TUGAS AKHIR – KS141501

NORMALISASI TEKS MEDIA SOSIAL MENGGUNAKAN WORD2VEC, LEVENSHTEIN DISTANCE, DAN JARO-WINKLER DISTANCE

SOCIAL MEDIA TEXT NORMALIZATION USING WORD2VEC, LEVENSHTEIN DISTANCE, AND JARO-WINKLER DISTANCE

STEZAR PRIANSYA

NRP 5213 100 131

Dosen Pembimbing:

Renny Pradina K., S.T., M.T., SCJP

DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Informasi

Institut Teknologi Sepuluh Nopember

Surabaya 2017

TUGAS AKHIR – KS141501

NORMALISASI TEKS MEDIA SOSIAL MENGGUNAKAN WORD2VEC, LEVENSHTEIN DISTANCE, DAN JARO-WINKLER DISTANCE

STEZAR PRIANSYA

NRP 5213 100 131

Dosen Pembimbing:

Renny Pradina K., S.T., M.T., SCJP

DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Informasi

Institut Teknologi Sepuluh Nopember

Surabaya 2017

FINAL PROJECT – KS 141501

SOCIAL MEDIA TEXT NORMALIZATION USING WORD2VEC, LEVENSHTEIN DISTANCE, AND JARO-WINKLER DISTANCE

STEZAR PRIANSYA
NRP 5213 100 131

Supervisor:
Renny Pradina K., S.T., M.T., SCJP

DEPARTMENT OF INFORMATION SYSTEMS
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2017

LEMBAR PENGESAHAN

NORMALISASI TEKS MEDIA SOSIAL MENGGUNAKAN WORD2VEC, LEVENSHTEIN DISTANCE, DAN JARO-WINKLER DISTANCE

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Departemen Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

STEZAR PRIANSYA
NRP. 5213 100 131

Surabaya, 19 Juli 2017

KEPALA DEPARTEMEN SISTEM INFORMASI

Dr. Ir. Aris Tjahyanto, M.Kom
NIP.19650310 199102 1 001

LEMBAR PERSETUJUAN

NORMALISASI TEKS MEDIA SOSIAL MENGGUNAKAN WORD2VEC, LEVENSHTEIN DISTANCE, DAN JARO-WINKLER DISTANCE

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat

Memperoleh Gelar Sarjana Komputer

pada

Jurusan Sistem Informasi

Fakultas Teknologi Informasi

Institut Teknologi Sepuluh Nopember

Oleh :

STEZAR PRIANSYA

NRP. 5213 100 131

Disetujui Tim Penguji: Tanggal Ujian: 10 Juli 2017
Periode Wisuda: September 2017

Renny Pradina K., S.T., M.T. **(Pembimbing I)**

Nur Aini R., S.Kom., M.Sc.Eng., Ph.D. **(Penguji I)**

Radityo Prasetianto W., S.Kom., M.Kom. **(Penguji II)**

NORMALISASI TEKS MEDIA SOSIAL MENGGUNAKAN WORD2VEC, LEVENSHTEIN DISTANCE, DAN JARO-WINKLER DISTANCE

Nama Mahasiswa	**:**	**Stezar Priansya**
NRP	**:**	**5213 100 131**
Jurusan	**:**	**Sistem Informasi**
Pembimbing 1	**:**	**Renny Pradina K., S.T., M.T., SCJP**

ABSTRAK

Sebagian besar pengguna internet di Indonesia menggunakan media sosial untuk mendapatkan pembaruan informasi secara rutin. Namun, frekuensi penggunaan media sosial yang tinggi, tidak sebanding dengan ejaan kalimat tidak baku (informal) yang digunakan dalam mengisi konten media sosial dengan maksud untuk memudahkan komunikasi. Ejaan bahasa yang digunakan tidak hanya mengganggu pengguna media sosial namun juga mempengaruhi pengolahan terhadap data konten media sosial tersebut yang biasa disebut Natural Language Processing.

Penelitian sebelumnya mencoba mengajukan konsep word2vec yang terbukti mampu menemukan cara untuk melakukan representasi vektor dari sebuah kata dengan waktu yang relative cepat dan dengan dataset yang cukup besar dan juga terdapat solusi berupa pembenaran/normalisasi teks menggunakan algoritma edit distance/levenshtein dan jaro-winkler distance.

Hasil yang dari training model word2vec yang didapatkan adalah model ke-8 dengan hasil akurasi 25%. Selain itu parameter yang sangat menentukan proses training ialah learning algorithm. Untuk pengujian sampel perngoreksian data, akurasi paling baik adalah 79,56% dengan threshold sebesar 70%.

Kata Kunci: Word Embedding, Word2Vec, Levenshtein Distance, Media Sosial, Natural Language Processing

SOCIAL MEDIA TEXT NORMALIZATION USING WORD2VEC, LEVENSHTEIN DISTANCE, AND JARO-WINKLER DISTANCE

Nama Mahasiswa	**:**	**Stezar Priansya**
NRP	**:**	**5213 100 131**
Jurusan	**:**	**Sistem Informasi**
Pembimbing 1	**:**	**Renny Pradina K., S.T., M.T., SCJP**

ABSTRACT

Most internet users in Indonesia use social media to obtain information regularly. However, high frequency of social media usage, is not comparable to non-standard (informal) sentences spelling that used in the social media content to facilitate communication nowadays. Those spelling language not only disrupts social media users but also updating process of social media content data that commonly called Natural Language Processing.

Previous research tried to propose word2vec concept which is proved able to find a way to perform vector representation of a word fairly fast with a large enough data sets. And there is also a solution like justification / normalization of the text using long distance editing algorithm/levenshtein and jaro-winkler distance editing algorithms.

The result of word2vec training model is the 8th model with 25% accuracy. In addition, the parameters that determine the training process is learning algorithm. For testing of data correction samples, the best accuracy is 79.56% with a threshold of 70%.

Keyword: Word Embedding, Word2Vec, Levenshtein Distance, Social Media, Natural Language Processing

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji dan syukur penulis tuturkan ke hadirat Allah SWT, Tuhan Semesta Alam yang telah memberikan kekuatan dan hidayah-Nya kepada penulis sehingga penulis mendapatkan kelancaran dalam menyelesaikan tugas akhir ini yang merupakan salah satu syarat kelulusan pada Departemen Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya.

Terima kasih penulis sampaikan kepada pihak-pihak yang telah mendukung, memberikan saran, motivasi, semangat, dan bantuan baik berupa materiil maupun moril demi tercapainya tujuan pembuatan tugas akhir ini. Tugas akhir ini tidak akan pernah terwujud tanpa bantuan dan dukungan dari berbagai pihak yang sudah melauangkan waktu, tenaga dan pikirannya. Secara khusus penulis akan menyampaikan ucapan terima kasih yang sebanyak-banyaknya kepada:

1. Bapak Priyono dan Ibu Dwi Kuswantini selaku kedua orang tua serta Satriya Pratama, Shafandi Priandana dan Syafarul Priwantoro selaku saudara kandung dari penulis yang tiada henti memberikan dukungan dan semangat.
2. Ibu Renny Pradina K., S.T., M.T., SCJP, selaku dosen pembimbing dan sebagai narasumber yang senantiasa meluangkan waktu, memberikan ilmu dan petunjuk, serta memotivasi untuk kelancaran tugas akhir.
3. Ibu Nur Aini R., S.Kom., M.Sc.Eng., Ph.D., dan Bapak Radityo Prasetianto W., S.Kom., M.Kom., selaku dosen penguji yang telah memberikan saran dan kritik untuk perbaikan tugas akhir.
4. Seluruh dosen Jurusan Sistem Informasi ITS yang telah memberikan ilmu yang bermanfaat kepada penulis.
5. Marina Safitri, Delina Rahayu dan M. Zuhri, selaku sahabat yang telah memberikan ilmu dan pencerahan terkait pengerjaan buku dan sistem dalam tugas akhir ini.

6. Marina Safitri, Shania Olivia, Pramita Lucianna, Provani Winda, Delina Rahayu, Chandra Surya, Ikhwan Aziz, M. Fahmi, Nadya Chandra, Caesar Gilang, Rani Oktavia, Bintang Setyawan, Tetha Valianta, Alvin Rahman yang telah mendukung dan menemani penulis dari masa mahasiswa baru hingga tugas akhir ini dapat diselesaikan.
7. Wisnu, Risa, Niko, Adnan, Harun, Lutfi, Kusnanta, serta para penghuni laboratorium ADDI yang telah menemani pengerjaan tugas akhir ini selama di laboratorium.
8. Aziz, Esther, April, Novi, Prima, Probo dan ASETSMALA selaku sahabat penulis semasa SMA, (hingga kini) yang memberikan dukungan serta doa dan bantuan.
9. Rekan-rekan BEM FTIf, HMSI dan BELTRANIS yang telah memberikan banyak kenangan manis dan pahit semasa kuliah.
10. Berbagai pihak yang tidak bisa disebutkan satu persatu yang telah turut serta menyukseskan penulis dalam menyelesaikan tugas akhir.

Penyusunan laporan ini masih jauh dari kata sempurna sehingga penulis menerima adanya kritik maupun saran yang membangun untuk perbaikan di masa yang akan datang. Semoga buku tugas akhir ini dapat memberikan manfaat bagi pembaca.

Surabaya, 04 Juli 2017

Penulis,

Stezar Priansya

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR KODE

Halaman ini sengaja dikosongkan

DAFTAR TABEL

BAB I
PENDAHULUAN

Pada bab pendahuluan akan diuraikan proses identifikasi masalah penelitian yang meliputi latar belakang masalah, perumusan masalah, batasan masalah, tujuan tugas akhir, manfaat kegiatan tugas akhir dan relevansi terhadap pengerjaan tugas akhir. Berdasarkan uraian pada bab ini, harapannya gambaran umum permasalahan dan pemecahan masalah pada tugas akhir dapat dipahami.

1.1 Latar Belakang Masalah

Indonesia merupakan salah satu negara dengan angka penggunaan internet terbesar di dunia. Menurut hasil survei APJII (Asosiasi Penyelanggara Jasa Internet Indonesia) pada tahun 2016, penetrasi pengguna internet di Indonesia mencapai 132,7 juta[1]. Hal ini merupakan kenaikan angka yang cukup besar dibandingkan pengguna internet pada tahun 2014 yang mencapai 88,1 juta[1]. Komposisi pengguna internet Indonesia peringkat 3 teratas berdasarkan usia adalah rentang usia 35 – 44 tahun, usia 25 – 34 tahun, dan usia 10 – 24 tahun[1]. Para pengguna internet di Indonesia memiliki alasan sangat beragam dalam menggunakan internet, tetapi alasan yang paling umum digunakan adalah untuk mendapatkan pembaruan informasi sebanyak 31,3 juta[1]. Untuk mendapatkan pembaruan informasi, pengguna mengakses berbagai macam konten yang tersedia pada internet. Media sosial merupakan konten terbanyak yang diakses oleh pengguna internet Indonesia dengan angka mencapai 129,2 juta pengguna[1]. Dari hal ini dapat disimpulkan bahwa 97,4% pengguna internet Indonesia menggunakan internet untuk mendapatkan informasi melalui media sosial.

Saat ini, media sosial di Indonesia dapat dianggap sebagai kebutuhan utama untuk pembaruan informasi dikarenakan maraknya informasi – informasi terbaru yang disebarkan melalui media seperti *Facebook*, *Twitter*, *Instagram*, dan lain

lain. *Facebook* sendiri merupakan media sosial paling populer di Indonesia dengan pengguna sebanyak 71,6 juta, kemudian *Twitter* yang berada pada peringkat 5 dengan pengguna sebanyak 7,2 juta[1]. Namun, pengguna media sosial yang banyak tidak sebanding dengan ejaan kalimat yang digunakan oleh pengguna media sosial tersebut. Ejaan bahasa yang digunakan pada media sosial terutama *Facebook* dan *Twitter* cenderung menggunakan ejaan yang tidak baku (informal) dengan alasan agar lebih komunikatif, santai serta akrab dalam berkomunikasi[2]. Tentunya hal ini akan membuat penyebaran informasi menjadi sedikit terganggu dikarenakan banyaknya ragam ejaan bahasa yang ada pada media sosial *Facebook* dan *Twitter*.

Informasi – informasi yang terdapat pada *Facebook* dan *Twitter* seringkali tidak hanya digunakan untuk melakukan pembaruan informasi saja namun juga seringkali digunakan kalangan akademisi untuk melakukan pemrosesan data teks agar menjadi sebuah *big insight* dan dapat berdampak bagi dunia akademisi maupun dunia secara global. Dalam melakukan pemrosesan data teks yang tidak terstruktur ini, para peneliti menggunakan metode yang disebut dengan Natural Language Processing atau yang biasa disingkat NLP. NLP adalah sebuah metode pembentukan model komputasi bahasa sebagai bentuk interaksi antara manusia dan komputer dengan perantaraan bahasa alami[3]. NLP berupaya untuk dapat memecahkan masalah untuk memahami bahasa alami manusia, dengan segala aturan gramatika dan semantiknya, serta mengubah bahasa tersebut menjadi repersentasi formal yang dapat diproses oleh komputer[3].

NLP pada setiap praktiknya memiliki beberapa tantangan antara lain, penandaan kelas kata yang sulit, segmentasi teks yang sulit, disambiguitas makna kata, ambiguitas sintaksis, masukan tidak sempurna atau tidak teratur, serta pertuturan dari kalimat yang susah diidentifikasi konteksnya[3]. Artinya, pemrosesan data teks pada media sosial *Facebook* dan *Twitter* yang notabene memiliki ejaan bahasa tidak baku yang menyebabkan

disambiguasi makna kata serta penulisan tidak baku/sempurna yang menyebabkan ambiguitas sintaksis dapat menjadi tantangan yang besar ketika menggunakan NLP. Banyak orang yang melakukan NLP tanpa memperhatikan permasalahan diatas, sehingga menyebabkan representasi dari setiap kata dalam satu korpus yang seharusnya memiliki makna yang sama menjadi berbeda. Hal ini akan dapat menyebabkan kinerja algoritma didalam NLP menjadi kurang efektif.

Banyak hal telah dilakukan untuk menghindari permasalahan tersebut seperti melakukan *data pre-processing* sebelum masuk kedalam *data processing* agar data teks yang diolah menjadi normal. Namun, hal tersebut sangat sulit dilakukan untuk ragam kata tidak baku yang ada di dalam Bahasa Indonesia karena sangat banyak variasi kata, semisal kata "Terima Kasih" memiliki beberapa padanan kata yang sama konteksnya yaitu "Makasih", "Makasi", "Maaci". Apabila hanya *data pre-processing* tidak dilakukan maka kata tersebut akan dihitung sebagai kata yang berbeda, namun meskipun juga telah dilakukan *data pre-processin*g dengan menghilangkan *stopwords*, maka kata "Makasi" dan "Maaci" tidak akan hilang dikarenakan tidak terdapat di kamus *stopwords* dan kita harus menambahkannya sendiri. Oleh karena itu, penulis mengajukan penelitian terkait normalisasi teks pada media sosial dengan pemodelan data media sosial menggunakan *Word2Vec*, *Levensthein Distance*, dan *Jaro-Winkler Distance*. Model *word2vec* akan membantu memberikan rekomendasi kata terdekat yang sesuai dengan kesamaan penggunaan, kedekatan konteks, maupun relasi antar kata-kata. Kemudian untuk semakin menambah keakuratan dalam menormalisasi kata teks maka ditambahkan algoritma *levensthein distance* dan *jarko-winkler distance* untuk mengukur kesamaan sintaks dari tiap kata yang ingin di normalisasi dengan rekomendasi kata-kata dari model *wod2vec.*

Dengan adanya tugas akhir ini diharapkan dapat melakukan normalisasi teks bahasa Indonesia sehingga mampu memberikan rekomendasi penyempurnaan kata dari kata yang

tidak baku sehingga kedepannya penggunaan NLP terutama untuk Bahasa Indonesia dari data media sosial akan menjadi lebih efektif lagi.

1.2 Perumusan Masalah

Rumusan Masalah dari penelitian ini adalah:

1. Bagaimana mendapatkan data posting dari *Facebook* dan *Twitter* serta komentar posting *Facebook*?
2. Bagaimana membuat model *word2vec* dari data *Facebook* dan *Twitter*?
3. Bagaimana menampilkan rekomendasi kata baku untuk sebuah pembenaran kata masukan yang tidak baku?

1.3 Batasan Masalah

1. Studi kasus yang digunakan pada penelitian ini adalah media sosial *Facebook* dan *Twitter*.
2. Data yang akan digunakan pada pengerjaan Tugas Akhir ini sebagai berikut:
   - Data posting dan komentar pada halaman *Facebook* dari bulan Agustus 2015 hingga Maret 2017
   - Data posting dari akun *Twitter* dari bulan Oktober 2016 hingga Maret 2017.
3. Penelitian ini merupakan eksperimen dari penerapan *word2vec* menggunakan data posting media sosial berbahasa Indonesia.
4. Bentuk pembenaran kata yang dilakukan hanya sebatas pembenaran sintaks apabila terdapat kesalahan penulisan tidak pada kata yang mengandung unsur bahasa asing dan juga tidak secara konteks yang sama.
5. Proses pembenaran kata masih belum memperhatikan waktu komputasi.
6. Pengujian yang dilakukan menggunakan sampel data dari dataset yang sudah tersedia.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Menerapkan metode *crawling* menggunakan *library Facebook4J* serta *Twitter4J* dengan memanfaatkan *Facebook* serta *Twitter API* untuk mendapatkan data yang diinginkan.
2. Membuat model *word2vec* dari data posting media sosial.
3. Memberikan rekomendasi pembenaran kata terhadap kata tidak baku yang telah dimasukkan.

1.5 Manfaat Penelitian

1. Bagi penulis, untuk mengetahui pemodelan dari sekumpulan dataset posting media sosial menggunakan *word2vec* serta mengetahui kata-kata terdekat dari setiap kata yang ada dan juga rekomendasi pembenaran katanya.
2. Bagi masyarakat, sebagai bentuk penelitian awal yang memungkinkan untuk terdapat penelitian-penelitian selanjutnya dan dapat dikembangkan kedalam beberapa hal seperti bisnis maupun rancang bangun aplikasi yang memanfaatkan pemodelan *word2vec* ini.

1.6 Relevansi

Relevansi tugas akhir ini terhadap laboratorium Akuisisi Data dan Diseminasi Informasi (ADDI) adalah karena tugas akhir ini berkaitan dengan penerapan mata kuliah bidang keilmuan laboratorium ADDI. Mata kuliah tersebut antara lain Sistem Cerdas, Sistem Pendukung Keputusan, dan Penggalian Data dan Analitika Bisnis.

Halaman ini sengaja dikosongkan

BAB II
TINJAUAN PUSTAKA

Pada bab ini akan membahas mengenai penelitian sebelumnya yang berhubungan dengan tugas akhir dan teori - teori yang berkaitan dengan permasalahan tugas akhir

2.1 Studi Sebelumnya

Tabel 2.2.1 menampilkan daftar penelitian sebelumnya yang mendasari tugas akhir ini

Tabel 2.2.1 Studi Sebelumnya

1. Efficient Estimation of Word Representations in Vector Space
Penulis/Tahun/Sumber: Tomas Mikolov, Greg Corrado, Kai Chen, dan Jeffrey Dean; 2013[4]
Metode: *- Skip Gram* *- Continuous Bags of Word*
Kesimpulan: Penelitian ini berhasil menemukan cara untuk melakukan representasi vektor dari sebuah kata dengan waktu yang relatif cepat dan dengan dataset yang cukup besar. Kemudian, vektor kata tidak hanya ditemukan kesamaan sintaks saja namun juga kesamaan semantik dari kata tersebut. Peneliti juga membandingkan akurasi hasilnya dengan teknik *neural networks* yang mana memiliki hasil yang lebih baik.
2. Distributed Representations of Words and Phrases and their Compositionality
Penulis/Tahun/Sumber: Tomas Mikolov, Greg Corrado, Kai Chen, dan Jeffrey Dean; 2013[5]

Metode: - *Skip Gram* - *Contiunous Bags of Word* - *Negative Sampling* - *Hierarchical Softmax*
Kesimpulan: Pada penelitian ini, penulis lebih memberikan tambahan agar model *Continuous Skip Gram* lebih memiliki representasi vektor yang lebih berkualitas dan meningkatkan kecepatan dari *training* dataset. Dalam penelitian ini juga membuktikan bahwa tambahan metode dapat diaplikasikan terhadap dataset yang cukup besar dengan waktu yang cukup singkat. Kemudian, pada penelitian ini menggunakan pendekatan *hierarchical softmax* dan *negative sampling*. Penulis menyarankan pada saat pembuatan model *word2vec* menggunakan *Skip-gram* dan *Negative Sampling.* Kemudian penulis juga membuat *library* yang bernama *Word2Vec* untuk mengimplementasikan metodenya.
3. From Word Embeddings to Item Recommendation
Penulis/Tahun/Sumber: Makbule Gulcin Ozsoy; 2016[6]
Metode: - *Word2vec* - *Content-based* - *Collaborative Filtering*

Kesimpulan: Pada penelitian ini membahas pembuatan sistem rekomendasi tempat berdasarkan lokasi jejaring sosial (LBSN) dengan menggunakan dataset *check-in foursquare* dengan menggunakan model *Word2Vec*. Selain itu, *word2vec* juga dikombinasikan menggunakan metode rekomendasi seperti *content-based* dan *collaborative filtering*. Penulis mengatakan bahwa *word2vec* cukup menjanjikan apabila digunakan sebagai metode sistem rekomendasi.
4. Adaptating the levenshtein distance to contextual spelling correction
Penulis/Tahun/Sumber: Aouragh Si Lhoussain, Gueddah Hicham, dan Yousfi Abdellah; 2015[5]
Metode: *- Bi-gram Language Model* *- Levenshtein distance*
Kesimpulan: Penelitian ini membahas bagaimana menyelesaikan masalah pengejaan yang salah pada bahasa Arab menggunakan algoritma *Levensthein Distance* dan *Bi-gram model*. Hasil yang diperoleh adalah kata yang salah dapat dikoreksi secara efektif.
5. Automatic spelling correction for Russian social media texts
Penulis/Tahun/Sumber: Sorokin, A A dan Shavrina, T O; 2016 [6]
Metode: *- Edit Distance* *- Logistic Regression*

Kesimpulan:
Penelitian ini membahas tentang bagaimana membuat sebuah sistem untuk mengoreksi kesalahan penulisan pada bahasa Rusia secara otomatis. Peneliti menggunakan metode *Edit Distance* untuk memilih kandidat kata yang tepat untuk pembenarannya. Kemudian, kandidat kata yang benar di ranking ulang kembali menggunakan *Logistic Regression*. *F1-Measure* dari hasil penelitian ini juga cukup tinggi yaitu 75%.

6. Text Normalization in Social Media: Progress, Problems and Applications for a Pre-Processing System of Casual English

Penulis/Tahun/Sumber:
Eleanor Clarka dan Kenji Arakia; 2011 [9]

Metode:
- *Rule-Based*

Kesimpulan:
Penelitian ini bertujuan untuk membuat sebuah sistem yang bernama CECS (*Casual English Conversion System*) yang mencakup metode *automated tokenization*, *word matching* dan teknik penggantian kata untuk menormalisasi teks berbahasa inggris pada media sosial *Twitte*r. Hasil penelitian menunjukkan bahwa rata-rata kesalahan setiap kalimat yang telah dibenarkan menurun secara substansial dari sekitar 15% menjadi kurang dari 5%.

7. Lexical Normalization for Social Media Text

Penulis/Tahun/Sumber:
Bo Han, Paul Cook, dan Timothy Baldwin; 2013[7]

Metode:
- *Lexcion*
- *Edit Distance*
- *Classifier*

Kesimpulan:
Data yang digunakan adalah data yang berasal dari *twitter* dan menggunakan *classifier* untuk mendeteksi *lexical variants* kemudian membentuk sebuah kandidat – kandidat kata yang benar berdasarkan *morphophonemic similarity*. Paper ini menggunakan *token-based lexical variant detection* dan *dictionary-based lexical normalization*. Hasil yang didapatkan menggunakan *individual dictionaries* adalah dengan *precission* 0.982 menggunakan *GHM-dict*. Untuk yang *combined dictionaries* hasil *F-score* terbaik adalah 0.723 menggunakan *HB-dict+GHM-dict+S-dict*. Kemudian untuk hybrid approaches dengan menggunakan *HB-dict+GHM-dict+S-dict+HB-norm* menghasilkan *recall* sebesar 0.791.

2.2 Dasar Teori

2.2.1 *Natural Language Processing*

Natural Language Processing (NLP) dapat didefinisikan sebagai pemrosesan otomatis/semi otomatis untuk mengkaji interaksi komputer dengan bahasa alami manusia yang seringkali digunakan dalam kehidupan sehari-hari. Proses komputasi bahasa direpresentasikan sebagai suatu rangkaian simbol yang memenuhi aturan tertentu. Dalam proses *natural language processing* terdapat beberapa kesulitan diantaranya sering terjadi ambiguitas atau makna ganda dan jumlah kosa kata dalam bahasa alami yang semakin besar dan berkembang dari waktu ke waktu. Jika dibandingkan dengan manusia, masalah ambiguitas tersebut didasarkan pada analisis konteks yang didukung oleh pengetahuan yang dimilikinya. NLP memodelkan pengetahuan terhadap bahasa, baik dari segi kata, bagaimana kata-kata bergabung menjadi suatu kalimat dan konteks kata dalam kalimat. Terdapat beberapa disiplin ilmu dari NLP yaitu sebagai berikut [8]:

- Fonetik/fonologi: Berhubungan dengan suara yang mampu menghasilkan kata yang dapat dikenali (*speech based system*)
- Morfologi : Pengetahuan terkait perbedaan tentang kata dan bentuknya (pembedaan kata dasar, prefiks dan sufiksnya)
- Sintaksis: Pengetahuan terkait urutan kata dalam pembentukan kalimat (urutan kalimat dengan subjek dan predikatnya)
- Semantik: Pengetahuan terkait arti suatu kata dan bagaimana kata tersebut membentuk arti kata dari kaliamat yang utuh
- Pragmatik: Pengetahuan tentang konteks kalimat/kata yang berhubungan erat dengan suatu keadaan

Hingga saat ini terdapat beberapa kajian yang telah dilakukan peneliti dalam bidang *Natural Language Processing* diantaranya adalah [9] : *anaphora(linguistics), automatic summarization, collocation extraction, Google Neural Machine Translation, Language identification, lemmatisation, linguistic empathy, speech tagging, phrase chunking, NER (Named-entity-recognation), semantics & text extraction, morphological segmentation, natural language understanding, discourse analysis*, dan *question answering*.

2.2.2 *Word Embedding*

Word Embedding adalah istilah yang digunakan dalam merepresentasikan sebuah kata kedalam sebuah *low-dimentional vector* (misal: 100 dimensi). Istilah *word embedding* awalnya dikenal dengan *word representation* yang muncul pada tahun 1986 oleh Rumelhart, Hinton, dan Williams[10]. Metode ini telah sukses digunakan dalam hal *statistical language modelling*[11]. Word embedding juga dikenal dengan nama Distributed Representation karena memiliki kerapatan, menggunakan *low-dimentional vector* dan *real valued*[12]. Ketika proses *training* dilakukan pada word embedding, maka hasil yang didapatkan salah satunya berupa kemiripan atau kedekatan antara kata maupun juga relasi yang

lainnya. Salah satu contoh *word embedding* yang cukup baru dan sangat populer adalah *word2vec*[13].

2.2.3 *Word2Vec*

Word2Vec adalah sebuah proyek yang mana mengimplementasikan metode *word embedding/word representation* yang dibuat oleh Tomas Mikolov, dkk[4][5]. Word2vec bertujuan untuk merepresentasikan kata-kata kedalam *vector* yang rapat dan memliki dimensi yang rendah.

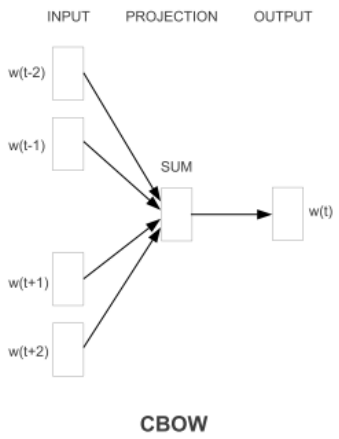
Saat ini, telah banyak model yang bertujuan untuk merepresentasikan kata-kata secara kontinyu seperti LDA *(Latent Dirichlet Allocation)* dan LSA *(Latent Semantic Analysis)*, namun Tomas Mikolov menggunakan sebuah metode yang mana hasil memodifikasi NNLM *(Neural Networks Language Model)* yang mana *neural networks* dapat bekerja dengan baik dalam menjaga keteraturan linear antara kata-kata dibandingkan dengan LSA dan lebih baik dalam menangani data besar dibandingkan dengan LDA[4]. Model yang digunakan ialah *Continuous Bag-of-Words* dan *Continuous Skip-gram*. Oleh karena itu, *word2vec* memiliki banyak keunggulan dibandingkan dengan metode *word representation* lainnya, seperti lebih cepat dalam proses *training*-nya, lebih efisien, dapat menangani dataset dengan skala yang besar. Berdasarkan eksperimen, keputusan krusial yang mampu mempengaruhi kinerja dari word2vec adalah pemilihan model yang digunakan, ukuran dari *vector, subsampling rate*, dan *training window*[13].

2.2.3.1 Arsitektur Word2Vec (*Log-Linear Models*)

Word2vec memiliki dua arsitektur model yang keduanya sama sama menggunakan model *log-linear*.

2.2.3.1.1 Continuous Bag-of-Words Model (CBOW)

Model ini adalah pengembangan dari *neural net language model* dengan mempunyai *input layer*, *projection layer*, dan *output layer* seperti gambar berikut ini.



Gambar 2.2.1 Arsitektur CBOW[4]

Pada Gambar 2.2.1 terlihat w(t-2), w(t-1), w(t+1), dan w(t+2) adalah kata-kata sebelum dan sesudah (disekitar) yang menjadi *input layer* kemudian masuk kedalam *projection layer* berupa SUM dan menghasilkan satu kata yang diprediksi sering muncul bersamaan dengan kata *input* tadi. CBOW pada *word2vec* digunakan untuk memprediksi sebuah kata (t) berdasarkan konteks kata-kata yang ada di sektiarnya (c), tujuan utamanya adalah memperbesar peluang P(t|c) pada sekumpulan *training* dataset. Contohnya adalah terdapat kalimat seperti "warga dari X telah melakukan demo hari ini", maka hasil dari X yang mana merupakan kata target (t) kemungkinan adalah nama tempat, kota, ataupun negara yang berhubungan secara semantik. Mikolov et al. juga telah menggambarkan *training complexity* dari CBOW pada persamaan berikut[4].

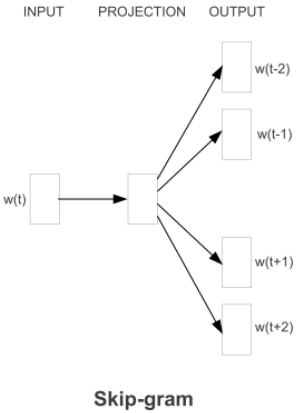
$$Q = N \times D + D \times \log_2(V) \quad \text{(persamaan 1)}$$

Kompleksitas pelatihan (Q) dihitung dengan hasil kali antara kata sebelum (N) dan hasil representasi kata (D) ditambah dengan hasil representasi kata (D) di kali hasil logaritma basis 2 dari *vocabulary* (V). Keuntungan menggunakan CBOW adalah proses *training* lebih cepat dibandingkan dengan *Skip-*

gram dan juga memiliki akurasi yang cukup baik pada kata yang jarang muncul.

2.2.3.1.2 Continuous Skip-Gram Model

Model *Skip-Gram* merupakan kebalikan dari Model CBOW. *Skip-gram* memiliki satu target kata sebagai masukannya dan memiliki luaran berupa konteks kata yang memiliki kedekatan dengan kata masukannya. Sehingga, *Skip-gram* memiliki gambaran arsitektur seperti ini.



Gambar 2.2.2 Arsitektur *Skip-Gram*[4]

Pada Gambar 2.2.2 terlihat bahwa *Skip-gram* adalah kebalikan dari CBOW, dimana w(t) sebagai sebuah kata *input* memasuki *projection layer* kemudian menghasilkan kata-kata disekitarnya, yaitu sebelum (w(t-1), w(t-2)) dan sesudah (w(t+1), w(t+2)). Contoh cara kerja dari *Skip-gram* ini adalah terdapat sebuah kalimat "aku sangat suka ….. ke Surabaya". Apabila kata dalam titik-titik tersebut adalah "pergi" dan kemudian dimasukkan sebagai kata masukan dari model *Skip-gram*, maka luaran yang dihasilkan adalah kata-kata yang sering muncul disekitarnya, seperti "aku", "sangat", "suka", "ke", dan "Surabaya". Selain itu, Mikolov et al. juga membuat menuliskan persamaan yang mana mencerminkan *training*

complexity dari model *Skip-gram* ini yang dapat dilihat pada gambar dibawah ini[4].

$$Q = C \times (D + D \times \log_2(V)) \quad \text{(persamaan 2)}$$

Kompleksitas pelatihan (Q) dihitung dengan hasil kali antara jarak maksimal antar kata (C) dan hasil penjumlahan dari representasi kata (D) ditambah dengan hasil representasi kata (D) di kali hasil logaritma basis 2 dari *vocabulary* (V).Menurut Mikolov, *Skip-gram* memiliki keuntungan dapat bekerja dengan baik pada data yang tidak terlalu besar dan dapat merepresentasikan kata-kata yang jarang/frasa dengan cukup baik.

2.2.3.2 Metode *Training*

Selain memiliki 2 model arsitektur, word2vec juga memiliki dua tipe *training*, yaitu *Negative Sampling* dan *Hierarchical Softmax*.

2.2.3.2.1 Negative Sampling

Negative Sampling adalah metode yang direkomendasikan oleh Mikolov dengan model *Skip-gram*. Metode ini muncul karena adanya modifikasi yang dilakukan pada paper kedua Mikolov untuk membuat proses *training* lebih mudah dan cepat [13] .

Pada *word2vec*, untuk mendapatkan hasil kemiripan yang tinggi dari hasil *dot product* diantara banyak vektor kata yang muncul bersamaan pada sebuah teks dan meminimalkan kemiripan pada kondisi yang sebaliknya, denominator harus menghitung kemiripan target kata (w) dengan seluruh konteks pada setiap konteks kata(c) dan memastikan bahwa kata yang muncul bersamaan akan memiliki kemiripan yang lebih besar daripada yang tidak. Vektor kata dengan jumlah komponen dan *vocabulary* yang besar akan menyebabkan beban yang semakin besar pada pemrosesan *neural network* dengan *hidden layer* dan *output layer* yang dimilikinya. Menjalankan algoritma *gradient descent* pada sebuah *neural network* dengan data tersebut akan memakan waktu yang sangat lama sehingga dibutuhkan *training* data dalam jumlah yang sangat besar untuk memberikan hasil yang baik dan menghindari *overfitting* data.

Dari permasalahan tersebut Mikolov membuat inovasi dengan memodifikasi dalam melakukan optimasi dengan menngunakan metode *Negative Sampling* yang hanya memilih pasangan kata dalam konteks c secara random sehingga memungkinkan proses *word2vec* akan lebih cepat. Setiap saat, sebuah kata akan semakin dekat dengan tetangganya, sedangkan sejumlah kecil kata lain (dipilih secara acak dari distribusi unigram di seluruh kata pada korpus) akan dijauhkan.

2.2.3.2.2 Hierarchical Softmax

Hierarchical Softmax merupakan pendekatan perhitungan yang efisien dari *full softmax*. Kelebihan dari metode ini, adalah untuk mengevaluasi hanya sekitar log2(W) nodes daripada harus mengevaluasi seluruh *output nodes* W pada *neural network* untuk mendapatkan kemungkinan distribusi. *Hierarchical Softmax* menggunakan representasi *binary tree* dari *output layer* dengan W Word sebagai *leaves* dan untuk setiap *nodes*, secara eksplisit merepresentasikan kemungkinan relative dari *child nodesnya*. Hal ini mendefinisikan pergerakan acak yang memberikan kemungkinan pada setiap kata.

Struktur *tree* yang digunakan pada *hierarchical softmax* memiliki dampak yang perlu dipertimbangkan dalam performanya. Mikolov menggunakan *binary Huffman tree*, yang memberikan kode pada kata yang sering muncul dari hasil *training*. Hasil pengamatan membuktikan bahwa dengan mengelompokkan kata secara bersamaan dengan kata kata yang memiliki frekuensi tinggi dapat berjalan dengan baik dan merupakan teknik yang mampu mempercepat pemrosesan *neural network*. [13]

2.2.4 Deeplearning4J

Deeplearning4J adalah *library* yang berisfat open source. Deeplearning4J adalah *distributed deep-learning project* dalam bahasa pemrograman Java dan Scala. Deeplearning4J dipelopori oleh orang-orang yang berada di Skymind, perusahaan yang berada di San Fransisco serta bergerak dibidang *business intelligence* dan *enterprise software*[14]. Deeplearning4J telah terintegrasi dengan Hadoop dan Spark.

Terdapat banyak sekali metode-metode yang disediakan dalam Deeplearning4J, antara lain *word2vec, neural networks, doc2vec,* dsb. Library ini digunakan dalam pengerjaan tugas akhir untuk membuat model *word2vec* dari data media sosial yang telah didapatkan. Selain itu juga digunakan untuk melakukan *pre-processing* data jika diperlukan.

2.2.5 *Levenshtein Distance*

Metode *Levensthein distance* merupakan salah satu cara untuk melakukan pengukuran perbedaan antara dua buah kata dalam bentuk string berupa jarak/distance. Distance ialah jumlah minimum dari operasi hapus, menyisipkan, atau substitusi yang dibutuhkan untuk merubah *string* awal menjadi *string* target[15]. Metode ini memiliki beberapa aturan dalam setiap operasi yang dilakukan dalam mengatahui perbedaan kata yaitu, 1 nilai untuk menghapus *substring* a pada operasi d(a,x), 1 nilai untuk penyisipan *substring* a pada operasi d(x,a), 1 nilai untuk substitusi *substring* a ke *substring* b pada operasi d(a,b), dan d(a,a) yang memiliki nilai 0 karena tidak terjadi perubahan. Nilai yang semakin besar dari *levensthein distance* menunjukkan perbedaan yang semakin tinggi pula.

2.2.6 *Jaro-Winkler Distance*

Algoritma Jaro-Winkler merupakan algoritma yang digunakan untuk membandingkan kesamaan atau perbedaan dari dua buah string. Penggunaan algoritma ini biasanya untuk mendeteksi duplikasi atau kesamaan kalimat/kata dalam sebuah dokumen. Algoritma ini awalnya bernama *Jaro Distance* yang dibuat oleh Matthew A. Jaro yang kemudian dikembangkan oleh William E. Winkler dan Thibaudeau dengan memodifikasi *Jaro Distance* untuk memberikan bobot yang lebih tinggi untuk *prefix* kemiripan[16].

Algoritma ini memiliki keunggulan dibandingkan dengan *Edit Distance*/Levenshtein Distance yaitu pada segi *quadratic runtime complexity.* Algoritma ini akan memberikan nilai 1 ketika dua buah string sama persis dan nilai 0 ketika tidak ada kesamaan. Algoritma ini memiliki tiga bagian yaitu[17]:

1. Mengitung panjang string,
2. Menemukan jumlah karakter yang sama di dalam kedua string, dan
3. Menemukan jumlah transposisi.

Rumus untuk menghitung dua string yang berbeda menurut algoritma ini adalah

$$dj = \frac{1}{3} \times \left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) \quad \text{(persamaan 3)}$$

dimana m adalah jumlah karakter yang sama persis, $|s_1|$ adalah panjang string s1, $|s_2|$ adalah panjang string s2, dan t adalah jumlah transposisi.

2.2.7 Media Sosial

Media sosial merupakan sebuah media yang menggunakan internet sebagai medianya dan memungkinkan penggunanya untuk berpartisipasi, saling berbagi, dan menciptkan konten di dunia virtual untuk menyampaikan pendapat ataupun informasi penting kepada publik. Jejaring sosial merupakan situs yang memungkinkan setiap orang memiliki halaman pribadi dan terhubung dengan teman-temannya untuk saling berbagai informasi dan berkomunikasi satu sama lain. Jejaring sosial sendiri adalah struktur sosial yang terdiri dari elemen individual atau organisasi yang mempertemukan kelompok yang berhubungan karena kesamaan sosialitas, visi, ide, dan lain-lain. Jejaring sosial terbesar antara lain *Facebook, Twitter, Instagram*. Media sosial mengajak siapa saja yang tertarik untuk saling berpartisipasi dengan memberi kontribusi dan *feedback* secara terbuka, memberi komentar, serta membagi informasi dalam waktu cepat dan tak terbatas.

2.2.7.1 *Facebook*

Facebook adalah sebuah layanan jejaring sosial dan situs *web* yang diluncurkan pada 4 Februari 2004. *Facebook* didirikan oleh Mark Zuckerberg, seorang mahasiswa Harvard kelahiran 14 Mei 1984 bersama rekan-rekan mahasiswanya. Menurut Jubilee Enterprise (2010: 79), Indonesia merupakan salah satu pengguna *Facebook* terbesar dengan jumlah pengguna sekitar

17,6 juta orang dan pada tahun 2016 pengguna *Facebook* di Indonesia mencapai 71,6 juta orang. Beberapa fitur dalam *Facebook* memungkinkan pengunanya untuk saling membuat informasi baru, memberikan komentar pada sebuah informasi, menyukai sebuah konten, serta membagikan informasi. Pengguna *Facebook* di Indonesia yang telah menjamur menjadi kekuatan tersendiri bagi *Facebook* sebagai tempat berbagi dan bertukar informasi terkini yang ada di Indonesia maupun di dunia.

Pengguna *Facebook* biasanya membagikan informasi melalui postingan diri sendiri dari pengguna tersebut maupun melalui halaman resmi yang telah dikelola oleh pengelola halaman tersebut. Umumnya, informasi yang dibagikan dapat berupa tulisan, gambar, maupun video. Sayangnya, untuk hal informasi yang disampaikan menggunakan tulisan sebagian besar tidak disertai dengan penggunaan tulisan yang baku, sehingga informasi yang didapat hanya dapat dimengerti konteksnya oleh manusia saja dan menjadi seulit dimengerti oleh komputer dalam hal pemrosesan data teks menggunakan *Natural Language Processing.*

2.2.7.2 *Twitter*

Twitter adalah suatu situs web layanan jaringan sosial dan mikroblog yang memberikan fasilitas bagi pengguna untuk mengirimkan "pembaharuan" berupa tulisan teks dengan panjang maksimum 140 karakter. *Twitter* didirikan pada Maret tahun 2006 oleh perusahaan rintisan Obvious Corp. *Twitter* dapat menjadi sumber yang sangat bermanfaat untuk mengumpulkan data yang digunakan dalam penelusuran informasi yang berkembang. Keragaman dari pengguna *Twitter* membuat bahan informasi tersebut memiliki nilai lebih. Pada September 2010, didapatkan 95 juta *tweet* per hari yang membuat *Twitter* mampu menjadi sumber informasi yang tepat untuk menganalisa informasi di media sosial.[18] Namun di sisi lain, konten dari *tweet* yang ada di dalam *Twitter* seringkali memiliki tata bahasa yang kurang baik dikarenakan batas huruf atau karakter yang dibatasi untuk sekali *tweet*.

2.2.8 *Crawling*

Crawling merupakan sebuah proses untuk mendapatkan informasi konten atau keseluruhan isi halaman yang terdapat pada suatu halaman *website* dan menyimpannya secara *offline*[19]. Dalam penelitian ini *crawling* dilakukan pada laman media sosial *Facebook* dan *Twitter*. *Crawling* bertujuan untuk medapatkan data berupa *tweet* dan posting yang bersumber dari media sosial *Twitter* dan *Facebook*. Data-data yang diambil adalah data publik berupa *tweet*/posting, waktu, pengguna, dan keterangan lain yang dibutuhkan. Semua informasi ini didapatkan melalui bantuan layanan dari *Twitter API* dan *Facebook API*.

2.2.8.1 Facebook4J

Untuk dapat mendapatkan informasi dari *Facebook* menggunakan *Facebook API*, *crawler* akan memanfaatkan *library* dari Java yang bernama Facebook4J sehingga proses mendapatkan informasi lebih mudah. Facebook4j merupakan *library Java* tidak resmi yang disediakan untuk mengintegrasikan aplikasi Java dengan *Facebook* API[20].

Facebook4J dibuat oleh Ryuji Yamasitha pada tahun 2012. Facebook4j merupakan *open source* yang juga terdapat pada library Java serta memiliki lisensi *Apache 2.0*.

2.2.8.2 Twitter4J

Untuk dapat mendapatkan informasi dari *Twitter* menggunakan *Twitter API*, crawler akan memanfaatkan library dari java yang bernama Twitter4J sehingga proses mendapatkan informasi lebih mudah. Twitter4j merupakan library Java tidak resmi *open source* untuk mengintegrasikan aplikasi Java dengan *Twitter* API yang tersedia sejak bulan Juni 2007[21]. Twitter4J dibuat oleh Yusuke Yamamoto serta memiliki lisensi *Apache 2.0*.

2.2.9 Kateglo

Kateglo (kamus, tesaurus dan glosarium) merupakan bentuk daring dari layanan kamus, tesaurus dan glosarium bahasa Indonesia. Data yang ada pada Kateglo berasal dari berbagai sumber yaitu KBBI III, KBBI IV, Wikipedia, Kateglo, serta kontirbusi publik.

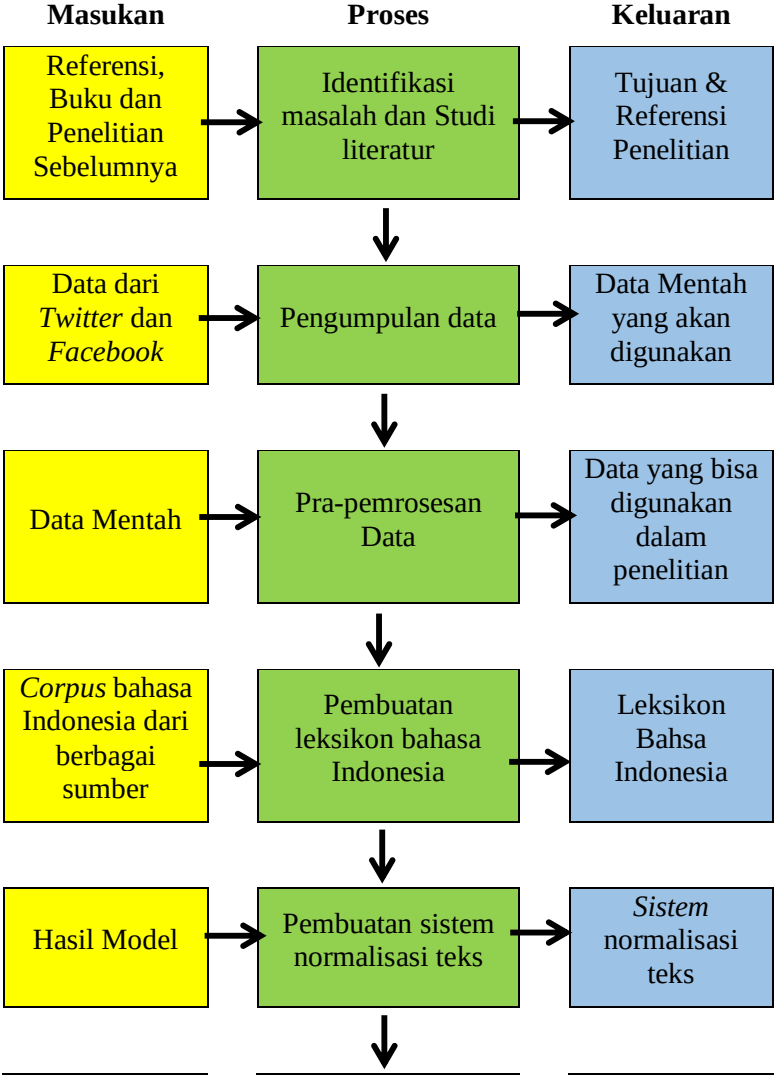
Kateglo adalah laman web yang menyediakan sumber terbuka berbasis PHP dan berbasis data MySQL dengan lisensi GPL. Isinya sendiri berlisensi CC-BY-NC-SA (Creative Commons Attribution Non Commercial ShareAlike), atau dengan kata lain penggunanya diizinkan untuk menyalin, menyebarkan, atau mengadaptasi isi Kateglo dengan bebas asal mencantumkan sumber isi, bukan untuk tujuan komersial, dan dalam lisensi yang sama atau serupa dengan lisensi kateglo[22].

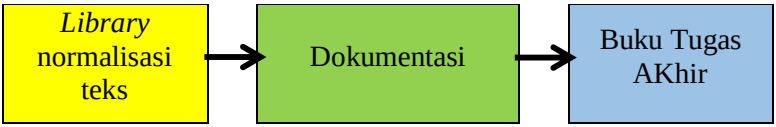
2.2.10 Wikitionary Indonesia

Wiktionary Bahasa Indonesia merupakan salah satu ensiklopedia kamus bahasa Indonesia yang berbasis online. Pertumbuhan kata yang terdapat di Wiktionary cukup pesat dikarenakan kontributor yang aktif menambahkan entri-entri kata beserta artinya kedalam system. Wiktionary bahasa Indonesia dimulai pada tahun 2004 dan saat ini telah memiliki 128.273 lema. Wiktionary juga tersedia dalam berbagai bahasa di dunia[23].

BAB III
METODOLOGI

3.1 Tahapan Pelaksanaan Tugas Akhir





Gambar 3.3.1 Metodologi Penelitian

3.1.1 Identifikasi Masalah dan Studi Literatur

Tahapan ini merupakan fase pertama dalam pengerjaan Tugas Akhir ini. Pada tahap ini dilakukan identifikasi masalah penggalian kebutuhan pengetahuan terkait dengan studi kasus yang akan diambil. Dikarenakan belum ada penelitian yang membahas mengenai hal ini maka literatur – literatur yang dijadikan referensi berasal dari luar negeri dan beragam metodenya. Literatur – literatur yang diambil adalah penelitian seputar *word2vec* dan metode-metode untuk normalisasi teks maupun *text correction.* Paper utama yang menjadi rujukan adalah tentang word2vec yang berjudul "*Efficient Estimation of Word Representations in Vector Space*" dan "*Distributed Representations of Words and Phrases and their Compositionality*". Kedua literatur tersebut membahas tentang cara baru dalam merepresntasikan sebuah kata-kata kedalam *vector low dimensional*, sehingga dapat diketahui hubungan antara kata-kata secara kesamaan konteks maupun hubungan lainnya. Metode ini dijadikan dasar untuk pembuatan model *word2vec* pada tugas akhir ini nantinya.

3.1.2 Proses Pengumpulan Data

Pada tahapan ini dilakukan pengumpulan data dari beberapa akun media sosial *Twitter* dan *Facebook*. Data yang dikumpulkan adalah data posting *Facebook* dari Agustus 2015 dan posting *Twitter* dari Agustus 2016. Secara umum, data yang dikumpulkan melalui *Twitter* maupun *Facebook* paling tidak harus sering melakukan posting dan juga memiliki tata bahasa yang baku maupun tidak baku. Daftar akun *Facebook* dan *Twitter* yang akan diambil data beserta alasannya dapat dilihat pada Tabel 3.3.1.

Tabel 3.3.1 Daftar Akun Media Sosial

Twitter	Facebook	Alasan
Pemerintahan		
Kemdikbud_RI	Kemdikbud.RI	Memiliki tata bahasa yang baku dan merupakan akun pemerintahan.
indtravel	Indonesia TravelINA	Memiliki tata bahasa yang baku dan merupakan akun pemerintahan.
Kementrian Agama	KementerianAgamaRI	Memiliki tata bahasa yang baku dan merupakan akun pemerintahan.
LPDP_RI	LPDPKemenkeu	Memiliki tata bahasa yang baku dan merupakan akun pemerintahan.
kemkominfo	Kemkominfo	Memiliki tata bahasa yang baku dan merupakan akun pemerintahan.
Sapawargasby	-	Akun pemerintah kota Surabaya yang melaporkan kondisi terkini kota Surabaya
Kemenpar_RI	kemenpar	Memiliki tata bahasa yang baku dan merupakan akun pemerintahan.
KemenBUMN	KementerianBUMNRI	Memiliki tata bahasa yang baku dan merupakan akun pemerintahan.
Kementrian ESDM	kesdm	Memiliki tata bahasa yang baku dan merupakan akun pemerintahan.
kemristekdikti	Dikti	Memiliki tata bahasa yang baku dan merupakan akun pemerintahan.
KemenkeuRI	KementerianKeuanganRI	Memiliki tata bahasa yang baku dan merupakan akun pemerintahan.

Twitter	Facebook	Alasan
	E-commerce	
bukalapak	Bukalapak	Cukup aktif dalam melakukan postingan. Bahasa yang digunakan cukup beragam mulai dari baku dan tidak baku.
tokopedia	tokopedia	Cukup sering melakukan posting dan bahasa yang digunakan beragam.
OLX_Indonesia	olxid	Cukup aktif dalam melakukan postingan. Bahasa yang digunakan cukup beragam mulai dari baku dan tidak baku.
	Radio	
-	Ag243	Akun radio di Kediri yang cukup aktif dalam memposting kondisi jalanan dan memiliki ragam bahasa baku dan tidak baku
E100ss	E100ss	Akun radio di Surabaya yang cukup aktif dalam memposting kondisi jalanan dan memiliki ragam bahasa baku dan tidak baku
radioelshinta	-	Akun radio yang cukup aktif dalam memposting kondisi jalanan dan memiliki ragam bahasa baku dan tidak baku
	Public Figure	
jokowi	Jokowi	Akun public figure yang cukup aktif dalam memposting dan memiliki tata bahasa yang baik.

Twitter	Facebook	Alasan
basuki_btp	AhokBTP	Akun public figure yang cukup aktif dalam memposting dan memiliki tata bahasa yang baik.
SBYudhoyono	SBYudhoyono	Akun public figure yang cukup aktif dalam memposting dan memiliki tata bahasa yang baik.
Media Online		
tempodotco	TempoMedia	Merupakan salah satu akun media sosial berita di Indonesia dan cukup aktif dalam melakukan postingan. Bahasa yang digunakan cukup beragam mulai dari baku dan tidak baku.
kaskus	Officialkaskus	Cukup aktif dalam melakukan postingan. Bahasa yang digunakan cukup beragam mulai dari baku dan tidak baku.
kompascom	KOMPAScom	Merupakan salah satu akun media sosial media di Indonesia dan cukup aktif dalam melakukan postingan. Bahasa yang digunakan cukup beragam mulai dari baku dan tidak baku.
kompasiana	KOMPASIANAcom	Merupakan salah satu akun media sosial berita di Indonesia dan cukup aktif dalam melakukan postingan. Bahasa yang digunakan cukup beragam mulai dari baku dan tidak baku.

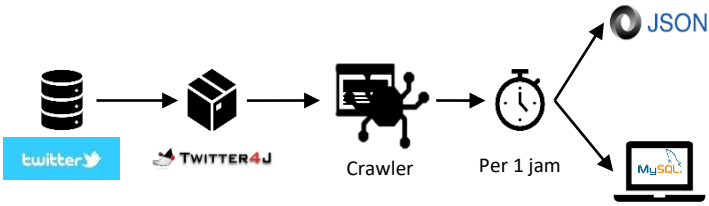
Twitter	Facebook	Alasan
GOAL_ID	goal.indonesia	Merupakan salah satu akun media sosial berita di Indonesia dan cukup aktif dalam melakukan postingan. Bahasa yang digunakan cukup beragam mulai dari baku dan tidak baku.
BBCIndonesia	bbc.indonesia	Merupakan salah satu akun media sosial berita di Indonesia dan cukup aktif dalam melakukan postingan. Bahasa yang digunakan cukup beragam mulai dari baku dan tidak baku.
okezonenews	OkezoneCom	Merupakan salah satu akun media sosial berita di Indonesia dan cukup aktif dalam melakukan postingan. Bahasa yang digunakan cukup beragam mulai dari baku dan tidak baku.
detikcom	detikcom	Merupakan salah satu akun media sosial berita di Indonesia dan cukup aktif dalam melakukan postingan. Bahasa yang digunakan cukup beragam mulai dari baku dan tidak baku.
TechinAsia_ID	techinasiaID	Cukup aktif dalam melakukan postingan. Bahasa yang digunakan cukup beragam mulai dari baku dan tidak baku.
Produk		
xiaomiindonesia	XiaomiIndonesia	Cukup aktif dalam melakukan postingan. Bahasa yang digunakan cukup beragam mulai dari baku dan tidak baku.

Twitter	Facebook	Alasan
UnileverIDN	unileverid	Cukup aktif dalam melakukan postingan. Bahasa yang digunakan cukup beragam mulai dari baku dan tidak baku.
TelkomIndonesia	TelkomIndonesia	Cukup aktif dalam melakukan postingan. Bahasa yang digunakan cukup beragam mulai dari baku dan tidak baku.
AXISgsm	AXISgsm	Cukup aktif dalam melakukan postingan. Bahasa yang digunakan cukup beragam mulai dari baku dan tidak baku.
PosIndonesia	posindonesia	Cukup aktif dalam melakukan postingan. Bahasa yang digunakan cukup beragam mulai dari baku dan tidak baku.
Telkomsel	telkomsel	Cukup aktif dalam melakukan postingan. Bahasa yang digunakan cukup beragam mulai dari baku dan tidak baku.
Universitas		
univ_indonesia	ui.ac.id	Akun perguruan tinggi yang memiliki bahasa cukup baku dan cukup aktif melakukan posting.
ITS_Surabaya	InstitutTeknologiSepuluhNopember	Akun perguruan tinggi yang memiliki bahasa cukup baku dan cukup aktif melakukan posting.
UGMYogyakarta	UGMYogyakarta	Akun perguruan tinggi yang memiliki bahasa cukup baku dan cukup aktif melakukan posting.

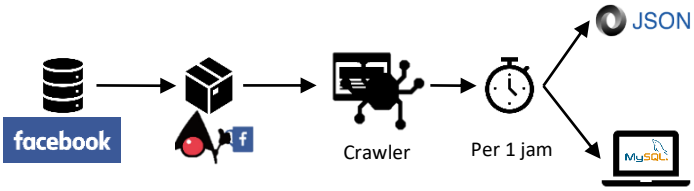
Twitter	Facebook	Alasan
itbofficial	instituttek nologiban dung	Akun perguruan tinggi yang memiliki bahasa cukup baku dan cukup aktif melakukan posting.
Organisasi		
Ind_Menga jar	Indonesia Mengajar	Memiliki tata bahasa yang cukup baku dan merupakan organisasi umum di Indonesia.
BEM_ITS	BEMITS Surabaya	Akun organisasi mahasiswa yang cukup aktif melakukan posting.

3.1.2.1 Desain *Crawler*

Data mentah tersebut akan dikumpulkan dengan menggunakan *crawler* yang dijalankan sesuai dengan jadwal, yaitu tiap satu jam sekali. *Crawler* yang digunakan memanfaatkan *library* java Twitter4J dan Facebook4J. Nantinya, data mentah yang telah dikumpulkan akan disimpan kedalam dua bentuk yaitu *database* MySQL dan JSON. Lebih jelasnya dapat dilihat pada arsitektur pada gambar Gambar 3.3.2 dan Gambar 3.3.3.



Gambar 3.3.2 Proses *crawling Twitter*

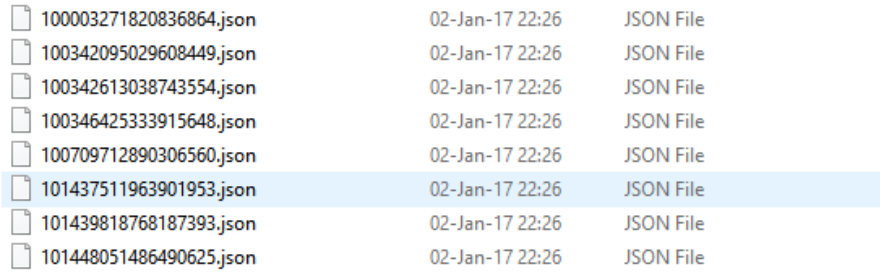


Gambar 3.3.3 Proses *crawling Facebook*

Selain itu, juga dapat dilihat contoh hasil *crawling* berupa JSON dan *database* MySQL seperti pada Gambar 3.3.4 dan Gambar 3.3.5.

id	message	account	latitude	longitude	created_time
784177628903116800	Good morning SUB CITY, my beloved city @e100ss	rusdyattamim	0.0	0.0	2016-10-07 06:44:50
784175032926806016	@SbyTrafficServ @e100ss @RTMCJatim TL MERRSTIKOM pagi sdhlebihtertib,mo	robbyanton	0.0	0.0	2016-10-07 06:34:31
784174253662883841	RT @Amelandoko: @e100ss jam 19.30-20.30 jalan Dharmawangsa (dpn Alfa Mart	fatkur83	0.0	0.0	2016-10-07 06:31:25
784174231470780418	RT @Amelandoko: @e100ss Erik (warga Gubeng Airlangga 1) menyebutkan 4 det	fatkur83	0.0	0.0	2016-10-07 06:31:20
784174228337594368	RT @Amelandoko: @e100ss setelah polisi datang, diketahui 4 orang yg mengaku	fatkur83	0.0	0.0	2016-10-07 06:31:19
784169897756463104	RT @Amelandoko: @e100ss setelah polisi datang, diketahui 4 orang yg mengaku	Mudjito8	0.0	0.0	2016-10-07 06:14:07
784169234200866817	Salute pd manajemen @KAI121 yg memperhatikan kebutuhan difabel, maju teru	dr_m_ardian	0.0	0.0	2016-10-07 06:11:29

Gambar 3.3.4 Hasil data yang disimpan pada MySQL



Gambar 3.3.5 Hasil data yang disimpan bentuk JSON

Untuk tugas akir ini, hasil yang akan digunakan adalah hasil *crawling* yang disimpan dalam bentuk database karena lebih mudah dan kolom yang diperlukan hanyalah isi dari postingan tersebut.

3.1.3 Pra-pemrosesan Data

Data yang telah di peroleh dari porses *crawling* akan masuk ke tahapan pra-pemrosesan data. Dimana setiap data yang akan digunakan pada proses *training* akan memasuki tahapan pra-pemrosesan data satu per satu.

3.1.3.1 Menggabungkan Dataset dan Menghapus data yang Terduplikasi

Data yang didapatkan dari *crawling* masih terdapat pada sumber yang terpisah sehingga perlu digabungkan untuk menjadi satu dataset. Selain itu, tidak menutup kemungkinan saat proses memasukkan data kedalam *database*, terdapat data yang terduplikasi sehingga harus dihapus. Dengan melalui

tahapan ini maka akan diperoleh data teks yang lebih layak dianalisa pada tahap berikutnya.

3.1.3.2 Menghapus Tanda Baca dan Simbol

Pada tahapan ini, setiap posting *Facebook* dan *Twitter* akan dihapus tanda bacanya dan simbol-simbol yang tidak bermakna. Tujuan dari proses ini adalah menghilangkan karakter yang tak bermakna serta memperkecil peluang kata yang memiliki makna sama namun berbeda penulisan karena penambahan tanda baca. Untuk contohnya dapat dilihat pada Tabel 3.3.2.

Tabel 3.3.2 Proses menghapus tanda baca dan simbol

Sebelum Pra-proses	Setelah Pra-proses
@SbyTrafficServ @e100ss @RTMCJatim TL MERRSTIKOM pagi sdhlebihtertib,mohon perhatikanR2 tdk pakaihelm,bykpelajarpakaiR2 apakahsurat lengkap?	TL MERRSTIKOM pagi sdhlebihtertib mohon perhatikanR2 tdk pakaihelm bykpelajarpakaiR2 apakahsurat lengkap
RT @Amelandoko: @e100ss jam 19.30-20.30 jalan Dharmawangsa (dpn Alfa Mart) padat. Terjadi pertikaian antara pengendara mobil dan beberapa…	jam 19 30 20 30 jalan Dharmawangsa dpn Alfa Mart padat Terjadi pertikaian antara pengendara mobil dan beberapa

3.1.3.3 Menghapus Data yang Terduplikasi

Data yang telah didapatkan akan dilakukan penghapusan terhadap data yang sama. Sehingga hanya didapatkan kata yang unik saja. Hal ini dilakukan untuk mengurangi adanya redudansi data sehingga data yang didapatkan menjadi lebih reliable.

3.1.3.4 Menggabungkan Baris yang Terpisah

Pada dasarnya beberapa postingan memiliki baris yang terpisah atau biasa disebut *new line.* Baris yang terpisah tersebut harus digabungkan sehingga memudahkan pemrosesan data menggunakan model *word2vec*. Berikut adalah contohnya pada Tabel 3.3.3.

Tabel 3.3.3 Proses menggabungkan baris

Sebelum Pra-proses	Setelah Pra-proses
muktamar nu rumuskan konsep islam nusantara islam yang tanpa pentungan. inilah konsep islam tanpa kekerasan itu http://m.suarasurabaya.net/kelanakota/detail.php?id=2rd5iab0l0skf1u7a4ru2jflp32015156524	muktamar nu rumuskan konsep islam nusantara islam yang tanpa pentungan. inilah konsep islam tanpa kekerasan itu http://m.suarasurabaya.net/kelanakota/detail.php?id=2rd5iab0l0skf1u7a4ru2jflp32015156524
guyonan gus ipul, pakde karwo dan jokowi di muktamar nu http://m.suarasurabaya.net/kelanakota/detail.php?id=2rd5iab0l0skf1u7a4ru2jflp32015156523	guyonan gus ipul, pakde karwo dan jokowi di muktamar nu http://m.suarasurabaya.net/kelanakota/detail.php?id=2rd5iab0l0skf1u7a4ru2jflp32015156523

3.1.3.5 *Tokenizing*

Tokenizing adalah proses membuat token/pemisahan per kata dari sebuah korpus. Nantinya token-token yang telah terbentuk digunakan untuk merepresentasikan sebuah kata kedalam vektor menggunakan *word2vec*. Berikut adalah contohnya pada Tabel 3.3.4.

Tabel 3.3.4 Proses *tokenizing*

Sebelum Pra-proses	Setelah Pra-proses

TL MERRSTIKOM pagi sdhlebihtertib mohon perhatikanR2 tdk pakaihelm bykpelajarpakaiR2 apakahsurat lengkap	{tl, merrstikom, pagi, sdhlebihtertib, mohon, perhatikanr2, tdk, pakaihelm, bykpelajarpakair2, apakahsurat, lengkap}
jam 19 30 20 30 jalan Dharmawangsa dpn Alfa Mart padat Terjadi pertikaian antara pengendara mobil dan beberapa	{jam, 19, 30, 20, 30, jalan, dharmawangsa, dpn, alfa, mart, padat, terjadi, pertikaian, antara, pengendara, mobil, dan, beberapa}

3.1.4 Pembuatan Leksikon Bahasa Indonesia

Sebelum melakukan pembuatan sistem normalisasi teks diperlukan pengecekan apakah sebuah kata termasuk dalam kamus bahasa Indonesia. Oleh karena itu diperlukan pembuatan leksikon bahasa Indonesia yang didapatkan dari *corpus-corpus* bahasa Indonesia yang ada di internet. Pengambilan data akan dilakukan dengan cara *crawling* maupun manual.

Selain itu leksikon bahasa Indonesia ini akan dilengkapi dengan kamus *mapping* untuk memetakan kata-kata yang tidak baku menjadi baku secara langsung. Nantinya leksikon bahasa Indonesia ini akan dibuat dalam bentuk *full-text search* yang telah terindeks menggunakan *Apache Solr* untuk mempercepat performa pencarian.

3.1.5 Pembuatan Model *Word2Vec*

Data yang telah melewati pra-proses akan dibuat model *word2vec*-nya menggunakan *library* DeepLearning4J. Pada tahapan ini akan dibuat beberapa skenario dengan mengubah beberapa parameter dalam proses *training* seperti:

- Arsitektur Model (CBOW/Skip-gram)
- Metode *Training* (*Negative Sampling*/*Hierarchical Softmax*)

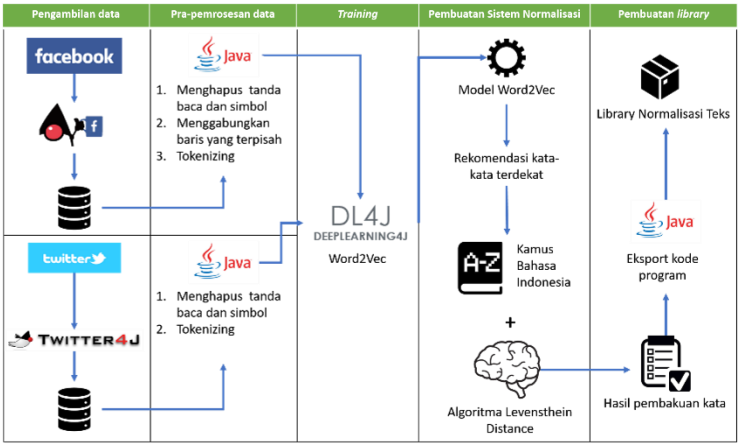
Tujuan mengubah parameter dari proses *training* untuk mengetahui model terbaik mana yang akan digunakan sebagai bahan membuat *library* normalisasi teks.

3.1.5.1 Analisis Model *Word2vec*

Dalam setiap proses *training* yang dilakukan akan terdapat akurasi pelatihan, yang mana menunjukkan kualitas dari model yang dihasilkan. Semakin besar tingkat akurasi maka semakin bagus model yang dihasilkan.

3.1.6 Pembuatan Sistem Normalisasi Teks

Pada tahapan ini, akan dibentuk sebuah sistem untuk menormalisasi teks yang telah di dapat dengan gambaran umum seperti gambar dibawah ini.



Gambar 3.3.6 Kerangka kerja *library*

Dari Gambar 3.3.6 dapat dilihat, model *word2vec* yang telah dibuat pada tahapan sebelumnya akan dimanfaatkan sebagai penyeleksi awal untuk menormalisasi teks dengan memilih kandidat-kandidat kata pembenaran yang sesuai konteks ataupun sintaks dari kata yang dimasukkan. Kemudian kandidat-kandidat tersebut di cocokkan dengan kamus bahasa Indonesia untuk mendapatkan informasi apakah setiap kandidat merupakan kata baku. Penyeleksian ketiga dilakukan dengan mengimplementasikan algoritma *Levensthein distance*, yaitu

menghitung kesamaan dari kata yang dimasukkan dengan kandidat-kandidat yang ada dan dicocokkan dengan kamus Bahasa Indonesia. Setelah terpilih, maka akan muncul hasil akhir kandidat kata mana yang tepat untuk menormalisasi kata masukan.

3.1.6.1 Analisis dan pengujian hasil normalisasi teks

Setelah library dibuat, maka proses analisis dan pengujian hasil normalisasi teks akan dilakukan dengan mengujikan data tes yang terdiri dari sekumpulan kalimat yang tidak baku dan kalimat baku kepada sistem. Pengujian dilakukan dengan cara memprediksi hasil luaran dari proses normalisasi kalimat tidak baku dibandingkan dengan kalimat baku yang telah dibuat manual. Nantinya, keakuratan dihitung dari seberapa banyak kata yang dapat dinormalisasi dengan benar dibandingkan dengan jumlah kata yang ada dalam satu kalimat.

3.1.7 Dokumentasi

Pada tahapan terakhir ini akan dilakukan pembuatan laporan dalam bentuk buku tugas akhir yang disusun sesuai format yang telah ditentukan. Buku ini berisi dokumentasi langkah-langkah pengerjaan tugas akhir secara rinci. Buku ini diharapkan dapat bermanfaat sebagai referensi untuk pengerjaan penelitian lain, serta sebagai acuan untuk pengembangan lebih lanjut terhadap topik penelitian yang serupa.

BAB IV
PERANCANGAN

Pada bab ini akan membahas mengenai perancangan dari luaran tugas akhir ini.

4.1 Akuisisi Data Media Sosial

Tahap awal perancangan dimulai dengan mengumpulkan seluruh data media sosial yang diperlukan untuk tahap pengolahan selanjutnya. Pengambilan data dilakukan melalui crawler yang dirancang untuk mengambil data setiap satu jam sekali. Data dikumpulkan dari 8 jenis akun facebook dan twitter yang telah dijabarkan dalam sub bab 3.1.2 yaitu pemerintahan, *e-commerce, public figure,* produk, universitas, organisasi, radio, dan media online.

4.2 Perancangan *Crawler*

Untuk mengambil data secara rutin dirancang sebuah *crawler* yang akan mengambil dan menyimpan data dari seluruh akun media sosial yang telah ditentukan. *Crawler* yang dibuat mengacu pada *library Facebook4J* untuk mengambil data *Facebook* dan *Twitter4J* untuk mengambil data *Twitter*.

4.2.1 Desain Database

Untuk melakukan perancangan *crawler,* maka perlu melakukan perancangan database untuk menyimpan data yang diambil, yaitu data media sosial *Facebook* dan *Twitter*.

4.2.1.1 Database *Facebook*

Data yang diambil dari *Facebook* adalah data posting dan komentar. Untuk dapat melakukan pemrosesan data perlu menyimpan atribut-atribut yang ada di posting dan komentar *Facebook*.

Atribut-atribut dari posting *Facebook* yang disimpan beserta tipe datanya dalam database *MySQL* dapat dilihat pada Tabel 4.1.

Tabel 4.1 Desain *database crawler post facebook*

Atribut	Tipe Data	Penjelasan	Batasan
fb_id	text	Berisi nomor unik dari sebuah posting di Facebook	Not null
message	text	Berisi konten dari sebuah posting Facebook	null
story	text	Berisi apakah terdapat story/note yang terdapat pada sebuah posting.	null
created_time	datetime	Tanggal sebuah posting dibuat	Not null
user	text	Berisikan nomor unik pengguna yang membuat posting	Not null

Atribut yang digunakan dipilih berdasarkan analisis kebutuhan data untuk proses selanjutnya. Atribut fb_id digunakan untuk mengidentifikasi konten facebook yang diambil secara unik, atribut *message* dipilih sebagai data utama yang akan digunakan dalam pemrosesan data di tugas akhir ini.

Selain data posting *Facebook*, dilakukan juga pengumpulan data komentar dari sebuah posting tersebut. Tabel 4.2 menunjukkan desain *database* dari komentar di *Facebook*.

Tabel 4.2 Desain *database crawler* komentar *facebook*

Atribut	Tipe Data	Penjelasan	Batasan
id_comment	text	Berisi nomor unik dari setiap komentar yang terdapat pada posting Facebook	Not null
id_post	text	Berisi nomor unik dari sebuah posting di Facebook	Not null
message	text	Berisikan konten dari komentar pada posting Facebook	null
account	text	Berisikan *username* dari pengguna yang memposting komentar	Not null
acoount_id	text	Berisikan nomor unik pengguna yang yang memposting komentar	Not null
created_time	datetime	Berisikan tanggal dibuat dari sebuah komentar	Not null

Atribut yang akan digunakan adalah *message*, karena berisikan konten dari komentar yang ada di *Facebook.*

4.2.1.2 Database *Twitter*

Data yang diambil dari *Twitter* adalah data tweet akun-akun yang ada diatas di sub bab 3.1.2. Tabel 4.3 menunjukkan desain database untuk menyimpan posting dari Twitter.

Tabel 4.3 Desain database crawler post twitter

Atribut	Tipe Data	Penjelasan	Batasan
id	text	Berisikan nomor unik dari setiap posting yang ada di twitter	Not null
message	text	Berisikan konten dari posting di Twitter	null
account	text	Berisikan nomor unik dari setiap pengguna yang melakukan posting di Twitter	Not null
latitude	text	Berisikan koordinat dari sebuah tempat dimana posting twitter diterbitkan	null
longitude	text	Berisikan koordinat dari sebuah tempat dimana posting twitter diterbitkan	null
created_time	datetime	Berisikan tanggal sebuah posting twitter diterbitkan	null

Dalam pemilihan atribut, atribut yang akan digunakan adalah atribut message saja dikarenakan hal itu sangat penting untuk pemrosesan data dalam tugas akhir ini.

4.2.2 Desain *Crawler*

Untuk melakukan pengambilan data dari media sosial, maka perlu dibuat sebuah *crawler*. Inti *source code* yang digunakan berasal dari sebuah library Facebook4J dan Twitter4J dengan basis bahasa pemrograman Java. Kemudian dilakukan sebuah kustomisasi agar dapat mengambil data pada banyak akun sekaligus dan dalam kurun waktu tertentu.

Setelah *source code* sebuah *crawler* telah selesai dibuat, maka dilakukan penjadwalan agar *crawling* dapat berlangsung 1 jam dalam tiap harinya.

4.3 Perancangan Pra-pemrosesan Data

Sebelum diproses, data harus terlebih dahulu mengalami proses persiapan data atau pra-pemrosesan. Data pra pemrosesan menunjukkan tipe-tipe proses yang menggunakan data mentah untuk ditransformasi ke suatu format yang lebih mudah dan efektif untuk kebutuhan rekomendasi agar dapat diolah dengan baik saat pembuatan model pada langkah selanjutnya. Berikut ini merupakan tahapan yang dilakukan pada pra-pemrosesan data berikut

4.3.1 Perancangan Penggabungan Dataset

Karena pada desain database dibuat terpisah antara *Facebook* dan *Twitter*, maka dataset perlu digabungkan terlebih dahulu agar dataset menjadi satu. Tentunya data yang digabungkan adalah data yang bertipe teks saja dan terdapat pada atribut *message*.

Untuk menggabungkannya dilakukan pengambilan data dari kedua database menggunakan query dalam Kode 4.1:

```
SELECT DISTINCT (message) FROM fb;//untuk FB
SELECT DISTINCT (message) FROM tw;//untuk Twitter
```

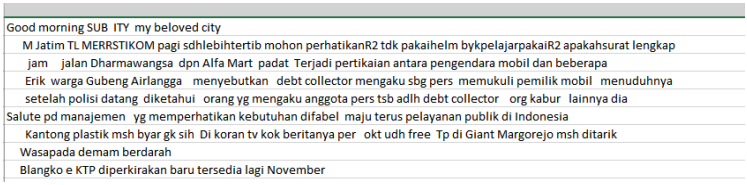
Kode 4.1 Kode mengambil dari tabel

Contoh data yang telah diambil menggunakan *query* diatas ditampilkan dalam Gambar 4.1.

message
Muktamar NU rumuskan konsep Islam Nusantara, Islam yang Tanpa Pent
Guyonan Gus Ipul, Pakde Karwo dan Jokowi di Muktamar NU. (odp-fk)htt
Sebanyak 94 TKI ilegal Dideportasi Malaysia. (odp-rt)http://m.suarasurab
Datang ke Muktamar, Jokowi Bagikan Kaos dan Kartu Indonesia Pintar. (o
21.45 : Hindari masuk Jombang Kota! Lalu lintas MACET TOTAL. Imas ke
Foto almarhum KH Abdurrahman Wahid alis Gus Dur sedang membuka a
21.00 : 4 Jalur MACET : 1.Simpang 3 Lakasantri;2.Depan Stasiun Wonokro
20.50 : Hindari masuk JL Kalidami! JL Kalidami - Unair Kampus B ada baza

Gambar 4.1 Data dalam database

Kemudian dipindahkan kedalam bentuk CSV melalui sebuah program Java yang memanfaatkan *library* OpenCSV. Setelah itu akan terbentuk file CSV yang memuat seluruh data gabungan yang dapat dilihat pada Gambar 4.2.



Good morning SUB ITY my beloved city
M Jatim TL MERRSTIKOM pagi sdhlebihtertib mohon perhatikanR2 tdk pakaihelm bykpelajarpakaiR2 apakahsurat lengkap
jam jalan Dharmawangsa dpn Alfa Mart padat Terjadi pertikaian antara pengendara mobil dan beberapa
Erik warga Gubeng Airlangga menyebutkan debt collector mengaku sbg pers memukuli pemilik mobil menuduhnya
setelah polisi datang diketahui orang yg mengaku anggota pers tsb adlh debt collector org kabur lainnya dia
Salute pd manajemen yg memperhatikan kebutuhan difabel maju terus pelayanan publik di Indonesia
Kantong plastik msh byar gk sih Di koran tv kok beritanya per okt udh free Tp di Giant Margorejo msh ditarik
Wasapada demam berdarah
Blangko e KTP diperkirakan baru tersedia lagi November

Gambar 4.2 Data dalam bentuk CSV

4.3.2 Perancangan Penghapusan Tanda Baca dan Simbol

Dalam data teks yang telah didapatkan dari media sosial, kurang lebih pasti mengandung banyak simbol serta tanda baca yang tidak memiliki arti dan kurang berpengaruh dalam proses training. Oleh karena itu, hal ini harus dihilangkan dalam dataset sehingga dataset menjadi bersih.

Untuk menghilangkan tanda baca dan simbol, maka dilakukan proses *replace* dengan menggunakan *regular expression* dalam bahasa Java. Simbol-simbol yang akan dihilangkan adalah:

1. Menghapus simbol “RT” pada posting Twitter
2. Menghapus singkatan untuk kata “untuk” yang menggunakan simbol “u/”
3. Menghapus jam yang mempunyai format “20:20”/”20.20”

4. Menghapus redaksi yang menulis berita, contoh: (Odp-pr)
5. Menghapus semua tag HTML, contoh: , <a>
6. Menghapus *mention* pada sebuah posting Twitter, contoh: @e100ss, @kemenpar
7. Menghapus *hashtag* pada setiap posting, contoh: #SAVENKRI, #YOLO
8. Menghapus link yang terdapat pada sebuah posting, contoh: https://t.co/tR546JnkoM
9. Menghapus semua karakter ASCII, seperti “$”, ”~”, ”@”
10. Menghapus semua karakter non-ASCII, seperti “é”,”'ö”, “ĉ”
11. Menghapus simbol derajat *celcius* yang digunakan untuk menunjukkan ukuran suhu dalam sebuah posting, seperti “°C”

Setelah semua tahapan diatas dilakukan, maka akan dihasilkan dataset seperti Gambar 4.3.

```
1  Seiring dengan berlalunya waktu maka seseorang akan makin terbiasa
   dengan kenyataan baru tersebut
2  Guyonan Gus Ipul Pakde Karwo dan Jokowi di Muktamar NU
3  Datang ke Muktamar Jokowi Bagikan Kaos dan Kartu Indonesia Pintar
4  Foto almarhum KH Abdurrahman Wahid alis Gus Dur sedang membuka
   amplop berisi uang Rp5 menjadi pusat perhatian pengunjung pameran
   foto yang digelar jelang Muktamar NU Foto  Fatkhurrohman Taufik
   Reporter Suara Surabaya
5    Hindari masuk JL Kalidami JL Kalidami  Unair Kampus B ada bazar
     Lalu lintas MAET karena jalur yang dari arah Karang Menjangan
     digunakan jadi  lajur
```

Gambar 4.3 Hasil penghapusan simbol

4.3.3 Perancangan Penggabungan Baris yang Terpisah

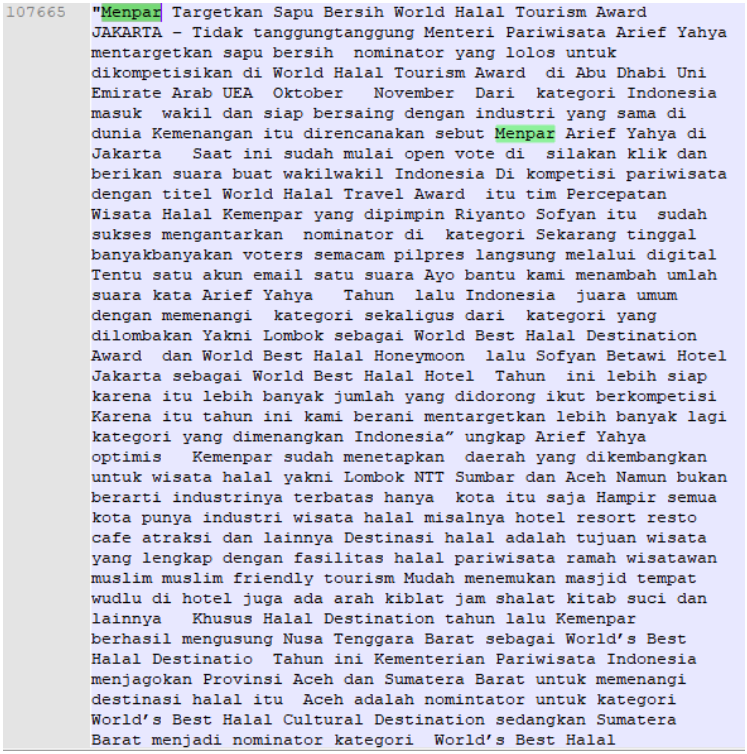
Untuk beberapa posting Facebook tertentu mempunyai konten yang mengandung lebih dari 1 paragraf sehingga antar paragraf dipisahkan oleh jarak atau *newline*. Dalam hal ini dataset harus dijadikan satu baris terlebih dahulu agar program yang dibuat nantinya tidak bingung ketika membaca dataset tersebut, karena pada dasarnya program membaca dataset dengan cara dibaca per baris.

Dengan memanfaatkan *regular expression*, maka nantinya baris yang awalnya terdiri dari beberapa paragraf akan menjadi satu paragraf yang dapat dilihat dari pada Gambar 4.4.

```
Menpar Targetkan Sapu Bersih World Halal Tourism Award 2016

JAKARTA – Tidak tanggung-tanggung, Menteri Pariwisata Arief Yahya
mentargetkan sapu bersih 12 nominator yang lolos untuk
dikompetisikan di World Halal Tourism Award 2016, di Abu Dhabi,
Uni Emirate Arab (UEA), 24 Oktober – 24 November 2016. Dari 16
kategori, Indonesia masuk 12 wakil dan siap bersaing dengan
industri yang sama di dunia. "Kemenangan itu direncanakan," sebut
Menpar Arief Yahya, di Jakarta.

Saat ini sudah mulai open vote, di http://bit.ly/VOTEindonesia,
silakan klik dan berikan suara buat wakil-wakil Indonesia. Di
kompetisi pariwisata dengan titel World Halal Travel Award 2016
itu, tim Percepatan Wisata Halal Kemenpar yang dipimpin Riyanto
Sofyan itu  sudah sukses mengantarkan 12 nominator di 12 kategori.
"Sekarang tinggal banyak-banyakan voters, semacam pilpres langsung
melalui digital. Tentu satu akun email, satu suara. Ayo bantu
kami, menambah umlah suara," kata Arief Yahya.

Tahun 2015 lalu, Indonesia  juara umum dengan memenangi 3 kategori
sekaligus dari 15 kategori yang dilombakan. Yakni Lombok sebagai
World Best Halal Destination Award 2015 dan World Best Halal
Honeymoon 2015, lalu Sofyan Betawi Hotel Jakarta sebagai World
Best Halal Hotel 2015. Tahun 2016 ini lebih siap, karena itu lebih
banyak jumlah yang didorong ikut berkompetisi. "Karena itu tahun
ini kami berani mentargetkan lebih banyak lagi kategori yang
dimenangkan Indonesia," ungkap Arief Yahya optimis.
```

Gambar 4.4 Contoh baris terpisah

Dari gambar diatas terlihat bahwa posting dari akun facebook Kemenpar memiliki beberapa paragraf, yang kemudian akan diubah menjadi satu baris menjadi seperti Gambar 4.5.



Gambar 4.5 Contoh hasil penggabungan baris

Potongan Gambar 4.5 menunjukkan hasil dari penggabungan beberapa paragraph untuk menjadi satu baris.

4.3.4 Perancangan Penghapusan Data yang Terduplikasi

Data yang didapatkan dari proses sebelumnya masih terdapat data yang sama, oleh karena itu harus dihapus salah satunya hingga setiap data yang ada merupakan data yang unik. Proses penghapusan data dilakukan dengan menggunakan *plugins* TextFX pada Notepad++. Hal ini dilakukan sebagai bentuk antisipasi jikalau masih terdapat kalimat yang terduplikasi.

4.3.5 Perancangan *Tokenizing*

Setelah semua pembersihan serta pemilihan atribut pada dataset telah dilakukan, maka selanjutnya adalah proses *tokenizing*.

Proses ini bertujuan untuk menjadikan satu data teks menjadi *token-token* (per kata) agar dapat diproses lebih lanjut menggunakan *Word2Vec*.

Tokenization dilakukan dengan cara menggunakan Class TokenizerFactory yang telah tersedia pada *library Word2vec Deeplearning4J*. Hasil dari proses *tokenization* ini nanti digunakan sebagai parameter proses *training* Word2Vec.

4.4 Perancangan Leksikon Bahasa Indonesia

Dalam melakukan pembenaran kata bahasa Indonesia yang sesuai, maka diperlukan juga pengayaan lekiskon Bahasa Indonesia agar jenis kata yang terdapat pada leksikon semakin banyak dan pembenaran kata yang dilakukan semakin akurat serta sesuai dengan kaidah bahasa Indonesia.

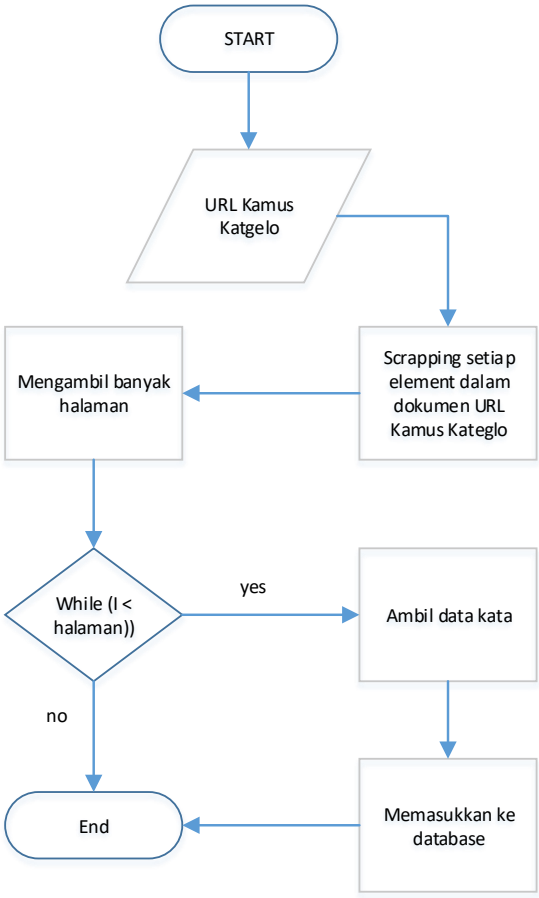
4.4.1 Perancangan Pengumpulan Data

Pengayaan dilakukan dengan cara mengumpulkan kata/*corpus* bahasa Indonesia yang terdapat di Internet maupun KBBI daring ataupun sistem lain yang tersebar di Internet yang mengandung kata bahasa Indonesia.

4.4.1.1 Data Kateglo

Kateglo (kamus, tesaurus dan glosarium) merupakan bentuk daring dari layanan kamus, tesaurus dan glosarium bahasa Indonesia. Data yang ada pada Kateglo berasal dari berbagai sumber yaitu KBBI III, KBBI IV, Wikipedia, Kateglo, serta kontirbusi publik.

Untuk mengumpulkan data dari kateglo maka dibuatkan sebuah program *scrapping* menggunakan bahasa PHP dan *framework* Lrdc Framework agar dapat mendapatkan semua daftar kata beserta atributnya dari *website* Kateglo. Untuk lebih jelas alurnya digambarkan pada Gambar 4.6.

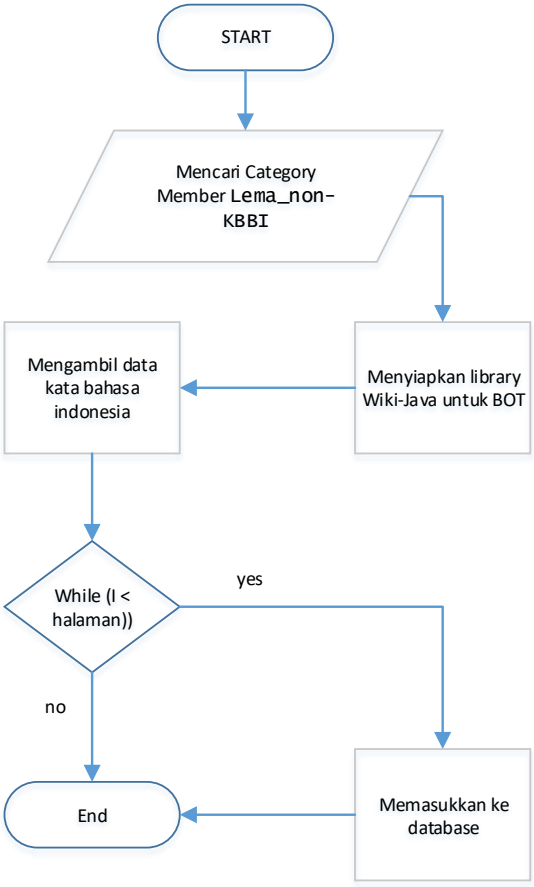


Gambar 4.6 Diagram Alur *Crawling* Kateglo

4.4.1.2 Data *Wiktionary* Indonesia

Wiktionary Bahasa Indonesia merupakan salah satu ensiklopedia kamus bahasa Indonesia yang berbasis online. Pertumbuhan kata yang terdapat di Wiktionary cukup pesat dikarenakan kontributor yang aktif menambahkan entri-entri kata beserta artinya kedalam sistem. Oleh karena itu untuk melengkapi data yang telah diambil dari Kateglo maka kata-kata yang ada dalam Wiktionary Indonesia juga akan diambil.

Pengambilan kata dalam Wiktionary Indonesia menggunakan library Java yang bernama Wiki-Java yang merupakan *bot* untuk Wiki. *Bot* yang dibuat akan mengambil data dari alamat sebuah artikel di Wiktionary yang kemudian akan disimpan kedalam database. Untuk lebih jelasnya dapat dilihat pada Gambar 4.7.



Gambar 4.7 Diagram Alur Wiktionary

4.4.1.3 *Google Translate* Bahasa Indonesia

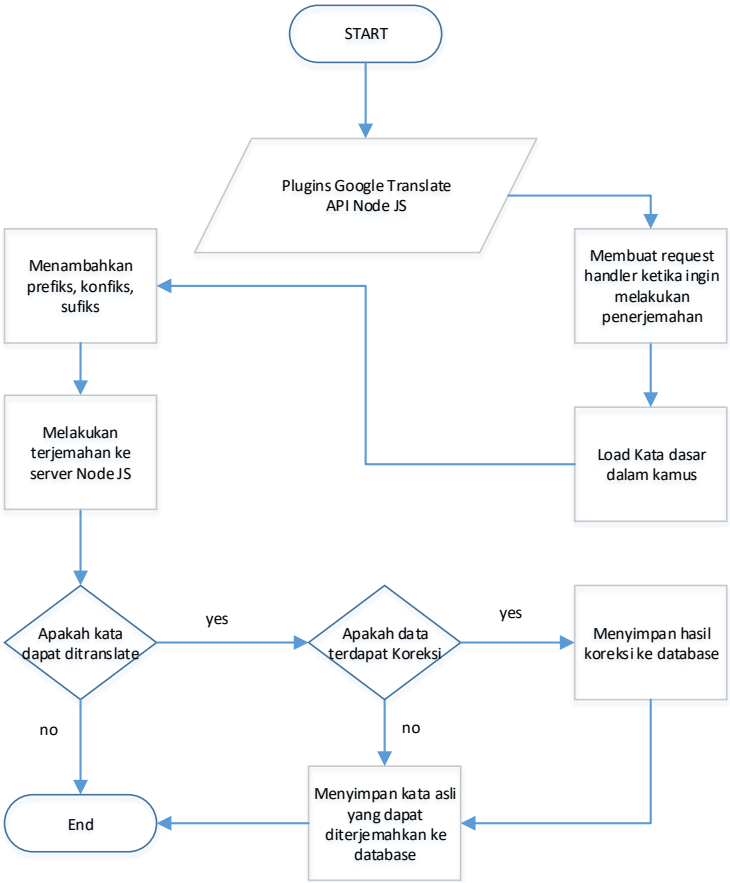
Pengambilan data selanjutnya ialah memanfaatkan layanan Google Translate. Dengan asumsi bahwa setiap kata yang benar dalam Bahasa Indonesia pasti memiliki terjemahan dalam Bahsa Inggris dan apabila tidak sesuai dengan ejaan maka terdapat pembenaran katanya. Hal ini dilakukan bertujuan untuk menutupi kekurangan pada sumber-sumber data seblumnya yang mana sedikit mengakomodir kata turunan dari sebuah kata dasar, misal kata yang berimbuhan, kata pasif maupun kata yang berulang.

Untuk dapat mengakses *Google Translate* dan dapat mengambil hasil terjemahannya maka dapat menggunakan library NodeJS yang bernama google-translate-api. Library tersebut bersifat gratis dan tidak terbatas. Untuk lebih lengkap library apa saja yang dipakai dapat dilihat pada Tabel 4.4.

Tabel 4.4 Library NodeJs

Nama Library	Deskripsi
express	Kerangka kerja web minimalis yang cepat dan tidak terbuka, untuk node.
Morgan	HTTP *request logger middleware* untuk node.js
Body-parser	Melakukan *parsing* dari *request bodies* yang ada dalam sebuah *middleware* sebelum *handlers*, terdapat pada bagian setelah *req.body property*.
Google-translate-api	Untuk melakukan *request* ke Google Translate API dari sebuah kata yang ingin diterjemahkan.

Cara untuk dapat mendapatkan kata bahasa Indonesia mana yang sesuai adalah dengan membuat daftar imbuhan yang ada dalam bahasa Indonesia mulai dari Prefiks, Konfiks hingga Sufiks dan kemudian dilakukan pengecekan satu per satu ke Google Translate untuk menghasilkan terjemahan bahasa Inggrisnya. Lebih jelasnya terdapat pada Gambar 4.8.



Gambar 4.8 Diagram Alur Google Translate

4.4.2 Perancangan Sistem Leksikon

Untuk membangun sistem leksikon bahasa Indonesia dalam tugas akhir ini dibutuhkan bantuan library untuk *indexing* dan

mendukung fitur *full-text search* yaitu *Solr*. Hal ini dibutuhkan agar proses pencarian kata dalam lekiskon/kamus dapat dilakukan dengan cepat dan akurat. Proses perancangan nantinya akan memindahkan dan menglakukan *indexing* data dari database leksikon yang telah terbentuk pada MySQL kedalam Solr.

4.5 Perancangan Pembuatan Model *Word2Vec*

Hasil tokenisasi yang berasal dari tahapan sebelumnya akan direpresntasikan kedalam vektor menggunakan Word2vec yang nantinya akan menghasilkan sebuah model melalui proses training. Terdapat 3 tahapan untuk melakukan pembuatan model hingga dapat menasilkan model terbaik yang akan digunakan sebagai acuan dalam pembuatan program normalisasi teks bahasa Indonesia.

4.5.1 Perancangan Pembagian Dataset

Tahapan pertama yang dilakukan adalah membagi dataset menjadi 2 bagian yaitu data training dan data sampel pengujian. Tujuan tahapan ini adalah melakukan *cross-validation* agar nantinya program yang memanfaatkan model dari *word2vec* dapat di evaluasi akurasinya dalam melakukan normalisasi. Data sampel untuk pengujian yang akan digunakan ialah sebanyak seribu data yang diambil dari data training.

4.5.2 Perancangan *Training Word2vec*

Tahapan *training word2vec* bertujuan untuk menghasilkan model *word2vec*. Untuk melakukan hal ini maka diperlukan bantuan dari library Deeplearning4J. *Library* ini berbasis Java dan memiliki banyak keunggulan salah satunya dapat mengubah beberapa parameter trainiing yang akan dilakukan.

Proses *training* akan dilakukan berulang kali dengan komposisi parameter yang berbeda-beda hingga menemukan model yang terbaik. Berikut adalah parameter yang akan diubah dalam proses *training*:

1. *Learning Algorithm*: parameter ini merupakan salah satu parameter penting karena akan menentukan kandidat-

kandidat yang dihasilkan dari proses prediksi menggunakan model yang telah terbentuk. Terdapat 2 *learning algorithm*, yaitu *Continuous Skip-Gram* dan *Continuous Bag of Words*.

2. *Layer Size*: parameter ini digunakan untuk menentukan dimensionalitas dari sebuah vector, nilainya mulai dari 0-1000.
3. *Window Size:* parameter yang menentukan berapa banyak kata setelah dan sesudah dari sebuah kata yang termasuk konteks dari kata tersebut.
4. *Training Algorithm:* dalam proses training word2vec mempunyai dua training algorithm yaiu *Heirarchical Softmax* (HS) dan *Negative Sampling* (NS). HS mampu bekerja dengan baik pada kata yang tidak sering muncul, sedangkan NS bekerja dengan baik pada kata yang sering muncul dan memiliki dimensi vektor yang rendah.
5. *MinimumWordFrequency:* parameter yang menentukan bahwa sebuah kata akan ditraining apabila kata tersebut muncul minimal dalam jumlah yang telah ditentukan.
6. *Iterations:* berapa banyak *neural net* dapat mengubah koefisiennya dalam sekali proses mini training.
7. *Epochs:* angka untuk menentukan berapa banyak proses training dilakukan.

4.5.3 Perancangan Evaluasi Model *Word2vec*

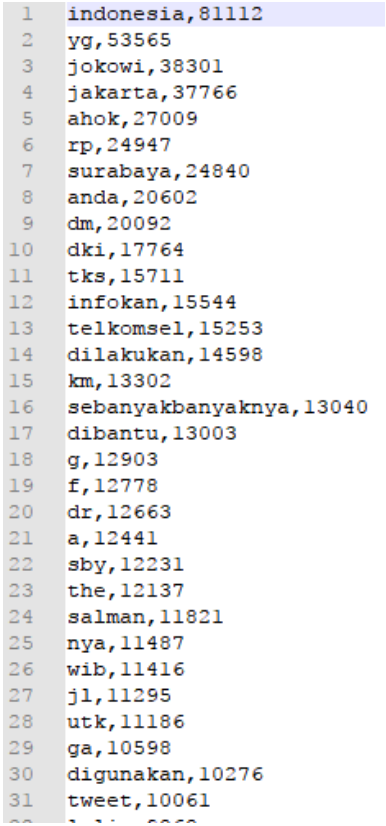
Dari tahapan training yang memilki kombinasi parameter akan terbentuk beberapa model. Model-model yang terbentuk akan dievaluasi satu per satu ketepatan dalam menghasilkan kandidat pembenaran kata dari seratus kata yang tidak baku. Sebelum melakukan evaluasi model, terdapat beberapa tahapan yang harus dilakukan.

4.5.3.1 Perancangan Menghitung Kemunculan Kata Baku dan Tidak Baku

Tahapan ini bertujuan untuk menghitung berapa kali sebuah kata tidak baku maupun baku muncul dalam data training yang telah ada. Perhitungan kemunculan kata ini dibantu oleh program Java dan daftar kata-kata baku yang sesuai dengan KBBI yang telah didapat dari pengambilan data pada *website*

Kateglo pada sub bab 4.4.1.1 untuk membedakan antara baku dan tidak baku.

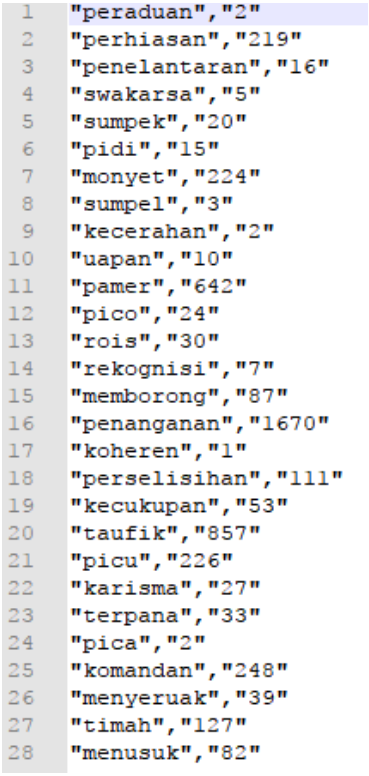
Daftar kata baku yang telah terbentuk dari proses ini tidak digunakan karena telah memenuhi KBBI, sedangkan daftar kata tidak baku akan diujikan kedalam model-model yang telah terbentuk agar dapat diprediksi pembenaran katanya. Contoh hasil daftar tidak baku dan baku dapat dilihat pada Gambar 4.9.



```
1  indonesia,81112
2  yg,53565
3  jokowi,38301
4  jakarta,37766
5  ahok,27009
6  rp,24947
7  surabaya,24840
8  anda,20602
9  dm,20092
10 dki,17764
11 tks,15711
12 infokan,15544
13 telkomsel,15253
14 dilakukan,14598
15 km,13302
16 sebanyakbanyaknya,13040
17 dibantu,13003
18 g,12903
19 f,12778
20 dr,12663
21 a,12441
22 sby,12231
23 the,12137
24 salman,11821
25 nya,11487
26 wib,11416
27 jl,11295
28 utk,11186
29 ga,10598
30 digunakan,10276
31 tweet,10061
```

Gambar 4.9 Contoh hasil kata tidak baku

Gambar diatas terdiri dari dua unsur yaitu kolom pertama kata tidak baku dan kolom kedua adalah frekuensi kemunculannya.



```
"peraduan","2"
"perhiasan","219"
"penelantaran","16"
"swakarsa","5"
"sumpek","20"
"pidi","15"
"monyet","224"
"sumpel","3"
"kecerahan","2"
"uapan","10"
"pamer","642"
"pico","24"
"rois","30"
"rekognisi","7"
"memborong","87"
"penanganan","1670"
"koheren","1"
"perselisihan","111"
"kecukupan","53"
"taufik","857"
"picu","226"
"karisma","27"
"terpana","33"
"pica","2"
"komandan","248"
"menyeruak","39"
"timah","127"
"menusuk","82"
```

Gambar 4.10 Contoh hasil kata baku

Gambar 4.10 terdiri dari dua unsur yaitu kolom pertama kata baku dan kolom kedua adalah frekuensi kemunculannya.

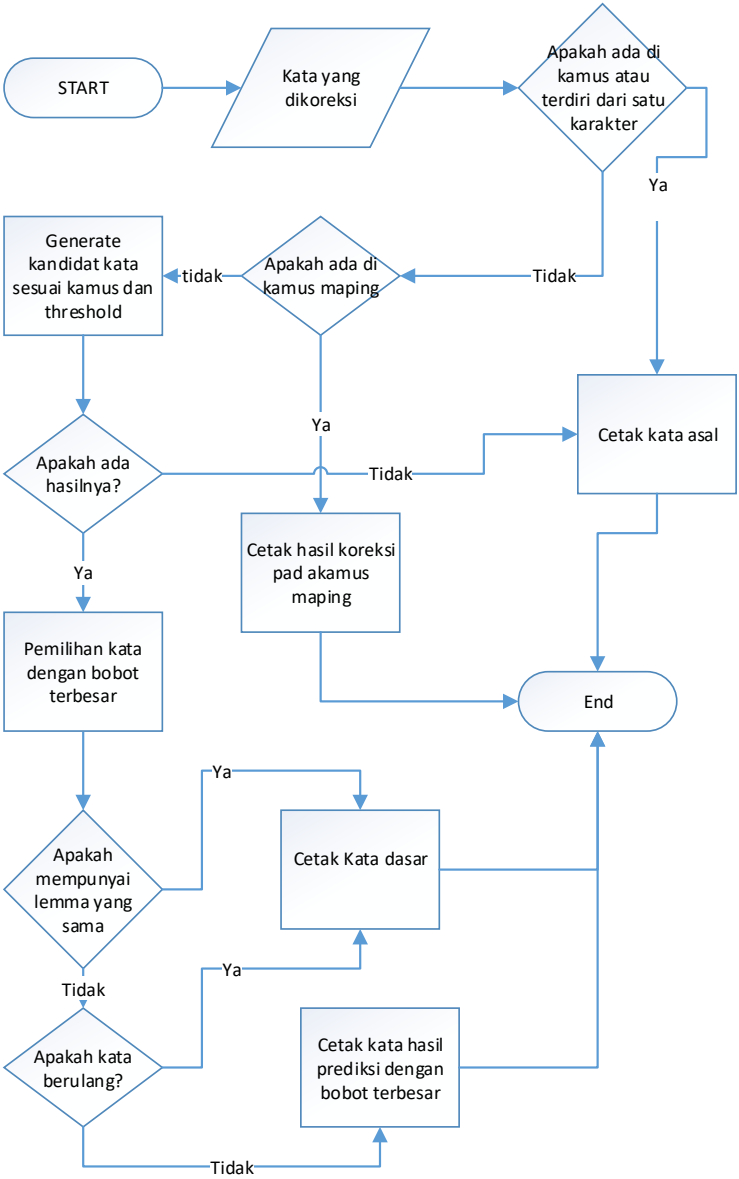
Dari tahapan inilah nantinya seratus kata yang berada dalam kamus akan diujikan terhadap model untuk menghitung tingkat akurasi model.

4.6 Perancangan Sistem Normalisasi Teks

4.6.1 Sistem Normalisasi Teks

4.6.1.1 Perancangan Sistem

Untuk melakukan perancangan sistem maka dibuatlah diagram alur, berikut adalah gambar diagram alur dari sistem.



Gambar 4.11 Diagram Alur Sistem Normalisasi Teks

Dari Gambar 4.11 dapat dilihat bahawa koreksi dilakukan per kata sehingga hingga menemukan hasil koreksinya. Apabila kata sudah ada di kamus atau terdiri dari satu karakter maka akan mengambalikan kata tersebut, jika tidak maka akan melalui tahapan pemeriksaan sesuai dengan gambar.

4.6.1.2 Pembobotan Hasil Prediksi

Mekanisme pembobotan yang akan dilakukan adalah dengan menggabung antara beberapa algoritma *distance* seperti *levenshtein distance* dan *jaro-winkler* serta mempertimbangkan hasil *word2vec*. Berikut adalah formula pembobotan.

$$Wtotal = (0.5 \times i) + (0.25 \times l) + (0.25 \times j) \quad \text{(persamaan 4)}$$

Dimana:
Wtotal = bobot total
i = posisi ditemukannya kandidat *word2vec*
l = skor *levenshtein distance*
j = skor *jaro-winkler distance*

4.6.2 Pengujian

4.6.2.1 Perancangan Prediksi Pembenaran Kata Tidak Baku dan Evaluasinya

Daftar kata tidak baku yang telah terbentuk diurutkan berdasarkan kemunculan kata yang paling banyak hingga sedikit kemudian diambil seribu teratas untuk diprediksi pembenaran katanya menggunakan model-model yang telah terbentuk.

Proses evaluasi dilakukan dengan cara menghitung secara manual berapa banyak kata tidak baku yang terdapat pembenarannya pada kandidat kata yang dihasilkan dibagi dengan total kata tidak baku yaitu seribu kata.

Dari proses evaluasi ini nantinya akan ditemukan model mana yang memiliki akurasi terbaik yang akan digunakan pada tahapan selanjutnya. Model terbaik memperlihatkan bahwa parameter-parameter yang digunakan telah sesuai dengan dataset yang ada.

4.6.2.2 Rancangan Pengujian

Pengujian yang akan dilakukan adalah dengan mengujikan data testing kedalam sistem normalisasi teks dengan melalui beberapa percobaan. Percobaan-percobaan yang akan dilakukan adalah dengan mengubah parameter threshold. Kemudian dari beberapa hasil yang didapat akan analisis dan kemudian diberikan kesimpulan.

Halaman ini sengaja dikosongkan

BAB V
IMPLEMENTASI

Pada bab ini, akan dijelaskan mengenai implementasi dari perancangan yang telah dilakukan sesuai dengan metode pengembangan yang dibuat. Bagian implementasi akan menjelaskan mengenai lingkungan implementasi, pembuatan fitur-fitur aplikasi dalam bentuk kode, serta pengujian aplikasi.

5.1 Lingkungan Implementasi

Pengembangan aplikasi ini menggunakan komputer dengan spesifikasi pada Tabel 5.1.

Tabel 5.1 Spesifikasi

Prosesor	Intel® Core™ i5-2400 CPU @ 3.10GHz
Memory	8 GB RAM
Sistem Operasi	*Windows 10 Pro*
Arsitektur Sistem	*64-bit Operating System, x64-based processor*

Aplikasi dikembangkan dengan menggunakan beberapa teknologi seperti editor, database, server, bahasa pemrograman, dan *library* yang disajikan dalam Tabel 5.2.

Tabel 5.2 Daftar *Library*

Webserver	Apache 2.4, NodeJS
Bahasa Pemrograman	PHP 5.6.28, Javascript, Java JDK 1.8.0
Database	MySQL

***Editor* (IDE)**	Sublime Text, Notepad++, IntellIJ, Netbeans
Browser	Google Chrome 56
Library	• DeepLearning4J: Word2Vec • Facebook4J • Twitter4J • Wiki-Java Bot • Lrdc Framework • google-translate-api • ExpressJS • OpenCSV • Json-simple • SolrJ • MySQL JDBC Connector • java-string-similarity

5.2 Pembuatan *Crawler*

5.2.1 *Facebook Crawler*

Dalam membuat *Facebook Crawler* diperlukan kustomisasi dari implementasi *library* Facebook4J. *Crawler* yang akan dibuat akan melakukan pengambilan data 41 akun Facebook mulai dari 1 Agustus 2015 hingga sekarang.

Untuk membuat Facebook Crawler ini terdiri dari 1 *class* yang mana mempunyai anggota *class* 1 *main method*, 5 *method* lainnya. Berikut adalah penjelasan dari masing-masing *method* yang ada.

```
public static Facebook getFacebookInstance(String[]
args) {
 ConfigurationBuilder cb = new ConfigurationBuilder();
 cb.setOAuthAppId(args[1]);
 cb.setOAuthAppSecret(args[2]);
 cb.setOAuthAccessToken(args[3]);
 cb.setJSONStoreEnabled(true);
 if (args.length > 4) {
   cb.setHttpProxyHost(args[4]);
   cb.setHttpProxyPort(Integer.parseInt(args[5]));
   cb.setHttpProxyUser(args[6]);
   cb.setHttpProxyPassword(args[7]);
 }
 FacebookFactory ff = new FacebookFactory(cb.build());
 return ff.getInstance(); }
```

Kode 5.1 *Get Facebook Instances*

Kode 5.1 menunjukkan isi kode dari *method* getFacebookInstances yang mana berguna untuk melakukan pemanggilan *Facebook Instance* melalui *Class FacebookFactory* yang mengandung konfigurasi *setOAuthAppId, setOAuthAppSecret, setOAuthAccessToken, setJSONStoreEnabled, setHttpProxyHost, setHttpProxyPort, setHttpProxyUser,* dan *setHttpProxyPassword*. Konfigurasi tersebut diambil dari *argument* yang dimasukkan ketika menjalankan program melalui *command line*.

```
public static Timestamp getMaxTimestamp(Connection db,
String akun) throws SQLException {
   Timestamp maxTS = null;
   Statement st = db.createStatement();
   ResultSet rs = st.executeQuery("SELECT
MAX(created_time) FROM fb_test5 WHERE user='" + akun +
"'");
   if (rs.next()) {
      maxTS = rs.getTimestamp(1);
    }
    rs.close();
    System.out.println(maxTS);
    return (maxTS);
}
```

Kode 5.2 Mengambil waktu terakhir

Kode 5.2 menunjukkan isi kode dari *method* getMaxTimestamp yang berfungsi untuk mengembalikan waktu posting paling baru sebuah user di *Facebook* yang telah disimpan didalam

database. Hal ini digunakan agar tidak adanya data yang terduplikasi dalam waktu yang sama karena program nantinya hanya akan memasukkan kedalam database apabila waktu dari posting yang telah di *crawl* lebih dari nilai yang dihasilkan method getMaxTimestamp.

```
private static LinkedList<String> readAkun(String
path) {
  FileReader fr = null;
  try {
    fr = new FileReader(path);
    BufferedReader textReader = new
BufferedReader(fr);
    String list;
    LinkedList<String> akun = new LinkedList<>();
    while ((list = textReader.readLine()) != null) {
         akun.add(list);
    }
    return akun;
  } catch (FileNotFoundException ex) {

Logger.getLogger(FBSched.class.getName()).log(Level.SE
VERE, null, ex);
   return null;
  } catch (IOException ex) {

Logger.getLogger(FBSched.class.getName()).log(Level.SE
VERE, null, ex);
   return null;
  } finally {
      try {
         fr.close();
      } catch (IOException ex) {

Logger.getLogger(FBSched.class.getName()).log(Level.SE
VERE, null, ex);
      }
  }
}
```

Kode 5.3 Membaca daftar akun FB

Kode 5.3 menunjukkan *method* readAkun yang berguna untuk membaca file daftar akun FB yang akan diambil datanya menggunakan *BufferedReader*. Kemudian akun yang telah dibaca per baris akan dimasukkan kedalam variabel akun yang bertipe *LinkedList<String>*. Setelah semua proses pembacaan selesai maka method akan mengembalikan nilai berupa variabel akun bertipe *LinkedList<String>*.

```
private static void storeJSON(String rawJSON, String
fileName) throws IOException {
        FileOutputStream fos = null;
        OutputStreamWriter osw = null;
        BufferedWriter bw = null;
        try {
            fos = new FileOutputStream(fileName);
            osw = new OutputStreamWriter(fos, "UTF-8");
            bw = new BufferedWriter(osw);
            bw.write(rawJSON);
            bw.flush();
        } finally {
            if (bw != null) {
                try {
                    bw.close();
                } catch (IOException ignore) {
                }
            }
            if (osw != null) {
                try {
                    osw.close();
                } catch (IOException ignore) {
                }
            }
            if (fos != null) {
                try {
                    fos.close();
                } catch (IOException ignore) {
                }
            }
        }
    }
```

Kode 5.4 Menyimpan JSON Post FB

Kode 5.4 menunjukkan isi dari *method* storeJSON yang berguna untuk menyimpan atau mengestrak JSON dari posting FB kedalam hardisk sehingga nantinya dapat dipergunakan sewaktu-waktu jika dibutuhkan.

Kode 5.5 menunjukkan isi dari *method* getCommentFB yang berguna menyimpan komentar dari post tertentu kedalam *database* tabel fb_comments. Pada method tersebut juga berisi proses pemanggilan *method* storeJSON yang digunakan untuk menyimpan file JSON dari raw JSON yang diambil oleh program. Selain itu juga terdapat pengambilan daftar komentar dari sebuah post yang akan disimpan satu per satu kedalam *database* menggunakan perulangan.

```
private static boolean getCommentFB(String id, Facebook
fb, Connection db) {
 try {
    PreparedStatement st2 = db.prepareStatement(
                "INSERT INTO comment (id_comment,
id_post, message, account, account_id, created_time)"
                + " VALUES (?, ?, ?, ?, ?, ?)");
    ResponseList<Comment> com = fb.getPostComments(id);
    String raw2 = DataObjectFactory.getRawJSON(com);
    String fileName2 = "facebook_json/" + id +
"_comments.json";
    storeJSON(raw2, fileName2);
    for (Comment co : com) {
       String com_id = co.getId();
       String com_message = co.getMessage();
       Timestamp com_createdTime = new
java.sql.Timestamp((co.getCreatedTime()).getTime());
       String com_akun = co.getFrom().getName();
       String com_akun_id = co.getFrom().getId();
       st2.setString(1, com_id);
       st2.setString(2, id);
       st2.setString(3, com_message);
       st2.setString(4, com_akun);
       st2.setString(5, com_akun_id);
       st2.setTimestamp(6, com_createdTime);
       try {
          st2.executeUpdate();
       } catch (SQLException e) {
          return false;
       }
     }
     return true;
   } catch (FacebookException ex) {
      return false;
   } catch (IOException ex) {
      return false;
   } catch (SQLException ex) {
      return false;
   }
 }
```

Kode 5.5 Menyimpan Komentar FB

Terdapat 6 kolom yang diisi ketika melakukan insert kedalam database, yaitu kolom *id_comment, id_post, message, account, account_id,* dan *created time.* 6 Kolom tersebut diisi oleh variabel-variabel yang didapat dari pengambilan tiap komentar dari sebuah post FB.

Kemudian untuk melakukan penyimpanan kedalam database diperlukan koneksi kedalam database dan menyiapkan query yang digunakan untuk memasukkan data kedalam database. Apabila seluruh proses dalam method berhasil dijalankan maka

method akan mengembalikan nilai Boolean true dan jika tidak maka false.

```
Facebook fb = getFacebookInstance(args);

Class.forName("com.mysql.jdbc.Driver");

String url = "jdbc:mysql://localhost:7717/fb_crawler";
String username = "root";
Connection   db   =   DriverManager.getConnection(url,
username, "");
int insertedRows = 0;

PreparedStatement st = db.prepareStatement("INSERT INTO
fb (fb_id, message, story, created_time, user)"
                    + " VALUES (?, ?, ?, ?, ?)");
```

Kode 5.6 Koneksi *Database* Crawler

Kode 5.6 menunjukkan pemanggilan *method* getFacebookInstances serta melakukan koneksi kedalam *database* fb_crawler. Kemudian menambahkan *prepared statement* yang digunakan untuk memasukkan post FB dan komentar FB.

```
LinkedList<String> listAkun = readAkun(args[0]);
if (listAkun.isEmpty()) {
   System.out.println("File Tidak Ditemukan");
   System.exit(0);
}
```

Kode 5.7 Membaca Akun

Kode 5.7 menunjukkan pemanggilan readAkun untuk membaca daftar akun dari file yang telah diinputkan pada argument pertama command line. Kemudian hasil pemanggilan disimpan kedalam LinkedList. Apabila variabel listAkun kosong maka program akan berhenti dikarenakan file tidak ditemukan.

Crawler yang dibuat terdapat dua jenis, yaitu yang pertama adalah untuk mengambil data Facebook mulai dari 1 Agustus 2015 hingga waktu tertentu dan untuk mengambil data secara berkala sejak waktu data terakhir yang ada pada *database*.

```
String date = "2015-08-01";
SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-
dd");
Calendar c = Calendar.getInstance();
c.setTime(sdf.parse(date));
c.add(Calendar.DATE, -1);
Date since2 = c.getTime();
```

Kode 5.8 Mengambil data mulai Agustus 2015

Kode 5.8 menunjukkan tanggal mulai *crawler* mengambil data posting *Facebook*. Data yang pertama kali diambil adalah posting pada tanggal 1 Agustus 2015, yang kemudian ditampung kedalam variabel since2.

```
Timestamp maxTS = getMaxTimestamp(db, akun);
SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-
dd");
Calendar c = Calendar.getInstance();
c.setTime(maxTS);
c.add(Calendar.DATE, -1);
Date since2 = c.getTime();
```

Kode 5.9 Mengambil data mulai tanggal terbaru pada database

Kode 5.9 menunjukkan tanggal mulai *crawler* mengambil data posting *Facebook*. Data yang pertama kali diambil adalah posting adalah sesuai dengan waktu terbaru yang ada di *Facebook*, jadi *crawler* akan mengambil data posting lebih dari tanggal terbaru.

```
c.setTime(new Date());
c.add(Calendar.DATE, 1);
Date until2 = c.getTime();
System.out.println("ambil data " + akun + " dari " +
since2 + " - " + new Date());
long diff = until2.getTime() - since2.getTime();
int interval = (int) (diff / (60 * 60 * 1000)) / 24;
c.setTime(maxTS);
```

Kode 5.10 Tanggal terakhir diambil

Kode 5.10 menunjukkan kode untuk menetapkan tanggal terakhir *crawler* melakukan pengambilan data, yaitu tanggal saat ini pada sistem. Kemudian interval antara tanggal awal pengambilan data dan terakhir dihitung agar dapat menentukan berapa kali perulangan yang dilakukan. Waktu program ditetapkan sesuai dengan tanggal awal pengambilan data.

```
for (int j = 0; j < interval; j++) {
 since = sdf.format(c.getTime());
 c.add(Calendar.DATE, 1);
 until = sdf.format(c.getTime());
 ResponseList<Post> posts;
 posts = fb.getPosts(akun, new
Reading().limit(100).since(since).until(until));
 new File("facebook_json").mkdir();
 for (Post post : posts) {
  Timestamp createdTime = new
java.sql.Timestamp((post.getCreatedTime()).getTime());
   if (maxTS == null || createdTime.after(maxTS)) {
   //if akan dihilangkan jika crawler dijalankan
pertama kali
    String fb_id = post.getId();
    String message = post.getMessage();
    String story = post.getStory();
    Post post1 = fb.getPost(post.getId());
    String raw1 = DataObjectFactory.getRawJSON(post);
    String fileName1 = "facebook_json/" +
post1.getId() + "_post.json";
    storeJSON(raw1, fileName1);
    getCommentFB(post1.getId(), fb, db);
    st.setString(1, fb_id);
    st.setString(2, message);
    st.setString(3, story);
    st.setTimestamp(4, createdTime);
    st.setString(5, akun);
    try {
     insertedRows += st.executeUpdate();

    } catch (SQLException e) {
     e.printStackTrace();
    }
    System.out.println(akun + ", " + createdTime + ":
sukses" + insertedRows);
    if(getCommentFB(fb_id, fb, db)==false){
     System.out.println("Gagal menambahkan comment");
    }else {
     System.out.println("Komen ditambahkan");
    }
   }
  }
 }
 System.out.println("Number of rows inserted: " +
insertedRows);
}
st.close();
```

Kode 5.11 Perulangan pengambilan data Facebook

Kode 5.11 adalah kode program untuk mulai mengambil data dari Facebook, yang dilakukan sebanyak nilai interval yang didapat dari selisih waktu mulai dengan waktu akhir pengambilan data. Untuk membaca post dari sebuah akun maka

method yang digunakan adalah getPost dengan parameter nama akun, opsi dari *Reading* yang mana disini menggunakan batas 100 post sejak waktu yang telah ditentukan hingga satu hari setelah waktu mulai. Apabila posting yang diambil waktunya dibuatnya setelah maxTS atau maxTS tidak ada isinya maka mulai diambil atribut-atribut post untuk disimpan. Atribut yang disimpan dapat dilihat pada Tabel 5.3.

Tabel 5.3 Variabel Fb

Nama Varibale	**Nama Method**
String fb_id	Post.getId()
String message	Post.getMessage()
String story	Post.getStory()
Timestamp createdTime	new java.sql.Timestamp((post.getCreatedTime()).getTime()
String akun	akun

Setelah semua variabel disimpan, maka kelima variabel diatas akan dimasukkan kedalam database pada tabel fb dan method getCommentFB dipanggil untuk mengambil komentar pada post tersebut.

```
java -Djdk.http.auth.tunneling.disabledSchemes="" -jar
FBSched.jar "akun1.txt" "551265581691308"
"8ab8b0b15efc40df10fc4de9d0da16c8"
"551265581691308|WlUm1zzZR7hW9OEf6Oi8HNMIguI"
"proxy.its.ac.id" "8080" "username@mhs.is.its.ac.id"
"password"
```

Kode 5.12 Perintah *crawler* facebook

Kemudian untuk menjalankan crawler tersebut akan dilakukan penjadwalan menggunakan Task Scheduler yang terdapat pada sistem operasi Windows yang akan menjalankan script dengan isi perintah seperti pada Kode 5.12.

5.2.2 *Twitter Crawler*

Pembuatan *crawler* untuk *Twitter* tidak jauh berbeda dengan Facebook. Terdapat beberapa kesamaan *class* member diantara kedua kode program *crawler* Facebook maupun Twitter. Kesamaannya adalah keduanya mempunyai method storeJSON, untuk itu method storeJSON tidak akan ditampilkan lagi pada sub bab ini.

Untuk perbedaannya adalah dari sisi waktu pengambilan data. Aturan dari Twitter API yang hanya memperbolehkan mengakses posting twitter hingga dua minggu sebelum hari pengambilan data. Jadi, semisal mengambil posting 3 minggu sebelumnya maka Twitter API akan mengembalikan nilai null.

```
public static Timestamp getMaxTimestamp(Connection db,
String akun) throws SQLException {
 Timestamp maxTS = null;
 String where = "";
 String[] tes = akun.split("\\s+OR\\s+");
 for (int i=0;i<tes.length;i++){
  String coba ="";
  if(i>0){
        coba = " OR `message` LIKE ";
  }
  else {
        coba = " `message` LIKE ";
  }
  where+=coba+"'%"+tes[i]+"%'";
 }
 Statement st = db.createStatement();
 ResultSet      rs      =      st.executeQuery("SELECT
MAX(created_time) FROM tw_test2 WHERE"+where);
 if (rs.next()) {
  maxTS = rs.getTimestamp(1);
 }
 rs.close();
 System.out.println(maxTS);
 return (maxTS);
}
```

Kode 5.13 Pengambilan waktu terakhir twitter

Kode 5.13 menunjukkan isi kode dari *method* getMaxTimestamp yang berfungsi untuk mengembalikan waktu posting paling baru sebuah user di *Twitter* yang telah disimpan didalam *database*. Hal ini digunakan agar tidak adanya data yang terduplikasi dalam waktu yang sama karena program nantinya hanya akan memasukkan kedalam *database*

apabila waktu dari posting yang telah di *crawl* lebih dari nilai yang dihasilkan method getMaxTimestamp.

```
public static Long getTweetId(Connection db, String
akun) throws SQLException {
 Long maxId = 0L;
 String where = "";
 String[] tes = akun.split("\\s+OR\\s+");
 for (int i=0;i<tes.length;i++){
        String coba ="";
        if(i>0){
         coba = " OR `message` LIKE ";
        }
        else {
         coba = " `message` LIKE ";
        }
         where+=coba+"'%"+tes[i]+"%'";
 }
 Statement st = db.createStatement();
 ResultSet rs = st.executeQuery("SELECT MAX(id) FROM
`tw_test2` WHERE"+where);
 if (rs.next()) {
     maxId = rs.getLong(1);
 }
 rs.close();
 System.out.println(maxId);
 return (maxId);
}
```

Kode 5.14 Mengambil id twitter terakhir

Kode 5.14 menunjukkan isi kode dari *method* getTweetId yang berfungsi untuk mengembalikan id dari posting paling baru sebuah user di *Twitter* yang telah disimpan didalam database. Nantinya data terbaru yang akan dimasukkan kedalam database hanyalah data yang mempunyai id paling terbaru dari nilai id terakhir pada *database*.

```
ConfigurationBuilder cb = new ConfigurationBuilder();
 cb.setOAuthConsumerKey("gOSXHsxvpfb7GblsXg6f8mh9f");

cb.setOAuthConsumerSecret("zfzkQSwsHx6mzLGjWeMT9iFo1SM
q2gNRhcAtEZA3o8hrzbWEsL");
 cb.setOAuthAccessToken("204791971-
oAb8H4W2F4o8FL3yexXQF8fyfawUCP4AXTkU93CP");

cb.setOAuthAccessTokenSecret("iS5fDqchnIdrm043Oo6EpzsI
JZxmt23AkQJcu78MeiZZZ");
 cb.setJSONStoreEnabled(true);
 if (args.length > 1) {
     cb.setHttpProxyHost(args[1]);
     cb.setHttpProxyPort(Integer.parseInt(args[2]));
     cb.setHttpProxyUser(args[3]);
     cb.setHttpProxyPassword(args[4]);
 }
 Twitter twitter = new
TwitterFactory(cb.build()).getInstance();
```

Kode 5.15 Get Twitter Instances

Kode 5.15 adalah sebuah potongan kode untuk membuat sebuah instances yang menhubungkan client dengan twitter api menggunakan *OAuth Consumer Secret* dan *OAuth Access Token* yang telah didapatkan dari website Twitter API. Kode diatas juga menjelaskan terdapat opsi untuk menambahkan *proxy server* beserta auntentikasinya apabila koneksi melewati sebuah *proxy*.

```
Class.forName("com.mysql.jdbc.Driver");
String url =
"jdbc:mysql://localhost:7717/twitter_crawler";
String username = "root";
Connection db = DriverManager.getConnection(url,
username, "");
int insertedRows = 0;
PreparedStatement st = db.prepareStatement(
  "INSERT INTO tw_test2 (id, message, account,
latitude, longitude, created_time)"
  + " VALUES (?, ?, ?, ?, ?, ?)");
```

Kode 5.16 Koneksi dengan *database* twitter

Kode 5.16 merupakan potongan kode program untuk melakukan koneksi dengan *database* twitter_crawler dan membuat sebuah *Prepared Statement* untuk memasukkan data kedalam *database*.

```
Query query = new Query(args[0]);
Timestamp maxTS = getMaxTimestamp(db, args[0]);
Long twId = getTweetId(db, args[0]);
query.setSinceId(twId);
query.setLang("id");
QueryResult result;
```

Kode 5.17 Set timestamp dan id

Kode 5.17 menunjukkan untuk melakukan penetapan tanggal terakhir dan id post terakhir sebagai dasar pengambilan data terbaru pada twitter.

Kode 5.18 digunakan untuk melakukan pengambilan data twitter sesuai dengan *query* pencarian yang ada. Pada implementasi program ini query pencarian yang dilakukan ialah mencari semua twitter yang mengandung *username* 1 atau lebih akun menggunakan fungsi OR. Data twitter yang didapatkan akan diperiksa terlebih dahulu apakah sudah lebih baru Id-nya dan waktunya agar bisa dimasukkan kedalam database. Pada proses ini juga sebuah file json Twitter akan disimpan kedalam hardisk untuk digunakan sebagai keperluan berikutnya. Proses ini akan berulang hingga hasil yang didapatkan dari pengambilan data twitter telah habis dan telah dimasukkan kedalam *database*.

```
do {
 result = twitter.search(query);
 new File("twitter_json").mkdir();
 List<Status> tweets = result.getTweets();
 for (Status tweet : tweets) {
  Timestamp createdTime = new java.sql.Timestamp(
    (tweet.getCreatedAt().getTime()
  ));
  if (maxTS == null || createdTime.after(maxTS)) {
   String user_id = tweet.getUser().getScreenName();
   String msg = tweet.getText();
   long id = tweet.getId();
   GeoLocation geo = tweet.getGeoLocation();
   Double lat = 0.0;
   Double longi = 0.0;
   if (geo != null) {
    lat = geo.getLatitude();
    longi = geo.getLongitude();
   }
   st.setLong(1, id);
   st.setString(2, msg);
   st.setString(3, user_id);
   st.setDouble(4, lat);
   st.setDouble(5, longi);
   st.setTimestamp(6, createdTime);
   System.out.println(
     "@" + tweet.getUser().getScreenName()
     + " - " + tweet.getText()
     + " - " + tweet.getCreatedAt());
   String rawJSON =
TwitterObjectFactory.getRawJSON(tweet);
   String fileName = "twitter_json/"
     + tweet.getId()
         + ".json";
   storeJSON(rawJSON, fileName);
   try {
    insertedRows += st.executeUpdate();
   } catch (SQLException e) {
    e.printStackTrace();
   }
     } else if (tweet.getId() == twId) {
   System.out.println(insertedRows);
   System.exit(0);
  }
 }
} while ((query = result.nextQuery()) != null);
System.out.println(insertedRows);
System.exit(0);
```

Kode 5.18 Pengambilan data twitter

Kode 5.19 Menunjukkan sebuah script yang digunakan untuk menjalankan twitter *crawler* dengan *query* pencarian "sapawarga OR e100ss OR radioelshinta". *Script* tersebut nantinya akan dijadwalkan menggunakan Task Scheduler yang terdapat pada Sistem Operasi Windows.

```
java -Djdk.http.auth.tunneling.disabledSchemes="" -jar
TwitterScheduler.jar "sapawargasby OR e100ss OR
radioelshinta" "proxy.its.ac.id" "8080"
"username@mhs.is.its.ac.id" "password"
```

Kode 5.19 Perintah penjadwalan twitter *crawler*

5.3 Pra-pemrosesan Data

5.3.1 Penggabungan Dataset

Data yang ada saat ini masih terpisah untuk masing-masing sosial media. Untuk melakukan proses training, dilakukan penggabungan data yang dituliskan dalam satu file csv.

```
String pathOutput = "
resources\\csv_merge_processed8.csv";
Class.forName("com.mysql.jdbc.Driver");
String url =
"jdbc:mysql://localhost:7717/twitter_crawler";
String url2 = "jdbc:mysql://localhost:7717/fb_crawler";
String username = "root";
String password = "";
Connection db = DriverManager.getConnection(url,
username, password);
Connection db2 = DriverManager.getConnection(url2,
username, password);
Statement st = db.createStatement();
ResultSet rs = st.executeQuery("SELECT DISTINCT message
FROM tw");
Statement st2 = db2.createStatement();
ResultSet rs2 = st2.executeQuery("SELECT DISTINCT
message FROM fb");
while(rs.next()){
 if(rs.getString(1)!=null){
  String word = removeSymbol(oneLine(rs.getString(1)));
  String[] myWord = {word};
  if(writeCSV(myWord, pathOutput)){
        System.out.println("Alhamdulillah, berhasil
nulis data :)");
  }
  else {
        System.out.println("Yah, gagal :(");
  }
 }
}
st.close();
while(rs2.next()){
 if(rs2.getString(1)!=null){
  String word2 =
removeSymbol(oneLine(rs2.getString(1)));
  String[] myWord2 = {word2};
  if(writeCSV(myWord2, pathOutput)){
    System.out.println("Alhamdulillah, berhasil nulis
data :)");
  }
```

```
  else {
        System.out.println("Yah, gagal :(");
  }
 }
}
st2.close();
```

Kode 5.20 Menggabungkan dataset fb dan twitter

Kode 5.20 menunjukkan proses penggabungan dataset yang diambil melalui masing-masing *database* yang menyimpan data *twitter* dan data *facebook*. Dari data tersebut diambil atribut message yang bertipe teks secara unik dari kedua dataset menggunakan *query* mysql. Sebelumnya sistem terlebih dahulu melakukan koneksi kedalam *database* twitter_crwler dan database fb_crawler. Hasil pembacaan data dituliskan dalam file csv dimana url direktorinya telah diinisiasi dalam variabel pathOutput. Sistem akan mengecek apakah method writeCSV dengan parameter myord, dan pathOutput berhasil dieksekusi, jika berhasil maka program akan mengembalikan pesan berhasil, begitupula sebaliknya sehingga nantinya akan didapatkan file csv yang memuat seluruh data gabungan dari dataset fb dan twitter

```
public static boolean writeCSV(String[] word, String
path){
 try {
  CSVWriter writer = new CSVWriter(
        new FileWriter(path, true), ',');
  writer.writeNext(word);
  writer.close();
  return true;
 } catch (IOException e) {
  e.printStackTrace();
  return false;
 }
}
```

Kode 5.21 Membuat file csv

Kode 5.21 menjelaskan proses pembuatan file CSV dengan menggunakan *library* openCSV. Method writeCSV dibuat untuk menuliskan file CSV baris demi baris. Method ini akan mengembalikan *array string* untuk setiap nilai dalam baris.

5.3.2 Penghapusan Tanda Baca dan Simbol

Data yang didapatkan dari hasil *crawling* masih mengandung berbagai karakter tak bermakna yang dapat mengurangi kualitas data pada proses training. Oleh karena itu dilakukan pembersihan data dengan cara menghilangkan berbagai tanda baca dan simbol tertentu sesuai penjabaran pada sub bab 4.3.2.

```
public static String removeSymbol(String word){
 String[] regex = {
  "(RT)",
  "(u\\/)",
  "([0-9]{2}[:.][0-9])\\w+", //menghapus jam
  "\\b[^\\D]+\\b", //menghapus angka
  "(?:\\(odp-[\\w]+\\)|\\(Odp-[\\w]+\\))", //menghapus
redaksi
  "<[^>]+>", //untuk html tags
  "(?:@[\\w_]+)", //untuk mention
  "(?:\\#+[\\w_]+[\\w\\'_\\-]*[\\w_]+)", //untuk
hashtags
  "http[s]?:(?:[a-z]|[0-9]|[$-
_@.&amp;+]|[!*\\(\\),]|(?:%[0-9a-f][0-9a-f]))+",
//untuk link
  "[\\\\\\\˜\\ÿ\\ð\\…\\“\\”\\[\\@\\~\\`\\#\\%\\-
\\+\\.\\^\\:\\,\\;\\&\\*\\(\\)\\$\\!\\/\\_\\<\\>\\?\\'
\\\"\\{\\}\\[\\]\\|\\=\\]]", //semua simbol
  "[^\\x00-\\x7F]+", //remove unicode gajelas
  "(˚C)",
 };
 for (String reg:regex) {
     word = word.replaceAll(reg, "");
 }
 return word;
}
```

Kode 5.22 Menghilangkan tanda baca dan simbol

Kode 5.22 menunjukkan implementasi method removeSymbol menggunakan fungsi regex (regular expression) untuk menghilangkan beberapa karakter yang disebutkan dan menggantinya dengan nilai null atau kosong sehingga data gabungan yang nantinya dituliskan dalam file csv tidak lagi mengandung karakter tak bermakna.

5.3.3 Penggabungan Baris

Proses ini dilakukan untuk menghilangkan *newline* atau jarak untuk menghindari ambiguitas sistem dalam membaca dataset setiap baris.

```
public static String oneLine(String word){
 word = word.replaceAll("\\n", " ");
 return word;
}
```

Kode 5.23. Menggabungkan baris dataset

Kode 5.23 berada dalam *class* oneLine yang juga memanfaatkan *regular expression* yaitu dengan fungsi replaceAll untuk menghapus baris baru dalam data dan menggantinya dengan string kosong sehingga baris yang awalnya terdiri dari beberapa paragraf akan menjadi satu paragraf.

```
String word = removeSymbol(oneLine(rs.getString(1)));
```

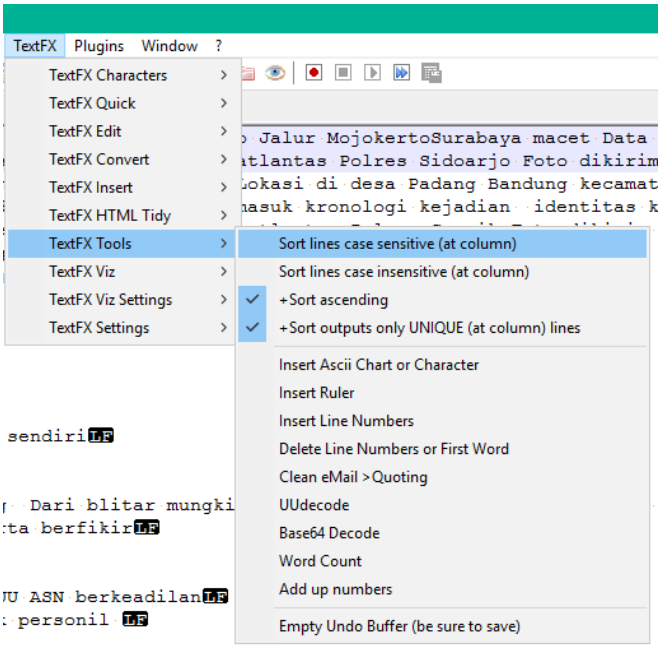
Kode 5.24 Panggil method untuk hapus simbol dan gabung paragraf

Kode 5.24 merupakan pemanggilan method untuk melakukan penggabungan kata dan penghapusan simbol yang telah dijelaskan pada sub bab sebelumnya dari sebuah string.

5.3.4 Penghapusan Kata yang Terduplikasi

Untuk menghapus kata yang terduplikasi, dapat dilakukan menggunakan plugin TextFX yang ada pada notepad++ Seperti pada gambar di bawah ini. Untuk mengaktifkan plugin ini dapat dilakukan dengan langkah berikut:

1. Buka menu *plugin*
2. Arahkan kursor pada *plug*in *manager*, pada dropdown menu, pilih show *plugin manager*
3. Di tab *Plugin Manager Avalilable*, pilih TextFX Characters, tandai kotak centang dan klik Install.



Kode 5.25 Menghapus data yang sama

Lalu Kode 5.25 menjelaskan cara untuk menggunakan plugin ini. Caranya dapat dilakukan beberapa step berikut:

1. Pada menu bar, pilih TextFX
2. Lalu pilih opsi *TextFX Tools* yang akan menampilkan beberapa pilihan.
3. Pastikan mencentang pilihan *Sort outputs only unique (at column) lines* untuk menghapus data duplikasi.
4. Untuk melihat hasilnya, tekan pilihan *Sort lines case sensitive (at column).*

5.3.5 Tokenizing

Tokenization dengan membuat pembuatan token token setiap kata dilakukan dengan cara menggunakan Class TokenizerFactory yang telah tersedia pada *library Word2vec Deeplearning4J*.

```
String word = removeSymbol(oneLine(rs.getString(1)));
TokenizerFactory t = new DefaultTokenizerFactory();
t.setTokenPreProcessor(new LowCasePreprocessor());
List<String> arrayWord =
t.create(word.toLowerCase()).getTokens();
```

Kode 5.26 Proses Tokenizing

Kode 5.26 menjelaskan pembuatan instance untuk proses tokenizing. Method setTokenPreProcessor digunakan untuk menetapkan token pre processor yang digunakan dengan setiap tokenizer. Preprocessor yang digunakan pada proses *tokenizing* disini adalah *LowCasePreprocessor* agar nantinya proses *tokenizing* tidak menghapus angka.

5.4 Pembuatan Leksikon Bahasa Indonesia

5.4.1 Pengumpulan Data

5.4.1.1 Pengumpulan Data Kateglo

```
<?php
ini_set('max_execution_time', 0);
require( 'csc_curl.php' );
require( 'csc_pdo.php' );
```

Kode 5.27. Konfigurasi awal *crawler*

Kode 5.27 menjelasakan untuk melakukan scraping data digunakan *library* dari Lrdc dari link https://github.com/cscpro/lrdc-fw/tree/master/library untuk kemudian disertakan ke dalam program dengan fungsi *require*. Konfigurasi max_execution_time dengan nilai 0, dimaksudkan agar tidak ada batas waktu yang dikenakan pada saat proses eksekusi script, kemudian memasukkan file dari *library* Lrdc yaitu csc_curl untuk melakukan fungsi cUrl dan csc_pdo untuk menghubungkan dengan database.

```
if(isset($_GET['kata'])){
 $kata = htmlentities($_GET['kata'], ENT_QUOTES);
 $url
='http://kateglo.com/index.php?op=1&phrase='.$kata.'&l
ex=&type=&src=&mod=dictionary&srch=Cari&p=';
 $c = new curl;
 $c->bc = $c->get( $url );
 $db = new csc_pdo('root','','kamusindo');
 $halaman = $c->xp('<ul class="pagination pagination-
sm" style="margin: 0px;">', '</ul>');
 $array = preg_split('/\n/',$halaman);
 $size = sizeof($array);
 $baru =
preg_match('/(?<=\&p\=).*(?=\")/',$array[$size-2],$m);
 $page = (int)$m[0];
 $daftar_kata = array();
```

Kode 5.28 Mendapatkan banyak halaman

Kode 5.28 digunakan untuk mendapatkan total halaman yang akan di *crawl* dari website Katgelo. Total halaman diambil dari element pagination pada halaman *website* kamus kateglo.

```
for($i=1;$i<=$page;$i++){
  $kata = array();
  $curl = new curl;
  $curl->bc = $curl->get($url.$i);
  $items = $curl->xp('<dl class="dl-horizontal">',
'</dl>');
  $array_items=preg_split('/\n/',$items);
  unset($array_items[sizeof($array_items)-1]);
  $count=0;
```

Kode 5.29 Memisahkan elemen turunan dari elemen dl

Dari jumlah halaman yang ada, maka tiap halaman akan diambil kontennya menggunakan perulangan. Konten utama yang akan diambil terdapat pada elemen <dl>. Oleh karena itu, Kode 5.29 menunjukkan cara untuk mendapatkan isi dari elemen <dl> yang kemudian dipisahkan menjadi *array* dengan nama *items*.

```
foreach($array_items as $item){
   $count++;
   if (strpos($item, '</dt>')) {

preg_match('/(?<="\>).*(?=\<\/a\>\<\/dt\>)/',$item,$ha
sil);
    if(sizeof($hasil)!=0){
     $kata['kata'] = $hasil[0];
    }
   }
```

Kode 5.30 Perulangan untuk mendapatkan kata dasar

Kode 5.30 menunjukkan dari setiap items yang telah dipisahkan dari element <dl> pada proses sebelumnya akan di periksa apakah termasuk kata dasar atau atribut lain seperti kelas kata dasar dan artinya. Jika dia mengandung elemen tag <dt> maka dia termasuk kata dasar dan disimpan sementara pada variabel *array* $kata[‘kata’].

```
else if (strpos($item, '</dd>')) {

preg_match('/(?<="\>).*?(?=\<\/span\>\s)/',$item,$hasil
);
    if(sizeof($hasil)!=0){
     //proses mencari lexicon, jika ada lexicon pasti
ada arti karena span diikuti arti
     $kata['lex'] = $hasil[0];
     $arti = NULL;
     $dom = new DOMDocument('1.0', 'UTF-8');
     $internalErrors =
libxml_use_internal_errors(true); //menghilangkan
warning parse
     $dom->loadHTML($item, LIBXML_HTML_NOIMPLIED |
LIBXML_HTML_NODEFDTD);
     libxml_use_internal_errors($internalErrors);
     $xpath = new DOMXpath($dom);
     foreach ($xpath->query('//dd/text()') as
$textNode) {
      $arti .= $textNode->nodeValue;
     }
     foreach ($xpath->query('//dd') as $textNode) {
      $arti2 .= $textNode->nodeValue;
     }
//cek apakah arti cuman berupa simbol tanda panah kanan
saja atau tidak
       if(strlen(trim(preg_replace('/[^A-Za-z0-9 ]/',
'', $arti)))!=0){
         $kata['arti'] = $arti2;
        } else{
      //bila artinya itu ternyata hasil koreksian dari
kata yg ada di kateglo
      $clean = trim(preg_replace('/ +/', ' ',
preg_replace('/[^A-Za-z0-9 ]/',
```

```
        ' ',
urldecode(html_entity_decode(strip_tags($item))))));
      $kata['actual'] = substr($clean, 2);
     }
    }
    else{
     // $DOM = new DOMDocument;
     //untuk melihat hasil pengoreksian dari kateglo
pada element yang tidak ada art dan lexiconnya
     // $str = $curl->xp('<dd>', '</dd>', $item);
     $clean = trim(preg_replace('/ +/', ' ',
preg_replace('/[^A-Za-z0-9 ]/',
      '',
urldecode(html_entity_decode(strip_tags($item))))));
     $kata['actual'] = $clean;
    }
   }
```

Kode 5.31 Mendapatkan kelas dan arti

Kode 5.31 digunakan untuk mendapatkan arti dan tipe dari kata dasar. Apabila kata dasar terdapat pada tag <dt> maka, arti dan kelas kata dasar terdapat pada tag <dd>. Jika di dalam tag <dd> terdapat tag maka kelas kata dapat diambil dari sebuah teks yang terkandung dalam tag dan disimpan sementara pada variabel $kata[‘lex’]. Sedangkan pengambilan arti menggunakan *DOMXPath* untuk mendapatkan seluruh isi teks dari tag <dd> dan akan disimpan kedalam $kata[‘arti’].

Selain itu potongan kode diatas juga akan memetakan secara otomatis apabila kata dasar merupakan kata yang salah dan memiliki pembenaran. Kata yang memiliki pembenaran kata ini biasanya memiliki ciri-ciri yaitu mengandung simbol tanda panah kanan. Setelah simbol tersebut ditemukan makan secara otomatis arti tidak akan disimpan dan diganti dengan penyimpanan sementara pada variabel $kata[‘actual’].

Kode 5.32 menunjukkan bagaimana proses penyimpanan hasil *crawling* kateglo kedalam database. Dalam hal ini dilakukan penyimpanan kedalam dua tabel, tabel yang pertama untuk menyimpan kata, kelas katanya serta arti dengan nama tabel katarev dan tabel kedua untuk menyimpan pembenaran kata apabila terdapat variabel $kata[‘actual’] yang kemudian disimpan pada tabel mapping. Setiap sekali proses pegambilan data pada satu halaman maka akan proses pengambilan halaman selanjutnya akan menunggu selama 3 detik. Hal dibuat untuk

menghindari proses yang terus menerus dalam pengiriman request kepada website.

```
if($count==2){
    $id=0;
    if(isset($kata['actual'])){
     $id = $db-
>i('mapping','kata,actual',[$kata['kata'],$kata['actua
l']]);
    }
    else{
     $id = $db-
>i('katarev','kata,lex,arti',[$kata['kata'],$kata['lex
'],$kata['arti']]);
    }

    if($id==0){
     echo 'gagal insert';
    }
    else {
     echo 'berhasil insert '.$id;
    }
    // array_push($daftar_kata, $kata);
    $count=0;
    $kata = array();
   }
  }
  sleep(3);
 }
}
```

Kode 5.32 Proses penyimpanan kedalam database dari Kateglo

Kode 5.33 merupakan sebuah *handler* apabila sebuah halaman tidak dapat diambil datanya. Apabila hal itu terjadi maka sistem akan menampilkan peringatan tidak ada hasil.

```
else{
 header('Content-Type: application/json');
 echo json_encode(['tidak ada hasil']);
}
```

Kode 5.33 Apabila tidak berhasil mendapatkan data kateglo

5.4.1.2 Pengumpulan Data *Wiktionary* Indonesia

Selain mengumpulkan data dari kateglo, pengumpulan juga dilakukan dari website Wiktionary Indonesia. Pengumpulan yang dilakukan hanyalah mengumpulkan Lema-lema yang tidak ada di KBBI dan berupa Kata turunan.

```
Wiki a = new Wiki("id.wiktionary.org");
a.login("Stezarpriansya", "stezarp12");
String[] hasil = a.getCategoryMembers("Lema_non-
KBBI");
Class.forName("com.mysql.jdbc.Driver");
String url = "jdbc:mysql://localhost:3306/kamusindo";
String username = "root";
Connection db = DriverManager.getConnection(url,
username, "");
int insertedRows = 0;
PreparedStatement st = db.prepareStatement(
  "INSERT INTO nonkbbi (kata)"
  + " VALUES (?)");
for (String res : hasil) {
  st.setString(1, res);
  try {
    insertedRows += st.executeUpdate();
  } catch (SQLException e) {
    continue;
  }
}
```

Kode 5.34 Pengambilan data Wiktionary

Kode 5.34 menunjukkan proses pengambilan lema non KBBI dari Wiktionary. Pertama membuat sebuah objek berupa *class* Wiki dengan parameter alamat *website* wiki yang akan diambil, pada kasus ini menggunakan "id.wiktionary.org". Kemudian melakukan proses *login* meggunakan akun yang telah terdaftar pada website tersebut. Karena data yang diambil merupakan lema non KBBI maka kategori member yang diambil adalah kategori lema non KBBI. Proses selanjutnya ialah memasukkan data yang telah diambil kedalam tabel nonkbbi

5.4.1.3 Pengumpulan Data Google Translate

```
const translate = require('google-translate-api');
var app = express();
app.use(morgan('dev'));
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: false }));
app.get('/translate', function(request,response){
 var query = request.query.word;
 var from = request.query.f;
 var to = request.query.t;
 translate(query, {from: from, to: to}).then(res => {
  var isTranslated = false;
  var initialText = query;
   var translatedText = res.text.toLowerCase();
   if(translatedText !== null){
    isTranslated = true;
   }
   var autoCorrected = res.from.text.autoCorrected;
   var correctedText =
res.from.text.value.toLowerCase().replace(/[^\w\s]/gi,
'');
   var didYouMean = res.from.text.didYouMean;
   response.setHeader('Content-Type',
'application/json');
   response.send(JSON.stringify(
    {
     initialText: initialText,
     translatedText: translatedText,
     isTranslated: isTranslated,
     autoCorrected: autoCorrected,
     correctedText: correctedText,
     didYouMean:didYouMean
    },
    null,
    3)
   );
 }).catch(err => {
   console.error(err);
 });
});
app.listen(3000);
console.log('Server started on port 3000');
module.exports = app;
```

Kode 5.35. Pengumpulan data google translate

Pengumpulan data google translate menggunakan *service* node js melalui *library* google-translate-api serta menggunakan *express* template. Dari beberapa parameter yang didapatkan seperti *word* (kata yang akan di cek), *from*(bahasa asal terjemahan), dan *to* (bahasa tujuan terjemahan). Fungsi *translate* melakukan pengecekan pada setiap kata yang didapatkan untuk mengetahui apakah kata yang diterjemahkan

memiliki makna, atau mengalami pembenaran dari google translate dengan mengambil respon *did you mean* pada google translate sebagai variabel *correctedText*. Sistem akan memeriksa nilai dari variabel *translatedText* yang didapatkan, apabila nilainya tidak sama dengan null, maka sistem memberikan nilai kembalian *true* terhadap variabel *isTranslated*. Seluruh data yang didapatkan dikirim dalam format data json berupa string dengan fungsi *Stringify* dengan 6 parameter objek yang disebutkan dalam Kode 5.35.

```
public static JSONObject readJsonFromUrl(String url)
throws IOException, JSONException {
 InputStream is = new URL(url).openStream();
 try {
     BufferedReader rd = new BufferedReader(
           new InputStreamReader(is,
Charset.forName("UTF-8")));
     String jsonText = readAll(rd);
     JSONObject json = new JSONObject(jsonText);
     return json;
 } finally {
     is.close();
 }
}
```

Kode 5.36. Membaca json google translate dari url

Kode 5.36 menunjukkan class untuk membaca hasil json yang sudah dibuat dari node js agar dapat dibaca oleh program java. Input dari program tersebut berupa url yang memuat hasil json google translate sebelumnya. Fungsi *BufferedReader* digunakan untuk membaca konten dari halaman url tersebut dan mengembalikannya dalam format json pada program java.

```
public TranslatedObject getTranslateWord(String
initWord, String from, String to) throws IOException {
 String url = getUrl()+initWord+"&f="+from+"&t="+to;
 JSONObject jsonObject = readJsonFromUrl(url);
 String translatedText = (String)
jsonObject.get("translatedText");
 String correctedText = (String)
jsonObject.get("correctedText");
 String initialText = (String)
jsonObject.get("initialText");
 boolean isTranslated = (Boolean)
jsonObject.get("autoCorrected");
 boolean autoCorrected = (Boolean)
jsonObject.get("isTranslated");
 boolean didYouMean = (Boolean)
jsonObject.get("didYouMean");
 TranslatedObject myObj = new TranslatedObject(
  translatedText,initialText,correctedText,
  isTranslated,autoCorrected,didYouMean);
 return myObj;
}
```

Kode 5.37. Mengambil elemen *json object*

Kode 5.37 menunjukkan class untuk mengambil nilai dari setiap elemen *json object* yang telah dibaca oleh program. Masing - masing nilai elemen disimpan dalam 6 variabel dengan tipe data yang telah ditentukan sesuai dengan kode diatas. Lalu sistem memanggil membuat sebuah *object* berdasarkan *class* TranslatedObject yang menyimpan nilai elemen dari objek json.

```
public static String getTranslation(String wordAfiks)
throws IOException {
 GoogleTranslate myTranslate = new GoogleTranslate();
 TranslatedObject myObj = myTranslate
   .getTranslateWord(wordAfiks,"id", "en");
 String resText = null;
 String correctedText = null;
 if (myObj.isTranslated()) {
    String translate = myObj.getTranslatedText();
    if(!translate.equalsIgnoreCase(wordAfiks)){
          if (myObj.isDidYouMean()||
             myObj.isAutoCorrected()) {
                correctedText =
myObj.getCorrectedText();
                resText = correctedText;
          } else {
                TranslatedObject myObj2 = myTranslate
                   .getTranslateWord(translate,"en",
"id");
                String translate2 =
myObj2.getTranslatedText()
                   .replaceAll("null", "");
```

```
        if(wordAfiks.equalsIgnoreCase(translate2)) {
                resText = wordAfiks;
              }
          }
          return resText;
   }
   else{
    System.out.println("Hasil Translasi sama!");
    return null;
   }
  } else {
     System.out.println("Gagal di translate!");
     return null;
  }
}
```

Kode 5.38 Mendapatkan translasi dari kata berimbuhan

Kode 5.38 menunjukkan potongan kode program untuk mendapatkan terjemahan kata dari sebuah kata yang telah diberi imbuhan. Sistem akan membuat objek berupa TranslatedObject apabila telah terdapat terjemahan katanya. Objek yang dibuat akan diperiksa apakah kata tersebut berhasil di terjemahkan kedalam bahasa inggris. Jika iya, maka akan diperiksa kembali apakah terdapat pengoreksian kata dari google, jika iya maka kata yang idkembalikan adalah kata pengoreksinya tadi, dan jika tidak akan mengembalikan nilai masukkanya.

```
public static LinkedList<String> sufiksMethod(String
rootword) throws IOException, ParseException {
 String sufiksTxt =
"F:\\Stezar\\Code\\TASte\\resources\\sufiks.txt";
 BufferedReader in = new BufferedReader(new
FileReader(sufiksTxt));
 LinkedList<String> hasil = new LinkedList<String>();
 String wordAfiks;
 String line;
 while((line=in.readLine())!= null){
   wordAfiks = rootword+line;//beda
   String trans = getTranslation(wordAfiks);
   if(trans!=null){
     hasil.add(trans);
   }
 }
 in.close();
 return hasil;
}
```

Kode 5.39 *Generate afiks*

Kode 5.39 menunjukkan cara untuk *generate* imbuhan pada kata dasar sehingga menjadi kata yang berimbuhan. Pada kode tersebut adalah contoh dalam membuat sufiks/akhiran. Perbedaannya hanyalah pada penambahan string, jika awalan maka string kata dasar ditambahkan diawal, jika akhiran ditambahkan diakhir, dan jika dia konfiks maka akan di pisah terlebih dahulu dan digabungkan keduanya pada kata dasar.

```
  Class.forName("com.mysql.jdbc.Driver");
  String url = "jdbc:mysql://localhost:3306/kamusindo";
  String username = "root";
  String password = "";
  Connection db = DriverManager.getConnection(url,
username, password);
  PreparedStatement st = db.prepareStatement(
     "INSERT INTO enrich (kata)"
     + " VALUES (?)");
  int insertedRows = 0;
```

Kode 5.40 Koneksi *database* untuk pengayaan

Kode 5.40 koneksi untuk *database* yang akan dimasukkan kedalam tabel *enrich* kata yang hasilnya berasal dari nilai kembalian method pada Kode 5.39 dan dijalankan seperti pada Kode 5.41.

```
String file =
"F:\\Stezar\\Code\\TASte\\resources\\kata.txt";
BufferedReader in = new BufferedReader(new
FileReader(file));
LinkedList<String> myHasil = new LinkedList<String>();
String line;
while((line=in.readLine())!= null){
  myHasil.addAll(prefiksMethod(line));
  myHasil.addAll(sufiksMethod(line));
  myHasil.addAll(konfiksMethod(line));
}
in.close();
Iterator<String> iter = myHasil.iterator();
while(iter.hasNext()){
  st.setString(1, iter.next());
  try {
    insertedRows += st.executeUpdate();
        System.out.println(insertedRows);
  } catch (SQLException e) {
        continue;
  }
}
```

Kode 5.41 Menjalankan translasi

Kode 5.41 akan membaca kata dasar dari file kemudian dimasukkan satu per satu kedalam method penambahan imbuhan dan diperiksa ke kamus. Hasilnya akan disimpan dalam Linkedlist dan akan dimasukkan kedalam database.

5.4.2 Pembuatan Solr Index

Dalam rangka mempercepat proses pencarian, maka data yang telah didapatkan sebelumnya akan digabungkan dan dilakukan *indexing* di dalam Solr.

5.4.2.1 Solr Kamusterbaru

Pertama yang dilakukan adalah membuat *core* Solr terlebih dahulu menggunakan perintah seperti pada Kode 5.42.

```
> .\solr create -c kamusterbaru
```

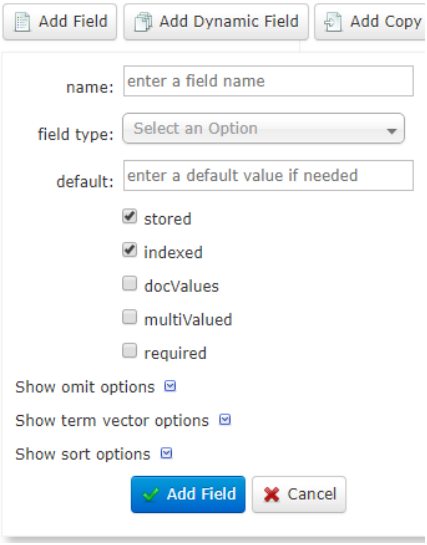
Kode 5.42 Membuat core kamusterbaru

Perintah tersebut akan membuat *core* baru yang bernama kamusterbaru untuk menyimpan semua gabungan leksikon bahasa Indonesia yang telah dikumpulkan pada tahap sebelunmnya.

Setelah *core* terbentuk maka mulai untuk membuat *field* pada *schema core,* disini *field*-nya terdiri dari:

- Kata_key: String (tidak di tokenisasi)
- Kata: Text (ditokenisasi)
- Lex: String
- Src: String
- Arti: Text

Untuk membuat field masuk ke menu *schema*, kemudian klik tombol *add field*, dan nantinya akan terdapat *form* untuk mengisikan *field* seperti Kode 5.43



Kode 5.43 Menambahkan *field* pada *schema*

Kemudian, untuk mengambil data dari *database* maka diperlukan *handler data import* dengan menambahkan file konfigurasi dengan nama data-config.xml. Isi file tersebut adalah seperti Kode 5.44.

```
<dataConfig>
 <dataSource                          type="JdbcDataSource"
driver="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost/kamusindo"       user="root"
password=""/>
 <document>
     <entity      name="kamusterbaru"      query="SELECT
kata,lex,type,src,arti FROM kamusterbaru">
         <field column="kata" name="kata" />
                <field column="kata" name="kata_key" />
         <field column="lex" name="lex" />
                <field column="type" name="type" />
         <field column="src" name="src" />
                <field column="arti" name="arti" />
        </entity>
```

```
  </document>
 </dataConfig>
```

Kode 5.44 Data-config.xml kamusterbaru

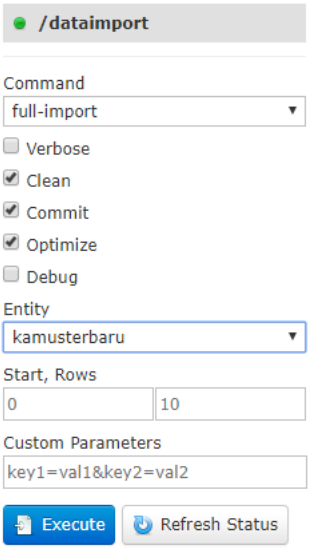
Kode 5.44 berguna memetakan kolom yang ada di tabel kedalam field yang telah dibuat di dalam core kamusterbaru.

Sebelum melakukan import data dari MySQL perlu menambahkan potongan requestHandler sesuai Kode 5.45 ke dalam file solr-config.xml.

```
<lib dir="${solr.install.dir:../../../..}/dist/"
regex="solr-dataimporthandler-.*\.jar" />
  <lib dir="${solr.install.dir:../../../..}/dist/"
regex="mysql-connector-java-.*\.jar" />

       <requestHandler name="/dataimport"
class="org.apache.solr.handler.dataimport.DataImportHa
ndler">
     <lst name="defaults">
          <str name="config">data-config.xml</str>
     </lst>
       </requestHandler>
```

Kode 5.45 *DataImport Handler*

Untuk melakukan proses import data, masuk ke menu Data Import, kemudian tekan tombol *execute* untuk memulai seperti Kode 5.46.



Kode 5.46 Import data dari MySQL

Kemudian menunggu hingga proses import berhasil dilakukan.

5.4.2.2 Solr Kamusmapping

Pertama yang dilakukan adalah membuat *core* Solr terlebih dahulu menggunakan perintah seperti pada Kode 5.47.

```
> .\solr create -c kamusmapping
```

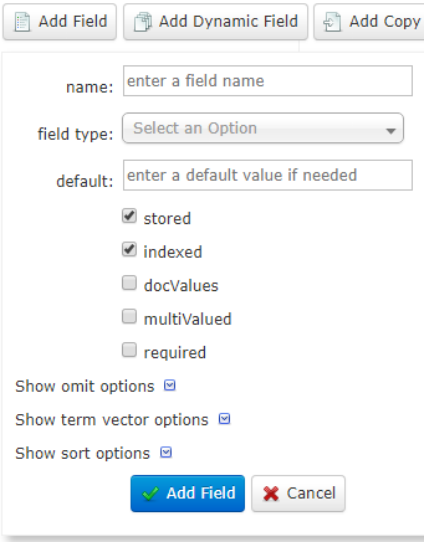
Kode 5.47 Membuat core kamusmaping

Perintah tersebut akan membuat *core* baru yang bernama kamusmapping untuk menyimpan data yang telah di indeks dari tabel kamusmapping dalam *database*.

Setelah *core* terbentuk maka mulai untuk membuat *field* pada *schema* core, disini *field*-nya terdiri dari:

- Kata: String (tidak di tokenisasi)
- Correction: String
- Id: unique integer
- Status: integer

Untuk membuat *field* masuk ke *menu schema*, kemudian klik tombol *add field*, dan nantinya akan terdapat *form* untuk mengisikan *field* seperti Kode 5.48.



Kode 5.48 Membuat field kamusmaping

Kemudian, untuk mengambil data dari *database* maka diperlukan *handler data import* dengan menambahkan file konfigurasi dengan nama data-config.xml. Isi file tersebut adalah seperti Kode 5.49.

```
<dataConfig>
 <dataSource                          type="JdbcDataSource"
driver="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost/kamusindo"       user="root"
password=""/>
 <document>
     <entity         name="mapping"         query="SELECT
id,kata,correction,status FROM mapping">
         <field column="id" name="id" />
                <field column="kata" name="kata" />
         <field column="correction" name="correction" />
                <field column="status" name="status" />
```

```
        </entity>
  </document>
</dataConfig>
```

Kode 5.49 Data-config.xml kamusmaping

Kode tersebut berguna memetakan kolom yang ada di tabel kedalam *field* yang telah dibuat di dalam core kamusterbaru. Tahapan berikutnya sama seperti proses pembuatan kamusterbaru yaitu menambahkan requestHandler dan mulai melakukan memasukkan serta mengindeks data MySQL.

5.5 Pembuatan Model *Word2Vec*

5.5.1 Pembagian Dataset

Yang pertama dilakukan ialah membagi dataset yang sudah ada menjadi dua yaitu training dan sampel pengujian. Data sampel pengujian yang diambil sebanyak seribu secara random dengan komposisi 500 data facebook dan 500 data twitter. Kemudian, setiap baris data tersebut dilakukan pengoreksian data per kalimat. Tahapan berikutnya dilakukan tokenisasi hingga terbentuk kumpulan token. Untuk melakukan pemisahan per token menggunakan kode program seperti pada Kode 5.50.

```
TokenizerFactory t = new DefaultTokenizerFactory();
t.setTokenPreProcessor(new
LowCasePreProcessor());//agar tidak menghapus angka

List<String> tokens =
t.create(line[1].toLowerCase()).getTokens();
tokens.removeAll(Arrays.asList("", null)); //remove
null element atau kosong

List<String> tokensLabeled =
t.create(line[1].toLowerCase()).getTokens();
tokensLabeled.removeAll(Arrays.asList("", null));
for (int i = 0; i < tokens.size(); i++) {
    String[] in = new String[]{line[0] + "",
tokens.get(i), ""};
    if (rw.writeCSV(in, outputTokenLabeled)) {
        System.out.println("Alhamdulillah :)");
    } else {
        System.out.println("Yah, gagal :(");
    }
}
if (tokens.size() == tokensLabeled.size()) {
    for (int i = 0; i < tokens.size(); i++) {
        String[] in = new String[]{posisi+"",
tokens.get(i), tokensLabeled.get(i)};
        if (rw.writeCSV(in, outputTokenLabeled)) {
            System.out.println("Alhamdulillah :)");
        } else {
            System.out.println("Yah, gagal :(");
        }
    }
} else {
    if (rw.writeCSV(new String[]{posisi+"", line[0],
line[1]}, cekToken)) {
        System.out.println("Alhamdulillah, berhasil
nambah periksaan :)");
    } else {
        System.out.println("Yah, gagal :(");
    }
}
posisi++;
```

Kode 5.50 Pembagian dataset

Kode diatas berjalan setelah melakukan pembacaan data testing yang telah diberi koreksi secara manual. Kemudian setiap baris yang dibaca akan dilakukan tokenisasi berbeda antar data asli dan data yang telah dikoreksi hingga terbentuk token. Hasil dari tokenisasi tersebut ditulis satu persatu kedalam file csv.

5.5.2 Menghitung Kemunculan Kata di Kamus dan Non-Kamus

Untuk dapat melakukan uji coba akurasi setiap model yang akan dibuat maka diperlukan sebuah data uji. Data uji yang dakan

digunakan berasal dari kata tidak baku yang paling sering muncul. Untuk menyimpan setiap kata dan kemunculannya maka diperlukan sebuah penyimpanan sementara. Hashmap merupakan salah satu bentuk penyimpanan yang tepat dikarenakn memiliki Key dan Value. Nantinya kata yang bersifat unik akan dijadikan Key dan frekuensinya akan dijadikan Value. Hashmap yang dibuat ada 2 untuk menyimpan kata sesuai kamu dan non-kamus.

```
String word = removeSymbol(oneLine(rs.getString(1)));
TokenizerFactory t = new DefaultTokenizerFactory();
t.setTokenPreProcessor(new CommonPreprocessor());
List<String> arrayWord =
t.create(word.toLowerCase()).getTokens();

for (String token:arrayWord) {
    if (myKamus.isSolrDict(URLEncoder.encode(token,
"UTF-8"))) {
        if (map1.containsKey(token)) {
            map1.put(token, map1.get(token) + 1);
        } else {
            map1.put(token, 1);
        }
    }
    else {
         if (map2.containsKey(token)) {
             map2.put(token, map2.get(token) + 1);
         } else {
             map2.put(token, 1);
         }
    }
}
i++;
```

Kode 5.51 Hitung frekuensi kata di kamus dan non-kamus

Kode 5.51 merupakan bentuk implementasi dari pemisahan kata ini. Awalnya data yang ada di ambil dari database dan di proses satu per satu menggunakan potongan kode tersebut. Setelah data di baca per baris maka akan dilakukan pembuangan simbol dan menjadikan satu baris. Kemudian hasilnya akan dipisah per token dan setiap token yang terbentuk akan diperiksa ke kamus. Apabila terdapat kata dikamus maka akan masuk ke object map1 dan apabila tidak dapat di kamus maka akan masuk ke object map2. Masing-masing dari object map akan dikeluarkan elemennya dan ditulis satu per satu kedalam file csv.

5.5.3 Training Word2Vec

Tahapan selanjutnya ialah proses *training* dataset untuk menghasilkan model *word2vec*. Training akan dilakukan dalam beberapa kali percobaan dengan menggunakan parameter yang berbeda untuk mencari hasil yang terbaik.

5.5.3.1 Percobaan *Training* 1

```
Word2Vec vec = new Word2Vec.Builder()
        .minWordFrequency(5)
        .iterations(1)
        .layerSize(100)
        .seed(42)
        .windowSize(5)
        .iterate(iter)
        .tokenizerFactory(t)
        .build(); //model 1
vec.fit();
WordVectorSerializer.writeWord2VecModel(vec,
model_v1.csv);
```

Kode 5.52 Parameter training 1

Kode 5.52 menunjukkan proses *training* yang pertama dengan menggunakan konfigurasi sebagai berikut:

Tabel 5.4 Percobaan 1

Parameter	Isian
Learning Algortihm	Skip-Gram
Training Algorithm	Negative Sample
Iterasi	1
Minimum Word Frequency	5
Context Window	5
Layer Size	100

Setelah melakukan konfigurasi seperti Tabel 5.4, maka sistem akan memanggil *method* fit() untuk memulai proses *training*. Setelah proses *training* dilakukan maka sistem akan menuliskan model tersebut kedalam file bernama “model_v1.csv”

5.5.3.2 Percobaan *Training* 2

```
Word2Vec vec = new Word2Vec.Builder()
    .elementsLearningAlgorithm(new CBOW<VocabWord>())
    .minWordFrequency(5)
     .iterations(1)
     .layerSize(100)
```

```
      .seed(42)
      .windowSize(5)
      .iterate(iter)
      .tokenizerFactory(t)
      .build(); //model 2
WordVectorSerializer.writeWord2VecModel(vec,
model_v2.csv);
```

Kode 5.53 Parameter training 2

Kode 5.53 menunjukkan proses *training* yang kedua dengan menggunakan konfigurasi sebagai berikut:

Tabel 5.5 Percobaan 2

Parameter	Isian
Learning Algortihm	CBOW
Iterasi	1
Minimum Word Frequency	5
Context Window	5
Layer Size	100

Setelah melakukan konfigurasi seperti Tabel 5.5, maka sistem akan memanggil method fit() untuk memulai proses *training*. Setelah proses *training* dilakukan maka sistem akan menuliskan model tersebut kedalam file bernama "model_v2.csv"

5.5.3.3 Percobaan *Training* 3

```
Word2Vec vec = new Word2Vec.Builder()
        .epochs(10)
        .negativeSample(10)
        .minWordFrequency(10)
        .iterations(1)
        .layerSize(100)
        .seed(42)
        .windowSize(10)
        .iterate(iter)
        .tokenizerFactory(t)
        .build(); //model 3
WordVectorSerializer.writeWord2VecModel(vec,
model_v3.csv);
```

Kode 5.54 Parameter Training 3

Kode 5.54 menunjukkan proses *training* yang ketiga dengan menggunakan konfigurasi sebagai berikut:

Tabel 5.6 Percobaan 3

Parameter	Isian
Learning Algortihm	Skip-Gram
Training Algorithm	Negative Sample (10)
Epoch	1
Iterasi	10
Minimum Word Frequency	5
Context Window	5
Layer Size	100

Setelah melakukan konfigurasi seperti Tabel 5.6, maka sistem akan memanggil *method* fit() untuk memulai proses *training*. Setelah proses *training* dilakukan maka sistem akan menuliskan model tersebut kedalam file bernama "model_v3.csv"

5.5.3.4 Percobaan *Training* 4

```
Word2Vec vec = new Word2Vec.Builder()
      .elementsLearningAlgorithm(new CBOW<VocabWord>())
      .epochs(10)
      .minWordFrequency(10)
      .iterations(1)
      .layerSize(100)
      .seed(42)
      .windowSize(5)
      .iterate(iter)
      .tokenizerFactory(t)
      .build(); //model 4
WordVectorSerializer.writeWord2VecModel(vec,
model_v4.csv);
```

Kode 5.55 Parameter training 4

Kode 5.55 menunjukkan proses *training* yang keempat dengan menggunakan konfigurasi sebagai berikut:

Tabel 5.7 Percobaan 4

Parameter	Isian
Learning Algortihm	CBOW
Training Algorithm	*Hierarchical Softmax*
Epoch	10

Iterasi	1
Minimum Word Frequency	10
Context Window	5
Layer Size	100

Setelah melakukan konfigurasi seperti Tabel 5.7, maka sistem akan memanggil *method* fit() untuk memulai proses *training*. Setelah proses *training* dilakukan maka sistem akan menuliskan model tersebut kedalam file bernama "model_v4.csv"

5.5.3.5 Percobaan *Training* 5

```
Word2Vec vec = new Word2Vec.Builder()
        .negativeSample(10)
        .minWordFrequency(10)
        .iterations(1)
        .layerSize(100)
        .seed(42)
        .windowSize(5)
        .iterate(iter)
        .tokenizerFactory(t)
        .build(); //model 5
WordVectorSerializer.writeWord2VecModel(vec,
model_v5.csv);
```

Kode 5.56 Parameter training 5

Kode 5.56 menunjukkan proses *training* yang kelima dengan menggunakan konfigurasi sebagai berikut:

Tabel 5.8 Percobaan 5

Parameter	Isian
Learning Algortihm	Skip-Gram
Training Algorithm	Negative Sample (10)
Epoch	1
Iterasi	5
Minimum Word Frequency	5
Context Window	100

Setelah melakukan konfigurasi seperti Tabel 5.8, maka sistem akan memanggil *method* fit() untuk memulai proses *training*. Setelah proses *training* dilakukan maka sistem akan menuliskan model tersebut kedalam file bernama "model_v5.csv"

5.5.3.6 Percobaan *Training* 6

```
Word2Vec vec = new Word2Vec.Builder()
      .elementsLearningAlgorithm(new CBOW<VocabWord>())
      .useHierarchicSoftmax(true)
      .minWordFrequency(10)
      .iterations(1)
      .layerSize(100)
      .seed(42)
      .windowSize(5)
      .iterate(iter)
      .tokenizerFactory(t)
      .build(); //model 6
WordVectorSerializer.writeWord2VecModel(vec,
model_v6.csv);
```

Kode 5.57 Parameter training 6

Kode 5.57 menunjukkan proses *training* yang keenam dengan menggunakan konfigurasi sebagai berikut:

Tabel 5.9 Percobaan 6

Parameter	Isian
Learning Algortihm	CBOW
Training Algorithm	Hierarchical Softmax
Epoch	1
Iterasi	5
Minimum Word Frequency	5
Context Window	100

Setelah melakukan konfigurasi seperti Tabel 5.9, maka sistem akan memanggil *method* fit() untuk memulai proses *training*. Setelah proses *training* dilakukan maka sistem akan menuliskan model tersebut kedalam file bernama “model_v6.csv”

5.5.3.7 Percobaan *Training* 7

```
Word2Vec vec = new Word2Vec.Builder()
        .epochs(10)
        .minWordFrequency(5)
        .iterations(3)
        .layerSize(100)
        .seed(42)
        .windowSize(10)
        .iterate(iter)
        .tokenizerFactory(t)
        .build(); //model 7
WordVectorSerializer.writeWord2VecModel(vec,
model_v7.csv);
```

Kode 5.58 Parameter training 7

Kode 5.58 menunjukkan proses *training* yang ketujuh dengan menggunakan konfigurasi sebagai berikut:

Tabel 5.10 Percobaan 7

Parameter	Isian
Learning Algortihm	Skip-Gram
Training Algorithm	Negative Sample
Epoch	10
Iterasi	3
Minimum Word Frequency	5
Context Window	10
Layer Size	100

Setelah melakukan konfigurasi seperti Tabel 5.10, maka sistem akan memanggil *method* fit() untuk memulai proses *training*. Setelah proses *training* dilakukan maka sistem akan menuliskan model tersebut kedalam file bernama "model_v7.csv"

5.5.3.8 Percobaan *Training* 8

```
Word2Vec vec = new Word2Vec.Builder()
      .elementsLearningAlgorithm(new CBOW<VocabWord>())
      .useHierarchicSoftmax(true)
      .layerSize(500)
      .seed(42)
      .windowSize(5)
      .minWordFrequency(5)
      .iterations(2)
      .epochs(10)
      .iterate(iter)
      .tokenizerFactory(t)
      .lookupTable(table)
      .vocabCache(cache)
      .workers(10)
      .build(); //model 8
WordVectorSerializer.writeWord2VecModel(vec,
model_v8.csv);
```

Kode 5.59 Parameter training 8

Kode 5.59 menunjukkan proses *training* yang kedelapan dengan menggunakan konfigurasi sebagai berikut:

Tabel 5.11 Percobaan 8

Parameter	Isian
Learning Algortihm	CBOW
Training Algorithm	Hierarchical Softmax
Epoch	10
Iterasi	2
Minimum Word Frequency	5
Context Window	5
Layer Size	500

Setelah melakukan konfigurasi seperti Tabel 5.11, maka sistem akan memanggil *method* fit() untuk memulai proses *training*. Setelah proses *training* dilakukan maka sistem akan menuliskan model tersebut kedalam file bernama "model_v8.csv"

5.5.3.9 Evaluasi Model

Untuk mengevaluasi model yang telah dibuat, maka akan menggunakan data uji berupa seratus kata non-kamus yang paling sering muncul.

```
Word2Vec word2Vec =
WordVectorSerializer.readWord2VecModel(namafile);
```

Kode 5.60 Membaca model Word2Vec

Sebelum melakukan proses prediksi maka model dibaca terlebih dahulu. Kode 5.60 menunjukkan proses dalam membaca model dari sebuah file dengan method readWord2VecModel.

```
String[] word = new String[2];
 word[0] = line2;
 String kataKamus="";
 Collection<String> hasil =
word2Vec.wordsNearest(line2, 500);
 int i = 0;
 for (String token : hasil) {
     if(i == 10){
         break;
     }
     if (myKamus.isSolrDict(URLEncoder.encode(token,
"UTF-8"))) { //cocokkan dengan kamus
         kataKamus += token + ",";
         i++;
     }
}
word[1] = kataKamus;
log.info("Closest words to "+line2+" on 1nd run: " +
hasil);
 if(writeCSV(word, file)){
     System.out.println("Alhmadulillah, berhasil :)");
 }
 else {
     System.out.println("Yah, gagal :(");
 }
```

Kode 5.61 Mencari 10 kandidat sesuai kamus

Setelah model dibaca maka kemudian file yang berisikan sertaus kata giliran dibaca. File tersebut dibaca per baris yang berisikan kata. Kode 5.61 menunjukkan token yang akan diprediksi kandidat kata terdekatnya bernama “line2”. Kemudian kata akan diprediksi menggunakan method wordNearest dan akan menghasilkan 500 kandidat kata. Hasil yang telah muncul akan diperiksa satu persatu menggunakan perulangan kedalam kamus. Jika kata yang sesuai di kamus sudah mencapai sepuluh token, maka perulangan akan berhenti dan sistem menuliskan kedalam file csv.

5.6 Pembuatan Sistem Normalisasi Teks

5.6.1 Sistem Normalisasi Teks

```
public void normalize(String filePath, String
pathOutputTokens, double threshold) throws IOException,
SolrServerException {
  Dictionary myKamus = new Dictionary();
  Word2vecPredict myWord2vec = new Word2vecPredict();
myWord2vec.setModel("model_v8.txt");
  ReadWrite rw = new ReadWrite();
  CSVReader reader = rw.readCSV(filePath);
  String[] line = null;
  try {
   while ((line = reader.readNext()) != null) {
  boolean resultDict = myKamus.isSolrDict(line[1]);
  String[] myWord;
  if (line[1].length() == 1 || resultDict) {
   //jika dia ada dikamus atau dia terdiri dari satu
huruf
   myWord = new String[]{line[0], line[1], line[2],
line[1], "1", "ada di kamus"};
  } else {
   //jika tidak ada, maka
   //cek prediksi kata pake word2vec
   //cek prediksi yang sesuai dengan kamus
   //bobotkan hasil prediksi yang sudah sesuai kamus
   String mapping = myKamus.isMapped(line[1], 1);
   if (mapping != null) {
    myWord = myWord = new String[]{line[0], line[1],
line[2], mapping, "1", "termaping"};
   } else {
    Collection<String> resWord2vec =
myWord2vec.getResult(line[1], 500);
    log.info("" + resWord2vec);
    LinkedList<SimilarityObject> myDistance =
myWord2vec.getSimilarity(10, line[1], resWord2vec,
threshold);
    System.out.println(myDistance.size());
    if (resWord2vec.isEmpty() || myDistance.isEmpty())
{
     if(line[2].equalsIgnoreCase(line[1])){
   myWord = new String[]{line[0], line[1], line[2],
line[1], "0", "tidak ada di kamus"};
     }else{
   myWord = new String[]{line[0], line[1], line[2],
line[1], "0", "gagal total"};
     }
    } else {
     System.out.println("Yang diperiksa kata : " +
line[1]);
     Collections.sort(myDistance); //sort tertinggi
     String max_word = myDistance.getFirst().getWord();
     double max_weight =
myDistance.getFirst().getWeight();
     if (line[2].equalsIgnoreCase(max_word)) {
    //jika hasil prediksi sama
    myWord = new String[]{line[0], line[1], line[2],
max_word, "1", "diprediksi sama"};
```

```
    myKamus.addMapping(line[1], max_word);
//menambahkan ke kamusmapping
     } else if (treatmentContainsLemma(line[1],
max_word)) {
    //jika kata dasarnya sama
    myWord = new String[]{line[0], line[1], line[2],
line[1], "1", "kata dasar sama"};
     } else if (line[2].split("-
")[0].equalsIgnoreCase(max_word) && line[2].split("-
")[1].equalsIgnoreCase(max_word)) {
    //untuk kata yang berulang
    myWord = new String[]{line[0], line[1], line[2],
max_word, "1", "diprediksi kata ulang"};
    myKamus.addMapping(line[1], max_word);
//menambahkan ke kamusmapping
     } else {
    //apapun hasilnya dikeluarkan
    myWord = new String[]{line[0], line[1], line[2],
max_word, "0", "tidak tepat"};
     }
    }
   }
  }
  if (rw.writeCSV(myWord, pathOutputTokens)) {
   System.out.println("Alhamdulillah, berhasil nambah
:)");
  } else {
   System.out.println("Yah, gagal :(");
  }
   }
   reader.close();
  } catch (IOException e1) {
   e1.printStackTrace();
  }
}
```

Kode 5.62 Sistem utama Normalisasi Teks

Kode 5.62 menunujukkan bahwa pertama kali yang harus dilakukan ialah membaca model word2vec terbaik. Dalam hal ini model yang dipakai adalah model_v8.txt. Kemudian sistem membaca file *input* yang berisi daftar token-token. Token tersebut diproses satu per satu untuk dicari pembenaran dari kata tersebut. Pertama akan diperiksa kedalam kamus, apabila telah ada maka hasilnya ataupun terdiri dari satu karakter adalah kata tersebut, dan jika tidak, lanjut memeriksa kedalam kamus mapping. Jika di dalam kamus *mapping* terdapat kata tersebut, maka hasilnya adalah kata yang telah dikoreksi dalam kamus maping. Apabila tidak termasuk, maka sistem akan memulai mencari 500 kandidat kata pada word2vec. Dari 500 kandidat tersebut yang diambil hanyalah sepuluh yang sesuai dengan kamus dan lebih dari sama dengan *threshold*. Jika dari proses

tidak ada hasilnya maka sistem akan mengembalikan kembali kata tersebut. Apabila ada maka memulai untuk mengambil nilai yang paling besar. Kemudian dari hasil nilai yang terbesar itu mulai mencocokkan dengan koreksi manualnya, apabila sama maka statusnya "prediksi sama", jika mengandung lemma maka "kata dasar sama", jika berulang maka "diprediksi kata ulang", dan jika tidak sama persis maka "tidak tepat". Kemudian hasil dari kesemuanya ditulis kedalam file csv.

5.6.1.1 Sistem Pembobotan

Pada awal membuat sistem ini digunakan algoritma levenshtein distance sebagai sistem bobot tunggal. Namun pada saat dicoba kebeberapa data secara random, hasil yang diperoleh justru tidak akurat, sehingga dibutuhkan parameter lain dalam pembobotan. Akhirnya ditentukanlah 3 parameter yaitu, posisi ditemukannya kandidat prediksi *word2vec*, skor *levenshtein distance*, serta *jaro-winkler distance*.

```
public double getWeight(double w2c, double lev, double
jaro){
    double bobot = (w2c*0.5)+(lev*0.25)+(jaro*0.25);
    return bobot;
}
```

Kode 5.63 Mekanisme Pembobotan

Kode 5.63 menunjukkan bahwa bobot yang diberikan terhadap 3 parameter ini berbeda, yaitu:

- Posisi Word2Vec: 0.5

  Hal ini dikarenakan semakin awal ditemukannya pada kumpulan kandidat kata *word2vec*, maka semakin memiliki kesamaan yang besar dengan kata yang diprediksi. Sehingga bobot yang diberikan juga besar mengingat ini hal utama dalam penelitian ini.

- Skor *Levenshtein Distance* & *Jaro-Winkler*: 0.25
  Bobot yang diberikan untuk algoritma *Levenshtein* ini dibuat sama dengan *Jaro-Winkler* karena kedua algoritma ini mempunyai keunggulan masing-masing. Keunggulan dari *Levenshtein* adalah mendukung transposisi dan substitusi dari sebuah karakter.

Sedangkan *Jaro-Winkler* memiliki keunggulan sangat cocok untuk perbandingan kata yang pendek, dikarenakan *Winkler* menyempurnakan algoritma ini dengan memberikan bobot yang lebih besar pada prefix kedua kata apabila sama.

5.6.2 Pengujian

5.6.2.1 Pengujian dengan Seribu Kata paling sering muncul non-kamus

```
System.out.println(line[1]);
String[] newWord = new String[7];
newWord[0] = line[0];
String[] listCandidate = line[1].split(",");
for (int i = 0 ; i < listCandidate.length-1; i++){
  newWord[1] = listCandidate[i];
  newWord[2] = i+"";
  //levenshtein
  double lev =  n.similarity(newWord[0], newWord[1]);
  //jaro winkler
  double jarko = jw.similarity(newWord[0], newWord[1]);
  //posisi word2vec
  double          w2c          =          1.0          -
((double)i/(double)(listCandidate.length-1));
  double bobot = (w2c*0.5)+(lev*0.25)+(jarko*0.25);
  newWord[3] = w2c+"";
  newWord[4] = lev+"";
  newWord[5] = jarko+"";
  newWord[6] = bobot+"";
  if(writeCSV(newWord, out)){
   System.out.println("Alhamdulillah, berhasil :)");
  }
  else {
   System.out.println("Yah, gagal :(");
  }
}
```

Kode 5.64 Implementasi pengujian seribu kata

Kode 5.64 menunjukkan proses dalam melakukan pengujian seribu kata yang paling sering muncul tetapi non-kamus. Pertama file uji akan dibaca per baris yang kemudian setiap hasil 10 kandidat dipisahkan menjadi array dengan split(). Setiap elemen *array* akan dihitung menggunakan bobot yang sudah dijelaskan pada sub bab sebelumnya. Untuk mencari skor *levenshtein* dan *jarko-winkler*, kandidat akan dibandingkan dengan kata non-kamus tadi. Kemudian bobot yang sudah dihitung akan disimpan ke variabel. Setelah semua variabel

terisi kedalam array newWord, maka *array* tersebut dituliskan kedalam file csv.

5.6.2.2 Pengujian Data Sampel Pengujian

Pengujian dilakukan pada data sampel dengan memperhatikan batasan/*threshold*. *Threshold* yang dimaksud adalah batas dari sebuah bobot perbandingan dua string antar kata asal dengan hasil prediksi *word2vec*. Jika memenuhi *threshold* maka kata itu yang akan diambil dan dijadikan 10 kandidat utama yang akan disorting kembali. Pengujian yang dilakukan menggunakan sistem normalisasi pada sub bab 5.6.1.

5.6.2.2.1 Threshold 65%

```
NormalizeText myNorm = new NormalizeText();
myNorm.normalize("resources\\token_baru.csv",
"resources\\output_baru_per_token_65.csv", 0.65);
```

Kode 5.65 Implementasi pengujian *threshold* 65%

Kode 5.65 menampilkan potongan kode program yang akan menjalankan pengujian sistem dengan menggunakan *threshold* bobot lebih dari atau sama dengan 65%. Mulanya program memanggil kelas NormalizeText dengan inisiasi objek bernama myNorm. Kemudian memanggil *method* normalize(). Setelah method ini menerima 3 parameter yaitu file testing, hasil testing, dan *threshold*. Nantinya file akan disimpan kedalam file output_baru_per_token_65.csv.

5.6.2.2.2 Threshold 70%

```
NormalizeText myNorm = new NormalizeText();
myNorm.normalize("resources\\token_baru.csv",
"resources\\output_baru_per_token_70.csv", 0.70);
```

Kode 5.66 Implementasi pengujian *threshold* 70%

Kode 5.66 menampilkan potongan kode program yang akan menjalankan pengujian sistem dengan menggunakan *threshold* bobot lebih dari atau sama dengan 70%. Mulanya program memanggil kelas NormalizeText dengan inisiasi objek bernama myNorm. Kemudian memanggil *method* normalize(). Setelah method ini menerima 3 parameter yaitu file testing, hasil testing, dan *threshold*. Nantinya file akan disimpan kedalam file output_baru_per_token_70.csv.

5.6.2.2.3 Threshold 75%

```
NormalizeText myNorm = new NormalizeText();
myNorm.normalize("resources\\token_baru.csv",
"resources\\output_baru_per_token_75.csv", 0.75);
```

Kode 5.67 Implementasi pengujian *threshold* 75%

Kode 5.67 menampilkan potongan kode program yang akan menjalankan pengujian sistem dengan menggunakan *threshold* bobot lebih dari atau sama dengan 75%. Mulanya program memanggil kelas NormalizeText dengan inisiasi objek bernama myNorm. Kemudian memanggil *method* normalize(). Setelah method ini menerima 3 parameter yaitu file testing, hasil testing, dan *threshold*. Nantinya file akan disimpan kedalam file output_baru_per_token_75.csv.

5.6.2.2.4 Threshold 80%

```
NormalizeText myNorm = new NormalizeText();
myNorm.normalize("resources\\token_baru.csv",
"resources\\output_baru_per_token_80.csv", 0.80);
```

Kode 5.68 Implementasi pengujian *threshold* 80%

Kode 5.68 menampilkan potongan kode program yang akan menjalankan pengujian sistem dengan menggunakan *threshold* bobot lebih dari atau sama dengan 80%. Mulanya program memanggil kelas NormalizeText dengan inisiasi objek bernama myNorm. Kemudian memanggil *method* normalize(). Setelah method ini menerima 3 parameter yaitu file testing, hasil testing, dan *threshold*. Nantinya file akan disimpan kedalam file output_baru_per_token_80.csv.

5.6.2.2.5 Threshold 85%

```
NormalizeText myNorm = new NormalizeText();
myNorm.normalize("resources\\token_baru.csv",
"resources\\output_baru_per_token_85.csv", 0.85);
```

Kode 5.69 Implementasi pengujian *threshold* 85%

Kode 5.69 menampilkan potongan kode program yang akan menjalankan pengujian sistem dengan menggunakan *threshold* bobot lebih dari atau sama dengan 85%. Mulanya program memanggil kelas NormalizeText dengan inisiasi objek bernama myNorm. Kemudian memanggil *method* normalize(). Setelah method ini menerima 3 parameter yaitu file testing, hasil

testing, dan *threshold*. Nantinya file akan disimpan kedalam file output_baru_per_token_85.csv.

5.6.2.2.6 Threshold 90%

```
NormalizeText myNorm = new NormalizeText();
myNorm.normalize("resources\\token_baru.csv",
"resources\\output_baru_per_token_90.csv", 0.90);
```

Kode 5.70 Implementasi pengujian *threshold* 90%

Kode 5.70 menampilkan potongan kode program yang akan menjalankan pengujian sistem dengan menggunakan *threshold* bobot lebih dari atau sama dengan 90%. Mulanya program memanggil kelas NormalizeText dengan inisiasi objek bernama myNorm. Kemudian memanggil *method* normalize(). Setelah method ini menerima 3 parameter yaitu file testing, hasil testing, dan *threshold*. Nantinya file akan disimpan kedalam file output_baru_per_token_90.csv.

BAB VI
HASIL DAN PEMBAHASAN

6.1 Data *Crawling*

6.1.1 Hasil Data *Facebook*

Proses pengambilan data yang didapatkan dari tanggal 1 Agustus 2015 hingga 25 Maret 2017 adalah sebanyak 377.138 post Facebook dan terdapat 361.494 data posting Facebook yang unik. Data post yang unik inilah yang digunakan untuk melakukan pembuatan model word2vec. Selain itu juga terdapat data komentar yang berhasil didapatkan sebanyak 3.131.989. Data komentar pada tugas akhir ini tidak akan digunakan karena memiliki struktur kalimat yang tidak jelas dan terdapat banyak campuran ragam bahasa yang digunakan.

6.1.2 Hasil Data *Twitter*

Proses pengambilan data yang didapatkan dari tanggal 7 Oktober 2016 hingga 25 Maret 2017 adalah sebanyak 1.022.562 post Twitter dan terdapat 828.391 data posting Twitter yang unik. Data post yang unik inilah yang digunakan untuk melakukan pembuatan model *word2vec*.

6.2 Data Percobaan

6.2.1 Hasil Data Setelah Pra-Proses

Data awal yang terdiri dari data twitter dan facebook memiliki jumlah total 1.202.529 data berubah menjadi 1.077.814 data setelah dilakukan beberapa tahapan pra-proses pada sub bab 4.3. Hal ini terjadi dikarenakan pada saat melewati tahapan penghapusan data yang terduplikasi menggunakan Notepad++ sehingga data yang memiliki kesamaan akan terhapus dan setiap baris nantinya merupakan data yang unik. Data yang dihasilkan nantinya akan terurut berdasarkan abjad. Namun, pada tahapan ini tidak Data yang berhasil dihasilkan dapat dilihat pada Gambar 6.1.

```
Hai Terkait info Tcash infokan nomor HP via DM Agar dibantu cek dan privasi terjaga
Hai Terkait info area 4G konfirmasi via DM apakah nomor yg digunakan 082293623xxx T
Hai Terkait info bonus Halofit konfirmasi via DM Apakah nomor HP 08118888xxx Agar d
Hai Terkait info cek bonus konfirmasi via DM apakah nomor yg digunakan 08126601xxx
Hai Terkait info dicapture infokan nomor HP via DM Agar dibantu cek dan privasi ter
Hai Terkait info gangguan infokan nomor HP dan lokasi detail via DM Agar dibantu ce
Hai Terkait info ganti kartu konfirmasi via DM Apakah nomor HP 082213080xxx Agar di
Hai Terkait info kartu 3G4G saya cek terlebih dahulu Mohon ditunggu Kaka
Hai Terkait info kartu sim infokan nomor yg digunakan via DM agar dibantu cek dan d
Hai Terkait info kartuHalo DM  Nomor  Nama Lengkap  TTL atau  Nama Ibu Kandung Ag
Hai Terkait info kode PUK Silakan infokan nomor nama TTLnama ibu kandung via DM aga
Hai Terkait info komunitas CUG konfirmasi via DM apakah nomor HP 082257533xxx Agar
Hai Terkait info kuota Flash infokan nomor HP via DM Agar dibantu cek dan privasi t
Hai Terkait info kuota Midnight kuota tersebut dapat digunakan pada semua jaringan
```

Gambar 6.1 Hasil Pra-Proses

Sehingga dapat disimpulkan seperti dalam Tabel 6.1 jumlah data yang berkurang setiap tahapan pra-prosesnya.

Tabel 6.1 Rekap Jumlah Data Hasil Pra-proses

Tahapan	**Jumlah Data**
Penggabungan dataset	1.202.529 data
Penghapusan simbol	1.202.529 data
Penggabungan baris yang terpisah	1.202.529 data
Penghapusan data yang terduplikasi	1.077.814 data
Tokenizing	1.077.814 data

Pada tahapan penggabungan dataset hingga penggabungan baris yang terpisah tidak terjadi pengurangan data dikarenakan memang tidak ada yang dihapus. Proses penghapusan simbol hanya menghapus simbol dalam satu data dan penggabungan baris terpisah menjadikan sebuah data menjadi satu baris, sehingga jumlah data tidak berkurang. Untuk tahapan penghapusan data yang terduplikasi terjadi banyak pengurangan data karena data yang terhapus bukan hanya yang memiliki kesamaan kalimat/terduplikasi saja namun juga menghapus data yang berupa baris kosong sisa dari penghapusan simbol pada tahapan sebelumnya. Untuk baris kosong yang terhapus sebanyak 3.542 data dan 121.173 data yang hilang akibat terduplikasi. Selanjutnya untuk tahapan

tokenizing tidak terjadi pengurangan data karena itu hanya proses pembuatan token dari data yang sudah ada.

6.2.2 Hasil Pembagian Data

Pembagian data yang dilakukan pada tahapan ini adlaah untuk melakukan pengujian. Data yang digunakan untuk pengujian merupakan sampel data yang diambil dari data training. Hal ini dikarenakan keterbatasan model word2vec yang tidak bisa memprediksi kata apabila kata tersebut tidak ada dalam *vocabulary model word2vec.* Sehingga data yang diambil berasal data training yang notabene telah tersimpan dalam *vocabulary.*

Hasil dari pembagian dataset ini adalah 1.077.814 untuk data training dan seribu untuk data sampel pengujian. Artinya setiap data testing diambil dari data training yang sudah ada. Kemudian dari setiap data sampel pengujian akan dilakukan pelabelan yang berisikan kalimat yang benar dan baku serta sesuai dengan Kamus Besar Bahasa Indonesia secara *manual*.

Untuk menguji seberapa jauh sistem dapat melakukan substitusi token dengan tepat, maka pengamatan hasil dilakukan pada token-token dari seribu data sampel tadi. Oleh karena itu, setelah dilakukan pelabelan, maka akan dibuat per token. Hasil dari pelabelan hingga tokenisasi dapat dilihat pada Gambar 6.2.

kalimat	raw_word	correction
1	tim	tim
1	spe	spe
1	itbsc	itbsc
1	kembali	kembali
1	menjadi	menjadi
1	jawara	jawara
1	dalam	dalam
1	cerdas	cerdas
1	cermat	cermat

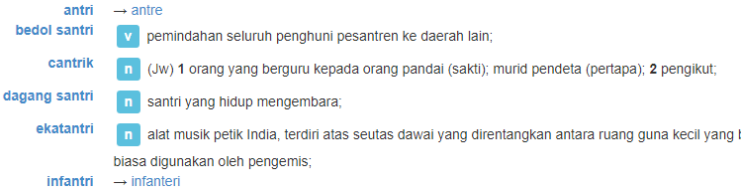
Gambar 6.2 Hasil pelabelan kata

Kolom kalimat menunjukkan berasal dari kalimat keberapakah token tersebut, kemudian kolom kedua berisikan token asli, dan kolom ketiga berisi token hasil koreksi manual. Hasil yang didapatkan adalah berisi 19.404 token.

6.3 Leksikon Bahasa Indonesia

6.3.1 Hasil Data Kateglo

Pada proses pengambilan data dari website Kateglo, sistem berhasil mengumpulkan sebanyak total 72.253 kata bahasa Indonesia yang unik beserta artinya. Pada tahap pengambilan data Katgelo ini disimpan kedalam dua tabel, yaitu kamusterbaru dan kamus *mapping*. Hal ini dikarenakan adanya konten pembenaran kata pada halaman Kateglo, selain itu merupakan konten kata dan artinya dalam kamus. Untuk melihat contoh tampilannya dapat dilihat pada Gambar 6.3.



Gambar 6.3 Tampilan Halaman Kamus Kateglo

Dalam gambar diatas terlihat bahwa terdapat dua tipe tampilan kamus, yan pertama kata “antri” dibenarkan menjadi “antre” dan kata lain seperti “ekasantri” beserta artinya. Konten yang memiliki tanda panah (right arrow) seperti “antri” dan “infanteri” akan masuk kedalam kamus *mapping* sedangkan konten yang lain seperti “ekasantri”, “cantrik”, “bedol santrik”, dan “dagang santri” akan masuk kedalam tabel kamusterbaru.

Jadi, pada akhirnya terdapat 4.598 kata yang telah termaping dengan kata yang benar serta 67.655 kata yang terdapat pada kamus seperti Gambar 6.4.

kata	lex	src	typ	arti
abaktinal	adj	KBBI3	r	adj (Bio) berkenaan dengan sisi tubuh yang tidak mengandung mulut, s
abakus	n	KBBI3	r	n 1 dekak-dekak; swipoa; 2 (Ars) lempeng datar di atas kepala tiang den
abal-abal	n	Kateglo	r	n 1 penjahat kelas kakap; 2 adj palsu; imitasi; 3 peti mati suku Batak;
aban	n	KBBI3	r	n (Antr) sebutan bagi duda dalam masyarakat Dayak Kayan;

Gambar 6.4 Data kamus Kateglo

Sedangkan untuk kata-kata yang telah terpetakan tadi akan berfungsi sebagai pengecekan pertama pada sistem normalisasi, jika kata yang diperiksa terdapat pada tabel maping, maka akan mengembalikan hasil sesuai dengan pembenaran kata pada tabel tersebut. Contoh data yang ada pada tabel mapping dapat dilihat pada Gambar 6.5.

34	aestetika	estetika	1
35	afdol	afdal	1
36	afkir	apkir	1
37	agel	agal	1
38	agribisnis	agrobisnis	1

Gambar 6.5 Hasil Mapping

Gambar diatas menunjukkan bahwa setiap kata yang tidak benar memiliki pembenaran katanya serta memiliki status. Status ini digunakan sebagai penanda jika pembenaran kata tersebut telah divalidasi kebenarannya. Hal ini dilakukan guna mengantisipasi penambahan data untuk kamus mapping yang didapatkan dari hasil pengoreksian yang dilakukan oleh sistem normalisasi nantinya.

6.3.2 Hasil Data Wiktionary

Dalam proses pengambilan data Wiktionary yang bertujuan untuk mendapatkan kata non-KBBI yang merupakan turunan kata dari kata dasar di KBBI, berhasil mendapatkan 14.743 kata bahasa Indonesia non-KBBI. Selain itu juga terdapat 225 kata yang tidak baku yang terdiri dari kata tanpa makna, ragam cakapan sehari-hari, dan penulisan singkatan yang kurang tepat seperti trims, yg, dan lain lain. Sehingga kata yang dapat digunakan berkurang menjadi 14.518. Contoh tampilan data yang telah diambil dapat dilihat pada Gambar 6.6. Dapat dilihat

bahwa gambar menunjukkan macam-macam kata turunan dari kata “abadi”.

id	kata
1	abadikan
2	abadikanlah
3	diabadikan
4	kauabadikan
5	kuabadikan
6	abahkan

Gambar 6.6 Hasil data non-KBBI Wiktionary

6.3.3 Hasil Data Google Translate

Proses pengambilan data dengan Google Translate ini bertujuan untuk memperkaya kosa kata kata turunan yang termasuk di bahasa Indonesia, dalam kata lain cara untuk mendapatkan *corpus* bahasa Indonesia. Untuk mendapatkan *corpus* tersebut perlu dilakukan beberapa tahapan, yaitu menambah kata dasar dalam KBBI dengan afiks yang ada dalam bahasa Indonesia. Daftar afiks yang ditambahkan dapat dilihat pada Tabel 6.2.

Tabel 6.2 Daftar Afiks

Prefiks	Sufiks	Konfiks
di	an	ke-an
me	I	pe-an
meng	kan	peng-an
ber	ku	pem-an
pe	mu	peny-an
per	nya	penge-an
peng	kah	per-an
ter	lah	se-nya

Ke	tah	ber-an
Se	man	be-an
	wati	di-kan
	wan	memper-kan
	at	me-i
	ika	di-i
	in	me-kan
	is	ber-kan
	ir	diper-kan
	ita	memper-i
	or	diper-i
	tas	ter-i
	si	
	ur	
	us	
	ris	
	isme	
	isida	
	isasi	

Dari daftar afiks diatas satu persatu akan digabung dengan kata dasar yang ada kemudian diperiksa kedalam Google Translate apakah ada hasil terjemahan atau tidak, dan apabila kata yang dikirimkan ke Google Translate tadi memiliki pembenaran kata atau dengan kata lain menggunakan fitur "*Google Did You Mean?*", maka yang dimasukkan ke dalam tabel adalah pembenaran katanya. Adanya proses pembenaran kata ini lah yang menyebabkan terdapat beberapa nama objek, tempat,

merk dagang, maupun nama orang bisa terdeteksi dan disimpan kedalam tabel.

Hasil pemeriksaan kata berimbuhan yang mempunyai hasil terjemahan bahasa inggris adalah sebanyak 6.225. Setelah dilakukan pemeriksaan terhadap hasil tersebut terdapat 2.491 kata yang tidak bermakna. Kata yang tidak bermakna biasanya merupakan kata bahasa inggris maupun nama-nama objek yang tidak diketahui. Sehingga hanya tersisa sebanyak 3.734 kata yang dapat ditambahkan kedalam leksikon.

Gambar 6.7 menunjukkan penyimpanan kata yang telah diperiksa terjemahan bahasa inggrisnya.

kata	translate
aborsi	abortion
absensi	absenteeism
absolutisme	absolutism
abstrakisme	abstractism
abstraksi	abstraction

Gambar 6.7 Hasil Google Translate

6.3.4 Pembahasan Hasil Leksikon Indonesia

Tujuan dari pengumpulan data yang dilakukan pada sub bab sebelumnya adalah untuk membuat leksikon/kamus bahasa Indonesia yang tidak hanya terdiri dari kata dasar saja tetapi juga kata turunanya. Oleh karena itu hasil-hasil yang telah didapatkan pada tahap sebelumnya digabungkan hingga menjadi total 82.462 dan terdapat satu kamus untuk memetakan kata yang tidak benar dengan data sebanyak 5.205 kata. Penambahan kamus yang terpetakan ini dikarenakan adanya beberapa pemetaan kata tambahan yang ditambahkan secara manual yang didapatkan dari beberapa *website* yang telah memetakan mana kata yang tidak baku menjadi baku seperti ebahasaidonesia[24] dan kbbi.web.id[25]. Beberapa daftar contoh kata yang ditambahkan dapat dilihat pada Tabel 6.3.

Tabel 6.3 Tambahan manual

Apotek → apotik
apoteker → apotiker
Analisis → analisa
Andal → handal
Antre → antri

Selain itu pada proses pembuatan leksikon bahasa Indonesia terdapat banyak sekali tantangan dalam melakukannya. Salah satunya adalah membedakan kelas kata ragam cakapan yang ada pada data kamus Kateglo, kelas kata kependekan, dan juga berupa sinonim terhadap sebuah kata. Hal itu semua dapat dilihat pada kolom arti seperti Gambar 6.8.


```
- {
      kata_key: "ajojing",
    - arti: [
          "n (cak) (akr) ayo jingkrak jingkrak; dansa dengan gerakan
          berjingkrak; "
      ]
  },
- {
      kata_key: "akal-akal",
    - arti: [
          "adj (cak) pura-pura; dibuat-buat; "
      ]
  },
- {
      kata_key: "akua",
    - arti: [
          "n (kp) akuades; "
      ]
  },
- {
      kata_key: "ala",
    - arti: [
          "l 1 atas; pada; kepada; akan; 2 adj tinggi; 3 (cak) secara;
          4 n (Huk) tanah yang tidak dikerjakan lagi, tetapi pemilik
          atau keturunannya masih mempunyai hak utama atas tanah itu; "
      ]
  }
```


Gambar 6.8 Tampilan leksikon tanpa filter pada Solr

Adanya kata yang mengandung arti seperti diatas menyebabkan ragam kata dengan konteks yang sama semakin banyak dan menyebabkan kata-kata cakapan sehari-hari seperti gue, lo, dan dah akan dihitung sebagai kata baku. Oleh karena itu perlu dilakukan penyaringan agar kata-kata yang mengandung arti tersebut tidak ditampilkan saat pemeriksaan kata baku. Untuk melakukan hal tersebut maka diperlukan penambahan Filter Query pada Query Solr, penambahannya adalah seperti Kode 6.1.

```
-(arti:(cak) OR arti:tandapanahsec OR arti:(kp))
```

Kode 6.1 Filter Query Solr

Dari potongan kode diatas dapat dilihat terdapat operator negasi yang menandakan bahwa tidak akan menampilkan arti yang mengandung (cak) untuk kelas cakapan, (kp) kelas kependekan, atau tandapanahsec untuk kata yang mempunyai pembenaran kata/sinonim. Dengan adanya hal ini maka dapat mengantisipasi kata-kata yang sebenarnya tidak baku namun terdapat pada kamus.

6.4 Model Word2Vec

6.4.1 Hasil Perhitungan Kemunculan Kata

Untuk dapat melakukan pengujian akurasi dari model yang dihasilkan dari word2vec, maka perlu dipisahkan antara kata yang sesuai kamus dan tidak beserta frekuensi kemunculannya.

6.4.1.1 Kemunculan Kata Sesuai Kamus

Proses perhitungan kemunculan kata yang hasilnya sesuai di kamus menghasilkan sebanyak 20.377 kata unik. Tabel 6.4 menunjukkan peringkat 10 teratas dari kata yang paling sering muncul pada data training yang sesuai dengan kamus.

Tabel 6.4 Frekuensi kemunculan kata sesuai kamus

Kata	Frekuensi
Di	191367
Dan	91000

yang	76601
Ini	69488
Ke	54929
Ada	54822
untuk	51399
Via	45306
Bisa	42969
Dari	39734

6.4.1.2 Kemunculan Kata Non-Kamus

Proses perhitungan kemunculan kata yang hasilnya tidak sesuai dengan kamus menghasilkan sebanyak 198.523 kata unik. Tabel 6.5 menunjukkan peringkat 10 teratas dari kata yang paling sering muncul pada data training.

Tabel 6.5 Frekuensi kemunculan kata non-kamus

Kata	Frekuensi
yg	52155
jokowi	32067
dm	20016
ahok	17687
tks	15712
infokan	15525
rp	13632
telkomsel	12911
sebanyakbanyaknya	12792
dibantu	12711

Salah satu kegunaan adanya proses ini adalah sebagai cara untuk mendapatkan kata-kata yang akan diujikan terhadap model-model word2vec yang telah dihasilkan. Pengujian akan

dilakukan pada seratus kata teratas dari hasil proses ini. Seratus kata yang diplih menjadi sampel dari kata-kata yang ada untuk mengukur seberapa tepat model dapat memprediksi pembenaran kata. Daftar kata dapat dilihat pada Tabel 6.6.

Tabel 6.6 Daftar kata tidak baku untuk sampel pengujian pemilihan model

Yang	km	manchester	gresik
jokowi	dki	dialami	blm
Dm	dgn	grapari	update
Ahok	digunakan	kpk	bali
Tks	bro	dg	in
infokan	jl	promo	flash
Rp	xxx	timnas	juventus
telkomsel	udah	jd	of
sebanyakbanyaknya	sidoarjo	online	gunakan
dibantu	jln	pastikan	bs
jakarta	the	sy	dpr
surabaya	aktivasi	dicek	pln
Utk	ditunggu	tl	jgn
Nya	united	anies	dpt
Sby	cc	live	klo
Dr	wib	dilakukan	lg
Ga	tokopedia	barcelona	twitter
tweet	bl	diberikan	lakukan
Lalin	pilkada	fc	freeport
salman	krn	gb	vs

gak	dlm	ri	chelsea
sms	internet	ektp	restart
tdk	city	ikuti	liverpool
hp	jg	its	madrid
sdh	tp	bukalapak	pd

6.4.2 Pembahasan Hasil Kemunculan Kata

Dari kata-kata yang tidak ada di kamus dengan frekuensi kemunculan seratus teratas di dominasi oleh kata-kata slang, singkatan, ragam cakapan dalam bentuk tulisan, brand, nama, tempat, dan juga akronim. Hal ini disebabkan karena data yang diambil berasal dari sosial media yang mana berbagai macam orang menuliskan pemikiran mereka dengan berbagai gaya, ada yang baku dan ada juga yang disingkat seperti gaya bercakapan. Untuk nama objek lain yang sering muncul seperti nama tempat, negara, brand, maupun instansi dapat disebabkan dari data yang diambil dari sebuah akun sosial media. Dikarenakan data yang diambil mayoritas adalah sebuah pemberitaan, maka banyak terdapat nama objek seperti yan telah disebutkan sebelumnya.

Dari hasil tahapan ini, nantinya seratus kata teratas ini akan digunakan untuk menguji akurasi dari sebuah model word2vec karena cukup dengan menggunakan serratus sampel sudah dapat meilihat pebandingan akurasi dari masing-masing model sedangkan untuk seribu teratas digunakan untuk menganalisis model yang terpilih dengan akurasi terbaik dengan tujuan untuk melihat posisi kemunculan kandidat serta kor dari masing-masing algoritma.

6.4.3 Hasil Percobaan Model Word2Vec

Setiap model yang didapatkan dari berbagai macam percobaan proses training, akan diujikan terhadap seratus kata teratas non-kamus yang telah dihasilkan pada tahapan sebelumnya. Dari kata-kata yang tidak baku tersebut akan diprediksi kandidat pembenarannya hingga sepuluh kandidat pertama yang sesuai

dengan kamus. Untuk menghitung akurasinya, rumus yang digunakan adalah dengan memepertimbangkan posisi, yaitu sebagai berikut:

$$A = \sum_{i=1}^{100} \left(1 - (\frac{\mathrm{i}}{10})\right) \qquad \text{(persamaan 5)}$$

Dari persamaan 5 dapat dilihat bahwa akurasi (A) didapat dari penjumlahan bobot dari posisi ditemukan kandidat kata yang benar. Posisi penemuan kandidat dilambangkan dengan huruf (i).

6.4.3.1 Uji Model 1

Dalam melakukan pengujian model 1 yang memiliki konfigurasi seperti pada Tabel 6.7 didapatkan hasil akurasi sebesar 4,5%

Tabel 6.7 Parameter Model 1

Parameter	Isian
Learning Algortihm	Skip-Gram
Training Algorithm	Negative Sample
Iterasi	1
Minimum Word Frequency	5
Context Window	5
Layer Size	100

Hal ini menunjukkan bahwa Skip-Gram memiliki akurasi yang cukup kecil karena mayoritas dari hasil yang diprediksi tidak sesuai dan yang brehasil pun menempati posisi yang agak jauh.

6.4.3.2 Uji Model 2

Dalam melakukan pengujian model 2 yang memiliki konfigurasi seperti pada Tabel 6.8 didapatkan hasil akurasi sebesar 17,5%.

Tabel 6.8 Parameter Model 2

Parameter	Isian
Learning Algortihm	CBOW
Iterasi	1
Minimum Word Frequency	5
Context Window	5
Layer Size	100

Hasil pengujian menunjukkan adanya peningkatan akurasi ketika memggunakan CBOW. Hal ini dikarenakan kata yang ditemukan pembenarannya lebih baik daripada model sebelumnya.

6.4.3.3 Uji Model 3

Dalam melakukan pengujian model 3 yang memiliki konfigurasi seperti pada Tabel 6.9 didapatkan hasil akurasi sebesar 4,5%.

Tabel 6.9 Parameter Model 3

Parameter	Isian
Learning Algortihm	Skip-Gram
Training Algorithm	Negative Sample (10)
Epoch	1
Iterasi	10
Minimum Word Frequency	5
Context Window	5
Layer Size	100

Hasil pengujian menunjukkan tidak ada perubahan walaupun telah ditambahkan iterasi sebanyak 10 dan *negative sample* sebanyak 10. Penambahan *negative sample* dilakukan sesuai dengan rekomendasi dari Mikolov bahwa hasil pengujian terbaik dilakukan pada konfigurasi Skip-Gram dan Negative Sample[4].

6.4.3.4 Uji Model 4

Dalam melakukan pengujian model 4 yang memiliki konfigurasi seperti pada Tabel 6.10 didapatkan hasil akurasi sebesar 23,5%.

Tabel 6.10 Parameter Model 4

Parameter	Isian
Learning Algortihm	CBOW
Training Algorithm	*Hierarchical Softmax*
Epoch	10
Iterasi	1
Minimum Word Frequency	10
Context Window	5
Layer Size	100

Sedangkan penambahan *epoch* pada CBOW dapat meningkatkan akurasi. Selain itu CBOW pada konfigurasi ini juga menggunakan *Hierarchical Softmax* sebagai *training algorithm*.

6.4.3.5 Uji Model 5

Dalam melakukan pengujian model 5 yang memiliki konfigurasi seperti pada Tabel 6.11 didapatkan hasil akurasi sebesar 5,4%.

Tabel 6.11 Parameter Model 5

Parameter	Isian
Learning Algortihm	Skip-Gram
Training Algorithm	Negative Sample (10)
Epoch	1
Iterasi	5
Minimum Word Frequency	5
Context Window	100

Meskipun telah ditambahkan iterasi sebanyak 5 tetap saja tidak ada perubahan yang signifikan dari model yang menggunakan Skip-Gram.

6.4.3.6 Uji Model 6

Dalam melakukan pengujian model 6 yang memiliki konfigurasi seperti pada Tabel 6.12 didapatkan hasil akurasi sebesar 24,9%.

Tabel 6.12 Parameter Model 6

Parameter	Isian
Learning Algortihm	CBOW
Training Algorithm	*Hierarchical* Softmax
Epoch	1
Iterasi	5
Minimum Word Frequency	5
Context Window	100

Penambahan iterasi menjadi 5 dan epoch menjadi 1 dapat membuat model 6 menjadi lebih baik akurasinya. Hal ini menunjukkan bahwa penambahan epoch serta iterasi dapat meningkatkan akurasi CBOW.

6.4.3.7 Uji Model 7

Dalam melakukan pengujian model 7 yang memiliki konfigurasi seperti pada Tabel 6.7 didapatkan hasil akurasi sebesar 2,9%.

Tabel 6.13 Parameter Model 7

Parameter	Isian
Learning Algortihm	Skip-Gram
Training Algorithm	Negative Sample
Epoch	10
Iterasi	3
Minimum Word Frequency	5
Context Window	10
Layer Size	100

Model 7 memperlihatkan bahwa Skip-Gram semakin tidak cocok untuk digunakan dalam studi kasus pembenaran kata karena akurasinya semakin menurun dari sebelumnya.

6.4.3.8 Uji Model 8

Dalam melakukan pengujian model 8 yang memiliki konfigurasi seperti pada Tabel 6.14 didapatkan hasil akurasi sebesar 25%.

Tabel 6.14 Parameter Model 8

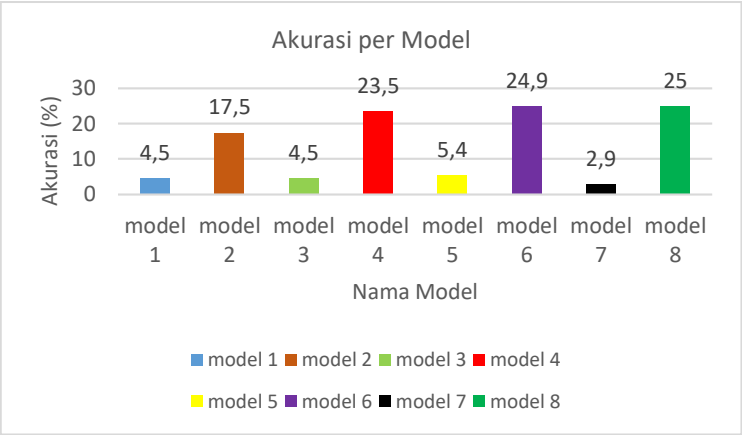
Parameter	Isian
Learning Algortihm	CBOW
Training Algorithm	Hierarchical Softmax
Epoch	10
Iterasi	2
Minimum Word Frequency	5
Context Window	5
Layer Size	500

Penambahan epoch dan iterasi pada model 6 menjadikan salah satu dasar penambahan epoch dan iterasi pada model 8. Yang terjadi adalah kenaikan akurasi yang cukup baik dari sebelumnya. Hal ini juga dikarenakan adanya penambahan *layer size* menjadi 500.

6.4.4 Pembahasan Hasil Percobaan Model Word2Vec

6.4.4.1 Model Word2Vec Terbaik

Dari hasil 8 kali percobaan dapat dilihat akurasi dari masing-masing model dalam Gambar 6.9.

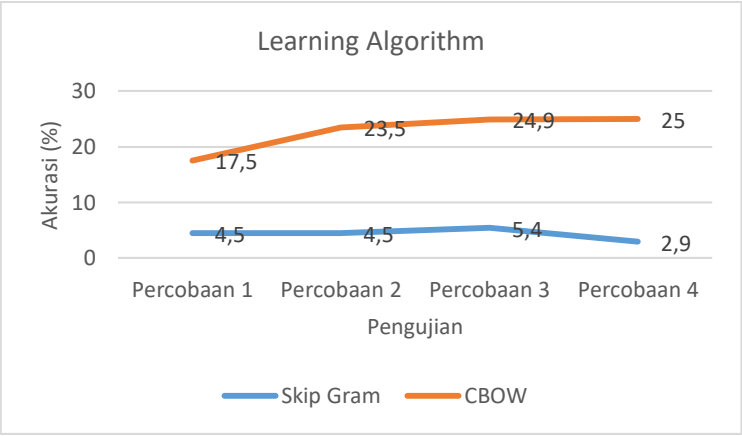


Gambar 6.9 Akurasi per Model

Dari gambar diatas dapat dilihat model dengan akurasi terbaik adalah model dari ke-8.

6.4.4.2 Parameter Penting Training Word2Vec

Sedangkan untuk perbandingan parameter *Learning Algorithm* yang digunakan di masing-masing percobaan dapat dilihat pada Gambar 6.10



Gambar 6.10 Perbandingan learning algorithm

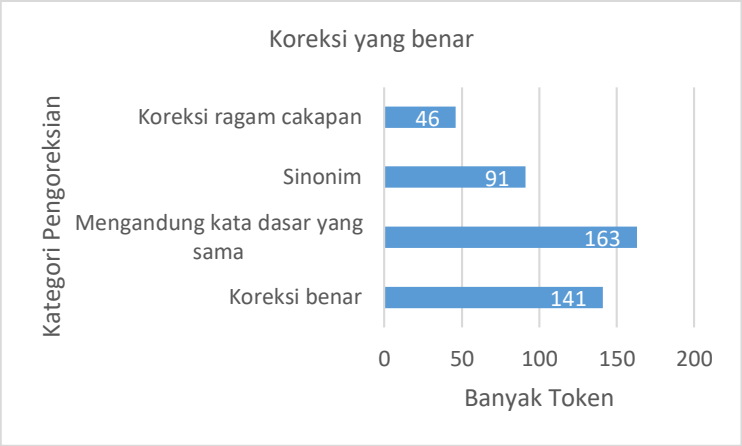
Gambar diatas menunjukkan bahwa CBOW merupakan algoritma yang cocok digunakan dalam kasus ini, dikarenakan tingkat akurasinya semakin meningkan sedangkan Skip-gram dalam setiap percobaannya mengalami penurunan.

Dari dua paparan diatas maka dapat disimpulkan bahwa model yang akan digunakan adalah model ke-8 dengan *learning algorithm* CBOW dengan parameter yang paling penting adalah *Learning Algorithm* karena dampak dari penggunaannya berdampak sangat signifikan terhadap akurasi yang dihasilkan.

6.5 Prediksi Normalisasi Teks

6.5.1 Hasil Pengujian dengan seribu Kata paling sering muncul non-kamus

Seribu kata teratas yang telah dipilih akan diujikan kembali terhadap model terpilih yaitu model ke-8. Pengujian dilakukan dengan memunculkan sepuluh kandidat kata terdekat/pembenarannya. Dengan metode skoring yang dilakukan seperti sub bab 4.6.1.2, hasilnya menunjukkan bahwa nilai skor tertinggi mencapai 97,49% dan terendah adalah 5%. Dari 10000 prediksi yang dilakukan terdapat 441 prediksi yang dinilai benar dan 9559 prediksi yang salah. Untuk komposisi dari koreksi yang benar dapat dilihat pada Gambar 6.12



Gambar 6.11 Grafik hasil koreksi yang benar dari 1000 kata

Gambar 6.12 menunjukkan contoh data dari hasil pengujian seratus data ini dengan kategori seperti Gambar 6.12.

token	correction	posisi	skor_w2c	skor_lev	skor_jarko	bobot	keterangan
international	internasional	0.00	1.00	0.923	0.976	97.49%	3
waspadai	waspada	0.00	1.00	0.875	0.987	96.56%	2
antrian	antrean	0.00	1.00	0.857	0.943	95.00%	1
karna	karena	0.00	1.00	0.833	0.961	94.86%	1
shalat	salat	0.00	1.00	0.833	0.950	94.58%	1
nelpon	telpon	0.00	1.00	0.833	0.889	93.06%	4
temen	teman	0.00	1.00	0.800	0.907	92.67%	1
narkoba	narkotika	0.00	1.00	0.667	0.921	89.68%	0

Gambar 6.12 Hasil testing seribu kata non-kamus

Gambar terdiri 8 kolom yang mana kolom token merupakan kata yang tidak ada dikamus, kolom correction untuk pembenaran katanya, kolom posisi adalah posisi ditemukan kandidat pembenarannya pada daftar kandidat word2vec, kolom skor_w2c untuk menilai posisi dari kandidat, kolom skor_lev untuk menilai similaritas kata dengan pembenarannya menggunakan *levenshtein distance*, kolom skor_jarko berisikan nilai dari perhitungan *jarko-winkler distance*, total bobot, dan kolom keterangan yang mana berisikan kategori pengoreksian. Berikut adalah daftar kategori pengoreksian dapat dilihat Tabel 6.15.

Tabel 6.15 Kategori hasil pengujian dalam seribu kata

Kategori	Keterangan
Salah	0
Koreksi benar	1
Mengandung kata dasar yang sama	2
Sinonim	3
Koreksi ragam cakapan	4

6.5.2 Pembahasan Hasil Pengujian Seribu Kata

Dari hasil yang telah dihasilkan terdapat beebrapa pembahasan yang dapat dilakukan.

6.5.2.1 Penentuan *Threshold*

Dari hasil diatas dapat dilihat dari setiap kategori yang berhasil dikoreksi dengan benar memiliki rentang skor tertinggi dan terendah yang dapat dilihat pada Tabel 6.16.

Tabel 6.16 Nilai maksimum-minimum hasil pengujian 1000 kata

Kategori	Max	Min
Koreksi benar	97,49%	53,33%
Mengandung kata dasar yang sama	96,56%	50,35%
Sinonim	86,34%	51,30%
Koreksi ragam cakapan	93,06%	53,75%
Rata-rata	**93,36%**	**52,18%**

Dari hasil rata-rata terendah maka dibuatlah *threshold* yang akan digunakan sebagai batasan kapan sebuah pengoreksian dikatakan berhasil dikoreksi. Nilai *threshold* awalnya yang digunakan menggunakan rata-rata dari nilai terendah yaitu 52,18%. Namun, nilai tersebut dirasa terlalu rendah karena ternyata apabila memakai nilai tersebut sebagai *threshold* maka dikhawatirkan akan semakin banyak kata yang nantinya sistem berhasil memprediksi dengan nilai yang melebihi *threshold* tapi seharusnya hasil prediksinya itu salah. Hal ini dapat dilihat pada nilai maksimum dari kategori salah pengoreksian mencapai

89,68%. Jadi, kemungkinan salah pengoreksian akan semakin besar.

Oleh karena itu threshold akan dinaikkan sesuai dengan rata-rata nilai minimum dari kategori satu dan empat dengan syarat kata hasil pengoreksian kata yang dipilih merupakan kandidat urutan pertama dalam hasil prediksi word2vec-nya. Kategori satu jelas dipilih karena merupakan hasil koreksi yang telah dilabeli benar. Sedangkan kategori empat dipilih karena sebenarnya koreksi yang dilakukan sudah benar namun dikoreksi sebagai ragam cakapan. Setelah dilakukan perhitungan dengan nilai minimum posisi 0 dari kategori satu adalah 65,83% dan untuk kategori empat adalah 72,22% maka rata-ratanya adalah 69,025%. Maka, nilai threshold yang dihasilkan adalah pembulatan keatas dari hasil rata-rata yaitu 70%. Selain itu untuk menambah realibilitas dari hasil percobaan, maka dilakukan beberapa percobaan *threshold* yaitu 65%, 70%, 75%, 80%, 90%. Dengan adanya percobaan threshold ini nantinya dapat menghasilkan rentang *threshold* yang bisa digunakan agar hasil yang dihasilkan lebih baik.

6.5.2.2 Penambahan Kata dalam Kamus Mapping

Pada sub bab sebelumnya telah terbentuk dua kamus/leksikon yaitu Kamus untuk Kata Bahasa Indonesia dan Kamus Mapping yang berisi kata tidak baku beserta koreksinya. Dari data yang telah dilabeli sebagai pengoreksian yang benar pada sub bab 6.5.1, maka kata tersebut beserta hasil koreksinya akan dimasukkan kedalam tabel kamus mapping dan segera dilakukan pengindeksan ulang dalam Solr.

6.5.2.3 Penambahan *Treatment*

6.5.2.3.1 Kata Berimbuhan

Dalam melakukan pengoreksian kata selain dicocokan kedalam kamus juga terdapat beberapa treatment yang digunakan untuk kata-kata yang telah diprediksi dengan sistem bukan lewat kamus. Salah satu kasus yang sering muncul adalah pengoreksian kata berimbuhan yang telah baku dengan bentuk

kata yang sama, namun dengan imbuhan yang berbeda, contohnya kata “diakses” yang dikoreksi menjadi “mengakses”.

Berdasarkan hasil 6.5.1 terdapat kategori 2 yaitu “mempunyai kata dasar yang sama” yang dengan peringkat kejadian sering muncul terbesar yaitu 163 dari seribu kasus. Kasus semacam ini disebabkan karena tidak semua varian konjugasi dari kata terdapat dalam kamus. Sedangkan untuk dapat menghasilkan semua kombinasi yang mungkin adalah suatu hal yang cukup susah dilakukan.

Dengan fakta bahwa dalam kasus variasi konjugasi semacam ini, kata token akan termuat seluruhnya pada token pengganti yang diusulkan ataupun sebaliknya, sebagai contoh:

- Abaikan → mengabaikan
- Tidur → tertidur
- Termakan → makan
- Diakses → mengakses

Maka, solusi yang dilakukan adalah melakukan pemeriksaan kata dasar menggunakan Lemmatisasi. Lemamatisasi digunakan karena kata kerja aktif dan pasif dalam bahasa Indonesia hanya dibedakan melalui imbuhan. Kata aktif dan pasif sama sekali tidak mengubah konteks, contoh:

- Adik *memanggil* temannya
- Adik *dipanggil* temannya

Dari contoh diatas dapat dilihat bahwa kata dasar dari dua kata miring bergaris bawah tersebut adalah panggil. Lemmatisasi akan mengembalikan kata token maupun usulan koreksi mnejadi kata dasar, sehingga apabila mempunyai kata dasar yang sama, maka kata usulan koreksi dianggap benar.

6.5.2.3.2 Kata Perulangan/Majemuk

Setelah dilakukan analisis pada hasil pengujian bab 6.5.1 ternyata terdapat salah satu isu yang muncul yaitu isu tokenisasi. Isu tokenisasi merupakan isu yang penting dalam melakukan pengoreksian data. Tokenisasi dilakukan sebelum proses pengoreksian. Dalam dalam penelitian ini, tokenisasi

dilakukan setelah pra-proses data. Masalah yang muncul adalah hilangnya simbol *dash* yang mana menandakan bahwa kata tersebut merupakan kata perulangan. Hal ini disebabkan karena penggantian simbol dash menjadi karakter kosong yang terjadi dalam tahap pra-proses data, contohnya:

- Tempat-tempat → tempattempat
- Sekolah-sekolah → sekolahsekolah

Adanya permasalahan diatas menyebabkan proses tokenisasi terpengaruh, karena nantinya terdapat perbedaan hasil dalam hal tokenisasi. Seharusnya, proses pembersihan data dari simbol-simbol tertentu dilakukan setelah proses tokenisasi dilakukan dan dengan aturan tertentu agar kejadian diatas tidak terulang. Namun, dalam penelitian ini terdapat fakta bahwa kata perulangan yang kehilangan tanda hubungnya dapat dikoreksi menjadi kata dasarnya, contoh:

- Tempattempat → tempat
- Sekolahsekolah → sekolah

Hal ini menandakan bahwa untuk beberapa kata perulangan mampu dikoreksi menjadi kata dasarnya.

Oleh karena itu, solusi yang dilakukan ketika melakukan uji coba pada data testing adalah dengan melakukan pemeriksaan pada kolom koreksi manual pada data testing yang dipisahkan menjadi *array* berdasarkan simbol (-) yang kemudian setiap elemen *array* tersebut akan diperiksa kemiripannya dengan hasil koreksi yang dilakukan oleh sistem. Apabila keduanya mirip maka dapat dipastikan bahwa kata tersebut adalah kata perulangan dan pengoreksian dianggap benar. Sebagai gambaran dapat dilihat pada contoh pada Tabel 6.17.

Tabel 6.17 Contoh treatment perulangan

Raw Text	***Manual Correction***	***Auto - correction***	**Status**
Tempattempat	Tempat-tempat	Tempat	benar

Tabel diatas menunjukkan bahwa hasil pembagian kata dari kata “tempattempat” menjadi [tempat, tempat]. Kemudian dari kedua hasil tersebut dibandingkan dengan kata dasar “tempat” yang mana menghasilkan status benar pengoreksiannya.

Selain menggunakan cara diatas, untuk kedepannya pemeriksaan kata ulang/majemuk dapat menggunakan cara lemmatisasi terlebih dahulu. Kata yang terdeteksi sebagai majemuk akan dilemmatisasi hingga terbentuk kata tunggalnya, baru dari kata tunggal itu diperiksa dikamus, jika tidak ada maka akan masuk ke proses pengoreksian kata. Namun hal ini dilakukan dengan catatan simbol *dash* (-) tidak dihilangkan agar sebuah kata dapat dideteksi sebagai kata majemuk.

6.5.3 Hasil Pengujian pada Data Sampel Pengujian

Tabel 6.18 Hasil Pengujian Data Sampel

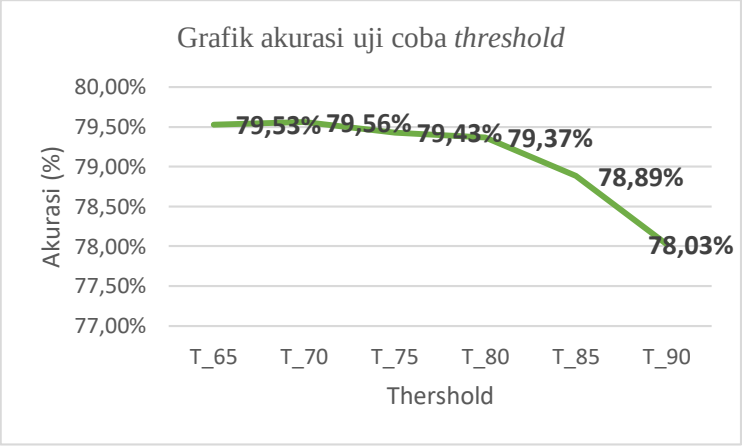
Kategori	T_65	T_70	T_75	T_80	T_85	T_90
Berhasil dikoreksi (Total)	**15432**	**15438**	**15412**	**15400**	**15307**	**15140**
Kata berhasil dikoreksi karena telah sesuai dengan kamus bahasa Indonesia	14385	14385	14385	14385	14385	14385
Kata berhasil dikoreksi karena telah terpetakan hasil pembenarannya dalam kamus mapping.	560	560	560	560	560	560
Kata berhasil dikoreksi karena memiliki lemma yang sama dengan hasil koreksi manual.	342	359	339	333	283	161
Kata berhasil dikoreksi yang mana kata dasarnya yang berasal dari kata ulang.	57	53	53	53	26	0
Kata berhasil dikoreksi dan sesuai dengan hasil koreksi manualnya.	88	81	75	69	53	34
Gagal dikoreksi (Total)	**3972**	**3966**	**3992**	**4004**	**4097**	**4264**

Gagal diprediksi, tidak ada di kamus bahasa Indonesia maupun daftar kandidat hasil prediksi *word2vec*.	481	1246	2372	3103	3482	3782
Berhasil dikoreksi, namun tidak sama/salah dengan hasil koreksi manual.	3381	2574	1402	617	281	108
Gagal karena tidak ada di kamus, maupun hasil *word2vec* dan juga tidak sama dengan label manual.	110	146	218	284	334	374
Rata-rata	**79.53%**	**79.56%**	**79.43%**	**79.37%**	**78.89%**	**78.03%**

Dari Tabel 6.18 dapat disimpulkan bahwa threshold dengan akurasi terbaik adalah *threshold* T_70 dengan akurasi mencapai 79,56%. Selain itu T_70 juga memiliki pengoreksian yang mengandung kata dasar terbanyak yaitu 359 kata. Kemudian untuk pengoreksian benar yang berisikan kata ulang paling banyak dimiliki oleh *threshold* T_65. Untuk pemrediksian yang tepat dengan label manual juga terbanyak adalah threshold T_65 dengan 88 kata.

Untuk tingkat kegagalan paling besar adalah pada *threshold* T_90. Jumlah kata yang tidak ada di kamus yaitu mencapai 3782. Sedangkan untuk ketidaktepatan dalam melakukan pengoreksian berada pada T_65 yaitu 3381 kata. Untuk pengoreksian yang tidak ada di kamus dan hasilnya berbeda dengan label manual paling banyak terdapat pada T_80 yaitu 284. Untuk lebih jelasnya dapat dilihat pada pembahasan yang ada di sub bab 6.5.4.

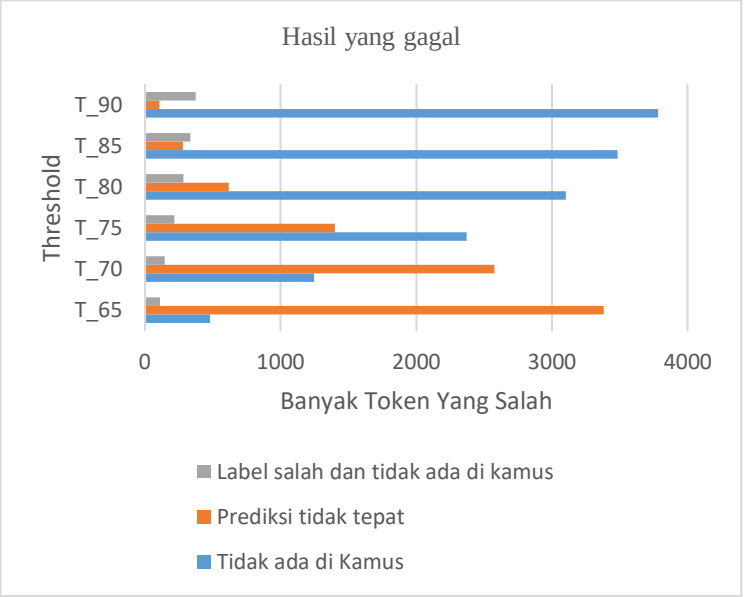
6.5.4 Pembahasan Hasil Pengujian



Gambar 6.13 Perbandingan Akurasi Hasil Uji

Gambar 6.13 menunjukkan bahwa semakin kecil *threshold* belum tentu memilki akurasi yang lebih bagus, dan semakin besar juga belum tentu memiliki akurasi yang lebih bagus.

Penurunan akurasi ini disebabkan adanya penambahan kata yang gagal diprediksi seperti pada Gambar 6.14.



Gambar 6.14 Hasil pengujian yang gagal

Semakin kecil threshold yang diberikan maka kata yang diprediksi semakin besar namun tidak tepat semua seperti Nampak pada T_65, sehingga jumlah prediksi yang tidak tepat semakin banyak. Sedangkan apabila threshold yang diberikan semakin besar maka kata dengan prediksi yang tidak tepat dapat diatasi namun menyebabkan kata lain tidak bisa diprediksi atau tidak terdapat dikamus seperti pada T_90. Contoh kasus dapat dilihat pada Gambar 6.15.

Token	Manual	T_90		T_85		T_80		T_75		T_70		T_65	
spe	spe	spe	tidak ada di kamus	spe	tidak ada di kamus	spe	tidak ada di kamus	spe	tidak ada di kamus	perban	tidak tepat	petroleum	tidak tepat
itbsc	itbsc	itbsc	tidak ada di kamus	itbsc	tidak ada di kamus	itbsc	tidak ada di kamus	itbsc	tidak ada di kamus	itbsc	tidak ada di kamus	institut	tidak tepat
ketemu	bertemu	ketemu	tidak ada di kamus	bertemu	diprediksi sama	bertemu	diprediksi sama	bertemu	diprediksi sama	bertemu	diprediksi sama	bertemu	diprediksi sama
nakanakk	nak-anakk	nakanakk	gagal dan tidak sesuai actual	nakanakk	gagal dan tidak sesuai actual	nakanakk	gagal dan tidak sesuai actual	nakanakk	gagal dan tidak sesuai actual	pahlawan	tidak tepat	anakku	tidak tepat
halhal	hal-hal	halhal	gagal dan tidak sesuai actual	halhal	gagal dan tidak sesuai actual	hal	diprediksi kata ulang	hal	diprediksi kata ulang	hal	diprediksi kata ulang	hal	diprediksi kata ulang
tks	erima kasi	tks	gagal dan tidak sesuai actual	tks	gagal dan tidak sesuai actual	tks	gagal dan tidak sesuai actual	tks	gagal dan tidak sesuai actual	tks	gagal dan tidak sesuai actual	tks	gagal dan tidak sesuai actual

Gambar 6.15 Contoh kegagalan prediksi

Dari gambar diatas dapat dilihat bahwa salah satu contoh dari kata “spe” pada *threshold* T_90 akan dideteksi sebagai kata yang tidak ada di kamus sedangkan pada T_65 dapat diprediksi namun tidak tepat, hal ini dikarenakan rendahnya *threshold* yang diberikan sehingga tidak mampu untuk memberikan kandidat pembenaran yang tepat. Sedangkan untuk keterangan gagal dan tidak sesuai dengan actual terjadi ketika kata yang dikembalikan tidak sesuai dengan koreksi manualnya atau sama dengan keteranagn label salah dan tidak ada di kamus. Apapun hasil yang didapat dari prediksi yang gagal akan dikembalikan sesuai kata semula kecuali untuk prediksi yang tidak tepat dikembalikan sesuai dengan hasil prediksinya walaupun tidak tepat.

Selain membandingkan hasil yang gagal, untuk menentukan threshold mana yang digunakan juga diperlukan membandingkan hasil yang benar diprediksi. Perbandingan hasil untuk yang kata dasar sama pada dilihat pada Gambar 6.16.

Token	Manual	T_90		T_85		T_80		T_75		T_70		T_65	
dinilai	dinilai	dinilai	tidak ada di kamus	dinilai	kata dasar sama	dinilai	kata dasar sama	dinilai	kata dasar sama	dinilai	kata dasar sama	dinilai	kata dasar sama
trima	terima	trima	gagal dan tidak sesuai actual	trima	gagal dan tidak sesuai actual	triliunan	tidak tepat	triliunan	tidak tepat	terima	kata dasar sama	terima	kata dasar sama
dibintangi	dibintangi	dibintangi	tidak ada di kamus	dibintangi	tidak ada di kamus	menangka	tidak tepat	dibintangi	kata dasar sama	dibintangi	kata dasar sama	diperankan	tidak tepat

Gambar 6.16 Contoh hasil kata dasar sama

Dari gambar diatas dapat dilihat bahwa tidak semua kasus threshold memiliki keterangan prediksi yang sama. Kasus "dinilai" pada T_90 mempunyai keterangan tidak ada dikamus karena tidak mendapatkan hasil apapun dari *word2vec* dengan threshold yang terlalu besar, sedangkan pada T_80-70 berhasil memprediksi dengan keterangan kata dasar sama yang. Selain itu pada T_65 untuk kasus "dibintangi" menghasilkan keterangan tidak tepat karena yang dihasilkan "diperankan" notabene adalah sinonimnya.

Untuk contoh kasus diprediksi kata ulang yang memiliki jumlah terbanyak pada T_65 sebanyak 57 kata dapat dilihat pada Gambar 6.17.

Token	Manual	T_90		T_85		T_80		T_75		T_70		T_65	
timtim	tim-tim	timtim	gagal dan tidak sesuai actual	timtim	gagal dan tidak sesuai actual	timtim	gagal dan tidak sesuai actual	timtim	gagal dan tidak sesuai actual	biota	tidak tepat	tim	diprediksi kata ulang
jauhjauh	jauh-jauh	jauhjauh	gagal dan tidak sesuai actual	jauhjauh	gagal dan tidak sesuai actual	jauhjauh	gagal dan tidak sesuai actual	jauhjauh	gagal dan tidak sesuai actual	jodoh	tidak tepat	jauh	diprediksi kata ulang
rungburu	rung-buru	rungburu	gagal dan tidak sesuai actual	rungburu	gagal dan tidak sesuai actual	rungburu	gagal dan tidak sesuai actual	rungburu	gagal dan tidak sesuai actual	bertempu	tidak tepat	burung	diprediksi kata ulang

Gambar 6.17 Contoh hasil diprediksi kata ulang

Dari gambar tersebut dapat dilihat pada kasus kata "timtim" untuk T_90-75 tidak dapat memprediksi kata itu dan mendeteksi sebagai kata yang tidak ada di kamus, sedangkan untuk T_70 diprediksi dengan tidak tepat dikarenakan posisi penemuan kandidat "biota" dalam *word2vec* lebih awal dibandingkan "tim", sehingga yang dikeluarkan adalah biota. Untuk T_65 berhasil memprediksi sebagai kata ulang dengan nilai keluaran "tim" yang mana menunjukkan bahwa perankingan pada T_65 paling besar adalah "tim" karena memiliki kesamaan sintaks.

Untuk contoh kasus diprediksi benar yang memiliki jumlah terbanyak pada T_65 sebanyak 88 kata dapat dilihat pada Gambar 6.18.

Token	Manual	T_90		T_85		T_80		T_75		T_70		T_65	
bikin	buat	bikin	tidak ada di kamus	bikin	tidak ada di kamus	bikin	tidak ada di kamus	hasilkan	tidak tepat	begini	tidak tepat	buat	diprediksi sama
msih	masih	msih	gagal dan tidak sesuai actual	msih	gagal dan tidak sesuai actual	msih	gagal dan tidak sesuai actual	msih	gagal dan tidak sesuai actual	mudi	tidak tepat	masih	diprediksi sama
vidio	video	vidio	gagal dan tidak sesuai actual	vidio	gagal dan tidak sesuai actual	vidio	gagal dan tidak sesuai actual	vidio	gagal dan tidak sesuai actual	lirikan	tidak tepat	video	diprediksi sama

Gambar 6.18 Contoh diprediksi sama

Dari gambar diatas dapat dilihat bahwa kata "bikin" tidak adapat dikoreksi pada T_90-80, namun menghasilkan koreksian yang tidak tepat pada T_75-70 dan menghasilkan koreksi yang tepat pada T_65 yang menghasilkan kata "buat". Hal ini terjadi karena terlalu rendahnya *threshold* pada T_65, yang menyebabkan kata-kata kandidat yang memiliki posisi lebih jauh daripada yang ditemukan pada T_70, mempunyai nilai yang sangat besar ketika diranking sesuai dengan bobot karena memiliki kesamaan sintaks.

Dari contoh-contoh kasus tersebut dapat ditarik kesimpulan bahwa *threshold* ideal adalah mempunyai rentang 65% – 70% agar tidak terlalu kecil dan tidak terlalu besar.

6.5.4.1 Hasil Temuan Lainnya

Selain menghasilkan kata-kata yang dapat diprediksi beserta *threshold* paling baik, tahapan ini juga menghasilkan bebrapa temuan baru. Pada threshold T_65 yang notabene terlalu rendah dapat menghasilkan kata prediksi yang berupa sinonimnya ataupun hasil terjemahan apabila kata masukannya adalah bahasa asing. Contohnya seperti pada Tabel 6.19.

Tabel 6.19 Daftar sinonim dan terjemahan

Singapore→singapura
Industry→industri
President→presiden
Nature→natural
Pergub→peraturan

Kerjasama→kesepakatan
Tercecer→berserakan
Digoreng→dimasak
Dipajang→dipamerkan
Polri→polisi
Amnesty→amnesti
Citacita→mimpi
Professional→profesional
Dibintangi→diperankan

Untuk pengoreksian yang menghasilkan terjemahan dikhawatirkan tidak sesuai dengan maksud dalam bahasa asingnya. Misalkan memang sebuah token adalah merupakan pembentuk frasa inggris, misal "International English Competition", apabila setiap token ini diterjemahkan, maka akan memiliki makna yang berbeda. Oleh karena itu, diperlukan adanya pendeteksi bahasa asing diawal sebelum melakukan pengoreksian kata agar kata bahasa asing dapat tersarin terlebih dahulu.

Namun hal ini tidak termasuk dalam lingkup tugas akhir ini karena tugas akhir ini hanyalah mengoreksi terkait kesamaan sintaks saja tidak pada lingkup kesamaan konteks/sinonim maupun terjemahannya, sehingga hasil-hasil seperti tampak pada Tabel 6.19 diabaikan dan tidak dianggap sebagai hasil koreksi yang benar.

BAB VII
KESIMPULAN DAN SARAN

Pada bab ini dibahas mengenai kesimpulan dari semua proses yang telah dilakukan dan saran yang dapat diberikan untuk pengembangan yang lebih baik.

7.1 Kesimpulan

Kesimpulan yang didapatkan dari proses pengerjaan tugas akhir yang telah dilakukan antara lain:

1. Pra-proses data adalah salah satu tahapan yang krusial. Dimana setiap data dari *facebook* dan *twitter* memiliki karakteristik yang berbeda, sehingga perlakuan yang diberikan adalah berbeda. Oleh karena itu, diperlukan adanya perlakuan yang berbeda ketika melakukan pra-proses data dari kedua sumber tersebut.
2. Proses penghapusan simbol dengan regular expression merupakan suatu hal yang sangat perlu diperhatikan. Karena apabila kita melakukan penghapusan yang salah maka makna kata bisa jadi berbeda. Oleh karena itu perlu adanya berkali-kali percobaan akan *regular expression* yang dibuat agar hasil lebih akurat.
3. Dalam membangun leksikon bahasa Indonesia total terdapat 82.462 kata dan terdapat satu kamus untuk memetakan kata yang tidak benar dengan data sebanyak 5.205. Banyaknya kata yang terdapat pada leksikon semakin membuat kosa-kata bahasa untuk pemeriksaan dan pembenaran kata ini menjadi lebih baik. Oleh karena itu perlu adanya lagi pengayaan serta evaluasi untuk kedua kamus ini agar hasil yang dihasilkan kedepannya lebih baik.
4. Dataset yang digunakan dalam penelitian ini mengandung total 20.377 kata unik dalam kamus dan 198.523 kata unik yang tidak ada dalam kamus. Oleh karena itu data sosial media merupakan data yang sangat banyak ragamnya serta mengandung probabilitas yang besar akan munculnya kata-kata non-kamus.

5. Dari berbagai macam hasil training dengan konfigurasi yang berbeda-beda dapat ditemukan model terbaik yang akan digunakan ialah model ke-8 dengan parameter *learning algorithm* CBOW, iterasi sebanyak 2, 5 *minimum word frequency*, *context window* sebanyak 5, *epoch* sebanyak 10, *layer size* sebesar 500 dimensi, dan menggunakan *hierarchical softmax*. Akurasi yang dimiliki oleh model ini apabila diujikan terhadap 100 kata non-kamus adalah 25%.
6. Parameter yang dianggap penting dalam proses *training* adalah penggunaan *learning algorithm*. *Learning algorithm* digunakan sesuai dengan tujuan digunakannya sebuah model. Karena tujuan dari model ini nantinya akan memprediksi sebuah kata menurut konteksnya maka hasilnya akan lebih baik jika memakai *Continous Bag of Words* daripada *Continous Skip-Gram*. Karena CBOW mampu memprediksi kata dari konteks yang diberikan.
7. Selama proses percobaan banyak hal-hal yang ditemui, salah satunya adalah perlunya *treatment* untuk beberapa kasus yang ditemui. *Treatment* yang digunakan dalam penelitian ini adalah untuk mendeteksi apabila memiliki lemma yang sama antara kata asli yang non-kamus dengan kata hasil koreksi sistem serta *treatment* untuk kata berulang/majemuk yang apabila hasil koreksinya berupa kata dasar kata berulang tersbeut maka dianggap benar.
8. Penentuan *threshold* juga menjadi salah satu parameter penting dalam melakukan pengoreksian menggunakan sistm normalisasi teks ini. *Threshold* yang diberikan tidak boleh terlalu kecil ataupun terlalu besar. Dalam 6 kali percobaan menggunakan threshold sebesar 65%, 70%, 75%, 80%, 85% dan 90% dapat disimpulkan bahwa percobaan dengan akurasi terbaik adalah dengan menggunakan threshold 70% dengan hasil akurasi sebesar 79,56%.
9. Penggunaan metode *Word2Vec*, *Levenshtein Distance* dan *Jaro-Winkler Distance* serta dikolaborasikan dengan pemeriksaan terhadap Leksikon Bahasa Indonesia bisa dianggap cukup baik dalam melakukan pengoreksian datam

dikarenakan hasil akhir dari pengujian menunjukkan angka yang cukup besar yaitu 79,56%.

7.2 Saran

Dari pengerjaan tugas akhir ini, adapun beberapa saran untuk pengembangan penelitian ke depan.

1. Dalam melakukan pengambilan data sebaiknya jika data yang diambil adalah data yang mempunyai lingkup topik yang sama.
2. Dalam melakukan pra-proses juga harus berbeda antar sumber, termasuk ketika melakukan penghapusan karakter tertentu menggunakan *regular expression*.
3. Data Kamus Bahasa Indonesia sebaiknya diambil dari versi terbaru yaitu Kamus Besar Bahasa Indonesia Edisi Kelima yang bersumber secara resmi dari Kementrian Pendidikan dan Kebudayaan Republik Indonesia (https://kbbi.kemdikbud.go.id).
4. Penambahan metode sebagai bentuk pembobotan agar hasil prediksi pengoreksian lebih baik juga dapat dilakukan, karena apabila menggunakan *word2vec*, *levenshtein distance*, dan *jaro-winkler distance* saja mungkin hasil prediksi yang diberikan masih terdapat banyak kesalahan.
5. Penambahan pengoreksian apabila sebuah kata tersebut kata majemuk, maka harus dicari lemmanya dulu baru diperiksa di kamus, jika tidak ada maka masuk ke proses pengoreksian. Selain itu penggunaan lemmatisasi juga tidak hanya berlaku pada kasus kata majemuk saja, namun untuk semua kata, apabila sebuah kata tidak ada di kamus, maka diperiksa apakah ada hasil lemmanya, jika lemmanya ada dikamus maka kata tersebut dianggap benar.
6. Penambahan pendeteksian bahasa asing diawal program agar bahasa asing tidak ikut dikoreksi.
7. Penambahan *post-tagging* untuk mendeteksi nama sebuah objek atau nama orang agar tidak diikutkan dalam proses pengoreksian.
8. Kemudian dalam pengembangannya nanti dapat dibuat sistem pengoreksian otomatis secara *realtime* dikarenakan

hal semacam itu belum banyak dikembangkan khususnya untuk bahasa Indonesia.

9. Algoritma utama dalam penelitian ini yaitu *word2vec* merupakan algoritma yang cukup baik dalam menghasilkan kandidat kata terdekat sesuai dengan konteks, sehingga bagus digunakan dalam hal membuat rekomendasi yang datanya berupa teks.

DAFTAR PUSTAKA

[1] Asosiasi Penyelnggara Jasa Internet Indonesia, "Survey Penetrasi & Perilaku Pengguna Internet Indonesia," Jakarta, 2016.

[2] D. Utami, "Karakteristik penggunaan bahasa pada status facebook," Universitas Sebelas Maret SUrakarta, 2010.

[3] J. Pustejovsky, *Natural Language Annotation for Machine Learning*. .

[4] T. Mikolov, G. Corrado, K. Chen, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *Proc. Int. Conf. Learn. Represent. (ICLR 2013)*, pp. 1–12, 2013.

[5] A. S. Lhoussain, G. Hicham, and Y. Abdellah, "Adaptating the levenshtein distance to contextual spelling correction," vol. 12, no. 1, pp. 127–133, 2015.

[6] A. A. Sorokin and T. O. Shavrina, "Automatic spelling correction for Russian social media texts," 2016.

[7] B. Han, P. Cook, and T. Baldwin, "Lexical Normalisation for Social Media Text," vol. V, no. 212, 2013.

[8] A. Copestack, "Natural Language Processing," *Nat. Lang. Process.*, pp. 2003–2004, 2004.

[9] A. Chopra, A. Prashar, and S. Chandresh, "Natural Language Processing," *Int. J. Technol. Enhanc. Emerg. Eng. Res.*, vol. 1, no. 4, pp. 131–134, 2013.

[10] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning represntations by back-propagating errors," *Nature*, vol. 323, 1986.

[11] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A Neural Probabilistic Language Model," vol. 3, pp. 1137–1155, 2003.

[12] J. Turian, L. Ratinov, Y. Bengio, and J. Turian, “Word Representations: A Simple and General Method for Semi-supervised Learning,” *Proc. 48th Annu. Meet. Assoc. Comput. Linguist.*, no. July, pp. 384–394, 2010.

[13] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Distributed Representations of Words and Phrases and their Compositionality,” *CrossRef List. Deleted DOIs*, vol. 1, pp. 1–9, 2013.

[14] Deeplearning4j Development Team, “Deeplearning4j: Open-source distributed deep learning for the JVM,” *Apache Software Foundation License 2.0*, 2015. [Online]. Available: http://deeplearning4j.org.

[15] D. Jurafsky, “Minimum Edit Distance,” *Natual Lang. Proc.*

[16] Y. Faranika, N. Nikentari, and H. Kurniawan, “SISTEM PENGUKUR KEMIRIPAN DOKUMEN MENGGGUNAKAN ALGORITMA JARO-WINKLER DISTANCE,” pp. 1–8.

[17] A. Kurniawati, “Implementasi Algoritma Jaro-Winkler Distance untuk Membandingkan Kesamaan Dokumen Berbahasa Indonesia.”

[18] V. Qazvinian, E. Rosengren, D. R. Radev, and Q. Mei, “Rumor has it : Identifying Misinformation in Microblogs,” *Conf. Empir. Methods Nat. Lang. Process.*, pp. 1589–1599, 2011.

[19] G. Pant, P. Srinivasan, and F. Menczer, “Crawling the Web.”

[20] A. P. Recommender, O. Social, N. Sites, and H. K. Kushwaha, “Personalized Recommender System,” vol. 93, no. 9, pp. 1–6, 2014.

[21] X. Wu, L. Bartram, and C. Shaw, “Plexus : An Interactive Visualization Tool for Analyzing Public Emotions from Twitter Data,” 2017.

[22] Kateglo, “No Title,” 2017. .

[23] Wiktionary, “No Title,” 2017. .

[24] Ebahsaindonesia, “No Title,” 2017. [Online]. Available: http://www.ebahasaindonesia.com.

[25] Kbbi.web.id, “No Title,” 2017. [Online]. Available: http://kbbi.web.id/.

Halaman ini sengaja dikosongkan

BIODATA PENULIS

Penulis lahir di Surabaya pada tanggal 17 Januari 1996. Merupakan anak kedua dari 4 bersaudara. Penulis telah menempuh beberapa pendidikan formal yaitu; SDN Pacarakeling V/186 Surabaya, SMP Negeri 1 Surabaya, dan SMA Negeri 5 Surabaya.

Pada tahun 2013 pasca kelulusan SMA, penulis melanjutkan pendidikan dengan jalur SBMPTN (Tulis) di Jurusan Sistem Informasi FTIf – Institut Teknologi Sepuluh Nopember (ITS) Surabaya dan terdaftar sebagai mahasiswa dengan NRP 5213100131. Selama menjadi mahasiswa, penulis mengikuti berbagai kegiatan kemahasiswaan seperti beberapa kepanitiaan serta pernah menjabat sebagai Staf Departemen Pengembangan Sumber Daya Mahasiswa BEM FTIf ITS dan pad atahun ketiga menjabat sebagai Vice CEO BEM FTIf ITS. Selain itu, kegiatan seperti Latihan Ketrampilan Manajemen Mahasiswa pun pernah diikuti hingga Tingkat Dasar. Di bidang akademik, penulis aktif menjadi asisten dosen dan asisten praktikum pada beberapa mata kuliah seperti Bahasa Pemrograman, Algoritma dan Struktur Data, Pengembangan Sumber Daya Perusahaan, dan Sistem Cerdas. Selain itu, pada tahun 2016 penulis menjadi peraih medali perak kategori poster PKM-KC pada perlombaan Pekan Ilmiah Nasional ke-29 di Institut Pertanian Bogor.

Pada tahun keempat, karena penulis memiliki ketertarikan di bidang pengolahan data, maka penulis mengambil bidang minat Akuisisi Data dan Diseminasi Informasi (ADDI). Penulis dapat dihubungi melalui *email* di stezarpriansya@gmail.com.

Halaman ini sengaja dikosongkan

LAMPIRAN A

Contoh Data Mentah Facebook

Messages
Muktamar NU rumuskan konsep Islam Nusantara, Islam yang Tanpa Pentungan. Inilah konsep Islam tanpa kekerasan itu. (odp-fk) http://m.suarasurabaya.net/kelanakota/detail.php?id=2rd5iab0l0skf1u7a4ru2jflp32015156524
Guyonan Gus Ipul, Pakde Karwo dan Jokowi di Muktamar NU. (odp-fk) http://m.suarasurabaya.net/kelanakota/detail.php?id=2rd5iab0l0skf1u7a4ru2jflp32015156523
Sebanyak 94 TKI ilegal Dideportasi Malaysia. (odp-rt) http://m.suarasurabaya.net/kelanakota/detail.php?id=ik878thocermn5g8a7r6esbdr72015156517
Datang ke Muktamar, Jokowi Bagikan Kaos dan Kartu Indonesia Pintar. (odp-rt) http://m.suarasurabaya.net/kelanakota/detail.php?id=ik878thocermn5g8a7r6esbdr72015156512
21.45 : Hindari masuk Jombang Kota! Lalu lintas MACET TOTAL. Imas kegiatan Muktamar NU. Sebaiknya gunakan jalur Ploso - Gedeg saja, Kawan. (odp-rt)
Foto almarhum KH Abdurrahman Wahid alis Gus Dur sedang membuka amplop berisi uang Rp5.000 menjadi pusat perhatian pengunjung pameran foto yang digelar jelang Muktamar NU. Foto : Fatkhurrohman Taufik Reporter Suara Surabaya (odp-rt) http://m.suarasurabaya.net/fokus/detail.php?id=ik878thocermn5g8a7r6es01082015156501&fokusid=613

Messages
21.00 : 4 Jalur MACET : 1.Simpang 3 Lakasantri; 2.Depan Stasiun Wonokromo. Imbas banyak taksi berhenti; 3.Singosari - Malang; 4.Jombang - Ploso. Imbas pembukaan muktamar NU ke 33. (odp-rt)
20.50 : Hindari masuk JL Kalidami! JL Kalidami - Unair Kampus B ada bazar. Lalu lintas MACET karena jalur yang dari arah Karang Menjangan digunakan jadi 2 lajur. (odp-rt)
Masih Ada Pilkada Paslon Tunggal, Tahapan Pilkada Lanjut Terus. (odp-rt) http://m.suarasurabaya.net/politik/detail.php?id=ik878thocermn5g8a7r6esbdr72015156515
#SSinfo : Unjuk rasa ribuan karyawan Migas Blok Cepu Bojonegoro ricuh, Sabtu (1/8/2015). Nana reporter Radio Suara Bojonegoro Indah melaporkan ribuan massa merusak kantor dan pos security. Empat Mobil digulingkan dan satu mobil dibakar. Unjuk rasa menuntut dibukanya kembali lima pintu utama masuk dan keluar karyawan. Saat ini, hanya satu pintu yang digunakan. Jadi karyawan harus berdesak-desakan untuk keluar,masuk dan istirahat. Tidak adanya jawaban dari pihak perusahaan memicu timbulnya kericuhan. Imbas kejadian ini, kegiatan diperusahaan diliburkan sampai batas waktu yang tidak bisa ditentukan. Foto : Dokumentasi Radio Suara Bojonegoro Indah. (odp-rt)
19.53 : Info awal : Kecelakaan di jelang FO Peterongan Jombang antara mobil dengan sepeda motor. Lalu lintas MACET. Belum ada polisi dilokasi. (odp-rt)

Messages
19.15 : 3 Jalur MACET malam ini: 1.Jembatan Karangpilang Baru dan lama - Sepanjang; 2.Kletek - Krian. Ada pick up mogok didepan SPBU Kletek; 3.Mojoagung. (odp-rt)
18.53 : Info awal : Kebakaran lahan kosong di depan Lenmarc. Api membesar. PMK sudah menuju lokasi. Foto : Enig Mia via e100. (odp-rt)
18.45 : Adzan Isya telah berkumandang untuk wilayah Surabaya dan sekitarnya. Selamat menunaikan ibadah sholat Isya, Kawan. (Odp-rt)
Suwarno, (50) seorang pemulung warga Bratang Gede, menemukan jenazah bayi di sungai Jagir, Jalan Jagir, Wonokromo, Surabaya, Sabtu (1/8/2015). (odp-rt) http://m.suarasurabaya.net/kelanakota/detail.php?id=ik878thocermn5g8a7r6esbdr72015156502
18.03 : 3 jalur MACET : 1.Simpang 4 Balongsari. Volume kendaraan tinggi; 2.Sukorejo - Purwosari; 3.Jombang - Nganjuk. (odp-rt)
17.52 : Hindari lewat JL Karang Menjangan! Lalu lintas MACET karena ada bazar. Gunakan jalur lain, Kawan. Foto : Aditya Surya Nata via @e100ss. (odp-rt)

Messages
17.40 : Update : Kondisi kijang innova L 1581 JV yang naik ke trotoar dan menabrak sepeda motor di JL Kendangsari - Rungkut. Kondisi pengendara sepeda motor belum diketahui. Foto : Petrus Budi Riyanto via e100. (odp-rt)
17.33 : Adzan maghrib telah berkumandang untuk wilayah Surabaya dan sekitarnya. Selamat menunaikan ibadah sholat maghrib Kawan. (Odp-rt)
KPU RI: Penundaan Pilkada Karena Perilaku Politik Elit Parpol. (odp-rt) http://m.suarasurabaya.net/politik/detail.php?id=ik878thocermn5g8a7r6esbdr72015156504
16.46 : Info awal : Kecelakaan di JL Kendangsari depan Kantor PDIP - Rungkut. Ada Mobil honda CRV naik ke trotoar dan infonya, kendaraan juga menabrak sepeda motor. Belum ada data kendaraan dan kronologi lengkap kejadian. (odp-rt)
16.35 : Purwosari-Malang MACET. Imbas ada trailler muat kepala pesawat yang berjalan pelan. Foto : Irul via e100. (odp-rt)
16.20 : Jalur-jalur MACET sore ini : 1.JL Mastrip-Karang Pilang MACET TOTAL. Imbas jalur Legundi ditutup; 2.Sepanjang - Kletek; 3.Brangkal - Mojokerto. Imbas jalur Bypass ditutup, diduga ada rombongan RI 1 yang menuju Jombang; 4.Beji - Pasuruan setelah Pasar Gondang MACET. (odp-rt)
Seorang penjual es tebu menemukan uang segepok yang terjatuh di dekat gerobaknya saat berjualan di kawasan Jl. Demak Surabaya, Jumat (31/7/2015) siang. Hingga, siang ini pemilik uang belum ditemukan. Jika sampai waktu lama tidak juga ada yang mengambil uang itu, rencananya akan disumbangkan ke

Messages
Masjid dekat rumahnya. (odp-rt) http://m.suarasurabaya.net/kelanakota/detail.php?id=ik878thocermn5g8a7r6esbdr72015156500
15.50 : Bundaran Waru MACET TOTAL SEGALA ARAH. Imbasnya masuk dan keluar tol Waru juga terhambat. Ekor dari arah Sidoarjo sudah sampai U-turn depan RS Mitra Keluarga Waru. Foto : Denny Setiyono via e100. (odp-rt)
15.34 : Info awal :Grand Max muat karung terguling di Tol Sidoarjo - Porong KM 36, posisi dilajur kanan. Lalu lintas masih belum terdampak. Foto : Hendarto Hutama via @e100ss. (odp-rt)
15.15 : Update : Kebakaran sampah limbah plastik milik PT Philips Indonesia Rungkut. Duta Komandan Pleton 4 PMK Rungkut menjelaskan, lokasi yang terbakar adalah area terbuka dan api tidak sampai menjalar ke bangunan pabrik. 3 Unit PMK diturunkan untuk mengatasi kebakaran ini. Api cepat dikuasai karena pabrik juga memiliki sistem hidran yang bagus. (odp-rt)
Muktamar NU akan Bahas BPJS yang Kontroversi. (odp-rt) http://m.suarasurabaya.net/kelanakota/detail.php?id=ik878thocermn5g8a7r6esbdr72015156495
14.49 : Adzan ashar telah berkumandang untuk wilayah Surabaya dan sekitarnya. Selamat menunaikan ibadah sholat ashar, Kawan. (odp-rt)
14.42 : Rangkuman Jalur MACET : 1.Warugunung - Karangpilang - Sepanjang. Imbas jalur Legundi-Wringin Anom ditutup. Foto : Amung Putra via e100' 2.Lidah - Wiyung; 3.HR Muhammad - Mayjend Sungkono;

Messages
4.Manukan - Lempung Tama - Balongsari; 5.Sebelum Simpang 4 Karang Lo Malang. Foto : Rufinus via e100; 6.Depan Ponpes Tebu Ireng Jombang. Imbas acara Muktamar NU. Foto : Joki via e100 (odp-rt)
Kemarau, Kebakaran Alang-Alang Terjadi di Beberapa Lokasi. (odp-rt) http://m.suarasurabaya.net/kelanakota/detail.php?id=ik878thocermn5g8a7r6esbdr72015156489
14.12: Update #kebakaran di Brebek. Bangunan yang terbakar berupa Pabrik Philips, di Jl Brebek Industri 5. Api sudah mulai mengecil, asap juga sudah mulai berkurang. Sudah ada petugas PMK d lokasi. Foto: Poendra via e100. (odp-pr)
14.00: Info #kebakaran di dekat PMK Rungkut, ada pabrik yang terbakar. Lokasi ada di Brebek 1 depan Tjokro dekat Philip. Sudah ada 3 unit PMK yang meluncur ke lokasi. Data dan kronologi masih belum diketahui. (odp-pr)
13.57: Info awal #kecelakaan di depan RS Orthopedi Citraland. Grand Livina warna abu-abu dengan Pick Up warna hitam. Posisi Pick Up melintang. Data dan kronologi belum diketahui. Info sudah diteruskan ke petugas. Foto: Donnie O via e100. (odp-pr)
13.43: 4 jalur ini padat cenderung Macet. 1. Rolak - Kalijaten padat. 2. Jembatan Sepanjang - Bukit Bambe Macet. 3. Simpang 3 TL Lakarsantri - Menganti padat. 4. Lidah Kulon - Simpang 3 Unesa Macet. (odp-pr)

Messages
13.16: Waspada #kebakaran ilalang di Tol KM 20 Waru arah Sidoarjo. Asap sedikit menganggu pendangan pengguna jalan. Info sudah diteruskan ke petugas. Foto: Anjar via e100. (odp-pr)
13.04: Info awal #penemuan jenazah bayi di dekat Kali Jagir. Sudah ada Satpol PP di lokasi. Lalu lintas padat karena banyak kendaraan yg mengurangi kecepatan untuk melihat. Foto: Sumarno via e100. (odp-pr)
Malam Ini, Muktamar NU Siap Dibuka Jokowi Foto: Fatkhurohman Taufik - Reporter Suara Surabaya (odp-pr) http://m.suarasurabaya.net/kelanakota/detail.php?id=aaqvvpqisq9baqleg3e89lg6302015156488
12.40: Hati-hati,kawan. Di Interchange turun Tol Waru ada mobil mogok. Posisi di lajur tengah. Foto: Santoso via @e100ss. (odp-pr)
12.15: Waspadai kepadatan di jalur-jalur ini,kawan. 1. Simpang 4 Babatan masih MACET. 2. Lenmarc arah HR Muhammad padat. 3. Singosari arah Malang padat. Foto: Ghifary via @e100ss. 4. Manukan arah Balongsari padat. 5. Dupak arah PGS padat. 6. Waru arah Trosobo padat. (odp-pr)
11.53: Update #kebakaran ilalang di sebelah barat Terminal Benowo. Ada 4 unit mobil PMK dari Kandangan, Pakal, dan Lakarsantri. Saat ini sedang melakukan pemadaman api yang cukup besar. Lalu lintas di sekitar lokasi padat. Foto: F Lopez via e100. (odp-pr)

Messages
11.39: Kumandang adzan Dzuhur sudah terdengar di Surabaya dan sekitarnya. Selamat menunaikan ibadah shalat Dzuhur, Kawan. (odp-pr)
11.29: Info #kecelakaan, kejadian sekitar pukul 11.15,Truk Terguling di Jl Ngagel Bagong Ginayan. Tidak ada korban. Saat ini ada forklift yang berusaha mengevakuasi Truk dibantu oleh warga. Lalu lintas Macet. Belum ada petugas di lokasi. Foto: Warkoppitulikur (odp-pr)
11.19: Info awal #kebakaran ilalang di sebalah barat Terminal Benowo. Api cukup besar. Di dekat ilalang yg terbakar banyak lapak milik pedagang. Info sudah diteruskan ke petugas PMK. (odp-pr)
11.08: 4 Jalur ini padat. 1. Wiyung 2 arah Macet. Foto: Priyo via e100. 2. Simpang 4 Babatan masih Macet. 3. Manukan Lor arah Margomulyo, depan Bibis 1 ada Truk Kontainer yang Mogok. Posisi Truk di lajur kiri. Lalu lintas Macet. 4. Pendem, Junrejo arah Batu Macet. (odp-pr)
Gunung Manam Meletus, Bandara Merauke Ditutup (odp-pr) http://m.suarasurabaya.net/kelanakota/detail.php?id=aaqvvpqisq9baqleg3e89lg6302015156483
10.49: 2 Jalur ini padat. 1. Simpang 4 Babatan Macet. Dari arah Unesa, ekor antrean sampai di Danau. 2. Flyover Arjosari arah Malang padat. (odp-pr)
10.23: Waspadai kepadatan di jalur-jalur ini, kawan. 1. Depan Ponpes Tebu Ireng padat. Foto: Rahman via e100.

Messages
2. Abdul Karim Rungkut arah Juanda Macet. Ekor antrean sampai di Rungkut Mapan. 3. Bundaran Aloha arah Surabaya padat. Foto: Santoso via @e100ss. 4. Segoromadu Gresik arah Surabaya ada Truk Mogok di lajur kanan. Lalu lintas padat. Antrean sampai Semen Gresik. (odp-pr)
#InspirasiSolusi Ikuti talkshow Inspirasi Solusi Sabtu (1/8/2015) pukul 10.00 - 11.00 WIB dengan topik "Mengidentifikasi dan Mengatasi Resiko Usaha" bersama narasumber DR Tri Siwi - Dosen UMKM dan Kewirausahaan Prodi Manajemen FEB Unair, dipandu penyiar Isa Anshori. Kawan bisa bergabung di 031-5600000. (odp-wd)
10.02: 3 Jalur ini padat. 1. Taman arah Krian padat cenderung Macet. 2. Prapen - Jemursari padat imbas ada pengerjaan penambalan jalan di lajur tengah di sesudah SPBU. 3. Pasar Tanah Merah Bangkalan 2 arah Macet. (odp-pr)
09.52: Jalur Legundi - Wringinanom masih Ditutup, ada pengerjaan jalan. Alternatifnya untuk kendaraan kecil, ada kantor pos kecil masuk belok kiri. Jalan terys menyusuri sungai, tembusnya nanti di dekat PT Malindo mendekati Jetis. (odp-pr)
09.28: Waspadai kepadatan di jalur-jalur ini, kawan. 1. Depan Pusdik Porong arah Malang ada Truk Colt Diesel Mogok. Posisi Truk ada di lajur tengah. Lalu lintas mulai depan Pasar Porong Lama sudah padat. Untuk pengguna jalan yg mengarah ke Malang, sebaiknya lewat arteri baru saja. 2. Depan Polsek Wiyung 2 arah padat.

Messages
3. Di dekat Rumadi Lidah ada tenda kegiatan warga. Lalu lintas Macet karena harus lewat bergantian. Dari arah Citraland sudah ditutup. 4. Ispatindo arah Krian padat. (odp-pr)
Tertibkan Lokalisasi Dolly dan Jarak, Satpol PP Amankan 9 Orang http://m.suarasurabaya.net/kelanakota/detail.php?id=aaqvvpqisq9baqleg3e89lg6302015156476
102 Kabupaten Kekeringan, BNPB Siapkan Rp75 Miliar (odp-pr) http://m.suarasurabaya.net/kelanakota/detail.php?id=aaqvvpqisq9baqleg3e89lg6302015156477
08.42: Update perempuan yang sebelumnya dilaporkan berjalan di jalur Tol KM 35 arah Surabaya. Saat ini sudah diamankan oleh petugas. Lalu lintas ramai lancar. (odp-pr)
08.35: Hati-hati,kawan. Di Mondoroko Singosari arah Malang ada Truk bermuatan Tebu yang mogok. Lalu lintas padat merambat. (odp-pr)

Contoh Data Mentah Twitter

Messages
Good morning SUB CITY, my beloved city @e100ss
@SbyTrafficServ @e100ss @RTMCJatim TL MERRSTIKOM pagi sdhlebihtertib,mohon perhatikanR2 tdk pakaihelm,bykpelajarpakaiR2 apakahsurat lengkap?
RT @Amelandoko: @e100ss jam 19.30-20.30 jalan Dharmawangsa (dpn Alfa Mart) padat. Terjadi pertikaian antara pengendara mobil dan beberapaâ€¦
RT @Amelandoko: @e100ss Erik (warga Gubeng Airlangga 1) menyebutkan 4 debt collector mengaku sbg pers, memukuli pemilik mobil & menuduhnyaâ€¦
RT @Amelandoko: @e100ss setelah polisi datang, diketahui 4 orang yg mengaku anggota pers tsb adlh debt collector. 2 org kabur & lainnya diaâ€¦
Salute pd manajemen @KAI121 yg memperhatikan kebutuhan difabel, maju terus pelayanan publik di Indonesia @e100ss https://t.co/8WuI2WQX9i
RT @nyonyachoi407: @e100ss Kantong plastik msh byar gk sih? Di koran&tv kok beritanya per 1 okt udh free. Tp di Giant Margorejo msh ditarikâ€¦
RT @M_AINUR_ROFIQ: Wasapada demam berdarah https://t.co/oGwivAMiMS @pakdekarwo1950 @gusipul4 @KemenDesa @KemenkesRI @e100ss @dprdkabmoker @â€¦
RT @M_AINUR_ROFIQ: Blangko e-KTP diperkirakan baru tersedia lagi November 2016 https://t.co/dST4CyXMLW @pakdekarwo1950 @Sapa_Kemendagri @diâ€¦

Messages
@e100ss https://t.co/s26PQGhwXc
Blangko e-KTP diperkirakan baru tersedia lagi November 2016 https://t.co/dST4CyXMLW @pakdekarwo1950 @Sapa_Kemendagri @dianalfianti @e100ss
@antoyudhie @e100ss @PTPERTAMINA iyup, dlm keadaan terpaksa merelakan isi pertalite
RT @marifanto: @pln_123 @e100ss daerah jalan panjang jiwo surabaya mati lampu . belum menyala.
Ini dari kemarin lusa masih belum berdiri, Jampirogo bypas Mojokerto... @e100ss https://t.co/4vBGuUMjnU
@pln_123 @e100ss daerah jalan panjang jiwo surabaya mati lampu . belum menyala.
Laka lantas tunggal bus po EKA @spbu bagor, sdh dlm penanganan petugas @lantasnganjuk @e100ss https://t.co/6Xg3FjTazf
@e100ss di jalan kalianak ada truck (container) ban bocor tepat di jalan (sopir tak ada). Pagi nanti berpotensi macet.
SAS Pimpin Pemkot #Tomohon Berguru ke Kota Pahlawan https://t.co/wsW0nxVj9W @Tri_Rismaharini @infosurabaya @SapawargaSby
@e100ss Jangan hanya sekedar membangun tapi lihat dulu potensi daerahnya

Messages
@e100ss @KontraS @kompascom Sedang nonton debat konyol d Jak TV "Presiden Indonesia hrs org Indo Asli" PPP menyerang Ahok, memalukan..
Greges lagi https://t.co/LtGF0ucP3B @KejaksaanRI @e100ss @kissfmjember @Humas_Jember @JatimPemprov @infolumajang @infojatim @Jbr134R
RT @hajar11980478: @e100ss
Mau dilantik ditahan duluan https://t.co/C600WCoWn0 @Hanura10 @wiranto1947 @JatimPemprov @e100ss @KejaksaanRI @lumajangone @infojatim
@MNCPlayMedia @infosurabaya @e100ss @playmedia_sby MNC Play kmi gangguan lg bgmn ini https://t.co/7VdYAl4kaB
RT @AtrosFarosd: @e100ss pasar patemon Bangkalan macet 2 arah.....hampir 2KM....belum ada petugas
@Amelandoko @e100ss lha kok ngawur ngono... Memalukan wartawan rek....
Cuma sekedar membantu , terimakasih @e100ss https://t.co/xY6EfhgADJ

Messages
?? https://t.co/RVIfB1gRGy
RT @irwanfajarsetia: @e100ss kawwsan dukuh tengah buduran sda hujan deras n listrik padam
@ainissfm @e100ss langganan klo macetny dsitu
@e100ss min ada info tentang kecelakaan di pacet pagi tadi ? Truk tangki menabrak spda motor .
RT @lnfosurabaya: Ini Alasan Kejati Tahan Wisnu Wardhana (odp-pr)... https://t.co/VNIp0B9d9mÂ : Ini Alasan Kejati Tahan Wisnu Wa... https:/â€¦
Ini Alasan Kejati Tahan Wisnu Wardhana (odp-pr)... https://t.co/VNIp0B9d9mÂ : Ini Alasan Kejati Tahan Wisnu Wa... https://t.co/ADAVZmmIfY
RT @bonek_klaten: Muehehe mantane sopo iki? ???? https://t.co/qFpPz15UVf
2x belanja di Ind*maret Alun2 Sidoarjo, 2x ada perbedaan harga di kasir/struk & di plakat rak. Harga struk > rak. Ada apa? @e100ss
@akma_k @e100ss mohon atensi dari @PDAMSurabaya
@Amelandoko @e100ss bekerja sesuai porsinya
@Amelandoko @e100ss wah pelanggaran itu namanya
RT @AgendaSurabaya: @e100ss Taman Nginden Intan lagi rame dengan kebut2an motor anak remaja. Mohon ditertibkan @RTMCJatim @TwetPolisi @Sapaâ€¦

Messages
@pln_123 @e100ss terima kasih atas perhatiannya....
Ada apa toh skr malam2 indrapura banyak arak2an sepeda motor mbleyer2? @e100ss
@febry_hr @e100ss msh ada non...
@e100ss PDAM diperum puri surya jaya gedangan masih belum nyala sdh 3 harian gmn nih apa tdk ada truk tangki air sementara
@pln_123 @e100ss terjadi lagi pemadaman bbrp rumah di Perum Puri Indah, Cemengkalang, Sidoarjo. mohon perhatiannya. sudah 3 hari berturut2.
@e100ss Listrik padam di Semolowaru Utara dan sekitarnya. Mohon segera dinyalakan. Sudah sangat sering sekali. Bagaimana PLN ini
Kebakaran Warung Ayam Nelongso di Jl Raya Mulyoagung Dau @e100ss https://t.co/jK0UKvlFx4
@nyonyachoi407 @e100ss di indomaret masih koq,,, diseluruh outlet mereka tetap membebankan kantong plastik berbayar 200 rupiah
Ini Alasan Kejati Tahan Wisnu Wardhana (odp-pr)... https://t.co/yBuX1SeaK3
@ruhanluthfi @e100ss @PTPERTAMINA tp bayar e pertalite kan
@Amelandoko @e100ss mangakne Kalau gak kuat bayar Mobil kembalikan.
@e100ss tlh hilang stnk a/n Mochammad Djuri dgn nopol L 6104 SU hilang disekitar grand city, apabila ada yg menemukan harap hub 081358434206

Messages
RT @akma_k: Kelurahan gundih sdh 2 hari air pdam tak mengalir,tolong PDAM sby kalo masih terkendala alirannya maka dpt kirim truk air u/ waâ€¦
RT @GmailElly: @e100ss tol skrng sering mct hrs nya phk jasa marga kl uda tau macet lngsung di pintu msk yg terdpt antrian pnjng itu lngsunâ€¦
Kelurahan gundih sdh 2 hari air pdam tak mengalir,tolong PDAM sby kalo masih terkendala alirannya maka dpt kirim truk air u/ warga @e100ss
Pacitan Berharap Segera Miliki Bandara (odp-pr)... https://t.co/99bMTxBX3n
RT @Okytriprstyo: @e100ss macet dan banyak polisi di daerah pasar kembang arah ke kedungdoro. Ada yg tau ada apa? #InfoSurabaya
@e100ss macet dan banyak polisi di daerah pasar kembang arah ke kedungdoro. Ada yg tau ada apa? #InfoSurabaya
RT @yudha_shanny: @e100ss Lalin kapas krampung landai lancar di ke 2 arah kawan
@e100ss setelah polisi datang, diketahui 4 orang yg mengaku anggota pers tsb adlh debt collector. 2 org kabur & lainnya diamankan polisi.
@e100ss Lalin kapas krampung landai lancar di ke 2 arah kawan

Messages
@e100ss Erik (warga Gubeng Airlangga 1) menyebutkan 4 debt collector mengaku sbg pers, memukuli pemilik mobil & menuduhnya membawa narkoba
@e100ss jam 19.30-20.30 jalan Dharmawangsa (dpn Alfa Mart) padat. Terjadi pertikaian antara pengendara mobil dan bâ€¦ https://t.co/3EDPy9qJ4C
Mohon idzin share @imandwi35 @e100ss @ainiSSfm #KelanaMedia https://t.co/3nRPTqJjRl
RT @e100ss: #Lazuardi Kawan akhir akhir ini, pemberitaaan heboh, soal Kanjeng Dimas taat pribadi, yang menghalalkan segala... https://t.coâ€¦
@DjokoToleee @e100ss kedung cowek
#Lazuardi @e100ss malam ini, @ainiSSfm mengupas tema, Menghalalkan Segala Cara. Gabung 031560000 sms 08553010055 https://t.co/Qa3LrC32le
RT @e100ss: 20.42: Update #kecelakaan di Kedung Cowek, Melibatkan 2 motor. Motor L 4546 ZK dikendarai Achmad Bauhaqi dengan... https://t.coâ€¦
#Lazuardi Kawan akhir akhir ini, pemberitaaan heboh, soal Kanjeng Dimas taat pribadi, yang menghalalkan segala... https://t.co/MCOJWPDWCY
KBS Segera Punya Dirut Definitif (odp-pr)... https://t.co/bNpb4G30nG

Messages
FPKB Kalah Telak Main Bola Dengan Wartawan (pr) https://t.co/tBBhahB5U2
Rame2 di jl. Dharmahusada & Gubeng Airlangga 2 kenapa ya? Sampe ada Polisi & Satpol PP (")(") @e100ss
RT @RamadityaRandu: @e100ss sebelum fly over lawang Dari arah surabayaa macet bgt ada apa ya?
20.42: Update #kecelakaan di Kedung Cowek, Melibatkan 2 motor. Motor L 4546 ZK dikendarai Achmad Bauhaqi deng... https://t.co/HAwf7ZNDuI
@republikaonline @nkarnadi: '@e100ss jalan ke arah pasar pucang dibongkar, menyaâ€¦ https://t.co/a872igOE88, see more https://t.co/cVuSYgLzI5
RT @PuspitaFM: #JMR #Now !!! #JaringRadio w/ @e100ss @TidarSaktiFM @LIIURFM @karimatafm1033 @Mandala964FM @mettafmsolo @RakosaFeMale @Radioâ€¦

Messages
@e100ss sebelum fly over lawang Dari arah surabayaa macet bgt ada apa ya?
#JMR #JaringRadio "Dewan desak Pemkot Malang untuk konsultasi cari kepastian hukum terkait pambangunan Jembatan Kedungkandang" Cc: @e100ss
#JMR #Now !!! #JaringRadio w/ @e100ss @TidarSaktiFM @LIIURFM @karimatafm1033 @Mandala964FM @mettafmsolo @RakosaFeMale @RadioGeFM dll

Messages
RT @bety_asbrt: @e100ss Hari ini hari istimewa semoga barokah u kita smua aamiin Tgl 6102016 dibaca dari kiri kanan, kanan kekiri juga saâ€¦
" BKKBN IMBAU MASYARAKAT HINDARI PERNIKAHAN DINI " Selengkapnya #JaringRadio cc @e100ss Pkl 20.45 WIB
20.42: Update #kecelakaan di Kedung Cowek, Melibatkan 2 motor. Motor L 4546 ZK dikendarai Achmad Bauhaqi dengan... https://t.co/sUqvLqccxf
@e100ss Taman Nginden Intan lagi rame dengan kebut2an motor anak remaja. Mohon ditertibkan @RTMCJatim @TwetPolisi @SapawargaSby
@e100ss @pdamgresik tlg @yayasan konsumen indonesia dengar keluhan kami warga perum puri asta kencana yg air @pdamgresik nya srg mati!!??
RT @RatehPW: @e100ss pasar pucang muacettt
RT @HIDAYAT42240300: @e100ss sudah ada petugas utk yg menangani kecelakaan arah tol gresik. Pengguna jalan monggo sabar, waspada radiator mâ€¦
RT @e100ss: 14.15 : Update : Posisi kendaraan yang terlibat kecelakaan beruntun di lajur kanan Tol Surabaya - Gresik KM 2.... https://t.coâ€¦
RT @ainissfm: Imbas laka km 2 arah Gresik, ekornya panjang terasa di km 0 atau sekitar tikungan dupak.Dari Satelit arah Perak mulai km 6 meâ€¦
RT @Jonwin14: @e100ss kapan selesai nya box culvert jemur ngawinan?

Messages
@e100ss harusnya di hukum kebiri juga
@e100ss kapan selesai nya box culvert jemur ngawinan?
@e100ss kami pelanggan @pdamgresik Merasa g percaya dg manajemen @pdamgresik Air koq mati dlm sebulan srg skali, sampe skrg blm ngalir??!!
RT @e100ss: 18.07: Info awal #kecelakaan di 100m setelah Simpang 4 Kenjeran arah Suramadu. Melibatkan 2 motor. Data belum... https://t.co/Dâ€¦
@e100ss buset nih orang apa raja
RT @herielectro: @e100ss sdh termasuk penistaan agama hrusnya bsa di hukum seberat beratnya,kasihan yg muslim
@e100ss @PTPERTAMINA (2) beli pertamax dikasih pertalite dg alasan bisa dicampur di spbu dharmahusada surabaya https://t.co/taAqGq6Y1A
@e100ss sdh termasuk penistaan agama hrusnya bsa di hukum seberat beratnya,kasihan yg muslim
@e100ss @PTPERTAMINA sdh 2 kali kejadian (1) beli pertamax dikasih pertalite dg alasan bnyk yg antri spbu sepanjangâ€¦ https://t.co/0ZBaLTmw8k
Cabuli 23 Siswa SMP, Sopir Angkot Dituntut 20 Tahun (pr) https://t.co/BLBEiNULJj
@pdamgresik sampe skrg air blm ngalir!!?? Kami sbg pelanggan sll yg didgr pipa sana sini pecah, apa benar? @e100ss

Messages
#SSInfo Pemprov Jawa Timur berkomitmen mencegah investasi ilegal, dengan cara melakukan perlindungan sektor... https://t.co/Fz6upIVXax
RT @LantasResMGTN: 16.22 Madiun arah Ngawi-Solo via Maospati lalin ramai lancar @e100ss https://t.co/wtlGSJ3cPT
@e100ss emng lebih mudah mempengaruhi orng dgn cara mnggunkn dalil2 agama.
Gus Ipul Pastikan Bocah 7 Tahun asal Tuban Tidak Dipasung (odp-pr)... https://t.co/IfVh3sCxfS

LAMPIRAN B

Contoh Data Sampel Pengujian

RAW TEXT	KOREKSI MANUAL
Tim SPE ITBSC Kembali menjadi Jawara dalam Cerdas Cermat Perminyakan seAsia Pasifik Tim the Society of Pretroleum Engineers Institut Teknologi Bandung Students Chapter SPE ITBSC kembali berhasil menyabet gelar juara pertama dalam kompetisi internasional Petrobowl APOGCE mengalah timtim lainya baik dari Indonesia maupun dari berbagai negara Tim SPESC ITB tersebut terdiri dari Muhammad Iffan Hannanu Chintya Rizkiaputri dan Arnold Rico Novrianto yang ketiganya berasal dari jurusan Teknik Perminyakan Kompetisi Petrobowl merupakan salah satu mata kegiatan dari APOGCE Asia Pacific Oil Gas Conference and Exhibition APOGCE tahun ini diselenggarakan atas kerjasama antara Society of Petroleum	Tim SPE ITBS Kembali menjadi Jawara dalam cerdas cermat Perminyakan seAsia Pasifik Tim the Society of Pretroleum Engineers Institut Teknologi Bandung Students chapter SPE ITBS kembali berhasil menyabet gelar juara pertama dalam kompetisi internasional Petrobowl APOGE mengalahkan tim-tim lainnya baik dari Indonesia maupun dari berbagai negara Tim SPES ITB tersebut terdiri dari Muhammad Iffan Hannanu hintya Rizkiaputri dan Arnold Rico Novrianto yang ketiganya berasal dari jurusan Teknik Perminyakan Kompetisi Petrobowl merupakan salah satu mata kegiatan dari APOGE Asia Pacific Oil Gas conference and Exhibition APOGE tahun ini diselenggarakan atas kerjasama antara Society of Petroleum Engineers SPE kolaborasi dengan

RAW TEXT	KOREKSI MANUAL
Engineers SPE kolaborasi dengan Ikatan Ahli Teknik Perminyakan Indonesia IATMI Pada tahun ini APOGCE terdiri dari beberpa kegiatan yaitu kegiatan Conference Student Paper Contest dan Petrobowl yang diselenggarkan di Nusa Dua Bali Petrobowl sendiri merupakan kompetisi cerdas cermat mengenai dunia oil and gas yang diperuntukan bagi para mahasiswa tingkat Asis Pasifik Kompetisi Petrobowl kali ini mengangkat tema Sustain and Gain Bending the Curve Kompetisi adu kecerdasan pengetahuan kemampuan dan kecepatan tentang dunia oil and gas ini mempertemukan tim dari berbagai nerara seperti Indonesia ITB UPN UGM SST Migas ITS dan UIR Malaysia Universiti …	Ikatan Ahli Teknik Perminyakan Indonesia IATMI Pada tahun ini APOGE terdiri dari beberpa kegiatan yaitu kegiatan conference Student Paper contest dan Petrobowl yang diselenggarkan di Nusa Dua Bali Petrobowl sendiri merupakan kompetisi cerdas cermat mengenai dunia oil and gas yang diperuntukan bagi para mahasiswa tingkat Asis Pasifik Kompetisi Petrobowl kali ini mengangkat tema Sustain and Gain Bending the curve Kompetisi adu kecerdasan pengetahuan kemampuan dan kecepatan tentang dunia oil and gas ini mempertemukan tim dari berbagai nerara seperti Indonesia ITB UPN UGM SST Migas ITS dan UIR Malaysia University …
Zidane dinilai memilik pengalaman untuk menghadapi situasi seperti ini	Zidane dinilai memilik pengalaman untuk menghadapi situasi seperti ini

RAW TEXT	KOREKSI MANUAL
Yuk kita Setiap tahun pemerintah membuat RKP yang menjadi arah pembangunan setahun ke depan Tema RKP tahun adalah Memacu Pembangunan Infrastruktur dan Ekonomi untuk Meningkatkan Kesempatan Kerja serta Mengurangi Kemiskinan dan Kesenjangan Antarwilayah Berdasarkan tema tersebut maka arah kebijakan tahun sebagai berikut peningkatan ekspor non migas barang dan jasa penyederhanaan perizinan dan penyediaan layanan investasi peningkatan ekstensifikasi dan intensifikasi perpajakan PNBP serta penyesuaian tarif penyempurnaan UU perpajakan dan PNPB penerapan reformasi kelembagaan	Yuk kita Setiap tahun pemerintah membuat RKP yang menjadi arah pembangunan setahun ke depan Tema RKP tahun adalah Memacu Pembangunan Infrastruktur dan Ekonomi untuk Meningkatkan Kesempatan Kerja serta Mengurangi Kemiskinan dan Kesenjangan Antarwilayah Berdasarkan tema tersebut maka arah kebijakan tahun sebagai berikut peningkatan ekspor non migas barang dan jasa penyederhanaan perizinan dan penyediaan layanan investasi peningkatan ekstensifikasi dan intensifikasi perpajakan PNBP serta penyesuaian tarif penyempurnaan UU perpajakan dan PNPB penerapan reformasi kelembagaan
Yuk awali harimu dengan kebahagiaan Belanja di Bukalapak setiap hari Senin bisa dapat potongan Rp100 ribu dengan kartu kredit UOB Card Promo ini berlaku untuk minimum belanja Rp1 juta Selengkapnya cek	Yuk awali harimu dengan kebahagiaan Belanja di Bukalapak setiap hari Senin bisa dapat potongan Rp100 ribu dengan kartu kredit UOB Card Promo ini berlaku untuk minimum belanja Rp1 juta Selengkapnya cek

RAW TEXT	KOREKSI MANUAL
Wujudnya memang aneh tapi kalau dengar suaranya pasti ketagihan	Wujudnya memang aneh tapi kalau dengar suaranya pasti ketagihan
woow	wow
Wakil Presiden Jusuf Kalla mendukung langkah PT Pelindo dalam meningkatkkan jumlah mobil crane di setiap pelabuhan	Wakil Presiden Jusuf Kalla mendukung langkah PT Pelindo dalam meningkatkkan jumlah mobil crane di setiap pelabuhan
Wahidin atau Rano Paduan kilat untuk mencoblos besok	Wahidin atau Rano Paduan kilat untuk mencoblos besok
Wah this durian so shiok best lah Apa saja katakata dalam peristilahan Hong Kong dan Singapura yang diserap di kamus baru Oxford	Wah this durian so shiok best lah Apa saja kata-kata dalam peristilahan Hong Kong dan Singapura yang diserap di kamus baru Oxford
Wah teknologi sudah semakin canggih aja ya	Wah teknologi sudah semakin canggih saja ya
Wah sudah cantik bakat menyanyi dan akting pintar pula Selamat ya Maudy D	Wah sudah cantik bakat menyanyi dan akting pintar pula Selamat ya Maudy D
Video Staf Khusus Ahok Kembali Diperiksa KPK	Video Staf Khusus Ahok Kembali Diperiksa KPK
VIDEO GOKIL Jangan Ngaca Sembarangan	VIDEO GOKIL Jangan mengaca Sembarangan
Video Gokil Rekor Push Up	Video Gokil Rekor Push Up

RAW TEXT	KOREKSI MANUAL
Usia kakek saya tahun baru saja ulang tahun kemarin Beliau masih sehat yang penting jangan minum bir	Usia kakek saya tahun baru saja ulang tahun kemarin Beliau masih sehat yang penting jangan minum bir
USBN telah di lalui dengan sukses Semoga hasil nya juga sesuai harapan Persiapkan untuk UBK anakanakku	USBN telah di lalui dengan sukses Semoga hasil nya juga sesuai harapan Persiapkan untuk UBK anak-anakku
Untung AHOK blm tua jd dipanggil KOH Kalo sdh tua manggilnya KONG jadinya KING ketemu KONG KING ketemu	Untung AHOK belum tua jadi dipanggil KOH kalau sudah tua manggilnya KONG jadinya KING ketemu KONG KING ketemu
Untuk pertama kalinya Pesta Persemakmuran akan digelar di kawasan Afrika	Untuk pertama kalinya Pesta Persemakmuran akan digelar di kawasan Afrika
Untuk meningkatkan kunjungan wisatawan dan mempromosikan pariwisata Ogan Komering Ulu OKU Selatan Dinas Kebudayaan dan Pariwisata Kabupaten Ogan Komering Ulu Selatan Provinsi Sumatera Selatan akan menyelenggarakan Festival Danau Ranau keXX pada November di Ponton dan Banding Agung Festival Danau Ranau adalah sebuah pertunjukan budaya yang	Untuk meningkatkan kunjungan wisatawan dan mempromosikan pariwisata Ogan Komering Ulu OKU Selatan Dinas Kebudayaan dan Pariwisata Kabupaten Ogan Komering Ulu Selatan Provinsi Sumatera Selatan akan menyelenggarakan Festival Danau Ranau ke-XX pada November di Ponton dan Banding Agung Festival Danau Ranau adalah sebuah pertunjukan budaya yang

RAW TEXT	KOREKSI MANUAL
menampilkan seni tari dan lagu daerah serta peragaan visualisasi kepariwisataan daerah yang mampu menambah nilai dan promosi pariwisata Selain itu diharapkan bahwa acara ini dapat mempresentasikan potensi wisata sebagai wadah informasi promosi dan pemasaran bagi pengelola obyek wisata meningkatkan kreatifitas para insan pariwisata juga untuk meningkatkan dan menggairahkan apresiasi masyarakat agar lebih mencintai budaya sendiri	menampilkan seni tari dan lagu daerah serta peragaan visualisasi kepariwisataan daerah yang mampu menambah nilai dan promosi pariwisata Selain itu diharapkan bahwa acara ini dapat mempresentasikan potensi wisata sebagai wadah informasi promosi dan pemasaran bagi pengelola obyek wisata meningkatkan kreativitas para insan pariwisata juga untuk meningkatkan dan menggairahkan apresiasi masyarakat agar lebih mencintai budaya sendiri
Untuk mencegah terjadinya tindakan anarkis pihak kepolisian berjanji akan mengerahkan pasukan untuk disiagakan pada saat penggusuran	Untuk mencegah terjadinya tindakan anarkis pihak kepolisian berjanji akan mengerahkan pasukan untuk disiagakan pada saat penggusuran
Untuk km yg suka helm dengan bentuk yg unik dan tidak biasa	Untuk kamu yang suka helm dengan bentuk yang unik dan tidak biasa
Untuk jangka panjang tidak akan mungkin bertahan bahwa hanya dua negara Uni Eropa Jerman dan Swedia yang menanggung sebagian besar pengungsi kata Komisaris Tinggi PBB	Untuk jangka panjang tidak akan mungkin bertahan bahwa hanya dua negara Uni Eropa Jerman dan Swedia yang menanggung sebagian besar pengungsi kata Komisaris Tinggi PBB

RAW TEXT	KOREKSI MANUAL
untuk Pengungsi Antonio Guterres kepada koran Jerman Die Welt	untuk Pengungsi Antonio Guterres kepada koran Jerman Die Welt
Universitas Islam Indonesia akan bertaraf Internasional terus apa bedanya	Universitas Islam Indonesia akan bertaraf Internasional terus apa bedanya
Uanguang kuno yang banyak diburu dan punya nilai tinggi	Uang-uang kuno yang banyak diburu dan punya nilai tinggi
Toleransi menjadi penawar ujian yang terjadi dalam sebuah rumahtangga tak terkecuali pada pernikahan mantan Presiden dan Ibu Negara Amerika Serikat	Toleransi menjadi penawar ujian yang terjadi dalam sebuah rumah tangga tak terkecuali pada pernikahan mantan Presiden dan Ibu Negara Amerika Serikat
Tips Sederhana Hadapi Cuaca Ekstrim di Madinah Suasana sekitar Masjid Nabawi Madinah Pinmas Cuaca Saudi Arabia jelang musim haji tahun ini memasuki musim panas Bahkan di Madinah suhu udara siang hari bisa mencapai derajat celsius Malam hari sekalipun masih terasa panas dengan suhu mencapai derajat celsius Ditambah dengan udara bercampur debu pasir halus dapat mengganggu pernapasan Menghadapi	Tips Sederhana Hadapi cuaca Ekstrim di Madinah Suasana sekitar Masjid Nabawi Madinah Pinmas cuaca Saudi Arabia jelang musim haji tahun ini memasuki musim panas Bahkan di Madinah suhu udara siang hari bisa mencapai derajat celsius Malam hari sekalipun masih terasa panas dengan suhu mencapai derajat celsius Ditambah dengan udara bercampur debu pasir halus dapat mengganggu pernapasan Menghadapi cuaca yang

RAW TEXT	KOREKSI MANUAL
cuaca yang relatif ekstrim seperti itu jemaah haji Indonesia dihimbau selalu menjaga diri dari halhal yang dapat mengurangi kesehatan fisik maupun psikis Hasil pantauan tim Media Center Haji MCH di Klinik Kesehatan Haji Indonesia KKHI di Madinah Sabtu sebagian pasien yang dirawat karena dehidrasi disorientasi dan kaki melepuh karena kepanasan Sebagian besar dari mereka karena kepanasan Bahkan kondisi tersebut bisa memicu kambuhnya penyakit bawaan dari Tanah Air kata Erwinsyah Erick salah seorang dokter jaga di KKHI Namun demikian dia mencoba memberikan beberapa tips sederhana agar jemaah haji dapat terhindar dari kondisi itu Pertama kata dia bahwa kemana pun jemaah haji pergi sebaiknya dia selalu membawa botol minum masker semprotan air dan kantong plastik Bila diluar ruangan minum air minimal satu gelas setiap jam dan semprot wajah setiap menit lanjut Erick Kedua bila masuk Masjid Nabawi masukan	relatif ekstrim seperti itu jemaah haji Indonesia dihimbau selalu menjaga diri dari hal-hal yang dapat mengurangi kesehatan fisik maupun psikis Hasil pantauan tim Media enter Haji MH di Klinik Kesehatan Haji Indonesia KKHI di Madinah Sabtu sebagian pasien yang dirawat karena dehidrasi disorientasi dan kaki melepuh karena kepanasan Sebagian besar dari mereka karena kepanasan Bahkan kondisi tersebut bisa memicu kambuhnya penyakit bawaan dari Tanah Air kata Erwinsyah Erick salah seorang dokter jaga di KKHI Namun demikian dia mencoba memberikan beberapa tips sederhana agar jemaah haji dapat terhindar dari kondisi itu Pertama kata dia bahwa kemana pun jemaah haji pergi sebaiknya dia selalu membawa botol minum masker semprotan air dan kantong plastik Bila diluar ruangan minum air minimal satu gelas setiap jam dan semprot wajah setiap menit lanjut Erick Kedua bila masuk Masjid Nabawi masukan sendal dalam plastik dan

RAW TEXT	KOREKSI MANUAL
sendal dalam plastik dan bawa masuk Letakkan dekat tempat jemaah shalat jangan ditinggal di luar atau tempat kotak sandal Hal itu bisa berpotensi lupa atau hilang Setelah berputarputar mencari sandal dan putus asa jemaah memutuskan pulang tanpa sandal melewati jalanan panas Hal itulah yang menyebabkan beberapa jemaah harus dilarikan ke KKHI untuk mendapat penanganan karena kulit kakinya melepuh Ketiga gunakan selalu masker bila bepergian Hindari juga kontak dengan unta dikhawatirkan terpapar virus MERS Hingga pagi ini jemaah haji telah mendarat di Madinah sebanyak Kloter mengangkut orang terdiri dari jemaah haji dan petugas kloter	bawa masuk Letakkan dekat tempat jemaah shalat jangan ditinggal di luar atau tempat kotak sandal Hal itu bisa berpotensi lupa atau hilang Setelah berputarputar mencari sandal dan putus asa jemaah memutuskan pulang tanpa sandal melewati jalanan panas Hal itulah yang menyebabkan beberapa jemaah harus dilarikan ke KKHI untuk mendapat penanganan karena kulit kakinya melepuh Ketiga gunakan selalu masker bila bepergian Hindari juga kontak dengan unta dikhawatirkan terpapar virus MERS Hingga pagi ini jemaah haji telah mendarat di Madinah sebanyak Kloter mengangkut orang terdiri dari jemaah haji dan petugas kloter
Timur Lenk penguasa Samarkand pernah dianggap sebagai penakluk dan raja yang paling ditakuti di dunia Menurut catatan para sejarawan Timur Lenk telah membunuh sekitar penduduk bumi	Timur Lenk penguasa Samarkand pernah dianggap sebagai penakluk dan raja yang paling ditakuti di dunia Menurut catatan para sejarawan Timur Lenk telah membunuh sekitar penduduk bumi

RAW TEXT	KOREKSI MANUAL
Tidak akan sembuh karena ketika terpasung dia tak bisa komunikasi dengan orang lain dan malah tertekan kata Djliteng Lalu harus bagaimana	Tidak akan sembuh karena ketika terpasung dia tak bisa komunikasi dengan orang lain dan malah tertekan kata Djliteng Lalu harus bagaimana
Terpidana korupsi yang juga mantan Anggota DPR M Nazaruddin memasuki Gedung KPK untuk menjalani pemeriksaan di Jakarta September ANTARA	Terpidana korupsi yang juga mantan Anggota DPR M Nazaruddin memasuki Gedung KPK untuk menjalani pemeriksaan di Jakarta September ANTARA
Ternyata Agan cuma sekecil ini dibandingkan sama yang lain	Ternyata Agan cuma sekecil ini dibandingkan sama yang lain

Data sampel pengujian Per-token

Posisi	Token	Koreksi
1	tim	tim
1	spe	spe
1	itbsc	itbsc
1	kembali	kembali
1	menjadi	menjadi

Posisi	Token	Koreksi
1	jawara	jawara
1	dalam	dalam
1	cerdas	cerdas
1	cermat	cermat
1	perminyakan	perminyakan
1	seasia	seasia
1	pasifik	pasifik
1	tim	tim
1	the	the
1	society	society
1	of	of
1	pretroleum	pretroleum
1	engineers	engineers
1	institut	institut
1	teknologi	teknologi

Posisi	Token	Koreksi
1	bandung	bandung
1	students	students
1	chapter	chapter
1	spe	spe
1	itbsc	itbsc
1	kembali	kembali
1	berhasil	berhasil
1	menyabet	menyabet
1	gelar	gelar
1	juara	juara
1	pertama	pertama
1	dalam	dalam
1	kompetisi	kompetisi
1	internasional	internasional
1	petrobowl	petrobowl

Posisi	Token	Koreksi
1	apogce	apoge
1	mengalah	mengalah
1	timtim	tim-tim
1	lainya	lainnya
1	baik	baik
1	dari	dari
1	indonesia	indonesia
1	maupun	maupun
1	dari	dari
1	berbagai	berbagai
1	negara	negara
1	tim	tim
1	spesc	spesc
1	itb	itb
1	tersebut	tersebut

Posisi	Token	Koreksi
1	terdiri	terdiri
1	dari	dari
1	muhammad	muhammad
1	iffan	iffan
1	hannanu	hannanu
1	chintya	chintya
1	rizkiaputri	rizkiaputri
1	dan	dan
1	arnold	arnold
1	rico	rico
1	novrianto	novrianto
1	yang	yang
1	ketiganya	ketiganya
1	berasal	berasal
1	dari	dari

Posisi	Token	Koreksi
1	jurusan	jurusan
1	teknik	teknik
1	perminyakan	perminyakan
1	kompetisi	kompetisi
1	petrobowl	petrobowl
1	merupakan	merupakan
1	salah	salah
1	satu	satu
1	mata	mata
1	kegiatan	kegiatan
1	dari	dari
1	apogce	apoge
1	asia	asia
1	pacific	pacific
1	oil	oil

Posisi	Token	Koreksi
1	gas	gas
1	conference	conference
1	and	and
1	exhibition	exhibition
1	apogce	apoge
1	tahun	tahun
1	ini	ini
1	diselenggarakan	diselenggarakan
1	atas	atas
1	kerjasama	kerjasama
1	antara	antara
1	society	society
1	of	of
1	petroleum	petroleum
1	engineers	engineers

LAMPIRAN C

Hasil Analisis Model 1

raw	posisi 0	posisi 1	posisi 2	posisi 3	posisi 4	posisi 5	posisi 6	posisi 7	posisi 8	posisi 9	hasil	bobot	akurasi
yg	pak	dah	koar	ini	ya	dan	elu	emang	cebong	ada	10	0	2,9
jokowi	presiden	menyopiri	pak	cebong	pres	ketawa	raja	kodok	kecebong	hina	10	0	
dm	email	alamat	spasi	silakan	pemenang	kirim	kuis	konfirmasi	format	beruntung	10	0	
ahok	purnama	menistakan	penista	petahana	gubernur	menista	pak	sandi	aho	pelapor	10	0	
tks	kerahasiaan	sakhi	keluhan	komplain	ayum	resi	mohon	hai	kejelasan	berlanggan	10	0	
infokan	kendala	komplain	keluhan	silahkan	menghubungi	silakan	mohon	uja	kasih	resi	10	0	
rp	miliar	juta	ribu	triliun	rupiah	uang	total	setara	per	bonus	4	0,6	

raw	posisi 0	posisi 1	posisi 2	posisi 3	posisi 4	posisi 5	posisi 6	posisi 7	posisi 8	posisi 9	hasil	bobot	akurasi
telkomsel	menukarkan	cek	kamu	saldo	pulsa	yuk	berlanggan an	gratis	ikutan	paket	10	0	
sebanyakbanyaknya	kuis	beruntung	saldo	berhadiah	hadiah	pemenang	ikutan	jawaban	kirim	spasi	10	0	
dibantu	kendala	menghubungi	komplain	keluhan	rekan	mohon	pengecekan	berkenan	terima	saran	10	0	
jakarta	gubernur	utara	baung	selatan	purnama	kawasan	kota	kebagusan	kemang	lembang	10	0	
surabaya	kota	wali	reporter	abidin	balai	buana	bandung	dulur	suara	bungkul	10	0	
utk	dan	untuk	ayo	d	mari	ini	berikan	beri	slah	pres	1	0,9	
nya	ya	dong	kalo	deh	pak	aja	kan	dah	ada	mau	10	0	
sby	ani	demokrat	negarawan	pidato	presiden	demokrasi	assalamualaikum	penyadapan	keadilan	pemimpin	10	0	
dr	prof	dekan	rektor	dosen	muhsin	emeritus	kemahasiswaan	kedokteran	fakultas	geografi	10	0	

raw	posisi 0	posisi 1	posisi 2	posisi 3	posisi 4	posisi 5	posisi 6	posisi 7	posisi 8	posisi 9	hasil	bobot	akurasi
ga	kalo	deh	aja	dah	emang	kayak	nih	banget	cuman	biar	10	0	
tweet	kuis	tagar	pemenang	jawaban	ikutan	komentar	pengumuman	ayo	kalian	akun	10	0	
lalin	tersendat	merayap	terpantau	padat	lancar	kepadatan	macet	arah	arus	tomang	10	0	
salman	raja	baginda	rombongan	arab	menyopiri	kedatangan	pangeran	sambut	kenegaraan	penyambutan	10	0	
gak	aja	kalo	deh	kayak	kan	emang	nih	banget	mau	tau	10	0	
sms	telepon	seluler	kirim	reguler	email	menghubungi	alamat	lacak	silahkan	isi	10	0	
tdk	slah	dan	gubris								10	0	
hp	alamat	seluler	kirim	telepon	dompet	folder	email	kamera	genggam	ketik	10	0	
sdh	pak	ini	dan	ada	di	uda	dah	tuk	d	ya	10	0	
km	tol	kilometer	macet	arah	ruas	lajur	padat	gerbang	merak	kemacetan	10	0	

raw	posisi 0	posisi 1	posisi 2	posisi 3	posisi 4	posisi 5	posisi 6	posisi 7	posisi 8	posisi 9	hasil	bobot	akurasi
dki	gubernur	petahana	pemenangan	purnama	urut	putaran	kampanye	debat	hidayat	cuti	10	0	
dgn	dan	ayo	d	ini	yuk	mari	untuk	cek	kalian	tagar	10	0	
digunakan	menggunakan	alat	pembuatan	memanfaatkan	keperluan	memungkinkan	otomatis	peralatan	kegunaan	penggunaan	10	0	
bro	deh	kayak	kalo	abis	biar	nih	dah	emang	taun	cakep	10	0	
jl	prapatan	jantan	pegangsaan	lenteng	perempatan	duren	raya	menteng	bubutan	cempaka	10	0	
xxx	menyutradarai	tara	sakhi	membintangi	gaga	bafta	dandi	drama	aktris	pertunangan	10	0	
udah	nih	aja	kalo	banget	deh	dong	buat	biar	kayak	emang	10	0	
sidoarjo	porong	waru	kawan	gempol	menur	magersari	jambangan	tadi	totok	kedung	10	0	
jln	pertigaan	macet	perempatan	belok	jalan	tersendat	lobang	arah	imbas	lajur	4	0,6	
the	to	on	are	has	we	be	were	his	had	not	10	0	

raw	posisi 0	posisi 1	posisi 2	posisi 3	posisi 4	posisi 5	posisi 6	posisi 7	posisi 8	posisi 9	hasil	bobot	akurasi
aktivasi	menukarkan	berlangganan	pulsa	paket	akses	modem	menukar	genggaman	sakhi	koneksi	10	0	
ditunggu	beruntung	kuis	pemenang	tunggu	pengumuman	kalian	ikutan	jawaban	besok	buruan	10	0	
united	ibra	manajer	arsenal	setan	ham	paul	brom	payet	bek	sir	10	0	
cc	n	pak	dan	ini	l	moga	lah	di	gara	d	10	0	
wib	pukul	sabtu	rabu	kamis	selasa	minggu	jumat	senin	sore	maret	10	0	
tokopedia	diskon	belanja	retur	cicilan	pulsa	beli	order	gratis	saldo	buruan	10	0	
bl	mania	mumpung	gajian	buruan	diskon	sahur	dong	biar	belanja	perlengkapan	10	0	
pilkada	pencoblosan	pemilih	petahana	kampanye	rekapitulasi	debat	mencoblos	golput	serentak	putaran	10	0	
krn	dan	bego	pak	ada	gubris	ini	di	lah	slah	pres	10	0	
dlm	perumus	berkeadilan	mari	sarikan	sobat	ralat	dan	pres	beri	nonkonvensional	10	0	

raw	posisi 0	posisi 1	posisi 2	posisi 3	posisi 4	posisi 5	posisi 6	posisi 7	posisi 8	posisi 9	hasil	bobot	akurasi
internet	koneksi	pengguna	aplikasi	mengakses	layanan	platform	fiber	jaringan	digital	browser	10	0	
city	stadium	arsenal	primer	blues	man	hart	liga	manajer	fa	brom	10	0	
jg	d	dan	dong	pak	dah	buat	tau	min	ketemu	kalo	10	0	
tp	dah	uda	kalo	pak	gubris	lah	emang	gara	dan	cuman	10	0	
manchester	ibra	arsenal	manajer	hart	paul	setan	primer	liga	kane	stadium	10	0	
dialami	menderita	jarang	fatal	kendala	faktor	berakibat	gangguan	kesalahan	seseorang	depresi	10	0	
graparai	menukarkan	sakhi	kiran	penukaran	g	menukar	halofit	cek	pulsa	berlanggananan	10	0	
kpk	korupsi	suap	pemberantasan	penyidik	saut	kasus	penyidikan	komisi	penyuap	menjerat	10	0	
dg	dan	pres	beri	berikan	n	masyakat	bincang	beker	moga	mar	10	0	

raw	posisi 0	posisi 1	posisi 2	posisi 3	posisi 4	posisi 5	posisi 6	posisi 7	posisi 8	posisi 9	hasil	bobot	akurasi
promo	diskon	pulsa	belanja	beli	gratis	buruan	cicilan	kehabisan	kode	kamu	10	0	
timnas	u	skuat	beruji	pemusatan	pemain	pelatih	naturalisasi	latihan	piala	merumput	10	0	
jd	emang	pak	dah	kalo	gara	eh	n	pres	gubris	aja	10	0	
online	ojek	gojek	taksi	konvensional	uber	angkutan	pengemudi	daring	pengguna	pemesanan	10	0	
pastikan	supaya	lupa	khawatir	sebaiknya	tenang	jangan	pilih	konfirmasi	perlengkapan	pantau	10	0	
sy	pak	pres	pa	dah	gubris	slah	gara	uda	dan	salut	10	0	
dicek	mengecek	menghubungi	pengececkan	tertera	tersimpan	berkirim	alamat	pengirim	hilang	konfirmasi	10	0	
tl	jantan	merambat	perempatan	pertigaan	macet	lingkar	seksi	prapatan	arah	merayap	10	0	
anies	sandi	urut	debat	agus	pemenangan	kampanye	putaran	petahana	pengusung	berkampanye	10	0	

raw	posisi 0	posisi 1	posisi 2	posisi 3	posisi 4	posisi 5	posisi 6	posisi 7	posisi 8	posisi 9	hasil	bobot	akurasi
live	menyiarkan	sesaat	kongkow	susunan	menayangkan	ketinggalan	cuplikan	talk	hot	radio	10	0	
dilakukan	melakukan	proses	adanya	hal	secara	beberapa	upaya	langkah	terhadap	pihak	10	0	
barcelona	real	leo	betis	menyanjung	bek	porto	agregat	liga	merengkuh	pepe	10	0	
diberikan	kepada	mendapat	pemberian	memberikan	apresiasi	memberi	penghargaan	mendapatkan	menerima	memperoleh	10	0	
fc	tandang	bek	striker	skuat	penggawa	kontra	laga	kekalahan	porto	klub	10	0	
gb	ram	obor	telpon	modem	bonus	prosesor	kebangetan	g	tablet	memori	10	0	
ri	wakil	menteri	menghadiri	kehormatan	ketua	rapat	perwakilan	pimpinan	dewan	sekretaris	10	0	
ektp	korupsi	dakwaan	kasus	pengusutan	suap	menyeret	duit	megaproyek	ijon	kejanggalan	10	0	
ikuti	ayo	yuk	ikutan	kuis	blog	cek	menarik	hadiah	lengkap	berhadiah	10	0	

raw	posisi 0	posisi 1	posisi 2	posisi 3	posisi 4	posisi 5	posisi 6	posisi 7	posisi 8	posisi 9	hasil	bobot	akurasi
its	almamater	kampus	kolaborator	gerigi	juang	selasar	mahasiswa	rek	integralistik	safaat	10	0	
bukalapak	belanja	diskon	buruan	gampang	kamu	lapak	cicilan	retur	ikutan	mania	10	0	
gresik	tuban	semen	petrokimia	delta	kedung	pakis	rembang	gempol	bubutan	sampang	10	0	
blm	di	resi	d	min	antri	sekalian	dah	ya	ada	uda	10	0	
update	terbaru	aplikasi	rangkum	info	kemarin	versi	unduh	hilang	rilis	mengunduh	10	0	
bali	dewata	wisata	kuta	badung	wisatawan	hotel	berlibur	berwisata	pulau	destinasi	10	0	
in	to	are	we	on	has	his	be	were	not	had	10	0	
flash	sale	modem	diskon	kece	oppo	mob	plus	buruan	spesial	ram	10	0	
juventus	porto	roma	dinamo	protagonis	rival	klub	liga	kiper	bek	gelandang	10	0	
of	to	are	has	on	his	be	we	not	were	he	10	0	

raw	posisi 0	posisi 1	posisi 2	posisi 3	posisi 4	posisi 5	posisi 6	posisi 7	posisi 8	posisi 9	hasil	bobot	akurasi
gunakan	menggunakan	kode	pilih	cara	tanpa	hemat	filter	pakai	gampang	perlengkapan	10	0	
bs	d	dan	order	berdering	uda	min	tau	pa	slah	dah	10	0	
dpr	fraksi	hamzah	komisi	ketua	dewan	anggota	pimpinan	paripurna	angket	usulan	10	0	
pln	tegangan	listrik	trafo	rayon	gardu	pemadaman	pembangkit	watt	padam	supervisor	10	0	
jgn	ya	lupa	jangan	ini	dan	mending	d	kalian	kelewatan	biar	2	0,8	
dpt	saran	slah	dan	bantu	komplain	pengaduan	beri	sekalian	pemilih	subsidi	10	0	
klo	dah	emang	kalo	uda	gue	lagian	cuman	ente	pak	eh	10	0	
lg	dah	kalo	aja	nih	ya	bentar	kece	sebentar	ampe	ketemu	10	0	
twitter	akun	tagar	media	sosial	komentar	mengunggah	foto	edit	berkicau	konten	10	0	

raw	posisi 0	posisi 1	posisi 2	posisi 3	posisi 4	posisi 5	posisi 6	posisi 7	posisi 8	posisi 9	hasil	bobot	akurasi
lakukan	sebaiknya	rutin	bantu	berhemat	stres	perlu	berolahraga	sebelum	kebiasaan	bercinta	10	0	
freeport	divestasi	konsentrat	arbitrase	renegosiasi	saham	pencatut	berunding	pencatutan	pertambangan	taipan	10	0	
vs	susunan	skor	duel	laga	pertandingan	kontra	imbang	stadium	lawan	melawan	10	0	
chelsea	arsenal	hart	blues	primer	kane	stadium	brom	liga	manajer	samir	10	0	
restart	pencet	dah	abis	folder	browser	mah	bentar	palinga n	tombol	log	10	0	
liverpool	arsenal	hart	stadium	primer	kane	blues	brom	gol	liga	tandang	10	0	
madrid	real	betis	pepe	protagonis	los	oblak	leo	agregat	liga	porto	10	0	
pd	demokrat	beri	keadilan	jujur	pres	partisipasi	konstruktif	perumus	berdedikasi	hidu	10	0	

Hasil Analisis Model 8

raw	posisi 0	posisi 1	posisi 2	posisi 3	posisi 4	posisi 5	posisi 6	posisi 7	posisi 8	posisi 9	hasil	bobot	akurasi
yg	yang	bila	jika	kalaupun	apabila	meski	sehingga	meskipun	karena	lantaran	0	1	25
jokowi	korsel	ambigu	ratas	saya	sentilan	penghinaan	bain	wakil	ketegasan	bapa	10	0	
dm	dadah	email	eksim	e	wasir	belu	absolut	idiot	nagara	pencucian	10	0	
ahok	sandi	dia	pan	haris	ia	petahana	provokatif	prapera dilan	agus	wari	10	0	
tks	o	bulbul	biara	pemenuhan	raras	puak	eksponen	kiasan	baskara	mempertentangkan	10	0	
infokan	memberitahukan	sita	konfirmasi	memboroskan	mengecek	menginformasikan	mengendurkan	menaruh	balik	mencadangkan	0	1	

raw	posisi 0	posisi 1	posisi 2	posisi 3	posisi 4	posisi 5	posisi 6	posisi 7	posisi 8	posisi 9	hasil	bobot	akurasi
rp	rupiah	pencakar	kubik	semesteran	banderol	aji	bulanan	tebusan	yen	sepeser	10	0	
telkomsel	piatu	kearifan	maskapai	tablig	menghemat	tip	saga	bohlam	seduh	hilir	10	0	
sebanyak banyaknya	sayang	terima	kol	lalu	tertolong	berterima	tender	mengundi	berlalu	berkemampuan	10	0	
dibantu	menertibkan	mempermudah	memantau	menolong	melakukan	berpesan	mengunjungi	menahan	menghindari	menyelesaikan	10	0	
jakarta	kemang	aula	kutub	ufuk	medan	kuningan	solok	sebelah	bogor	rawa	10	0	
surabaya	solo	malang	bandung	medan	jember	jombang	sampang	sana	singkawang	indonesia	10	0	
utk	untuk	tuk	demi	guna	usai	sebelum	agar	sekaligus	sambil	setelah	0	1	
nya	lo	dong	emang	ya	aja	dah	bobok	bacaan	bingit	kalo	10	0	
sby	saya	cendana	ani	mega	ratas	dia	budi	pesing	membatik	kukang	10	0	

raw	posisi 0	posisi 1	posisi 2	posisi 3	posisi 4	posisi 5	posisi 6	posisi 7	posisi 8	posisi 9	hasil	bobot	akurasi
dr	dari	rektor	dekan	medik	geohidrologi	ketu	humas	abdu	madani	mawon	0	1	
ga	tak	enggak	tidak	kagak	dah	emang	kalo	belum	tanpa	papi	0	1	
tweet	tulisan	tayangan	kemunculan	alkitab	tanda	buku	belatung	wawasan	kurusan	persamaan	10	0	
lalin	lintas	lalim	terpantau	kokas	mede	bubutan	bruk	sedingin	geneng	lamela	10	0	
salman	ampat	kodok	sal	bagindа	pengajar	kristus	penukaran	abang	generasi	pengusutan	10	0	
gak	enggak	tak	tidak	kagak	belum	banget	emang	kalo	papi	sebentar	0	1	
sms	email	telepon	telpon	kerabat	langsung	kuta	permata	transkrip	imperial	mengadukan	10	0	
tdk	tak	tidak	enggak	tanpa	jangan	kagak	belum	kiranya	lum	barangkali	1	0,9	
hp	telepon	alamat	baterai	tablet	browser	oppo	laptop	perangkat	kamera	aplikasi	10	0	

raw	posisi 0	posisi 1	posisi 2	posisi 3	posisi 4	posisi 5	posisi 6	posisi 7	posisi 8	posisi 9	hasil	bobot	akurasi
sdh	uda	dah	setelah	tuk	sebelum	sudah	bakal	usai	ekskavasi	untuk	5	0,5	
km	kilometer	mil	meter	gerigi	kapal	merak	realisasi	sentimeter	membentang	kolaborator	0	1	
dki	jabar	serentak	pusat	selatan	impas	kemalasan	teluk	naim	pembelotan	reformasi	10	0	
dgn	dengan	tuk	demi	tanpa	terhadap	dalam	lewat	n	buat	dan	0	1	
digunakan	menghasilkan	memanfaatkan	menyediakan	tersedia	membeli	berfungsi	dijual	memproduksi	menciptakan	menggunakan	10	0	
bro	kak	om	kalo	geladi	sortir	bisul	layung	salur	bang	mumet	10	0	
jl	jalan	tugu	wisma	puruk	gang	apotik	pis	meruwat	galur	auditorium	0	1	
xxx	bafta	bawah	urut	urutan	lapangan	blok	lingkungan	ampat	sana	wilayah	10	0	
udah	deh	dah	nih	dong	sudah	banget	bakalan	ya	aja	mending	4	0,6	

raw	posisi 0	posisi 1	posisi 2	posisi 3	posisi 4	posisi 5	posisi 6	posisi 7	posisi 8	posisi 9	hasil	bobot	akurasi
sidoarjo	subang	malang	bandung	jombang	sampang	dupak	bogor	legok	wreda	pendopo	10	0	
jln	jalan	jalanan	nyepi	masjid	kebun	arena	jalur	labirin	tugu	pis	0	1	
the	welut	lala	tipar	kancil	his	war	minor	instrumental	memorandum	memble	10	0	
aktivasi	mengaktifkan	menonaktifkan	memecahkan	muri	puting	pemecahan	segmentasi	matahari	pengaktifan	perpindahan	10	0	
ditunggu	bersabar	tunggu	menunggu	tungu	kebagian	menyita	kemuraan	beruntung	menghadirkan	menyisakan	10	0	
united	cendekia	kerukunan	diameter	degeneratif	terentang	rohaniwan	penyingkiran	unjuk	dirgantara	wereng	10	0	
cc	alamak	bualan	colong	manjakani	emang	purnajual	carik	spakbor	retoris	tropi	10	0	
wib	was	degeneratif	persero	nonstop	terpantau	tergelak	selebihnya	laden	cendekia	kronis	10	0	

raw	posisi 0	posisi 1	posisi 2	posisi 3	posisi 4	posisi 5	posisi 6	posisi 7	posisi 8	posisi 9	hasil	bobot	akurasi
tokopedia	toko	sela	order	situ	sana	lapak	diskon	penjual	pasaran	belanja	10	0	
bl	sampean	sampeyan	pembekalan	piket	amien	ting	tertembus	q	beasiswa	pascaperang	10	0	
pilkada	pemilihan	pencoblosan	kampanye	gubernur	pengusungan	pemilih	pencalonan	payet	petahana	inaugurasi	10	0	
krn	karena	lantaran	jika	meskipun	bila	apabila	sehingga	kalo	meski	kalaupun	0	1	
dlm	dalam	usai	jelang	untuk	terhadap	pada	demi	tuk	sebelum	melalui	0	1	
internet	aplikasi	jaringan	konten	iklan	teknologi	televisi	modem	komputer	mengunduh	browser	10	0	
city	cendekia	rohaniwan	kerukunan	terentang	degeneratif	penyingkiran	diameter	persero	wereng	dirgantara	10	0	
jg	tuk	kalo	eh	loh	usai	kelincahan	lum	tubrukan	y	jika	10	0	

raw	posisi 0	posisi 1	posisi 2	posisi 3	posisi 4	posisi 5	posisi 6	posisi 7	posisi 8	posisi 9	hasil	bobot	akurasi
tp	meskipun	walau	padahal	dan	lantaran	tapi	kalo	eh	meski	gara	5	0,5	
manchester	man	bermotor	gandaria	volume	berwajib	bajakan	obral	penumpukan	sun	ingkar	10	0	
dialami	terjadi	bersangkutan	dimaksud	menimpa	berlangsung	ada	melanda	mengalami	berujung	menggambarkan	10	0	
grapari	gerai	kota	punggung	bioskop	panti	kafe	mal	stadion	diskotek	selatan	10	0	
kpk	polisi	penyidik	persidangan	penyidikan	penuntutan	pengusutan	kejaksaan	ma	supervisi	suap	10	0	
dg	dengan	tuk	demi	dan	n	tanpa	terhadap	serta	tunaaksa	lewat	0	1	
promo	diskon	potongan	pulsa	buruan	penawaran	loh	hadiah	gratis	tiket	kejutan	10	0	
timnas	garuda	paviliun	republik	aron	beruji	berpaspor	menghargakan	membidani	sineas	pos	10	0	

raw	posisi 0	posisi 1	posisi 2	posisi 3	posisi 4	posisi 5	posisi 6	posisi 7	posisi 8	posisi 9	hasil	bobot	akurasi
jd	jadi	menjadi	sebagai	selaku	memper	menyandang	mengasapi	ama	tongkol	kusuk	0	1	
online	konvensional	daring	gratis	digital	khusus	mandiri	sosial	uber	keseluruhan	mengglobal	10	0	
pastikan	imbau	jamin	ajak	anggap	minta	menegaskan	menyadari	menurut	menilai	tunggu	10	0	
sy	saya	dia	kami	ipung	gue	tuk	sekeping	y	ku	dadak	0	1	
dicek	mengecek	mengetahui	periksa	teratasi	mengunjungi	memantau	merasakan	melihat	menunggu	terlacak	10	0	
tl	perempatan	belokan	meramb at	pertigaan	merabat	dahi	topangan	pendapa	dum	belok	10	0	
anies	anis	novel	bolak	putar	perambban	kate	damai	berlainan	uni	metro	0	1	
live	darmabakti	net	duh	kemenangan	mikron	semburat	blakblakan	selipan	ampe	timpal	10	0	

raw	posisi 0	posisi 1	posisi 2	posisi 3	posisi 4	posisi 5	posisi 6	posisi 7	posisi 8	posisi 9	hasil	bobot	akurasi
dilakukan	berlangsung	melakukan	terjadi	mengikuti	membedakan	menghasilkan	melaksanakan	menjalani	menggelar	dimaksud	10	0	
barcelona	porto	bhayangkara	bayangan	vaksin	pertembakauan	yasmin	arsenal	aral	lumrah	ela	10	0	
diberikan	memberikan	memberi	memperoleh	berikan	menawarkan	menghasilkan	membutuhkan	memiliki	mendapatkan	menyediakan	10	0	
fc	pincang	pembudidaya	brom	penerang	veteran	aktivitas	kisruh	bakau	navigator	penerbang	10	0	
gb	obor	nitrogen	perakitan	kawalan	universal	diraja	hitungan	per	epidemiologi	olah	10	0	
ri	rapat	indonesia	gedung	sareh	tatapan	instruksi	sorong	paripurna	kantor	stan	10	0	
ektp	dugaan	pengusutan	tersandung	suap	pura	teguh	sertifikat	air	abang	urukan	10	0	
ikuti	ikutan	pantau	mengikuti	intip	lihat	simak	tonton	adakan	cek	hapus	10	0	

raw	posisi 0	posisi 1	posisi 2	posisi 3	posisi 4	posisi 5	posisi 6	posisi 7	posisi 8	posisi 9	hasil	bobot	akurasi
its	himpunan	resimen	kampus	selasar	jurusan	open	pengabdian	rahmat	globalisasi	magang	10	0	
bukalapak	sana	situ	lapak	diskon	belanja	indonesia	pasaran	bidang	sela	iklan	10	0	
gresik	malang	bandung	tegal	bogor	solok	garut	padang	areal	jombang	gersik	10	0	
blm	belum	tak	tidak	enggak	sudah	akhirnya	kapan	kagak	pinangan	suda	0	1	
update	pantau	mendiskusikan	mencermati	rilis	menebak	terbaru	memperbarui	mengumumkan	ikutan	menghapus	10	0	
bali	sana	lombok	situ	kediaman	hotel	sulut	bawah	kawasan	lokasi	dunia	10	0	
in	to	be	kebolehan	tepukan	on	blues	merenda	gulma	uis	aura	10	0	
flash	daur	kunyah	perancangan	kaji	kencang	eco	menonaktifkan	hidroponik	pemantauan	repot	10	0	

raw	posisi 0	posisi 1	posisi 2	posisi 3	posisi 4	posisi 5	posisi 6	posisi 7	posisi 8	posisi 9	hasil	bobot	akurasi
juventus	arsenal	mu	kedigdayaan	zenit	sepakan	viola	kegelapan	mereka	ibra	kompatriot	10	0	
of	blues	on	to	eviden	has	winter	wiyata	merelakan	gama	dragon	10	0	
gunakan	pakai	menggunakan	bawa	memakai	pilih	punya	menemukan	memanfaatkan	memilih	membeli	10	0	
bs	bisa	dapat	mampu	usah	sanggup	mau	perlu	akan	boleh	becus	0	1	
dpr	parlemen	dewan	fraksi	formatur	ma	demokrat	ting	kursi	inseminasi	pan	10	0	
pln	listrik	pemadaman	megawatt	kelistrikan	gas	uap	pasokan	pembangkit	distribusi	pelanggan	10	0	
jgn	jangan	tak	asbes	aurat	tidak	berpendirian	mengaum	sampurna	pantang	dah	0	1	
dpt	dapat	bisa	mampu	perlu	akan	patut	mari	sanggup	harus	siap	0	1	
klo	kalo	dah	emang	uda	kalau	biar	kok	karena	ama	eh	4	0,6	

raw	posisi 0	posisi 1	posisi 2	posisi 3	posisi 4	posisi 5	posisi 6	posisi 7	posisi 8	posisi 9	hasil	bobot	akurasi
lg	oppo	laptop	modem	kamera	y	pemadatan	intel	ama	eh	proyektor	10	0	
twitter	laman	akun	konten	telepon	tagar	iklan	situs	media	pilang	foto	10	0	
lakukan	melakukan	menemukan	alami	mengikuti	merasakan	terjadi	menjalani	membuat	merencanakan	mengadakan	10	0	
freeport	renegosiasi	dirgantara	semen	penambangan	divestasi	negosiasi	garuda	pupuk	serbuan	fi	10	0	
vs	kontra	versus	melawan	lawan	tekuk	memodifikasi	puji	jelang	tuding	tantang	1	0,9	
chelsea	arsenal	mu	porto	lawakan	tandang	keonaran	sepakan	ibra	ofensif	keperakan	10	0	
restart	jemawa	tipe	penanggulangan	spektrum	riak	terlacak	mandala	sangka	kunyah	puspita	10	0	
liverpool	porto	bhayangkara	arsenal	vaksin	bayangan	pertembakauan	seblang	tandang	lumrah	yasmin	10	0	
madrid	betis	gum	el	ihsan	bunuh	bulus	ibra	amina	prasarana	kepulauan	10	0	

raw	posisi 0	posisi 1	posisi 2	posisi 3	posisi 4	posisi 5	posisi 6	posisi 7	posisi 8	posisi 9	hasil	bobot	akurasi
pd	pada	tuk	jelang	terhadap	dalam	kemalingan	menuhankan	usai	melalui	kaukus	0	1	

Hasil Uji Seribu Data Model 8

token	correction	posisi	skor_w2c	skor_lev	skor_jarko	bobot	keterangan
international	internasional	0.00	1.00	0.923	0.976	97.49%	3
waspadai	waspada	0.00	1.00	0.875	0.987	96.56%	2
antrian	antrean	0.00	1.00	0.857	0.943	95.00%	1
karna	karena	0.00	1.00	0.833	0.961	94.86%	1
maruf	makruf	0.00	1.00	0.833	0.956	94.72%	1
shalat	salat	0.00	1.00	0.833	0.950	94.58%	1
smoga	semoga	0.00	1.00	0.833	0.950	94.58%	1
manfaatkan	memanfaatkan	0.00	1.00	0.833	0.949	94.56%	2

token	correction	posisi	skor_w2c	skor_lev	skor_jarko	bobot	keterangan
nggak	enggak	0.00	1.00	0.833	0.944	94.44%	1
menangkan	memenangkan	0.00	1.00	0.818	0.950	94.21%	2
lanjutkan	melanjutkan	0.00	1.00	0.818	0.939	93.94%	2
luncurkan	meluncurkan	0.00	1.00	0.818	0.939	93.94%	2
anies	anis	0.00	1.00	0.800	0.953	93.83%	1
truck	truk	0.00	1.00	0.800	0.953	93.83%	1
social	sosial	0.00	1.00	0.833	0.911	93.61%	1
adzan	azan	0.00	1.00	0.800	0.940	93.50%	1
china	cina	0.00	1.00	0.800	0.940	93.50%	1
laporkan	melaporkan	0.00	1.00	0.800	0.933	93.33%	2
ujan	hujan	0.00	1.00	0.800	0.933	93.33%	2
wujudkan	mewujudkan	0.00	1.00	0.800	0.933	93.33%	2
nelpon	telpon	0.00	1.00	0.833	0.889	93.06%	4
temen	teman	0.00	1.00	0.800	0.907	92.67%	1
umroh	umrah	0.00	1.00	0.800	0.907	92.67%	1

token	correction	posisi	skor_w2c	skor_lev	skor_jarko	bobot	keterangan
lakukan	melakukan	0.00	1.00	0.778	0.926	92.59%	2
rayakan	merayakan	0.00	1.00	0.778	0.926	92.59%	2
pemenangnya	pemenang	0.00	1.00	0.727	0.975	92.56%	2
penyebabnya	penyebab	0.00	1.00	0.727	0.975	92.56%	2
tutorialnya	tutorial	0.00	1.00	0.727	0.975	92.56%	2
mall	mal	0.00	1.00	0.750	0.942	92.29%	3
toll	tol	0.00	1.00	0.750	0.942	92.29%	1
ramaikan	meramaikan	0.00	1.00	0.800	0.892	92.29%	2
resmikan	meresmikan	0.00	1.00	0.800	0.892	92.29%	2
klo	kalo	0.00	1.00	0.750	0.925	91.88%	4
dri	dari	0.00	1.00	0.750	0.925	91.88%	1
lengkapnya	lengkap	0.00	1.00	0.700	0.970	91.75%	2
nonton	menonton	0.00	1.00	0.750	0.917	91.67%	2
nunggu	menunggu	0.00	1.00	0.750	0.917	91.67%	2
dilaporkan	melaporkan	0.00	1.00	0.800	0.867	91.67%	2

token	correction	posisi	skor_w2c	skor_lev	skor_jarko	bobot	keterangan
nanya	tanya	0.00	1.00	0.800	0.867	91.67%	1
perhatikan	memperhatikan	0.00	1.00	0.769	0.890	91.47%	2
perpanjang	memperpanjang	0.00	1.00	0.769	0.890	91.47%	2
harusnya	seharusnya	0.00	1.00	0.800	0.850	91.25%	2
dapatkan	mendapatkan	0.00	1.00	0.727	0.909	90.91%	2
donk	dong	0.00	1.00	0.750	0.883	90.83%	1
perpanjang	perpanjangan	1.00	0.90	0.833	0.991	90.60%	2
nikmati	menikmati	0.00	1.00	0.778	0.831	90.21%	2
diperpanjang	memperpanjang	0.00	1.00	0.769	0.834	90.09%	2
telp	telpon	0.00	1.00	0.667	0.933	90.00%	4
perbaiki	memperbaiki	0.00	1.00	0.727	0.867	89.87%	2
narkoba	narkotika	0.00	1.00	0.667	0.921	89.68%	0
sholat	solat	1.00	0.90	0.833	0.950	89.58%	1
adlh	adalah	0.00	1.00	0.667	0.911	89.44%	1
ok	oke	0.00	1.00	0.667	0.911	89.44%	1

token	correction	posisi	skor_w2c	skor_lev	skor_jarko	bobot	keterangan
yaa	ya	0.00	1.00	0.667	0.911	89.44%	1
yah	ya	0.00	1.00	0.667	0.911	89.44%	1
waktunya	waktu	0.00	1.00	0.625	0.938	89.06%	2
ikuti	ikutan	0.00	1.00	0.667	0.893	89.00%	2
sampe	sampai	0.00	1.00	0.667	0.893	89.00%	1
tegaskan	menegaskan	0.00	1.00	0.700	0.858	88.96%	2
gini	begini	0.00	1.00	0.667	0.889	88.89%	2
ngga	enggak	0.00	1.00	0.667	0.889	88.89%	1
smua	semua	1.00	0.90	0.800	0.940	88.50%	1
sampe	ampe	1.00	0.90	0.800	0.933	88.33%	4
smoga	moga	1.00	0.90	0.800	0.933	88.33%	1
diprediksi	memprediksi	0.00	1.00	0.727	0.801	88.20%	2
informasikan	menginformasikan	0.00	1.00	0.750	0.778	88.19%	2
lalin	lalim	1.00	0.90	0.800	0.920	88.00%	0
kembalikan	mengembalikan	0.00	1.00	0.692	0.827	87.98%	2

token	correction	posisi	skor_w2c	skor_lev	skor_jarko	bobot	keterangan
targetkan	menargetkan	0.00	1.00	0.727	0.789	87.90%	2
sholat	salat	0.00	1.00	0.667	0.840	87.67%	1
login	log	0.00	1.00	0.600	0.907	87.67%	0
datangi	mendatangi	0.00	1.00	0.700	0.805	87.62%	2
dipanggil	panggil	1.00	0.90	0.778	0.926	87.59%	2
mba	mbak	1.00	0.90	0.750	0.942	87.29%	1
diaktifkan	mengaktifkan	0.00	1.00	0.667	0.822	87.22%	2
sepakbola	persepakbolaan	0.00	1.00	0.643	0.844	87.17%	2
anaknya	anak	0.00	1.00	0.571	0.914	87.14%	2
caranya	cara	0.00	1.00	0.571	0.914	87.14%	2
infonya	info	0.00	1.00	0.571	0.914	87.14%	2
namanya	nama	0.00	1.00	0.571	0.914	87.14%	2
resinya	resi	0.00	1.00	0.571	0.914	87.14%	2
satunya	satu	0.00	1.00	0.571	0.914	87.14%	2
tinggalkan	meninggalkan	0.00	1.00	0.750	0.735	87.13%	2

token	correction	posisi	skor_w2c	skor_lev	skor_jarko	bobot	keterangan
tingkatkan	meningkatkan	0.00	1.00	0.750	0.735	87.13%	2
selamatkan	menyelamatkan	0.00	1.00	0.692	0.790	87.06%	2
diberikan	memberikan	0.00	1.00	0.700	0.778	86.96%	2
sampaikan	menyampaikan	0.00	1.00	0.667	0.810	86.92%	2
miliki	memiliki	1.00	0.90	0.750	0.925	86.88%	2
kalahkan	mengalahkan	0.00	1.00	0.636	0.837	86.84%	2
saksikan	menyaksikan	0.00	1.00	0.636	0.837	86.84%	2
serahkan	menyerahkan	0.00	1.00	0.636	0.837	86.84%	2
inilah	itulah	0.00	1.00	0.667	0.800	86.67%	1
tdk	tak	0.00	1.00	0.667	0.800	86.67%	1
ditunggu	tunggu	1.00	0.90	0.750	0.917	86.67%	2
gini	ini	1.00	0.90	0.750	0.917	86.67%	2
udah	dah	1.00	0.90	0.750	0.917	86.67%	1
hadapi	menghadapi	0.00	1.00	0.600	0.867	86.67%	2
hadiri	menghadiri	0.00	1.00	0.600	0.867	86.67%	2

token	correction	posisi	skor_w2c	skor_lev	skor_jarko	bobot	keterangan
one	monel	0.00	1.00	0.600	0.867	86.67%	0

Contoh Hasil Pengujian Sampel berdasarkan Threshold 90

posisi	**tokens**	**manual**	**90**		
			sistem	**stat**	**ket**
1	tim	tim	tim	1	ada di kamus
1	spe	spe	spe	0	tidak ada di kamus
1	itbsc	itbsc	itbsc	0	tidak ada di kamus
1	kembali	kembali	kembali	1	ada di kamus
1	menjadi	menjadi	menjadi	1	ada di kamus
1	jawara	jawara	jawara	1	ada di kamus
1	dalam	dalam	dalam	1	ada di kamus
1	cerdas	cerdas	cerdas	1	ada di kamus

posisi	tokens	manual	90		
			sistem	stat	ket
1	cermat	cermat	cermat	1	ada di kamus
1	perminyakan	perminyakan	perminyakan	1	ada di kamus
1	seasia	seasia	seasia	0	tidak ada di kamus
1	pasifik	pasifik	pasifik	1	ada di kamus
1	tim	tim	tim	1	ada di kamus
1	the	the	the	0	tidak ada di kamus
1	society	society	society	0	tidak ada di kamus
1	of	of	of	0	tidak ada di kamus
1	pretroleum	pretroleum	pretroleum	0	tidak ada di kamus
1	engineers	engineers	engineers	0	tidak ada di kamus
1	institut	institut	institut	1	ada di kamus
1	teknologi	teknologi	teknologi	1	ada di kamus
1	bandung	bandung	bandung	1	ada di kamus
1	students	students	students	0	tidak ada di kamus

posisi	tokens	manual	90		
			sistem	stat	ket
1	chapter	chapter	chapter	0	tidak ada di kamus
1	spe	spe	spe	0	tidak ada di kamus
1	itbsc	itbsc	itbsc	0	tidak ada di kamus
1	kembali	kembali	kembali	1	ada di kamus
1	berhasil	berhasil	berhasil	1	ada di kamus
1	menyabet	menyabet	menyabet	1	ada di kamus
1	gelar	gelar	gelar	1	ada di kamus
1	juara	juara	juara	1	ada di kamus
1	pertama	pertama	pertama	1	ada di kamus
1	dalam	dalam	dalam	1	ada di kamus
1	kompetisi	kompetisi	kompetisi	1	ada di kamus
1	internasional	internasional	internasional	1	ada di kamus
1	petrobowl	petrobowl	petrobowl	0	tidak ada di kamus

posisi	tokens	manual	90		
			sistem	stat	ket
1	apogce	apoge	apogce	0	gagal dan tidak sesuai actual
1	mengalah	mengalah	mengalah	1	ada di kamus
1	timtim	tim-tim	timtim	0	gagal dan tidak sesuai actual
1	lainya	lainnya	lainya	0	gagal dan tidak sesuai actual
1	baik	baik	baik	1	ada di kamus
1	dari	dari	dari	1	ada di kamus
1	indonesia	indonesia	indonesia	1	ada di kamus
1	maupun	maupun	mau	1	termaping
1	dari	dari	dari	1	ada di kamus
1	berbagai	berbagai	berbagai	1	ada di kamus
1	negara	negara	negara	1	ada di kamus
1	tim	tim	tim	1	ada di kamus

posisi	tokens	manual	90		
			sistem	stat	ket
1	spesc	spesc	spesc	0	tidak ada di kamus

Contoh Hasil Pengujian Sampel berdasarkan Threshold 85

posisi	tokens	manual	85		
			sistem	stat	ket
1	tim	tim	tim	1	ada di kamus
1	spe	spe	spe	0	tidak ada di kamus
1	itbsc	itbsc	itbsc	0	tidak ada di kamus
1	kembali	kembali	kembali	1	ada di kamus
1	menjadi	menjadi	menjadi	1	ada di kamus
1	jawara	jawara	jawara	1	ada di kamus
1	dalam	dalam	dalam	1	ada di kamus

posisi	tokens	manual	85		
			sistem	stat	ket
1	cerdas	cerdas	cerdas	1	ada di kamus
1	cermat	cermat	cermat	1	ada di kamus
1	perminyakan	perminyakan	perminyakan	1	ada di kamus
1	seasia	seasia	seasia	0	tidak ada di kamus
1	pasifik	pasifik	pasifik	1	ada di kamus
1	tim	tim	tim	1	ada di kamus
1	the	the	the	0	tidak ada di kamus
1	society	society	society	0	tidak ada di kamus
1	of	of	of	0	tidak ada di kamus
1	pretroleum	pretroleum	pretroleum	0	tidak ada di kamus
1	engineers	engineers	engineers	0	tidak ada di kamus
1	institut	institut	institut	1	ada di kamus
1	teknologi	teknologi	teknologi	1	ada di kamus
1	bandung	bandung	bandung	1	ada di kamus

posisi	tokens	manual	85		
			sistem	stat	ket
1	students	students	students	0	tidak ada di kamus
1	chapter	chapter	chapter	0	tidak ada di kamus
1	spe	spe	spe	0	tidak ada di kamus
1	itbsc	itbsc	itbsc	0	tidak ada di kamus
1	kembali	kembali	kembali	1	ada di kamus
1	berhasil	berhasil	berhasil	1	ada di kamus
1	menyabet	menyabet	menyabet	1	ada di kamus
1	gelar	gelar	gelar	1	ada di kamus
1	juara	juara	juara	1	ada di kamus
1	pertama	pertama	pertama	1	ada di kamus
1	dalam	dalam	dalam	1	ada di kamus
1	kompetisi	kompetisi	kompetisi	1	ada di kamus
1	internasional	internasional	internasional	1	ada di kamus
1	petrobowl	petrobowl	petrobowl	0	tidak ada di kamus

posisi	tokens	manual	85		
			sistem	stat	ket
1	apogce	apoge	apogce	0	gagal dan tidak sesuai actual
1	mengalah	mengalah	mengalah	1	ada di kamus
1	timtim	tim-tim	timtim	0	gagal dan tidak sesuai actual
1	lainya	lainnya	lainya	0	gagal dan tidak sesuai actual
1	baik	baik	baik	1	ada di kamus
1	dari	dari	dari	1	ada di kamus
1	indonesia	indonesia	indonesia	1	ada di kamus
1	maupun	maupun	mau	1	termaping
1	dari	dari	dari	1	ada di kamus
1	berbagai	berbagai	berbagai	1	ada di kamus
1	negara	negara	negara	1	ada di kamus
1	tim	tim	tim	1	ada di kamus

posisi	tokens	manual	85		
			sistem	stat	ket
1	spesc	spesc	spesc	0	tidak ada di kamus

Contoh Hasil Pengujian Sampel berdasarkan Threshold 80

posisi	tokens	manual	80		
			sistem	stat	ket
1	tim	tim	tim	1	ada di kamus
1	spe	spe	spe	0	tidak ada di kamus
1	itbsc	itbsc	itbsc	0	tidak ada di kamus
1	kembali	kembali	kembali	1	ada di kamus
1	menjadi	menjadi	menjadi	1	ada di kamus
1	jawara	jawara	jawara	1	ada di kamus
1	dalam	dalam	dalam	1	ada di kamus

posisi	tokens	manual	80		
			sistem	stat	ket
1	cerdas	cerdas	cerdas	1	ada di kamus
1	cermat	cermat	cermat	1	ada di kamus
1	perminyakan	perminyakan	perminyakan	1	ada di kamus
1	seasia	seasia	seasia	0	tidak ada di kamus
1	pasifik	pasifik	pasifik	1	ada di kamus
1	tim	tim	tim	1	ada di kamus
1	the	the	the	0	tidak ada di kamus
1	society	society	society	0	tidak ada di kamus
1	of	of	of	0	tidak ada di kamus
1	pretroleum	pretroleum	pretroleum	0	tidak ada di kamus
1	engineers	engineers	engineers	0	tidak ada di kamus
1	institut	institut	institut	1	ada di kamus
1	teknologi	teknologi	teknologi	1	ada di kamus
1	bandung	bandung	bandung	1	ada di kamus

posisi	tokens	manual	80		
			sistem	stat	ket
1	students	students	students	0	tidak ada di kamus
1	chapter	chapter	chapter	0	tidak ada di kamus
1	spe	spe	spe	0	tidak ada di kamus
1	itbsc	itbsc	itbsc	0	tidak ada di kamus
1	kembali	kembali	kembali	1	ada di kamus
1	berhasil	berhasil	berhasil	1	ada di kamus
1	menyabet	menyabet	menyabet	1	ada di kamus
1	gelar	gelar	gelar	1	ada di kamus
1	juara	juara	juara	1	ada di kamus
1	pertama	pertama	pertama	1	ada di kamus
1	dalam	dalam	dalam	1	ada di kamus
1	kompetisi	kompetisi	kompetisi	1	ada di kamus
1	internasional	internasional	internasional	1	ada di kamus
1	petrobowl	petrobowl	petrobowl	0	tidak ada di kamus

posisi	tokens	manual	80		
			sistem	stat	ket
1	apogce	apoge	apogce	0	gagal dan tidak sesuai actual
1	mengalah	mengalah	mengalah	1	ada di kamus
1	timtim	tim-tim	timtim	0	gagal dan tidak sesuai actual
1	lainya	lainnya	lainya	0	gagal dan tidak sesuai actual
1	baik	baik	baik	1	ada di kamus
1	dari	dari	dari	1	ada di kamus
1	indonesia	indonesia	indonesia	1	ada di kamus
1	maupun	maupun	mau	1	termaping
1	dari	dari	dari	1	ada di kamus
1	berbagai	berbagai	berbagai	1	ada di kamus
1	negara	negara	negara	1	ada di kamus
1	tim	tim	tim	1	ada di kamus

posisi	tokens	manual	80		
			sistem	stat	ket
1	spesc	spesc	spesc	0	tidak ada di kamus

Contoh Hasil Pengujian Sampel berdasarkan Threshold 75

posisi	tokens	manual	75		
			sistem	stat	ket
1	tim	tim	tim	1	ada di kamus
1	spe	spe	spe	0	tidak ada di kamus
1	itbsc	itbsc	itbsc	0	tidak ada di kamus
1	kembali	kembali	kembali	1	ada di kamus
1	menjadi	menjadi	menjadi	1	ada di kamus
1	jawara	jawara	jawara	1	ada di kamus
1	dalam	dalam	dalam	1	ada di kamus

posisi	tokens	manual	75		
			sistem	stat	ket
1	cerdas	cerdas	cerdas	1	ada di kamus
1	cermat	cermat	cermat	1	ada di kamus
1	perminyakan	perminyakan	perminyakan	1	ada di kamus
1	seasia	seasia	seasia	0	tidak ada di kamus
1	pasifik	pasifik	pasifik	1	ada di kamus
1	tim	tim	tim	1	ada di kamus
1	the	the	the	0	tidak ada di kamus
1	society	society	society	0	tidak ada di kamus
1	of	of	of	0	tidak ada di kamus
1	pretroleum	pretroleum	pretroleum	0	tidak ada di kamus
1	engineers	engineers	engineers	0	tidak ada di kamus
1	institut	institut	institut	1	ada di kamus
1	teknologi	teknologi	teknologi	1	ada di kamus
1	bandung	bandung	bandung	1	ada di kamus

posisi	tokens	manual	75		
			sistem	stat	ket
1	students	students	students	0	tidak ada di kamus
1	chapter	chapter	chapter	0	tidak ada di kamus
1	spe	spe	spe	0	tidak ada di kamus
1	itbsc	itbsc	itbsc	0	tidak ada di kamus
1	kembali	kembali	kembali	1	ada di kamus
1	berhasil	berhasil	berhasil	1	ada di kamus
1	menyabet	menyabet	menyabet	1	ada di kamus
1	gelar	gelar	gelar	1	ada di kamus
1	juara	juara	juara	1	ada di kamus
1	pertama	pertama	pertama	1	ada di kamus
1	dalam	dalam	dalam	1	ada di kamus
1	kompetisi	kompetisi	kompetisi	1	ada di kamus
1	internasional	internasional	internasional	1	ada di kamus
1	petrobowl	petrobowl	petrobowl	0	tidak ada di kamus

posisi	tokens	manual	75		
			sistem	stat	ket
1	apogce	apoge	apogce	0	gagal dan tidak sesuai actual
1	mengalah	mengalah	mengalah	1	ada di kamus
1	timtim	tim-tim	timtim	0	gagal dan tidak sesuai actual
1	lainya	lainnya	iklankan	0	tidak tepat
1	baik	baik	baik	1	ada di kamus
1	dari	dari	dari	1	ada di kamus
1	indonesia	indonesia	indonesia	1	ada di kamus
1	maupun	maupun	mau	1	termaping
1	dari	dari	dari	1	ada di kamus
1	berbagai	berbagai	berbagai	1	ada di kamus
1	negara	negara	negara	1	ada di kamus
1	tim	tim	tim	1	ada di kamus
1	spesc	spesc	spesc	0	tidak ada di kamus

Contoh Hasil Pengujian Sampel berdasarkan Threshold 70

posisi	tokens	manual	70		
			sistem	stat	ket
1	tim	tim	tim	1	ada di kamus
1	spe	spe	perban	0	tidak tepat
1	itbsc	itbsc	itbsc	0	tidak ada di kamus
1	kembali	kembali	kembali	1	ada di kamus
1	menjadi	menjadi	menjadi	1	ada di kamus
1	jawara	jawara	jawara	1	ada di kamus
1	dalam	dalam	dalam	1	ada di kamus
1	cerdas	cerdas	cerdas	1	ada di kamus
1	cermat	cermat	cermat	1	ada di kamus
1	perminyakan	perminyakan	perminyakan	1	ada di kamus
1	seasia	seasia	diana	0	tidak tepat
1	pasifik	pasifik	pasifik	1	ada di kamus

posisi	tokens	manual	70		
			sistem	stat	ket
1	tim	tim	tim	1	ada di kamus
1	the	the	the	0	tidak ada di kamus
1	society	society	society	0	tidak ada di kamus
1	of	of	of	0	tidak ada di kamus
1	pretroleum	pretroleum	pretroleum	0	tidak ada di kamus
1	engineers	engineers	engineers	0	tidak ada di kamus
1	institut	institut	institut	1	ada di kamus
1	teknologi	teknologi	teknologi	1	ada di kamus
1	bandung	bandung	bandung	1	ada di kamus
1	students	students	students	0	tidak ada di kamus
1	chapter	chapter	puteri	0	tidak tepat
1	spe	spe	perban	0	tidak tepat
1	itbsc	itbsc	itbsc	0	tidak ada di kamus
1	kembali	kembali	kembali	1	ada di kamus

posisi	tokens	manual	70		
			sistem	stat	ket
1	berhasil	berhasil	berhasil	1	ada di kamus
1	menyabet	menyabet	menyabet	1	ada di kamus
1	gelar	gelar	gelar	1	ada di kamus
1	juara	juara	juara	1	ada di kamus
1	pertama	pertama	pertama	1	ada di kamus
1	dalam	dalam	dalam	1	ada di kamus
1	kompetisi	kompetisi	kompetisi	1	ada di kamus
1	internasional	internasional	internasional	1	ada di kamus
1	petrobowl	petrobowl	internasional	0	tidak tepat
1	apogce	apoge	apogce	0	gagal dan tidak sesuai actual
1	mengalah	mengalah	mengalah	1	ada di kamus
1	timtim	tim-tim	biota	0	tidak tepat
1	lainya	lainnya	lainnya	1	diprediksi sama

posisi	tokens	manual	70		
			sistem	stat	ket
1	baik	baik	baik	1	ada di kamus
1	dari	dari	dari	1	ada di kamus
1	indonesia	indonesia	indonesia	1	ada di kamus
1	maupun	maupun	mau	1	termaping
1	dari	dari	dari	1	ada di kamus
1	berbagai	berbagai	berbagai	1	ada di kamus
1	negara	negara	negara	1	ada di kamus
1	tim	tim	tim	1	ada di kamus
1	spesc	spesc	spesc	0	tidak ada di kamus

Contoh Hasil Pengujian Sampel berdasarkan Threshold 65

posisi	tokens	manual	65		
			sistem	stat	ket
1	tim	tim	tim	1	ada di kamus
1	spe	spe	petroleum	0	tidak tepat
1	itbsc	itbsc	institut	0	tidak tepat
1	kembali	kembali	kembali	1	ada di kamus
1	menjadi	menjadi	menjadi	1	ada di kamus
1	jawara	jawara	jawara	1	ada di kamus
1	dalam	dalam	dalam	1	ada di kamus
1	cerdas	cerdas	cerdas	1	ada di kamus
1	cermat	cermat	cermat	1	ada di kamus
1	perminyakan	perminyakan	perminyakan	1	ada di kamus
1	seasia	seasia	diana	0	tidak tepat
1	pasifik	pasifik	pasifik	1	ada di kamus
1	tim	tim	tim	1	ada di kamus

posisi	tokens	manual	65		
			sistem	stat	ket
1	the	the	the	0	tidak ada di kamus
1	society	society	society	0	tidak ada di kamus
1	of	of	of	0	tidak ada di kamus
1	pretroleum	pretroleum	pretroleum	0	tidak ada di kamus
1	engineers	engineers	alumni	0	tidak tepat
1	institut	institut	institut	1	ada di kamus
1	teknologi	teknologi	teknologi	1	ada di kamus
1	bandung	bandung	bandung	1	ada di kamus
1	students	students	students	0	tidak ada di kamus
1	chapter	chapter	puteri	0	tidak tepat
1	spe	spe	petroleum	0	tidak tepat
1	itbsc	itbsc	institut	0	tidak tepat
1	kembali	kembali	kembali	1	ada di kamus
1	berhasil	berhasil	berhasil	1	ada di kamus

posisi	tokens	manual	65		
			sistem	stat	ket
1	menyabet	menyabet	menyabet	1	ada di kamus
1	gelar	gelar	gelar	1	ada di kamus
1	juara	juara	juara	1	ada di kamus
1	pertama	pertama	pertama	1	ada di kamus
1	dalam	dalam	dalam	1	ada di kamus
1	kompetisi	kompetisi	kompetisi	1	ada di kamus
1	internasional	internasional	internasional	1	ada di kamus
1	petrobowl	petrobowl	internasional	0	tidak tepat
1	apogce	apoge	balon	0	tidak tepat
1	mengalah	mengalah	mengalah	1	ada di kamus
1	timtim	tim-tim	tim	1	diprediksi kata ulang
1	lainya	lainnya	lainnya	1	diprediksi sama
1	baik	baik	baik	1	ada di kamus

posisi	tokens	manual	65		
			sistem	stat	ket
1	dari	dari	dari	1	ada di kamus
1	indonesia	indonesia	indonesia	1	ada di kamus
1	maupun	maupun	mau	1	termaping
1	dari	dari	dari	1	ada di kamus
1	berbagai	berbagai	berbagai	1	ada di kamus
1	negara	negara	negara	1	ada di kamus
1	tim	tim	tim	1	ada di kamus
1	spesc	spesc	spesc	0	tidak ada di kamus