

# **LAPORAN AKHIR**

## ***Indonesian Named-entity Recognition using Conditional Random Fields (CRF)***



### **Pengusul:**

1. 12S17009 – Prince Ephraim Prabowo Silaban
2. 12S17055 – Joshua Koko Sarasi Manurung
3. 12S18001 – Cindy Angelia Siregar
4. 12S18003 – Citra Hutajulu
5. 12S18017 – Putri Yohana Panjaitan
6. 12S18035 – Angelina Naomi Christina Sinaga

**11S4037 – PEMROSESAN BAHASA ALAMI  
FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO  
INSTITUT TEKNOLOGI DEL  
2021**

# DAFTAR ISI

DAFTAR GAMBAR.....	3
DAFTAR TABEL .....	4
BAB 1 PENDAHULUAN.....	5
1.1 Latar Belakang.....	5
1.2 Tujuan.....	6
1.3 Ruang Lingkup.....	6
BAB 2 ISI .....	7
2.1 Analisis .....	7
2.1.1 Analisis Data .....	7
2.1.2 Analisis Metode.....	7
2.2 Desain .....	8
2.2.1 <i>Text Preprocessing</i> .....	8
2.2.2 <i>Feature Extraction</i> .....	9
2.2.3 <i>Evaluation and Results</i> .....	9
2.3 Implementasi .....	10
2.3.1 <i>Text Preprocessing</i> .....	10
2.3.2 <i>Case Folding</i> .....	10
2.4 <i>Feature Extraction</i> .....	11
2.6 <i>Modeling</i> .....	13
2.6.1 <i>Accuracy CRF Model</i> .....	14
BAB 3 PENUTUP .....	15
3.1 Pembagian Tugas dan Tanggung Jawab.....	15
3.2 Kesimpulan .....	16
3.3    Saran .....	17
DAFTAR PUSTAKA.....	18

## DAFTAR GAMBAR

Gambar 1. Analisis Metode .....	8
Gambar 2. Confusion Matrix .....	9
Gambar 3. Kode Program Case Folding .....	11
Gambar 4. Kode Program Feature Extraction.....	12
Gambar 5. Kode Program Hasil.....	13
Gambar 6. Kode Program Modeling.....	13
Gambar 7. Kode Program Accuracy CRF Model .....	14

## DAFTAR TABEL

Table 1. Case Folding .....	8
Table 2. Contoh Case Folding .....	11
Table 3. Pembagian Tugas dan Tanggung Jawab .....	15

# BAB 1

## PENDAHULUAN

Bab pendahuluan berisi penjelasan terkait latar belakang, tujuan, manfaat, dan ruang lingkup pengerjaan proyek.

### 1.1 Latar Belakang

Pada dokumen teks mengandung banyak informasi seperti nama orang, nama organisasi dan nama tempat. Pada umumnya untuk memperoleh informasi dari dalam dokumen teks secara manual dengan cara membaca seluruh isi dokumen. Apabila dokumen teks tersebut sangat panjang, maka dibutuhkan waktu yang lama untuk memperoleh informasi yang terdapat dalam dokumen tersebut. Oleh karena itu *Named Entity Recognition* dapat digunakan untuk memperoleh informasi pada dokumen teks secara otomatis dan dapat dilakukan dalam waktu yang singkat [1]. Pengenalan entitas bernama (*Named Entity Recognition* atau NER) merupakan salah satu bentuk pemrosesan bahasa alami (*Natural Language Processing* atau NLP), subbidang kecerdasan buatan (*Artificial Intelligence* atau AI) yang dapat digunakan untuk memperoleh entitas bernama seperti nama orang, organisasi dan tempat pada sebuah dokumen teks secara otomatis [2]. *Named entity recognition* (NER) diterapkan pada identifikasi entitas dalam artikel berita, algoritma pencarian agar berjalan lebih efektif dan efisien, pengembangan rekomendasi konten untuk suatu media, dan penanganan *customer support*.

Selama lima tahun terakhir, penelitian terkait NER sudah banyak dilakukan dalam berbagai metode dan tujuan yang digunakan. Metode yang digunakan dapat dikategorikan dalam dua kategori pendekatan, yaitu pendekatan *rule-based* dan pendekatan *machine learning*. Metode *rule-based* menggunakan aturan yang telah didefinisikan berdasarkan pengetahuan linguistik dengan analisis yang dilakukan pada tingkatan sintaksis dan semantik. Metode tersebut memiliki keterbatasan karena aturan sintaksis dan semantik harus didefinisikan sebanyak mungkin untuk memperoleh hasil yang optimal.

Saat ini, penelitian mengenai *Named Entity Recognition* untuk Bahasa Indonesia telah banyak dilakukan. Pada Tahun 2018, William G. melakukan sebuah penelitian *Named Entity*

*Recognition* Bahasa Indonesia dengan menggunakan algoritma *Deep learning* [4]. Pada penelitian tersebut, data yang digunakan berasal dari DBPedia Indonesia sebagai data berlabel dan artikel berita dari situs berita Indonesia (*kompas.com*, *cnnindonesia.com*, *tempo.com*) sebagai data tidak berlabel dengan menggunakan arsitektur *hybrid bidirectional LSTM (BLSTM)*. Hasil dari penelitian tersebut dapat disimpulkan bahwa jumlah dataset yang digunakan dalam penelitian masih sedikit sehingga hasil eksperimen melaporkan bahwa kinerja Deep Learning masih cukup rendah untuk beberapa label. Oleh karena itu, pada penelitian selanjutnya dapat mengumpulkan dan membangun dataset yang lebih banyak. Dengan memiliki ukuran dataset yang besar, dapat mendukung kinerja yang lebih baik. Untuk penelitian ini juga akan menggunakan metode supervised learning untuk menghasilkan kinerja yang lebih baik.

## **1.2 Tujuan**

Adapun tujuan dari pengerjaan proyek ini adalah untuk meningkatkan performa teks *summarization* menghasilkan model *Named Entity Recognition* Bahasa Indonesia yang digunakan untuk meningkatkan performa sistem pencarian yang terdapat pada kalimat atau teks Bahasa Indonesia. Dan juga untuk mengidentifikasi / mengklasifikasi entitas yang sangat berguna saat ingin mengekstrak informasi.

## **1.3 Ruang Lingkup**

Ruang lingkup yang akan dibahas dalam penelitian ini adalah:

1. Masukan pada model NER berupa Bahasa Indonesia sesuai dengan pedoman Penulisan Umum Ejaan Bahasa Indonesia (PUEBI).
2. Dataset yang digunakan pada proyek bernama dataset SINGGALANG.

## 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

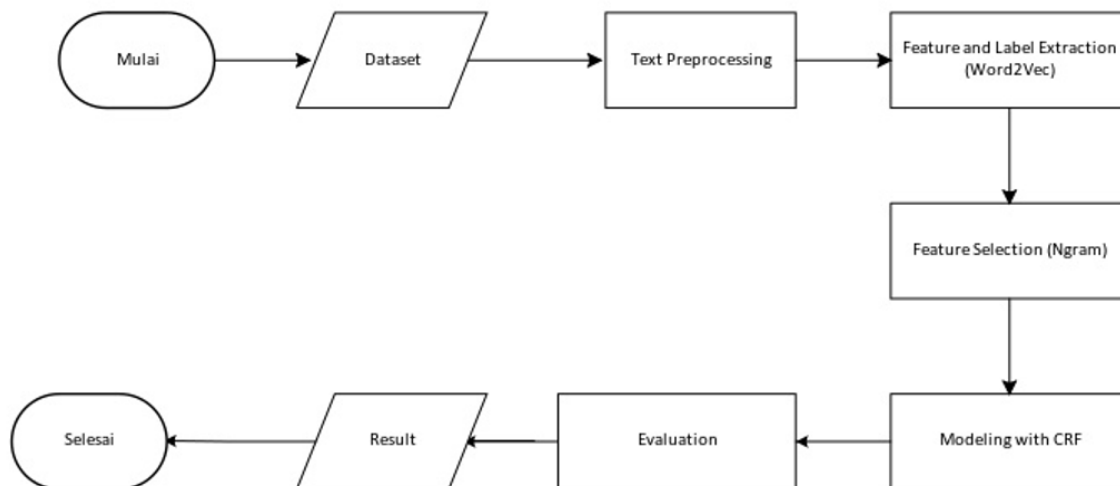
*Dataset* yang digunakan dalam proyek ini diperoleh dari link berikut: <https://github.com/ialfina/ner-dataset-modified-dee/tree/master/singgalang>. *Dataset* yang digunakan adalah SINGGALANG.tsv yang terdiri dari 1.527.224 *records*. *Dataset* tersebut terdiri dari 4 entitas, yaitu *others*, *organization*, *place* dan *people*.

##### 2.1.2 Analisis Metode

Analisis sentimen berkaitan dengan bidang yang lebih luas seperti pengolahan bahasa alami, komputasi linguistik, dan *text mining* untuk tujuan menganalisis, *sentiment*, pendapat, sikap, evaluasi, penilaian dan emosi seseorang berkenaan pada topik, produk, layanan, organisasi, atau kegiatan tertentu [3]. Sentimen adalah perasaan, sikap, evaluasi, atau emosi yang terkait dengan suatu opini. Dalam melakukan analisis *Named Entity Recognition*, diperlukan suatu metode atau klasifikasi. Dalam proyek ini, untuk mengklasifikasikan entitas bernama digunakan metode CRF (*Conditional Random Fields*). CRF memiliki banyak kelebihan dibandingkan model probabilitas lain seperti *Hidden Markov Model* (HMM) dan *Maximum Entropy Markov Model* (MEMM). CRF mengatasi permasalahan ketergantungan asumsi yang tinggi pada HMM. Hal ini karena CRF dapat menentukan sendiri seberapa banyak fitur yang diinginkan untuk membangun sebuah model CRF tidak seperti HMM yang bersifat lokal dimana setiap kata hanya bergantung pada label saat ini dan setiap label sebelumnya [4].

## 2.2 Desain

Pada sub bab ini dijelaskan desain pemrosesan bahasa alami yaitu analisis sentimen pada teks yang ditampilkan dalam bentuk *flowchart* atau diagram alir seperti ditunjukkan pada Gambar X.



Gambar 1. Analisis Metode

### 2.2.1 Text Preprocessing

*Text 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* [5]. 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*.

#### 2.2.1.1 Case Folding

Setelah data dibersihkan, pada tahapan ini dilakukan proses konversi data teks yang terdapat pada *dataset* menjadi ke dalam bentuk *lower case*. Hal ini untuk memudahkan preproses data di tahap selanjutnya. Berikut adalah contoh *case folding*:

Table 1. Case Folding

<i>Text</i>	<i>Case Folding</i>
Ia bergabung dengan Gerakan Perwira Bebas	ia bergabung dengan gerakan perwira bebas



### 2.2.2 Feature Extraction

Pada bagian ini dilakukan *feature extraction*. *Feature extraction* atau ekstraksi fitur merupakan suatu pengambilan ciri (*feature*) dari suatu bentuk yang nantinya nilai yang didapatkan akan dianalisis untuk proses selanjutnya. Berikut adalah kode program yang digunakan untuk melakukan *feature extraction*. Dalam CRF, setiap fungsi fitur adalah fungsi yang menerima input sebuah kalimat, *S* dan posisi *i* dari sebuah kata dalam kalimat.

### 2.2.3 Evaluation and Results

Pada tahapan ini proses evaluasi dari hasil yang didapatkan, dilakukan dengan *confusion matrix*. *Confusion matrix* memberikan informasi perbandingan hasil klasifikasi yang dilakukan model dengan klasifikasi yang sebenarnya.

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	<b>TP</b> (True Positive)	<b>FP</b> (False Positive) Type I Error
	0 (Negative)	<b>FN</b> (False Negative) Type II Error	<b>TN</b> (True Negative)

Gambar 2. Confusion Matrix

Terdapat 4 istilah yang digunakan untuk merepresentasikan hasil klasifikasi, yaitu:

1. *True Positive* (TP) merupakan data positif yang diprediksi benar.
2. *True Negative* (TN) merupakan data negatif yang diprediksi benar.
3. *False Positive* (FP) merupakan data negatif namun diprediksi sebagai data positif.
4. *False Negative* (FN) merupakan data positif namun diprediksi sebagai data negatif.

Untuk mengukur *performance metrics* dari *confusion matrix* dapat dilakukan dengan menghitung nilai *accuracy*, *precision*, *Recall*, dan *F1-Score*.

- (1) *Accuracy* menggambarkan seberapa akurat model dapat mengklasifikasikan dengan benar.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

(2) *Precision* menggambarkan tingkat keakuratan antara data yang diminta dengan hasil prediksi yang diberikan oleh model.

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

(3) *Recall* menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi.

$$Recall = \frac{TP}{TP + FN}$$

## 2.3 Implementasi

Pada subbab ini dijelaskan pengimplementasian pemrosesan bahasa alami, yaitu *indonesian entity recognition* menggunakan *Conditional Random Fields*.

### 2.3.1 Text Preprocessing

Pada bagian ini akan dibahas *text preprocessing* yang dilakukan sebelum digunakan dalam pemodelan, mencakup *teks preprocessing*, *case folding*, *stemming*, dan *lemmatization*. *Text 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 *teks preprocessing* untuk menghilangkan kata-kata pada teks atau dokumen yang mengandung beberapa format yang keberadaannya tidak penting dalam *text mining* [1].

### 2.3.2 Case Folding

Pada tahapan ini dilakukan proses konversi data teks yang terdapat pada *dataset* menjadi ke dalam bentuk *lower case*. Hal ini untuk memudahkan preproses data di tahap selanjutnya. Berikut adalah contoh *case folding*:

```

• Case Folding

In [26]: dataset_prep['token'] = dataset_prep['token'].str.lower()

In [28]: dataset_prep.sample(10)

Out[28]:

```

	token	entitas bernama	kalimat
328862	dalam	O	10879
1264404	yang	O	41804
793267	mencari	O	25999
1350249	tahun	O	44692
1259538	sony	Person	41630
760575	barat	O	24880
941552	desa	O	30996
506721	kebasen	Place	16533
328675	doe	O	10872
664370	memiliki	O	21615

Gambar 3. Kode Program Case Folding

Table 2. Contoh Case Folding

<i>Text</i>	<i>Case Folding</i>
Ia bergabung dengan Gerakan Perwira Bebas	ia bergabung dengan gerakan perwira bebas

## 2.4 Feature Extraction

Pada bagian ini dilakukan *feature extraction*. *Feature extraction* atau ekstraksi fitur merupakan suatu pengambilan ciri (*feature*) dari suatu bentuk yang nantinya nilai yang didapatkan akan dianalisis untuk proses selanjutnya. Berikut adalah kode program yang digunakan untuk melakukan *feature extraction*. Dalam CRF, setiap fungsi fitur adalah fungsi yang menerima input sebuah kalimat, *S* dan posisi *i* dari sebuah kata dalam kalimat.

Baris *pseudocode* untuk proses *feature function* dapat dilihat sebagai berikut:

```
In [40]: # A class to retrieve the sentences from the dataset
class getsentence(object):

    def __init__(self, data):
        self.n_sent = 1.0
        self.data = data
        self.empty = False
        agg_func = lambda s: [(w, t) for w, t in zip(s["token"].values.tolist(),
                                                    s["entitas bernama"].values.tolist())]

        self.grouped = self.data.groupby("kalimat").apply(agg_func)
        self.sentences = [s for s in self.grouped]

In [41]: getter = getsentence(dataset_prep)

In [42]: sentences = getter.sentences
#this is how a sentence will look like.
print(sentences[1])

[('oleh', 'O'), ('dunia', 'O'), ('barat', 'O'), ('ia', 'O'), ('dianggap', 'O'), ('sebagai', 'O'), ('orang', 'O'), ('yang', 'O'), ('sangat', 'O'), ('berpengaruh', 'O'), ('di', 'O'), ('mesin', 'Place'), ('dan', 'O'), ('di', 'O'), ('timur', 'O'), ('tengah', 'O'), ('dalam', 'O'), ('sejarah', 'O'), ('modern', 'O'), ('.', 'O'), ('O')]
```

Gambar 4. Kode Program Feature Extraction

```
In [43]: def word2features(sent, i):
    word = sent[i][0]
    postag = sent[i][1]

    features = {
        'bias': 1.0,
        'word.lower()': word.lower(),
        'word[-3:]': word[-3:],
        'word[-2:]': word[-2:],
        'word.isupper()': word.isupper(),
        'word.istitle()': word.istitle(),
        'word.isdigit()': word.isdigit(),
    }
    if i > 0:
        word1 = sent[i-1][0]
        features.update({
            '-1:word.lower()': word1.lower(),
            '-1:word.istitle()': word1.istitle(),
            '-1:word.isupper()': word1.isupper(),
        })
    else:
        features['BOS'] = True

    if i < len(sent)-1:
        word1 = sent[i+1][0]
        features.update({
            '+1:word.lower()': word1.lower(),
            '+1:word.istitle()': word1.istitle(),
            '+1:word.isupper()': word1.isupper(),
        })
    else:
        features['EOS'] = True

    return features
```

## 2.5 Hasil

Pada sub bab ini dijelaskan evaluasi kuantitatif berdasarkan implementasi pemrosesan bahasa alami menggunakan CRF (*Conditional Random Field*).

```

y_true_combined,
y_pred_combined,
labels = [class_indices[cls] for cls in tagset],
target_names = tagset
), accuracy_score(y_true_combined, y_pred_combined)

In [64]: y_pred = [tagger.tag(xseq) for xseq in X_test]

In [65]: report, accuracy = ner_report(y_test, y_pred)

In [66]: print(report)

```

	precision	recall	f1-score	support
0	0.98	0.99	0.98	134129
Organisation	0.82	0.75	0.78	1503
Person	0.86	0.65	0.74	4078
Place	0.90	0.83	0.86	8125
micro avg	0.97	0.97	0.97	147835
macro avg	0.89	0.81	0.84	147835
weighted avg	0.97	0.97	0.97	147835
samples avg	0.97	0.97	0.97	147835

```

In [67]: print(accuracy)
0.9696418304190483

```

Gambar 5. Kode Program Hasil

## 2.6 Modeling

Pada bagian dilakukan pemodelan terhadap analisis sentimen dengan algoritma CRF.

```

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.1, random_state = 42)

5.Modeling and Evaluation

In [55]: !pip install python-crfsuite
Collecting python-crfsuite
  Downloading python_crfsuite-0.9.7-cp37m-manylinux1_x86_64.whl (743 kB)
    |████████████████████████████████████████| 743 kB 24.9 MB/s eta 0:00:01
Installing collected packages: python-crfsuite
Successfully installed python-crfsuite-0.9.7

In [56]: import pycrfsuite

In [57]: trainer = pycrfsuite.Trainer(verbose = True)
for xseq, yseq in zip(X_train, y_train):
    trainer.append(xseq, yseq)

In [58]: trainer.set_params({
    'c1': 1.0, # coefficient for L1 penalty
    'c2': 1e-3, # coefficient for L2 penalty
    'max_iterations': 100, # stop earlier

    # include transitions that are possible, but not observed
    'feature.possible_transitions': True
})

In [59]: trainer.params()
Out[59]: ['feature.minfreq',
          'feature.possible_states',
          'feature.possible_transitions',
          'c1',
          'c2',

```

Gambar 6. Kode Program Modeling

```
'period',
'delta',
'linesearch',
'max_linesearch']

In [60]: trainer.train('singgalang-ner.crfsuite')

Feature generation
type: CRFid
feature.minfreq: 0.000000
feature.possible_states: 0
feature.possible_transitions: 1
0....1....2....3....4....5....6....7....8....9....10
Number of features: 298811
Seconds required: 2.459

L-BFGS optimization
c1: 1.000000
c2: 0.001000
num_memories: 6
max_iterations: 100
epsilon: 0.000010
stop: 10
delta: 0.000010
linesearch: MoreThuente
linesearch.max_iterations: 20

In [61]: trainer.logparser.last_iteration
Out[61]: {'active_features': 29208,
          'error_norm': 381.285675,
          'feature_norm': 340.704479,
          'linesearch_step': 1.0,
          'linesearch_trials': 1,
          'loss': 106791.92172,
          'num': 100,
          'scores': {},
          'time': 0.899}

In [62]: tagger = pycrfsuite.Tagger()
```

### 2.6.1 Accuracy CRF Model

Pada bagian ini dilakukan pembangunan model CRF. Berikut adalah kode program untuk model CRF.

```
In [64]: y_pred = [tagger.tag(xseq) for xseq in X_test]
```

```
In [65]: report, accuracy = ner_report(y_test, y_pred)
```

```
In [67]: print(accuracy)
```

0.9696418304190483

Gambar 7. Kode Program Accuracy CRF Model

## 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.

*Table 3. Pembagian Tugas dan Tanggung Jawab*

<i>Name</i>	<i>Role</i>	<i>Task</i>
Prince Silaban	<i>Data Analyst</i>	Berperan dalam mengumpulkan, mengidentifikasi, menafsirkan serta menganalisis data, model, dan strategi yang efisien untuk digunakan dalam pengerjaan proyek.
	<i>Programmer</i>	Berperan dalam mengimplementasikan code untuk membangun sistem dan melakukan pengujian terhadap sistem yang sudah dibangun.
Joshua Koko Manurung	<i>Data Analyst</i>	Berperan dalam mengumpulkan, mengidentifikasi, menafsirkan serta menganalisis data, model, dan strategi yang efisien untuk digunakan dalam pengerjaan proyek.
	<i>Programmer</i>	Berperan dalam mengimplementasikan code untuk membangun sistem dan melakukan pengujian terhadap sistem yang sudah dibangun
Cindy Siregar	<i>Data Analyst</i>	Berperan dalam mengumpulkan, mengidentifikasi, menafsirkan serta menganalisis data, model, dan strategi yang efisien untuk digunakan dalam pengerjaan proyek.
	<i>Programmer</i>	Berperan dalam mengimplementasikan

		code untuk membangun sistem dan melakukan pengujian terhadap sistem yang sudah dibangun.
Citra Hutajulu	<i>Data Analyst</i>	Berperan dalam mengumpulkan, mengidentifikasi, menafsirkan serta menganalisis data, model, dan strategi yang efisien untuk digunakan dalam pengerjaan proyek.
	<i>Programmer</i>	Berperan dalam mengimplementasikan code untuk membangun sistem dan melakukan pengujian terhadap sistem yang sudah dibangun
Putri Yohana Panjaitan	<i>Data Analyst</i>	Berperan dalam mengumpulkan, mengidentifikasi, menafsirkan serta menganalisis data, model, dan strategi yang efisien untuk digunakan dalam pengerjaan proyek.
	<i>Programmer</i>	Berperan dalam mengimplementasikan <i>code</i> untuk membangun sistem dan melakukan pengujian terhadap sistem yang sudah dibangun.
Angelina Naomi Sinaga	<i>Data Analyst</i>	Berperan dalam mengumpulkan, mengidentifikasi, menafsirkan serta menganalisis data, model, dan strategi yang efisien untuk digunakan dalam pengerjaan proyek.
	<i>Programmer</i>	Berperan dalam mengimplementasikan code untuk membangun sistem dan melakukan pengujian terhadap sistem yang sudah dibangun

### 3.2 Kesimpulan

Penerapan *Entity Recognition* pada *Indonesian Named-Entity* dengan menerapkan *Conditional Random Fields* (CRF) dapat menganalisis data yang ditampilkan pada *dataset* diimplementasikan menggunakan pemodelan *Conditional Random Field* (CRF). Setelah dievaluasi analisis dengan pemodelan tersebut menghasilkan hasil akurasi secara berurutan 0,96.



Berdasarkan hasil pemodelan tersebut pemodelan yang lebih akurat dan lebih sesuai untuk menganalisis entitas yang diberikan label *organization*, *place* dan *people*, tersebut adalah pemodelan dengan *Conditional Random Field* dengan hasil akurasi 0,96.

### 3.3 Saran

Beberapa saran sekiranya dapat membantu meningkatkan efektivitas dari analisis pada Indonesian *Named Entity Recognition* sebagai berikut:

1. Penambahan jumlah data sekiranya dapat membantu dalam memperbanyak kosa kata dalam *data training*.
2. Pemodelan yang dilakukan dapat dilakukan dengan menggunakan algoritma yang lainnya untuk menghasilkan prediksi analisis data Singgalang dalam *Indonesian Named Entity Recognition* yang lebih akurat, misalnya dengan menggunakan Bi-LSTM, *Maximum Entropy*, dll.

## **DAFTAR PUSTAKA**

- [1] K. K. P. Hadi Permana, "NAMED ENTITY RECOGNITION USING BIDIRECTIONAL LSTMCRF METHODS IN INDONESIAN TEXT," no. 1, p. 8, 2019.
- [2] A. P. Nita Patil, "Named Entity Recognition using Conditional Random Fields," no. 3, p. 7, 2019.
- [3] P. Mitra, "Conditional Random Field Based Named Entity," vol. 1, no. 2, p. 4, 2010.
- [4] R. Goswami, "CRF-based Clinical Named Entity Recognition using clinical NLP," vol. 3, no. 1, p. 9, 2020.
- [5] S. E. NAINGGOLAN, "NAMED ENTITY RECOGNITION FOR HERBAL PLANTS," no. 1, p. 7, 2019.