LAPORAN AKHIR PROYEK

Indonesian Named-entity Recognition using spaCy

Disusun oleh:

1. 12S17028 – Andri Reimondo Tamba
2. 12S18006 – Mei Kristina Panjaitan
3. 12S18013 – Yudika Purba
4. 12S18021 – Lastri Sari Naomi Marbun

11S4037 – PEMROSESAN BAHASA ALAMI

FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO

INSTITUT TEKNOLOGI DEL

2021

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

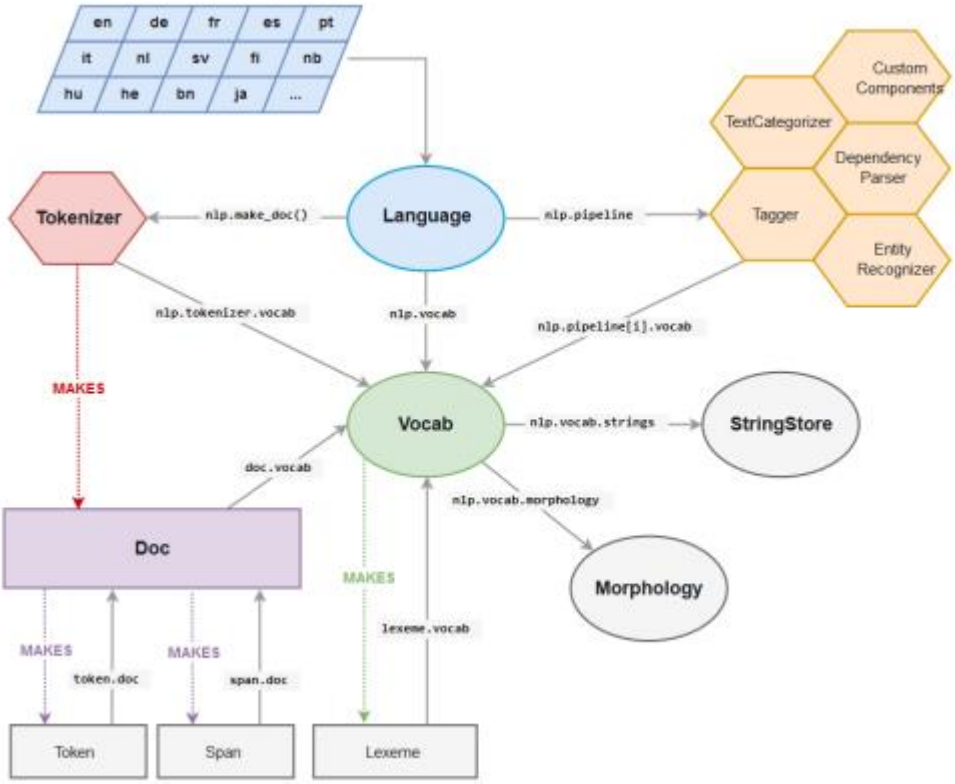
BAB 1
PENDAHULUAN

Bagian ini berisi latar belakang, tujuan, manfaat, ruang lingkup. Tim pengembang, serta kepentingan utama dalam pengerjaan Proyek dalam mata kuliah PBA (Pemrosesan Bahasa Alami) dalam topic Indonesian Named-entity Recognition

1.1 Latar Belakang

Named-entity Recognition merupakan bagian dari riset Natural Language Processing (NLP) yang digunakan untuk mengekstrak informasi serta nama orang, organisasi, lokasi dan waktu. Dengan tujuan misalkan untuk memperoleh informasi dalam dokumen teks secara manual, manusia harus membaca seluruh isi dokumen, apabila dokumen teks memiliki teks panjang diperlukannya waktu sangat lama dalam mendapatkan informasi tersebut dengan itu Named-entity Recognition ini memperoleh informasi dengan waktu yang singkat. [1]

Dalam pendekatan Named-entity Recognition dalam proyek ini menggunakan entity tagging namun ada kekurangan yang dimiliki yaitu pada proses automatic tagging dimana tidak dapat melakukan tag entitas pada kata serta frasa yang dimiliki entitas, untuk meningkatkan kemampuan automatic tagging dengan menggunakan *library spaCy* dimana spacy ini untuk mengembangkan dan men-training NER untuk memproses data language mengenali berbagai entitas bernama atau numerik, termasuk orang, organisasi, bahasa, acara, dll. Selain entitas default ini, juga dapat menambahkan kelas arbitrary ke model NER, dengan melatih (training) model untuk memperbaikinya dengan contoh terlatih (trained) yang lebih baru. Berikut Model Architecture SpaCy: [2]



Gambar 1. Spacy Library Architecture

1.2 Tujuan

Tujuan dari penelitian ini adalah :

a. Membangun data set dengan menggunakan library spacy
b. Membuat model dependency parser dan named entity recognition menggunakan library spacy bahasa indonesia
c. mengevaluasi model dependency parser dan named entity recognition

1.3 Manfaat

Adapun yang menjadi manfaat proyek adalah sebagai berikut:

a. Penulis dapat menambah wawasan dan pengalaman secara langsung dalam

membangun sebuah model untuk untuk mengekstrak informasi serta nama orang, organisasi, lokasi dan waktu.

b. Sebagai referensi, jika ingin dilakukan pengembangan terhadap proyek.

1.4 Ruang Lingkup

Ruang lingkup dalam pengerjaan proyek ini adalah menggunakan metode library spacy dengan data sampel yang diperoleh dari github.com.

BAB 2
ISI

Pada bab ini dijelaskan analisis yang dilakukan terhadap data dan metode, desain pemrosesan bahasa alami yang ditampilkan dalam bentuk *flowchart* atau diagram alir, implementasi berupa kode program dan cuplikan hasil, dan hasil evaluasi kuantitatif terhadap implementasi pemrosesan bahasa alami yang dilakukan.

2.1 Analisis

Pada subbab ini dijelaskan analisis yang dilakukan terhadap data dan metode yang digunakan dalam mengimplementasikan pemrosesan bahasa alami.

2.1.1 Analisis Data

Dataset yang digunakan dalam proyek ini diperoleh dari link berikut: https://github.com/ialfina/ner-dataset-modified-dee/blob/master/singgalang/SINGGALANG.tsv *dataset* yang digunakan adalah SINGGALANG.csv dimana data terdiri dari 1478268 *rows* dan 2 *columns*. *Dataset* SINGGALANG.csv memiliki 2 column yang berisi data *token_0* dan *BIO_tags_0*.

Tabel 1. Atribut pada Dataset

No.	Nama atribut	Tipe Atribut	Deskripsi
1.	*token_0*	object	Berisi teks isi dataset
2.	*BIO_tag_0*	object	Entitas dari teks pada dataset

Berikut adalah distribusi *entitas* teks pada *dataset* SINGGALANG.tsv ditunjukkan pada Gambar 1. Dapat dilihat bahwa entitas Person (96.5%) lebih banyak daripada entitas Place, Organisation dan O (other) atau entitas lainnya.

```
['IamenjabatsebagaiPresidenketigaMesirpadaperiode15Oktober1970hinggaterbunuhnyapada6Oktober1981.OlehduniaBaratiadianggapsebagai
orangyangsangatberpengaruhdiMesirdandiTimurTengahdalamsejarahmodern.SadatdilahirkandiMitAbuAl-KumAl-MinufiyahMesirdalamsebuahke
luargaMesir-sudanyangmiskindengan12saudaralaki-lakidanperempuan.IalulusdariAkademiMiliterKerajaandiKairopada1938danditempatkand
iKorpsIsyarat.IabergabungdenganGerakanPerwiraBebasyangbertekaduntukmembebaskanMesirdarikekuasaanBritaniaRaya.',
 {'entities': [(31, 36, 'place'),
   (153, 158, 'place'),
   (193, 198, 'person'),
   (234, 239, 'place'),
   (341, 349, 'place'),
   (351, 356, 'place'),
   (457, 462, 'place'),
   (475, 483, 'place'),
   (483, 487, 'place')]}]
```

```
['IamenjabatsebagaiPresidenketigaMesirpadaperiode15Oktober1970hinggaterbunuh
nyapada6Oktober1981.OlehduniaBaratiadianggapsebagaiorangyangsangatberpengaru
hdiMesirdandiTimurTengahdalamsejarahmodern.SadatdilahirkandiMitAbuAl-KumAl-M
inufiyahMesirdalamsebuahkeluargaMesir-sudanyangmiskindengan12saudaralaki-lak
idanperempuan.IalulusdariAkademiMiliterKerajaandiKairopada1938danditempatkan
diKorpsIsyarat.IabergabungdenganGerakanPerwiraBebasyangbertekaduntukmembebas
kanMesirdarikekuasaanBritaniaRaya.',
 {'entities': [(31, 36, 'place'),
   (153, 158, 'place'),
   (193, 198, 'person'),
   (234, 239, 'place'),
   (341, 349, 'place'),
   (351, 356, 'place'),
   (457, 462, 'place'),
   (475, 483, 'place'),
   (483, 487, 'place')]}]
```

Gambar 2. Persentase entitas data PERSON pada Dataset

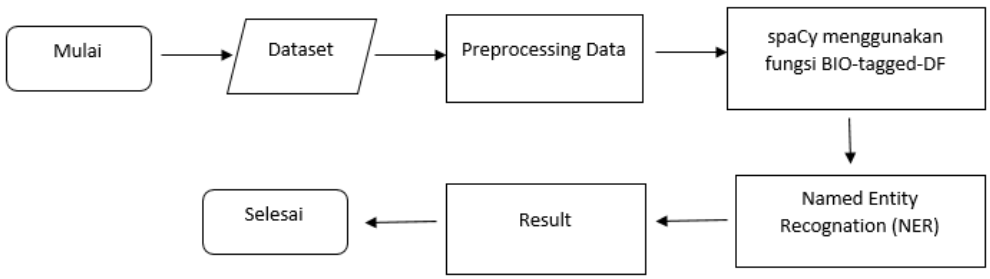
2.1.2 Analisis Metode

*(NER) Natural Language Processin*g (NLP) merupakan salah satu cabang ilmu AI yang berfokus pada pengolahan bahasa natural dengan pendekatan komputerisasi yang digunakan untuk menganalisis data, teks, ucapan, dan lain-lain. Oleh karena itu NLP dapat melakukan pengolahan bahasa alami seperti bahasa manusia pada umumnya. Spacy merupakan sebuah library bahasa pemrograman Python. Library spacy merupakan sebuah library *natural language process*ing (NLP) open source yang bisa dipergunakan untuk membuat sebuah sistem NLP (*Natural Language Processing*). Salah satu fasilitas yang dimiliki *spacy* adalah model Text Classification yang dapat mendeteksi kalimat berdasarkan sentimen negatif, positif dan netral. Untuk menjalankan library ini dibutuhkan dataset Bahasa Indonesia yang diolah menggunakan fasilitas dari *Spacy*. spaCy Named entity recognition akan mengklasifikasikan token dengan kategori seperti person names, organization, locations, medical codes, time expression, quantities, monetary values dan lainnya. Model spacy sudah tersedia dalam beberapa bahasa diantaranya Bahasa Inggris, Bahasa Jerman, Bahasa Spanyol, Bahasa Portugis, Bahasa Prancis, Bahasa Italia, dan Bahasa Belanda. Untuk pengembangan Text Classification Bahasa Indonesia sejauh ini belum ada dikembangkan melalui *Library Spacy*. Dalam melakukan analisis mengelompokkan entitas data teks diperlukan metode. Dalam proyek ini untuk mengelompokkan dan memberikan label entitas pada teks entitas *Person, Organisation, Place, GPE* dan *O (Other*) dengan menggunakan model spacy bahasa indonesia yaitu model menggunakan BIO-tagged-DF kemudian model akan memberikan label pada teks yang telah diinput sebelumnya.

SpaCy adalah pustaka sumber terbuka (open source library) untuk Natural Language Processing tingkat lanjut yang ditulis dengan Python. Ini dapat digunakan untuk membangun ekstraksi informasi atau sistem Natural Language Processing atau untuk memproses teks untuk pembelajaran yang mendalam. SpaCy menampilkan sistem pengenalan entitas statistik yang sangat cepat, yang memberikan label ke rentang token yang berdekatan. SpaCy berfokus pada menyelesaikan sesuatu daripada lebih pendekatan akademis, SpaCy disertai dengan satu algoritme Parts-of-Speech dan Named-entity Recognition. Ini juga berarti bahwa paket tersebut tidak dipenuhi dengan fitur yang tidak perlu.

2.2 Desain

Pada sub bab ini dijelaskan desain pemrosesan bahasa alami yaitu menggunakan metode spaCy pada teks yang ditampilkan dalam bentuk *flowchart* atau diagram alir seperti ditunjukkan pada Gambar 3.



Gambar 3. Desain model spaCy NER menggunakan BIO-tagged-DF

2.2.1 Data Preprocessing

Data Preprocessing merupakan tahap pemrosesan awal yang dilakukan sebelum teks (*dataset*) diolah ke tahap selanjutnya. Data yang diperoleh masih dalam format yang tidak terstruktur, dimana masih terdapat *noise* pada *dataset* tersebut. Selain itu, data masih dalam format *raw data* sehingga tidak memungkinkan untuk melakukan analisis pada *raw data*. Oleh karena itu perlu dilakukan *data preprocessing* untuk menghilangkan kata-kata pada teks atau dokumen yang mengandung beberapa format yang keberadaannya tidak penting dalam *text mining* [1].

2.2.1.1 Tokenization

Tujuan dari tahapan ini adalah untuk memecah teks menjadi unit yang relevan, seperti *words,* simbol, *phrases,* atau elemen bermakna lainnya yang disebut juga dengan *token*. Pada tahap *tokenization,* setiap kalimat akan dipisahkan per kata berdasarkan tanda spasi antar kalimat. Setelah setiap kata dipisahkan, maka tanda baca akan dihapus. Contoh *tokenization* adalah sebagai berikut.

Tabel 2. Tokenization

Text	***Tokenization***
Ia menjabat sebagai Presiden ketiga Mesir	ia, menjabat, sebagai, presiden, ketiga, mesir

2.3 Implementasi

Pada sub bab ini dijelaskan mengimplementasikan pemrosesan bahasa alami, yaitu *Indonesian Named-entity Recognition* menggunakan *SpaCy*.

Pada bagian ini akan dibahas *data preprocessing* yang dilakukan sebelum digunakan dalam pemodelan, mencakup *tokenization*.

Pada dataset dibawah ini, kita ketahui bahwa “Mesir” diberikan tag sebagai place atau tempat, dan dibawahnya ada organization dan person. Tujuan kami adalah membuat data dari dataset dibawah ini akan dibuat suatu model, supaya jika diberikan text data, misalnya “Saya pergi ke Mesir” untuk memunculkan nama entitas dari keterangan dari tag nya. Mesir itu berposisi sebagai keterangan tempat.

```
df.head(25)
```

	token_0	BIO_tag_0
0	Ia	O
1	menjabat	O
2	sebagai	O
3	Presiden	O
4	ketiga	O
5	Mesir	b-place
6	pada	O
7	periode	O
8	15	O
9	Oktober	O
10	1970	O
11	hingga	O
12	terbunuhnya	O
13	pada	O
14	6	O
15	Oktober	O
16	1981	O
17	.	O
18	Oleh	O
19	dunia	O
20	Barat	O
21	ia	O
22	dianggap	O
23	sebagai	O
24	orang	O

Gambar 4. Tampilan df.head

Pertama dilakukan Import Data seperti gambar di bawah ini.

```
#mengimport library yang diperlukan untuk membaca data
import pandas as pd
import numpy as np
df = pd.read_csv("SINGGALANG_.csv", sep = ';')
```

Gambar 5. Tampilan Code Import Data

Selanjutnya melakukan perintah df.head, dimana kegunaan fungsi head pada pandas adalah untuk mendapatkan n baris data teratas.

```
df.head(25)
```

	token_0	BIO_tag_0
0	Ia	O
1	menjabat	O
2	sebagai	O
3	Presiden	O
4	ketiga	O
5	Mesir	b-place
6	pada	O
7	periode	O
8	15	O
9	Oktober	O
10	1970	O
11	hingga	O
12	terbunuhnya	O
13	pada	O
14	6	O
15	Oktober	O
16	1981	O
17	.	O
18	Oleh	O
19	dunia	O
20	Barat	O
21	ia	O
22	dianggap	O
23	sebagai	O
24	orang	O

Jadi dilakukan pemeriksaan terhadap tagnya, dimana indeks pertama sampai indeks seterusnya.

Kemudian dilakukan import library, seperti gambar di bawah ini.

```
# import library untuk preprocessing data
from nltk.tokenize import word_tokenize
import itertools
import re
import pandas as pd
import numpy as np
import string
from tqdm import tqdm
```

Gambar 6. Tampilan code import library untuk import data

1. Fungsi dibawah ini digunakan untuk mengubah BIO-tagged-DF kedalam format SpaCy Annotations.

Spacy itu memiliki format tersendiri untuk teksnya. Format yang digunakan untuk membuat annotationnya harus dalam bentuk seperti ini yang harus diwajibkan jika menggunakan spacy.

Tadi pada bagian df.head dilakukan pemeriksaan terhadap tagnya, dimana indeks pertama sampai indeks seterusnya menggunakan code di bawah ini, degan tujuan untuk mengetahui kata itu ditempatkan di urutan ke berapa. Jadi dari dataset, diketahui, bahwa dia berada di indeks pertama, “ia menjabat sebagai presiden,,,” di bagian df.head tadi. Jadi inti dari code dibawah ini adalah untuk mengurutkan data dari token token ini, supaya bisa menghasiilkan kata berurutan seperti pada hasil Training.

```
    # mengisi entitas dengan non 'O' annotations
    for row in range(len(dictTemp['token'])):
        if dictTemp['BIO_tag'][row] != 'o':
            entities.append((dictTemp['start_idx'][row],
                             dictTemp['end_idx'][row],
                             dictTemp['BIO_tag'][row]))


    start = []
    end = []
    BIO = []
    i = 0
    while i < len(entities):
        try:
            if entities[i][2][2:] == entities[i+1][2][2:]:
                if entities[i][2][0] is 'b':
#                     print('start1', entities[i][0])
                    start.append(entities[i][0])
                    i += 1
                    if entities[i][2][0] is 'e':
#                         print('end1a', entities[i][1])
                        end.append(entities[i][1])
                        BIO.append(entities[i][2][2:])
                        i += 1
                        continue
                    elif entities[i][2][0] is 'i':
                        for j in range(i, len(entities)):
                            if entities[j][2][0] is not 'e' and j < len(entities)-1:
#                                 print('sana', entities[j])
                                continue
                            elif entities[j][2][0] is 'e':
#                                 print('end1b', entities[j][1])
                                end.append(entities[j][1])
                                BIO.append(entities[j][2][2:])
                                i = j+1
                                break
```

Gambar 7. Tampilan code mengisi entitas dengan non 'O' annotations

```
                            else:
                                assert 1 == 0, \
                                    "Something error in the BIO-tag you wrote. Error BIO tag: '{}'" \
                                    .format(entities[j][2])
                    elif entities[i][2][0] is 'b':
#                         print('end1b', entities[i-1][1])
                        end.append(entities[i-1][1])
                        BIO.append(entities[i-1][2][2:])
                        continue

#                         print('ss',i,j)
            else:
#                 print('start2a', entities[i][0], i)
                start.append(entities[i][0])
#                 print('end2a', entities[i][1], i)
                end.append(entities[i][1])
                BIO.append(entities[i][2][2:])
                i += 1
        except IndexError:
#             print('start2b', entities[i][0], i)
            start.append(entities[i][0])
#             print('end2b', entities[i][1], i)
            end.append(entities[i][1])
            BIO.append(entities[i][2][2:])
            i += 1

    enti['entities'] = [(i,j,k) for i,j,k in zip(start, end, BIO)]
    return [text, enti]
```

2. Digunakan annotations, dimana adalah setiap tag yang digunakan untuk memberi keterangan pada setiap entity

Code di bawah ini menunjukkan annotationnya, dimana anotasi ini adalah keterangan dari setiap entity nya. Ada oragnization, person dan place.

```
# annotations adalah setiap tag yang digunakan untuk memberi keterangan pada setiap entity
annotations = sorted(['organisation',
                      'person',
                      'place',
                      ])
annotations
```

Gambar 8. Tampilan Code Annotation

Code di bawah ini digunakan untuk menjalankan code spacy format tadi. Jadi data frame nya di convert dan disertakan anotationnya.

```
train_data = convert_to_spaCyformat(df, annotations)
```

Gambar 9. Tampilan Code Train Data

3. Training

Setelah menggunakan format yang telah ditentukan oleh spacy, maka hasilnya adalah seperti ini.Selain itu harus dilakukan kombinasi, teks pertama tagnya apa, teks kedua, dst. Supaya kiya bisa mengetahui urutan indeks dari teks. Misalnya, Mesir berada pada indeks 31,36 dengan nama entitas Place. Fungsi di bawah ini berfungsi untuk mengubah data dari bentuk csv menjadi bentuk spacy format annotation supaya bisa digunakan untuk melatih model.

```
# data yang digunakan untuk training data
train_data
```

```
['IamenjabatsebagaiPresidenketigaMesirpadaperiode15Oktober1970hinggaterbunuhnyapada6Oktober1981.OlehduniaBaratiadianggapsebagai
orangyangsangatberpengaruhdiMesirdandiTimurTengahdalamsejarahmodern.SadatdilahirkandiMitAbuAl-KumAl-MinufiyahMesirdalamsebuahke
luargaMesir-sudanyangmiskindengan12saudaralaki-lakidanperempuan.IalulusdariAkademiMiliterKerajaandiKairopada1938danditempatkand
iKorpsIsyarat.IabergabungdenganGerakanPerwiraBebasyangbertekaduntukmembebaskanMesirdarikekuasaanBritaniaRaya.',
 {'entities': [(31, 36, 'place'),
   (153, 158, 'place'),
   (193, 198, 'person'),
   (234, 239, 'place'),
   (341, 349, 'place'),
   (351, 356, 'place'),
   (457, 462, 'place'),
   (475, 483, 'place'),
   (483, 487, 'place')]}]
```

Gambar 10. Tampilan code data yang digunakan untuk training data

4. Membangun Model

Membangun model Named Entity Recognition nya menggunakan spacy.

Berikut adalah library yang digunakan :

```
# import library yang digunakan untuk membuat model Named Entity Recognition
import spacy
from spacy import displacy
from spacy.util import minibatch, compounding
from spacy.util import decaying
from thinc.optimizers import Adam
import random
from matplotlib import pyplot as plt
```

Gambar 11. Tampilan Code library yang digunakan untuk membangun NER

Optimizer digunakan untuk meng-custom model dengan tujuan mendapatkan hasil yang lebih baik.

5. Selanjutnya fungsi ini digunakan untuk membuat optimizer pada pelatihan model

Dengan tujuan mengoptimalkan spacy nya tadi, untuk memperoleh model yang terbaik untuk NER nya. Selajutnya dilakukan train pada data, jika sudah dibuat parameter train pada model nya.

```
# fungsi ini digunakan untuk membuat optimizer pada pelatihan model
def custom_optimizer(optimizer, learn_rate=0.0001, beta1=0.9, beta2=0.999, eps=1e-8, L2=1e-6, max_grad_norm=1.0):
    """
    Function to customizer spaCy default optimizer
    """

    optimizer.learn_rate = learn_rate
    optimizer.eps = eps
    optimizer.L2 = L2

    return optimizer

def train_spacy(data,
                iterations,
                learn_rate=0.001,
                beta1=0.9,
                beta2=0.999,
                eps=1e-8,
                L2=1e-4,
                max_grad_norm=1.0):
    """Load the model, set up the pipeline and train the entity recognizer."""

    TRAIN_DATA = data
    nlp = spacy.blank('id')  # create blank Language class
    # create the built-in pipeline components and add them to the pipeline
    # nlp.create_pipe works for built-ins that are registered with spaCy
    if 'ner' not in nlp.pipe_names:
        ner = nlp.add_pipe('ner')


    # add labels
    for _, annotations in TRAIN_DATA:
         for ent in annotations.get('entities'):
            ner.add_label(ent[2])

    # get names of other pipes to disable them during training
    other_pipes = [pipe for pipe in nlp.pipe_names if pipe != 'ner']
    with nlp.disable_pipes(*other_pipes):  # only train NER
```

Gambar 12. Tampilan Code fungsi ini digunakan untuk membuat optimizer pada pelatihan model

```
        # additional lines
        optimizer = nlp.begin_training()
        optimizer = custom_optimizer(optimizer, learn_rate=learn_rate)
        # Define decaying dropout
#         dropout = decaying(0.8, 0.2, 1e-6)

#         optimizer = nlp.begin_training()
        loss_list = []
        for itn in range(iterations):
            print("Starting iteration " + str(itn))
            random.shuffle(TRAIN_DATA)
            losses = {}

            # batch up the examples using spaCy's minibatch
            batches = minibatch(TRAIN_DATA, size=compounding(4.0, 64.0, 1.001))

            for batch in batches:
                texts, annotations = zip(*batch)
                example = []
                # Update the model with iterating each text
                for i in range(len(texts)):
                    doc = nlp.make_doc(texts[i])
                    example.append(Example.from_dict(doc, annotations[i]))

                # Update the model
                nlp.update(example, drop=0.5, losses=losses)

#             for text, annotations in TRAIN_DATA:
#                 nlp.update(
#                     [text],  # batch of texts
#                     [annotations],  # batch of annotations
#                     drop=next(dropout),  # dropout - make it harder to memorise data
#                     sgd=optimizer,  # callable to update weights
#                     losses=losses)

            print(losses)
            loss_list.append(losses)

    return nlp, loss_list
```

Pada training model ini, disini kami menggunakan 50 iterasi, maksudnya dilakukan perulangan sampai 50 kali dengan tujuan untuk mendapatkan model terbaik.

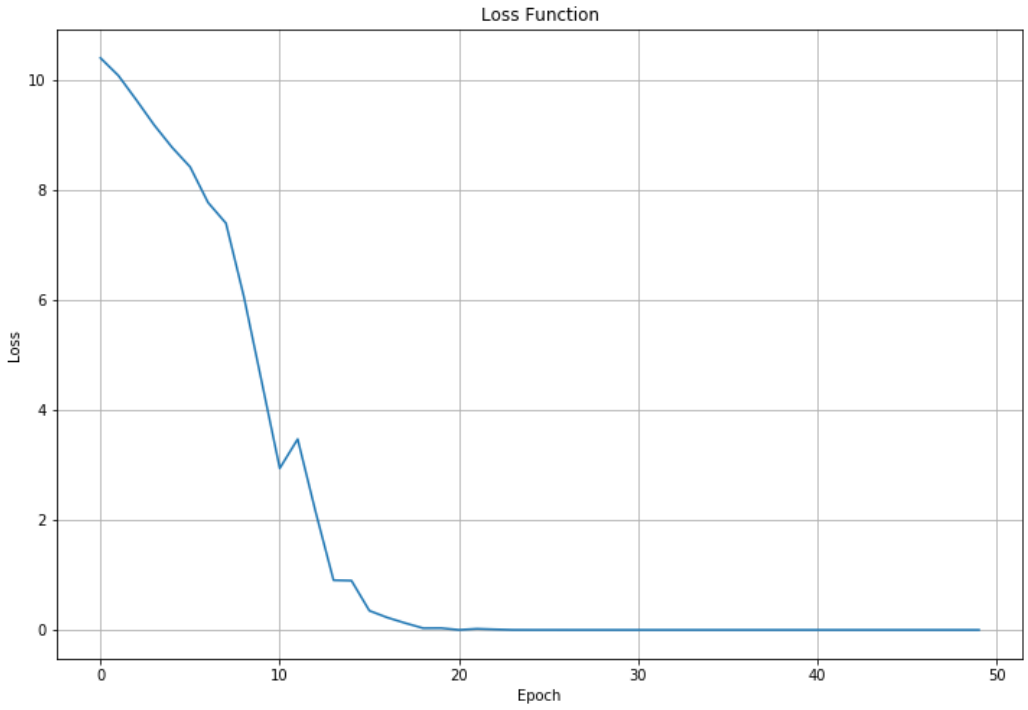
6. Evaluasi

Mencari model terbaik itu ada di epoch berapa dan loss berapa.

```
from matplotlib import pyplot as plt
```

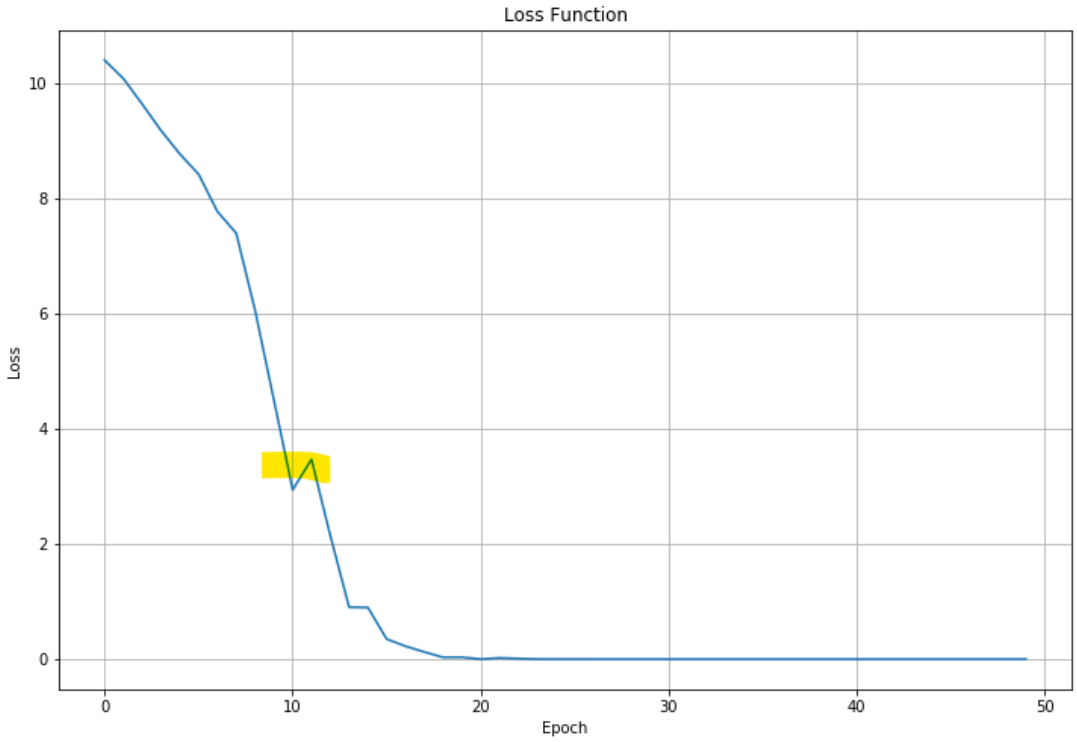
```
# hasil visualisasi training model
plt.figure(figsize=(12,8))
plt.plot([i['ner'] for i in loss])
plt.grid()
plt.title('Loss Function')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.show()
```

Gambar 13. Tampilan Code hasil visualisasi training model



Gambar 14. Tampilan hasil visualisasi training model

Gambar di atas merupakan patahan.



Pada posisi seperti di ataslah, model yang kita train tadi optimal, yaitu berada pada epoch 10-20 dan pada data loss pada k2 sampai 4.

7. Jika hasil evaluasinya sudah muncul, maka sudah bisa disimpan

```
# menyimpan model yang telah dibuat
# nama model adalah Model_NER
modelfile = input("Enter your Model Name: ")
model.to_disk(modelfile)
```

Gambar 15. Tampilan Code untuk menyimpan model

```
# import model untuk menampilkan hasil Named Entity Recognition
from spacy import load, displacy
```

```
link_to_model = "Model_NER"
loaded_model = load(link_to_model)
```

Gambar 16. Tampilan Code untuk meload model

Setelah di load,selanjutnya kita test.

```
# Test text yang digunakan adalah potongan kata berikut
test_text = """Ia menjabat sebagai Presiden ketiga Mesir"""
```

Gambar 17. Tampilan Code untuk melakukan test

Untuk dispacy, sama seperti tadi, harus diubah ke dalam notasi yang ditentukan Spacy. Dikarenakan kita tidak menggunakan data frame csv, namun hanya teks. Oleh karena itu, kita hanya membutuhkan library dispacy dibawah ini maka kita render dalam bentuk entitas supaya dibagi bagi.

```
In [31]: # Membagi setiap kata dalam test text kedalam bentuk entity dengan menggunakan displacy
         doc = loaded_model(test_text)
         displacy.render(doc, style="ent")
```

Ia menjabat sebagai Presiden ketiga Mesir

Gambar 18. Tampilan Code untuk membagi kata

2.4 Hasil

Pada subbab ini dijelaskan evaluasi kuantitatif berdasarkan implementasi pemrosesan bahasa alami, yaitu Indonesian Named-entity Recognition menggunakan spaCy.

Hasil teksnya kita lihat dari model nya tadi yang sudah di prediksi, sehingga muncullah, bahwa entitas yang punya tag itu hanya Mesir yaitu “Place”.

Pada hasil yang diperoleh, kami melakukan test pada teks “Ia menjabat sebagai presiden ketiga Mesir”.

Hasilnya menunjukkan bahwa :

1. Teks “Ia menjabat sebagai presiden ketiga” tidak mempunyai tag.

2. Mesir memiliki tag place

Jadi intinya, yang dimunculkan hanya token atau entitas yang punya anotasi di antara oragnization, person dan place seperti yang telah disebutkan diatas.

Dimana 31 36 menyatakan indeks Mesir di dalam dataset yang tadi.

Gambar 19. Tampilan Code yang menampilkan hasil Test

Sehingga tujuan pada proyek ini sudah tercapai

BAB 3
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 2. Pembagian Tugas dan Tanggung Jawab .

Name	*Role*	*Task*
Andri Reimondo Tamba	*Data Analyst*	Berperan dalam mengumpulkan, mengidentifikasi, menafsirkan serta menganalisis data, model, dan strategi yang efisien untuk digunakan dalam pengerjaan proyek.
	Programmer	Berperan dalam mengimplementasikan code untuk membangun sistem dan melakukan pengujian terhadap sistem yang sudah dibangun.
Mei Kristina Panjaitan	*Data Analyst*	Berperan dalam mengumpulkan, mengidentifikasi, menafsirkan serta menganalisis data, model, dan strategi yang efisien untuk digunakan dalam pengerjaan proyek.
	Programmer	Berperan dalam mengimplementasikan code untuk membangun sistem dan melakukan pengujian terhadap sistem yang sudah dibangun.
Yudika Purba	*Data Analyst*	Berperan dalam mengumpulkan, mengidentifikasi, menafsirkan serta menganalisis data, model, dan strategi yang efisien untuk digunakan dalam pengerjaan proyek.

	Programmer	Berperan dalam mengimplementasikan *code* untuk membangun sistem dan melakukan pengujian terhadap sistem yang sudah dibangun.

Lastri Sari Naomi Marbun	*Data Analyst*	Berperan dalam mengumpulkan, mengidentifikasi, menafsirkan serta menganalisis data, model, dan strategi yang efisien untuk digunakan dalam pengerjaan proyek.
	Programmer	Berperan dalam mengimplementasikan *code* untuk membangun sistem dan melakukan pengujian terhadap sistem yang sudah dibangun.

3.2 Kesimpulan

Dengan menggunakan Spacy ini disimpulkan bahwa dalam metode ini dapat mempermudah dalam pemrosesan data yang menyelesaikan berbagai tugas yang sangat luas untuk mengidentifikasi jenis kata dan mengekstraksi entitas bernama hingga membuat model anda sendiri untuk dianalisis . Pada proses Name entity recognition dalam pengenalanan entitas waktu dalam kejadian pengambilan atau pengenalan entitas, kemudian dengan Library spacy untuk named entity recognition (NER) pada pendekatan rule based method mampu mengekstrak informasi.

3.3 Saran

Adapun saran untuk pengembangan penelitian berikutnya adalah sebagai berikut:

1. Menggunakan metode NLP (Natural Language Processing) lainnya pada proses Named Entity Recognition (NER) untuk mendapatkan ekstraksi informasi dan klasifikasi sebagai perbandingan.

2. Menambahkan dan menggunakan data yang lebih banyak sehingga mempengaruhi hasil dari evaluasi akhir.

3. Melakukan perubahan pada parameter yang akan diukur dalam penelitian lain.

4. Menerapkan metode lain dengan memanfaatkan Library Python yang lain untuk dapat memvisualisasikan data geografis berupa peta.

DAFTAR PUSTAKA

[1] *NAMED ENTITY RECOGNITION (NER) BAHASA INDONESIA MENGGUNAKAN CONDITIONAL RANDOM FIELD DAN POS-TAGGING,2018*

[2] *EKSTRAKSI INFORMASI BERITA ONLINE DENGAN NAMED ENTITY RECOGNITION (NER) DAN RULE-BASED UNTUK VISUALISASI PENYAKIT TROPIS DI INDONESIA*