

Implementasi Neural Network untuk Pembuatan Chatbot Menggunakan Dataset Pertanyaan Mahasiswa

Immanuel Ruben

Informatika, Fakultas Industri Kreatif Institut Teknologi dan Bisnis Kalbis
Jalan Pulomas Selatan Kav, 22 Jakarta Timur, 13210
Email : immanuelrubenn@gmail.com

Abstract: This research develops a machine learning model for chatbot and chatbot applications. Chatbot is an area of application of Machine Learning and Natural Language Processing. Modelling is done by using the Deep Learning algorithm. The training data used for training is obtained by observation in the student manual of Kalbis Institute. The model created produces an accuracy of 98% and loss value of 0.05275. The model created was tested in the form of a chatbot application by giving various questions relating to topic in the student manual of Kalbis Institute.

Keywords: Chatbot, Natural Language Processing, Deep Learning, Machine Learning

Abstrak: Penelitian ini membuat model pembelajaran mesin untuk chatbot serta aplikasi chatbot. Chatbot merupakan bidang penerapan Pembelajaran Mesin dan Natural Language Processing. Pembuatan model dilakukan dengan menggunakan algoritma Deep Learning. Data latih yang digunakan untuk pembelajaran algoritma didapatkan dengan cara observasi pada buku panduan mahasiswa Kalbis Institute. Model yang dibuat menghasilkan akurasi sebesar 98% dan loss value sebesar 0,05275. Model yang dibuat diuji dalam bentuk aplikasi "chatbot" dengan cara diberikan berbagai pertanyaan yang berhubungan dengan hal-hal dalam buku panduan mahasiswa Kalbis Institute dalam Bahasa Indonesia.

Kata kunci: Chatbot, Natural Language Processing, Deep Learning, Machine Learning

I. PENDAHULUAN

Buku panduan mahasiswa merupakan salah satu media yang berperan penting dalam perguruan tinggi karena sebagai sumber informasi mengenai tata tertib dan persyaratan yang berlaku dalam perguruan tinggi tersebut. Buku panduan ini biasanya berbentuk buku atau dokumen digital yang bisa didapatkan melalui website kampus. Hanya saja, buku panduan berisi terlalu banyak informasi karena menjelaskan setiap aturan secara detail. Terkadang pengguna harus membaca semua halaman untuk mendapatkan informasi yang dibutuhkan. Hal ini membuat akses ke informasi menjadi kurang praktis dan kurang interaktif.

Mahasiswa juga dapat bertanya langsung ke fasilitas yang disediakan oleh pihak kampus yaitu Layanan Mahasiswa. Mahasiswa dapat bertanya langsung ke petugas di Layanan Mahasiswa, melalui telepon, atau melalui email. Namun Layanan Mahasiswa memiliki jam kerja oleh sebab itu Layanan Mahasiswa tidak dapat menjawab pertanyaan mahasiswa diluar jam kerja. Semenjak Indonesia terutama Jakarta terkena dampak dari pandemi virus Corona, pihak kampus memutuskan untuk melakukan pembelajaran dari rumah. Mahasiswa tidak dapat ke kampus untuk bertanya langsung ke Layanan Mahasiswa dan hanya dapat bertanya melalui telepon atau email.

Perkembangan aplikasi “chatbot” saat ini banyak dimanfaatkan untuk membantu pekerjaan manusia. Aplikasi “chatbot” telah digunakan dalam banyak hal, termasuk memberikan edukasi, hiburan, dan masih banyak lagi. Aplikasi “chatbot” juga telah terbukti menjadi alat yang berguna dalam membantu kegiatan manusia. Satu contoh yaitu pada pengembangan yang dilakukan oleh Marwan Noor Fauzy dan Kusri. Mereka menunjukkan potensi pada chatbot untuk diaplikasikan untuk menjadi *virtual assistant* pada pusat layanan informasi akademik [1].

Dalam pengembangan aplikasi “chatbot”, tantangan tersulitnya adalah membuat model yang dapat menyesuaikan tanggapan dengan konteks. Hubungan semantik antara pertanyaan pengguna dan respons yang sesuai merupakan elemen yang penting untuk pemodelan percakapan dalam aplikasi “chatbot”. Respons yang dihasilkan merupakan hasil deteksi kata kunci pada pertanyaan pengguna dan menghasilkan jawaban yang dianggap paling cocok [2]. Tantangan yang lain adalah pengumpulan data berupa pertanyaan dan jawaban yang nantinya berguna sebagai pengetahuan dasar dari aplikasi “chatbot”. Selain itu pemilihan algoritma yang tepat juga penting agar aplikasi “chatbot” dapat berjalan dengan efektif.

Dengan menggunakan kombinasi dari teknologi *Natural Language Processing* (NLP) dan *machine learning*, memungkinkan komputer untuk mengekstrak makna dari pertanyaan yang diberikan pengguna. Kombinasi dari kedua teknologi ini juga memungkinkan untuk melatih aplikasi “chatbot” dalam menghadapi pertanyaan yang bermacam-macam. Dengan adanya data *training* yang berguna untuk memberi pengetahuan dasar yang lebih baik dan memungkinkan aplikasi “chatbot” untuk menjawab pertanyaan yang tepat [2]. Cara seperti ini biasa

dikenal dengan sebutan *rule based chatbot*.

Penelitian ini mengenai pembuatan model pembelajaran mesin dan aplikasi “chatbot” yang dapat mendeteksi pertanyaan-pertanyaan sederhana dan dapat menjawab pertanyaan terkait persoalannya yaitu yang berhubungan dengan buku panduan mahasiswa. Dalam kasus ini, model pembelajaran mesin dan aplikasi “chatbot” akan dapat mendeteksi pertanyaan seputar aturan-aturan dan persyaratan yang ada dalam perguruan tinggi dan dapat memberi jawaban yang tepat berdasarkan buku panduan mahasiswa.

Maka persoalan pada penelitian ini adalah mengenai bagaimana menerapkan model pembelajaran mesin pada aplikasi “chatbot” sebagai pendukung fasilitas layanan mahasiswa pada Kalbis Institute?

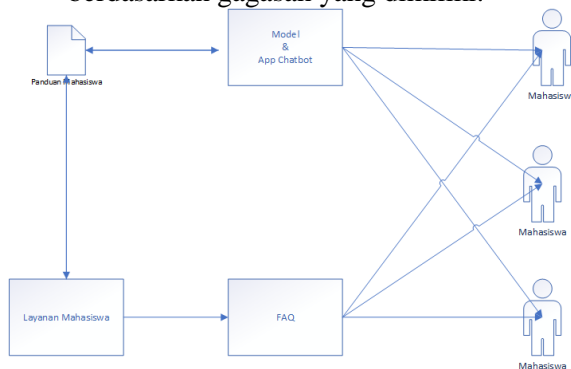
Penelitian ini dilakukan dengan tujuan untuk membuat model pembelajaran mesin pada aplikasi “chatbot” yang dapat menjawab pertanyaan pengguna seputar topik-topik yang ada di dalam buku panduan mahasiswa Kalbis Institute.

II. METODE PENELITIAN

Penelitian ini membahas mengenai pengembangan model pembelajaran mesin untuk aplikasi “chatbot” yang dapat digunakan sebagai alat pendukung fasilitas layanan mahasiswa dalam menjawab pertanyaan mahasiswa seputar topik-topik yang ada di dalam buku panduan mahasiswa. Aplikasi “chatbot” dapat dimanfaatkan untuk menjadi alat bantu bagi pengguna dalam mencari informasi. Aplikasi “chatbot” harus diberi pengetahuan mengenai informasi tertentu agar dapat menjawab pertanyaan pengguna dengan tepat.

Dengan gagasan tersebut, maka dibuat model pembelajaran mesin untuk

aplikasi "chatbot" yang dapat menjawab pertanyaan pengguna seputar topik-topik yang sering ditanyakan. Model dalam aplikasi "chatbot" ini diharapkan dapat memprediksi jawaban yang tepat sehingga dapat mendukung fasilitas layanan mahasiswa dan memudahkan pengguna untuk mencari informasi yang ada di dalam buku panduan mahasiswa. Bahasa pemrograman Python akan digunakan untuk membangun model dan aplikasi ini. Dibutuhkan juga perangkat lunak Anaconda yang berbasis *desktop* yang didalamnya terdapat *library Tensorflow, Natural Language Toolkit (NLTK), Sastrawi, dan Numpy*. Metode yang digunakan dalam pengembangan aplikasi adalah metode inkremental. Hasil akhir aplikasi ini berupa aplikasi "chatbot" yang ditampilkan dalam bentuk *Command Line Interface (CLI)*. Di bawah ini merupakan rancangan kerangka pemikiran yang dirancang berdasarkan gagasan yang dimiliki.

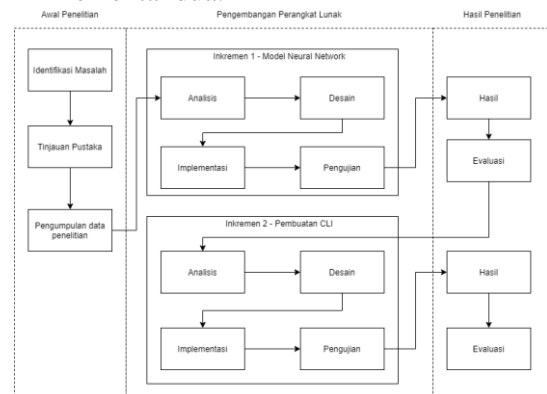


Gambar 1 Kerangka Pemikiran

Agar model pada aplikasi "chatbot" dapat memprediksi jawaban dari pertanyaan dengan tepat, diperlukan dataset yang berisikan jawaban beserta pertanyaannya untuk dilatih menggunakan algoritma *Deep Neural Network*. Dataset ini berfungsi sebagai pengetahuan dasar model aplikasi "chatbot" yang nantinya akan digunakan sebagai acuan dalam memprediksi jawaban. Dataset untuk aplikasi "chatbot" berasal dari buku panduan mahasiswa Kalbis Institut yang digunakan sebagai acuan untuk jawaban

dari pertanyaan-pertanyaan yang dibuat peneliti. Dataset yang telah dilatih menggunakan *deep learning* dan akan menghasilkan model. Model ini dapat digunakan untuk memprediksi jawaban dari pertanyaan pengguna agar aplikasi "chatbot" dapat memberi jawaban yang tepat. Pertanyaan yang dimasukkan oleh pengguna dilakukan pra pemrosesan terlebih dahulu sehingga menyisakan kata-kata pokok yang memungkinkan sistem untuk memberikan jawaban yang tepat kepada pengguna terkait pertanyaan yang diajukan.

Dalam pengembangan aplikasi, peneliti menggunakan metode inkremental. Terdapat dua tahapan dalam penelitian ini, yaitu inkremental satu dan inkremental dua.



Gambar 2 Proses Penelitian dengan Metode Inkremental

1. Inkremental Satu

Tahap inkremental satu dilakukan untuk melatih dataset, membentuk model, dan mendapatkan akurasi dari model yang terbentuk.

1.1. Analisis

Tahap awal penelitian adalah mempersiapkan dataset yang nantinya berguna sebagai pengetahuan dasar bagi aplikasi "chatbot". Dataset ini disusun manual dari buku panduan mahasiswa Kalbis Institute tahun 2019/2020 sebagai acuan dari jawaban-jawaban yang memungkinkan aplikasi "chatbot" untuk menjawab pertanyaan dengan tepat. Peneliti juga menambahkan pertanyaan-

pertanyaan yang berhubungan dengan topik dari jawaban yang ada secara manual karena tidak adanya data percakapan riil antara layanan mahasiswa dengan mahasiswa. Dataset yang dibuat peneliti terdiri dari 124 pertanyaan, 30 jawaban, yang dibagi menjadi 30 kelompok.

Pada tahap selanjutnya, peneliti menganalisis perangkat yang dibutuhkan untuk mendukung kinerja aplikasi. Perangkat keras yang digunakan adalah komputer dengan spesifikasi sebagai berikut:

1. Processor : Intel Core i7
2. RAM : 16 GB
3. Storage : 500 GB SSD
4. VGA : GeForce GTX 1660 Super

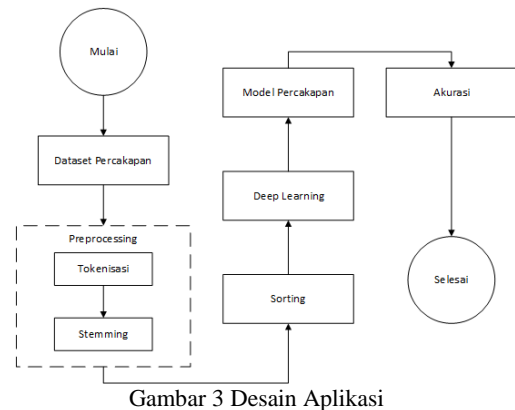
Perangkat lunak yang digunakan adalah sebagai berikut :

1. Windows 10 Pro 64 bit
2. Anaconda
3. Spyder
4. Python
5. Modul *Tensorflow*, *Natural Language Toolkit (NLTK)*, *Sastrawi*, dan *Numpy*

Proses selanjutnya adalah membuat model untuk aplikasi “chatbot” berdasarkan analisis dari metode-metode yang ada. Berdasarkan studi literatur yang ada, metode *Deep Learning* dipilih oleh peneliti untuk mendeteksi konteks yang ada didalam pertanyaan pengguna dan memilih jawaban yang tepat.

1.2. Desain

Proses ini dilakukan untuk membuat desain alur dari logika model. Di bawah ini merupakan tahapan yang akan dilakukan untuk membuat model dan aplikasi “chatbot”.



Gambar 3 Desain Aplikasi

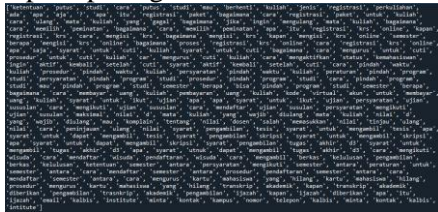
Langkah awal yang dilakukan adalah melakukan pra pemrosesan pada dataset yang sudah disiapkan. Tahap awal pra pemrosesan yang dilakukan adalah dengan tokenisasi. Tokenisasi dilakukan terlebih dahulu agar data teks dapat diolah lebih mudah. Tokenisasi mengubah setiap kalimat yang ada menjadi token kata. Selanjutnya adalah *stemming* yang berfungsi untuk mengubah kata berimbuhan menjadi kata dasar. Lalu dilakukan sorting agar kata-kata yang sudah dilakukan pra pemrosesan dapat lebih mudah diolah.

Data hasil pra pemrosesan kemudian akan diurutkan berdasarkan alfabet. Setelah diurutkan, data tersebut diubah bentuknya menjadi angka dengan menggunakan teknik *bag of word* agar dapat diolah ke dalam algoritma *neural network*. Struktur *neural network* yang digunakan terdiri dari 2 *hidden layer* dengan 8 *neuron* di setiap *hidden layer*. Selanjutnya data hasil pra pemrosesan akan diolah di dalam *neural network* yang telah disiapkan dan akan menghasilkan model. Model ini yang akan digunakan untuk memprediksi jawaban dari pertanyaan pengguna nanti.

1.3. Implementasi

Pada tahap inkremental satu, tahap awal dalam pengolahan teks adalah pra pemrosesan data teks. Dataset yang sudah disiapkan dilakukan pra pemrosesan berupa tokenisasi dan stemming. Tokenisasi akan mengubah

teks dalam dataset menjadi token kata seperti pada gambar di bawah ini.



Gambar 4 Hasil Tokenisasi

Proses selanjutnya adalah melakukan *stemming* pada data teks yang telah di-tokenisasi. Stemming akan mengubah seluruh kata yang ada menjadi kata dasar seperti pada gambar di bawah ini.



Gambar 5 Hasil Stemming

Selanjutnya data yang telah dilakukan pra pemrosesan akan diurutkan berdasarkan alfabet. Setelah itu data teks harus diubah bentuknya menjadi array menggunakan metode *bag of words*. Contoh implementasi *bag of words* dengan dokumen teks:

Teks 1 = “Budi suka bermain bola. Budi suka makan mie instan. Budi suka main bola dan makan mie instan”

Teks 2 = “Budi makan bola”

Data teks ini harus dilakukan tokenisasi untuk memisah tiap kata menjadi:

“Budi”, “suka”, “bermain”, “bola”, “Budi”, “suka”, “makan”, “mie”, “instan”, “Budi”, “suka”, “main”, “bola”, “dan”, “makan”, “mie”, “instan”

Selanjutnya list dari kata-kata tersebut akan direpresentasikan dalam bentuk angka dengan cara berikut:

BoW1 = { “Budi” : 3 , “suka” : 3 , “bermain” : 1 , “bola” : 2 , “makan” : 2 , “mie” : 2 , “instan” : 2 , “main” : 1 , “dan” : 1 };

BoW2 = { “Budi” : 1 , “suka” : 0 , “bermain” : 0 , “bola” : 1 , “makan” : 1 ,

“mie” : 0 , “instan” : 0 , “main” : 0 , “dan” : 0 };

Sehingga akan menghasilkan array seperti berikut:

BoW1 = [3,3,1,2,2,2,1,1]

BoW2 = [1,0,0,1,1,0,0,0,0]

Tahap selanjutnya adalah memasukkan data *bag of words* ke dalam neural network untuk menghasilkan model yang dapat digunakan dalam memprediksi jawaban dari pertanyaan pengguna.

1.4. Pengujian

Berikut ini adalah proses-proses yang diuji setelah model terbentuk. Semua proses pada inkremental satu akan dijabarkan pada tabel 1.

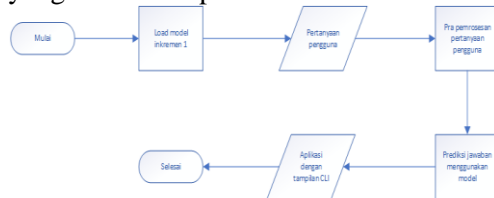
Tabel 1 Skenario Pengujian Inkremental Satu

No	Proses	Data Masukan	Proses	Pengamatan
1	Pengambilan dataset	File json dataset	Melakukan pengambilan dataset berupa teks dari file json	Dapat mengambil data teks dari file json
2	Tokenisasi	Data teks	Melakukan pemotongan kalimat menjadi token kata	Dapat menghasilkan token kata dari kalimat
3	Stemming	Data teks yang telah dilakukan tokenisasi	Melakukan perubahan kata yang berimbuhan menjadi kata dasar	Dapat mengubah kata berimbuhan menjadi kata dasar
4	Sorting	Data teks yang telah dilakukan tokenisasi dan stemming	Melakukan pengurutan kata berdasarkan alfabet	Dapat mengurutkan kata berdasarkan alfabet
5	Mengubah bentuk teks menjadi array	Data teks yang telah di sorting	Mendapatkan hasil konversi dari teks yang telah disorting	Dapat mengubah teks menjadi array
6	Training menggunakan neural network	Data teks yang bentuknya sudah berubah menjadi array	Mendapatkan hasil akurasi diatas 80%	Dapat menghasilkan akurasi training sebesar 99%

Pengujian ini dilakukan untuk memastikan proses-proses pada inkremental satu dapat berjalan dengan baik. Proses-proses pada tabel 1 bertujuan untuk membuat model dengan akurasi yang baik agar dapat mendeteksi pertanyaan pengguna dan memprediksi jawaban yang tepat atas jawaban yang diberikan.

2. Inkremental Dua

Tahap inkremental dua dilakukan untuk menambahkan *command line interface* (CLI) aplikasi “chatbot”. Tampilan CLI akan disesuaikan dengan kebutuhan tampilan yang dievaluasi pada inkremen satu.



Gambar 6 Inkremental Dua

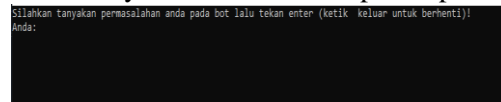
Pada gambar 6 diatas dijelaskan proses-proses yang dilakukan pada inkremental dua. Proses ini merupakan proses lanjutan dari inkremen satu dimana pada inkremental dua hanya menambahkan fungsi untuk model memprediksi jawaban dari pertanyaan pengguna dan menampilkannya dalam tampilan CLI.

2.1. Analisis

Karena fokus tujuan utama aplikasi ini adalah untuk mengetes kemampuan model dalam memprediksi dan menjawab pertanyaan pengguna, maka harus ada fungsi untuk dapat memprediksi jawaban atas pertanyaan pengguna dan merespon pertanyaan pengguna yang dimasukkan. Jadi, peneliti akan membuat fungsi untuk mengirim pertanyaan menggunakan tombol “enter” pada keyboard dan fungsi untuk memprediksi jawaban dari pertanyaan pengguna.

2.2. Desain

Tahap ini dilakukan dengan membuat atribut aplikasi pada tahap sebelumnya ke dalam bentuk prototipe.



Gambar 7 CLI

Pada gambar 7 merupakan tampilan CLI aplikasi “chatbot”. Pengguna dapat mengisi pertanyaan di kolom “Anda:”, setelah itu pengguna dapat menekan tombol enter pada keyboard. Jawaban dari pertanyaan pengguna akan ditampilkan dibawahnya. Untuk keluar dari aplikasi, pengguna mengetik “keluar” lalu tekan enter pada keyboard, maka aplikasi “chatbot” akan berhenti.

2.3. Implementasi

Tahap ini dilakukan untuk merepresentasikan desain dari tahap sebelumnya ke dalam kode python. Teks pertanyaan dari pengguna akan dilakukan pra pemrosesan terlebih dahulu lalu mengubah bentuk teks menjadi array menggunakan metode *bag of words*. Di bawah ini merupakan kode program untuk melakukan pra pemrosesan sekaligus implementasi metode *bag of words* pada teks pertanyaan pengguna.

Selanjutnya adalah pembuatan fungsi yang berguna untuk membuat tampilan pada CLI serta memprediksi jawaban dari pertanyaan pengguna dan menampilkan hasil prediksi tersebut.

2.4. Pengujian

Pengujian dilakukan pada seluruh fungsi yang terdapat di dalam CLI untuk memastikan seluruh fungsi dapat berjalan sesuai yang diharapkan. Pengujian ini juga dilakukan untuk mengevaluasi model yang terbentuk dapat dijalankan dan memberi respon yang tepat terhadap pertanyaan pengguna.

Tabel 2 Skenario Pengujian Inkremental Dua

No	Fungsi	Proses
1	Judul	Menampilkan kalimat judul/pembuka pada aplikasi
2	Pertanyaan	Memasukkan pertanyaan pada kolom "Anda:?"
3	Jawaban	Merespon pertanyaan pengguna dengan memberikan jawaban berupa teks
4	Keluar	Memberhentikan aplikasi
5	Model pembelajaran mesin	Memprediksi jawaban dengan tepat

```

Adam | epoch: 999 | loss: 0.14587 - acc: 0.9732 -- iter: 048/124
Training Step: 7988 | total loss: 0.13263 | time: 0.005s
Adam | epoch: 999 | loss: 0.13263 - acc: 0.9759 -- iter: 064/124
Training Step: 7989 | total loss: 0.12034 | time: 0.007s
Adam | epoch: 999 | loss: 0.12034 - acc: 0.9783 -- iter: 080/124
Training Step: 7990 | total loss: 0.10901 | time: 0.008s
Adam | epoch: 999 | loss: 0.10901 - acc: 0.9805 -- iter: 096/124
Training Step: 7991 | total loss: 0.10350 | time: 0.010s
Adam | epoch: 999 | loss: 0.10350 - acc: 0.9762 -- iter: 112/124
Training Step: 7992 | total loss: 0.09427 | time: 0.012s
Adam | epoch: 999 | loss: 0.09427 - acc: 0.9786 -- iter: 124/124
-
Training Step: 7993 | total loss: 0.08597 | time: 0.001s
Adam | epoch: 1000 | loss: 0.08597 - acc: 0.9807 -- iter: 016/124
Training Step: 7994 | total loss: 0.08256 | time: 0.002s
Adam | epoch: 1000 | loss: 0.08256 - acc: 0.9764 -- iter: 032/124
Training Step: 7995 | total loss: 0.07534 | time: 0.004s
Adam | epoch: 1000 | loss: 0.07534 - acc: 0.9787 -- iter: 048/124
Training Step: 7996 | total loss: 0.06872 | time: 0.006s
Adam | epoch: 1000 | loss: 0.06872 - acc: 0.9809 -- iter: 064/124
Training Step: 7997 | total loss: 0.06297 | time: 0.008s
Adam | epoch: 1000 | loss: 0.06297 - acc: 0.9828 -- iter: 080/124
Training Step: 7998 | total loss: 0.05782 | time: 0.009s
Adam | epoch: 1000 | loss: 0.05782 - acc: 0.9845 -- iter: 096/124
Training Step: 7999 | total loss: 0.05298 | time: 0.011s
Adam | epoch: 1000 | loss: 0.05298 - acc: 0.9861 -- iter: 112/124
Training Step: 8000 | total loss: 0.05275 | time: 0.012s
Adam | epoch: 1000 | loss: 0.05275 - acc: 0.9812 -- iter: 124/124
    
```

Gambar 8 Hasil Training

III. HASIL DAN PEMBAHASAN

Pada penelitian ini, dataset disimpan dalam bentuk file JSON seperti pada gambar 8. Dataset ini akan diproses menggunakan metode *Deep Learning* yang nantinya akan menghasilkan model pembelajaran mesin.

```

{"context": {
  "tag": "pembukaan",
  "prompt": ["Pembukaan program studi", "cara pembuka studi", "apa beasiswa kuliah?"],
  "response": ["Beasiswa diberikan untuk studi, jika ada. Tidak melibatkan registrasi perkuliahan selama 3 (tiga) semester."],
},
  "tag": "registrasi",
  "prompt": ["Mau registrasi perkuliahan ada apa aja?"],
  "response": ["Registrasi perkuliahan ada 2, yaitu: 1. Registrasi secara paket 1/2. Registrasi secara RPS online"],
},
  "tag": "registrasi paket",
  "prompt": ["Papa itu registrasi paket", "bagaimana cara registrasi paket untuk kuliah?"],
  "response": ["Registrasi secara paket ini. Didaftarkan secara otomatis bagi mahasiswa yang sudah memenuhi persyaratan"],
},
  "tag": "alasan tidak",
  "prompt": ["Cara ulang mata kuliah yang gagal bagaimana", "jika ingin mengulang mata kuliah bagaimana?"],
  "response": ["Mata kuliah yang akan diulang, dapat diambil pada saat Semester Anterlainatau saat mahasiswa melaku"],
},
}
    
```

Gambar 8 Contoh Dataset

Berikut ini adalah hasil dari pembuatan model. Gambar 9 adalah proses pembuatan model dengan melatih dataset. Training dilakukan sesuai dengan jumlah *epoch* yang sudah ditentukan. Peneliti menentukan jumlah *epoch* sebanyak 1000 kali karena pada *epoch* 1000 menghasilkan akurasi sebesar 98% dan nilai *loss* yang kecil.

Pada gambar 9 di bawah ini merupakan tampilan CLI aplikasi "chatbot". Pengguna dapat langsung mengisi pertanyaan yang nanti akan muncul di sebelah tulisan "Anda:?". Setelah pengguna mengajukan pertanyaan, pengguna dapat menekan tombol enter pada keyboard untuk mendapatkan respon berupa jawaban mengenai pertanyaan yang diajukan.



Gambar 9 Tampilan CLI

Pada gambar 10 di bawah ini merupakan tampilan setelah pengguna memasukkan pertanyaan. Pertanyaan tersebut akan langsung dideteksi dan sistem akan memprediksi jawaban lalu menampilkan jawaban yang sesuai dengan topik pertanyaan yang diajukan berdasarkan presentase prediksi tertinggi.



Gambar 10 Tes Aplikasi “chatbot”

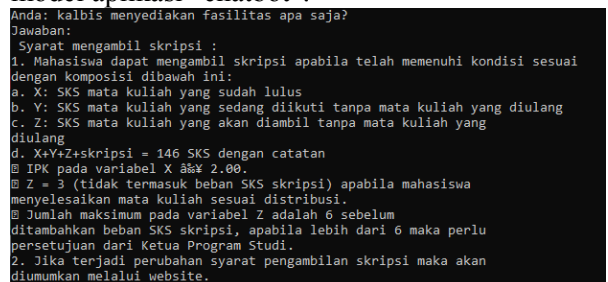
Pada tabel 3 di bawah ini, peneliti melakukan pengujian kepada model aplikasi “chatbot” dalam menjawab pertanyaan riil yang diajukan. Pertanyaan-pertanyaan yang diajukan adalah pertanyaan yang tidak ada di dalam *dataset* tetapi dengan topik pertanyaan yang ada di dalam *dataset*. Hal ini dilakukan untuk memastikan bahwa model dapat memprediksi jawaban dengan tepat. Pertanyaan-pertanyaan yang berhasil dijawab dengan benar akan diberikan hasil “Valid”.

Tabel 3 Pengujian Respon pada Model

No	Pertanyaan	Hasil
1	Ktm saya hilang, cara membuat baru gimana?	VALID
2	Nilai d maksimal berapa?	VALID
3	Kapan pengisian krs online?	VALID
4	Bagaimana prosedur untuk cuti?	VALID
5	Bagaimana cara pindah waktu kuliah?	VALID
6	Semester berapa mengisi krs online?	VALID
7	Ingin pindah program studi, bagaimana caranya?	VALID
8	Saya ingin memperbaiki nilai saya karena dosen saya salah memasukkan nilai, bagaimana caranya?	VALID
9	Syarat mengikuti semester antara apa saja?	VALID
10	Syarat untuk mengambil skripsi apa?	VALID
11	Bagaimana cara memilih peminatan?	VALID
12	Cara mengulang matkul yang mendapat nilai d bagaimana?	VALID

13	Bagaimana cara membayar uang kuliah?	VALID
14	Syarat untuk dapat cuti apa saja?	VALID
15	Kapan mengambil berkas kelulusan?	VALID
16	Apa saja syarat mengejakan tesis?	VALID
17	Cara mengikuti ujian susulan bagaimana?	VALID
18	Saya ingin aktif kembali setelah cuti, bagaimana caranya?	VALID
19	Bagaimana cara mendaftar wisuda?	VALID
20	Pembayaran uang kuliah kemana?	VALID

Jika topik pertanyaan yang diajukan tidak terdapat di dalam *dataset*, maka model akan memilih jawaban berdasarkan presentase prediksi tertinggi dan ditampilkan oleh aplikasi “chatbot” ini. Gambar 11 di bawah ini merupakan contoh dari kurangnya pengetahuan pada model aplikasi “chatbot”.



Gambar 11 Kurangnya Pengetahuan Model

IV. SIMPULAN

Kesimpulan dari hasil analisis dan uji coba yang dilakukan, maka dapat diambil kesimpulan sebagai berikut:

- Model yang dibuat menghasilkan nilai akurasi sebesar 98% dan *loss value* sebesar 0.05275 dengan jumlah training yang dilakukan sebanyak 1000 kali.
- Model diterapkan dalam aplikasi “chatbot” yang menggunakan bahasa Indonesia dan tampilan CLI.
- Model pada aplikasi “chatbot” dapat memprediksi pertanyaan seputar hal-hal yang ada dalam buku panduan mahasiswa dengan tepat dengan jumlah

pertanyaan yang diajukan sebanyak 20 pertanyaan yang berbeda-beda.

d. Kurangnya pengetahuan pada dataset menyebabkan aplikasi "chatbot" tidak dapat menjawab pertanyaan dengan tepat.

DAFTAR RUJUKAN

- [1] M. N. Fauzy dan Kusriani, "Chatbot menggunakan metode fuzzy string matching sebagai virtual assistant pada pusat layanan informasi akademik," *J. Inf. Politek. Indonusa Surakarta*, vol. 5, hlm. 61–67, 2019.
- [2] J. Ghorpade-Aher, R. Kontamwar, S. Kukreja, T. Karpe, dan S. Kakkad, "An overview of NLP based Chatbots," *Univers. Rev.*, vol. VIII, no. II, hlm. 177–181, 2019.
- [3] A. Elcholiqi dan A. Musdholifah, "Chatbot in Bahasa Indonesia using NLP to Provide Banking Information," *IJCCS (Indonesian J. Comput. Cybern. Syst.)*, vol. 14, no. 1, hlm. 91, Jan 2020, doi: 10.22146/ijccs.41289.
- [4] T. A. Zuraiyah, D. K. Utami, dan D. Herlambang, "IMPLEMENTASI CHATBOT PADA PENDAFTARAN MAHASISWA BARU MENGGUNAKAN RECURRENT NEURAL NETWORK," *J. Ilm. Teknol. dan Rekayasa*, vol. 24, no. 2, hlm. 91–101, Apr 2019, doi: 10.35760/tr.2019.v24i2.2388.
- [5] R. Khan dan A. Das, *Build Better Chatbots*. Berkeley, CA: Apress, 2018.
- [6] T. Beysolow II, *Applied Natural Language Processing with Python*. Berkeley, CA: Apress, 2018.
- [7] L. Deng dan Y. Liu, "Epilogue: Frontiers of NLP in the Deep Learning Era," in *Deep Learning in Natural Language Processing*, Singapore: Springer Singapore, 2018, hlm. 309–326.
- [8] Lane. Hobson dan H. Hanes, *Natural Language Processing in Action: Understanding, analyzing, and generating text with Python*. Manning Publications, 2019.
- [9] A. C. Muller dan S. Guido, *Introduction to Machine Learning with Python*. 2017.
- [10] M. Taylor, *Neural Networks: A Visual Introduction for Beginners*. 2017.
- [11] F. Chollet, *Deep Learning with Python*. Manning Publications, 2018.