NAMED ENTITY RECOGNITION MENGGUNAKAN METODE *BIDIRECTIONAL LSTM-CRF* PADA TEKS BAHASA INDONESIA

Hadi Permana[1], Ken Kinanti Purnamasari[2]

[1,2]Universitas Komputer Indonesia
Jl. Dipati Ukur No.112-116, Lebakgede, Coblong, Kota Bandung, Jawa Barat 40132
E-mail : hadi.prmn@gmail.com[1], ken.kinanti@email.unikom.ac.id[2]

ABSTRAK

Named Entity Recognition (NER) atau pengenalan entitas bernama adalah salah satu bagian atau tugas dari *natural language processing (nlp)*. Tujuan dari *NER* adalah untuk mengidentifikasi atau mengklasifikasi sebuah entitas misalnya nama orang, organisasi, waktu, lokasi dan sesuatu entitas lain dalam sebuah teks yang sangat berguna dalam kasus ekstraksi informasi. Pada penelitian ini, metode yang digunakan adalah metode *Bidirectional LSTM-CRF*. *Bidirectional LSTM* menggabungkan konteks sebelumnya dan konteks setelahnya dengan memproses data dari dua arah yang selanjutnya diklasifikasi menggunakan *CRF*. Terdapat beberapa proses yang dilakukan, yaitu *preprocessing*; tokenisasi, pemberian fitur(*InitCap, AllCap, AllLower, Digits, ContaintsDigits, Punctuation*) dan selanjutnya dilakukan *training* dan *testing* dari data hasil *prepocessing*. Berdasarkan hasil pengujian menggunakan data *training* sebanyak 25.709 kata dan *testing* 9.406, metode *bidirectional* LSTM-CRF memperoleh akurasi sebesar 87.77%.

Kata Kunci : *Named Entity Recognition*, NER, *Bidirectional LSTM-CRF, Natural language Processing.*

1. PENDAHULUAN

Named Entity Recognition (NER) atau pengenalan entitas bernama adalah salah satu bagian atau tugas dari *natural language processing (nlp)*. Tujuan dari *NER* adalah untuk mengidentifikasi atau mengklasifikasi sebuah entitas misalnya nama orang, organisasi, waktu, lokasi dan sesuatu entitas lain dalam sebuah teks [1]. NER dapat digunakan dalam kasus *natural language processing* lainnya, seperti ekstraksi informasi dan pembangkit pertanyan otomatis.

Penelitian tentang *NER* sudah dilakukan di Bahasa Indonesia. Salah satunya adalah penelitian yang menggunakan metode *HMM* dalam kasus pembangkit pertanyaan otomatis [2] dengan hasil akurasi yaitu 42,54%. Rendahnya akurasi pada penelitian tersebut disebabkan oleh adanya kata-kata ambigu yang tidak terdeteksi.

Sementara itu, dalam metode pembelajaran mesin terdapat metode yang terbukti mendapatkan pencapaian performa paling tinggi*(state-of-the-art)* dalam kasus *NER* [3], yaitu *Bidirectional LSTM-CRF*. *Bidirectional LSTM* menggabungkan konteks sebelumnya dan konteks setelahnya dengan memproses data dari dua arah [1] yang selanjutnya diklasifikasi menggunakan *CRF* [3]. Pada penelitian yang dilakukan oleh Guillaume Lample, dkk [3] mengkomparasikan dua *neural* arsitektur dalam mengatasi *NER*, yaitu *Stack-LSTM* (*S-LSTM*) dan *Bidirectional LSTM-CRF*. Hasil dari komparasi pada dataset bahasa Inggris, *S-LSTM* mendapatkan akurasi 90,33% dan *Bidirectional LSTM-CRF* mendapatkan akurasi 90.94%. Pada penelitian lain yang dilakukan oleh T Anh Le, dkk [1] metode *Bidirectional LSTM-CRF* mendapatkan akurasi 87,17% dibandingkan dengan metode NeuroNER yang mendapatkan akurasi 85.37% untuk dataset Gareev's. Dari penelitian-penelitian yang telah dilakukan terbukti bahwa metode *Bidirectional LSTM-CRF* tersebut memiliki akurasi yang cukup tinggi. Oleh karena itu, dalam penelitian ini akan digunakan metode *Bidirectional LSTM-CRF* untuk menangani kasus pada teks bahasa Indonesia.

Berdasarkan Uraian tersebut maka pada penelitian ini akan dilakukan Implementasi metode *Bidirectional LSTM-CRF* pada kasus Named Entity Recognition dalam teks bahasa Indonesia dengan batasan data yang akan digunakan pada penelitian ini adalah artikel berita politik dari berbagai sumber.

2. ISI PENELITIAN

Menjelaskan tentang *named entity recognition*, berita, korpus, metode penelitian, alur proses, data masukan, tokenisasi, pemberian fitur, algoritma *long short term memory (LSTM), bidirectional LSTM, conditional random field* dan hasil pengujian.

2.1 *Named Entity Recognition*

Named Entity Recognition atau *NER* merupakan salah satu tugas dari *natural language processing* [1]

yang bertujuan dalam mengenali unit informasi seperti nama termasuk nama orang, organisasi dan nama lokasi, dan ekspresi numerik termasuk waktu, tanggal, dan ekspresi persen [1]. Jadi, *named entity recognition* bertujuan untuk mengenali entitas apapun yang memiliki sebuah informasi nama.

2.2 Berita

Berita (*news*) adalah laporan mengenai suatu peristiwa atau kejadian yang terbaru (aktual), laporan mengenai fakta -fakta yang aktual, menarik perhatian, dinilai penting, atau luar biasa [8]. Pada penelitian ini, berita yang digunakan dikhususkan pada kategori tertentu, yaitu berita politik. Hal tersebut dilakukan karena pada sistem NER (*Name Entity Recognition*) pada penelitian sebelumnya menggunakan berita dengan kategori politik.

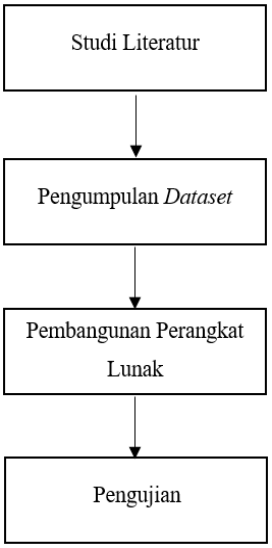
2.3 Korpus

Korpus menurut Kamus Besar Bahasa Indonesia (KBBI) adalah kumpulan ujaran yang tertulis atau lisan yang digunakan untuk menyokong atau menguji hipotesis tentang struktur bahasa. Korpus juga bisa diartikan sebagai data yang dipakai sebagai sumber bahan penelitian.

Pada penelitian ini korpus yang digunakan adalah hasil dari penelitian Rusliani [2] dan Fachri [5].

2.4 Metode Penelitian

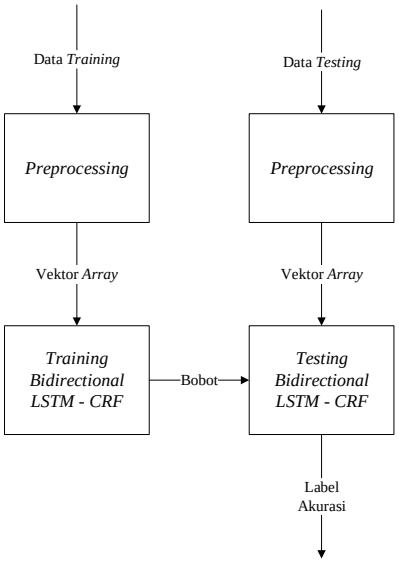
Metodologi penelitian yang digunakan adalah metode Deskriptif [6]. Terdapat empat tahap alur kerja penelitian, yaitu studi literatur, pengumpulan *dataset*, pembangunan perangkat lunak, serta pengujian. Berikut adalah alur kerja penelitian yang digunakan.



Gambar 1. Alur Penelitian

2.5 Proses

Terdapat serangkaian proses sistem yang ada pada pembangunan sistem NER. Berikut alur proses pada sistem *NER* menggunakan *bidirectional LSTM-CRF*.



Gambar 2. Alur Proses

Alur proses pada gambar 3. diawali dengan memasukkan dataset berupa artikel berita yang telah disimpan dengan format *file .txt.* pada tahap *preprocessing* terdapat dua proses yaitu tokenisasi kata dan ekstraksi fitur. Proses tokenisasi kata digunakan untuk merubah dari suatu barisan kata menjadi token kata. Proses ekstrakti fitur digunakan untuk mengenali ciri dari token kata menjadi sebuah vektor agar bisa diproses oleh suatu algoritma. Proses *training* dan *testing* menggunakan *Bidirectional Long Short Term Memory (Bi – LSTM)* dengan *output layer* menggunakan *Conditional Random Field (CRF)*. Input untuk *Bi-LSTM* berupa vektor-vektor dari sebuah token yang telah di beri fitur. Proses *training* pada *Bi-LSTM-CRF* menghasilkan bobot yang akan digunakan pada proses *testing*. Proses *testing* akan mendapatkan label dari setiap token kata dan akurasi dari metode yang digunakan.

2.6 Data Masukan

Data masukan yang digunakan dalam proses *NER* adalah data *training* dan data *testing* yang diambil dari berbagai situs berita online seperti kompas.com, detik.com dan cnnindonesia.com. data dari berbagai situs tersebut digabungkan disimpan ke dalam bentuk file *.txt*. Berikut adalah contoh data masukan yang digunakan pada penelitian ini.

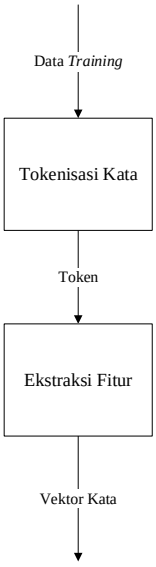
Terpidana mati kasus narkoba yang merupakan warga negara Perancis, Serge Atlaoui, mengajukan upaya hukum lanjutan berupa

peninjauan kembali ke Pengadilan Negeri Tangerang. Sebelumnya, terpidana mati asal Filipina, Mary Jane Fiesta Veloso, juga telah mengajukan langkah serupa.

Gambar 3. Contoh Data Masukan

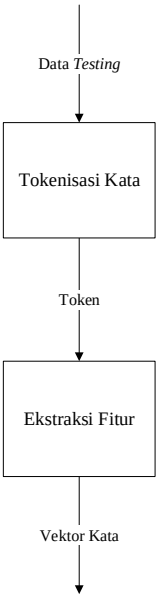
2.7 Praproses

Pada tahap *preprocessing* ini dilakukan tokenisasi kata kemudian dilakukan pemberian fitur pada kata untuk mendapatkan vektor kata. Berikut adalah blok diagram dari praposes data *training*.



Gambar 4. Praproses *training*

Sama halnya dengan data *testing*, data di tokenisasi dan kemudian dilakukan pemberian fitur untuk mendapatkan vektor kata. Berikut adalah blok diagram dari praproses data *testing*.



Gambar 5. Praproses *testing*

2.8 Tokenisasi

Tokenisasi kata merupakan tahap pemisahan teks menjadi kata per kata dengan menggunakan sebuah pemisah sehingga menjadi kumpulan token, token dapat berupa kata, simbol atau angka [7]. Dalam pemisahan kata tersebut dalam penelitian ini menggunakan sebuah *library nltk.word_tokenize*. Tahap yang dilakukan *library* tersebut adalah sebagai berikut.

1. Memisahkan token dengan space atau tab
2. Karakter tanda baca dianggap sebagai token terpisah
3. Kata dan tanda baca dianggap satu token jika diikuti dengan kata lainnya, mis : S.T, 2.3, dan lain sebagainya.

Berikut adalah tokenisasi yang dilakukan.

Tabel 1. Tokenisasi

No	Token	No	Token
1	Terpidana	23	Negeri
2	Mati	24	Tangerang
3	kasus	25	.
4	narkoba	26	sebelumnya
5	yang	27	,
6	merupakan	28	terpidana
7	warga	29	mati
8	negara	30	asal
9	Perancis	31	Filipina
10	,	32	,
11	Serge	33	Mary
12	Atlaoui	34	Jane
13	,	35	Fiesta
14	mengajukan	36	Veloso
15	upaya	37	,
16	hukum	38	juga
17	lanjutan	39	telah
18	berupa	40	mengajukan
19	peninjauan	41	langkah
20	kembali	42	serupa
21	Ke	43	.
22	Pengadilan		

2.9 Fitur

Setiap kata atau token diubah menjadi sebuah vektor agar dapat terbaca oleh metode yang digunakan. Fitur yang digunakan adalah *spelling feature* [4] dengan mengambil enam fitur berdasarkan dengan pertimbangan dari dataset yang digunakan. Berikut adalah fitur yang digunakan.

Tabel 2. Fitur

No	Fitur	Keterangan
1	InitCap	Mengenali setiap token yang hurufnya diawali dengan kapital.
2	AllCap	Mengenali setiap token yang semua hurufnya kapital.

3	AllLower	Mengenali setiap token yang semuanya huruf kecil.
4	Digits	Mengenali setiap token yang semuanya digit.
5	ContaintsDigits	Mengenali setiap token yang mengandung digit.
6	Punctuation	Mengenali setiap token yang mengandung tanda baca

Setiap fitur pada tabel 2 akan memberikan nilai 1 atau 0.

1. InitCap akan bernilai 1 jika token terdapat huruf kapital diawal kata dan jika tidak akan bernilai 0. Contohnya "Seorang","Pengamat", dan lain sebagainya.
2. AllCap akan bernilai 1 jika token terdapat semua huruf kapital dan jika tidak akan bernilai 0. Contohnya "NI", "IBM", dan lain sebagainya.
3. AllLower akan bernilai 1 jika token terdapat semua hurufnya kecil semua dan jika tidak akan bernilai 0. Contohnya "tahun", "yang", dan lain sebagainya.
4. Digits akan bernilai 1 jika token adalah angka dan jika tidak akan bernilai 0. Contohnya "2014", "21", dan lain sebagainya.
5. Gabungan Huruf dan Angka bernilai 1 jika token terdapat huruf dan angka jika tidak akan bernilai 0.
6. Punctuation bernilai satu jika token terdapat sebuah tanda baca dan jika tidak bernilai 0. Contohnya "24-Aug".

Tabel 3. Contoh Fitur

No	Kata	Fitur					
		F1	F2	F3	F4	F5	F6
1	Terpidana	1	0	0	0	0	0
2	mati	0	0	1	0	0	0
3	kasus	0	0	1	0	0	0
4	narkoba	0	0	1	0	0	0
5	yang	0	0	1	0	0	0
6	merupakan	0	0	1	0	0	0
7	warga	0	0	1	0	0	0
8	negara	0	0	1	0	0	0
9	Perancis	1	0	0	0	0	0
10	,	0	0	0	0	0	1
11	Serge	1	0	0	0	0	0
12	Atlaoui	1	0	0	0	0	0
13	,	0	0	0	0	0	1
14	mengajukan	0	0	1	0	0	0
15	upaya	0	0	1	0	0	0
16	hukum	0	0	1	0	0	0
17	lanjutan	0	0	1	0	0	0
18	berupa	0	0	1	0	0	0
19	peninjauan	0	0	1	0	0	0
20	kembali	0	0	1	0	0	0
21	ke	0	0	1	0	0	0
22	Pengadilan	1	0	0	0	0	0
23	Negeri	1	0	0	0	0	0
24	Tangerang	1	0	0	0	0	0
25	.	0	0	0	0	0	1

2.10 *Long Short Term Memory* (LSTM)

Recurrent neural networks (RNN) adalah sebuah metode yang beroperasi untuk data *sequence.* Metode ini mengambil input dari vektor *sequence* (x_1, x_2, ..., x_n) dan menjadi sequence yang lain (h_1, h_2, ..., h_n) [3]. metode RNN tidak bisa digunakan dalam long-term dependency yang menimbulkan masalah vanishing gradient [1]. Oleh sebab itu *long short term memory* (LSTM) dikembangkan untuk mengatasi permasalahan *vanishing gradient* [1]. LSTM mengganti hidden unit pada arsitektur RNN dengan unit yang bisa disebut *memory block* yang terdiri dari empat komponen yaitu : *input gate*, *output gate*, *forget gate*, dan *memory cell* [3]. Rumus dari ke-empat komponen tersebut adalah sebagai berikut.

$$i_t = \sigma\,(W_i x_t + U_i h_{t-1} + b_i) \quad (1)$$

Input gate (i_t) digunakan untuk mengubah nilai pada *cell state* (c_t) dengan W_i dan U_i adalah bobot matriks yang dikalikan dengan vektor x_t dan h_{t-1}.

$$f_t = \sigma\,(W_f x_t + U_f h_{t-1} + b_f) \quad (2)$$

Forget gate (f_t) digunakan untuk menghapus informasi dari *cell state* (c_t). Informasi dari hidden state sebelum (h_{t-1}) dan inputan (x_t) dihitung dengan fungsi sigmoid (σ) dengan nilai keluaran antara 0 dan 1. Jika nilai keluaran mendekati 0 maka informasi akan di hapus, dan jika nilai keluaran mendekati 1 maka informasi akan disimpan.

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh\,(W_c x_t + U_c h_{t-1} + b_c) \quad (3)$$

Cell state sebelumnya(c_{t_1}) dikali ($\odot$) dengan *forget gate* (f_t), terdapat kemungkinan nilai dari hasil perkalian tersebut akan menurun jika dikalikan dengan nilai (f_t) yang mendekati 0. Lalu nilai dari i_t dikali ($\odot$) dengan fungsi *cell state* saat ini untuk mendapatkan nilai *cell state* saat ini dengan nilai antara 1 sampai -1.

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

Sama halnya dengan input gate (i_t), output gate (o_t) digunakan untuk menentukan nilai dari hidden state baru (h_t).

$$h_t = o_t \odot \tanh(c_t) \quad (5)$$

Nilai dari output gate (o_t) dikalikan dengan hasil dari fungsi tanh untuk *cell state* (c_t). Yang mana *hidden state* baru dan *cell state* baru akan digunakan untuk perhitungan pada langkah selanjutnya (t).

Dimana σ adalah fungsi sigmoid dengan persamaan

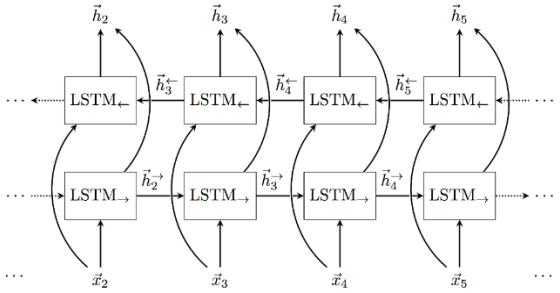
$$f(x) = \frac{1}{1+\exp(-x)} \quad (6)$$

Dan tanh dengan persamaan

$$tanh(x) = \frac{2}{1+\exp(-2x)} - 1 \quad (7)$$

2.11 *Bidirectional LSTM*

Bidirectional LSTM memanfaatkan konteks sebelumnya dan konteks setelahnya dengan memproses data dari dua arah dengan *hidden layer* terpisah [10] [1]. *Forward layer* untuk merepresentasikan konteks sebelumnya, dan *backward layer* untuk merepresentasikan konteks setelahnya [1]. Keluaran dari kombinasi dua arah *hidden layer* $\vec{h}_t$ dan $\overleftarrow{h_t}$ adalah : $y_t = W_{\vec{h}y}\vec{h}_t + W_{\overleftarrow{h}y}\overleftarrow{h_t}$.



Gambar 6. Arsitektur *Bidirectional* LSTM

2.12 *Conditional Random Field*(CRF)

Conditional Random Field merupakan sebuah model probabilistik yang digunakan untuk memprediksi data terstruktur yang telah digunakan dalam berbagai tugas, seperti *computer vision, natural language processing*.

Model CRF melakukan *training* untuk memprediksi sebuah vektor y {y0,y1,y2,..,yT} dari sebuah kalimat X {x0, x1, x2,..xT} dengan

$$p(y|x) = \frac{e^{score(x,y)}}{\sum_{y'} e^{score(x,y')}} \quad (8)$$

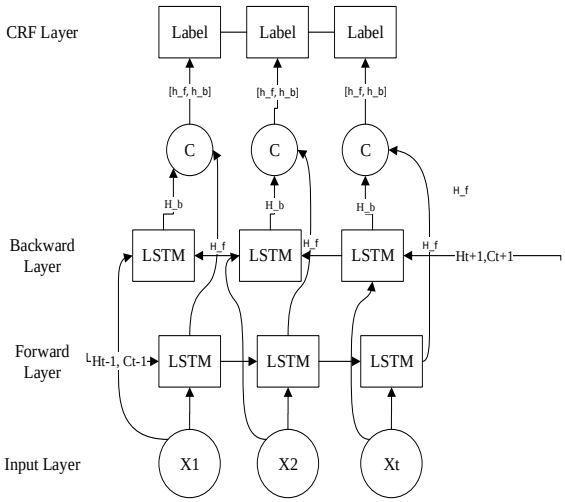
Dimana untuk mencari $score(x,y)$ dapat menggunakan

$$score(x,y) = \sum_{i=0}^{T} A_{yi,yi+1} + \sum_{i=1}^{T} P_{i,yi} \quad (9)$$

Dimana $A_{yi,yi+1}$ adalah probabilitas emisi yang mewakili skor transisi dari tag *i* ke tag *j*. $P_{i,yi}$ adalah probabilitas transisi yang mewakili skor transisi dari tag *j* ke kata *i*.

2.13 *Bidirectional LSTM-CRF*

Bidirectional LSTM-CRF adalah sebuah kombinasi metode antara *bidirectional LSTM* dan model *CRF*. Vektor kata di hitung dengan *bidirectional LSTM* untuk menghasilkan skor yang mewakili kemungkinan tag pada setiap kata didalam kalimat. Artinya $P_{i,yi}$ dari persamaan 9 diganti dengan hasil dari *Bidirectional LSTM*[1]. Sehingga pada model CRF hanya menghitung $A_{yi,yi+1}$.



Gambar 7. Arsitektur *Bidirectional LSTM-CRF*

2.14 Hasil Pengujian

Hasil pengujian akurasi didapatkan dari dua skenario pengujian, yaitu mengubah jumlah *epoch* dan mengubah nilai *learning rate*. Berikut merupakan hasil pengujian akurasi yang dilakukan terhadap data *testing.*

2.14.1 Skenario Pengujian 1

Pada skenario pengujian 1, pengujian metode dilakukan dengan mengubah nilai *epoch* menjadi 40, dan nilai *learning rate* menjadi 0.01, 0.015, 0.02 dan 0.001.

a. *Epoch* 40 dan *learning rate* 0.01

Berikut adalah hasil prediksi sistem dari enam entitas yaitu *person, organization, time, quantity location* dan *other*.

Tabel 4. Hasil Prediksi

Name Entiy	**Hasil Sistem**	**Prediksi Benar**
Person	239	669
Organization	73	296
Time	237	254
Quantity	38	42
Location	112	183
Other	7448	7962

Total	**8147**	**9406**

Dari tabel berikut akurasi yang didapat bisa dihitung dengan membagi total hasil sistem dengan total prediksi benar, kemudian hasil bagi tersebut dikalikan dengan seratus persen. Maka akurasi dari pengujian dengan mengubah nilai *epoch* menjadi 40 dan *learning rate* menjadi 0.01 adalah 86.61%

b. *Epoch* 40 dan *learning rate* 0.015

Berikut adalah hasil prediksi sistem dari enam entitas yaitu *person, organization, time, quantity location* dan *other*.

Tabel 5. Hasil Prediksi

Name Entiy	**Hasil Sistem**	**Prediksi Benar**
Person	274	669
Organization	53	296
Time	236	254
Quantity	38	42
Location	105	183
Other	7502	7962
Total	**8208**	**9406**

Dari tabel berikut akurasi yang didapat bisa dihitung dengan membagi total hasil sistem dengan total prediksi benar, kemudian hasil bagi tersebut dikalikan dengan seratus persen. Maka akurasi dari pengujian dengan mengubah nilai *epoch* menjadi 40 dan *learning rate* menjadi 0.015 adalah 87.26%

c. *Epoch* 40 dan *learning rate* 0.02

Berikut adalah hasil prediksi sistem dari enam entitas yaitu *person, organization, time, quantity location* dan *other*.

Tabel 6. Hasil Prediksi

Name Entiy	**Hasil Sistem**	**Prediksi Benar**
Person	258	669
Organization	60	296
Time	235	254
Quantity	38	42
Location	103	183
Other	7507	7962
Total	**8201**	**9406**

Dari tabel berikut akurasi yang didapat bisa dihitung dengan membagi total hasil sistem dengan total prediksi benar, kemudian hasil bagi tersebut dikalikan dengan seratus persen. Maka akurasi dari pengujian dengan mengubah nilai *epoch* menjadi 40 dan *learning rate* menjadi 0.02 adalah 87.18%

d. *Epoch* 40 dan *learning rate* 0.001

Berikut adalah hasil prediksi sistem dari enam entitas yaitu *person, organization, time, quantity location* dan *other*.

Tabel 7. Hasil Prediksi

Name Entiy	**Hasil Sistem**	**Prediksi Benar**
Person	205	669
Organization	42	296
Time	236	254
Quantity	38	42
Location	95	183
Other	7587	7962
Total	**8203**	**9406**

Dari tabel berikut akurasi yang didapat bisa dihitung dengan persamaan membagi total hasil sistem dengan total prediksi benar, kemudian hasil bagi tersebut dikalikan dengan seratus persen. Maka akurasi dari pengujian dengan mengubah nilai *epoch* menjadi 40 dan *learning rate* menjadi 0.01 adalah 87.21%. berikut adalah tabel dari hasil pengujian untuk skenario 1.

Tabel 8. Skenario Pengujian 1

Epoch	***Learning rate***	**Akurasi(%)**
40	0.01	86.61
40	0.015	87.26
40	0.02	87.18
40	0.001	87.21

Berdasarkan dari hasil pengujian skenario 1 dapat diambil kesimpulan bahwa nilai *epoch* dan *learning rate* yang memiliki akurasi paling tinggi adalah nilai *epoch* 40 dan nilai *learning rate* 0.015 dengan akurasi sebesar 87.26%.

2.14.2 Skenario Pengujian 2

Pada skenario pengujian 2, pengujian metode dilakukan dengan mengubah nilai *epoch* menjadi 50, dan nilai *learning rate* menjadi 0.01, 0.015, 0.02 dan 0.001.

a. *Epoch* 50 dan *learning rate* 0.01

Berikut adalah hasil prediksi sistem dari enam entitas yaitu *person, organization, time, quantity location* dan *other*.

Tabel 9. Hasil Prediksi

Name Entiy	**Hasil Sistem**	**Prediksi Benar**
Person	338	669
Organization	57	296
Time	232	254
Quantity	38	42
Location	99	183
Other	7478	7962

Total	**8242**	**9406**

Dari tabel berikut akurasi yang didapat bisa dihitung dengan membagi total hasil sistem dengan total prediksi benar, kemudian hasil bagi tersebut dikalikan dengan seratus persen. Maka akurasi dari pengujian dengan mengubah nilai *epoch* menjadi 50 dan *learning rate* menjadi 0.01 adalah 87.62%

b. *Epoch* 50 dan *learning rate* 0.015

Berikut adalah hasil prediksi sistem dari enam entitas yaitu *person, organization, time, quantity location* dan *other*.

Tabel 10. Hasil Prediksi

Name Entiy	**Hasil Sistem**	**Prediksi Benar**
Person	328	669
Organization	58	296
Time	236	254
Quantity	38	42
Location	96	183
Other	7457	7962
Total	**8213**	**9406**

Dari tabel berikut akurasi yang didapat bisa dihitung dengan membagi total hasil sistem dengan total prediksi benar, kemudian hasil bagi tersebut dikalikan dengan seratus persen. Maka akurasi dari pengujian dengan mengubah nilai *epoch* menjadi 50 dan *learning rate* menjadi 0.015 adalah 87.32%.

c. *Epoch* 50 dan *learning rate* 0.02

Berikut adalah hasil prediksi sistem dari enam entitas yaitu *person, organization, time, quantity location* dan *other*.

Tabel 11. Hasil Prediksi

Name Entiy	**Hasil Sistem**	**Prediksi Benar**
Person	327	669
Organization	61	296
Time	234	254
Quantity	38	42
Location	102	183
Other	7491	7962
Total	**8253**	**9406**

Dari tabel berikut akurasi yang didapat bisa dihitung dengan membagi total hasil sistem dengan total prediksi benar, kemudian hasil bagi tersebut dikalikan dengan seratus persen. Maka akurasi dari pengujian dengan mengubah nilai *epoch* menjadi 50 dan *learning rate* menjadi 0.02 adalah 87.74%

d. *Epoch* 50 dan *learning rate* 0.001

Berikut adalah hasil prediksi sistem dari enam entitas yaitu *person, organization, time, quantity location* dan *other*.

Tabel 12. Hasil Prediksi

Name Entiy	**Hasil Sistem**	**Prediksi Benar**
Person	217	669
Organization	40	296
Time	237	254
Quantity	38	42
Location	93	183
Other	7631	7962
Total	**8256**	**9406**

Dari tabel berikut akurasi yang didapat bisa dihitung dengan membagi total hasil sistem dengan total prediksi benar, kemudian hasil bagi tersebut dikalikan dengan seratus persen. Maka akurasi dari pengujian dengan mengubah nilai *epoch* menjadi 50 dan *learning rate* menjadi 0.001 adalah 87.77%. berikut adalah tabel dari hasil pengujian untuk skenario 2.

Tabel 13. Skenario Pengujian 2

Epoch	***Learning rate***	**Akurasi(%)**
50	0.01	87.62
50	0.015	87.31
50	0.02	87.61
50	0.001	87.77

Berdasarkan dari hasil pengujian skenario 2 dapat diambil kesimpulan bahwa nilai *epoch* dan *learning rate* yang memiliki akurasi paling tinggi adalah nilai *epoch* 50 dan nilai *learning rate* 0.001 dengan akurasi sebesar 87.77%.

3. PENUTUP

Berdasarkan hasil dari dua skenario pengujian yang telah dilakukan, maka dapat diperoleh kesimpulan bahwa akurasi dari metode *bidirectional LSTM-CRF* pada sistem *NER* bahasa Indonesia adalah sebesar 87,77% . Akurasi tersebut didapatkan dari susunan parameter yang memiliki batas *epoch* 50, *learning rate* sebesar 0,001.

Adapun saran untuk penelitian selanjutnya adalah penambahan data dengan entitas nama, organisasi, *time* dan *quantity* dikarenakan pada *dataset* yang dipakai hampir 80% adalah data dengan entitas *other*. Juga bisa dilakukan penambahan fitur agar bisa mendeteksi entitas dengan lebih akurat.

DAFTAR PUSTAKA

[1]G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami dan C. Dyer, “Neural Architectures for

Named Entity Recognition," arxiv:1603.01360v3, 2016.
[2]A. L. T., A. M. Y. dan B. M. S., "Application of a Hybrid Bi-LSTM-CRF model to the task of Russian Named Entity Recognition," 2017.
[3]Rusliani, "Named Entity Recognition Pada Teks Berbahasa Indonesia Untuk Pembangkit Pertanyaan Otomatis," UNIKOM, Bandung, 2017.
[4]Z. Huang, W. Xu dan K. Yu, "Bidirectional LSTM-CRF Models for Sequence Tagging," arxiv:1508.01991v1, 2015.
[5]M. Fachri, "Pengenalan Entitas Bernama Pada Teks Bahasa Indonesia Menggunakan Hidden Markov Model," Universitas Gadjah Mada, Yogyakarta, 2014.
[6]Z. A. Hasibuan, Metodologi Penelitian Pada Bidang Ilmu Komputer Dan Teknoogi Informasi, Depok: Universitas Indonesia, 2007.
[7]K. K. Purnamasari dan I. S. Suwardi, "Rule-based Part of Speech Tagger for Indonesian Language," IOP Conference Series: Materials Science and Engineering, no. 1, pp. 1-4, 2018.
[8]K. Budiman, "Dasar - Dasar Jurnalistik," dalam Pelatihan Jurnalistik, Jawa, Info Jawa, 2005, pp. 1 - 4.
[9]F. Amin, "Sistem Temu Kembali Informasi dengan Metode Vector Space Model," Jurnal Sistem Informasi Bisnis, vol. 02, pp. 78-83, 2012.
[10]M. Maimaiti, A. Wumaier, K. Abiderexiti dan T. Yibulayin, "Bidirectional Long Short-Term Memory Network with a Conditional Random Field Layer for Uyghur Part-Of-Speech Tagging," 2017.
[11]D. Bell, "An introduction to the Unified Modeling Language," IBM, 15 June 2003. [Online]. Available: https://www.ibm.com/developerworks/rational/library/769.html. [Diakses 4 October 2018].
[12]S. W. Ambler, "Introduction to the Diagrams of UML 2.X," Agile Modeling, [Online]. Available: http://www.agilemodeling.com/essays/umlDiagrams.htm. [Diakses 4 October 2018].
[13]F. A. Aslam dan H. N. Mohammed, "Efficient Way Of Web Development Using Python And Flask," International Journal of Advanced Research in Computer Science, vol. 6, no. 2, pp. 54-57, 2015.
[14]I. A. Diana, Sistem Komunikasi, Bandung: Universitas Pendidikan Indonesia, 2012.
[15]K. Xu, Z. Zhou, T. Hao dan W. Liu, "A Bidirectional LSTM and Conditional Random Fields Approach to Medical Named Entity Recognition," 2017.
[16]Z. Liu, B. Tang, X. Wang dan Q. Chen, "De-identification of clinical notes via reccurent neural network and conditional random field," Journal of Biomedical Informatics, 2017.
[17]A. Solihin, PEMROGRAMAN WEB DENGAN PHP DAN MYSQL, Penerbit Budi Luhur, 2016.
[18]K. Xu, Z. Zhou, T. Hao dan W. Liu, "A Bidirectional LSTM and Conditional Random Fields Approach to Medical Named Entity Recognition," 2018.
[19]A. Setioaji, L. Muflikhah dan M. A. Fauzi, "Named Entity Recognition Menggunakan Hidden Markov Model dan Algoritma Viterbi pada Teks Tanaman Obat," Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, vol. I, no. 12, pp. 1858-1864, 2017.
[20]N. Jaariyah, "Pengenalan Entitas Bernama pada Teks Bahasa Indonesia Menggunakan Conditional Random Fields," Perpustakaan UNIKOM, Bandung, 2017.
[21]M. I. Tiarasani, "Pengenalan Entitas Bernama Pada Artikel Berita Berbahasa Indonesia Menggunakan Metode Hidden Markov Model Dan Rule Based," UGM, Yogyakarta, 2018.
[22]I. Irfana, "Pengenalan Entitas Bernama Pada Artikel Berita Bahasa Indonesia Menggunakan Metode Berbasis Aturan," UGM, Yogyakarta, 2015.