TUGAS AKHIR – IF184802

PENGGUNAAN *CLUSTER IMPORTANCE* DAN *NAMED ENTITY RECOGNITION* UNTUK PENENTUAN TRENDING ISSUE DATA TWITTER DALAM PERINGKASAN BERITA MULTIDOKUMEN

REINARDUS WANDYA KRESNAPRABOWO
NRP 05111540000091

Dosen Pembimbing I
Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.

Dosen Pembimbing II
Anny Yuniarti S.Kom., M.Comp.Sc.

Departemen Teknik Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020

TUGAS AKHIR – IF184802

PENGGUNAAN CLUSTER IMPORTANCE DAN NAMED ENTITY RECOGNITION UNTUK PENENTUAN TRENDING ISSUE DATA TWITTER DALAM PERINGKASAN BERITA MULTIDOKUMEN

REINARDUS WANDYA KRESNAPRABOWO
NRP 05111540000091

Dosen Pembimbing I
Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.

Dosen Pembimbing II
Anny Yuniarti S.Kom., M.Comp.Sc.

Departemen Teknik Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020

(Halaman ini sengaja dikosongkan)

UNDERGRADUATE THESIS – IF184802

USE OF CLUSTER IMPORTANCE AND NAMED ENTITY RECOGNITION FOR DETERMINING TRENDING ISSUES TWITTER DATA IN MULTIDOCUMENT NEWS SUMMARIZATION

REINARDUS WANDYA KRESNAPRABOWO
NRP 05111540000091

First Advisor
Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.

Second Advisor
Anny Yuniarti S.Kom., M.Comp.Sc.

Department of Informatics Engineering
Faculty of Intelligent Electrical and Informatics Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2020

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN

PENGGUNAAN *CLUSTER IMPORTANCE* DAN *NAMED ENTITY RECOGNITION* UNTUK PENENTUAN *TRENDING ISSUE* DATA TWITTER DALAM PERINGKASAN BERITA MULTIDOKUMEN

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Komputasi Cerdas dan Visi
Program Studi S-1 Departemen Teknik Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Oleh:
REINARDUS WANDYA KRESNAPRABOWO
NRP: 05111540000091

Disetujui oleh Pembimbing Tugas Akhir:

1. Dr.Eng. Chastine Fatichah, S.Kom., M.Kom. ..... (Pembimbing 1)
   (NIP. 19751220 200112 2 002)
2. Anny Yuniarti S.Kom., M.Comp.Sc. ..... (Pembimbing 2)
   (NIP. 19810622 200501 2 002)

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN INSTITUT TEKNOLOGI SEPULUH NOPEMBER FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS DEPARTEMEN TEKNIK INFORMATIKA

SURABAYA
Januari, 2020

(Halaman ini sengaja dikosongkan)

PENGGUNAAN *CLUSTER IMPORTANCE* DAN *NAMED ENTITY RECOGNITION* UNTUK PENENTUAN *TRENDING ISSUE* DATA TWITTER DALAM PERINGKASAN BERITA MULTIDOKUMEN

Nama Mahasiswa	**: Reinardus Wandya Kresnaprabowo**
NRP	**: 05111540000091**
Jurusan	**: Departemen Teknik Informatika Fakultas Teknologi Elektro dan Informatika Cerdas**
Dosen Pembimbing 1	**: Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.**
Dosen Pembimbing 2	**: Anny Yuniarti S.Kom., M.Comp.Sc.**

ABSTRAK

Peringkasan berita multidokumen merupakan salah satu bentuk pengaplikasian machine learning yaitu pada sistem temu kembali informasi (information retrieval). Peringkasan berita multidokumen bertujuan untuk membuat sebuah berita yang kaya akan informasi. Maka dilakukan proses peringkasan berita dengan masukan sebuah kumpulan berita dan dengan keluaran sebuah ringkasan berita (peringkasan berita multidokumen) dengan harapan pembaca tidak perlu membaca banyak artikel berita untuk mengetahui informasi penting didalamnya.

Proses peringkasan berita multidokumen terdiri dari beberapa fase. Salah satu fase terpenting pada peringkasan berita yang bersifat ekstraktif adalah pembobotan kalimat. Beberapa metode yang digunakan diantaranya posisi kalimat, centroid, dan kemiripan kalimat dengan judul, yang menitikberatkan pada fitur berita itu sendiri. Hal ini menyebabkan peringkasan berita multidokumen menjadi kurang koheren (tidak padu) terutama bila kumpulan berita dalam sebuah topik memiliki isu yang berbeda.

Pada tugas akhir ini, penulis mengusulkan penggunaan Named Entity Recognition (NER) dan Cluster Importance (CI)

untuk melakukan ekstraksi isu pada kumpulan tweet dengan topik tertentu untuk meningkatkan kualitas hasil ringkasan otomatis.

Trending Issue yang telah diekstraksi dengan menggunakan NER dan CI dalam peringkasan berita multidokumen ini berhasil meningkatkan nilai ROUGE dan F1 ringkasan berita hingga 2%.

Kata kunci: Pengenalan Entitas Bernama, *Penambangan Teks, Twitter, Peringkasan Berita Multidokumen*.

USE OF CLUSTER IMPORTANCE AND NAMED ENTITY RECOGNITION FOR DETERMINING TRENDING ISSUES TWITTER DATA IN MULTI DOCUMENT NEWS SUMMARIZATION

Student's Name : Reinardus Wandya Kresnaprabowo
Student's ID : 05111540000091
Department : Department of Informatics Engineering
Faculty of Intelligent Electrical and
Informatics Technology
First Advisor : Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.
Second Advisor : Anny Yuniarti S.Kom., M.Comp.Sc.

ABSTRACT

Multidocument news summarizaion is one form of application of machine learning in the information retrieval system. Multidocument news summarizaion aims to create a news that is rich in information. The process of summarizing with input a collection of news and output a summary (multidocument news summarizaion) have a goals that readers do not need to read many news articles to find important information in it.

The multidocument news summary process consists of several phases. One of the most important phases of summarizing news (extractive method) is sentence weighting. Some of the methods used include sentence position, centroid, and sentence similarity to the title, which emphasizes the news feature itself. This causes the summary of multidocument news becomes less coherent (not coherent), especially if the collection of news on a topic has a different issue.

In this thesis, the authors propose the use of Named Entity Recognition (NER) and Cluster Importance (CI) to extract issues on a collection of tweets with certain topics to improve the quality of automatic summary results.

Trending Issues that have been extracted using NER and CI in summarizing multi-document news have succeeded in

increasing the value of ROUGE and F1 news summary by up to 2%.
Keywords: ***Named Entity Recognition, Text Data, Trending issue, Twitter, Multi Document News Summarization.***

KATA PENGANTAR

Puji syukur kepada Tuhan yang Maha Esa. Karena berkat dan rahmat-Nya, penulis dapat menyelesaikan Tugas Akhir yang berjudul:

" PENGGUNAAN *CLUSTER IMPORTANCE* DAN *NAMED ENTITY RECOGNITION* UNTUK PENENTUAN *TRENDING ISSUE* DATA TWITTER DALAM PERINGKASAN BERITA MULTIDOKUMEN"

Segala proses pengerjaan Tugas Akhir ini tidak terlepas dari bantuan dan dukungan banyak pihak. Oleh karena itu, penulis mengucapkan terima kasih dan penghormatan sebesar-besarnya kepada:

1. Kedua orangtua penulis, kakak, dan anggota keluarga lainnya yang telah memberikan dukungan doa, moral, dan material kepada penulis;
2. Dr.Eng. Chastine Fatichah, S.Kom., M.Kom. dan Anny Yuniarti S.Kom., M.Comp.Sc. sebagai pembimbing I dan II yang telah membimbing penulis dalam menyelesaikan Tugas Akhir;
3. Dr. Eng. Darlis Herumurti, S.Kom., M.Kom. selaku Ketua Departemen Informatika ITS, Dr. Radityo Anggoro, S.Kom, M.Sc. selaku Ketua Program Studi Sarjana Teknik Informatika ITS, dan seluruh dosen dan karyawan Departemen Informatika ITS yang telah memberikan pelajaran dan pengalaman baik dalam ranah kognitif maupun afektif selama penulis menjalani masa kuliah di Informatika ITS;
4. Forum Komunikasi Laboratorium Teknik Computer-Informatika, Khususnya Laboratorium Algoritma & Pemrograman (Alpro) dan Laboratorium Komputasi Cerdas & Visi (KCV) yang telah membantu penulis

mengerjakan Tugas Akhir dan menyediakan tempat di laboratorium tersebut;

5. Silvyana Eka Melinda Mali, Aditya Pratama, serta teman-teman yang secara personal telah membantu & memberi dukungan moral pada penulis selama pengerjaan Tugas Akhir ini bahkan sejak pertama memasuki masa perkuliahan;
6. Teman-teman kontrakan GAES dan Wardug (GIRAS ITS) yang mewarnai masa perkuliahan penulis;
7. Komunitas Helman Salvation Ministry, khususnya keluarga B7, Edita Tanoyo, dan Edwin Wen yang telah menempa pola pikir dan kehidupan penulis sebagai seorang pelayan yang berintegritas;
8. Keluarga Mahasiswa Katolik St. Ignasius Loyola (khususnya segenap kabinet KMK INISIATOR KMK ITS periode 17/18), Komunitas Young Interdenomination Society, dan Bandung ITS yang memberi inspirasi bagi penulis untuk terus berkembang menjadi pemimpin, kakak, dan mentor yang baik;
9. Seluruh mahasiswa Informatika ITS angkatan 2015 serta semua pihak yang yang telah turut membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa laporan Tugas Akhir ini masih memiliki banyak kekurangan. Oleh karena itu dengan segala kerendahan hati penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan penulis kedepannya. Selain itu, penulis berharap laporan Tugas Akhir ini dapat berguna bagi pembaca secara umum.

Surabaya, Januari 2020

DAFTAR ISI

(Halaman ini sengaja dikosongkan)

DAFTAR TABEL

(Halaman ini sengaja dikosongkan)

DAFTAR KODE SUMBER

(Halaman ini sengaja dikosongkan)

DAFTAR GAMBAR

(Halaman ini sengaja dikosongkan)

BAB I
PENDAHULUAN

1.1 Latar Belakang

Peringkasan berita multidokumen merupakan salah satu bentuk pengaplikasian *machine learning* yaitu pada sistem temu kembali informasi (*information retrieval*). Ringkasan berita merupakan sebuah teks yang terdiri dari satu atau lebih kalimat yang mampu menyampaikan informasi penting dari sebuah berita [1]. Untuk membuat sebuah berita yang kaya informasi, maka dilakukan proses peringkasan berita dengan masukan sebuah kumpulan berita dan dengan keluaran sebuah ringkasan berita (peringkasan berita multidokumen).

Proses peringkasan berita multidokumen terdiri dari beberapa fase. Salah satu fase terpenting pada peringkasan berita yang bersifat ekstraktif adalah pembobotan kalimat. Beberapa metode yang digunakan diantaranya posisi kalimat, centroid, dan kemiripan kalimat dengan judul, yang menitikberatkan pada fitur berita itu sendiri. Hal ini menyebabkan peringkasan berita multidokumen menjadi kurang koheren (tidak padu) terutama bila kumpulan berita dalam sebuah topik memiliki isu yang berbeda.

Twitter adalah sebuah media sosial di mana pengguna twitter dapat dengan bebas menge-*tweet* apapun dalam 140 karakter. *Tweets*, dengan kalimatnya yang tidak formal, sering kali kaya akan informasi yang berarti. Namun banyaknya noise dan ketidakformalan *tweets* ini membuat performa pengolahan bahasa natural (*natural language processing*) standar menjadi berkurang. Oleh karena itu, dilakukan sebuah sistem *part-of-speech-tagging* -berupa NER (*Named Entity Recognition*) untuk meningkatkan kualitas ekstraksi informasi pada *tweets*.

NER adalah salah satu metode pengolahan kata, di mana kata-kata dalam sebuah kalimat akan dibagi menjadi entitas / kategori tertentu untuk meningkatkan pemahaman makna dalam sebuah kalimat. Pada tugas akhir ini, penulis mengusulkan penggunaan NER (*Named Entity Recognition*) dan *Cluster*

Importance untuk melakukan ekstraksi isu pada tweet dengan topik tertentu untuk menentukan *Trending issue* dalam sebuah topik. Penggunaan *tweet* tersebut didasari oleh karakteristik tweet, yaitu terdiri dari paling banyak 140 karakter namun kaya akan informasi. Dengan ekstraksi isu ini, diharapkan tingkat keterkaitan hasil ringkasan berita multidokumen menjadi semakin baik.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini adalah sebagai berikut:

1. Bagaimana menemukan *Trending issue* dalam sebuah kumpulan *tweet* dengan topik tertentu dengan menggunakan *Named Entity Recognition* dan *Cluster Importance*?
2. Bagaimana mengimplementasikan *Trending issue* dalam melakukan peringkasan berita multidokumen?
3. Bagaimana mengevaluasi penggunaan *Named Entity Recognition* dan *Cluster Importance* untuk menemukan *Trending issue* dalam melakukan peringkasan berita multidokumen?

1.3 Batasan Permasalahan

Masalah yang diselesaikan pada tugas akhir ini terbatas pada batas-batas sebagai berikut:

1. Berita dan *tweet* yang digunakan yaitu berita dan *tweet* berbahasa Indonesia yang dikumpulkan berdasarkan periode waktu dan topik yang sama.
2. Peringkasan yang digunakan bersifat ekstraktif, dengan masukan berupa kumpulan berita dan keluaran berupa kumpulan kalimat.
3. Penyusunan ringkasan tidak mempertimbangkan urutan kalimat dalam sistematika kepenulisan.
4. Tidak membahas hubungan semantik antar kalimat.

1.4 Tujuan

Tujuan tugas akhir ini adalah membuat sistem yang dapat melakukan peringkasan berita multidokumen dengan teknik pembobotan kalimat berdasarkan fitur berita dan *Trending issue* yang diekstraksi dari *tweets* dengan topik tertentu dengan menggunakan *Named Entity Recognition*.

1.5 Manfaat

Tugas akhir ini diharapkan dapat menambah prespektif dan kemampuan yang ada dalam peringkasan berita multidokumen dengan meningkatkan kualitas hasil peringkasan agar hasil peringkasan lebih koheren. Sehingga dapat diterapkan dalam sistem temu kembali informasi untuk dapat menemukan infromasi sebanyak-banyaknya dalam waktu yang sesingkat-singkatnya.

1.6 Metodologi

Pembuatan tugas akhir ini dilakukan dengan menggunakan metodologi sebagai berikut:

1.6.1 *Penyusunan Proposal Tugas Akhir*

Pengerjaan tugas akhir ini diawali dengan penyusunan proposal tugas akhir yang berisi pendahuluan, tinjauan pustaka, dan metodologi yang akan digunakan pada pembuatan tugas akhir. Pendahuluan terdiri dari latar belakang, rumusan dan batasan masalah yang akan diangkat. Tinjauan pustaka menjadi referensi dalam proses pembuatan tugas akhir. Metodologi berisi tahap-tahap pengerjaan tugas akhir secara keseluruhan.

1.6.2 *Studi Literatur*

Studi literatur dilakukan dalam rangka mengumpulkan informasi yang berkaitan dengan penyelesaian tugas akhir. Informasi yang di cari dalam tugas akhir ini adalah informasi yang berkaitan dengan *Natural Language Processing*, *Named Entity Recognition*, dan peringkasan berita multidokumen.

1.6.3 *Implementasi Perangkat Lunak*

Tahap ini dilakukan sesuai dengan analisis dan desain yang sesuai dengan yang telah dijabarkan sebelumnya. Pada tahap ini dilakukan desain rancangan perangkat lunak pengumpul *tweets* dan berita dengan topik tertentu, pengolahan kata dengan *Named Entity Recognition*, dan peringkasan berita multidokumen dengan menggunakan bahasa pemrograman *Python 3* yang mendukung untuk penggunaan Keras, Tensorflow, Anago, dan Spacy.

1.6.4 *Pengujian dan Evaluasi*

Tahap pengujian dan evaluasi dilakukan dengan menggunakan berita dan *tweet* algoritma ROUGE-N yaitu ROUGE untuk mengetahui tingkat recall sebuah ringkasan berita terhadap kumpulan berita yang diringkas.

1.6.5 *Penyusunan Buku*

Pada tahap ini dilakukan penyusunan buku yang menjelaskan seluruh konsep, teori dasar dari metode yang digunakan, implementasi, serta hasil yang telah dikerjakan sebagai dokumentasi dari pelaksanaan tugas akhir.

1.7 Sistematika Penulisan Laporan

Sistematika penulisan laporan tugas akhir adalah sebagai berikut:

Bab I Pendahuluan

Bab ini berisikan penjelasan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika penulisan dari pembuatan Tugas Akhir.

Bab II Tinjauan Pustaka

Bab ini berisi kajian teori dari metode dan algoritma yang digunakan dalam penyusunan Tugas Akhir ini. Secara garis besar, bab ini berisi tentang *Multi*

Document Summarization, Trending issue Twitter, Named Entity Recognition, serta *library* yang digunakan.

Bab III Perancangan Sistem

Bab ini berisi pembahasan mengenai perancangan sistem pengumpulan data, ekstraksi *Trending issue*, dan pembobotan kalimat yang digunakan untuk melakukan peringkasan berita multidokumen.

Bab IV Implementasi

Bab ini membahas implementasi dari perancangan yang telah dibuat pada bab sebelumnya. Penjelasan berupa kode yang digunakan untuk proses implementasi.

Bab V Uji Coba Dan Evaluasi

Bab ini membahas tahapan uji coba, kemudian hasil uji coba dievaluasi terhadap kinerja dari sistem yang dibangun.

Bab VI Kesimpulan dan Saran

Bab ini merupakan bab yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan, masalah-masalah yang dialami pada proses dan tertulis saat pengerjaan tugas akhir, dan saran untuk pengembangan solusi ke depannya.

(Halaman ini sengaja dikosongkan)

BAB II
TINJAUAN PUSTAKA

Bab ini berisi tinjauan penulis mengenai teori, metode, dan informasi lainnya yang berguna dalam menyelesaikan tugas akhir ini. Informasi ini bertujuan untuk memberikan gambaran umum dan diharapkan dapat mendukung sistem yang dibangun.

Peringkasan dokumen merupakan metode pengolahan informasi secara otomatis. berdasarkan jumlah dokumen yang diproses, peringkasan dokumen dibagi menjadi peringkasan dokumen tunggal dan peringkasan multidokumen. Peringkasan multidokumen menghasilkan sebuah ringkasan yang memuat semua informasi pada setiap dokumen yang pada sebuah kumpulan dokumen.

Beberapa algoritma yang digunakan untuk melakukan peringkasan multidokumen pada yaitu MEAD (*Centroid based multi-document summarization*). MEAD adalah sebuah peringkasan dokumen berita yang memaksimalkan fitur yang dimiliki oleh dokumen yaitu centroid, posisi, dan kemiripan kalimat terhadap kalimat utama [3]. Selain MEAD juga dapat dilakukan klasterisasi menggunakan *global importance dan local importance*. Kedua metode tersebut menggunakan fitur yang ada pada dokumen itu sendiri. Pada dokumen berita dapat terjadi kemunculan lebih dari satu isu (multiple issue) pada topik yang sama. Isu yang beragam ini menyebabkan hasil ringkasan berita multidokumen menjadi kurang koheren (keterpaduan makna). Walaupun banyak isu yang muncul pada topik yang sama, kemungkinan hanya beberapa isu yang menjadi pokok pembicaraan (*Trending issue*). *Trending issue* ini kemudian dijadikan pertimbangan dalam menyeleksi kalimat penting pada proses peringkasan berita multidokumen.

2.1 Named Entity Recognition

Named Entity Recognition (NER) adalah bagian dari ekstraksi informasi yang melihat dan mengklasifikasi entitas-entitas pada teks yang tidak terstruktur ke dalam kategori yang didefinisikan. Beberapa kategori diantaranya nama orang, lokasi, organisasi, keterangan waktu, jumlah, dan kata sifat. NER digunakan secara meluas dalam bidang ekstraksi informasi, menjawab pertanyaan, peringkasan teks, temu kembali informasi, dan berbagai aplikasi yang berkaitan pemrosesan Bahasa natural (*Natural Language Processing*) [2].

2.2 Cluster Importance

Cluster Importance digunakan untuk menemukan kelompok (cluster) yang paling representative. Pada proses peringkasan dokumen, hal ini dilakukan pada fase penyusunan ringkasan. Secara sederhana, metode yang digunakan adalah pengurutan dokumen berdasarkan jumlah kalimat yang terdapat pada masing-masing kelompok. Kelompok dengan jumlah kalimat paling banyak dianggap kelompok yang paling penting dan akan diletakkan di awal ringkasan. Namun pengurutan tingkat kepentingan sebuah kelompok berdasarkan jumlah kalimat memiliki kelemahan. Kelompok dapat mengandung kalimat yang sama. Sebuah kelompok juga bisa terdiri dari banyak kalimat pendek yang tidak informatif.

Menghadapi masalah di atas, dikembangkan sebuah metode untuk meningkatkan performa *Cluster Importance* [4] Kalimat dianggap sebagai tweet pada saat ekstraksi trending issue dan bobot akan diberikan pada cluster berdasarkan jumlah kata penting yang terkandung dalam cluster. Setiap kata penting atau kata kunci dianggap sebagai sebuah cluster. Kata penting diseleksi berdasarkan jumlah kemunculan kata pada tweet atau dokumen input (count(k)). Count(k) yang terseleksi nilainya harus lebih besar dari nilai ambang (threshold). Sehingga bobot sebuah cluster dihitung dengan menjumlahkan bobot dari seluruh kata penting yang muncul dalam sebuah cluster, cara

penghitungan bobot cluster (W(CI)) selengkapnya dapat dilihat pada persamaan sebagai berikut:

$$bobot\ cluster\ CI, W(CI) = \sum_{w \in C} \log(1 + count(k)) \qquad (2.1)$$

2.3 Pembobotan Kalimat

Pembobotan kalimat adalah sebuah tahap yang penting dalam peringkasan dokumen dengan teknik ekstraksi. Kalimat dengan bobot yang memenuhi ambang batas akan dimasukkan ke dalam ringkasan karena dianggap kalimat yang penting. Berikut ini adalah fitur yang digunakan untuk menentukan bobot sebuah kalimat dalam dokumen berita.

2.3.1 *Word Frequency*

Semakin sering sebuah kata muncul dalam sebuah berita, maka kata tersebut dianggap sebagai kata penting. Sehingga semakin sering sebuah kata muncul dalam sebuah dokumen, maka bobotnya akan semakin tinggi.

Setelah dilakukan ekstraksi term dari dokumen, kemudian membuat *ranking term* berdasarkan kemunculan term pada dokumen. Term yang memiliki bobot diatas ambang batas akan dimasukkan ke dalam *Word Freaquency List* (*WFList*) yang akan digunakan sebagai fitur pembobotan kalimat.

2.3.2 *TF-IDF*

TF (*Term Frequency*) menunjukkan seberapa sering sebuah term muncul dalam sebuah dokumen. Perbedaan panjang setiap dokumen menyebabkan perbedaan kemungkinan kemunculan sebuah term. Karena itu *TF* dibagi dengan panjang dokumen sebagai bentuk normalisasi.

IDF (*Inverse Document Frequency*) menunjukkan seberapa penting sebuah term. Menghitung *TF* dimulai dengan menganggap semua term sama penting. Namun banyak term seperti “yaitu”, “adalah” dan “yang” yang muncul dalam banyak

dokumen namun memiliki tingkat kepentingan yang rendah. Sehingga kita perlu menurunkan bobot term yang muncul berulang kali dan menaikkan bobot yang hanya muncul dalam sedikit dokumen. Berbeda dengan TF yang semakin sering frekuensi kata muncul maka nilai semakin besar, dalam IDF, semakin sedikit frekuensi kata muncul dalam dokumen, maka makin besar nilainya. Dengan D adalah jumlah semua dokumen dalam koleksi dan df_j adalah jumlah dokumen yang mengandung term (t_j), maka rumus IDF dapat dilihat pada (2.2) .

$$IDF_j = \log\left(\frac{D}{df_j}\right) \tag{2.2}$$

$$TF - IDF_{Ij}, W_{ij} = tf_{ij} x\ idf_j \tag{2.3}$$

Rumus TF-IDF didapatkan dengan mengalikan nilai TF dengan nilai IDF (2.3) Dimana W_{ij} adalah bobot term (t_j) terhadap dokumen (d_i). Sedangkan tf_{ij} adalah jumlah kemunculan term (t_j) dalam dokumen (d_i). D adalah adalah jumlah semua dokumen yang ada dalam database dan d_{fj} adalah jumlah dokumen yang mengandung term (t_j) (minimal ada satu kata yaitu term (t_j)).

2.3.3 *Posisi Kalimat*

Pembobotan sebuah kalimat dapat dilakukan berdasarkan posisinya dalam sebuah dokumen. Pada penelitian [5] dijelaskan bahwa kalimat yang posisinya berada di awal dokumen memiliki skor lebih besar dari kalimat dengan posisi di akhir. Mengutip pernyataan Baxendale bahwa kebanyakan kalimat awal sebuah paragraf adalah kalimat utama (*topic sentence*).

Walaupun dalam kenyataan sering ditemui paragraf dengan kalimat utama di tengah maupun di akhir, namun teknik yang paling banyak digunakan dalam berita online adalah “piramida terbalik”. Dalam teknik ini, berita diawali dari hal yang dianggap paling penting, baru diakhiri oleh kalimat-kalimat pelengkap. Hal

ini menjadi alasan bahwa pemberian bobot lebih pada kalimat yang berada di posisi awal berkaitan dengan kalimat utama..

2.3.4 *Kemiripan dengan Judul*

Dalam penulisan sebuah berita, judul adalah hal yang penting. Sebuah judul berita minimal mengandung unsur Subjek Predikat dan Object yang bisa diambil dari kutipan isi berita. Hal ini yang mendasari pembobotan kalimat berdasarkan kemiripan dengan judul. Semakin tinggi tingkat kemiripan sebuah kalimat dengan judul, maka kalimat itu akan dianggap semakin penting

2.3.5 *Kemiripan dengan Trending issue*

Tren (KBBI) adalah gaya mutakhir. Sedangkan isu (KBBI) adalah masalah yang dikedepankan (untuk ditanggapi dsb.). Sehingga *Trending issue* dapat didefinisikan sebagai permasalahan yang terbaru yang dikedepankan untuk diperbincangkan atau menjadi pembahasan.

Studi [6] mengelompokkan *tweets* berdasarkan kesamaan isi, kemudian menyeleksi satu kelompok *tweets* dengan skor tertinggi untuk dijadikan sebgai *Trending Topic*. Isu dapat diidentifikasi dari kata kunci (*keyword*) yang muncul dalam kelompok *tweet* dengan isu yang sama [7]. Kata kunci dapat dipahami sebagai unit terkecil dari beberapa term, yang dapat mewakili, merangkum, dan mengidentifikasi pokok pikiran dalam sebuah teks. (Abilhoa & de Castro, 2014).

Berdasarkan definisi isu dan *Trending Topic*, *Trending issue* bisa didapatkan dengan mencari isu dengan skor yang tertinggi diantara isu-isu yang ada. Pada tugas akhir ini, pemberian bobot skor pada kelompok isu akan menggunakan *Cluster Importance*.

Mendeteksi *Trending Topic* dari *tweets* dilakukan dengan mengelompokkan *tweets* berdasarkan kesamaan topik dengan metode *K-Means*. Setelah itu, dilakukan pembobotan dengan *TF-IDF*. Kelompok dengan bobot terbesar akan digunakan sebagai

Trending Topic yang nantinya akan dilakukan ekstraksi isu dengan mengekstraksi kata kunci (*keyword*) dari kelompok *tweets* tersebut.

2.3.6 *Cosine Similarity*

Cosine Similarity menunjukkan nilai kemiripan sebuah kalimat dengan kalimat lainnya dengan menggunakan nilai cosinus sudut antara dua vector. Jika terdapat dua vector (dokumen di ¬dan dj) yang mengandung kata-kata (k) maka nilai cosinus antara dua pasangan dokumen tersebut dapat dihitung menggunakan persamaan 2.4 berikut:

$$Cosine(A,B) = \frac{\sum_{n=1}^{j}(nA \times nB)}{\sqrt{\sum_{n=1}^{j}(nA)^2} \times \sqrt{\sum_{n=1}^{j}(nB)^2}} \quad (2.4)$$

dengan j = |A ∩ B|, nA = jumlah kemunculan kata indeks ke-n dari daftar kata pada kalimat A. dan nB = jumlah kemunculan kata indeks ke-n dari daftar kata pada kalimat B.

2.4 Python

Python adalah bahasa pemograman yang populer. Python sering dimanfaatkan dalam pengembangan web, perangkat lunak, penelitian, dan system scripting. Python dapat digunakan untuk menangani data besar dan melakukan operasi matematika yang kompleks. Python bekerja di berbagai *platform* seperti Windows, Mac, Linux, Raspberry Pi, dan lain-lain. Python dirancang untuk mudah dibaca, yaitu memiliki sintaks yang sederhana dan menggunakan bahasa inggris [1].

2.5 Keras

Keras adalah *high-level neural networks* API, yang ditulis dalam bahasa pemograman Python dan mampu berjalan di atas *TensorFlow* dan *Theano*. Keras dikembangkan dalam rangka memungkinkan eksperiman dilakukan dengan cepat. Keras dapat berjalan baik di CPU dan GPU. Keras berisi banyak implementasi *neural network* yang umum digunakan, fungsi aktivasi, *optimizer*,

dan *tool* lain yang memudahkan dalam pengolahan citra dan data teks [2].

2.6 TensorFlow

TensorFlow adalah *library open source* untuk pembuatan program yang membutuhkan komputasi numerik berkinerja tinggi. *TensorFlow* dikembangakan oleh tim Google Brain. *TensorFlow* menyediakan fungsi-fungsi *machine learning* dan *deep learning*, dan dapat dijalankan dalam CPU atau GPU [3].

2.7 ROUGE

Recall-Oriented Understudy for Gisting Evaluation (ROUGE) adalah sebuah metrik yang digunakan untuk mengevaluasi perangkat lunak peringkasan otomatis dan penerjemahan mesin di dalam *natural language processing*. ROUGE membandingkan ringkasan atau terjemahan yang telah dihasilkan secara otomatis terhadap referensi ringkasan atau terjemahan yang dibuat oleh manusia (sebagai *ground truth)*.

BAB III
PERANCANGAN SISTEM

Sistem yang dibuat pada tugas akhir ini terdiri dari pengumpulan data (*tweets* dan berita) dan sistem peringkasan berita multidokumen.

3.1 Perancangan Data

Data yang digunakan sebagai masukan pada sistem ini terdiri dari dua jenis data yaitu *tweets* dan kumpulan berita.

3.1.1 *Data Twitter*

Data Twitter digunakan sebagai masukan awal dari sistem peringkasan berita multidokumen yang akan digunakan dalam pencarian *trending issue*. Pencarian data Twitter ini dimulai dengan menentukan sebuah topik (misalnya "Lamborghini Koboi") dan melakukan *scrapping* dengan *library* Twint. Contoh data Twitter yang digunakan dapat dilihat pada Tabel 3.1.

Tabel 3.1 Contoh *Tweets* (dengan *Query*: Lamborghini Koboi)

	Username	Text
1	muriskah	Polres Jaksel Terus Telusuri Indikasi Peralihan Kepemilikan Lamborghini oleh Sang Koboi https://ift.tt/2Sxi6w7Â pic.twitter.com/ra8zsQeyZy
2	Berita7	Manager Showroom Ungkap Pengusaha 'Koboi' Adalah Pemilik Asli Lamborghini - http://bit.ly/37t4arxÂ pic.twitter.com/TvE3DU6zENt
3	okezonenews	Polres Jaksel Terus Menelusuri Indikasi Peralihan Kepemilikan Lamborghini oleh Sang Koboi #TauCepatTanpaBatas #BeritaTerkini #BeritaTerkini #NewsUpdate . https://news.okezone.com/read/2019/12/28/338/2147035/polres-jaksel-terus-menelusuri-indikasi-peralihan-kepemilikan-lamborghini-oleh-sang-koboiÂ â€¦

3.1.2 *Data Berita*

Berita yang digunakan sebagai penyusun ringkasan berita pada sistem ini dikumpulkan dari beberapa portal berita online dengan waktu yang berdekatan dengan pengumpulan *tweet*. Pengumpulan yang bersamaan ini, diharapkan keselarasan antara hasil berita yang dikumpulkan dengan isu yang didapatkan dari *tweets*. Berita dikumpulkan dalam format .xml (Gambar 3.1). Berita yang dikumpulkan memiliki atribut judul, id (nama file), tanggal, kata kunci, dan isi.

```
<artikel>

<judul>Pengemudi Lamborghini Todong Pelajar di Kemang Positif Ganja</judul>

<id>Lamborghini_CNN</id>

<tanggal>CNN Indonesia | Selasa, 24/12/2019 16:38 WIB</tanggal>

<tag>lamborghini, aksi koboi pengemudi lamborghini, polda metro jaya</tag>

<isi>Jakarta, CNN Indonesia -- Pngendara mobil Lamborghini yang melakukan penodongan kepada dua pelajar di
Kemang, Jakarta Selatan positif menggunakan narkotika jenis ganja. Pengemudi berinisial AM itu saat ini
sudah ditetapkan sebagai tersangka. "Tim reserse Polres Jakarta Selatan coba mendalami kemungkinan pelaku
mabok atau tidak. Ternyata positif ganja," kata Kepala Bidang Humas Polda Metro Jaya Kombes Yusri Yunus di
Polres Metro Jakarta Selatan, Kebayoran Lama, Jakarta Selatan pada Selasa (24/12). Kendati demikian, tidak
ditemukan barang bukti narkotika di mobil maupun rumah AM. Namun polisi terus mendalami kemungkinan
penyalahgunaan narkotika oleh AM ini. Polisi menyebut AM merupakan seorang pengusaha properti. Ia juga
merupakan pemilik Lamborghini jenis Gallardo tersebut. Atas perbuatannya ia dijerat dengan Pasal 335 KUHP
tentang perbuatan tidak menyenangkan dengan ancaman pidana 1 tahun penjara. Peristiwa penodongan oleh
pengemudi Lamborghini terhadap dua pelajar SMA asal Jakarta itu terjadi di kawasan Kemang pada Sabtu
(21/12). Saat itu salah satu pelajar berinisial A bersama temannya, I, ingin membeli kopi di kawasan
tersebut. Dalam perjalanan, keduanya melihat mobil Lamborghini oranye. Keduanya lalu saling bercanda
mengenai Lamborghini tersebut. Keduanya juga tertawa sambil terus bercanda. Diduga, pemilik mobil mewah itu
tidak terima, lalu menodongkan pistol kepada mereka. Karena insiden tersebut, pihak korban melaporkan AM ke
Polres Metro Jakarta Selatan. Polisi menangkap AM di kediamannya.</isi>

<link>https://www.cnnindonesia.com/nasional/20191224141322-12-459636/pengemudi-lamborghini-todong-pelajar-
di-kemang-positif-ganja</link>

</artikel>
```

Gambar 3.1 Contoh Data Berita

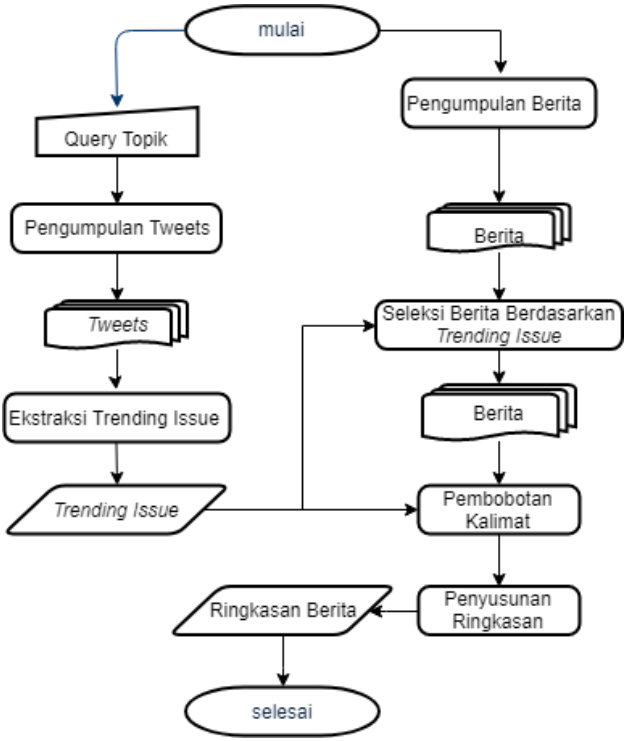
3.2 Desain Umum Sistem

Sistem yang dibangun ini memiliki beberapa proses utama diantaranya pengumpulan data, praproses data, pelatihan dan pengujian model *Named Entity Recognition*, ekstraksi *Trending issue* dan peringkasan berita multidokumen. Diagram alir dari sistem ditunjukan pada Gambar 3.2.

Tahap ekstraksi *trending issue* terdiri dari beberapa tahap yaitu filtrasi tweet berentitas dengan menggunakan metode

Named Entity Recognition dan pemilihan *cluster* dengan metode *Cluster Importance*.

Trending issue yang telah didapatkan akan digunakan sebagai fitur pembobotan kalimat untuk membentuk ringkasan berita.



Gambar 3.2 Diagram Alir Sistem yang akan Dibangun

3.2.1 *Praproses Data*

3.2.1.1 Praproses Data Twitter

Kumpulan *tweets* yang sebelumnya telah dikumpulkan akan digunakan untuk mencari trending isu. Sebelum melalui proses ekstraksi trending isu, ada data *tweets* harus diproses

terlebih dahulu. Tahap praproses *tweets* yang bertujuan agar proses setelahnya dapat berjalan dengan lancar ini dilakukan dengan menormalisasi *tweets*.

Praproses data Twitter ini dibagi menjadi dua tahap, yaitu sebelum *tweet* difiltrasi menggunakan NER. Pada tahap praproses ini, *tweets* yang kosong & mengandung isi kurang dari satu kata dibuang. Beberapa hal yang tidak akan diproses selanjutnya seperti *hashtag*, tautan, video, gambar, dan *emoticon* pun akan dihapus dari *tweets*.

Tahap praproses *tweets* ini tidak termasuk *case folding*, *stemming*, *tokenizing* dan penghapusan *stopwords*. Hal ini disebabkan karena proses tersebut akan mempengaruhi hasil filtrasi NER pada tahap selanjutnya.

Tahap kedua adalah tahap setelah *tweet* difiltrasi menggunakan NER. Pada tahap praproses ini, pada data *tweet* diterapkan *case folding*, *stemming*, *tokenizing*, dan penghapusan *stopwords.*

3.2.1.2 Praproses Data Berita

. Berbeda dengan praproses data *tweets*, tahap praproses data berita dimulai dari pembentukan struktur data (merubah dari XML ke *DataFrame*), penanganan *unicode* dan karakter spesial, dengan penghapusan *stopword*, *tokenizing*, dan diakhiri *stemming*,

3.2.2 *Tahap Perancangan Arsitektur NER*

Setelah melakukan praproses terhadap data *tweets* dan berita, akan dilakukan perancangan arsitektur NER, akan dilakukan pelatihan dan pengujian dengan menggunakan data latih NER. Data yang digunakan merupakan kumpulan entitas bernama (*Named Entity*) berbahasa Indonesia Tabel (3.2). terdiri dari nama orang, nama tempat, nama organisasi, dan bilangan.

Tabel 3.2 Contoh Pelabelan Entitas Bernama

Token	Bambang	Pamungkas	anggota	Timnas	Indonesia
Label	PER	PER	O	ORG	ORG

Arsitektur NER yang digunakan pada Anago adalah LSTM (*Long Short-Term Memory)* di mana urutan kata dan konteks sangat diperhatikan dan akan mempengaruhi hasil pelatihan. Oleh karena itu, label pada data training akan dibuat menjadi format BIO (*Begin, Inside, Outside*) seperti pada Tabel (3.3). Format BIO merupakan bagian dari skema penandaan *(tagging scheme)* BILOU (*Begin, Inside, Outside, Unit).*

Tabel 3.3 Contoh Pelabelan dengan Format BIO

Token	Putusan	Mahkamah	Konstitusi	bersifat	final
Label	O	B-ORG	I-ORG	O	O

Setelah menyiapkan data training, selanjutnya model akan dibuat dengan menggunakan *Tensorflow* dan Keras.

Berdasarkan hasil uji coba, sistem deteksi kejadian menggunakan penggabungan NeuroNER dan RCNN bekerja dengan sangat baik dengan nilai rata-rata precision, recall, dan f-measure masing-masing 94,87%, 92,73%, dan 93,73% Pembangunan Arsitektur Sistem [4]

3.2.2.1 Ekstraksi *Trending issue Twitter*

Ekstraksi *Trending issue Twitter* terdiri dari beberapa bagian yaitu filtrasi *tweets* dengan menggunakan *Named Entity Recognition (NER)* yang akan menghasilkan kelompok *tweets* yang berentitas. Kemudian dilanjutkan dengan pengelompokkan *tweets* menggunakan Kmeans. Setelah terbentuk kelompok-kelompok berdasarkan isi tweet, dilanjutkan dengan mencari kata representative dari setiap kelompok. Pembobotan Kalimat

Pembobotan kalimat adalah bagian yang terpenting dalam penyusunan ringkasasn berita yang bersifat ekstraktif yang dilakukan dengan menjumlakan semua bobot kalimat.

Pembobotan dimulai dengan menghitung *Term-Frequency Inverse Document Frequency*. Perhitungan TF-IDF pada peringkasan berita multidokumen tentunya berbeda dengan perhitungan TF-IDF pada peringkasan berita satu dokumen. Pada tugas akhir ini selain TF-IDF, posisi kalimat, dan kemiripan dengan judul, kemiripan dengan *Trending issue* dan kandungan entitas bernama juga digunakan sebagai pembobotan kalimat.

3.2.2.2 Penyusunan Ringkasan Berita

Berdasarkan bobot kalimat yang sudah didapatkan pada tahap sebelumnya, kalimat diurutkan mulai dari bobot tertinggi. Jumlah kalimat yang akan diringkas dapat ditentukan dari awal, maupun dapat menggunakan ambang batas bobot kalimat (kalimat yang dimasukkan ke ringkasan hanya yang berbobot di atas ambang batas)

(Halaman ini sengaja dikosongkan)

BAB IV
IMPLEMENTASI

Bab ini menjelaskan mengenai implementasi perangkat lunak dari rancangan sistem yang telah dibahas pada Bab 3 meliputi kode program dalam perangkat lunak. Selain itu, implementasi dari tiap proses, parameter masukan, keluaran, dan beberapa keterangan yang berhubungan dengan program juga dijelaskan.

4.1 Lingkungan Implementasi

Dalam mengimplementasikan aplikasi pengenalan ekspresi manusia diperlukan beberapa perangkat pendukung sebagai berikut.

4.1.1 *Perangkat Keras*

Implementasi tugas akhir ini menggunakan desktop *personal computer* (PC) Dell Inspiron 15 7000. Sistem operasi yang digunakan adalah Windows 10 64-bit. PC yang digunakan memiliki spesifikasi Intel Core i7-7700HQ dengan kecepatan 2,8 GHz (8CPU), *Random Access Memory* (RAM) sebesar 8 GB, dan mempunyai *Graphics Processing Unit* (GPU) yaitu NVIDIA GeForce GTX 1050 Ti sebesar 8 GB.

4.1.2 *Perangkat Lunak*

PC dari sisi perangkat lunak memiliki spesifikasi antara lain menggunakan bahasa pemograman Python 3.6, dilengkapi dengan *library* antara lain Tensorflow, Keras, Anago, Pandas, NLTK, Spacy dan Scikit-learn.

4.2 Implementasi Praproses Data

Pada subbab ini akan dijabarkan implementasi pada tahap praproses data *tweets* dan berita.

4.2.1 *Implementasi Praproses Tweets*

Praproses data *tweets* dibagi menjadi dua tahap. Praproses data *tweets* pertama diimplementasikan pada Kode Sumber 4.1.. Praproses dilakukan pada dengan menggunakan library html untuk melakukan penanganan pada karakter *unicode*, dan dilanjutkan dengan membersihkan *emoticon*, tautan, dan tanda baca yang berpotensi mengganggu proses selanjutnya.

```
1.  def preprocessing_tweets_first(text):
2.      clean = html.unescape(clean)
3.      clean = re.sub("[0-9]", "", clean)
4.      clean = re.sub(r"http\S+", "", clean)
5.      clean = clean = dEmojify(clean)
6.      clean = " ".join(re.findall("[#a-zA-
    Z]{3,}", clean))
7.      return clean
```

Kode Sumber 4.1 Implementasi Praproses Data *Tweet* Tahap Pertama

Praproses data *tweets* kedua diimplementasikan pada Kode Sumber 4.2. Praproses kedua ini menerapkan *case folding*, *stemming*, *tokenizing*, dan penghapusan *stopwords* dengan menggunakan Sastrawi, serta penanganan kata-kata tidak baku dengan menggunakan kamus kata tidak baku (*slang dictionary*).

```
1.  def preprocessing_tweets_second(text):
2.      clean = text.lower()
3.      clean = stopword(clean)
4.      clean = normalize_slang_words(clean)
5.
6.      factory = StemmerFactory()
7.      stemmer = factory.create_stemmer()
8.      clean   = stemmer.stem(clean)
9.      return clean
```

Kode Sumber 4.2 Praproses Data *Tweet* Tahap Kedua

4.2.2 *Implementasi Praproses Berita*

Praproses berita diimplementasikan pada Kode Sumber 4.3. Praproses berita dimulai dengan membaca kumpulan file berita (dari XML menjadi Pandas *DataFrame*),

```
1.  directory = "./ berita/"
2.  df = pd.DataFrame(columns=['judul','id','tanggal','
    kata_kunci','ísi'])
3.  for berita in os.listdir(directory):
4.      filename = directory+berita
5.      xtree = et.parse(filename).getroot()
6.      judul = xtree.find('judul').text
7.      ids = xtree.find('id').text
8.      tanggal = xtree.find('tanggal').text
9.      kata_kunci = xtree.find('kata_kunci').text
10.     isi = xtree.find('isi').text
11.     df = df.append({'judul':judul,'id':ids,'tanggal
    ':tanggal,'kata_kunci':kata_kunci,'ísi':isi},ignore
    _index=True)
```

Kode Sumber 4.3 Membaca Data Berita

Seperti pada praproses *tweets*, penanganan *unicode* dan karakter spesial dilakukan dengan menggunakan *library* html. Sedangkan untuk *stopwords removal, stemming dan tokenizing* diimplementasikan dengan bantuan *library* Sastrawi dengan tambahan *stopwords* Bahasa Indonesia di luar yang telah disediakan oleh Sastrawi.

```
1.  def preprocessing_news(text):
2.      clean = text.lower()
3.      clean = stopword(clean)
4.
5.      factory = StemmerFactory()
6.      stemmer = factory.create_stemmer()
7.      clean   = stemmer.stem(clean)
8.      return clean
```

Kode Sumber 4.4 Praproses Data Berita

4.3 Implementasi Pelatihan dan Pengujian Anago

```
os.environ['PYTHONHASHSEED'] = '0'
np.random.seed(42)
rn.seed(12345)

model = anago.Sequence(char_emb_size=25,
          word_emb_size=100, char_lstm_units=25,
          word_lstm_units=100, dropout=0.5,
          char_feature=True, crf=True,
          batch_size=20, optimizer='adam',
          learning_rate=0.001, lr_decay=0.9,
          clip_gradients=5.0, max_epoch=30,
          early_stopping=True, patience=3,
          train_embeddings=True,
          max_checkpoints_to_keep=5,
          log_dir=None)
```

Kode Sumber 4.5 Deklarasi dan Mengatur Parameter Model Anago

```
namaFileTrain = "train.txt"
namaFileValid = "valid.txt"
namaFileTest = "test.txt"
x_train, y_train = load_data_and_labels(namaFileTrain)
x_valid, y_valid = load_data_and_labels(namaFileValid)
x_test, y_test = load_data_and_labels(namaFileTest)

model.train(x_train, y_train, x_valid, y_valid)
```

Kode Sumber 4.6 Memuat File dan Melatih Model

4.4 Implementasi Ekstraksi *Trending issue*

Ekstraksi *Trending issue* dimulai dengan memfilter *tweets* yang sudah melewati praproses tahap pertama dengan menggunakan model Anago yang sudah dilatih sebelumnya.

```
## filtering tweets with entity
```

```
tweets_with_entity = pd.DataFrame(columns=['tweet',
 'clean'])
for index, row in tweets.iterrows():
#     index+=1
    if len(row['clean'].split()) >= 3: # filtering
tweets that contain < 3 words
        if row['clean'] not in tweets_with_entity['
clean'].tolist(): # filtering exactly same tweet
            result = AnagoModel.analyze(row['clean'
].split())
            if result.get("entities") != []:
                tweets_with_entity = tweets_with_en
tity.append({'tweet':row['tweet'], 'clean':row['cle
an']}, ignore_index=True)
```

Kode Sumber 4.7 Pembentukan Vector TF-IDF *Tweets*

Tahap filtrasi menghasilkan kumpulan *tweets* yang mengandung entitas bernama. Pada kelompok *tweets* inilah akan dilakukan tahap praproses kedua. Setelah dilakukan praproses ke dua, selanjutnya dibentuk vector yang berisi TF-IDF *tweets* dengan mengimplementasikan Kode Sumber 4.7.

```
from sklearn.feature_extraction.text
                    import TfidfVectorizer
tfidf_vectorizer = TfidfVectorizer(max_df=0.8,
                    max_features=200000,
                    min_df=0.2, stop_words=stopwords,
                    use_idf=True,
                    tokenizer=tokenize_only,
                    ngram_range=(1,3))
tfidf_matrix = tfidf_vectorizer.fit_transform(
                    tweets['text'].tolist())
```

Kode Sumber 4.8 Pembentukan Vector TF-IDF *Tweets*

Setelah melewati tahap praproses kedua, kelompok *tweets* dengan entitas bernama tersebut akan dikelompokkan dengan menggunakan K-Means clustering.

```
13. true_k = 5
14. model = KMeans(n_clusters=true_k, init='k-
    means++', max_iter=100, n_init=1)
15. model.fit(tfidf_matrix)
```

Kode Sumber 4.9 Pembentukan Model Clustering *Tweets*

Pada setiap kelompok yang terbentuk, akan dipilih *terms* yang dianggap mewakilkan kelompok tersebut berdasarkan jarak terdekat dengan *centroids*. Kumpulan *terms* inilah yang dianggap sebagai *Trending issue.*

```
1.  print("Top terms per cluster:")
2.  order_centroids = model.cluster_centers_.argsort(
        )[:, ::-1]
3.  terms = vectorizer.get_feature_names()
4.  for i in range(true_k):
5.      print("Cluster %d:" % i),
6.      for ind in order_centroids[i, :10]:
7.          print(' %s' % terms[ind]),
8.      print
```

Kode Sumber 4.10 Pembentukan Vector TF-IDF *Tweets*

4.5 Implementasi Peringkasan Berita

Setelah mendapatkan *Trending issue*, dilakukan filtrasi pada kumpulan berita (Kode Sumber 4.8). Berita akan dibagi menjadi dua, yaitu berita yang mengandung konten yang relevan dengan *Trending issue*, dan berita yang tidak mengandung konten yang relevan dengan *Trending issue*.Kumpulan berita yang mengandung konten yang relevan dengan *Trending issue* ini yang selanjutnya akan sebuah ringkasan.

```
9.  def news_filtering(news_collection, terms):
10.     trending_news = list()
11.     for news in news_collection:
12.         result = False
13.         for sent in news:
14.             result = any (term in terms for word
                                  in sent)
15.             if result:
16.                 trending_news.append(news)
17.                 continue
```

Kode Sumber 4.11 Filtrasi Berita Berdasarkan *Trending issue*

Berita yang sudah terkelompokkan kini akan ditinjau satu persatu untuk dilalkukan pembobotan kalimat. Pada tahap ini akan dihitung bobot masing-masing fitur kalimat dalam berita.

Fitur pertama yang dihitung adalah *term frequency*

```
18. def frequency_scores(self, article_text):
19.         response = self._tfidf.transform
                       ([article_text])
20.         feature_names = self._tfidf.
                       get_feature_names()
21.         word_prob = {}
22.         for col in response.nonzero()[1]:
23.             word_prob[feature_names[col]] =
                       response[0, col]
24.         if DEBUG:
25.             print(word_prob)
26.
27.         sent_scores = []
28.         for sentence in self.split_into_sentences
                        (article_text):
29.             score = 0
30.             sent_tokens = self.
                        tokenize_and_stem(sentence)
31.             for token in (t for t in sent_tokens
                             if t in word_prob):
32.                     score += word_prob[token]
33.
34.             sent_scores.append(score /
```

```
                        len(sent_tokens))
        return sent_scores
```

Kode Sumber 4.12 Perhitungan *Term Frequency* Kalimat Berita

Fitur selanjutnya yang dihitung adalah TF-IDF. Perhitungan TF-IDF dalam peringkasan berita multidokumen berbeda dengan menghitung TF-IDF dalam sebuah dokumen saja. Dalam peringkasan berita multidokumen, model TF-IDF dibangun bagi setiap kata dalam sebuah berita terhadap seluruh koleksi berita yang diringkas.

```
def build_TFIDF(self):
        directory = "./ScrapeNews/news/"
        token_dict = {}
        for news in os.listdir(directory):
            token_dict[news] = news.isi
        self._tfidf = TfidfVectorizer
                    (tokenizer=self,
                    stop_words=stopwords)
        tdm = self._tfidf.fit_transform
                    (token_dict.values())
```

Kode Sumber 4.13 Perhitungan TF-IDF Kalimat Berita

Setelah mendapatkan nilai TF-IDF, akan dihitung nilai berdasarkan posisi kalimat dalam sebuah berita.

```
def position_score(self, i, size):
    relative_position = i / size
    if 0 < relative_position <= 0.1:
        return 0.17
    elif 0.1 < relative_position <= 0.2:
        return 0.23
    elif 0.2 < relative_position <= 0.3:
        return 0.14
    elif 0.3 < relative_position <= 0.4:
        return 0.08
    elif 0.4 < relative_position <= 0.5:
        return 0.05
    elif 0.5 < relative_position <= 0.6:
```

```
            return 0.04
        elif 0.6 < relative_position <= 0.7:
            return 0.06
        elif 0.7 < relative_position <= 0.8:
            return 0.04
        elif 0.8 < relative_position <= 0.9:
            return 0.04
        elif 0.9 < relative_position <= 1.0:
            return 0.15
        else:
            return 0
```

Kode Sumber 4.14 Perhitungan Bobot Berdasarkan Posisi Kalimat

Fitur selanjutnya adalah kemiripan kalimat dengan judul.

```
def headline_score(self, headline, sentence):
        count = 0.0
        for word in sentence:
            if word in title:
                count += 1.0
        score = count / len(title)
        return score
```

Kode Sumber 4.15 Perhitungan Bobot Berdasarkan Kemiripan Kalimat dengan Judul Berita

Pembobotan kalimat diakhiri penjumlahan semua bobot.

```
def score(self, article):
        headline = article[0]
        sentences = self.
                split_into_sentences(article[1])
        frequency_scores = self.frequency_scores
                (article[1])
        for i, s in enumerate(sentences):
            headline_score = self.headline_score
                    (headline, s) * 1.5
            length_score = self.length_score
                    (self.split_into_words
                    (s)) * 1.0
            position_score = self.position_score
```

```
                    (float(i+1),
                    len(sentences)) * 1.0
82.             frequency_score =
                    frequency_scores[i] * 4
83.             score = (headline_score +
                    frequency_score +
                    length_score +
                    position_score) / 4.0
84.             self._scores[s] = score
```

Kode Sumber 4.16 Perhitungan Bobot Total Kalimat

BAB V
UJI COBA DAN EVALUASI

Bab ini membahas mengenai hasil uji coba sistem yang telah dirancang dan dibuat. Uji coba dilakukan untuk mengetahui kinerja sistem dengan lingkungan uji coba yang telah ditentukan.

5.1 Lingkungan Uji Coba

Implementasi tugas akhir ini menggunakan desktop *personal computer* (PC) Dell Inspiron 15 7000. Sistem operasi yang digunakan adalah Windows 10 64-bit. PC yang digunakan memiliki spesifikasi Intel Core i7-7700HQ dengan kecepatan 2,8 GHz (8CPU), *Random Access Memory* (RAM) sebesar 8 GB, dan mempunyai *Graphics Processing Unit* (GPU) yaitu NVIDIA GeForce GTX 1050 Ti sebesar 8 GB.

PC dari sisi perangkat lunak memiliki spesifikasi antara lain menggunakan bahasa pemograman Python 3.6, dilengkapi dengan *library* antara lain Tensorflow, Keras, Anago, Pandas, NLTK, dan Spacy.

5.2 Dataset

Pada tugas akhir ini, data yang digunakan adalah data tweet dengan query “Lamborghini Koboi” (542 tweets), ”Yasonna Laoly” (425 tweets) dan “Gerhana Matahari Cincin” (4491 tweets) dengan bahasa Indonesia.

Data berita dikumpulkan dari berbagai portal berita *online*. Masing-masing topik terdiri dari 6 artikel berita yang berasal dari portal berita berbeda.

5.3 Hasil Praproses

Praproses dilakukan dua kali, yaitu pada tweet dan pada berita.

5.3.1 *Hasil Praproses Tweet*

Pada tahap praproses *tweet* pertama dilakukan penanganan pada karakter *unicode*, dan dilanjutkan dengan membersihkan *emoticon*, tautan, dan tanda baca. Hasil praproses tahap pertama dapat dilihat pada Tabel 5.1

Tabel 5.1 Hasil Praproses Tweet Tahap Pertama (Kolom "clean")

	Text	Clean
0	#News: Polres Jaksel Terus Telusuri Indikasi Peralihan Kepemilikan Lamborghini oleh Sang Koboi https://ift.tt/2Sxi6w7	Polres Jaksel Terus Telusuri Indikasi Peralihan Kepemilikan Lamborghini Sang Koboi
1	Manager Showroom Ungkap Pengusaha 'Koboi' Adalah Pemilik Asli Lamborghini https://ift.tt/2F3t0lu	Manager Showroom Ungkap Pengusaha Koboi Adalah Pemilik Asli Lamborghini
2	TERKUAK! Deretan Kasus yang Bayangi 'Koboi' Lamborghini https://www.youtube.com/watch?v=zx5XKPAOH1c …Berita lainnya hanya di aplikasi tvOne connect #tvOneNews pic.twitter.com/tAa9ems7Wy	TERKUAK Deretan Kasus Bayangi Koboi Lamborghini Berita aplikasi tvOne connect
3	Sejumlah Aib 'Koboi' Lamborghini Kemang yang Todong Pelajar Terungkap https://www.wowkeren.com/berita/tampil/00289731.html … pic.twitter.com/euPMgK7i5iKatholik. Kristen. Khonghucu. Isla...	Sejumlah Aib Koboi Lamborghini Kemang Todong Pelajar Terungkap

5.3.2 *Hasil Praproses Berita*

Berita yang dikumpulkan berbentuk kumpulan file XML (Extensible Markup Language). Agar data dapat diproses dengan lebih mudah, maka struktur data kumpulan berita dirubah menjadi bentuk Pandas *DataFrame* seperti pada Tabel 5.2.

Tabel 5.2 Contoh Berita dalam Bentuk DataFrame

No	Judul	Isi
1	Pengemudi Lamborghini Todong Pelajar di Kemang Positif Ganja	Jakarta, CNN Indonesia -- Pngendara mobil Lamborghini yang melakukan penodongan kepada dua pelajar di Kemang, Jakarta Selatan positif (…)
2	Bak Koboi Jalanan, Pengemudi Lam borghini Todong Pelajar di Kemang	Jakarta - Aksi arogan dilakukan oleh pengendara mobil mewah Lamborghini di kawasan Kemang, Jakarta Selatan. Pengendara mobil sport itu (…)

No	Judul	Isi
3	Geger Aksi Koboi Pengemudi Lamborghini Todong Pelajar di Kemang	Kasus penodongan 2 pelajar di Kemang, Jakarta Selatan, oleh seorang pengendara Lamborghini kembali menjadi contoh arogansi pemilik mobil (…)

Pada bentuk *DataFrame*, pada berita, diterapkan penanganan *unicode* dan karakter spesial, penghapusan tanda baca, *casefolding*, *tokenizing*, dan *stemming*. Perbedaan berita sebelum dilakukan praproses dan setelahnya dapat dilihat di Tabel 5.3.

Tabel 5.3 Contoh Berita setelah tahap praproses pertama

No	Judul	Isi
1	Pengemudi Lamborghini Todong Pelajar di Kemang Positif Ganja	jakarta cnn indonesia pngendara mobil lamborghini yang melakukan penodongan kepada dua pelajar kemang jakarta selatan positif menggunakan (…)
2	Bak Koboi Jalanan, Pengemudi Lam borghini Todong Pelajar di Kemang	jakarta aksi arogan dilakukan oleh pengendara mobil mewah lamborghini kawasan kemang jakarta selatan pengendara mobil sport itu bahkan sempat (…)
3	Geger Aksi Koboi Pengemudi Lamborghini Todong Pelajar di Kemang	kasus penodongan pelajar kemang jakarta selatan oleh seorang pengendara lamborghinikembali menjadi contoh arogansi pemilik mobil yang (…)

5.4 Skenario Uji Coba

Skenario uji coba berguna untuk menemukan parameter yang menghasilkan performa model dan metode yang paling optimal. Metode dan parameter yang tepat akan memberikan hasil yang lebih baik pada saat proses uji coba. Hasil terbaik dari suatu skenario uji coba akan digunakan untuk skenario uji coba berikutnya. Ada 3 macam skenario uji coba yang akan dilakukan yaitu:

1. Uji coba penggunaan NER untuk filtrasi *Tweets*
2. Uji coba penggunaan *Cluster Importance* pada pemilihan *cluster* pada ekstraksi *trending issue*
3. Uji coba penggunaan *trending issue* untuk pembobotan kalimat

5.4.1 *Uji coba penggunaan NER dalam filtrasi tweet*

Berikut adalah beberapa *tweet* hasil filtrasi dengan menggunakan NER:

1. Kalimat: Kok gaya koboi munkin krn orang Polisi menangkap pengemudi Lamborghini warna oranye menodong pelajar pistol Kemang Jakarta Selatan

Table 5.4 Contoh Hasil Filtrasii *Tweets* menggunakan NER

Entitas	**Type**
orang Polisi	OBJ
pelajar	GPE
Jakarta Selatan	GPE

2. Kalimat: Bak Koboi Jalanan Pengemudi Lamborghini Todong Pelajar Kemang detikcom DivHumas Polri KomnasHAM KomnasPA

Table 5.5 Contoh Hasil Filtrasii *Tweets* menggunakan NER

Entitas	**Type**
pelajar	GPE
DivHumas Polri KomnasHAM	OBJ

Berdasarkan contoh di atas, dapat dilihat bahwa kesalahan identifikasi entitas terdapat pada kata “pelajar” yang bukan merupakan lokasi geografis namun diidentifikasi sebagai “GPE”. Hal ini terjadi pada setiap kali kata “pelajar” muncul dalam kalimat.

Kesalahan yang berulang kali ini dapat dihilangkan dengan menggati data latih model Anago. Hal ini membuktikan bahwa harus dilakukan optimalisasi pada data latih Anago.

5.4.2 *Uji coba penggunaan Cluster Importance dan contain most tweet pada pemilihan cluster pada ekstraksi Trending issue*

Uji coba ini dilakukan dengan dataset *tweet* dengan topik "Lamborghini Koboi". Berikut adalah jumlah nilai *Cluster Importance* pada masing-masing *cluster*:

Table 5.6 Nilai CI masing-masing cluster

Cluster	Nilai CI
0	54.8897
1	81.0189
2	187.3932
3	34.4085
4	16.9889

Pada tabel di atas dapat dilihat bahwa *cluster* dengan nilai CI terbesar adalah *cluster* 2.

Adapun daftar isu pada *cluster* 2 tersebut adalah: 'ajar', 'todong', 'kemudi', 'lamborghini', 'koboi', 'kemang', 'jalan', 'aksi', 'senjata' (kata-kata sudah mengalami proses *stemming*).

Dapat dilihat bahwa isu yang terpilih merupakan isu yang sangat relevan dengan topik "Lamborghini Koboi".

5.4.3 *Uji coba penggunaan trending issue untuk pembobotan kalimat*

Hasil evaluasi dari peringkasan berita dengan topik "Lamborghini Koboi" tanpa dan dengan menggunakan pembobotan *trending issue* dapat dilihat pada Tabel 4.4.

Evaluasi yang dilakukan dengan mencari nilai ROUGE dan F1 setiap ringkasan berita otomatis terhadap *ground truth* (dengan metode Average dan Best).

Adapun *ground truth* yang digunakan pada tugas akhir ini adalah ringkasan berita yang dilakukan secara manual oleh pakar (tenaga profesional) Bahasa Indonesia.

Tabel 5.7 Evaluasi hasil ringkasan otomatis tanpa dan dengan pembobotan *trending issue*

		Tanpa Pembobotan *Trending Issue*			Dengan Pembobotan Trending Issue		
		P	R	F1	P	R	F1
Average	ROUGE-1	56.00	55.45	55.72	54.00	53.47	53.73
	ROUGE-2	46.46	46.00	46.23	47.47	47.00	47.24
	ROUGE-3	42.86	42.42	42.64	45.92	45.45	45.69
	ROUGE-4	39.18	38.78	38.97	44.33	43.88	44.10
	ROUGE-l	44.00	43.56	43.78	50.00	49.50	49.75
	ROUGE-w	39.21	15.42	22.14	44.06	17.33	24.88
Best	ROUGE-1	56.00	55.45	55.72	54.00	53.47	53.73
	ROUGE-2	46.46	46.00	46.23	47.47	47.00	47.24
	ROUGE-3	42.86	42.42	42.64	45.92	45.45	45.69
	ROUGE-4	39.18	38.78	38.97	44.33	43.88	44.10
Average		45.22	42.43	43.30	47.75	44.64	45.62

Berdasarkan kedua gambar di atas, terlihat bahwa hasil pengukuran ROUGE-1 ringkasan dengan pembobotan *trending issue* lebih rendah dibandingkan tanpa pembobotan *trending issue.*

Namun jika ditinjau lebih lanjut, dapat dilihat bahwa nilai-nilai ROUGE yang lain pada ringkasan otomatis dengan pembobotan *trending issue* berada sekitar 2% di atas ringkasan otomatis tanpa pembobotan *trending issue.*

BAB VI
KESIMPULAN DAN SARAN

Bab ini membahas tentang kesimpulan yang didasari oleh hasil uji coba yang telah dilakukan pada bab sebelumnya. Kesimpulan nantinya sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditujukan untuk pengembangan penelitian lebih lanjut di masa depan.

6.1 Kesimpulan

Pada pengerjaan tugas akhir ini setelah melalui tahap perancangan aplikasi, implementasi metode, serta uji coba, diperoleh kesimpulan sebagai berikut:

1. *Trending issue* dalam sebuah kumpulan tweet dapat ditemukan dengan bantuan *Named Entity Recognition* untuk melakukan filtrasi tweet berentitas (dengan nilai F1 92.35% yang menghasilkan tweets berentitas dan *Cluster Importance* untuk menentukan kelompok tweet yang paling representatif.

   Hal ini dibuktikan dengan filtrasi *Named Entity Recognition*, dan *trending issue* yang dihasilkan dari pemilihan *cluster* dengan *Cluster Importance* sangat relevan dengan topik yang diinginkan.
2. Dalam peringkasan berita multi dokumen, *trending issue* digunakan sebagai bobot pada tahap pembobotan kalimat dengan bobot relatif 4 kali bobot lainnya (setara dengan bobot TF-IDF)
3. Evaluasi penggunaan *Named Entity Recognition* dan *Cluster Importance* untuk menemukan *Trending issue* dalam dalam melakukan peringkasan berita multidokumen dapat dilakukan dengan mengukur nilai ROUGE dan F1.

   Di mana terbukti bahwa dengan penggunaan *Named Entity Recognition* dan *Cluster Importance* maka kualitas peringkasan berita multidokumen (nilai ROUGE dan F1) dapat meningkat hingga 2%.

6.2 Saran

Saran yang diberikan untuk pengembangan sistem pengenalan Penggunaan *Cluster Importance* dan *Named Entity Recognition* untuk Penentuan *Trending issue* dalam Peringkasan Berita Multidokumen, yaitu:

1. Membuat sistem yang dapat mengumpulkan berita dari banyak portal berita secara otomatis.
2. Membuat sistem yang bisa diakses secara mudah dan nyaman seperti berbasis *web* atau Android.

DAFTAR PUSTAKA

[1] "About Python," Python, [Online]. Available: https://www.python.org/about/. [Diakses 30 November 2018].

[2] "Keras: The Python Deep Learning library," Keras, [Online]. Available: https://keras.io/. [Diakses 30 November 2018].

[3] "TensorFlow," TensorFlow, [Online]. Available: https://www.tensorflow.org/. [Diakses 30 November 2018].

[4] F. N. Putra dan C. Fatichah, "Klasifikasi jenis kejadian menggunakan kombinasi NeuroNER dan Recurrent Convolutional Neural Network pada data Twitter," *Jurnal Ilmiah Teknologi Sistem Informasi,* vol. 4, no. 2, pp. 81-90, 2018.

[5] N. Hayatin dan C. Fatichah, "PEMBOBOTAN KALIMAT BERDASARKAN FITUR BERITA DAN TRENDING ISSUE UNTUK PERINGKASAN MULTI DOKUMEN BERITA," *JUTI: Jurnal Ilmiah Teknologi Informasi,* vol. 13, pp. 38-44, 2015.

[6] D. R. Radev, H. Jing, M. Styś dan D. Tam, "Centroid-based summarization of multiple documents," *Information Processing & Management,* vol. 40, no. 6, pp. 919-938, 2004.

[7] A. Purwarianti dan A. S. Wibawa, "Indonesian Named-entity Recognition for 15 Classes Using Ensemble Supervised Learning," *Procedia Computer Science ,* vol. 81, pp. 221-228, 2016.

[8] K. Sarkar, "Sentence Clustering-based Summarization of Multiple Text Documents," *International Journal of*

Computing Science and Communication Technologies, vol. 2, 2009.

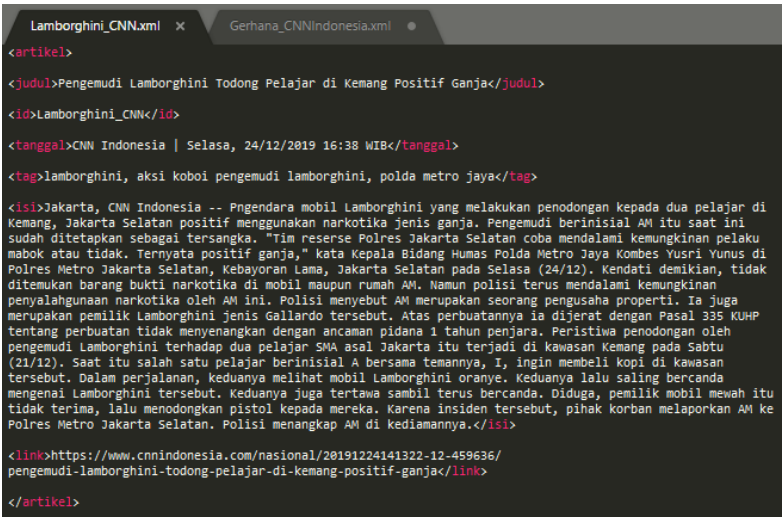
[9] J.-P. Mei dan C. Chen, "A new subtopic-based extractive approach for text summarization," *Knowl Inf Syst,* 2011.

[10] D. Kim, D. Kim, S. Kim, M. Jo dan E. Hwang, "SNS-based Issue Detection and Related News Summarization Scheme," 2014.

[11] G. Ifrim, B. Shi dan I. Brigadir, "Event Detection in Twitter using Aggressive Filtering and Hierarchical Tweet Clustering," *CEUR Workshop Proceedings ,* vol. 1150.

LAMPIRAN

L. 1 Dataset: contoh input berita



Lamborghini_CNN.xml | Gerhana_CNNIndonesia.xml

```
<artikel>

<judul>Pengemudi Lamborghini Todong Pelajar di Kemang Positif Ganja</judul>

<id>Lamborghini_CNN</id>

<tanggal>CNN Indonesia | Selasa, 24/12/2019 16:38 WIB</tanggal>

<tag>lamborghini, aksi koboi pengemudi lamborghini, polda metro jaya</tag>

<isi>Jakarta, CNN Indonesia -- Pngendara mobil Lamborghini yang melakukan penodongan kepada dua pelajar di
Kemang, Jakarta Selatan positif menggunakan narkotika jenis ganja. Pengemudi berinisial AM itu saat ini
sudah ditetapkan sebagai tersangka. "Tim reserse Polres Jakarta Selatan coba mendalami kemungkinan pelaku
mabok atau tidak. Ternyata positif ganja," kata Kepala Bidang Humas Polda Metro Jaya Kombes Yusri Yunus di
Polres Metro Jakarta Selatan, Kebayoran Lama, Jakarta Selatan pada Selasa (24/12). Kendati demikian, tidak
ditemukan barang bukti narkotika di mobil maupun rumah AM. Namun polisi terus mendalami kemungkinan
penyalahgunaan narkotika oleh AM ini. Polisi menyebut AM merupakan seorang pengusaha properti. Ia juga
merupakan pemilik Lamborghini jenis Gallardo tersebut. Atas perbuatannya ia dijerat dengan Pasal 335 KUHP
tentang perbuatan tidak menyenangkan dengan ancaman pidana 1 tahun penjara. Peristiwa penodongan oleh
pengemudi Lamborghini terhadap dua pelajar SMA asal Jakarta itu terjadi di kawasan Kemang pada Sabtu
(21/12). Saat itu salah satu pelajar berinisial A bersama temannya, I, ingin membeli kopi di kawasan
tersebut. Dalam perjalanan, keduanya melihat mobil Lamborghini oranye. Keduanya lalu saling bercanda
mengenai Lamborghini tersebut. Keduanya juga tertawa sambil terus bercanda. Diduga, pemilik mobil mewah itu
tidak terima, lalu menodongkan pistol kepada mereka. Karena insiden tersebut, pihak korban melaporkan AM ke
Polres Metro Jakarta Selatan. Polisi menangkap AM di kediamannya.</isi>

<link>https://www.cnnindonesia.com/nasional/20191224141322-12-459636/
pengemudi-lamborghini-todong-pelajar-di-kemang-positif-ganja</link>

</artikel>
```

L 1. Contoh berita topik “Lamborghini Koboi”
(diambil dari portal berita CNNIndonesia)

L. 2 Dataset: contoh *ground truth* berita

Polda Metro Jaya telah menahan pengemudi mobil lamborghini yang melakukan aksi menggunakan pistol bak seorang "koboi" terhadap dua pelajar di kawasan Jakarta Selatan. Peristiwa penodongan oleh pengemudi Lamborghini terhadap dua pelajar SMA asal Jakarta itu terjadi di kawasan Kemang pada Sabtu (21/12). Pasca penangkapan pemilik supercar Lamborghini berinisial AM yang beraksi bak koboi jalanan di kawasan Kemang, polisi melakukan penggeledahan rumah AM dan menyita sejumlah barang bukti lain. Polisi turut menyita sejumlah hewan langka dilindungi yang telah diawetkan. Selain itu polisi juga menemukan sejumlah amunisi peluru senjata api laras panjang yang diduga digunakan pelaku untuk berburu. Mobil Lamborghini yang dipakai pelaku mengancam 2 pelajar dengan senjata api di kawasan Kemang memiliki surat-surat yang diduga palsu.

L. 3 Hasil output berita (tanpa pembobotan *trending issue*)

Kabid Humas Polda Metro Jaya Kombes Yusri Yunus menyatakan tindakan pengemudi lamborgini melepaskan tembakan ke atas dengan pistol sebanyak tiga kali karena tidak terima dengan ucapan pelajar. Kedatangan anggota kepolisian dari Polres Metro Jakarta Selatan ini untuk mencari barang bukti lain dan mencari bukti dugaan pelanggaran hukum lainnya, selain aksi bkoboi pelaku di kawasan Kemang lalu. Polda Metro Jaya telah menahan pengemudi mobil lamborghini yang melakukan aksi menggunakan pistol bak seorang "koboi" terhadap dua pelajar di kawasan Jakarta Selatan. Pasca penangkapan pemilik supercar Lamborghini berinisial AM yang beraksi bak koboi jalanan di kawasan Kemang, polisi melakukan penggeledahan rumah AM dan menyita sejumlah barang bukti lain. "Anak saya selama beberapa menit ditodong, disumpah serapah (makian binatang), disuruh jongkok, tengkurap terus berkali-kali, A bilang bukan saya yang ngomong, pelaku bilang jangan banyak omong tengkurap

L. 4 Hasil output berita (dengan pembobotan *trending issue*)

Polda Metro Jaya telah menahan pengemudi mobil lamborghini yang melakukan aksi menggunakan pistol bak seorang "koboi" terhadap dua pelajar di kawasan Jakarta Selatan. Pasca penangkapan pemilik supercar Lamborghini berinisial AM yang beraksi bak koboi jalanan di kawasan Kemang, polisi melakukan penggeledahan rumah AM dan menyita sejumlah barang bukti lain. Kabid Humas Polda Metro Jaya Kombes Yusri Yunus menyatakan tindakan pengemudi lamborgini melepaskan tembakan ke atas dengan pistol sebanyak tiga kali karena tidak terima dengan ucapan pelajar. Kedatangan anggota kepolisian dari Polres Metro Jakarta Selatan ini untuk mencari barang bukti lain dan mencari bukti dugaan pelanggaran hukum lainnya, selain aksi koboi pelaku di kawasan Kemang lalu. Aksi koboi jalanan bermula saat pelajar berinisial A dan I tengah asyik bercanda di pinggir Jalan Kemang Selatan, Jakarta pada Sabtu sore 21 Desember 2019

BIODATA PENULIS

Bernama Reinardus Wandya K., penulis lahir di Kota Bandung pada 07 Desember 1997. Penulis yang sedang menyelesaikan masa Pendidikan S1 Informatika di Institut Teknologi Sepuluh Nopember (Surabaya) ini menempuh pendidikan mulai dari TK Pandu Bandung (2001-2003), SD Pandu Bandung (2003-2009), SMP Santa Angela Bandung (2009-2012), dan SMA Santa Angela Bandung (2012-2015).

Selama menempuh Pendidikan S1, penulis aktif dalam beberapa kegiatan, kepanitiaan dan organisasi di lingkup jurusan, diantaranya Staf Departemen Kewirausahaan HMTC ITS 2016-2017, Staf Departemen REEVA Schematics ITS 2016, dan Staf Ahli Humas Schematics ITS 2017. Penulis juga mengikuti beberapa kegiatan, kepanitiaan dan organisasi di luar jurusan, diantaranya Ketua Biro Catholic Community Departemen Eksternal KMK St. Ignasius Loyola ITS 2017/2018, Wakil Ketua Forum Daerah Bandung ITS (Forda BandITS), dan anggota Komunitas Heman Salvation Ministry (HSM).

Penulis dapat dihubungi melalui e-mail reinard7@gmail.com