Journal of Information System, Graphics, Hospitality and Technology
Volume 03, Nomor 01, Maret 2021
P-ISSN : 2622-1594
E-ISSN : 2685-449X

Multilabel Text Classification Menggunakan SVM dan Doc2Vec Classification pada Dokumen Berita Bahasa Indonesia

Kristian Indradiarta Gunawan, Joan Santoso
Teknik Informatika, Institut Sains dan Teknologi Terpadu Surabaya
E-mail: kristian.indra.1412@gmail.com, joan@stts.edu

***Abstrak*—Seiring dengan berkembangnya informasi yang ada di sekitar dengan pesat, maka jenis informasi yang ada pun menjadi sangat bervariasi dan sangat banyak jumlahnya, dan akan semakin terus bertambah. Dengan kondisi tersebut, kita akan mengalami kesulitan untuk mengenali jenis dari informasi tersebut satu persatu. Oleh karena itu dengan adanya proses klasifikasi teks dan dokumen sangatlah membantu untuk memilah dan mengenali informasi-informasi apa saja yang ada, baik informasi yang lama maupun informasi yang baru dan belum pernah ditemui sebelumnya. Bertujuan untuk dapat mengidentifikasi dan mengklasifikasikan dokumen-dokumen berita dalam bahasa Indonesia ke dalam beberapa kategori sekaligus, maka dibuatlah sebuah penelitian berupa sistem untuk menangani klasifikasi dokumen teks dalam bahasa Indonesia. Sistem tersebut akan memproses berita-berita yang diberikan, dan kemudian akan memberikan 2 kategori yang paling mendekati terhadap isi dari berita tersebut. Sistem dibuat dengan menggunakan *Python*, memanfaatkan *Doc2Vec* untuk mengambil fitur dataset, dan SVM untuk melakukan klasifikasi terhadap banyak kelas. Dataset yang digunakan adalah kumpulan dokumen berupa berita-berita yang diperoleh dari CNN Indonesia tahun 2016-2017, dan terbagi dalam 5 kategori berita utama, yaitu: Politik, Ekonomi, Teknologi, Olahraga, dan Hiburan. Dikarenakan sedikitnya literatur untuk klasifikasi text dalam bahasa Indonesia, maka pada penelitian ini hanya menargetkan akurasi sebesar 70% saja. Namun dari hasil ujicoba, akurasi yang diperoleh melebihi 90%. Hasil prediksi untuk kelas dokumen pun memiliki tingkat keberhasilan yang tinggi. Dengan penggunaan dataset dan penanganan preprocessing yang tepat untuk dokumen bahasa Indonesia, maka hasil yang dicapai bisa lebih bagus dan akurat.**

***Kata Kunci*—Bahasa Indonesia, Doc2Vec, Klasifikasi Teks, Multilabel, SVM**

I. PENDAHULUAN

Setiap harinya informasi yang ada di sekitar berkembang dengan pesat. Dan perkembangan itu terjadi di berbagai segi, baik dari segi kualitas maupun kuantitas. Variasi informasinya pun selalu bertambah, sehingga akan selalu muncul jenis informasi yang baru yang mungkin tidak pernah ditemui sebelumya. Terlebih di era teknologi saat ini, persebaran informasi bisa terjadi dengan sangat cepat, sehingga informasi yang bisa didapatkan menjadi lebih banyak dan sangat variatif.

Dengan pesatnya perkembangan informasi, apabila terdapat informasi yang sangat banyak, akan menjadi sulit untuk mengenali jenis dari informasi tersebut satu persatu. Oleh karena itu dengan adanya proses klasifikasi teks sangatlah membantu untuk memilah dan mengenali informasi-informasi apa saja yang ada, baik informasi yang lama maupun informasi yang baru dan belum pernah ditemui sebelumnya.

Permodelan dari klasifikasi teks ada banyak, dan dipelajari lebih lanjut dalam NLP. Dalam hal pemilahan dokumen dengan jumlah yang sangat banyak, klasifikasi teks sangat membantu dengan memberikan label kepada tiap dokumen yang diproses. Namun terkadang satu label saja dapat membuat kesalahan informasi, dikarenakan terkadang ada beberapa label yang dapat diberikan untuk sebuah dokumen. Jika sebuah dokumen dapat memiliki beberapa label, maka dapat meningkatkan informasi label dokumen tersebut dan dalam proses pemilahannya dapat dimanfaatkan untuk lebih lanjut.

Untuk dapat melakukan klasifikasi dan memberikan beberapa kelas / kategori untuk sebuah dokumen, digunakanlah SVM sebagai classifier. Penggunaan SVM dengan tepat memungkinkan untuk dapat melakukan klasifikasi terhadap banyak kelas, karena itu dapat dimanfaatkan dalam pemberian multilabel pada sebuah dokumen.

Doc2Vec merupakan pengembangan lanjutan dari *Word2Vec*, apabila Word2Vec mengklasifikasi kata atau text, maka Doc2Vec berguna untuk mengklasifikasikan kumpulan kata / kalimat, dan umumnya disebut dengan dokumen. Doc2Vec Digunakan untuk menentukan nilai vector dari sebuah dokumen, dimana nilai tersebut dapat dimanfaatkan untuk kepentingan klasifikasi yang mana akan menggunakan SVM. Sehingga dengan perpaduan doc2vec dan SVM, dapat menghasilkan klasifikasi yang tepat dan jelas untuk proses klasifikasinya.

Untuk itulah dikembangkan sebuah sistem untuk menangani pelabelan dokumen sehingga kumpulan dokumen yang ada dapat dikelompokkan dan dimanfaatkan sesuai dengan kategorinya.Adapun tujuan dari pemanfaatan ini adalah untuk membantu memilah dokumen berita dengan cara memberikan beberapa label / class untuk sebuah

DOI : 10.37823/insight.v3i01.126

dokumen, dan kemudian mengelompokkan dokumen-dokumen tersebut berdasarkan labelnya untuk dapat dimanfaatkan lebih lanjut.

Penelitian yang dilakukan adalah klasifikasi multi-label terhadap kumpulan berita-berita dalam bahasa Indonesia yang menggunakan SVM dan memanfaatkan doc2vec sebagai feature extraction. Dengan mempertimbangkan paper yang ditulis oleh Nanak Chand[1] mengenai ketepatan SVM yang cukup tinggi yaitu di atas 90%, maka target *F1Score* pada pemanfaatan ini hanyalah sebesar 70%. Persentase tersebut lebih rendah dibanding hasil penelitian sebelumnya dikarenakan beberapa faktor, salah satunya adalah minimnya literasi klasifikasi dokumen untuk bahasa Indonesia, sehingga penulis perlu melakukan penelitian dan mengujinya terlebih dahulu.

Di dunia internasional, klasifikasi dokumen telah banyak dilakukan, tetapi sebagian besar dilakukan untuk dokumen Bahasa Inggris, dimana struktur kata dan kalimatnya berbeda dengan Bahasa Indonesia. Untuk saat ini penelitian klasifikasi dokumen pada bahasa Indonesia masih sangat terbatas dibandingkan dengan penelitian klasifikasi dokumen pada bahasa Inggris, sehingga hal tersebut mendorong penulis untuk melakukan penelitian dengan topik klasifikasi dokumen untuk Bahasa Indonesia.

II. Dasar Teori

Pada bagian ini akan dijelaskan mengenai teori-teori dasar yang melandasi pembuatan pemanfaatan pada Tesis ini. Adapun teori-teori yang akan dibahas meliputi *machine learning*, *Natural Language Processing*, *Support Vector Machine (SVM)*, *word2vec*, *doc2vec*, bahasa pemrograman *Python*, dan library-library Python yang akan digunakan.

A. Machine Learning

Machine learning adalah bagian dari Artificial Intelligence (AI) / Kecerdasan buatan yang memfasilitasi kemampuan sistem untuk belajar dan berkembang secara otomatis dari pengalaman tanpa perlu disuruh secara langsung. Machine learning berfokus pada pengembangan program komputer yang dapat mengakses data dan menggunakannya untuk belajar sendiri.

Proses pembelajaran dimulai dari observasi atau data, seperti contoh-contoh, pengalaman langsung, atau instruksi, dengan tujuan untuk mencari pola pada data dan membuat keputusan yang lebih baik di masa depan berdasarkan pada contoh-contoh yang telah diberikan. Tujuan utamanya adalah untuk membiarkan komputer belajar secara otomatis tanpa campur tangan atau bantuan manusia dan dapat menyesuaikan tindakan sesuai kebutuhan.

Algoritma klasik machine learning melakukan pendekatan berdasarkan analisa semantik yang meniru kemampuan manusia untuk memahami arti teks. Meskipun biasanya memberikan hasil yang lebih cepat dan lebih akurat untuk mengidentifikasi peluang yang menguntungkan atau risiko berbahaya, mungkin juga memerlukan waktu dan sumber daya tambahan untuk melatihnya dengan benar. Menggabungkan machine learning dengan AI dan teknologi kognitif dapat membuatnya lebih efektif dalam memproses informasi dalam jumlah yang besar.

B. Natural Language Processing

Natural Language Processing atau NLP adalah salah satu bidang dari *Artificial Intelligence* yang memberi mesin kemampuan untuk membaca, memahami, dan memperoleh makna dari bahasa manusia. NLP adalah disiplin yang berfokus pada interaksi antara *data science* dan bahasa manusia, dan berkembang ke banyak industri. Saat ini, NLP berkembang pesat berkat peningkatan besar dalam akses ke data dan peningkatan daya komputasi, yang memungkinkan praktisi mencapai hasil yang berarti di berbagai bidang seperti perawatan kesehatan, media, keuangan, dan sumber daya manusia.

Segala sesuatu yang diungkapkan (baik secara lisan maupun tertulis) mengandung banyak informasi. Topik yang dipilih, nada, pilihan kata-kata, semuanya menambahkan beberapa jenis informasi yang dapat ditafsirkan dan diambil nilainya. Secara teori, dapat dipahami dan bahkan diprediksi perilaku manusia dengan menggunakan informasi tersebut.

Tetapi permasalahannya adalah satu orang dapat menghasilkan ratusan atau ribuan kata dalam satu deklarasi, dimana tiap kalimatnya memiliki tingkat kompleksitas yang beragam. Jika ingin mengukur dan menganalisis beberapa ratus, ribuan, atau jutaan orang atau deklarasi dalam suatu geografi tertentu, maka situasinya terlalu besar dan tidak dapat dikelola dengan baik. Data yang dihasilkan dari percakapan adalah contoh data yang tidak terstruktur. Data tidak terstruktur tidak cocok dengan struktur baris dan kolom tradisional dari database relasional, dan mewakili sebagian besar data yang ada di dunia nyata, yaitu berantakan dan sulit untuk dimanipulasi.

Namun demikian, berkat kemajuan dalam disiplin ilmu seperti machine learning, revolusi besar sedang terjadi terkait topik ini. Sekarang ini bukan lagi tentang mencoba menafsirkan teks atau pidato berdasarkan kata kuncinya (cara mekanik kuno), tetapi tentang memahami makna di balik kata-kata itu (cara kognitif). Dengan cara ini memungkinkan untuk mendeteksi kiasan seperti ironi, atau bahkan melakukan *sentiment analysis*.

C. Support Vector Machine

SVM (*Support Vector Machine*) adalah permodelan dari supervised learning yang bertujuan untuk menganalisa data yang akan digunakan untuk klasifikasi dan analisa regresi. Dengan diberikan contoh data-data training, setiap data akan ditandai sebagai salah satu dari beberapa kategori / fitur yang diberikan. Dari hasil proses training dengan menggunakan algoritma training tertentu, terbentuklah sebuah model yang akan memasangkan contoh data baru kepada salah satu kategori yang telah dipelajari sebelumnya.

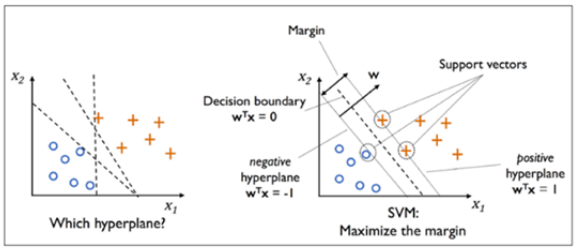
Dalam pemodelan klasifikasi, SVM memiliki konsep yang lebih matang dan lebih jelas secara matematis dibandingkan dengan teknik-teknik klasifikasi lainnya. SVM juga dapat mengatasi masalah klasifikasi dan regresi dengan *linear* maupun *non linear*. Pada dasarnya SVM digunakan untuk menentukan sebuah data termasuk dalam kelas yang mana dari hanya 2 kelas yang diberikan.

SVM digunakan untuk mencari *hyperplane* terbaik dengan memaksimalkan jarak antar kelas. *Hyperplane*

[1] Chand, Nanak, Preeti Mishra, C. Rama Krishna, Emmanuel Shubhakar Pilli, Mahesh Chandra Govil. *A Comparative Analysis of SVM and its Stacking with other Classification Algorithm for Intrusion Detection*.

 DOI : 10.37823/insight.v3i01.126

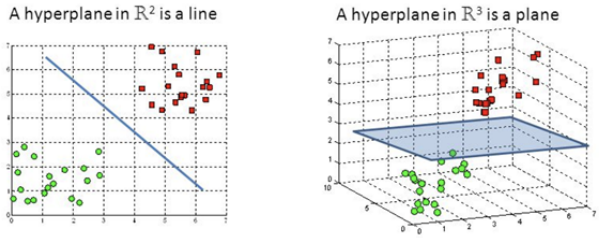
adalah sebuah fungsi yang dapat digunakan sebagai pemisah antar kelas. Dalam 2-D fungsi yang digunakan untuk klasifikasi antar kelas disebut sebagai *line whereas*, fungsi yang digunakan untuk klasifikasi antas kelas dalam 3-D disebut *plane similarly*, sedangan fungsi yang digunakan untuk klasifikasi di dalam ruang kelas dimensi yang lebih tinggi di sebut *hyperplane*.



Gambar. 1. Perbedaan Hyperplane pada bidang fitur 2D dan 3D

Hyperplane yang ditemukan SVM diilustrasikan seperti dapat dilihat pada Gambar 1, dimana posisinya berada ditengah-tengah antara dua kelas, artinya jarak antara hyperplane dengan objek-objek data berbeda dengan kelas yang berdekatan (terluar) yang diberi tanda bulat kosong dan positif. Dalam SVM objek data terluar yang paling dekat dengan *hyperplane* disebut *support vector*. Objek yang disebut support vector paling sulit diklasifikasikan dikarenakan posisi yang hampir tumpang tindih (*overlap*) dengan kelas lain. Mengingat sifatnya yang kritis, hanya *support vector* inilah yang diperhitungkan untuk menemukan hyperplane yang paling optimal oleh SVM.

Pada ilustrasi gambar 2, Hyperplane pada bidang 2 dimensi adalah sebuah garis yang memisahkan data-data yang ada. Sedangkan untuk 3-dimensi, hyperplane adalah sebuah bidang datar yang membagi kedua data, sehingga dapat dikenali perbedaan antara kedua data tersebut.

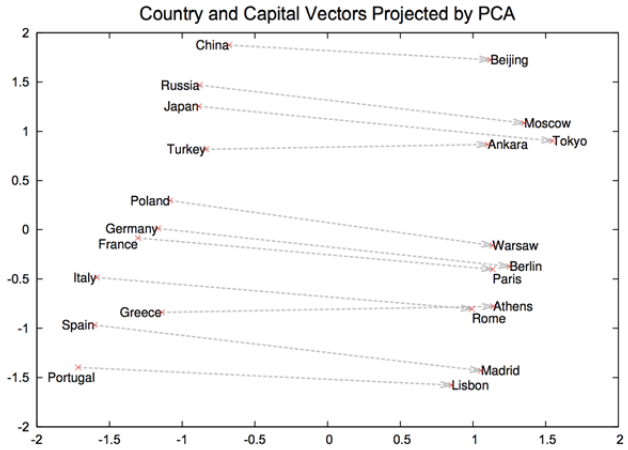


Gambar. 2. Perbedaan Hyperplane pada bidang fitur 2D dan 3D

D. Word2Vec

Word2vec adalah salah satu metode embedding word yang berguna untuk merepresentasikan kata menjadi sebuah vektor dengan panjang N[2]. Misalnya kata "**Indonesia**" di representasikan menjadi sebuah vektor dengan panjang 5 yaitu: [0.2, 0.4, -0.8, 0.9, -0.5]. Vektor tersebut tidak hanya merepresentasikan kata secara sintaktik tapi juga secara semantik atau secara makna.

Sebagai contoh, apabila word2vec dilatih menggunakan korpus yang cukup lengkap, maka vektor representasi dari kata "Indonesia" akan berdekatan dengan vektor "Jakarta" sebagaimana vektor "Perancis" akan berdekatan dengan vektor "Paris". Dengan kata lain, model word2vec akan memahami bahwa "Indonesia" dan "Jakarta memiliki hubungan yang sama dengan "Perancis" dan "Paris" yaitu negara dan ibukotanya.



Gambar. 3. Contoh penggunaan Word2vec untuk Negara dan Ibukota

Word2Vec menggunakan neural network untuk mendapatkan vektor tersebut. Arsitektur Word2vec hanya terdiri dari 3 layer yaitu Input, Projection *(Hidden Layer*), dan Output. Input pada Word2vec berbentuk *one-hot encoded vector* dengan panjang = jumlah kata unik pada data training. Terdapat 2 jenis arsitektur neural network dari Word2Vec yaitu "Skip-gram" dan "Continous Bag of Word" (CBOW).

E. Doc2Vec

Paragraph Vector (Doc2Vec) adalah pengembangan dari Word2Vec[3], dan target dari doc2vec adalah membuat representasi vektor dari sebuah dokumen. Jika kata-kata dalam kalimat memiliki struktur (grammar), sebuah dokumen tidak memiliki struktur yang logis. Untuk permasalahan ini, sebuah vektor lain (Paragraph ID) perlu ditambahkan agar cocok dengan permodelan word2vec. Hanya inilah perbedaan antara word2vec dengan doc2vec.

Dalam pengaplikasiannya, terdapat 2 pendekatan yang akan dilakukan oleh Doc2Vec, yaitu PV-DM *(Paragraph Vector-Distributed Memory*), dan DBOW (*Distributed Bag of Words*).

1) Paragraph Vector-Distributed Memory

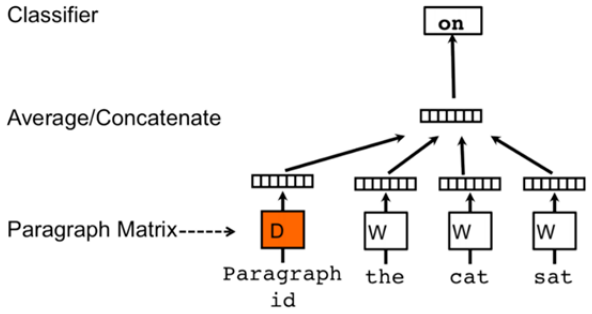
Ide utama dari PV-DM model terinspirasi dari *Continous Bag of Words* dari Word2Vec. Pada CBOW model dari Word2Vec, model belajar untuk memprediksi inti kata berdasarkan dari konteks. Sebagai contoh, jika diberikan kalimat "Kucing duduk di atas sofa", CBOW model akan belajar untuk memprediksi kata "duduk" saat diberikan konteks: Kucing, di atas, sofa. Mirip seperti itu, pada PV-DM, ide utamanya adalah: Mengambil sampel secara acak dari kata-kata yang berurutan dari paragraf, dan memprediksi inti kata dari sampel acak tersebut dengan mengambil id paragraf dan konteks kata-kata sebagai input.

[2] https://medium.com/@afrizalfir/mengenal-word2vec-af4758da6b5d, diakses tanggal 25 November 2020

[3] https://medium.com/wisio/a-gentle-introduction-to-doc2vec-db3e8c0cce5e, Diakses tanggal 25 November 2020

DOI : 10.37823/insight.v3i01.126

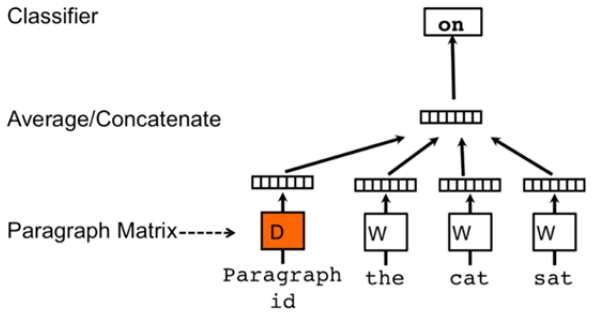
Dengan melihat pada model diagram PV-DM pada Gambar 4 akan memberikan sedikit pencerahan. Pada model tersebut, dapat terlihat matrix paragraph, Average / Concatenate, dan bagian classifier. Matrix Paragraf adalah matrix dimana tiap kolomnya mewakili vektor dari paragraf. Average/Concatenate memiliki arti entah vector kata dan paragraph termasuk averaged atau concatenated. Terakhir bagian *classifier* adalah lapisan vektor yang tersembunyi (salah satu dari *concatenated / averaged*) sebagai input dan memprediksi bagian tengah dari kata.



Gambar. 4. Model PV-DM

Ketika melakukan training untuk vektor kata W, vektor dokumen D juga ikut dilatih. Dan pada akhir training, vektor dokumen D menyimpan representasi angka dari dokumen tersebut. Model ini dinamakan PV-DM, karena bertindak seperti ingatan (memory) yang berusaha mengingat apa yang hilang dari konteks saat ini (topic dari paragraf). Jika vektor kata merepresentasikan konsep dari sebuah kata, maka vektor dokumen bertujuan untuk merepresentasikan konsep dari sebuah dokumen.

2) Distributed Bag of Words



Gambar. 5. Model DBOW

Model DBOW sedikit berbeda dari PV-DM model. DBOW model mengabaikan konteks kata-kata pada input, dan memaksakan model untuk memprediksi kata-kata secara acak dari paragraf *output*. Untuk contoh di atas, katakan model belajar dari memprediksi 2 kata sampel. Jadi, untuk mempelajari vektor dari dokumen, 2 kata diambil sebagai sampel dari {Kucing, duduk, di atas, sofa}.

Hanya terdapat 1 perbedaan antara *skip-gram* dan DBOW, yaitu DBOW mengambil ID dokumen (Paragraph ID) sebagai input, dan mencoba memprediksi berbagai contoh kata-kata acak dari dokumen tersebut.

F. Python

Python dibuat oleh programmer Belanda bernama Guido Van Rossum. Python adalah salah satu bahasa pemrograman yang dapat melakukan eksekusi sejumlah instruksi multi guna secara langsung (*interpretatif*) dengan metode orientasi objek (*Object Oriented Programming*) serta menggunakan semantik dinamis untuk memberikan tingkat keterbacaan *syntax*. Sebagian lain mengartikan Python sebagai bahasa yang kemampuan, menggabungkan kapabilitas, dan sintaksis kode yang sangat jelas, dan juga dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif.

Walaupun Python tergolong bahasa pemrograman dengan level tinggi, nyatanya Python dirancang sedemikian rupa agar mudah dipelajari dan dipahami. Python memiliki tata bahasa dan script yang sangat mudah untuk dipelajari. Python memiliki sistem pengelolaan data dan memori otomatis, modul yang selalu diperbarui, dan juga memiliki banyak fasilitas pendukung. Python banyak diaplikasikan pada berbagai sistem operasi seperti Linux, Microsoft Windows, Mac OS, Android, Symbian OS, Amiga, Palm dan lain-lain.

G. Python Library

Dalam penggunaannya, Python didukung oleh berbagai macam *library* yang dikembangkan oleh orang lain ataupun komunitas. Sebagian besar library / package dapat di-install dengan menggunakan 'pip'. PIP adalah *Standard Package Manager* untuk Python, yang membantu untuk melakukan instalasi dan mengatur *package* tambahan yang bukan merupakan dari *Python Standard Library*.

1) Gensim

Gensim adalah library dari Phyton yang digunakan untuk *topic modelling*, *document indexing*, dan *similiarity retrieval* dengan *corpora* yang besar. Target utamanya adalah untuk memfasilitasi bidang *Natural Language Processing* (NLP) dan *Information Retrieval* (IR). Gensim memiliki banyak fungsi yang dapat dimanfaatkan di bidang NLP, namun dalam proses pengerjaan Tesis ini, fitur yang digunakan dari library Gensim adalah *feature extraction* dengan menggunakan Doc2Vec

Untuk dapat menghasilkan *vector* dari sebuah dokumen, maka berikut adalah langkah-langkah yang perlu dilakukan:

1) Menyiapkan dataset yang sudah melalui preprocessing
2) Memberi label / tag pada dataset yang akan diproses
3) Membuat model Doc2Vec
4) Membangun kosakata / vocabulary
5) Melatih model Doc2Vec dengan dataset yang sudah diberi tag
6) Menguji model Doc2Vec

2) NLTK

NLTK (*Natural Language Toolkit*) adalah platform untuk membantu program Python agar dapat bekerja dengan data NLP. NLTK menyediakan interface yang mudah digunakan ke 50 lebih sumber korpora dan leksikal seperti WordNet, bersama dengan berbagai library pemrosesan teks yang digunakan untuk klasifikasi, tokenisasi, stemming, parsing, dan sebagainya.

NLTK memiliki dokumentasi lengkap yang memperkenalkan dasar-dasar pemrograman bersama topik

 DOI : 10.37823/insight.v3i01.126

dalam komputasi linguistik, ditambah dokumentasi API yang komprehensif. Karena itu NLTK cocok untuk digunakan oleh berbagai kalangan. NLTK tersedia untuk Windows, Mac OS X, dan Linux. Yang terbaik dari semuanya, NLTK adalah proyek gratis, *open source*, dan berbasis komunitas.

3) Sastrawi

Sastrawi adalah library Python sederhana yang memungkinkan untuk mereduksi kata-kata infleksi dalam Bahasa Indonesia (Bahasa Indonesia) menjadi bentuk dasarnya (stem). Library Sastrawi yang akan digunakan ini adalah percabangan dari proyek Sastrawi asli yang ditulis dalam bahasa PHP.

Terdapat 2 fungsi utama yang digunakan dari library sastrawi, yaitu stemming dan stopwords removal. Proses stemming menggunakan bantuan library Sastrawi karena Sastrawi sudah mendukung proses stemming untuk bahasa Indonesia. Begitu pula dengan stopwords removal, dimana Sastrawi sudah memiliki daftar stopwords dalam bahasa Indonesia yang bisa tinggal digunakan.

III. TINJAUAN PUSTAKA

Pada bagian ini akan meninjau beberapa paper yang dijadikan sebagai acuan dari pembuatan Tesis ini. Adapun judul-judul paper yang akan dibahas adalah sebagai berikut: Support Vector Machines for Multi-class Classification, A Comparative Analysis of SVM and its Stacking with other Classification Algorithm for Intrusion Detection, dan An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation.

A. Support Vector Machines for Multi-class Classification

Support Vector Machines (SVM) utamanya didesain untuk masalah klasifikasi dengan 2 kelas. Namun pada beberapa paper disebutkan bahwa kombinasi dari *n* SVM dapat digunakan untuk menyelesaikan permasalahan klasifikasi dengan *n* kelas, namun prosesnya membutuhkan perhatian yang khusus karena cukup kompleks. Pada paper ini, perbesaran masalah dari penggunaan beberapa SVM dibahas dan difokuskan terhadap permasalah tersebut. Beragam metode normalisasi diajukan untuk dapat menyelesaikan permasalahan dan efisiensinya diukur secara empirik.

Berbagai cara klasifikasi diusulkan dalam paper ini, salah satunya adalah stacking SVM dengan teknik klasifikasi lainnya. Di sisi lainnya, skema *one-per-class decomposition* digantikan oleh skema yang lebih bagus yang didasarkan pada *error-correcting codes*.

Semua eksperimen yang dilaporkan pada bagian ini berdasarkan pada dataset dari Machine Learning Repository di Irvine. Nilai yang tertera adalah persentase dari kesalahan klasifikasi, dirata-rata dari 10 eksperimen.

Pada paper ini, permasalahan dari normalisasi hasil keluaran dari beberapa SVM menjadi sorotan utama agar dapat dijadikan perbandingan. Teknik normalisasi yang berbeda diusulkan dan dilakukan eksperiman. Metode lain yang lebih bagus membuat penggunaan *binary classifier* untuk resolusi dari permasalahan klasifikasi multi-class. Eksperimen dari pendekatan-pendekatan tersebut dengan SVM maupun teknik pembelajaran lainnya adalah pekerjaan skala besar yang sedang dikerjakan dan akan diberikan pada versi akhir dari paper ini.

B. A Comparative Analysis of SVM and its Stacking with other Classification Algorithm for Intrusion Detection

Mendeteksi keanehan / *anomaly* adalah fokus utama dari paper ini. SVM adalah salah satu algoritma klasifikasi yang bagus dan diaplikasikan secara khusus untuk Intrusion Detection. Bagaimanapun performanya dapat ditingkatkan secara signifikan ketika diaplikasikan dengan metode klasifikasi lainnya. Dan pada paper ini dilakukan analisa perbandingan dari performa SVM ketika dilakukan stacking dengan algoritma classifier lainnya.

Paper ini melakukan berbagai eksperimen untuk menemukan kombinasi mana yang paling bagus ketika dipasangkan dengan SVM. Dataset yang digunakan adalah NSL-KDD yang tersedia untuk umum dan memiliki 11,850 record dan 42 atribut.

Sebagai kesimpulan, pendeteksian keanehan menggunakan teknik machine learning adalah salah satu dari area penelitian yang sedang berkembang. Kemampuan adaptasi dan pembelajaran dari algoritma machine learning cukup mengagumkan di mata para peneliti yang bekerja di bidang yang berbeda. Tujuan dari pembelajaran ini adalah untuk menemukan pasangan multi-classifier terbaik untuk mengalahkan classifier SVM. Ditemukan bahwa tidak semua classifier meningkatkan performa SVM. Stacking dari SVM dan Random Forest memberikan hasil yang terbaik karena Random Forest adalah classifier yang sangat bagus untuk melakukan tugas klasifikasi yang kompleks.

C. An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation

Doc2vec diajukan oleh Le dan Mikolov (2014) sebagai pengembangan dari word2vec untuk mempelajari *embedding* pada level dokumen. Meskipun hasil pada paper aslinya cukup menjanjikan, namun orang-orang lain berjuang untuk menghasilkan ulang hasil tersebut. Paper ini memberikan sebuah evaluasi emporik dari doc2vec untuk 2 tugas, dan menemukan bahwa doc2vec memiliki performa yang kuat ketika menggunakan model yang dilatih pada *external corpora* yang besar, dan dapat dikembangkan lebih jauh lagi dengan menggunakan word embedding sang telah dilatih sebelumnya.

Vektor paragraph (doc2vec) adalah pengembangan sederhana dari word2vec untuk mengembangkan pembelajaran dari embedding urutan kata demi kata. Pada paper ini akan digunakan istilah "*document embedding*" untuk mengacu pada penyematan dari urutan kata, tanpa memperhatikan *granularity*.

Doc2Vec diajukan dalam 2 bentuk, dbow (Distributed Bag-Of-Words) dan dmpv (Distributed Memory Paragraph Vector). Dbow merupakan model yang lebih sederhana dan mengabaikan urutan kata, sedangkan dmpv adalah model yang lebih kompleks dengan parameter lebih. Meskipun ditemukan bahwa metode dmpv yang berdiri sendiri adalah model yang lebih baik, orang-orang lainnya melaporkan hasil yang berlawanan. Doc2vec juga telah dilaporkan telah memproduksi performa *sub-par disbanding* dengan metode rata-rata vektor yang berdasarkan pada eksperimen informal.

 DOI : 10.37823/insight.v3i01.126

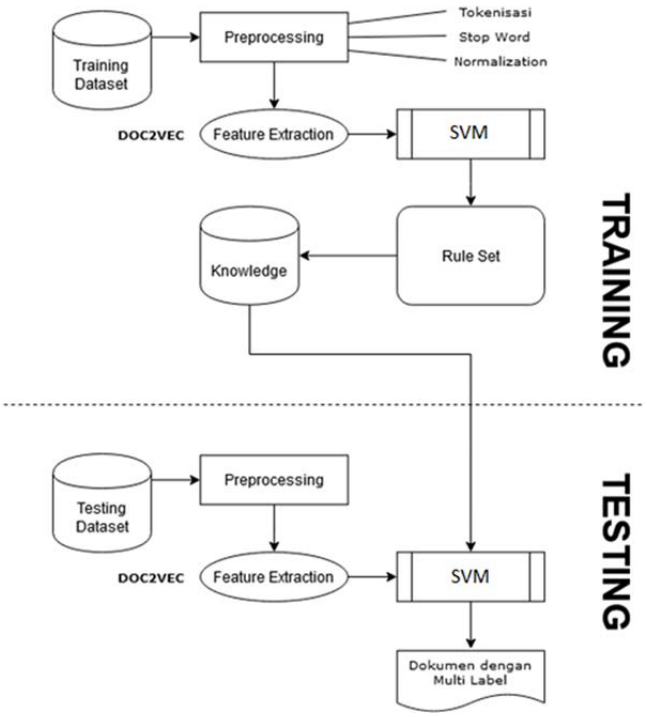
Dengan adanya ketidakjelasan mengenai efektifitas sesungguhnya dari doc2vec dan kebingungan mengenai perbedaan performa antara dbow dan dmpv, maka paper ini bertujuan untuk memberikan pencerahan terhadap beberapa pertanyaan empirik antara lain: 1) Seberapa efektif doc2vec pada pengaturan tugas yang berbeda-beda?; 2) Manakah yang lebih baik antara dmpv dan dbow?; 3) Apakah mungkin untuk memperbaiki doc2vec melalui optimasi *hyper-parameter* atau dengan word embedding yang telah dilatih sebelumnya?; dan 4) Dapatkan doc2ved digunakan sebagai metode alternative seperti word2vec?

Dalam paper ini dilakukan beberapa eksperimen dan membandingkan dengan metode-metode lainnya untuk menemukan pasangan mana yang paling bagus dan lebih optimal, sehingga dapat kita manfaatkan untuk kepentingan lainnya.

Dan sebagai kesimpulan, telah dilakukan 2 tugas untuk mengevaluasi kualitas dari document embedding yang dipelajari oleh doc2vec secara empirik, sebagaimana dibandingkan dengan 2 metode lainnya: word2vec word vector averaging dan model n-gram. Secara keseluruhan ditemukan bahwa dbow memiliki model yang lebih baik disbanding dmpv. Doc2vec juga memiliki performa yang kuat bahkan ketika dilatih menggunakan external corpora yang besar, dan memiliki keuntungan dari word embedding yang telah dilatih sebelumnya

IV. METODOLOGI PENELITIAN

Pada bab ini akan dibahas mengenai langkah-langkah penyelesaian penelitian dari sistem yang dibuat, meliputi pembahasan arsitektur sistem, pembahasan tiap proses yang dilakukan, beserta contoh data yang sedang diproses.



Gambar. 6. Blok Diagram

A. Arsitektur Sistem

Arsitektur sistem dari penelitian yang dibuat dapat dilihat pada Gambar 6. Dari gambar blok diagram tersebut, dapat dilihat secara garis besar bahwa proses implementasi ujicoba nantinya akan dibagi menjadi 2 bagian besar, yaitu bagian training dimana sistem akan mempelajari banyak dataset untuk dijadikan acuan pembetukan ruleset, kemudian bagian testing dimana hasil dari ruleset yang telah diperoleh akan diterapkan pada dokumen yang lain untuk menentukan klasifikasi dari dokumen tersebut.

Dalam proses pembuatannya, sistem dibuat dengan menggunakan bahasa pemrograman Phyton. Untuk membantu dalam proses-proses klasifikasi berikutnya maka akan digunakan library seperti Gensim, NLTK, Sastrawi, dan Sklearn. Semua library yang digunakan juga merupakan library yang dapat diakses dari bahasa Phyton.

Untuk tiap bagiannya secara garis besar akan dilakukan beberapa proses penting, yaitu adalah persiapan dataset, preprocessing, feature extraction, training dan testing menggunakan SVM. Tiap proses tersebut akan dibahas lebih lanjut pada bagian berikutnya.

B. Dataset

Tahap pertama dari serangkaian alur sistem yang dibuat adalah persiapan dataset. Dataset yang digunakan adalah sekumpulan dokumen berupa berita-berita dalam bentuk teks sejumlah 1000 dokumen yang terbagi dalam 5 kategori berita utama. Tiap dokumen berisi kata-kata dalam bahasa Indonesia dengan kisaran jumlah antara 300-500 kata per dokumennya. Dataset yang digunakan diperoleh dari CNN Indonesia tahun 2016-2017. Dataset tersebut telah dilabeli sebelumnya dengan salah satu dari 5 kategori, yaitu: ekonomi, hiburan, olahraga, politik, dan teknologi. 5 kategori ini nanti yang akan menjadi kelas dalam proses klasifikasi berikutnya.

Dari dataset yang berjumlah 1000 tersebut, sudah termasuk dataset dari 5 kelas dengan perbandingan yang rata, berarti terdapat 200 dataset dari masing-masing kelas pada kumpulan dataset tersebut.

C. Preprocessing

Sebagai langkah awal dari proses ini adalah tahap preprocessing, dimana tiap dataset akan disiapkan terlebih dahulu sebelum menerapkan feature extraction. Tujuan dari preprocessing terhadap dataset ini adalah untuk lebih menajamkan lagi dataset yang akan digunakan dengan cara menghapus hal-hal yang akan mengganggu nantinya sehingga hasil yang diperoleh bisa maksimal.

Pada pembuatan sistem ini akan digunakan 4 metode preprocessing, yaitu: *casefolding*, *stopwords removal*, *stemming*, dan *tokenization*. Secara garis besar, langkah yang dilakukan untuk tiap dokumennya adalah:

a. Membaca isi file dan menyimpannya ke dalam variabel
b. Melakukan proses preprocessing terhadap variabel tersebut
c. Menyimpan isi variabel ke dalam file kembali

Sehingga hasil akhir dari proses preprocessing ini adalah sekumpulan dataset yang telah melalui proses preprocessing, dan sudah siap untuk diproses ke tahapan berikutnya, yaitu feature extraction.

D. Feature Extraction (Doc2Vec)

Sampai pada tahap ini, semua dataset telah melalui proses preprocessing dan siap digunakan untuk proses-proses

 DOI : 10.37823/insight.v3i01.126

berikutnya. Feature extraction pada pembuatan sistem ini menggunakan doc2vec yang dibantu oleh library Gensim. Tujuan utama dari doc2vec ini adalah mendapatkan nilai vector sebagai perwakilan fitur dari dokumen yang diberikan. Dalam implementasinya bagian ini dibedakan menjadi 2 yaitu pelatihan model doc2vec dan eksekusi doc2vec. Dari setiap dataset yang diberikan akan diwakilkan fiturnya oleh 50 vector yang akan digunakan pada proses berikutnya.

E. SVM

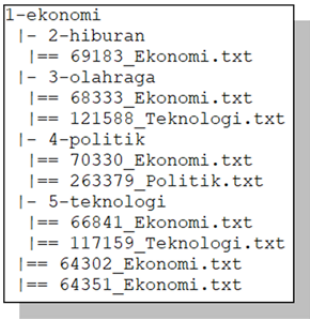
Setelah model doc2vec terbentuk, maka berikutnya adalah memasukkan fitur-fitur tersebut ke dalam SVM. Umumnya SVM digunakan hanya untuk memisahkan 2 kelas saja, antara true atau false. Untuk dapat melakukan klasifikasi terhadap 2 kelas atau lebih, maka perlu dilakukan training terpisah untuk tiap kelasnya. Namun pada prakteknya, SVM bisa melakukan training tersebut secara bersamaan dengan cara memasukkan semua dataset dan kelasnya secara bersamaan, dan kemudian dilakukan training terhadap dataset tersebut.

Setelah proses testing selesai, maka berikutnya mencetak hasil akurasi yang terdiri dari nilai *accuracy*, *precision*, *recall*, dan f1. Sampai pada tahap ini, SVM sudah selesai dilatih dan dapat memprediksi dataset baru yang akan diberikan.

F. Prediksi dan Klasifikasi

Setelah training SVM sudah selesai dilakukan, maka yang akan dilakukan berikutnya adalah melakukan prediksi untuk tiap dataset, seberapa dekat dataset tersebut dengan 5 kelas yang telah dilatihkan sebelumnya. Sebagai pembuktian, maka akan dilakukan prediksi ulang terhadap setiap dataset yang digunakan sebelumnya.

Dari hasil prediksi, maka akan diambil 2 kelas yang memiliki persentase tertinggi. Kemudian file dokumen dataset akan dipindahkan ke folder baru untuk memisahkan tiap dataset ke kelas hasil prediksinya. Contoh hasil akhir dari pemindahan dataset dapat dilihat Gambar 7.



Gambar. 7. Contoh Hasil Klasifikasi File Untuk Kelas Ekonomi

Pada gambar tersebut dapat dilihat pembagian dataset hasil klasifikasi untuk kelas ekonomi. Semua file yang berada pada folder ini berarti memiliki probabilitas tertinggi sebagai kelas ekonomi. File yang ada pada folder ini menandakan bahwa dataset tersebut hanya memiliki 1 kelas prediksi saja dan tidak memiliki prediksi untuk kelas ke-2. Tidak adanya prediksi kelas ke-2 bisa disebabkan karena rendahnya probabilitas kelas ke 2 yang berada di bawah threshold.

Nama file yang ada pada gambar tersebut menandakan kelas awal yang telah ditentukan untuk dataset tersebut. Sebagai contoh untuk dataset dengan nama 121588_Teknologi.txt, berarti awalnya dilabeli sebagai berita kelas teknologi. Tetapi setelah melalu proses klasifikasi, ternyata diidentifikasi sebagai berita kelas ekonomi, dan juga berita kelas olahraga. Hal tersebut bisa terjadi ketika terdapat kata-kata yang mirip digunakan pada kelas lain dan digunakan pada berita tersebut.

V. PEMBAHASAN HASIL UJI COBA

Pada bab ini akan dibahas mengenai proses ujicoba yang telah dilakukan. Adapun yang akan dibahas pada bab ini dimulai dari skenario ujicoba, hasil dari tiap skenario ujicoba beserta contoh dan hasil dari ujicobanya, analisa ujicoba, dan kesimpulan dari proses ujicoba yang telah dilakukan.

A. Skenario Ujicoba

Secara garis besar, proses ujicoba dilakukan sebanyak 4x dengan persentase dataset training dan testing yang berbeda-beda, seperti dapat dilihat pada tabel I. Untuk setiap ujicoba nanti akan dilakukan perhitungan terhadap ketepatan dari proses ujicoba tersebut.

TABEL I
SKENARIO UJICOBA

Uji Coba	Training (%)	Testing (%)
1	90	10
2	80	20
3	70	30
4	60	40

Semua ujicoba akan menggunakan dataset yang sama, hanya persentasenya saja yang berbeda. Dataset yang digunakan sejumlah 1000 dokumen, dimana didalanya terkandung 5 kelas dengan perbandingan yang sama besar. Berarti pada 1000 dokumen tersebut terdapat 200 dokumen dari masing-masing kelas. Kelima kelas yang digunakan adalah: ekonomi, hiburan, olahraga, politik, dan teknologi.

Disertakan juga contoh berita baru yang akan diujicobakan pada sistem yang telah dibuat, yaitu adalah berita tentang kemenangan Joe Bidden dalam pemilihan presiden di Amerika yang diambil dari kompas.com, dan berita tentang pendukung Khofifah dalam pilkada Surabaya yang diambil dari cnnindonesia.com, dan keduanya adalah berita politik. Adapun untuk keperluan ujicoba, berita tentang Joe Bidden akan diwakili dengan data A, dan berita tentang pendukung Khofifah akan diwakili dengan data B.

B. Hasil Ujicoba

Pada bagian ini akan dibahas mengenai hasil dari ujicoba yang telah dilakukan baik pada dataset yang telah disiapkan, maupun pada dokumen berita baru yang akan diujicobakan. Akan dijabarkan mengenai proporsi dataset, dan hasil perhitungan untuk ketepatan hasil dari ujicoba. Untuk dokumen berita baru, akan dituliskan juga persentase prediksi kelas dari dokumen tersebut. Berikut adalah hasil dari ujicoba ke-1 dan ke-4.

 DOI : 10.37823/insight.v3i01.126

1) Ujicoba 1

Pada ujicoba ke-1, perbandingan antara data training dan testing adalah 90:10 dari total sebanyak 1000 dataset. Hal tersebut berarti data yang digunakan untuk training adalah sebesar 180 untuk tiap kelasnya, sehingga total data yang digunakan untuk training adalah sebesar 900. Dan sisanya adalah 20 dataset dari tiap kelas dengan total 100 dataset akan digunakan sebagai testing.

TABEL II
HASIL PERHITUNGAN AKURASI PADA UJICOBA 1

Dataset	Accuracy	Precision	Recall	F1	Dokumen	% Dokumen
1000	0,9800	0,9828	0,9783	0,9800	31	3,1
	0,9900	0,9920	0,9900	0,9907	23	2,3
	0,9800	0,9843	0,9767	0,9799	31	3,1
	0,9900	0,9920	0,9900	0,9908	29	2,9
	0,9900	0,9923	0,9867	0,9892	22	2,2
Average	0,9860	0,9887	0,9843	0,9861	27,2	2,72

Pada Tabel II dapat dilihat hasil perhitungan akurasi dari ujicoba yang dilakukan. Hasil rata-rata perhitungan akurasi, precision, recall, dan F1score yang diperoleh cukup tinggi, yaitu dengan hasil rata-rata di atas 90%. Dari ujicoba ini terdapat kurang lebih 27 dokumen yang diidentifikasi memiliki multi-class.

TABEL III
HASIL PREDIKSI DATA BARU PADA UJICOBA 1

Iterasi	Prediksi Data A	Prediksi Data B
1	62.14% Politik & 31.40% Hiburan	95.00% Politik
2	63.06% Politik & 31.18% Hiburan	95.79% Politik
3	60.67% Politik & 33.19% Hiburan	95.74% Politik
4	61.00% Politik & 32.92% Hiburan	94.90% Politik
5	62.29% Politik & 31.74% Hiburan	95.47% Politik

Selain perhitungan akurasi, ujicoba juga melibatkan 2 data baru dan memprediksi kelas dari kedua data tersebut dengan menggunakan sistem yang telah dibangun. Dapat dilihat pada Tabel III bahwa hasil prediksi untuk data A adalah sekitar 60% untuk kelas politik, dan 30% untuk kelas hiburan. Sedangkan untuk data B diprediksi sekitar 95% adalah berita politik. Analisa dan pembahasan lebih lanjut mengenai hasil dari klasifikasi data baru ini akan dibahas pada bagian berikutnya.

2) Ujicoba 4

Pada ujicoba ke-4, perbandingan antara data training dan testing adalah 60:40 dari total sebanyak 1000 dataset. Hal tersebut berarti data yang digunakan untuk training adalah sebesar 120 untuk tiap kelasnya, sehingga total data yang digunakan untuk training adalah sebesar 600. Dan sisanya adalah 80 dataset dari tiap kelas dengan total 400 dataset akan digunakan sebagai testing.

TABEL IV
HASIL PERHITUNGAN AKURASI PADA UJICOBA 4

Dataset	Accuracy	Precision	Recall	F1	Dokumen	% Dokumen
1000	0,9675	0,9672	0,9670	0,9670	60	6
	0,9725	0,9724	0,9721	0,9722	75	7,5
	0,9700	0,9702	0,9697	0,9696	71	7,1
	0,9725	0,9727	0,9718	0,9722	62	6,2
	0,9650	0,9651	0,9646	0,9645	68	6,8
Average	0,9695	0,9695	0,9690	0,9691	67,2	6,72

Pada Tabel IV dapat dilihat hasil perhitungan akurasi dari ujicoba yang dilakukan. Hasil rata-rata perhitungan akurasi, precision, recall, dan F1score yang diperoleh cukup tinggi, yaitu dengan hasil rata-rata di atas 90%, kurang lebih sama dengan hasil perhitungan akurasi pada ujicoba 3. Dari ujicoba ini terdapat kurang lebih 67 dokumen yang diidentifikasi memiliki multi-class, cukup meningkat jika dibandingkan dengan hasil dari ujicoba 3.

TABEL V
HASIL PREDIKSI DATA BARU PADA UJICOBA 4

Iterasi	Prediksi Data A	Prediksi Data B
1	58.68% Hiburan & 17.20% Ekonomi	94.76% Politik
2	59.08% Hiburan & 17.31% Ekonomi	95.23% Politik
3	58.79% Hiburan & 16.89% Ekonomi	95.08% Politik
4	59.36% Hiburan & 17.02% Ekonomi	94.70% Politik

Selain perhitungan akurasi, ujicoba juga melibatkan 2 data baru dan memprediksi kelas dari kedua data tersebut dengan menggunakan sistem yang telah dibangun. Dapat dilihat pada Tabel V bahwa hasil prediksi untuk data A adalah sekitar 58% untuk kelas hiburan, dan 17% untuk kelas ekonomi. Sedangkan untuk data B diprediksi sekitar 94% adalah berita politik. Nilai prediksi untuk data A jauh berbeda jika dibandingkan dengan hasil dari ujicoba 3. Analisa dan pembahasan lebih lanjut mengenai hasil dari klasifikasi data baru ini akan dibahas pada bagian berikutnya.

C. Hasil Klasifikasi

Setelah semua dataset diklasifikasi sesuai kelasnya masing-masing, maka diperolehlah data seperti terlihat pada Tabel VI. Pada sebelah kiri adalah kelas pilihan pertama, dan sebelah atas adalah kelas pilihan kedua. Dari 1000 dataset yang diuji dan diklasifikasikan ulang, terdapat beberapa dataset yang ternyata lebih condong ke kelas lain daripada kelas awalnya, atau dataset tersebut memiliki kelas kedua yang berbeda dari kelas awalnya.

TABEL VI
PEMETAAN HASIL KLASIFIKASI

1 \ 2	Ekonomi	Hiburan	Olahraga	Politik	Teknologi	Subtotal
Ekonomi	193	1	0	4	3	201
Hiburan	0	200	1	0	0	201
Olahraga	1	1	196	1		199
Politik	3	1	0	196	1	201
Teknologi	5	3		1	189	198
Subtotal	202	206	197	202	193	1000

Data yang terlihat pada Tabel VII adalah dataset yang tidak sesuai dengan kelas awalnya. Kelas awal dari dataset tersebut malah menjadi kelas kedua, dan hasil prediksi menentukan kelas lain sebagai kelas utama pada dataset tersebut. Pada salah satu dataset tersebut dapat dilihat bahwa sebetulnya memang betul berita tersebut adalah berita teknologi, tetapi karena isi beritanya menggunakan kata-kata yang digunakan di kelas politik seperti: dpr, ite, pasal, ayat, revisi, hokum, dan sebagainya, mengakibatkan berita tersebut dikenali sebagai berita dengan kelas yang berbeda.

TABEL VII
PEMETAAN HASIL KLASIFIKASI KELAS KEDUA

1 \ 2	Ekonomi	Hiburan	Olahraga	Politik	Teknologi
Ekonomi	0	0	0	0	1
Hiburan	0	0	1	0	0
Olahraga	0	0	0	0	0
Politik	0	0	0	0	1
Teknologi	0	0	0	0	0

DOI : 10.37823/insight.v3i01.126

D. Analisa Ujicoba Data Baru

Setelah melakukan berbagai ujicoba, terlebih terhadap data baru di luar dataset, bisa dilihat bahwa untuk data A memiliki akurasi yang rendah. Bahkan untuk persentase testing yang rendah, terkadang terjadi *misclassification* / kesalahan klasifikasi. Sedangkan dibanding dengan data B, tingkat akurasi cukup tinggi yaitu di atas 90% dan hasil klasifikasinya tepat. Berikut adalah hasil dari analisa penyebab perbedaan tingkat akurasi tersebut.

1) Perbedaan sumber berita

Data A diambil dari kompas.com, sedangkan data B diambil dari cnnindonesia.com. Meskipun kedua berita tersebut seharusnya termasuk kategori politik, namun hasil prediksi dari kedua data tersebut terpaut cukup jauh. Hal ini bisa disebabkan karena pada sumber berita yang berbeda, maka kosakata dan gaya pemberitaan yang digunakan pada berita tersebut juga akan berbeda. Hal ini mengakibatkan rendahnya akurasi untuk data A karena satu-satunya data dengan sumber yang berbeda. Berbeda dengan B yang memiliki sumber yang sama dengan dataset awal yang digunakan sehingga memiliki akurasi yang tinggi.

2) Perbedaan wilayah berita

Penyebab lain adalah karena wilayah pemberitaan yang berbeda, dimana data A adalah berita untuk wilayah Amerika Serikat, sedangkan data B adalah berita untuk wilayah Indonesia. Perbedaan wilayah juga memiliki potensi penggunaan kata atau istilah yang berbeda, seperti misalnya nama Joe Bidden ataupun Donald Trump tidak pernah muncul pada dataset yang telah disiapkan sebelumnya. Karena dari frekuensi kemunculan nama orang saja dapat menentukan kategori dari berita tersebut apakah termasuk politik atau olahraga.

3) Perbedaan waktu rilis berita

Penyebab ketiga perbedaan tingkat akurasi adalah waktu saat berita rilis. Dataset yang digunakan adalah berita pada tahun 2016-2017, sedangkan data A dan data B adalah berita baru pada tahun 2020. Perbedaan waktu ini dapat memunculkan istilah baru yang belum ada pada saat tahun dataset rilis. Contohnya adalah kata-kata seperti *virus*, *corona*, *covid*, *electoral*, dan kata-kata lainnya.

E. Kesimpulan Hasil Ujicoba

Setelah melalui berbagai ujicoba yang telah dilakukan, maka dapat ditarik kesimpulan sebagai berikut:

- Hipotesa awal dari Tesis ini adalah keberhasilan klasifikasi dokumen sebesar 70%. Hipotesis tersebut tercapai karena hasil akhir perhitungan akurasi dari ujicoba cukup tinggi dan melebihi target, yaitu di atas 90%. Hal ini bisa disebabkan oleh sumber dataset yang digunakan untuk melatih doc2vec, melatih SVM, dan untuk testing semuanya menggunakan sumber data yang sama yaitu dari CNN pada rentang waktu tertentu, sehingga kosakata yang digunakan pada dokumen-dokumen berita tersebut memiliki pola tertentu ataupun kata-kata yang mirip antar dokumen.
- Hasil akurasi dari penelitian sebelumnya oleh Nanak Chand yang juga menggunakan metode SVM cukup tinggi, yaitu di atas 90%. Dan pada hasil ujicoba kali ini juga mendapat akurasi yang cukup bagus yaitu juga di atas 90%. Dari kedua data tersebut dapat disimpulkan bahwa dengan menggunakan SVM terbukti mampu melakukan klasifikasi terhadap beberapa kelas sekaligus dengan tingkat akurasi yang tinggi.
- Untuk data baru yang akan diprediksi, baiknya menggunakan data dengan sumber dan jenis yang sama agar dokumen berita tersebut dapat dikenali dan diklasifikasi dengan baik. Adanya perbedaan sumber maupun waktu dapat menyebabkan perbedaan kosakata yang digunakan, sehingga hasil dari training sebelumnya menjadi kurang optimal.
- Untuk lebih menambah kosakata, sebaiknya dataset yang disiapkan berasal dari beberapa sumber terpisah, sehingga ketika memprediksi data berita baru, akan dapat terprediksi dengan baik.
- Dari hasil klasifikasi, ditemukan bahwa kelas Ekonomi, Hiburan, dan Politik lebih kuat karena banyak memprediksi dataset sebagai kelasnya. Sedangkan kelas Olahraga dan Teknologi lebih lemah dan banyak terprediksi sebagai kelas lain. Hal ini disebabkan kumpulan kosakata pada kelas yang lebih lemah tersebut cenderung lebih umum, sehingga ketika bertemu dengan kosakata yang spesifik untuk kelas lain, dataset tersebut menjadi dikenali lebih condong ke kelas lain itu.

VI. KESIMPULAN

Kesimpulan diperoleh setelah melalui proses pengembangan dan ujicoba dari sistem yang dikerjakan, dengan mengamati setiap data yang telah disiapkan, proses yang dilakukan, dan hasil yang diperoleh. Berikut merupakan beberapa kesimpulan yang didapat:

- Dataset yang digunakan untuk training model Doc2Vec memiliki peranan penting karena akan menjadi dasar dari penentuan vector untuk data / dokumen baru nantinya. Hal tersebut menentukan kosakata yang dimiliki oleh model Doc2Vec dan akan dapat mempengaruhi fitur serta prediksi yang dihasilkan. Apabila dokumen baru yang akan dicari vektornya berasal dari sumber yang berbeda atau memiliki kosakata yang baru dan belum pernah dikenali sebelumnya oleh model Doc2Vec, maka akurasi klasifikasi dari vector yang terbentuk dari model Doc2Vec tersebut menjadi rendah dan memungkinkan terjadinya kesalahan pada saat prediksi.
- Melihat hasil ujicoba yang memiliki tingkat akurasi yang cukup tinggi dan prediksi yang tepat, deretan vektor yang dihasilkan Doc2Vec mampu mewakili sebuah dokumen dengan baik untuk dapat digunakan sebagai fitur klasifikasi, dan SVM dapat melakukan klasifikasi serta prediksi sebuah data terhadap beberapa kelas sekaligus dengan baik.

DAFTAR PUSTAKA

[1] A Gentle Introduction to Doc2Vec, https://medium.com/wisio/a-gentle-introduction-to-doc2vec-db3e8c0cce5e, Diakses tanggal 25 November 2020

[2] A.muis, Imelda, Muhammad Affandes. *Penerapan Metode Support Vector Machine (SVM) Menggunakan Kernel Radial Basis Function (RBF) Pada Klasifikasi Tweet*. Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, UIN Sultan Syarif Kasim Riau, 2015

[3] Ariadi, Dio, Kartika Fithriasari. Klasifikasi Berita Indonesia Menggunakan Metode Naive Bayesian Classification dan Support Vector Machine Dengan Confix Stripping Stemmer. Jurnal Sains dan Seni ITS, Surabaya, 2016

 DOI : 10.37823/insight.v3i01.126

[4] Arifin, Agus Zainal, Ari Novan Setiono. *Klasifikasi Dokumen Berita Kejadian Berbahasa Indonesia dengan Algoritma Single Pass Clustering.* Jurusan Teknik Informatika, Institut Teknologi Sepuluh November (ITS), Surabaya, 2002
[5] C. Chang, S. Lee and C. Lai, "Weighted word2vec based on the distance of words," 2017 International Conference on Machine Learning and Cybernetics (ICMLC), Ningbo, China, 2017, pp. 563-568, doi: 10.1109/ICMLC.2017.8108974.
[6] Cara Kerja Word2Vec, https://medium.com/@afrizalfir/mengenal-word2vec-af4758da6b5d, Diakses tanggal 25 November 2020
[7] Chand, Nanak, Preeti Mishra, C. Rama Krishna, Emmanuel Shubhakar Pilli, Mahesh Chandra Govil. *A Comparative Analysis of SVM and its Stacking with other Classification Algorithm for Intrusion Detection*. Department of Computer Science and Engineering, National Institute of Technical Teachers, Chandigarh, India.
[8] Even-Zohar, Y.,(2002), Introduction to Text Mining, Supercomputing.
[9] Evolution and Future of NLP, https://www.xenonstack.com/blog/evolution-of-nlp/ Diakses tanggal 25 Oktober 2020
[10] Februariyanti, Herny, Eri Zuliarso. *Klasifikasi Dokumen Berita Teks Bahasa Indonesia Menggunakan Ontologi*. Fakultas Teknologi Informasi, Universitas Stikubank, Semarang, 2012.
[11] Fradkin, Dmitriy, Ilya Muchnik. *Support Vector Machines for Classification*. 2000 Mathematics Subject Classification. 62H30.
[12] Gensim Tutorial, https://www.machinelearningplus.com/nlp/gensim-tutorial, Diakses tanggal 25 Oktober 2020
[13] H. İ. Çelenli, "Application of paragraph vectors to news and tweet data," 2018 26th Signal Processing and Communications Applications Conference (SIU), Izmir, 2018, pp. 1-4, doi: 10.1109/SIU.2018.8404232.
[14] H. Arslan, O. Kaynar and S. Şahİn, "Classification of Customer Demands by Using Doc2Vec Feaure Extraction Method," 2019 27th Signal Processing and Communications Applications Conference (SIU), Sivas, Turkey, 2019, pp. 1-4, doi: 10.1109/SIU.2019.8806452.
[15] Han, J. and Michelline Kamber, (2006). Data Mining: Concepts and Techniques, 2nd. Morgan Kaufmann.
[16] J. Gao, Y. He, X. Zhang and Y. Xia, "Duplicate short text detection based on Word2vec," 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 2017, pp. 33-37, doi: 10.1109/ICSESS.2017.8342858.
[17] Latent Dirichlet Allocation (LDA), https://socs.binus.ac.id/2018/11/29/latent-dirichlet-allocation-lda/, diakses tanggal 26 Oktober 2020
[18] Lau, Jey Han, Timothy Baldwin. *An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation*. Dept of Computing and Information Systems, IBM Research, The University of Melbourne.
[19] Lott, B. (2012), Survey of Keyword Extraction Techniques, Study Literature Project: Principles of Artificially Intelligent, University of New Mexico, Albuquerque.
[20] M. Bilgin and İ. F. Şentürk, "Sentiment analysis on Twitter data with semi-supervised Doc2Vec," 2017 International Conference on Computer Science and Engineering (UBMK), Antalya, Turkey, 2017, pp. 661-666, doi: 10.1109/UBMK.2017.8093492.
[21] Mani, Inderjeet. Automatic Summarization Volume 3 of Natural Language Processing. 2001.
[22] Mayoraz, Eddy, Ethem Alpaydin. *Support Vector Machines for Multi-class Classification*. Dept of Computer Engineering, Bogazici University, Istanbul, Turkey.
[23] Menahem, Eitan, Lior Rokach, Yuval Elovici. *Troika – an Improved Stacking Schema for Classification Tasks*. Information Systems Engineering Department, Ben Gurion University, Israel.
[24] Mengenal Machine Learning, http://www.postmedya.com/default/mengenal-lebih-dalam-tentang-machine-learning/, Diakses tanggal 25 Oktober 2020
[25] Mengenal Topic Modelling, https://toolbox.kurio.co.id/topic-modeling-696d7ba2592f, diakses tanggal 25 Oktober 2020
[26] Multi-Class text classification With doc2vec Logistic Regression, https://towardsdatascience.com/multi-class-text-classification-with-doc2vec-logistic-regression-9da9947b43f4, Diakses tanggal 7 November 2020
[27] NLP Examples, https://www.bloomreach.com/en/blog/2019/09/natural-language-processing.html, Diakses tanggal 25 Oktober 2020
[28] P. Karvelis, D. Gavrilis, G. Georgoulas and C. Stylios, "Topic recommendation using Doc2Vec," 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 2018, pp. 1-6, doi: 10.1109/IJCNN.2018.8489513.
[29] Pakana, Fitrio "Perancangan Dan Pembuatan Aplikasi Pencarian Dokumen Berbasis Web Dengan Penerapan Metode Suffix Tree Clustering Pada Result Set", Tugas Akhir, Teknik Informatika, Institut Teknologi Sepuluh Nopember Surabaya, 2001
[30] R. Nath Nandi, M. M. Arefin Zaman, T. Al Muntasir, S. Hosain Sumit, T. Sourov and M. Jamil-Ur Rahman, "Bangla News Recommendation Using doc2vec," 2018 International Conference on Bangla Speech and Language Processing (ICBSLP), Sylhet, Bangladesh, 2018, pp. 1-5, doi: 10.1109/ICBSLP.2018.8554679.
[31] Support Vector Machine, https://scikit-learn.org/stable/modules/svm.html, Diakses tanggal 30 Oktober 2020
[32] SVM Classification in Python, https://www.datacamp.com/community/tutorials/svm-classification-scikit-learn-python, Diakses tanggal 30 Oktober 2020
[33] Text Preprocessing dengan Python, https://medium.com/@ksnugroho/dasar-text-preprocessing-dengan-python-a4fa52608ffe, Diakses tanggal 26 November 2020
[34] Tokenisasi, https://id.wikipedia.org/wiki/Tokenisasi, Diakses tanggal 26 November 2020
[35] W. Tian, J. Li and H. Li, "A Method of Feature Selection Based on Word2Vec in Text Categorization," 2018 37th Chinese Control Conference (CCC), Wuhan, China, 2018, pp. 9452-9455, doi: 10.23919/ChiCC.2018.8483345.
[36] W. Yue and L. Li, "Sentiment Analysis using Word2vec-CNN-BiLSTM Classification," 2020 Seventh International Conference on Social Networks Analysis, Management and Security (SNAMS), Paris, France, 2020, pp. 1-5, doi: 10.1109/SNAMS52053.2020.9336549.
[37] Widiastuty, Nelly Indriani, Ednawati Rainarli, Kania Evita Dewi. *Peringkasan dan Support Vector Machiine pada Klasifikasi Dokumen*. Universitas Komputer Indonesia, 2017
[38] Y. Safali, G. Nergız, E. Avaroğlu and E. Doğan, "Deep Learning Based Classification Using Academic Studies in Doc2Vec Model," 2019 International Artificial Intelligence and Data Processing Symposium (IDAP), Malatya, Turkey, 2019, pp. 1-5, doi: 10.1109/IDAP.2019.8875877.

 DOI : 10.37823/insight.v3i01.126