

Part of Speech Tagging Pada Teks Bahasa Indonesia dengan BiLSTM + CNN + CRF dan ELMo

Muhamad Kurniawan¹, Kusrini Kusrini², Muhammad Rudyanto Arief³

Program Studi Magister Teknik Informatika

Universitas Amikom Yogyakarta

Yogyakarta, Indonesia

e-mail: ¹muh.wan@hotmail.com, ²kusrini@amikom.ac.id, ³rudy@amikom.ac.id

Diajukan: 8 Februari 2021; Direvisi: 30 Juni 2021; Diterima: 4 Agustus 2021

Abstrak

Part of Speech Tagging atau POS Tag merupakan salah satu proses untuk mengelompokkan kata berdasarkan kelas kata seperti: kata benda, kata kerja, atau kata sambung. Kegunaan POS Tag antara lain dapat bermanfaat pada analisis sentimen, pengenalan entitas bernama, dan konversi teks ke suara. Dalam melakukan POS Tag, jika dilakukan secara manual dapat menghabiskan banyak waktu, oleh karena itu dibuatlah sistem berbasis machine learning untuk mengotomatisasi proses ini. Pada penelitian ini, dilakukan POS Tag dengan menerapkan transfer learning dengan model Embedding from Language Model (ELMo). Model ELMo cukup populer digunakan pada dataset bahasa Inggris karena dapat memberi hasil akurasi yang memuaskan namun pada dataset bahasa Indonesia belum ada paper yang membahas tentang model ini. Melalui penelitian ini ingin dilihat bagaimana performa ELMo pada dataset bahasa Indonesia. Model yang digunakan untuk menyelesaikan masalah POS Tag adalah model berbasis BiLSTM. Pada penelitian ini, juga ingin dilihat bagaimana performa model jika ditambahkan CNN setelah BiLSTM. Selain itu, juga diteliti bagaimana performa dari tiap lapisan ELMo. Dari penelitian ini diperoleh bahwa metode BiLSTM + CNN + CRF dengan embedding ELMo lapisan pertama memiliki akurasi terbaik, dengan nilai 95.62%. Selain itu, diperoleh bahwa penambahan CNN setelah BiLSTM dapat meningkatkan akurasi serta mengurangi overfitting pada masalah POS Tag bahasa Indonesia.

Kata kunci: Data mining, POS Tag, Deep learning, Pengolahan bahasa alami, Embedding.

Abstract

Part of Speech Tagging or POS Tag is a process for grouping words based on word classes such as: nouns, verbs, or conjunctions. The use of POS Tags, among others, can be useful in sentiment analysis, recognition of named entities, and conversion of text to speech. In doing POS Tag, if it is done manually, it can take a lot of time, therefore a machine learning based system is created to automate this process. In this study, POS Tag was carried out by implementing transfer learning with the Embedding from Language Model (ELMo) model. The ELMo model is quite popular for use on the English dataset because it can provide satisfactory accuracy results, but in the Indonesian language dataset, there is no paper that discusses this model. Through this research, we want to see how the performance of ELMo on the Indonesian language dataset. The model used to solve the POS Tag problem is a BiLSTM-based model. In this study, we also want to see how the model performs if CNN is added after BiLSTM. In addition, it is also investigated how the performance of each ELMo layer is. From this research, it was found that the BiLSTM + CNN + CRF method with the first layer of ELMo embedding has the best accuracy, with a value of 95.62%. In addition, it was found that the addition of CNN after BiLSTM can improve accuracy and reduce overfitting in Indonesian POS Tag problems.

Keywords: Data mining, POS Tag, Deep learning, Natural language processing, Embedding.

1. Pendahuluan

Pengolahan bahasa alami adalah proses untuk memperoleh pengetahuan dari data berupa teks. Data tersebut dapat diperoleh dari berbagai sumber seperti sosial media, blog, review produk, dan video. Terdapat beberapa jenis pengolahan bahasa alami, contohnya analisis sentimen, tanya-jawab, *named entity recognition*, klasifikasi teks, dan POS Tag [1].

Pada POS Tag, metode dilakukan dengan mengelompokkan kata berdasarkan kelas kata, misalnya: kata benda, kata kerja, atau kata sambung. POS Tag digunakan untuk analisis teks dan dapat membantu dalam beberapa masalah pengolahan bahasa alami. Proses POS Tag biasanya dilakukan secara otomatis, dengan menggunakan model pembelajaran mesin yang telah dilatih terlebih dahulu. Terdapat beberapa metode yang biasanya digunakan untuk POS Tag, contohnya Hidden Markov, Viterbi, Brill Tagger, dan metode yang saat ini cukup populer *deep learning* [2], [3].

Terdapat beberapa penelitian terdahulu tentang POS Tag. Munarko dan Azhar melakukan penelitian POS Tag pada *tweet* bahasa Indonesia. Model yang digunakan adalah *Cyclic Dependency Network* [4]. *Dataset* yang digunakan diambil dari Twitter. Kemudian, pada penelitian ini ingin dilihat bagaimana performa model yang telah dilatih pada *dataset* dengan bahasa formal dan informal. Diperoleh akurasi pada data formal sebesar 95,42%, data informal sebesar 92,42%, dan data gabungan sebesar 90,69%.

Selain itu, Sabloak juga melakukan penelitian tentang POS Tag bahasa Indonesia [5]. Metode yang digunakan adalah Viterbi [3]. Sedangkan *dataset* yang digunakan adalah IDN Tagged Corpus. Diperoleh akurasi rata-rata sebesar 93,23%. Pada penelitian yang penulis lakukan, *dataset* yang digunakan juga IDN Tagged Corpus, sedangkan metode yang digunakan berbasis *deep learning*.

Metode *deep learning* telah lama ditemukan dan telah banyak pengembangan arsitektur dari metode ini. Contoh metode *deep learning* adalah *Convolution Neural Network* (CNN) yang biasa digunakan untuk klasifikasi gambar kemudian *Deep Belief Network* yang digunakan pada penemuan obat baru, setelah itu *Long Short Term Memory* (LSTM) yang biasa digunakan pada pengolahan bahasa alami [6], [7].

Pada penelitian ini, penulis ingin melakukan penelitian POS Tag dengan metode berbasis LSTM. Terdapat penelitian sebelumnya mengenai POS Tag dengan metode berbasis LSTM. Kurniawan dan Aji telah melakukan penelitian mengenai POS Tag dengan metode LSTM [8]. *Dataset* yang digunakan adalah IDN Tagged Corpus. Metode yang digunakan adalah BiLSTM-CRF dengan *embedding* berbasis CNN. Diperoleh akurasi terbaik sebesar 97,47%.

Selain itu, penulis akan menggunakan CRF pada keluaran model karena pada penelitian sebelumnya, penggunaan CRF dapat menghasilkan akurasi yang lebih baik dibandingkan menggunakan Softmax. Ma dan Hovy telah melakukan penelitian tentang POS Tag dengan model LSTM+CNN+CRF [9]. Kemudian *dataset* yang digunakan adalah Penn Treebank WSJ yang merupakan *dataset* bahasa Inggris. Diperoleh bahwa model LSTM+CNN+CRF dengan *Embedding Glove* memiliki akurasi terbaik, yaitu 97,55%.

Seperti penelitian Ma dan Hovy, pada penelitian ini penulis juga akan meneliti penggunaan model CNN [9]. Cukup banyak penelitian POS Tag yang menggunakan metode BiLSTM+CRF, namun pada penelitian ini penulis menambahkan model CNN setelah BiLSTM untuk melihat pengaruhnya terhadap performa model. Belum ada penelitian mengenai penggunaan CNN setelah BiLSTM pada *dataset* POS Tag bahasa Indonesia, sehingga penulis tertarik untuk melakukan penelitian ini.

Selain itu, penulis akan menggunakan *embedding* ELMo (*Embedding from Language Model*) dari penelitian [10], [11]. Mereka telah melatih *embedding* ELMo pada berbagai bahasa, salah satunya Indonesia. Penelitian yang menggunakan ELMo untuk pengolahan bahasa alami pada teks bahasa Indonesia belum ada sehingga penulis ingin melihat performa *embedding* tersebut dalam masalah POS Tag bahasa Indonesia. ELMo memiliki performa yang sangat baik dalam menyelesaikan masalah pengolahan bahasa alami pada *dataset* bahasa Inggris, sehingga penulis juga berharap, ELMo dapat memberikan performa yang sangat baik pada *dataset* bahasa Indonesia. Pada penelitian ini, penulis menggunakan bahasa pemrograman Python dan Jupyter Notebook yang disediakan oleh kaggle.com.

2. Metode Penelitian

2.1. IDN Tagged Corpus Dataset

Dataset yang digunakan adalah IDN Tagged Corpus yang merupakan hasil penelitian dari Arawinda [12]. *Dataset* ini berisi kumpulan kalimat sebanyak 10.030 yang telah dilabeli POS Tag. Terdapat 23 label POS Tag pada *dataset* ini yang dapat di lihat pada Tabel 1.

Tabel 1. POS Tag pada *dataset* IDN Tagged Corpus.

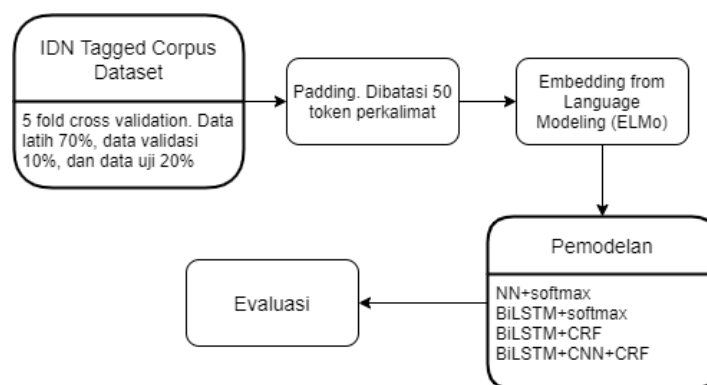
Tag	Deskripsi	Contoh
CC	<i>Coordinating conjunction</i>	dan, tetapi, atau
CD	<i>Cardinal number</i>	dua, juta, enam, 7916, sepertiga, 0,025, 0,525, banyak, kedua, ribuan, 2007, 25
OD	<i>Ordinal number</i>	ketiga, ke-4, pertama
DT	<i>Determiner / article</i>	Para, Sang, Si
FW	<i>Foreign word</i>	<i>climate change, terms and conditions</i>

Tag	Deskripsi	Contoh
IN	<i>Preposition</i>	dalam, dengan, di, ke, oleh, pada, untuk
JJ	<i>Adjective</i>	bersih, panjang, hitam, lama, jauh, marah, suram, nasional, bulat
MD	<i>Modal and auxiliary verb</i>	boleh, harus, sudah, mesti, perlu
NEG	<i>Negation</i>	tidak, belum, jangan
NN	<i>Noun</i>	monyet, bawah, sekarang, rupiah
NNP	<i>Proper noun</i>	Boediono, Laut Jawa, Indonesia, India, Malaysia, Bank Mandiri, BBKP, Januari, Senin, Idul Fitri, Piala Dunia, Liga Primer, Lord of the Rings: The Return of the King
NND	<i>Classifier, partitive, and measurement noun</i>	orang, ton, helai, lembar
PR	<i>Demonstrative pronoun</i>	ini, itu, sini, situ
PRP	<i>Personal pronoun</i>	saya, kami, kita, kamu, kalian, dia, mereka
RB	<i>Adverb</i>	sangat, hanya, justru, niscaya, segera
RP	<i>Particle</i>	pun, -lah, -kah
SC	<i>Subordinating conjunction</i>	sejak, jika, seandainya, supaya, meski, seolah-olah, sebab, maka, tanpa, dengan, bahwa, yang, lebih ... daripada ..., semoga
SYM	<i>Symbol</i>	IDR, +, %, @
UH	<i>Interjection</i>	brengsek, oh, ooh, aduh, ayo, mari, hai
VB	<i>Verb</i>	merancang, mengatur, pergi, bekerja, tertidur
WH	<i>Question</i>	siapa, apa, mana, kenapa, kapan, di mana, bagaimana, berapa
X	<i>Unknown</i>	statemen
Z	<i>Punctuation</i>	"...", ?, .

Dalam penelitian ini, penulis menggunakan *dataset* yang telah dibuat menjadi 5 set data dengan metode *5 fold cross validation*. *Dataset* ini merupakan hasil dari penelitian [8] dan tersedia secara publik. *Dataset* ini dipisah menjadi data latih sebanyak 70%, data *development* sebanyak 10% dan data uji sebanyak 20%.

2.2. Desain Penelitian

Dalam penelitian ini penulis menggunakan *dataset* IDN Tagged Corpus dengan perlakuan seperti yang telah dijelaskan pada bagian 2.1. Sebelum dilatih, teks ditokenisasi kemudian tiap kalimat dibatasi sebanyak 50 *token*. Jika suatu kalimat memiliki kurang dari 50 *token*, maka akan diberi *padding*, jika lebih dari 50 maka kalimat 51 dan seterusnya akan dipotong. Kemudian, penulis menggunakan model *pretrained Embeddings from Language Models (ELMo)* yang berasal dari penelitian [10]. Setelah itu, data dilatih dengan model berbasis LSTM, selain itu penulis juga menambah model CNN untuk melihat bagaimana performanya. Pada atas model juga terdapat 2 model yang ingin dibandingkan, yaitu Softmax dan CRF. Penjelasan mengenai metode yang penulis gunakan dijelaskan pada bagian selanjutnya.



Gambar 1. Alur penelitian.

2.3. Preprocessing

Pada *dataset* IDN Tagged Corpus, jumlah kata pada tiap kalimat berbeda-beda sedangkan untuk dimasukkan ke model *deep learning* diperlukan dimensi atau panjang kalimat yang sama. Penulis membuat semua kalimat memiliki panjang 50 kata atau *token*. Untuk kalimat yang memiliki panjang kurang dari 50, penulis memberikan *padding* setelah kalimat, sedangkan kalimat yang lebih dari 50 kata, penulis memotong pada kata ke 51 dan seterusnya. Pemilihan 50 kata karena jika lebih dari 50 kata, penggunaan RAM dapat melebihi kapasitas, sehingga proses pelatihan tidak dapat dijalankan.

Pada penelitian ini penulis tidak melakukan *preprocessing data* seperti *stemming* atau *lemmatization*. Selain itu, penulis tidak mengubah semua kata menjadi huruf kecil karena informasi seperti huruf kapital dapat berguna untuk proses klasifikasi POS Tag.

2.4. Embedding from Language Models

Dalam pengolahan bahasa alami, data yang diolah berupa teks. Teks dapat berupa kumpulan kata, namun cukup sulit untuk mengolah data tersebut karena komputer tidak dapat memahami kata seperti manusia. Oleh karena itu, dibuatlah suatu teknik sehingga kata dapat diolah dan dimasukkan dalam model pengolahan bahasa alami. Teknik yang cukup populer dalam mengubah kata menjadi bahasa yang siap diolah komputer adalah *embedding*.

Embedding merupakan teknik untuk merepresentasikan suatu kata dalam bentuk vektor. Kelebihan *embedding* adalah, pada tiap kata yang mirip akan memiliki vektor yang berdekatan. Misalnya kata “pergi” dan “berangkat” akan memiliki vektor yang mirip karena kedua kata tersebut memiliki makna yang sama. Beberapa contoh teknik *embedding* adalah Word2vec, Glove, Fasttext, BERT, dan ELMo.

Embeddings from Language Models atau ELMo merupakan metode *embedding* yang dibuat oleh Allen Institute. Tidak seperti Word2Vec atau Glove yang memiliki representasi kata yang tetap pada berbagai kalimat, metode ini akan memperhatikan keseluruhan kata dalam satu kalimat sebelum menentukan nilai representasi kata. Metode ini dapat mengatasi masalah polisemi, di mana suatu kata memiliki makna lebih dari satu. Contoh polisemi dapat dilihat pada kalimat di bawah ini:

- i. Gina adalah anak pertama dari Pak Juna.
- ii. Gina hampir terpeleset dari anak tangga.

Pada kedua kalimat di atas terdapat kata “anak” namun memiliki dua pengertian yang berbeda. Pada kalimat pertama anak memiliki arti benar-benar anak sedangkan kalimat kedua anak berarti salah satu bagian dari tangga.

ELMo menggunakan *bidirectional LSTM* dalam mempelajari kata dan konteks pada dokumen. Dengan penggunaan *bidirectional LSTM*, ELMo tidak hanya mengerti kata selanjutnya, namun juga kata sebelumnya. ELMo menggunakan model CNN pada lapisan pertama kemudian pada lapisan selanjutnya menggunakan 2 lapisan *bidirectional LSTM*. Selain itu, model ini menggunakan *Language Modeling* dalam melatih *embedding*-nya [8].

Pada penelitian ini, penulis akan menggunakan *embedding* ELMo (*Embedding from Language Model*) dari penelitian [11], [13]. Mereka telah melatih *embedding* ELMo pada berbagai bahasa, salah satunya Indonesia. Untuk ELMo bahasa Indonesia, *embedding* dilatih pada *dataset* Indonesian Conll 2017.

2.5. Perancangan Model

Seperti yang dijelaskan pada awal metode penelitian, pada penelitian ini penulis menggunakan model *pretrained* ELMo yang berasal dari penelitian [11], [13]. Penulis juga mencoba membandingkan beberapa model klasifikasi seperti CNN dan BiLSTM. Pada atas model juga terdapat 2 model yang ingin dibandingkan, yaitu Softmax dan CRF. Semua proses pengolahan data ini menggunakan Python Notebook dari Kaggle. Arsitektur model yang digunakan pada penelitian ini dapat dilihat pada Tabel 2.

Tabel 2. Arsitektur model *deep learning* pada penelitian ini.

NN	BiLSTM+softmax	BiLSTM+CRF	BiLSTM+CNN+CRF
Dense(Unit=64)	Bidirectional LSTM(Unit=256)	Bidirectional LSTM(Unit=256)	Bidirectional LSTM(Unit=256)
Dense(Aktivasi=Softmax)	Dropout(0.5)	Dropout(0.5)	Conv1D(Filter =16)
	Dense(Aktivasi=Softmax)	CRF	Conv1D(Filter =4)
			Dropout(0.5)
			CRF

2.5.1. Long Short Term Memory

Long Short Term Memory atau LSTM merupakan salah satu tipe jaringan syaraf tiruan yang biasa digunakan pada data yang bersifat sekuensial, seperti video, teks, dan data *time-series*. Dalam LSTM terdapat 3 bagian, yaitu *input gate*, *output gate*, dan *forget gate*. Rumus dari ketiga bagian tersebut dapat diekspresikan sebagai berikut [10]:

$$f_t = \sigma(W_f x_t + U_f h(t-1) + b_f) \tag{1}$$

$$i_t = \sigma(W_i x_t + U_i h(t-1) + b_i) \tag{2}$$

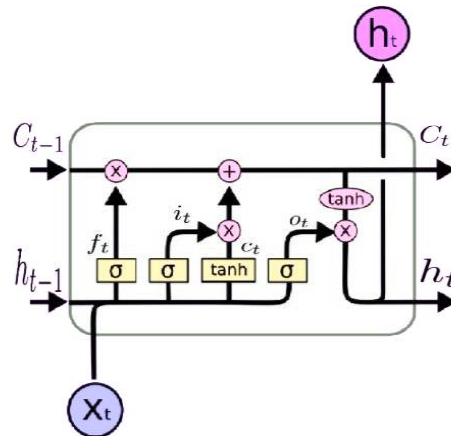
$$C_t = \tanh(W_c x_t + U_c h_{(t-1)} + b_c) \tag{3}$$

$$o_t = \sigma(W_o x_t + U_o h_{(t-1)} + b_o) \tag{4}$$

$$h_t = o_t * \tanh(C_t) \tag{5}$$

Dengan f_t adalah *forget gate*, i_t adalah *input gate*, C_t merupakan *cell state*, kemudian o_t adalah *output gate*, dan h_t adalah keluaran final.

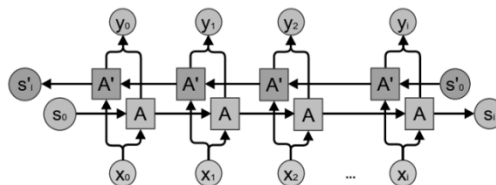
LSTM sangat baik dalam menyelesaikan berbagai masalah pengolahan bahasa alami, terutama masalah yang perlu memperhatikan konteks kalimat. Hal ini disebabkan LSTM memiliki memori sehingga dapat memperoleh informasi sebelumnya dan memperhitungkan keseluruhan kalimat. Kelebihan LSTM dari model sebelumnya yaitu RNN, LSTM dapat menyimpan informasi dalam waktu yang cukup lama.



Gambar 2. Arsitektur Long Short Term Memory.

2.5.2. Bidirectional Long Short Term Memory

Bidirectional Long Short Term Memory atau BiLSTM merupakan salah satu tipe jaringan syaraf tiruan yang merupakan variasi dari LSTM di mana data diproses tidak hanya alur maju (dari masa lalu ke masa depan) namun juga alur mundur (dari masa depan ke masa lalu) [11]. Dengan cara tersebut, model ini dapat melihat konteks kalimat dengan lebih baik.



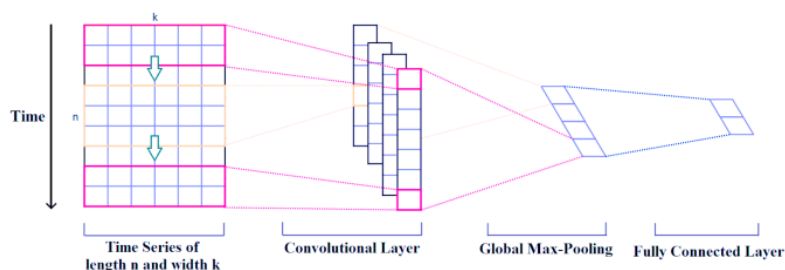
Gambar 3. Arsitektur BiLSTM.

Gambar 2 memperlihatkan arsitektur BiLSTM, di mana x adalah *input*, kemudian A adalah model LSTM dengan alur maju, sedangkan A' adalah LSTM dengan alur mundur. Hasil dari A dan A' kemudian diperhitungkan untuk menghasilkan *output* y .

2.5.3. Convolutional Neural Network

Convolutional Neural Network atau biasa disingkat CNN merupakan salah satu tipe jaringan syaraf tiruan yang memiliki lapisan konvolusi untuk melakukan ekstraksi fitur, di mana proses ini bermanfaat untuk memperoleh informasi secara maksimal. Walaupun populer digunakan pada pemrosesan citra, CNN juga biasa digunakan pada pengolahan bahasa alami. CNN yang biasa digunakan pada pengolahan bahasa alami adalah CNN 1 dimensi [9]. Arsitektur CNN dapat dilihat pada Gambar 1.

Pada CNN 1D ini, *kernel* konvolusi atau filter akan bergerak dalam 1 dimensi, di mana sumbu tersebut dapat berupa waktu. Hasil konvolusi akan menghasilkan lapisan dengan jumlah kedalaman tertentu yang bergantung pada jumlah filter yang diaplikasikan, semakin banyak filter maka fitur yang diekstrak akan semakin banyak. Kemudian pada lapisan *max pooling*, nilai pada lapisan konvolusi akan direduksi untuk mempercepat proses komputasi. Hasil dari *max pooling* kemudian masuk ke *fully connected layer* untuk diklasifikasi.



Gambar 4. Arsitektur *Convolutional Neural Network* 1 dimensi.

2.5.4. Conditional Random Field

Conditional Random Field (CRF) adalah model yang digunakan untuk memprediksi data yang bersifat sekuensial. Metode ini menggunakan informasi kontekstual dari label sebelumnya, sehingga meningkatkan jumlah informasi model dan dapat membuat prediksi yang baik. Contoh aplikasi CRF pada pengolahan bahasa alami adalah pada pelabelan nama pada *text* (*named entity recognition*), ekstraksi kutipan pada jurnal, identifikasi sumber opini, dan sebagainya.

Conditional Random Field merupakan model yang bersifat diskriminatif, di mana model ini mengambil variabel *input* *x* kemudian menentukan label *y* dari variabel *x* tersebut. Contoh dari model diskriminatif adalah *logistic regression*. Kebalikan dari model diskriminatif adalah model generatif, di mana model ini menentukan bagaimana variabel label *y* secara probabilistik menghasilkan variabel *input* *x*. Contoh dari model generatif adalah Naive Bayes dan Hidden Markov Model [12].

Model CRF dapat diekspresikan pada rumus berikut:

$$P(y|x) = \frac{\exp(w \cdot x, y)}{\sum_y \exp(w \cdot \psi(x, y))} \tag{6}$$

Di mana *y* adalah label data dan *x* adalah *input data*. Sedangkan ψ adalah fungsi fitur dan *w* adalah bobot.

2.6. Evaluasi

Evaluasi dilakukan dengan menghitung nilai akurasi. Perhitungan akurasi dapat dilihat pada rumus di bawah:

Tabel 3. *Confusion Matrix*.

Aktual	Prediksi	
	Positif	Negatif
Positif	TP	FN
Negatif	FP	TN

- TP = Jumlah hasil prediksi positif dan hasil sebenarnya positif
- FN = Jumlah hasil prediksi negatif namun hasil sebenarnya positif
- FP = Jumlah hasil prediksi positif namun hasil sebenarnya negatif
- TN = Jumlah hasil prediksi negatif dan hasil sebenarnya negatif

Rumus perhitungan akurasi adalah sebagai berikut:

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FN+FP} \tag{7}$$

3. Hasil dan Pembahasan

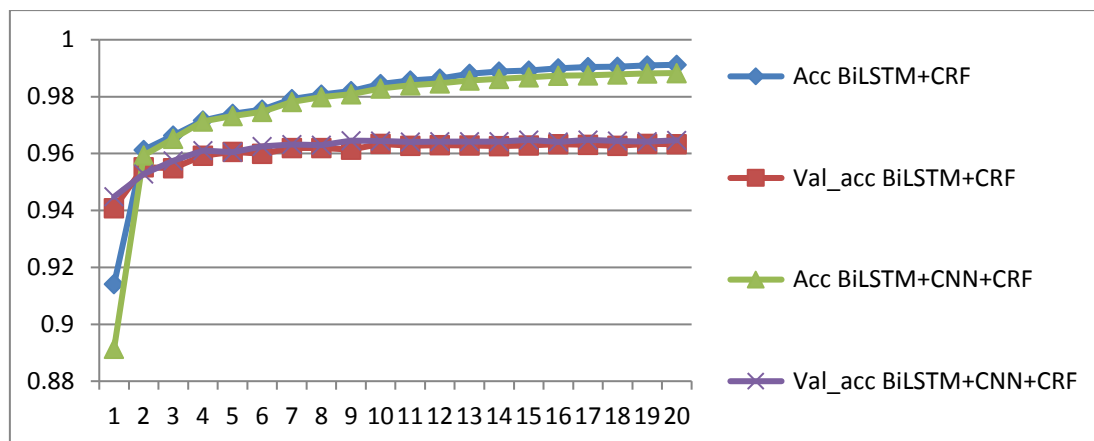
Pada penelitian ini, penulis melakukan POS Tagging pada *dataset* IDN Tagged Corpus dengan model *deep learning*. Model yang penulis gunakan berbasis LSTM dan penulis juga menambah model CNN setelah BiLSTM. Proses pelatihan menggunakan *dataset* IDN Tagged Corpus dengan metode 5-fold *cross validation* yang pembagiannya telah disediakan oleh [8]. Lama pelatihan adalah 20 *epoch*, kemudian tiap model dievaluasi dengan menghitung nilai akurasinya. Nilai akurasi yang ditampilkan pada Tabel 4 dan Tabel 5 merupakan nilai akurasi dari data uji.

Tabel 4. Performa berbagai model pada 5 set data uji yang diambil dari *dataset* IDN Tagged Corpus dengan metode *5 fold cross validation*.

Model	Akurasi					Rata-rata
	Set 1	Set 2	Set 3	Set 4	Set 5	
ELMo+NN+softmax	93.31	93.19	93.42	93.55	92.75	93.24
ELMo+BiLSTM+softmax	93.88	93.91	94.01	94.10	93.39	93.86
ELMo+BiLSTM+CRF	93.82	93.96	94.10	94.13	93.40	93.88
ELMo+BiLSTM+CNN+CRF	94.08	93.96	94.13	94.14	93.46	93.95

Tabel 4 memperlihatkan hasil pengujian dengan metode *5-fold cross validation* pada beberapa model. Dapat dilihat bahwa model ELMo+BiLSTM+CNN+CRF memiliki akurasi tertinggi, yaitu 93.92%. Kemudian diikuti model ELMo+BiLSTM+CRF sebesar 93.88%. Hal ini menunjukkan bahwa penambahan CNN setelah BiLSTM dapat meningkatkan performa model pada masalah POS Tag bahasa Indonesia.

Gambar 5 memperlihatkan grafik akurasi pada pelatihan model BiLSTM+CRF dan BiLSTM+CNN+CRF. Pada *epoch* pertama, akurasi latih pada kedua model tersebut berada di bawah akurasi validasi, kemudian pada *epoch* kedua dan seterusnya, akurasi latih melebihi akurasi validasi. Kemudian akurasi latih terus meningkat sedangkan akurasi validasi stagnan pada *epoch* 10 dan seterusnya. Hal ini menunjukkan *overfitting*. *Overfitting* dapat dilihat jika jarak antara grafik akurasi latih dan akurasi validasi cukup besar[14]. Kemudian, terlihat bahwa pada model BiLSTM+CNN+CRF memiliki jarak antara akurasi latih dan akurasi validasi yang lebih kecil dibandingkan model BiLSTM+CRF, hal ini menunjukkan bahwa penambahan CNN setelah BiLSTM dapat mengurangi *overfitting*.



Gambar 5. Grafik akurasi latih dan akurasi validasi pada BiLSTM+CRF dan BiLSTM+CNN+CRF.

Dari tabel 4 dapat diketahui bahwa penggunaan BiLSTM menghasilkan akurasi yang lebih baik dibandingkan *Neural Network* tradisional (NN). Terdapat peningkatan akurasi pada BiLSTM sebesar 0.62% dibandingkan NN. Hal ini disebabkan model BiLSTM memperhitungkan data-data di sekitarnya sehingga dapat memprediksi dengan lebih baik.

Kemudian dari tabel 4 juga diketahui bahwa model BiLSTM+CRF mengungguli model BiLSTM+softmax dengan perbedaan nilai akurasi rata-rata sebesar 0.02%. Dari 5 set data uji, model BiLSTM+CRF mengungguli 4 set data sebesar 0.01%-0.09% dan hanya set pertama yang kalah dengan nilai 0.06%. Dari informasi tersebut, dugaan penulis bahwa penggunaan CRF dapat meningkatkan akurasi adalah benar.

Pada penelitian ini, penulis juga membandingkan performa lapisan pada ELMo. ELMo memiliki 3 lapisan, di mana lapisan pertama menggunakan model CNN kemudian lapisan kedua dan ketiga menggunakan BiLSTM. Karena menggunakan model LSTM, lapisan kedua dan ketiga akan memperhitungkan kata-kata sekitarnya dalam menghasilkan vektor. Secara *default*, vektor dari ketiga lapisan tersebut dirata-ratakan dan pada penelitian di atas, vektor tersebut yang dipakai.

Tabel 5. Performa berbagai lapisan ELMo pada *dataset* IDN Tagged Corpus

Model	Akurasi 1	Akurasi 2	Akurasi 3	Akurasi 4	Akurasi 5	Rata-rata
ELMo(rata-rata)	94.08	93.96	94.13	94.14	93.46	93.95
+BiLSTM+CNN+CRF						

Model	Akurasi 1	Akurasi 2	Akurasi 3	Akurasi 4	Akurasi 5	Rata-rata
ELMo (lapisan 1) +BiLSTM+CNN+CRF	95.51	95.59	95.80	96.04	95.14	95.62
ELMo (lapisan 2) +BiLSTM+CNN+CRF	93.69	93.60	93.67	93.73	92.92	93.52
ELMo (lapisan 3) +BiLSTM+CNN+CRF	93.43	93.38	93.56	93.63	92.78	93.36

Pada tabel 5 memperlihatkan performa masing-masing layer pada ELMo. Terlihat bahwa lapisan 1 menghasilkan akurasi terbaik, yaitu 95.62%, kemudian diikuti lapisan rata-rata, lapisan 2 dan lapisan 3 sebesar 93.95%, 93.52% dan 93.36% secara berurutan. Dari informasi tersebut dapat diambil kesimpulan bahwa fitur ELMo dalam menghasilkan vektor representasi kata yang kontekstual tidak berguna pada penelitian POS Tag bahasa Indonesia ini. Hal ini disebabkan lapisan kedua dan ketiga menghasilkan nilai akurasi yang lebih rendah dibandingkan lapisan pertama, yang mana lapisan pertama ELMo yang menggunakan model CNN tidak menghasilkan vektor representasi kata yang kontekstual.

Tabel 6 memperlihatkan perbandingan performa model terbaik terhadap penelitian lainnya. Terlihat bahwa model penulis mengalahkan model Fasttext+BiLSTM+CRF [15], di mana Fasttext+BiLSTM+CRF memiliki akurasi 95.4% sedangkan model penulis 95.6%. Namun dibandingkan model IndoBERT, model penulis kalah dengan selisih 1.2%. Model ELMo kalah dibandingkan IndoBERT, namun model ini masih cukup relevan digunakan, terutama jika dipakai pada sumber daya terbatas. Hal ini disebabkan model ELMo yang penulis gunakan memiliki ukuran model yang kecil, yaitu sekitar 400 Mb, sedangkan model IndoBERT dan Fasttext memiliki ukuran yang cukup besar, yaitu lebih dari 1 Gb.

Kemudian model ELMo masih dapat ditingkatkan performanya dengan melatih menggunakan *dataset* yang lebih terkontrol. Model IndoBERT dilatih pada *dataset* dengan sumber yang relevan dan terkontrol, sedangkan pada model ELMo dari penelitian [17] menggunakan *dataset* Indonesian Conll 2017 namun tidak jelas dari mana saja *dataset* diambil. Ke depannya dapat dilakukan penelitian dengan melatih ulang model ELMo menggunakan *dataset* bahasa Indonesia dari sumber yang lebih diawasi.

Tabel 6. Performa berbagai model pada *dataset* IDN Tagged Corpus.

Model	Akurasi
IndoBERT[15]	96.8%
Fasttext+BiLSTM+CRF[15], [16]	95.4%
Model penulis	95.62%

4. Kesimpulan dan Saran

Dari penelitian tersebut diperoleh bahwa penambahan CNN setelah BiLSTM dapat meningkatkan nilai akurasi serta mengurangi *overfitting* pada masalah POS Tag bahasa Indonesia. Diperoleh bahwa model BiLSTM+CNN+CRF menghasilkan akurasi sebesar 93.95% sedangkan BiLSTM+CRF sebesar 93.88%. Selain itu, penggunaan CRF secara umum menghasilkan akurasi yang lebih baik dibandingkan menggunakan softmax.

Kemudian lapisan *embedding* ELMo pertama menghasilkan akurasi terbaik dengan nilai akurasi 95.62% dengan model BiLSTM+CNN+CRF. Sedangkan lapisan kedua dan ketiga memiliki nilai akurasi 93.52% dan 93.36%. Perlu penelitian lebih lanjut untuk mengetahui mengapa nilai akurasi pada lapisan pertama (*embedding* non kontekstual) lebih baik daripada lapisan kedua dan ketiga (*embedding* kontekstual). Kemudian untuk penelitian ke depannya juga dapat dengan melatih *embedding* ELMo dari awal dengan *dataset* bahasa Indonesia yang banyak dan terkontrol agar menghasilkan keluaran representasi kata yang lebih baik.

Daftar Pustaka

- [1] A. Kulkarni and A. Shivananda, *Natural Language Processing Recipes: Unlocking Text Data with Machine Learning and Deep Learning using Python*. 2019.
- [2] D. E. Cahyani and M. J. Vindiyanto, "Indonesian Part of Speech Tagging using Hidden Markov Model - Ngram Viterbi," in *2019 4th International Conference on Information Technology, Information Systems and Electrical Engineering, ICITISEE 2019*, Nov. 2019, pp. 353–358, doi: 10.1109/ICITISEE48480.2019.9003989.
- [3] R. S. Yuwana, E. Suryawati, and H. F. Pardede, "On Empirical Evaluation of Deep Architectures for Indonesian POS Tagging Problem," *2018 Int. Conf. Comput. Control. Informatics its Appl. Recent Challenges Mach. Learn. Comput. Appl. IC3INA 2018 - Proceeding*, no. November 2018, pp. 204–208, 2019, doi: 10.1109/IC3INA.2018.8629531.

-
- [4] Y. Munarko, Y. Azhar, M. Balqis, and S. Ekawati, "POS Tagger Tweet Bahasa Indonesia," *Kinetik*, vol. 2, no. 1, pp. 9–16, 2017, doi: 10.22219/kinetik.v2i1.169.
- [5] N. Sabloak, "Part-of-Speech (POS) Tagging Bahasa Indonesia Menggunakan Algoritma Viterbi," no. x, pp. 1–11, 2016.
- [6] G. Hinton, "Deep belief networks," *Scholarpedia*, vol. 4, no. 5, p. 5947, 2009, doi: 10.4249/scholarpedia.5947.
- [7] P. Goyal, S. Pandey, and K. Jain, *Deep Learning for Natural Language Processing: Creating Neural Networks with Python*. 2018.
- [8] K. Kurniawan and A. F. Aji, "Toward a Standardized and More Accurate Indonesian Part-of-Speech Tagging," *Proc. 2018 Int. Conf. Asian Lang. Process. IALP 2018*, pp. 303–307, 2019, doi: 10.1109/IALP.2018.8629236.
- [9] X. Ma and E. Hovy, "End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF," *54th Annu. Meet. Assoc. Comput. Linguist. ACL 2016 - Long Pap.*, vol. 2, pp. 1064–1074, 2016, doi: 10.18653/v1/p16-1101.
- [10] W. Che, Y. Liu, Y. Wang, B. Zheng, and T. Liu, "Towards Better UD Parsing: Deep Contextualized Word Embeddings, Ensemble, and Treebank Concatenation," *CoNLL 2018 - SIGNLL Conf. Comput. Nat. Lang. Learn. Proc. CoNLL 2018 Shar. Task Multiling. Parsing from Raw Text to Univers. Depend.*, pp. 55–64, 2018, doi: 10.18653/v1/K18-2005.
- [11] E. V. Murhaf Fares, Andrey Kutuzov, Stephan Oepen, "Word Vectors, Reuse, and Replicability: Towards a Community Repository of Large-text Resources," *Proc. 21st Nord. Conf. Comput. Linguist. NoDaLiDa, 22-24 May 2017, Gothenburg, Sweden*, no. 131, pp. 271–276, 2017.
- [12] A. Dinakaramani, F. Rashel, A. Luthfi, and R. Manurung, "Designing an Indonesian part of speech tagset and manually tagged Indonesian corpus," *Proc. Int. Conf. Asian Lang. Process. 2014, IALP 2014*, pp. 66–69, 2014, doi: 10.1109/IALP.2014.6973519.
- [13] M. E. Peters *et al.*, "Deep contextualized word representations," *NAACL HLT 2018 - 2018 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.*, vol. 1, pp. 2227–2237, 2018, doi: 10.18653/v1/n18-1202.
- [14] M. Cogswell, F. Ahmed, R. Girshick, L. Zitnick, and D. Batra, "Reducing Overfitting In Deep Networks By Decorrelating Representations."
- [15] F. Koto, A. Rahimi, J. H. Lau, and T. Baldwin, "IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP," Online. Accessed: Feb. 08, 2021. [Online]. Available: <https://huggingface.co/>.
- [16] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural Architectures for Named Entity Recognition." Accessed: Feb. 08, 2021. [Online]. Available: <https://github.com/>.