 291/300
26/26 [==============================] - 0s 2ms/step - loss: 0.0841 - accuracy: 0.9766
Epoch 292/300
26/26 [==============================] - 0s 1ms/step - loss: 0.1730 - accuracy: 0.9531
Epoch 293/300
26/26 [==============================] - 0s 2ms/step - loss: 0.2108 - accuracy: 0.9375
Epoch 294/300
26/26 [==============================] - 0s 2ms/step - loss: 0.1200 - accuracy: 0.9453
Epoch 295/300
26/26 [==============================] - 0s 2ms/step - loss: 0.3202 - accuracy: 0.9062
Epoch 296/300
26/26 [==============================] - 0s 2ms/step - loss: 0.1627 - accuracy: 0.9531
Epoch 297/300
26/26 [==============================] - 0s 2ms/step - loss: 0.2449 - accuracy: 0.9375
Epoch 298/300
26/26 [==============================] - 0s 1ms/step - loss: 0.3779 - accuracy: 0.8828
Epoch 299/300
26/26 [==============================] - 0s 2ms/step - loss: 0.2008 - accuracy: 0.9531
Epoch 300/300
26/26 [==============================] - 0s 2ms/step - loss: 0.1332 - accuracy: 0.9688
```

Gambar 9. Hasil *compile* model



Gambar 10. Grafik *training accuracy* dan *training loss*

Pengujian sistem

Pengujian sistem dilakukan untuk memastikan sistem yang telah dibangun dapat berjalan dengan baik sesuai dengan analisis dan perancangan. Pada tahap pengujian ini, model yang telah diimplementasi akan diuji coba dengan melakukan proses tanya jawab seputar peraturan yang ada dalam aplikasi peraturanku. Pengujian akan dilakukan dengan menguji 35 kalimat pertanyaan atau percakapan untuk melihat apakah sistem *chatbot* dapat menjawab dengan sesuai atau tidak. Dari 35 kalimat pertanyaan didapatkan 33 kalimat percakapan yang berhasil menampilkan respon dengan benar dan sesuai dan terdapat 2 kalimat percakapan yang menampilkan respon tidak sesuai. Pengujian model sistem *chatbot* tersebut menghasilkan sebuah akurasi yang cukup tinggi yaitu sebesar **94,28%** dan kesalahan **5,71%.** Model Perhitungan akurasi dan kesalahan tersebut didapatkan dengan menggunakan perhitungan (3) dan (4). Berdasarkan performa model pada proses training dan testing, model yang dikembangkan menggunakan algoritma ANN menggunakan pendekatan NLP tidak menunjukan overfitting.

Overfitting terjadi ketika pelatihan model menghasilkan nilai akurasi yang tinggi, sedangkan akurasi pada proses pengujian model menunjukan hasil yang rendah. Hal ini dapat terjadi karena pada proses training mengabaikan *noise* atau *varians* (10).

$$Akurasi = \frac{Jumlah\ pengujian\ yang\ benar}{Jumlah\ seluruh\ pengujian} x100\ \% \quad (3)$$

$$Akurasi = \frac{33}{35} x100\ \%$$
$$Akurasi = 94{,}28\ \%$$

$$Kesalahan = \frac{Jumlah\ pengujian\ yang\ tidak\ benar}{Jumlah\ seluruh\ pengujian} x100\ \% \quad (4)$$

$$Kesalahan = \frac{2}{35} x100\ \%$$
$$Kesalahan = 5{,}71\ \%$$

Berikut ini merupakan contoh daftar hasil pengujian dari 5 pertanyaan ditampilkan pada Tabel 2.

Tabel 2. Hasil Pengujian Chatbot

No.	Inputan User	Respon Chatbot	Keterangan
1.	Assalamualaikum	Waalaikumsalam Wr. Wb	Berhasil
2.	Halo	Hai, ada yang bisa aku bantu?	Berhasil
3.	Disini ada peraturan apa aja ya?	Saat ini aplikasi Peraturanku memiliki Peraturan Daerah, Peraturan Gubernur dan Keputusan Gubernur.	Berhasil
4.	Apa ada peraturan daerah mengenai pajak?	Peraturan Daerah tentang Pajak Hiburan terdapat pada Peraturan Daerah Nomor 3 Tahun 2015. Silahkan kembali ke menu utama dan pilih Peraturan Daerah.	Berhasil
5.	Apa ada peraturan tentang perdagangan?	Peraturan Daerah, Peraturan Gubernur dan Keputusan Gubernur.	Gagal

Proses Konversi

Setelah melakukan pelatihan dan pengujian dari model sistem *chatbot*, pada tahapan ini akan dilakukan proses konversi model. Model yang yang sebelumnya telah dilatih dan diuji akan dikonversi menjadi model dengan format **.tflite** atau TensorFlow Lite yang dibantu oleh *library* TensorFlow. Proses ini dilakukan agar model yang telah dilatih dan diuji dapat diintegrasikan pada aplikasi *mobile* "Peraturanku". Proses ini akan mengonversi model pada *file* **"chatbot.h5"** menjadi **"chatbot.tflite".** Proses konversi tersebut akan ditampilkan pada Gambar 11. sebagai berikut.

```
import tensorflow as tf

tflite_converter = tf.lite.TFLiteConverter.from_keras_model(model)
tflite_model = tflite_converter.convert()

with open('chatbot.tflite', 'wb') as f:
  f.write(tflite_model)
```

Gambar 11. Proses konversi model

4. Kesimpulan

Berdasarkan hasil yang di dapat dari keseluruhan rangkaian penelitian ini, didapatkan beberapa kesimpulan mengenai uji coba dan implementasi sistem *chatbot* dengan menggunakan algoritma ANN, diantaranya:

1. Algoritma ANN dapat diimplementasikan untuk membangun sebuah model *chatbot* sehingga mampu memberikan respon yang sesuai dengan inputan pengguna dengan tingkat akurasi dari hasil uji coba sebesar 94.28%.
2. Tingkat akurasi dapat dipengaruhi dengan banyaknya sampel data, dimana semakin banyak data yang dimiliki maka tingkat akurasi akan semakin tinggi.

Pada proses *training* data akan dihasilkan sebuah model dengan format .h5, namun model tersebut tidak dapat diimplementasikan pada aplikasi *mobile*. Agar model dapat digunakan dalam aplikasi *mobile*, maka model yang telah diuji akan dikonversikan menggunakan TensorFlow Lite sehingga menjadi model dengan format .tflite

Ucapan Terima Kasih

Terima kasih kepada Kementrian Pendidikan dan Kebudayaan Indonesia serta bekerja sama dengan Google, Gojek, Tokopedia, dan Traveloka yang telah mendanai dan menyelenggarakan program Bangkit Academy 2021 sehingga penelitian Tugas Akhir ini dapat terlaksana.

Daftar Pustaka

[1] A. Ahmad, "Mengenal *artificial intelligence*, *machine learning*, *neural network*, dan *deep learning*," dalam *Jurnal Teknologi Indonesia*, 2017.

[2] G. A. Lambonan, R. Sengkey, dan X. B. N. Najoan, "Rancang bangun aplikasi ensiklopedia hukum indonesia berbasis android," dalam *Jurnal Teknik Informatika*, 14(3), 2019.

[3] B. R. Ranoliya, N. Raghuwanshi, dan S. Singh, "*Chatbot for university related faqs*," dalam *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, hlm. 1525-1530, 2017.

[4] P. Devikar, "*Transfer learning for image classification of various dog breeds*," dalam *International Journal of Advanced Research in Computer Engineering and Technology (IJARCET)*, hlm. 2707-2715, 2016.

[5] D. A. C. Rachman, R. Goejantoro dan F. D T. Amijaya, "Implementasi *Text Mining* Pengelompokkan Dokumen Skripsi Menggunakan Metode *K-Means Clustering*," dalam *Jurnal EKSPONENSIAL*, 11(2), hlm. 167-174, 2020.

[6] M. Uma, V. Sneha, G. Sneha, J. Bhuvana, dan B. Bharati, "*Formation of SQL from Natural Language Query using NLP*," dalam *International Conference on Computational Intelligence in Data Science (ICCIDS)*, hlm. 1-5, 2019.

[7] W. T. H. Putri dan R. Hendrowati, "Penggalian Teks Dengan Model *Bag of Words* Terhadap Data Twitter," dalam *Jurnal Muara Sains, Teknologi, Kedokteran, dan Ilmu Kesehatan*, 2(1), hlm. 129-138, 2018.

[8] Y. I. Al Barsh, *et al*, "MPG *Prediction Using Artificial Neural Network*," dalam *International Journal of Academic Information Systems Research (IJAISR)*, hlm. 7-16, 2020.

[9] F. Mustakim, Fauziah dan N. Hayati, "Algoritma *Artificial Neural Network* pada *Text-based Chatbot Frequently Asked Question* (FAQ) *Web* Kuliah Universitas Nasional," dalam *Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)*, 5(4), hlm. 439-447, 2021.

[10] Webb G.I. (2011) Overfitting. In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-30164-8_623