

IMPLEMENTASI *CHATBOT* PADA PENDAFTARAN MAHASISWA BARU MENGGUNAKAN *RECURRENT NEURAL NETWORK*

¹Tjut Awaliyah Zuraiyah, ²Dian Kartika Utami, ³Degi Herlambang

^{1,2,3}Program Studi Ilmu Komputer, Fakultas Matematika Dan Ilmu Pengetahuan Alam, Universitas Pakuan

Jl. Pakuan, Tegallega, Kota Bogor, Jawa Barat, 16143

¹tjut.awaliyah@unpak.ac.id, ²diankartikautami@unpak.ac.id, ³ddegih@gmail.com

Abstrak

Chatbot adalah perangkat lunak yang dapat berkomunikasi dengan manusia menggunakan bahasa alami. Model percakapan menggunakan kecerdasan buatan agar mampu memahami ucapan pengguna dan memberi tanggapan yang relevan dengan masalah yang dibahas oleh pengguna. Pendaftaran mahasiswa baru memerlukan banyak informasi mengenai prosedur pendaftaran di perguruan tinggi. Website pendaftaran online di Universitas Pakuan masih sebatas berisi informasi umum. Penelitian ini bertujuan untuk membuat suatu aplikasi *Chatbot* otomatis yang dapat berkomunikasi dengan manusia mengenai informasi pendaftaran mahasiswa baru di Universitas Pakuan menggunakan *Recurrent Neural Network (RNN)* untuk klasifikasi teks. Aplikasi *Chatbot* diimplementasikan menggunakan bahasa pemrograman Python dan Telegram API. Tahapan pada implementasi *Chatbot* terdiri dari preprocessing, transformasi data ke format .JSON, pelatihan data, bag of word dan full connection. Pengujian aplikasi *Chatbot* menggunakan data sebanyak 251 kalimat pertanyaan tentang pendaftaran mahasiswa baru di Universitas Pakuan. Hasil pengujian menunjukkan bahwa *Chatbot* dapat menjawab pertanyaan mengenai pendaftaran mahasiswa baru dengan akurasi sebesar 88%, presisi sebesar 95% dan recall sebesar 92%.

Kata Kunci: *Chatbot*, pendaftaran mahasiswa baru, *RNN*, Python, Telegram API.

Abstract

Chatbot is software that can communicate with humans using natural language. The conversation model uses artificial intelligence to be able to understand the user's words and provide responses that are relevant to the problem discussed by the user. Enrollment for new students requires a lot of information about registration procedures in tertiary institutions. The online registration website at Pakuan University is still limited to general information. This study aims to create an automated *Chatbot* application that can communicate with humans about new student registration information at Pakuan University using the *Recurrent Neural Network (RNN)* for text classification. The *Chatbot* application is implemented using the Python programming language and the Telegram API. The stages in the *Chatbot* implementation consist of preprocessing, data transformation to .JSON format, data training, bag of words and full connection. Testing the *Chatbot* application uses data as much as 251 question sentences about enrolling new students at Pakuan University. The test results show that *Chatbot* can answer questions regarding registration of new students with an accuracy of 88%, a precision of 95% and a recall of 92%.

Keywords: *Chatbot*, registration of new students, Python, Telegram API.

PENDAHULUAN

Sebagian besar masyarakat ingin mendapat informasi yang mudah, cepat dan akurat. Pendaftaran calon mahasiswa baru memerlukan banyak informasi prosedur pendaftaran di setiap perguruan tinggi. Sarana informasi pendaftaran *online* di Universitas Pakuan melalui halaman *website* masih sebatas informasi umum. Akan tetapi, calon pendaftar membutuhkan komunikasi dua arah untuk menanyakan informasi lebih detail lagi. Terdapat satu tempat ruang informasi untuk melayani calon pendaftar yang ingin bertanya lebih lanjut namun tidak efisien karena pertanyaan berulang dan serupa berkaitan dengan pendaftaran. Untuk mendukung kebutuhan informasi yang akurat dan *up-to-date* terkait dengan pendaftaran mahasiswa baru, maka diperlukan sebuah sistem berbasis teknologi informasi yang dapat merangkum dengan baik berbagai data pendaftaran serta menampilkan informasinya pada pengguna.

Chatbot juga dikenal sebagai *talkbot*, *chatterbox*, *Bot*, *IM bot* atau *artificial conversational entity*. *Chatbot* adalah program komputer yang meniru percakapan manusia dalam format alami termasuk teks atau bahasa lisan menggunakan teknik kecerdasan buatan seperti *Natural Language Processing* (NLP), gambar dan pemrosesan video, serta analisis audio [1].

Chatbot dapat digunakan dalam bidang pendidikan sebagai media pembelajaran. Penelitian yang dilakukan oleh Wijaya,

Sarosa, dan Tolle mengenai pembuatan *Chatbot* sebagai media pembelajaran bahasa Java pada *Google Classroom* dan *Facebook Messenger*. Hasil pengujian *Chatbot* menunjukkan bahwa nilai ujian siswa lebih tinggi pada kelas yang menggunakan *Chatbot* dibandingkan kelas yang menggunakan pembelajaran secara konvensional [2]. Penelitian lain mengenai *Chatbot* sebagai *virtual assistant* berbasis *website* menggunakan AIML yang memberikan informasi kepada mahasiswa mengenai Program Studi Teknik Informatika, Universitas Muhamadiyah Malang. Selain itu, pada *Chatbot* dapat ditambahkan pengetahuan baru jika terdapat pertanyaan yang tidak terdapat pada basis data *Artificial Linguistic Internet Computer Entity* (ALICE). Akurasi *Chatbot* yang dibuat pada penelitian tersebut sebesar 80% [3].

Beberapa *Chatbot* tersedia tetapi pengguna harus melakukan pengaturan frase kunci secara manual, mengusulkan dan mengembangkan pertanyaan yang sering diajukan [4]. *Recurrent Neural Networks* (RNN) merupakan arsitektur *deep learning* yang populer serta sangat menjanjikan untuk menyelesaikan berbagai persoalan yang terkait dengan NLP. *Chatbot* secara otomatis menanggapi pengguna menggunakan RNN dalam bentuk *Long Short-Term Memory* (LSTM) untuk klasifikasi teks [5]. Model RNN digunakan agar mesin dapat memahami bahasa manusia, mulai dari cara berkomunikasi, mendengarkan, mengenali percakapan, hingga memahami tata bahasa dan aksen [6].

Pada penelitian ini dibuat suatu aplikasi *Chatbot* otomatis yang dapat berkomunikasi dengan pengguna mengenai informasi pendaftaran mahasiswa baru di Universitas Pakuan. Implementasi aplikasi *Chatbot* pada penelitian ini menggunakan *Recurrent Neural Network* (RNN).

METODE PENELITIAN

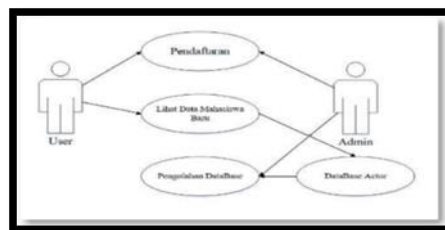
Tahap Perencanaan

Tahap proses perencanaan ini merupakan tahapan awal yang berupa pengumpulan data yang dibutuhkan dalam pembuatan *Chatbot* untuk calon mahasiswa baru Universitas Pakuan. Pada tahap ini mencari permasalahan yang terjadi menyangkut pendaftaran calon mahasiswa baru serta menentukan bagaimana solusi pemecahan masalahnya. Tahapan ini dilakukan untuk mendapatkan data pendaftaran

dan proses pendaftaran secara langsung melalui wawancara dan studi literatur. Penelitian ini mengumpulkan data secara langsung ke lapangan menggunakan teknik pengumpulan data wawancara pegawai BAAK maupun calon mahasiswa baru sehingga dapat diambil data yang diperlukan guna kepentingan penelitian ini. Selanjutnya dilakukan pembuatan sebuah dokumen yaitu kuisisioner yang akan diberikan ke setiap responden guna mencatat setiap jawaban yang nantinya akan menjadi sebuah sumber data penelitian ini.

Tahap perancangan

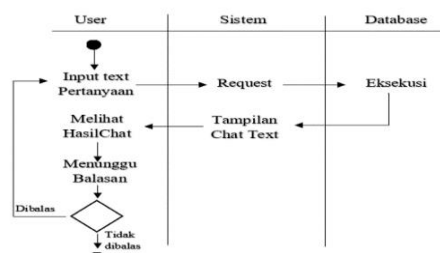
Tahapan perancangan secara detail digunakan untuk menentukan sistem yang akan dibangun meliputi pembuatan *use case diagram*, *activity diagram* dan *sequence diagram*. Gambar 2 menunjukkan *use case diagram* untuk pendaftaran mahasiswa baru.



Gambar 2. *Use Case Diagram*

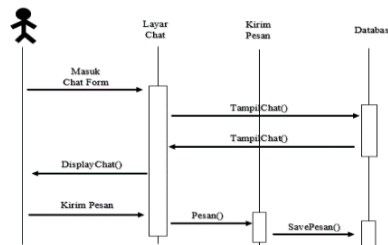
Activity diagram merupakan model analisis yang digunakan atau menggambarkan sebuah proses aktivitas. Gambar 3 menunjukkan

activity diagram aplikasi *Chatbot* pendaftaran mahasiswa baru.



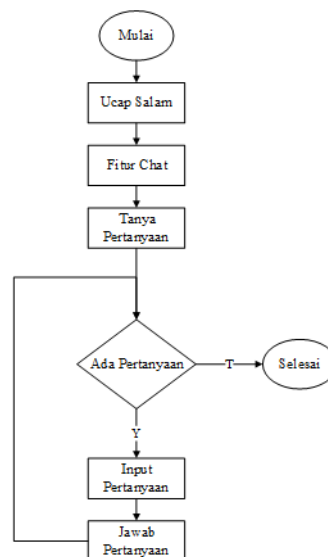
Gambar 3. *Activity Diagram Chatbot*

Sequence diagram menggambarkan menerima sebuah *message*. Gambar 4 objek dan relasinya termasuk kronologi memperlihatkan *sequence diagram Chatbot* (urutan) perubahan secara logis setelah pendaftaran mahasiswa baru.



Gambar 4. *Sequence Diagram Chatbot*

Flowchart sistem dibangun guna dapat memberikan gambaran seutuhnya dari sistem terlihat pada Gambar 5. kebutuhan data yang ada. Adapun *flowchart* sistem terlihat pada Gambar 5.



Gambar 5. *Flowchart Sistem*

Tahap Implementasi

Pembangunan aplikasi ini menggunakan pemrograman bahasa Python. Pengujian data menggunakan *tool tensorflow* atau *deep learning* guna mendapat tingkat akurasi tinggi untuk jawaban dari pertanyaan yang dimasukkan pengguna. Koneksi terhadap *Telegram API* digunakan sebagai media

hosting untuk platform *Telegram* secara *realtime*.

Implementasi *Chatbot* secara detail terbagi menjadi beberapa tahapan, yaitu tahap *preprocessing*, tahap transformasi data menjadi *file .JSON*, lalu tahap pelatihan data, tahap *bag of words*, dan tahap terakhir *full connection*.

a. *Text Preprocessing*

Tahapan *text preprocessing* adalah tahapan dimana aplikasi melakukan seleksi data yang diproses pada setiap dokumen. Proses *preprocessing* meliputi 4 tahapan yaitu *case folding*, *tokenizing*, *filtering* dan *stemming*.

b. Transformasi Data Menjadi File .JSON

Transformasi data menjadi sebuah file .JSON diperlukan untuk sistem agar dapat membaca setiap pola pertanyaan yang muncul beserta target jawabannya. Cara mentransformasikan data ke dalam file .JSON ialah dengan memisahkan setiap kata yang ada didalam data menjadi beberapa bagian yaitu:

1. *Tags* (kategori) merupakan sebuah pengetahuan atau kategori yang menjadi acuan sistem dalam menentukan respon.
2. *Pattern* (pola) adalah sebuah rangkaian huruf yang diharapkan sesuai atau cocok dengan satu atau bahkan lebih dengan masukan pengguna.
3. *Responses* (output) adalah hasil jawaban yang akan dikeluarkan berdasarkan *index tags* dan *pattern* yang ditentukan oleh sistem.

c. Pelatihan Data

Setelah data terkumpul, sistem menguji data agar mendapat tingkat akurasi *output* yang baik dan memuaskan. Langkah pelatihan data sebagai berikut:

1. Mengubah bentuk pola pada data .JSON ke dalam bentuk matriks.

2. *Flattering* atau memasukan setiap baris matriks ke dalam *node input layer* yang akan memberi sinyal ke *hidden layer*.

3. Menilai *output* setelah selesai perhitungan dari *input layer* ke *hidden layer* yang dimana akan diberikan setiap bobot oleh sistem.

d. *Bag of Words*

Pada tahap ini dilakukan transformasi nilai *bag of words* dengan mentransformasi teks *input* dari pengguna menjadi bentuk bilangan biner.

e. *Full Connected*

Setelah mendapat nilai biner dari teks *input*, maka dimasukan nilai biner tersebut ke dalam *input layer* yang nantinya akan memberi sinyal kepada *hidden layer* dari arsitektur *neural network*.

Tahap Pengujian

Pengukuran performa dari aplikasi *Chatbot* yang dibuat pada penelitian ini berdasarkan hasil *confusion matrix* dengan menghitung tingkat *accuracy*, *precision* dan *recall*.

a. *Accuracy*

Akurasi merupakan rasio prediksi benar (positif dan negatif) dengan keseluruhan data. Perhitungan akurasi menggunakan Persamaan (1)

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} * 100\% \quad (1)$$

b. *Precision*

Presisi merupakan rasio prediksi benar

positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. Perhitungan presisi menggunakan Persamaan (2).

$$Presisi = \frac{TP}{FP + TP} * 100\% \quad (2)$$

c. *Recall*

Recall merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif. *Recall* dihitung menggunakan Persamaan (3).

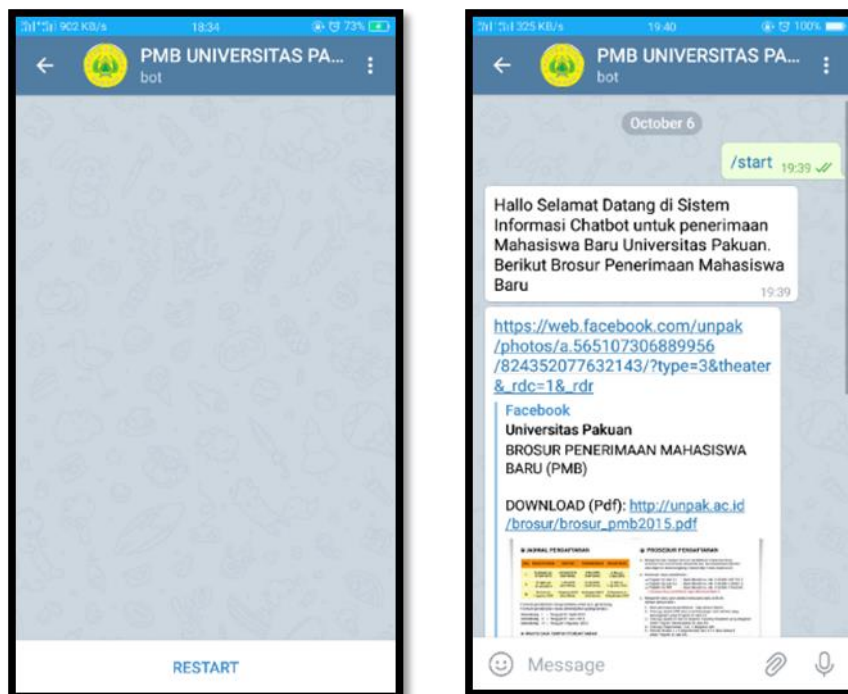
$$Recall = \frac{TP}{FN + TP} * 100\% \quad (3)$$

HASIL DAN PEMBAHASAN

Pada aplikasi *Chatbot* pendaftaran

mahasiswa baru Universitas Pakuan terdapat halaman *chat* atau pesan menggunakan *platform* Telegram. Dengan menggunakan *RNN* sistem dengan mudah menjawab pertanyaan dari pengguna. Apabila *Chatbot* tidak mengerti apa yang dimasukkan oleh pengguna maka *Chatbot* akan menanyakan kembali dan mengoreksi pertanyaan yang diajukan oleh pengguna sesuai dengan format sistem yang ada.

Halaman *chat* Telegram merupakan tampilan awal ketika pengguna membuka aplikasi Telegram terlihat pada Gambar 6 (a) dan (b). Pada saat pengguna membuka kontak *Chatbot* maka sistem akan otomatis mengirimkan gambar mengenai pendaftaran mahasiswa baru Universitas Pakuan.



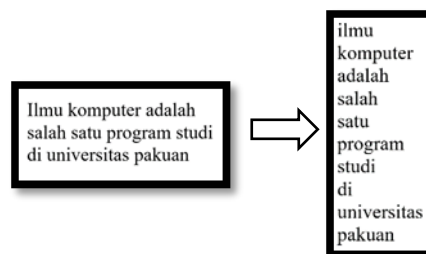
Gambar 6. (a) Halaman Telegram, (b) Halaman *Chatbot Start*

Pada tahap *text preprocessing* dimulai dengan tahap *case folding*. Dalam penelitian ini *case folding* digunakan untuk mengubah semua huruf dalam dokumen menjadi huruf kecil. Selanjutnya dilakukan tahap *tokenizing* yang merupakan tahapan pemotongan *string input* berdasarkan tiap kata yang menyusunnya. Contoh hasil tahap *tokenizing* dapat dilihat pada Gambar 7.

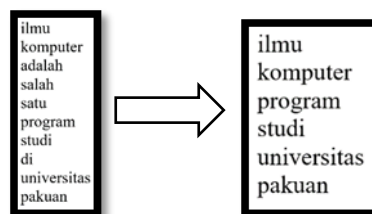
Tahap *text preprocessing* selanjutnya yaitu tahap *filtering*. Pada tahap ini digunakan

stoplist (membuang kata kurang penting) atau *wordlist* (menyimpan kata penting). Contoh *filtering* dapat dilihat pada Gambar 8.

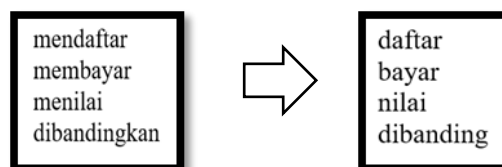
Tahap *stemming* diperlukan untuk memperkecil jumlah indeks yang berbeda dari suatu dokumen dan juga untuk melakukan pengelompokan kata-kata lain yang memiliki kata dasar dan arti yang serupa namun memiliki bentuk yang berbeda. Gambar 9 merupakan contoh hasil tahap *stemming* *Chatbot*.



Gambar 7. Hasil Tahap *Tokenizing*



Gambar 8. Hasil Tahap *Filtering*



Gambar 9. Hasil Tahap *Stemming*

Setelah data terkumpul lalu sistem menguji data agar mendapat tingkat *output* yang baik dan memuaskan. Tabel 1

merupakan contoh data pelatihan yang digunakan pada penelitian ini.

Tabel 1. Sample Pelatihan Data

Tags		Pattern		Target
Salam	Hei	apa kabar	hello	Ada yang bisa dibantu?
Daftar	bagaimana cara daftar	ilmu komputer	fasilitas	Lihat Brosur ini...
Terimakasih	terimakasih	thanks	makasih	Kapan pun!

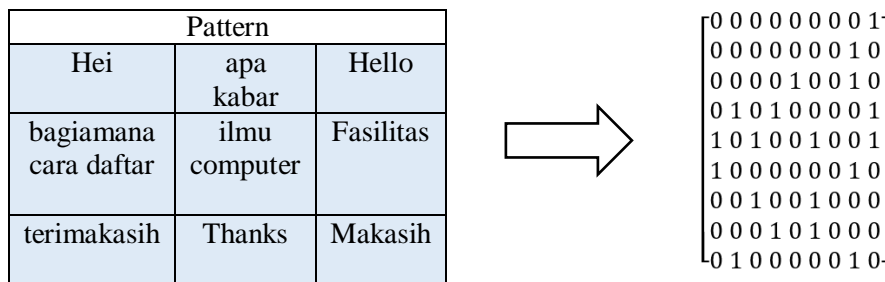
Tabel 1 merupakan contoh sebuah file .JSON akan digunakan dalam pelatihan data. Pelatihan data dilakukan agar mesin dapat mengenali *pattern* atau pola teks *input* pengguna yang nanti akan digunakan untuk membantu menjawab pertanyaan dari pengguna pada sistem *Chatbot*. Langkah-langkah pada pelatihan data sebagai berikut:

1. Mengubah bentuk pola pada data

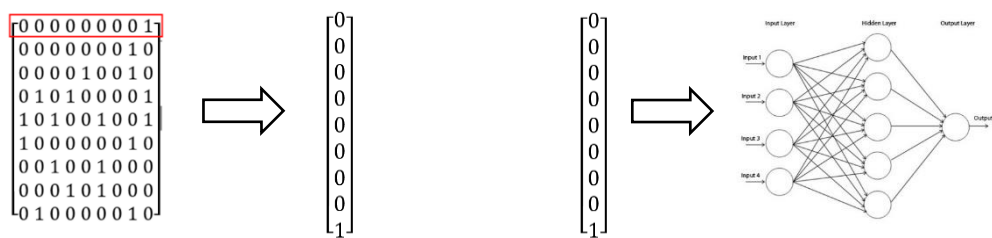
.JSON ke dalam bentuk matriks seperti pada Gambar 10.

2. *Flattering* atau memasukan setiap baris matriks ke dalam *node input layer* yang akan memberi sinyal ke *hidden layer*. Contoh hasil *flattering* dapat dilihat pada Gambar 11.

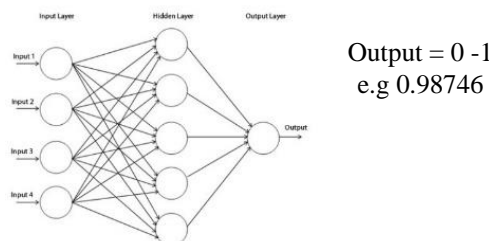
3. Nilai *output* pelatihan data terlihat pada Gambar 12.



Gambar 10. Hasil Mengubah Pola ke Matriks



Gambar 11. Hasil *Flattering*



Gambar 12. Nilai *Output* Pelatihan Data

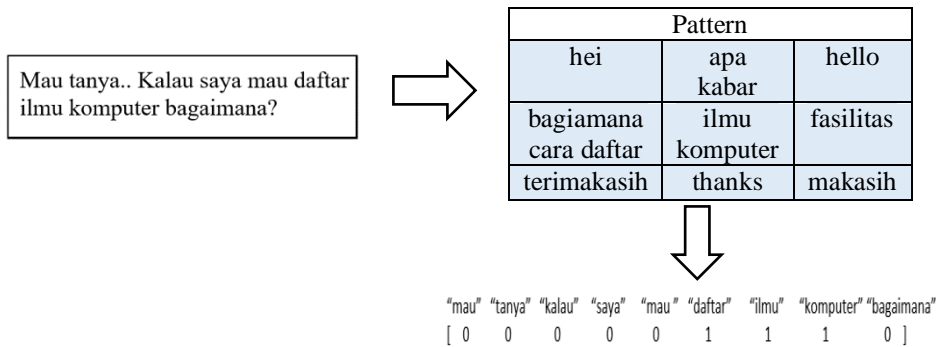
Bobot pada hidden layer tidak dapat dimasukkan langsung oleh pengguna karena bobot merupakan nilai yang otomatis diberikan oleh sistem, dan disimpan untuk dijadikan tingkat akurasi oleh pola sebelumnya.

Tahap selanjutnya dilakukan transformasi nilai *bag of words*. Contoh hasil transformasi nilai *bag of words* terlihat pada Gambar 13.

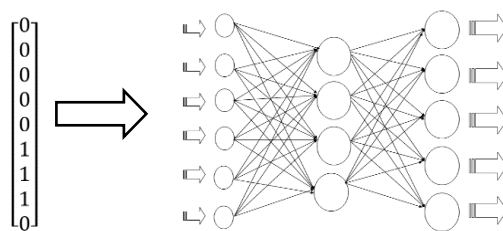
Setelah mendapat nilai biner dari teks *input*, maka pada tahap *full connected*, nilai biner tersebut dimasukan ke dalam *input layer*

yang nantinya akan memberi sinyal kepada *hidden layer* dari arsitektur *neural network* seperti pada Gambar 14. Nilai *output* ditentukan oleh sistem setelah melakukan perhitungan dari *input layer* ke *hidden layer*, sehingga dapat menentukan jawaban apa yang akan diberikan oleh sistem *Chatbot*.

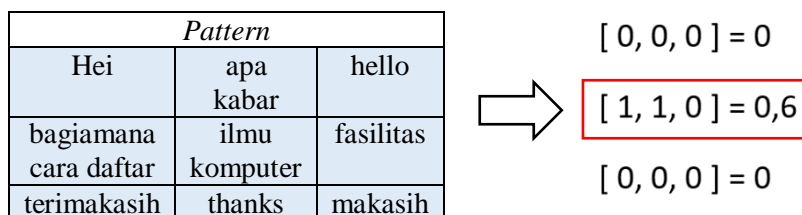
Setelah didapatkan nilai-nilai output dari setiap pola, kemudian diambil nilai yang paling tinggi untuk menjawab pertanyaan dilihat dari hasil penentuan respon seperti pada Gambar 15.



Gambar 13. Hasil Transformasi Nilai *Bag of Words*



Gambar 14. Tahap *Full Connected*



Gambar 15. Contoh Penentuan Respon

Jumlah data secara keseluruhan data uji sebanyak 50 data. Kemudian sebanyak 300 data, untuk pengujian data didapatkan hasil uji seperti terlihat pada latih sebanyak 250 dan akan diuji dengan Tabel 2.

Tabel 2. Nilai *Confusion Matrix*

	<i>Positive</i>	<i>Negative</i>
<i>Positive</i>	40	3
<i>Negative</i>	2	4

Pengukuran *performance* berdasarkan menghitung tingkat *accuracy*, *precision* dan hasil *confusion matrix* pada Tabel 2 dengan *recall* sebagai berikut:

$$\text{Akurasi} = \frac{40 + 4}{40 + 5 + 3 + 2} = \frac{44}{50} = 88\%$$

$$\text{Presisi} = \frac{40}{40 + 2} = \frac{40}{42} = 95\%$$

$$\text{Recall} = \frac{40}{40 + 3} = \frac{40}{43} = 92\%$$

Berdasarkan hasil perhitungan performa aplikasi, aplikasi memiliki akurasi sebesar 99%, presisi sebesar 95%, dan *recall* sebesar 92%.

KESIMPULAN DAN SARAN

Pada penelitian ini telah berhasil mengimplementasikan aplikasi *Chatbot* menggunakan *Recurrent Neural Networks* (RNN) untuk pelatihan data yang dapat membantu menemukan jawaban yang sesuai dari pertanyaan pengguna. JSON digunakan untuk menyimpan data serta menampilkan data-data yang tersimpan menggunakan perangkat lunak Python untuk membantu dalam pembuatan dan pengembangan aplikasi ini. Pada nilai-nilai pada *confusion matrix* didapatkan tingkat akurasi aplikasi sebesar

88%, presisi sebesar 95% dan *recall* sebesar 92%. Pembuatan aplikasi ini bertujuan untuk mempermudah pengguna dalam mendapatkan informasi yang cepat dan akurat secara *realtime*. Pengguna dapat langsung mendapatkan informasi saat itu juga dengan menggunakan aplikasi Telegram tanpa harus datang jauh ke kampus yang cukup jauh bagi sebagian orang.

Penelitian selanjutnya dapat mengembangkan aplikasi *Chatbot* dari sisi keamanan aplikasi. Selain itu, pada penelitian selanjutnya dapat menggunakan metode *deep learning* lainnya sehingga memberikan hasil akurasi yang lebih baik.

DAFTAR PUSTAKA

- [1] K. Bala, M. Kumar, S. Hulawale, dan S. Pandita, "Chat-bot for college management

- system using A.I,” *International Research Journal Of Engineering and Technology (IRJET)*, vol. 4, no. 11, hal. 2030 – 2033, 2017.
- [2] M. H. Wijaya, M. Sarosa, dan H. Tolle, “Rancang bangun Chatbot pembelajaran Java pada Google Classroom dan Facebook Messenger,” *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, vol. 5, no. 3, hal. 287 – 296, 2018.
- [3] Maskur, “Perancangan Chatbot pusat informasi mahasiswa menggunakan AIML sebagai virtual assistant berbasis web”, *KINETIK*, vol. 1, no. 3, hal. 123 – 128, 2016.
- [4] S. P. Lende dan M. M. Raghuwanshi, “Question answering system on education acts using NLP techniques,” Dipresentasikan pada IEEE World Conference on Futuristic Trends in Research and Innovation for Social Welfare, Coimbatore, India, 2016.
- [5] P. Muangkammuen, N. lutiruk, dan K. R. Saikaew, “Automated Thai-FAQ Chatbot using RNN-LSTM,” In Proc. 22nd IEEE International Computer Science and Engineering Conference (ICSEC), 2018, hal. 1 – 4.
- [6] P. Tanwar, T. V. Prasad, dan K. Datta, “An effective reasoning algorithm for question answering system,” *International Journal of Advanced Computer Science and Applications (IJACSA) Special Issue on Natural Language Processing*, hal. 52 – 57, 2014.