LAPORAN AKHIR

11S4037 – Pemrosesan Bahasa Alami

Indonesian Named Entity Recognition using POS Tagging

Disusun oleh:

12S18020 Dita L. Sastri Sihombing

12S18029 Estomihi Rascana Sirait

12S18061 Angela Friscilia Simamora

PROGRAM STUDI SARJANA SISTEM INFORMASI

FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO

INSTITUT TEKNOLOGI DEL

2021

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I PENDAHULUAN

Pada bab ini menyajikan latar belakang, tujuan, manfaat, dan ruang lingkup pengerjaan proyek.

1.1 Latar Belakang

Named Entity Recognition merupakan kata benda yang mengacu pada jenis individu tertentu seperti nama organisasi, nama orang, nama lokasi, dan sebagainya [1]. *Named Entity Recognition* melibatkan pemrosesan teks dan mengidentifikasi kemunculan kata atau ekspresi tertentu sebagai milik kategori tertentu *Named Entity* (NE). Perangkat lunak pengenalan *named entity* berfungsi sebagai alat pemrosesan awal yang penting untuk tugas-tugas seperti ekstraksi informasi, pengambilan informasi, dan aplikasi pemrosesan teks lainnya. Apa yang dianggap sebagai *named entity* bergantung pada aplikasi yang menggunakan anotasi. Salah satu aplikasi tersebut adalah pengambilan dokumen atau penerusan dokumen otomatis: dokumen yang dicatat dengan informasi *named entity* dapat dicari lebih akurat daripada teks mentah [2].

Dalam beberapa dokumen teks untuk memperoleh banyak informasi yang penting seperti nama orang, nama lokasi, nama organisasi yang dimana dalam dokumen tentu dilakukan dengan manual yaitu membaca keseluruhan teks yang ada, diperlukan waktu yang banyak lagi jika sebuah dokumen sangat panjang. Di saat sekarang ini sudah banyak dilakukan pembahasan terkait *Named Entity*, mengingat bahwa entitas dari sebuah dokumen itu penting dan dengan upaya yang dilakukan membuat *Named Entity Recognition* dapat digunakan untuk mendeteksi informasi secara otomatis sehingga tidak perlu menghabiskan banyak waktu untuk membaca dokumen teks keseluruhan. *Named Entity Recognition* dapat diimplementasikan pada machine translation, question answering dan semantic web.(Leonandya, 2015) [1]. Namun dalam beberapa sumber penelitian terdapat kekurangan pada proses *automatic tagging*. *Automatic tagging* merupakan proses untuk melakukan *tagging* pada setiap kata atau frasa dengan jenis entitasnya. Maka dari itu dalam proyek ini mencoba meningkatkan kemampuan *automatic tagging* dalam mengimplementasikan *POS-Tagging* dengan beberapa aturan tambahan pada proses automatic tagging tersebut. Dengan mengimplementasikan *POS-Tagging* dengan tujuan untuk memperoleh seluruh kata atau frasa yang memiliki kemungkinan mempunyai jenis entitas dan selanjutnya kata

atau frasa tersebut akan dilakukan pengecekkan dengan *rule* yang telah disediakan dan akan dilakukan tag entitas berdasarkan aturan yang digunakan.

1.2 Rumusan Masalah

Berdasarkan pembahasan masalah yang telah dibahas sebelumnya, dimana terdapat kekurangan pada saat proses *automatic tagging* pada pelabelan banyak kata atau frasa. Maka dari itu, diperlukan metode untuk meningkatkan proses kerja *automatic tagging* dalam melakukan pelabelan.

1.3 Tujuan

Tujuan proyek ini adalah:

1. Menghasilkan model *Named Entity Recognition* dengan *tag entitas* pada kata dalam dokumen teks Bahasa Indonesia.
2. Memperkaya pengetahuan bagi penulis maupun pembaca terkait *Named Entity Recognition* dengan *POS-Tagging.*

1.4 Manfaat

Manfaat proyek ini adalah:

1. Memberikan model *Named Entity Recognition* dengan tag entitas yang sesuai pada kata atau frasa yang terdapat dalam dokumen teks Bahasa Indonesia dengan memanfaatkan pemrosesan bahasa alami.
2. Sebagai tahapan awal dalam *Information Extraction* Bahasa Indonesia

1.5 Ruang Lingkup

Ruang lingkup dari proyek ini adalah:

1. Membangun sistem *Named Entity Recognition* yang digunakan untuk menganalisis teks Bahasa Indonesia
2. Menggunakan set data berisi teks Bahasa Indonesia, yaitu SINGGALANG.tsv (https://github.com/ialfina/ner-dataset-modified-dee/tree/master/singgalang)

1.6 Sistematika Penyajian

Adapun sistematika penyajian dari proyek ini adalah:

Bab 1. Pendahuluan, membahas tentang latar belakang, pertanyaan penelitian, tujuan, manfaat, ruang lingkup, dan sistematika penyajian penelitian.

Bab 2. Isi, menjelaskan mengenai teori-teori yang berkaitan dengan proyek diantaranya metode yang digunakan, proses dan perangkat (tools) terkait dengan tujuan penelitian.

Bab 3. Rencana, menjelaskan terkait jadwal kerja pengerjaan proyek dan pembagian tugas.

BAB II ISI

Pada bab ini dijelaskan analisis yang dilakukan terhadap data dan metode, desain pemrosesan bahasa alami, implementasi kode program serta hasil dari proyek pemrosesan bahasa alami. Dalam proyek PBA ini adapun dilakukan analisis dataset baik dengan manual dan menggunakan kode program untuk memahami konteks antar variabel, dan untuk memahami data. Kemudian dilakukan data preprocessing yang terdiri dari cleaning dan tokenization. Metode Pos tagging akan diimplementasikan dengan data yang dihasilkan yang merupakan data yang sudah bersih dan ter tokenisasi. Pos tagging adalah proses memberikan label atau tag atau entitas pada kata yang menghasilkan data yang berkualitas dengan informasi yang jelas pemberian tag entitas. Ide perbaikan berdasarkan pendekatan yang dilakukan dalam penyelesaian proyek ini seperti data set yang akan dianalisis lebih baik menggunakan csv dari pada tsv agar tidak terlalu banyak kendala error dengan kode program seperti load data, selanjutnya perlu dilakukan data preprocessing berulang ulang untuk membersihkan data sehingga lebih mudah diimplementasikan pos tagging. Karena pekerjaan proyek ini mengimplementasikan pendekatan pos tagging yang belum berhasil untuk melabeli data.

2.1 Analisis

Pada sub bab ini membahas mengenai analisis yang dilakukan terhadap data dan metode yang digunakan. Dataset yang digunakan dalam proyek yaitu SINGGALANG.tsv diperoleh melalui link berikut : https://github.com/ialfina/ner-dataset-modified-dee/tree/master/singgalang. Dalam file SINGGALANG.tsv tersebut berisi data dengan identifikasi sebagai *'Word'* dan *'Entity'* sebagai berikut :

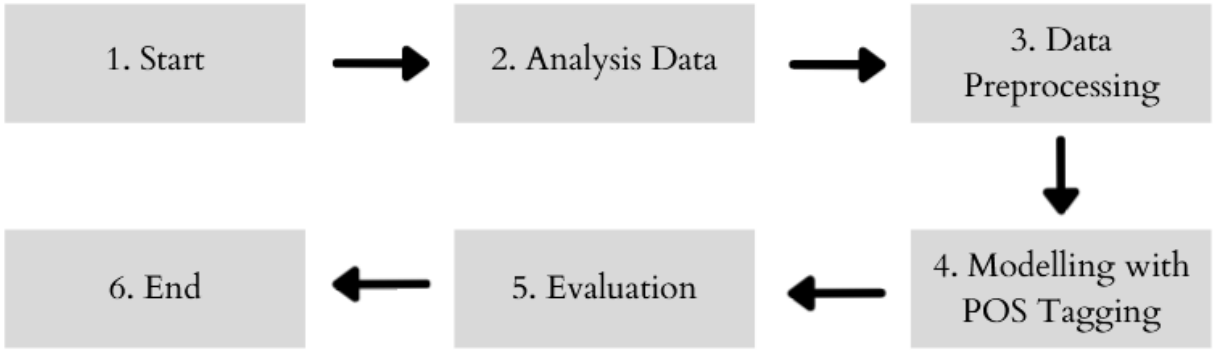
	Word	Entity
0	la	O
1	menjabat	O
2	sebagai	O
3	Presiden	O
4	ketiga	O
...	...	...
712933	zgjidhet	O
712934	presidente	O
712935	e	O
712936	Republikës	O
712937	'	O

712938 rows × 2 columns

Gambar 1. Dataset SINGGALANG.tsv

2.2 Desain

Pada sub bab ini dijelaskan desain pemrosesan bahasa alami ditampilkan dalam bentuk flowchart atau diagram alir sebagai berikut.



Gambar 2. Desain Analisis NER

2.2.1 Analysis Data

Dengan melakukan analisis terhadap data yang digunakan bertujuan dalam memahami data, mendapat konteks data serta memahami hubungan antara variabel. Berikut merupakan info serta deskripsi terkait dataset yang akan digunakan.

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 712938 entries, 0 to 712937
Data columns (total 2 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   Word    689461 non-null  object
 1   Entity  689905 non-null  object
dtypes: object(2)
memory usage: 10.9+ MB
```

Gambar 3. Info Dataset SINGGALANG.tsv

```
dataset.describe()
```

	Word	Entity
count	689461	689905
unique	59555	4
top	,	O
freq	40084	625260

Gambar 4. Deskripsi Dataset SINGGALANG.tsv

2.2.2 Data Preprocessing

Preprocessing data dilakukan untuk mengubah data mentah atau biasa dikenal dengan raw data yang dikumpulkan dari berbagai sumber menjadi informasi yang lebih bersih dan bisa digunakan untuk pengolahan selanjutnya.

2.2.2.1 Data Cleaning

Data set yang baru dikumpulkan pasti memiliki *missing value* atau *noise*. Hal ini karena proses pengumpulan datanya tidak sempurna sehingga ada banyak bagian yang tidak relevan dan hilang. Metode yang harus digunakan untuk mengatasi *missing value* dan *noise* yaitu dengan melakukan *data cleaning* diterapkan untuk menghilangkan noise, memperbaiki ketidakkonsistenan data.

2.2.2.2 Tokenization

Tokenisasi merupakan proses memotong kata dengan memecah mecah kalimat (*segmenting*) dan setiap kata akan di normalisasi untuk disesuaikan dengan standar tertentu. Tokenisasi juga menjadi bagian segmenting sentences, yaitu pada teks panjang dengan membagi kalimat per kalimat dengan karakter tanda baca. Pada proyek ini tokenisasi dilakukan dengan menerapkan kode program tokenization pada set data.

2.2.3 Modelling with POS Tagging

Model merupakan hasil dari training tapi bukan datanya. namun lebih mirip pada sebuah rule/suatu fungsi matematika yang nantinya digunakan untuk memprediksi suatu tag/label dari data yg diberikan. Named Entity Recognition dapat memperoleh informasi seperti nama orang, tempat dan organisasi pada sebuah teks. Dalam menghasilkan model, metode yang digunakan adalah metode POS-tagging untuk meningkatkan jumlah data berlabel. Dalam proyek ini peningkatan data berlabel akan digunakan pada teks data SINGGALANG.

2.2.4 Evaluation

Setelah model selesai dibangun lalu dilakukan melakukan evaluasi untuk mengukur keakuratan dari sistem. Dalam tahap evaluasi digunakan parameter precision yang merupakan perbandingan antara jumlah entitas yang berhasil di tag dengan benar oleh model dengan jumlah entitas yang berhasil di tag. Untuk persamaan dari nilai precision dapat dilihat sebagai berikut.

$$precision = \frac{TP}{TP+FP}$$

Dengan :

- TP (*True Positive*) adalah jumlah entitas yang berhasil di tag dengan benar oleh model.
- FP (*False Positive*) adalah jumlah entitas yang berhasil di tag oleh model tetapi bernilai salah.

2.3 Implementasi

Pada bab ini dijelaskan pengimplementasian pemrosesan bahasa alami yaitu *Named Entity Recognition* menggunakan *POS Tagging*.

2.3.1 Data Preprocessing

Pada bagian ini dijelaskan *data preprocessing* yang dilakukan sebelum digunakan dalam pemodelan, mencakup *data cleaning*, *tokenization* dan *modelling with POS Tagging*.

2.3.1.1 Data Cleaning

Pada bagian ini dilakukan *data cleaning* dengan melakukan pemeriksaan terhadap *missing value* dari *dataset*. Berikut merupakan kode program dalam mendeteksi *missing value* pada *dataset* yang digunakan.

```
# detection missing values

dataset.isnull()
```

Gambar 5. Kode Program Deteksi *Missing Value* pada Dataset SINGGALANG.tsv

Dilakukan *run* dengan kode program deteksi *missing value* pada dataset SINGGALANG dapat dilihat bahwa *missing value* pada data tersebut *false* seperti pada gambar berikut.

```
dataset.isnull()
```

	Word	Entity
0	False	False
1	False	False
2	False	False
3	False	False
4	False	False
...	...	...
712933	False	False
712934	False	False
712935	False	False
712936	False	False
712937	False	False

712938 rows × 2 columns

Gambar 6. Hasil Deteksi *Missing Value* pada Dataset SINGGALANG.tsv

Setelah dilakukan deteksi pada *missing value*, berikut kode program untuk melakukan penjumlahan *missing value* pada dataset.

```
# data aggregation

dataset.isnull().sum()
```

Gambar 7. Kode Program Peringkas*an Missing Value* pada Dataset SINGGALANG.tsv

Hasil penjumlahan *missing value* pada train data ditunjukkan pada gambar berikut.

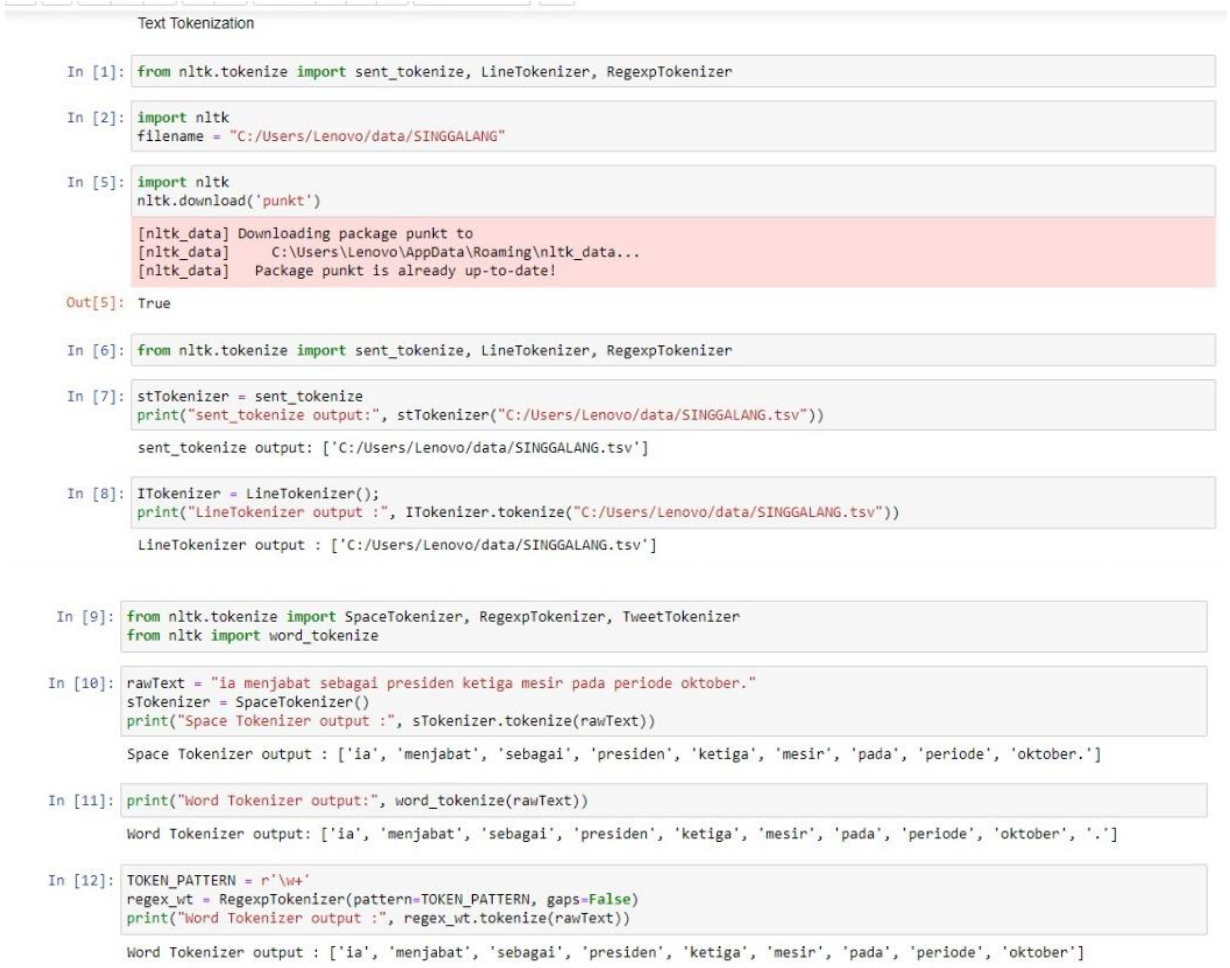
```
dataset.isnull().sum()

Word       23477
Entity     23033
dtype: int64
```

Gambar 8. Hasil Penjumlahan *Missing Value* pada Dataset SINGGALANG.tsv

2.3.1.2 Tokenization

Pada bagian ini dilakukan tokenisasi terhadap data SINGGALANG sebagai berikut.



Text Tokenization

```
In [1]: from nltk.tokenize import sent_tokenize, LineTokenizer, RegexpTokenizer

In [2]: import nltk
        filename = "C:/Users/Lenovo/data/SINGGALANG"

In [5]: import nltk
        nltk.download('punkt')

        [nltk_data] Downloading package punkt to
        [nltk_data]     C:\Users\Lenovo\AppData\Roaming\nltk_data...
        [nltk_data]   Package punkt is already up-to-date!

Out[5]: True

In [6]: from nltk.tokenize import sent_tokenize, LineTokenizer, RegexpTokenizer

In [7]: stTokenizer = sent_tokenize
        print("sent_tokenize output:", stTokenizer("C:/Users/Lenovo/data/SINGGALANG.tsv"))

        sent_tokenize output: ['C:/Users/Lenovo/data/SINGGALANG.tsv']

In [8]: ITokenizer = LineTokenizer();
        print("LineTokenizer output :", ITokenizer.tokenize("C:/Users/Lenovo/data/SINGGALANG.tsv"))

        LineTokenizer output : ['C:/Users/Lenovo/data/SINGGALANG.tsv']

In [9]: from nltk.tokenize import SpaceTokenizer, RegexpTokenizer, TweetTokenizer
        from nltk import word_tokenize

In [10]: rawText = "ia menjabat sebagai presiden ketiga mesir pada periode oktober."
         sTokenizer = SpaceTokenizer()
         print("Space Tokenizer output :", sTokenizer.tokenize(rawText))

         Space Tokenizer output : ['ia', 'menjabat', 'sebagai', 'presiden', 'ketiga', 'mesir', 'pada', 'periode', 'oktober.']

In [11]: print("Word Tokenizer output:", word_tokenize(rawText))

         Word Tokenizer output: ['ia', 'menjabat', 'sebagai', 'presiden', 'ketiga', 'mesir', 'pada', 'periode', 'oktober', '.']

In [12]: TOKEN_PATTERN = r'\w+'
         regex_wt = RegexpTokenizer(pattern=TOKEN_PATTERN, gaps=False)
         print("Word Tokenizer output :", regex_wt.tokenize(rawText))

         Word Tokenizer output : ['ia', 'menjabat', 'sebagai', 'presiden', 'ketiga', 'mesir', 'pada', 'periode', 'oktober']
```

Gambar 9. Kode Program *Tokenization* pada Dataset SINGGALANG.tsv

BAB III PENUTUP

Pada bab ini dijelaskan mengenai pembagian tugas dan tanggung jawab dalam pengerjaan proyek, kesimpulan yang diperoleh, dan saran terhadap proyek ke depannya.

3.1 Pembagian Tugas dan Tanggung Jawab

Pada subbab ini dijelaskan pembagian tugas dan tanggung jawab dari setiap anggota dalam pengerjaan proyek.

Tabel 1. Pembagian Tugas dan Tanggung Jawab

Name	Task
Estomihi Rascana Sirait	Programmer - Mengimplementasikan code untuk membangun sistem - Menguji sistem yang telah dibuat, agar aplikasi yang dibuat bisa bermanfaat untuk pengguna
Dita L. Sastri Sihombing	System Analyst - Mengumpulkan serta menganalisis data - Membuat diagram alir, dan spesifikasi yang akan digunakan
Angela Friscilia Simamora	System Analyst - Mengumpulkan serta menganalisis data - Membuat diagram alir, dan spesifikasi yang akan digunakan

3.2 Kesimpulan

Apabila metode POS Tagging pada dataset SINGGALANG dapat diimplementasikan maka dapat menganalisis entitas dan mengidentifikasi person, location dan time. POS Tagging pada proyek ini hanya dapat dilakukan tokenisasi, sehingga tidak dapat meningkatkan jumlah data berlabel.

3.3 Saran

Sebagai saran dari proyek ini diharapkan pada penelitian selanjutnya dapat dilakukan Na*med Entity Recognition* pada jenis entitas yang lebih beragam serta diharapkan sistem dapat melakukan *entity-tagging* dengan lebih baik.

DAFTAR PUSTAKA

[1]	A. WILLYAWAN, "NAMED ENTITY RECOGNITION (NER) BAHASA INDONESIA," p. 54, 2018.
[2]	M. M. d. C. G. Andrei Mikheev, "Named Entity Recognition without Gazetteers," p. 8, 1999.