Text Summarization pada Berita Bola Menggunakan *Library Natural Language Toolkit (NLTK)* Berbasis Pemrograman Python

Edoardo Jofan Rifano^{1,*}, Abd. Charis Fauzan², Ahmad Makhi³, Egy Nadya⁴, Zainun Nasikin⁵, Fatra Nonggala Putra⁶

Program Studi Ilmu Komputer, Universitas Nahdlatul Ulama Blitar, Indonesia

¹ejrmanutd20@gmail.com; ²abdcharis@unublitar.ac.id; ³ahmadmakhy@gmail.com; ⁴egy.nadya@gmail.com;

⁵zainunnasikin0@gmail.com; fatranp@unublitar.ac.id

* corresponding author

ARTIKEL INFO

Article history

Diterima: 25 Oktober 2020

Direvisi: 20 Januari 2020

Diterbitkan: 30 April 2020

Keywords

Text Summarization

NLTK

Pemrograman python

ABSTRAK

Tujuan dari penulisan jurnal ini untuk memudahkan pembaca dalam meringkas berita khususnya berita bola dengan menggunakan *library NLTK* pada Pemrograman Python. NLTK, adalah paket program *open source* modul, dan menyediakan komputasi siap pakai khususnya bahasa linguistik. NLTK mencakup pemrosesan bahasa alami simbolik dan statistik, dan dihubungkan dan berorientasi ke *copra*. Meng-import *library NLTK* dengan mengetikkan: `from nltk.corpus import stopwords, from nltk.stem import PorterStemmer, from nltk.tokenize import word_tokenize, sent_tokenize` untuk memanggil metode pada coding. Hasil yang diperoleh bahwa dengan menggunakan *library NLTK* dapat mempermudah pembaca dalam menemukan informasi yang paling penting dalam sebuah berita bola. Terdapat 15 kalimat teks asli yang setelah dilakukan peringkasan dengan *library NLTK* memperoleh hasil ringkasan menjadi 6 kalimat inti berita.

PENDAHULUAN

Model peringkasan teks adalah pendekatan yang dapat digunakan untuk mengompres artikel teks besar menjadi lebih kecil dan lebih ringkas ringkasan. Ini pada dasarnya menggunakan prinsip dan algoritma Pemrosesan Bahasa Alami untuk memahami artikel dan menghasilkan ringkasan yang lebih kecil dan efisien. Peringkasan otomatis adalah proses meringkas suatu dokumen dengan program komputer untuk menemukan kalimat terpenting dari dokumen aslinya [10]. Ada 2 cara dalam peringkasan teks, yaitu ekstraksi dan abstraksi. Ekstraksi adalah meringkas teks dengan cara memilih beberapa bagian teks, lalu digabungkan menjadi sebuah ringkasan. Metode ekstraksi ini adalah metode yang simpel, namun kekurangannya adalah ringkasan yang dihasilkan cenderung acak dan sulit dipahami. Sementara abstraksi akan menghasilkan ringkasan yang lebih mudah dipahami karena prosesnya tidak hanya dengan menggabungkan kata-kata pada setiap kalimat [5].

NLTK merupakan rangkaian perpustakaan dan program pengolahan bahasa simbolik dan statistik alami (NLP) untuk bahasa inggris yang ditulis dalam bahasa Pemrograman Python. NLTK dimaksudkan untuk mendukung penelitian dan pengajaran di NLP yang termasuk ilmu linguistik empiris, ilmu kognitif, kecerdasan buatan, pencarian informasi, dan pembelajaran mesin. NLTK telah berhasil digunakan sebagai alat belajar pribadi, dan sebagai alat perancah dan sistem riset bangunan [9].

Pada penelitian ini data yang digunakan adalah berita bola yang dapat diakses secara online, alasannya pertandingan bola pada zaman sekarang sangat diminati berbagai

kalangan. Oleh karena itu banyak sekali media online menyajikan atau memberitakan berita bola khususnya dalam bentuk artikel, yang terkadang informasi tersebut membuat seseorang kesulitan untuk memahami teksnya yang terlalu panjang. Sehingga memerlukan banyak waktu dalam memahami isi berita dan terkadang membuat pembaca menjadi malas untuk membacanya. Penelitian yang pernah dilakukan Setyadi [7] menggunakan text summarization menggabungkan dua metode yaitu metode Graph dan Ant Colony Optimization, yang dikombinasikan dengan penggunaan empat fitur kalimat. Fitur Kalimat tersebut adalah, kemiripan antar kalimat, kalimat yang menyerupai judul dokumen, TF-ISF dan TF-IDF. Pengujian dilakukan untuk mendapatkan Cosine Similarity antara ringkasan sistem dengan ringkasan manual dari ahli. Uji coba menunjukkan bahwa ringkasan sistem memiliki kemiripan 78,43% dibandingkan dengan ringkasan ahli, dengan tingkat kompresi 78,2% [7]. Ada juga penelitian lain oleh Gede [3] yang menggunakan *Automated Text Summarization* (ATS) yang merupakan aplikasi berbasis komputer untuk menghasilkan ringkasan dari sebuah artikel tanpa menghilangkan tujuan utama dari artikel tersebut. Penelitian itu mengembangkan metode ATS berbasis graf (*Graph-Based Summarization Algorithm*) untuk dokumen berbahasa Indonesia yang menggunakan Algoritma Genetika sebagai penyeleksi kalimat yang tidak sesuai dengan ide pokok dari teks. Dari hasil pengujian diperoleh kesimpulan bahwa untuk proses dokumen yang sama, waktu proses Algoritma Genetika dipengaruhi oleh penentuan nilai generasi, populasi dan elitism. Nilai generasi dan populasi yang ditetapkan berbanding lurus dengan waktu proses dan penggunaan sumber daya. Kemudian penelitian text summarization dari Saraswati [6] menggunakan MMR (*Maximum Marginal Relevance*) meringkas dokumen dengan menghitung kesamaan (*similarity*) antara kalimat dengan kalimat dan antara kalimat dengan *query*. Berdasarkan hasil pengujian didapatkan hasil rata-rata presisi terbaik pada peringkat ke lima sebesar 0,96 untuk hasil sistem temu kembali informasi. Hasil pengujian terbaik dari rata-rata *precision*, *recall*, *f-measure* dan akurasi masing-masing sebesar 0,70, 0,75, 0,70 dan 74,17. Metode yang digunakan sudah cukup baik untuk mendapatkan dokumen yang relevan dengan query dan memperoleh ringkasan berdasarkan judul yang sesuai dengan isi dari dokumen [6].

Dari berbagai penelitian yang sudah pernah dilakukan, *text summarization* masih belum banyak yang mengulas tentang bagaimana *library* NLTK ini dapat digunakan dalam program peringkasan text yang dapat memudahkan pembaca berita dalam mendapatkan informasi penting dalam sebuah text berita. Untuk itu, penelitian ini dibuat agar pembaca lebih mudah memahami isi teks berita yang terlalu panjang dan berbelit-belit dalam menyampaikan informasinya.

METODE

Penelitian ini menggunakan *library* NLTK, yang merupakan paket program open source modul, tutorial dan set masalah, menyediakan komputasi siap pakai khusus bahasa linguistik. NLTK mencakup pemrosesan bahasa alami simbolik dan statistik, dan dihubungkan dan berorientasi ke corpora [4]. NLTK menggabungkan alat pemrosesan bahasa (*tokenizer*, *stemmer*, *taggers*, *parser* sintaksis, analisa semantik) dan set data standar (korpora dan alat untuk mengakses korpora secara efisien dan seragam) [8][11]. *Natural Language Processing* menggunakan beberapa teknik untuk *automated generation*, memanipulasi, dan menganalisis bahasa manusia atau *natural language* [2]. Tahapan-tahapan yang perlu dilakukan untuk melakukan peringkasan teks berita bola menggunakan *library* NLTK adalah seperti Gambar 1 berikut;

Gambar 1. Tahapan penelitian

Gambar 1 adalah alur dari tahapan penelitian *text summarization* menggunakan *Library NLTK* pada berita bola.

Pencarian Data

Dalam penelitian ini, digunakan data berupa berita bola yang diambil dari koran Jawa Pos edisi 20 September 2019 yang berjudul “Doa dari mantan untuk mantan” (diakses pada 07-12-2019). Alasan mengapa menggunakan data berita bola adalah karena peneliti berharap pembaca berita tidak terlalu menghabiskan banyak waktu untuk membaca berita bola, jadi pembaca berita bisa menggunakan waktu untuk melanjutkan kerja lagi ataupun untuk beraktifitas lainnya. Beberapa dari berita bola menyajikan informasi yang cukup panjang dan cukup banyak menyita waktu dalam membacanya. Berita bola ini cukup sering dikunjungi semua kalangan pecinta olahraga sepakbola terkhusus fans-fans fanatik yang kadang akibat kesibukannya hingga tidak sempat melihat pertandingan ataupun menonton pertandingan bola secara langsung sehingga memilih membaca berita bola sebagai solusi. Banyak sekali sumber berita dalam proses pencarian data yaitu berita bola yang akan diringkas dengan cara mengambil berita bola dari koran atau mengunjungi situs berita bola yang terpercaya.

Konfigurasi Library NLTK Pada Python

```
ca. Command Prompt - pip install nltk

C:\Users\Dell>python --version
Python 3.5.2

C:\Users\Dell>pip install numpy
Requirement already satisfied (use --upgrade to upgrade): numpy in c:\users\dell\appdata\local\programs\python\python35\lib\site-packages
You are using pip version 8.1.1, however version 9.0.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:\Users\Dell>pip install nltk
Collecting nltk
  Downloading nltk-3.2.5.tar.gz (1.2MB)
    100% |#####| 1.2MB 38kB/s
Requirement already satisfied (use --upgrade to upgrade): six in c:\users\dell\appdata\local\programs\python\python35\lib\site-packages (from nltk)
Building wheels for collected packages: nltk
  Running setup.py bdist_wheel for nltk ... |
```

Gambar 2. Instalasi NLTK pada python

Sebelum menginstal *library* NLTK terlebih dahulu memasang Python pada sistem operasi. Berikut adalah langkah-langkahnya instalasi *library* NLTK;

1. Langkah 1 buka cmd. Kemudian ketikkan python –version
2. Langkah 2 instal Numpy dengan mengetik perintah. pip install numpy
3. langkah 3 instal NLTK. pip install nltk. Contohnya seperti Gambar 2.

Selanjutnya unduh paket tambahan pada modul NLTK dengan mengetikkan perintah seperti pada Gambar 3.

```
>>> import nltk
>>> nltk.download()
```

Gambar 3. Download NLTK pada Python

Setelah mengetikkan perintah seperti yang ada pada Gambar 3, lalu pilih paket yang dibutuhkan untuk di-install, atau bisa juga mengunduh semua paket yang ada.

Gambar 4. Pilihan paket *library*

Setelah paket-paket pada Gambar 4 terinstal, maka *library* NLTK siap untuk digunakan.

Cara Menggunakan Library NLTK Untuk Peringkasan Text

Gambar 5. Contoh Berita Bola.

Gambar 5 adalah contoh berita bola yang peneliti gunakan untuk diringkas. Berita tersebut diambil dari jawa pos.

```
# Implementation from https://dev.to/davidisrawi/build-a-quick-summarizer-with-
python-and-nltk
import nltk
```

```
from nltk.corpus import stopwords
```

```
text_str = ""
```

BEKASI, Jawa Pos - Persija Jakarta mendapatkan hasil yang memalukan di kandang sendiri. Macan Kemayoran (julukan Persija) dipaksa menyerah dengan skor 0-1 saat menjamu tim pemuncak klasemen Bali United di Stadion Patriot Candrabhaga, Kota Bekasi, Jawa Barat, kemarin sore (19/9).

Gol semata wayang Serdadu Tridatu (julukan Bali United) yang bersarang ke gawang Persija dilesakkan Malvin Platje pada menit ke-72. Kekalahan Persija mendapat simpati dari pelatih Bali United Alessandro Stefano "Teco" Cugurra Rodrigues. Sebagai mantan pelatih Persija. Teco sedih melihat prestasi Ramdani Lestaluhu dan kawan-kawan musim ini. Apalagi, Persija berkompetisi dengan status juara bertahan. "Posisi Persija tahun ini tidak bagus," ucapnya.

Kendati sedang terpuruk. Teco optimistis mantan timnya itu bisa bangkit. Persija masih memiliki dua pertandingan putaran pertama yang belum dijalani. Yaitu, melawan Borneo FC dan Semen Padang. "Karena itu, saya pikir Persija masih punya banyak kesempatan untuk keluar dari posisi ini. Saya akan men-support Persija dari jauh. Sebab, di Persija, banyak teman saya," ujarnya. .

Penjaga gawang sekaligus kapten Macan Kemayoran Andritany Ardhayasa sangat menyesalkan kekalahan kemarin. Setelah kekalahan menyakitkan itu, penjaga gawang tim nasional Indonesia tersebut berjanji segera membawa timnya keluar dari situasi sulit.

```
'''
```

```
def _create_frequency_table(text_string) -> dict:
```

```
'''
```

```
we create a dictionary for the word frequency table.
```

```
For this, we should only use the words that are not part of the stopWords array.
```

```
Removing stop words and making frequency table
```

```
Stemmer - an algorithm to bring words to its root word.
```

```
:rtype: dict
```

```
'''
```

```
stopWords = set(stopwords.words("english"))
```

```
words = word_tokenize(text_string)
```

```
ps = PorterStemmer()
```

```
freqTable = dict()
```

```
for word in words:
```

```
    word = ps.stem(word)
```

```
    if word in stopWords:
```

```
        continue
```

```
    if word in freqTable:
```

```
        freqTable[word] += 1
```

```
    else:
```

```
        freqTable[word] = 1
```

```
return freqTable
```

```
def _score_sentences(sentences, freqTable) -> dict:
    """
    score a sentence by its words
    Basic algorithm: adding the frequency of every non-stop word in a sentence divided by total no
    of words in a sentence.
    :rtype: dict
    """

    sentenceValue = dict()

    for sentence in sentences:
        word_count_in_sentence = (len(word_tokenize(sentence)))
        word_count_in_sentence_except_stop_words = 0
        for wordValue in freqTable:
            if wordValue in sentence.lower():
                word_count_in_sentence_except_stop_words += 1
            if sentence[:10] in sentenceValue:
                sentenceValue[sentence[:10]] += freqTable[wordValue]
            else:
                sentenceValue[sentence[:10]] = freqTable[wordValue]

        if sentence[:10] in sentenceValue:
            sentenceValue[sentence[:10]] = sentenceValue[sentence[:10]] /
            word_count_in_sentence_except_stop_words

    return sentenceValue
```

```
def _find_average_score(sentenceValue) -> int:
    """
    Find the average score from the sentence value dictionary
    :rtype: int
    """

    sumValues = 0
    for entry in sentenceValue:
        sumValues += sentenceValue[entry]

    average = (sumValues / len(sentenceValue))

    return average
```

```
def _generate_summary(sentences, sentenceValue, threshold):
    sentence_count = 0
    summary = ""

    for sentence in sentences:
        if sentence[:10] in sentenceValue and sentenceValue[sentence[:10]] >= (threshold):
            summary += " " + sentence
            sentence_count += 1

    return summary
```



```
def run_summarization(text):  
    # 1 Create the word frequency table  
    freq_table = _create_frequency_table(text)  
  
    # 2 Tokenize the sentences  
    sentences = sent_tokenize(text)  
  
    # 3 Important Algorithm: score the sentences  
    sentence_scores = _score_sentences(sentences, freq_table)  
  
    # 4 Find the threshold  
    threshold = _find_average_score(sentence_scores)  
  
    # 5 Important Algorithm: Generate the summary  
    summary = _generate_summary(sentences, sentence_scores, 1 * threshold)  
  
    return summary  
  
if __name__ == '__main__':  
    result = run_summarization(text_str)  
    print(result)
```

Hasil ringkasan

Sebagai mantan pelatih Persija. Apalagi, Persija berkompetisi dengan status juara bertahan. "Posisi Persija tahun ini tidak bagus, ucapnya. Kendati sedang terpuruk. Yaitu, melawan Borneo FC dan Semen Padang. "Karena itu, saya pikir Persija masih punya banyak kesempatan untuk keluar dari posisi ini. Saya akan men-support Persija dari jauh.

HASIL

Setelah melalui tahapan-tahapan yang telah dilakukan maka penelitian ini memperoleh hasil seperti pada Gambar 6.

Gambar 6. hasil ringkasan berita bola

Dengan menggunakan *library* NLTK dapat mempermudah pembaca dalam menemukan informasi yang paling penting dalam sebuah berita bola, pada berita terdapat 15 kalimat pada teks asli, setelah dilakukan peringkasan dengan *library* NLTK memperoleh hasil ringkasan menjadi 6 kalimat inti berita.

PEMBAHASAN

Dalam pembahasan kali ini peneliti akan memberikan 4 langkah untuk membangun *Summarizer* dengan menggunakan *library* NLTK yaitu:

1. Hapus stopword untuk analisis
2. Buat tabel frekuensi kata - berapa kali setiap kata muncul dalam teks
3. Menetapkan skor untuk setiap kalimat tergantung pada kata-kata yang dikandungnya dan tabel frekuensi
4. Membuat ringkasan berdasarkan skor tertinggi setiap kalimat

Ada dua perpustakaan NLTK yang akan diperlukan untuk membangun *summarizer* yang efisien.

```
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize, sent_tokenize
```

Gambar 7. Impor *library* *stopword*

Pada Gambar 7 program akan mengimpor *library* *stopword* dari *nltk.corpus* dan *word_tokenize, sent_tokenize* dari *nltk.tokenize*. *Corpus* berarti kumpulan teks. Ini bisa berupa kumpulan data tertentu. Dalam hal ini, peneliti akan menggunakan kumpulan data *stopword* yang telah ditentukan sebelumnya. Pada dasarnya, ini membagi teks menjadi serangkaian token. Ada tiga *tokenizer* utama - kata, kalimat, dan *tokenizer regex*. Untuk project ini, peneliti hanya akan menggunakan *tokenizer* kata dan kalimat. Pertama, peneliti membuat dua larik, satu untuk kata *stopword*, dan satu lagi untuk setiap kata dalam kalimat.

```
stopWords = set(stopwords.words("english"))
words = word_tokenize(text)
```

Gambar 8. Kamus untuk tabel frekuensi kata

Pada Gambar 8 peneliti membuat kamus untuk tabel frekuensi kata, yang hanya boleh menggunakan kata-kata yang bukan bagian dari larik *stopWords*.

```
freqTable = dict()
for word in words:
    word = word.lower()
    if word in stopWords:
        continue
    if word in freqTable:
        freqTable[word] += 1
    else:
        freqTable[word] = 1
```

Gambar 9. Kamus *freqTable*

Pada Gambar 9 membuat kamus *freqTable* untuk setiap kalimat untuk mengetahui kalimat mana yang memiliki informasi paling relevan dengan keseluruhan tujuan teks. Karena peneliti sudah memiliki *tokenizer* kalimat, jadi peneliti hanya perlu menjalankan metode *sent_tokenize()* untuk membuat larik kalimat. Kedua, peneliti membutuhkan kamus untuk menjaga skor setiap kalimat, dan menelusuri kamus untuk menghasilkan ringkasan.


```
sentences = sent_tokenize(text)
sentenceValue = dict()
```

Gambar 10. Skor kata

Pada gambar 10 setiap kalimat akan di-skor tergantung banyak kata yang sama dalam setiap kalimat.

```
for sentence in sentences:
    for wordValue in freqTable:
        if wordValue[0] in sentence.lower():
            if sentence[:12] in sentenceValue:
                sentenceValue[sentence[:12]] += wordValue[1]
            else:
                sentenceValue[sentence[:12]] = wordValue[1]
```

Gambar 11. Proses penyeimbangan skor

Gambar 11 menjelaskan bahwa kalimat yang panjang akan lebih diuntungkan daripada kalimat pendek. Untuk mengatasinya maka bagilah setiap skor kalimat dengan jumlah kata dalam kalimat dengan menggunakan “[:12]” agar skor yang diperoleh setiap kalimat menjadi seimbang.

```
sumValues = 0
for sentence in sentenceValue:
    sumValues += sentenceValue[sentence]

# Average value of a sentence from original text
average = int(sumValues / len(sentenceValue))
```

Gambar 12. Hasil peringkasan teks

Gambar 12 menjelaskan bahwa untuk hasil dari ringkasan ini peneliti membuatnya menjadi setengah dari text aslinya

```
from nltk.stem import PorterStemmer
ps = PorterStemmer()
```

Gambar 13. *Stemmer* untuk membawa kata ke kata dasar

Pada gambar 13 peneliti menggunakan *stemmer*, sebuah algoritma untuk membawa kata ke kata dasarnya. Untuk mengimplementasikan *Stemmer*, dapat menggunakan *library stemmer* NLTK. Stemming/*stemmer* adalah bagian dari prapengolahan dari analisis sentimen yang mendeteksi dan menghilangkan kata imbuhan [1].

```
summary = ''
for sentence in sentences:
    if sentence[:12] in sentenceValue and sentenceValue[sentence[:12]] > (1.5 * average):
        summary += " " + sentence
```

Gambar 14. Skoring hasil *stemming*

Pada Gambar 14 setiap kata yang sudah melalui proses stemming akan ditambahkan ke *freqTable* yang kemudian akan di hitung nilai skornya, kalimat yang memiliki skor tertinggi akan di ambil untuk hasil ringkasan sebanyak setengah dari seluruh isi text berita aslinya.

KESIMPULAN

Peringkasan Teks adalah pendekatan yang dapat digunakan untuk meng-*compress* artikel teks besar menjadi lebih kecil dan lebih ringkas. Berdasarkan penelitian dengan menggunakan *library* NLTK untuk meringkas berita bola diperoleh hasil bahwa dengan *library* NLTK dapat mempermudah pembaca dalam menemukan informasi yang paling penting dalam sebuah Berita bola, seperti terlihat pada berita pertama terdapat 15 kalimat pada teks asli setelah dilakukan peringkasan dengan *library* NLTK memperoleh hasil ringkasan menjadi 6 kalimat yang menjadi pokok inti dari isi berita tersebut. Dalam mencari sumber berita hendaknya memperhatikan kebenaran (fakta) dan juga sumber yang jelas, karena pada zaman sekarang banyak berita-berita yang beredar itu sumbernya tidak jelas dan kebanyakan hoax, oleh karena itu harus jeli dalam memilih berita. Terutama disini berita bola sebagai sumber penelitian. Pilihlah berita di situsterpercaya seperti kompas.com, bola.net, sumberbola.com, bola.okezone.com, liputan6.com agar berita yang didapat lebih terpecaya.

REFERENSI

- [1] Agastya, I Made Artha. 2018. "Pengaruh Stemmer Bahasa Indonesia Terhadap Peforma Analisis Sentimen Terjemahan Ulasan Film." *Jurnal Tekno Kompak* 12(1): 18.
- [2] Fuadi, Kholid. 2013. "Pengenalan NLP (Natural Language Processing) Dengan Python Contents." : 1-13.
- [3] Gede, Aditra Pradyana, and Komang Ari Mogi I. 2014. "Implementasi Automated Text Summarization Untuk Dokumen Tunggal Berbahasa Indonesia Dengan Menggunakan Graph-Based." *Jurnal Ilmiah NERO* 1(2): 33-46.
- [4] Loper, Edward, and Steven Bird. 2002. "Nltk." (March): 63-70.]
- [5] Romadhony, Ade, Z R Fariska, Novi Yusliani, and Luciana Abednego. 2017. "Text Summarization Untuk Dokumen Berita Berbahasa Indonesia." *Konferensi Nasional ICT-M Politeknik Telkom*: 408-14.
- [6] Saraswati, Nirmala Fa'izah, Indriati, and Rizal Setya Perdana. 2018. "Peringkasan Teks Otomatis Menggunakan Metode Maximum Marginal Relevance Pada Hasil Pencarian Sistem Temu Kembali Informasi Untuk Artikel Berbahasa Indonesia." *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya* 2(11): 5494-5502.
- [7] Setyadi, I Wayan Adi, Duman Care Khrisne, and I Made Arsa Suyadnya. 2018. "Automatic Text Summarization Menggunakan Metode Graph Dan Metode Ant Colony Optimization." 17(1): 124-30.
- [8] Xue, Nianwen. 2010. "Book Review, Natural Language Processing with Python." *Natural Language Engineering* 17(3): 419-24. http://www.journals.cambridge.org/abstract_S1351324910000306.
- [9] Yao, Jiawei. 2019. "Automated Sentiment Analysis of Text Data with NLTK." *Journal of Physics: Conference Series* 1187(5).
- [10] F. N. Putra, A. Effendi, and A. Z. Arifin, "Pembobotan Kata berdasarkan Kluster untuk Peringkasan Otomatis Multi Dokumen," *J. Linguist. Komputasional*, vol. 1, no. 1, pp. 17-22, 2018.
- [11] F. N. Putra and C. Fatichah, "Klasifikasi jenis kejadian menggunakan kombinasi NeuroNER dan Recurrent Convolutional Neural Network pada data Twitter," *Regist. J. Ilm. Teknol. Sist. Inf.*, vol. 4, no. 2, pp. 81-90, 2018.