Pengembangan Model Klasifikasi Dokumen Artikel Teks Berita Olahraga dan Bukan Olahraga dalam Bahasa Indonesia Menggunakan Algoritma Support Vector Machine

Muhammad Alif Iman Aulia[1)], Yulia Ery Kurniawati[2)]

Informatika, Fakultas Ilmu Komputer dan Desain, Institut Teknologi dan Bisnis Kalbis
Jalan Pulomas Selatan Kav. 22, Jakarta 13210
[1)] *Email: alif.iman98@gmail.com*
[2)]*Email: yulia.kurniawati@kalbis.ac.id*

Abstract: *The purpose of classification is to inform the machine (algorithm) of group division or in other words, teach the machine how to divide the group. The classification process uses the Support Vector Machine (SVM) method. SVM is used because it has a good reputation in the classification. This study aims to classify sports news and non-sports news using the SVM algorithm and with the addition of n-gram features and Term Frequent Inverse Document Frequency (TF-IDF). The results showed an accuracy of 92%.*
Keywords: *Classification, news, n-gram, SVM, TF-IDF*

Abstrak: *Tujuan dari klasifikasi yakni untuk memberi tahu kepada mesin (algoritma) pembagian kelompok atau dalam kata lain mengajari mesin cara membagi kelompoknya. Proses klasifikasi menggunakan metode Support Vector Machine (SVM). SVM digunakan karena memiliki reputasi yang baik dalam klasifikasi. Penelitian ini bertujuan untuk mengklasifikasikan berita olahraga dan berita bukan olahraga menggunakan algoritma SVM dan dengan penambahan fitur n-gram dan Term Frequent Inverse Document Frequency (TF-IDF). Hasil penelitian menunjukkan akurasi sebesar 92%.*
Kata kunci: *Berita, Klasifikasi, n-gram, SVM, TF-IDF*

I. PENDAHULUAN

Pada saat ini, dokumen teks berita telah mengalami evolusi dari dokumen yang berbentuk fisik menjadi elektronik, sehingga banyak perusahaan media sekarang berfokus pada *website* berita yang bisa diakses oleh khalayak orang. Banyaknya data elektronik, khususnya dalam dokumen teks, menghasilkan banyak penelitian dalam bidang pengklasifikasian teks. Fakta bahwa sebagian besar data disimpan dalam bentuk teks dan pertumbuhan eksponensial dari data, memicu peningkatan jumlah penelitian penambangan teks. Kondisi tersebut dapat menyebabkan adanya era ledakan informasi [1]. Klasifikasi artikel berita adalah salah satu metode dalam domain teks yang banyak dipelajari. Misalnya, klasifikasi berita dalam Bahasa Arab [2], klasifikasi tipe surat kabar menggunakan *Naive Bayes Classification (NBC), Maximum Entropy*, dan *Probabilistic Classification* [3]. Ada banyak penelitian dalam klasifikasi artikel berita digital yang telah dilakukan sampai sekarang dan banyak algoritma telah diimplementasikan dalam rangka untuk menciptakan sebuah *classifier* yang memiliki akurasi mendekati 100%, metode klasifikasi teks telah diusulkan seperti *Decision Tree*, *Support Vector Machine* (SVM),

Artificial Neural Network, Bayesian Generative, KNN, dll. [4]. Algoritma seperti *Decision Tree*, SVM[5], dan *K-Nearest Neighbors* (KNN) [6-7] adalah algoritma yang paling umum digunakan dalam bidang studi pengklasifikasian teks. Algoritma tersebut juga dapat diterapkan untuk beberapa bahasa yang berbeda. Pendekatan statistik juga dapat diimplementasikan dalam penelitian ini [8]. Berdasarkan latar belakang di atas, maka algoritma lain harus diuji untuk mendapatkan hasil yang optimal. Penelitian ini akan menggunakan metode klasifikasi SVM untuk mengklasifikasikan artikel teks berita olahraga dan bukan olahraga. Penelitian ini menggunakan *dataset* sebanyak 1144 yang terdiri dari 572 data teks berita olahraga.

II. METODE PENELITIAN

A. *Text Preprocessing*

Text Preprocessing adalah bagian penting dari sistem *Natural Language Processing* (NLP), karena karakter, kata, dan kalimat yang diidentifikasi pada tahap ini adalah unit dasar yang akan diteruskan ke semua tahap pemrosesan lebih lanjut, dari analisis dan penandaan komponen, seperti analisis morfologi dan bagian dari penandaan ucapan, melalui aplikasi, seperti pencarian informasi dan sistem terjemahan mesin. Kumpulan kegiatan di mana dokumen teks sudah diproses sebelumnya. Karena data teks sering berisi beberapa format khusus seperti format angka, format tanggal, dan kata-kata paling umum yang tidak mungkin membantu klasifikasi teks seperti preposisi, artikel, dan kata benda bisa dihilangkan [9].

Text Preprocessing dapat dibagi menjadi beberapa tahap yaitu:

1. *Case Folding*

Case Folding adalah proses untuk mengkonversi kata atau keseluruhan teks menjadi bentuk standar (biasanya huruf kecil atau *lowercase*). Proses ini dilakukan untuk memastikan bahwa kata "ORANG", "orang", "OraNg ", "oranG", atau variasi kasus lain akan ditafsirkan sebagai arti kata yang sama [10].

2. *Remove Number*

Remove Number adalah proses untuk menghapus angka pada data yang telah diperoleh. Contoh dari *remove number* yakni untuk memastikan kata “Orang10” menjadi “Orang”.

3. *Punctuation Removal*

Punctuation removal adalah proses untuk menghapus tanda baca yang ada pada kalimat atau kata. Contoh dari *punctuation removal* yakni untuk memastikan bahwa kata “rumah?!@#” menjadi “rumah”.

4. *Remove Multiple Whitespace*

Remove multiple whitespace adalah proses penghapusan spasi yang ganda atau lebih dari satu. Contoh dari *remove multiple whitespace* yakni menjadikan kata “hotel “ menjadi “hotel”.

5. *Word Tokenize*

Word Tokenize adalah proses memecah aliran teks menjadi kata, frasa, simbol, atau elemen bermakna lainnya yang disebut token. Tujuan *word tokenize* adalah untuk eksplorasi kata-kata dalam sebuah kalimat. *Word tokenize* berguna baik dalam linguistik (di mana ia adalah bentuk segmentasi teks), dan dalam ilmu komputer, di mana ia membentuk bagian dari analisis leksikal. Penggunaan utama *word tokenize* adalah mengidentifikasi kata kunci yang bermakna. Inkonsistensi dapat berupa format angka dan waktu yang berbeda. Masalah lain adalah singkatan dan akronim yang harus diubah menjadi bentuk standar [9].

6. *Stop word removal*

Banyak kata dalam dokumen berulang sangat sering muncul tetapi pada dasarnya tidak berarti, karena kata yang digunakan berfungsi untuk

menggabungkan kata dalam sebuah kalimat. Secara umum dipahami bahwa kata penghenti tidak berkontribusi pada konteks atau isi dokumen, karena frekuensi kemunculannya yang tinggi, kehadiran mereka dalam klasifikasi teks menghadirkan kendala dalam memahami konten dokumen.

Stop word sangat sering digunakan kata-kata umum seperti 'dan', 'adalah', 'ini' dll. Kata ini tidak berguna dalam klasifikasi dokumen sehingga harus dihilangkan. Proses ini juga mengurangi data teks dan meningkatkan kinerja sistem. Setiap dokumen teks membahas kata-kata ini yang tidak diperlukan untuk aplikasi penambangan teks maupun klasifikasi teks [9].

7. *Stemming*

Stemming adalah proses pemetaan dan penguraian berbagai bentuk (*variants*) dari suatu kata menjadi bentuk kata dasarnya (*stem*) [11]. Tujuan dari proses *stemming* adalah menghilangkan imbuhan-imbuhan baik itu berupa prefiks, sufiks, maupun konfiks yang ada pada setiap kata. *Stemming* adalah bagian dari studi linguistik dalam morfologi dan AI pengambilan dan ekstraksi informasi. Pengetahuan *stemming* dan AI mengekstrak informasi yang bermakna dari sumber yang luas seperti data besar atau internet karena bentuk tambahan dari sebuah kata yang berhubungan dengan subjek mungkin perlu dicari untuk mendapatkan hasil. terbaik. *Stemming* juga merupakan bagian dari *query* dan mesin pencari internet mengenali, mencari, dan mengambil lebih banyak bentuk kata menghasilkan lebih banyak hasil. Ketika sebuah bentuk kata diakui itu dapat memungkinkan untuk mengembalikan hasil pencarian yang dinyatakan mungkin telah terjawab. Informasi tambahan yang diambil adalah mengapa berasal merupakan bagian integral dari permintaan pencarian dan pengambilan informasi [12].

B. *Support Vector Machine*

SVM merupakan salah satu metode dalam *supervised learning* yang digunakan untuk klasifikasi. SVM digunakan untuk mencari *hyperplane* yang paling baik dengan memaksimalkan jarak antar kelas [13]. SVM adalah salah satu teknik yang relatif baru dibandingkan dengan teknik lainnya, tetapi memiliki performansi lebih baik di bidang aplikasi *bioinformatics*, klasifikasi teks, pengenalan tulisan tangan, dan lain-lain [14].

Algoritma SVM menggunakan seperangkat fungsi matematika yang didefinisikan sebagai kernel. Fungsi kernel adalah untuk mengambil data sebagai input dan mengubahnya menjadi bentuk yang diperlukan. Algoritma SVM yang berbeda menggunakan berbagai jenis fungsi kernel. Salah satu fungsinya yaitu kernel *linear*. Kernel *linear* berguna ketika berhadapan dengan *large sparse data vector*.

Pada kernel *linear* digunakan rumus dengan persamaan:

*f(x) = B(0) + sum(ai * (x,xi))*

(1) Pada persamaan 2.1 menjelaskan bahwa penghitungan produk dalam dari vektor input baru (x) dengan semua vektor pendukung dalam data *train.* Koefisien B(0) dan ai (untuk setiap input) harus diperkirakan data data *train* dengan algortima pembelajaran [14].

C. *Cross Validation*

Cross validation adalah teknik yang melibatkan sampel tertentu dari *dataset* yang tidak dilatih modelnya. *Cross validation* adalah metode statistik yang digunakan untuk mengevaluasi kinerja model yang telah dibuat, dimana data dipisahkan menjadi dua yaitu data proses pembelajaran dan data validasi. Model akan dilatih oleh data proses pembelajaran dan divalidasi. Biasanya

k-fold cross validation digunakan karena dapat mengurangi waktu komputasi dengan menjaga ketetapan dan keakuratan estimasi [15].

D. *Term Frequent Inverse Document Frequency (*TF-IDF)

Metode TF-IDF merupakan metode untuk menghitung bobot setiap kata yang sangat umum untuk digunakan pada *information retrieval*. Metode ini juga terkenal mudah, efisien, dan memiliki hasil yang akurat [16]. Metode *Term Frequency Inverse Document Frequency* (TF-IDF) adalah cara pemberian bobot hubungan suatu kata (*term*) terhadap dokumen. TF-IDF ini adalah sebuah ukuran statistik yang digunakan untuk mengevaluasi pentingnya sebuah kata di dalam sebuah dokumen atau dalam sekelompok kata. Untuk dokumen tunggal tiap kalimat dianggap sebagai dokumen. Frekuensi kemunculan kata di dalam dokumen yang diberikan menunjukkan seberapa penting kata itu di dalam dokumen tersebut. Frekuensi dokumen yang mengandung kata tersebut menunjukkan seberapa umum kata tersebut. Bobot kata semakin besar jika sering muncul dalam suatu dokumen dan semakin kecil jika muncul dalam banyak dokumen [17]. Pada algoritma TF-

IDF digunakan rumus untuk menghitung bobot (W) masing masing dokumen terhadap kata kunci dengan persamaan[18] :

$$Wdt = tfdt * Idft \quad (2)$$

Dimana Wdt= bobot dokumen ke-d terhadap kata ke-t tfdt = banyaknya kata yang dicari pada sebuah dokumen. Idft = *Inversed Document Frequency* (log (N/df)), N = total dokumen, df = banyak dokumen yang mengandung kata yang dicari.

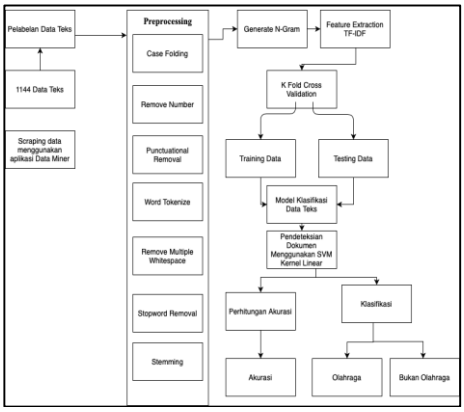
E. N-Gram

Model probabilistik *N-gram*, merupakan model yang digunakan untuk memprediksi kata berikutnya yang mungkin dari kata *N-1* sebelumnya. Model statistika dari urutan kata ini seringkali disebut juga sebagai model bahasa (*language models / LMs).* Model estimasi seperti *N-gram* memberikan probabilitas kemungkinan pada kata berikutnya yang mungkin dapat digunakan untuk melakukan kemungkinan penggabungan pada keseluruhan kalimat. Model *N-gram* merupakan model yang penting dalam setiap pemrosesan suara ataupun bahasa baik untuk memperkirakan probabilitas kata berikutnya maupun keseluruhan *sequence*. *N-gram* cukup esensial pada banyak hal dimana kata perlu diartikan dengan lebih tepat mengingat terkadang ada input yang ambigu maupun gangguan (*noise*). model *N-gram* juga memegang peranan amat penting dalam NLP, seperti *part-of-speech tagging, natural language generation,* dan *word similarity.*

F. Kerangka Pemikiran

Penelitian ini membahas tentang pengklasifikasian teks menggunakan metode klasifikasi *Support Vector Machine* (SVM) dan fitur *Term Frequent Inverse Document Frequent* (TF-IDF). Berdasarkan studi literatur yang telah dilakukan, maka dengan adanya gagasan yang dimiliki peneliti membuat aplikasi yang dapat melakukan pengklasifikasian teks berita olahraga dan bukan olahraga. Aplikasi ini diharapkan dapat menambah wawasan tentang pengklasifikasian teks. Untuk melakukan pengklasifikasian teks, aplikasi butuh teks berita. Aplikasi ini dibangun menggunakan bahasa pemrograman python. Untuk menggunakan aplikasi dibutuhkan perangkat lunak untuk pengembangan aplikasi, metodologi yang digunakan adalah metode inkremental dan akan dibagi menjadi dua inkremental, inkremental satu untuk pembuatan model klasifikasi teks dan inkremental dua untuk pembuatan

graphic user interface (GUI). Di bawah ini merupakan rancangan kerangka pemikiran berdasarkan gagasan yang dimiliki.



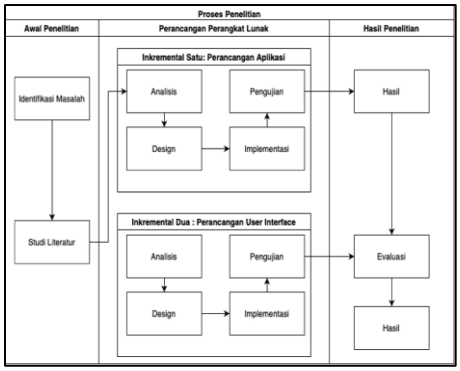
Gambar 1 Kerangka Pemikiran

Kerangka pemikiran yang ada pada Gambar 1 menunjukkan proses-proses tahapan penelitian. Tahap inkremental satu mencakup proses yang berisi pembelajaran teks menggunakan 1144 *dataset*. Proses pengumpulan *dataset* dilakukan dengan menggunakan aplikasi bernama *Data Miner*. Selanjutnya dilakukannya pelabelan pada data yang dilakukan secara manual. Data dibagi menjadi dua label, yaitu label berita olahraga dan label berita bukan olahraga. Label berita olahraga memiliki 572 data teks, sedangkan label berita bukan olahraga sebanyak 572 data teks. Proses selanjutnya adalah data teks akan memasuki tahap *preprocessing.* Proses ini terdiri dari *case folding, remove number, punctuational removal, remove multiple whitespace, word tokenize*, *stopword removal,* dan *stemming*. Proses selanjutnya yaitu penerapan *n-gram* dilanjutkan dengan ekstraksi fitur TF-IDF, proses setelahnya yaitu pembagian data yang digunakan untuk mendapatkan nilai akurasi dan *recall*, pembagian data dilakukan sebanyak 75% sebagai data latih dan 25% sebagai data uji. Rata-rata nilai akurasi dan *recall* di dapat dengan menggunakan algoritma SVM dipadukan dengan *k-fold cross validation* setelahnya dihasilkan model untuk pengklasifikasian teks.

Tahap inkremental dua mencakup proses pengujian menggunakan data teks asal yang diambil dari *website* portal berita. Data teks tersebut mengalami *preprocessing* yang sama, lalu data teks tersebut akan dideteksi dengan mencocokan dengan data latih sebelumnya dengan model yang telah dibuat. Hasil akhir aplikasi ini berupa *file* klasifikasiteks.py dan berjalan secara *offline*.

G. Proses Penelitian

Pada proses penelitian, aplikasi akan dibuat menggunakan metode inkremental untuk digunakan dalam pengembangan aplikasi. Model inkremental dipilih karena pengerjaannya dapat dilakukan secara bertahap. Tahap penelitian ini terdiri dari dua tahap, yaitu tahap inkremental satu dan tahap inkremental dua. Metode inkremental pada tahapan pertama merupakan tahapan *core* atau inti pembuatan model untuk aplikasi ini, sedangkan inkremental tahap kedua berfokus kepada pembuatan tampilan.



Gambar 2 Proses penelitian dengan model inkremental

H. Inkremental Satu

Tahap inkremental satu dikerjakan untuk perancangan aplikasi dan melatih data menggunakan algoritma terkait sehingga didapatkan

model untuk melakukan klasifikasi pada data teks.

1. Analisis

Tahap pertama yang dilakukan pada penelitian ini adalah mencari data teks yang akan digunakan sebagai dataset. Jumlah data teks yang didapat berjumlah 1144 yang terdiri dari 572 data teks berita olahraga dan 572 teks berita bukan olahraga. Pemilihan data teks dilakukan sesuai dengan pengelompokannya, data teks yang digunakan didapat dari website berita online. Dalam proses pengambilan data teks menggunakan aplikasi bernama *Data Miner.*

Penelitian ini menggunakan peralatan yang dibutuhkan untuk mendukung proses pengerjaan aplikasi, perangkat yang digunakan akan dituliskan pada Tabel 1

Tabel 1 Perangkat pembuatan aplikasi

Nama Perangkat Keras	*Spesifikasi*	*Perangkat lunak*
Laptop	*Processor : 2,5 GHz Dual-Core Intel Core i5 Memory : 8 GB Graphic : Intel HD Graphics 4000 1536 MB Storage : 500 GB SSD*	*Mac OS Catalina Anaconda Spyder Python 3.4*

Dalam pengembangan aplikasi, dibutuhkan perangkat keras dan perangkat lunak. Perangkat keras laptop digunakan untuk mendapatkan *dataset* dan juga digunakan untuk mengolah *dataset*. Laptop yang digunakan sudah mempunyai perangkat lunak seperti yang disebut pada Tabel 1. Bahasa pemrograman yang digunakan untuk membangun aplikasi pengklasifikasian teks adalah bahasa pemrograman python 3.7 yang akan diimplementasikan dengan menggunakan perangkat lunak PyCharm dan juga Spyder.

Perangkat keras yang digunakan untuk menjalankan aplikasi yang telah dibuat berspesifikasi seperti yang telah dituliskan pada Tabel 2.

Tabel 2 Perangkat pengguna aplikasi

Nama Perangkat Keras	*Spesifikasi*	*Perangkat lunak*
Laptop Macbook Pro 2012	*Processor : 2,5 GHz Dual-Core Intel Core i5 Memory : 8 GB Graphic : Intel HD 4000 1536 MB Storage : 500 GB SSD*	*Mac OS Catalina Anaconda Spyder Python 3.4*
Laptop MSI GE62VR 7RF	*Processor : Intel Core i7-7700 HQ Memory : 16 GB Graphic : GTX 1060 3GB GDDR5 Storage : 128 GB SSD*	*Windows 10 64 Bit Anaconda Spyder Python 3.4*

Proses selanjutnya adalah menganalisis metode yang dapat digunakan untuk membuat logika model untuk pengelompokan data teks berdasarkan bobot kata. Berdasarkan studi literatur yang telah dilakukan, penelitian dilakukan dengan metode klasifikasi SVM menggunakan *n-gram* dan TF-IDF.

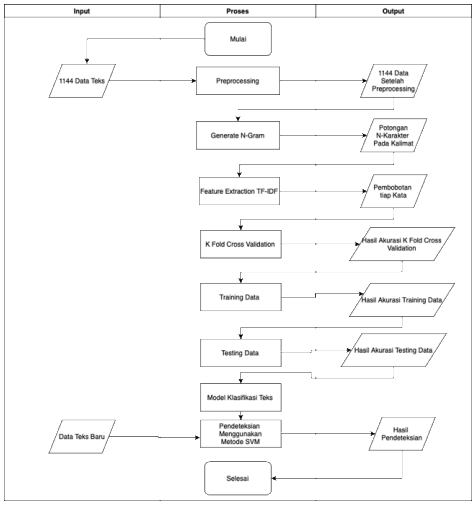
Sebelum dilakukannya pengklasifikasian menggunakan metode dan fitur tertentu, data teks akan dikelompokkan menjadi dua kelompok yakni berita olahraga dan berita bukan olahraga.

Pengelompokan ini digunakan karena penelitian ini menggunakan metode *supervised learning*, artinya data telah memiliki label yang telah dibuat sebelum dilakukan proses pembelajaran. Data teks yang digunakan sebagai data pembelajaran maupun data pengetesan akan mengalami *preprocessing.*

2. Desain

Proses ini dilakukan untuk membuat alur dari logika model. Langkah awal yang akan dilakukan adalah menerapkan tahap *preprocessing* pada 1144 data teks. Tahapan *preprocessing* yang dilakukan terhadap data tersebut meliputi proses *case folding, remove number, punctuational removal, remove multiple whitespace, word tokenize, stopword removal, dan stemming*. Gambar 3 menunjukkan tahapan yang akan dilakukan pada saat aplikasi dijalankan.

Data teks untuk pembanding akan ditempatkan pada satu *file* bernama dataset.csv, Untuk pemisahan data teks pada *file* csv dibuat kolom yang bernama kelas dengan isi yang mencakup dua kategori yaitu olahraga dan bukan olahraga. Tahap selanjutnya adalah melakukan *preprocessing* yang terdiri dari *case folding, remove number, punctuational removal, remove multiple whitespace, word tokenize, stopword removal,* dan *stemming*. Tahapan yang sudah disebutkan diharapkan dapat meningkatkan akurasi dari pengklasifikasian teks. Setelah melakukan tahapan preprocessing pada seluruh data teks yang akan digunakan sebagai data pembelajaran, tahapan selanjutnya yakni penerapan feature n-gram. Tahapan ini berfungsi untuk membantu perhitungan ketepatan nilai akurasi klasifikasi. Selanjutnya *feature extraction* TF-IDF, ini dilakukan untuk mengetahui nilai *term* kemunculan kata dalam suatu dokumen. Setelahnya fitur-fitur tersebut akan digabungkan dengan algoritma SVM dan akan membentuk model baru, model ini digunakan untuk pendeteksian data teks.



Gambar 3 Inkremental Satu

3. Implementasi

Pada inkremental satu, tahapan awal dalam pengolahan data teks adalah *case folding*, yaitu mengkonversi keseluruhan teks menjadi *lower case*. Setelah proses *case folding* dilanjutkan dengan proses *remove number* yakni penghapusan nomor pada teks setelahnya, selanjutnya *punctuational removal* yaitu untuk menghilangkan tanda baca pada teks.

Setelah proses ini, penelitian dilanjutkan dengan *remove multiple whitespace* yakni proses penghapusan spasi yang lebih dari satu. Tahap selanjutnya yaitu *word tokenize* yang berfungsi untuk membuat kalimat dipecah menjadi kata atau frasa. Selanjutnya yaitu *stemming* yang berfungsi untuk menjadikan kata menjadi bentuk dasarnya dan menghilangkan imbuhan didepan maupun dibelakang kata.

Preprocessing terakhir yaitu *stopword removal* yang berfungsi untuk menghilangkan kata yang tidak dibutuhkan dalam proses pengklasifikasian. Selanjutnya yakni proses *feature n-gram* yang dilakukan untuk mendapatkan n-karakter yang

diperoleh dari kata yang telah di *preprocessing*. Setelah *n-gram* dilanjutkan dengan feature *extraction* TF-IDF berfungsi untuk pembobotan nilai *term*, dilanjutkan dengan menggabungkan metode klasifikasi SVM dengan fitur *n-gram* dan TF-IDF yang menjadi model untuk pendeteksian teks.

4. Pengujian

Pada inkremental satu akan menguji dengan metode *whitebox*.

Tabel 3 Tabel pengujian *whitebox* pada inkremental satu

Nama Proses	*Skenario*	*Kode*	*Harapan*
Preprocessing	*Menyeleksi data yang akan diproses pada setiap dokumen sesuai kebutuhan pengklasifikasian.*	*Melakukan penyetaraan teks menjadi lower case, menghilangkan angka pada kata, penghapusan tanda baca, penghapusan spasi yang lebih dari satu, pemisahan kalimat menjadi kata per kata, penghapusan kata yang tidak terlalu dibutuhkan, menguraikan kata menjadi bentuk dasarnya dan menghilangkan imbuhannya.*	*Mendapatkan data teks yang lebih terstruktur untuk bisa diklasifikasikan dengan algortima SVM dipadukan dengan fitur n-gram dan TF-IDF*
Training data & validation	*Membagi dataset untuk training dan menggunakan 10 fold cross validation, lalu dilakukannya testing dan validation.*	*Membagi teks yang telah dilakukan preprocessing, menyusun arsitektur SVM, melakukan training dan validation.*	*Mendapatkan akurasi training dan validation, mendapatkan model yang digunakan untuk prediksi.*
Testing	*Menguji model dengan data teks selain data yang digunakan untuk training dan validation.*	*Memuat model, memuat data testing, memprediksi data testing, membandingkan hasil testing dengan label.*	*Mengetahui kemampuan model yang telah dibuat dalam memprediksi data teks masuk ke kategori tertentu.*

Tabel 3 menjelaskan bahwa seluruh proses inkremental satu dilakukan pengujiannya dengan menggunakan *whitebox.* Terdapat tiga proses yang dilakukan dalam pengujian ini yakni *preprocessing, training data & validation* dan *testing.* Setiap proses dilakukan pengujian sesuai dengan skenario yang telah dibuat, skenario yang telah dibuat diwujudkan dengan kode yang telah diimplementasikan. Hasil dari kode kemudian akan dibandingkan dengan harapan.

I. Inkremental Dua

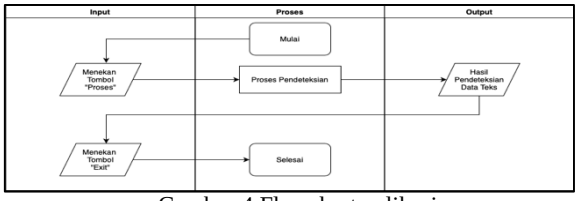
1. Analisis

Tahapan terakhir pada penelitian ini adalah pembuatan tampilan untuk memudahkan pengguna dalam mengoperasikan aplikasi ini. Pada tahap inkremental satu, penelitian hanya fokus dalam fungsi untuk pembuatan model aplikasi, ini dilakukan agar penelitian dapat berjalan sesuai dengan rancangan awal, yaitu melakukan pengklasifikasian teks pada dokumen teks.

Pada tahap ini akan dibuat tampilan aplikasi. Aplikasi ini hanya memiliki satu tampilan, tampilan tersebut adalah tombol yang digunakan untuk mengklasifikasi teks, tombol *restart* aplikasi dan tombol keluar aplikasi.

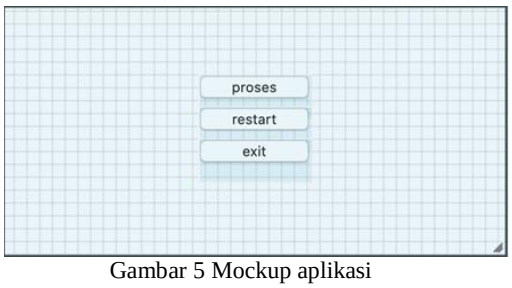
2. Desain

Dari analisis pada inkremental dua, peneliti membuat alur penggunaan aplikasi yang diperuntukan untuk user dengan menggunakan *flowchart.* Alur tersebut dibuat agar memudahkan peneliti dalam mengembangkan tampilan aplikasi serta memudahkan pengguna dalam mengoperasikan aplikasi. Berikut gambar *flowchart* aplikasi.



Gambar 4 Flowchart aplikasi

Di tahap penggunaan aplikasi, *user* menginput teks berita terlebih dahulu *user* menekan tombol "proses" untuk memproses dan mengklasifikasikan jenis berita tersebut dengan fungsi yang sudah dibuat pada inkremental satu. Hasil klasifikasi akan muncul berupa akurasi dan prediksi teks tersebut olahraga atau bukan olahraga. *User* dapat mengulangi langkah yang sama dengan memencet tombol *restart* untuk menguji teks lainnya. Untuk keluar dari aplikasi tekan tombol "exit". Berikut merupakan tampilan mockup aplikasi beserta contoh hasilnya.



Gambar 5 Mockup aplikasi

3. Implementasi

Tahap implementasi dilakukan untuk menggambarkan GUI ke dalam bentuk *code*. Penelitian ini menggunakan bahasa pemrograman python, sehingga perancangan GUI dapat menggunakan *library* khusus yang sudah disediakan untuk merancang GUI.

4. Pengujian

Tabel 4 Menggunakan font Times New Roman 9 pt

Nama Proses	*Skenario*	*Kode*	*Harapan*
Memprediksi data teks berdasarkan teks yang telah diinput oleh pengguna aplikasi.	*Menginput data teks yang akan diklasifikasikan, memprediksi data teks, menampilkan hasil prediksi data teks.*	*Memprediksi input teks yang ingin diklasifikasikan dan hasil akan ditampilkan.*	*Mendapatkan hasil prediksi teks berdasarkan teks yang di input.*

Tabel 3.4 menjelaskan bahwa seluruh proses inkremental dua diuji dengan menggunakan *whitebox.* Terdapat satu proses yang dilakukan dalam pengujian yaitu memprediksi data teks yang telah di input oleh pengguna aplikasi, proses ini dilakukan dengan pengujian mengikuti skenario yang telah dibuat, skenario yang telah dibuat diwujudkan dengan kode yang telah diimplementasikan. Hasil dari kode kemudian akan dibandingkan dengan harapan.

III. HASIL DAN PEMBAHASAN

A. Inkremental Satu

Pada tahap inkremental satu dilakukan pengujian dari fungsi yang dibuat menggunakan bahasa pemrograman python

1. Hasil

Tabel 5 Tabel hasil pengujian *whitebox* inkremental satu

Nama Proses	*Skenario*	*Kode*	*Harapan*	*Hasil*
Preprocessing	*Menyeleksi data yang akan diproses pada setiap dokumen*	*Melakukan penyetaraan teks menjadi lower case, menghilangkan angka pada kata, penghapusan tanda baca, penghapusan spasi yang lebih dari satu, pemisahan kalimat menjadi kata per kata, penghapusan kata yang tidak terlalu dibutuhkan, mengurai kan kata menjadi bentuk dasarnya dan menghilangkan imbuhannya.*	*Mendapatkan data teks yang lebih terstruktur untuk bisa diklasifikasikan dengan algortima SVM dipadukan dengan fitur n-gram dan TF-IDF.*	*Valid*
Training Data & validation	*Membagi dataset untuk training dan menggunakan 10 fold cross validation, lalu dilakukannya testing dan validation*	*Membagi teks yang telah dilakukan preprocessing, menyusun arsitektur SVM, melakukan training dan validation.*	*Mendapatkan akurasi training dan validation, mendapatkan model yang digunakan untuk prediksi*	*Valid*
Testing	*Menguji model dengan data teks selain data yang digunakan untuk training dan validation.*	*Memuat model, memuat data testing, memprediksi data testing, membandingkan hasil testing dengan label.*	*Mengetahui kemampuan model yang telah dibuat dalam memprediksi data teks masuk ke kategori tertentu.*	*Valid*

Pada Tabel 5 dijelaskan bahwa pengujian *whitebox* pada inkremental satu memiliki tiga proses yang dilakukan. Proses pengujian meliputi preprocessing, training data & validation, testing. Seluruh proses pengujian dengan metode *whitebox* berhasil dicapai berdasarkan skenario yang telah diharapkan dengan menggunakan kode yang melakukan proses tersebut.

B. Inkremental Dua

Pada tahap inkremental dua dilakukan pembuatan GUI menggunakan *library* tkinter.

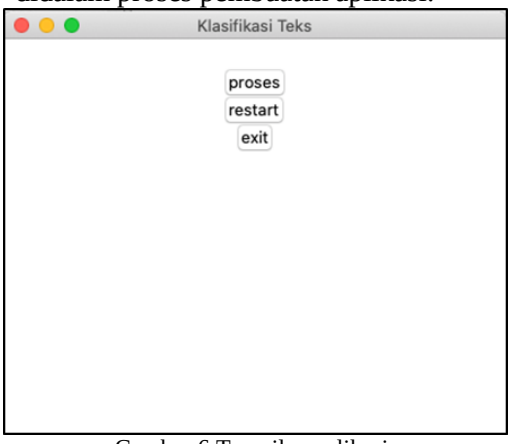
1. Hasil

Nama Proses	*Skenario*	*Kode*	*Harapan*	*Hasil*
Memprediksi data	*Menginput data teks yang*	*Memprediksi input*	*Mendapatkan hasil*	*Valid*

teks berdasarkan teks yang telah diinput oleh pengguna aplikasi.	*akan diklasifikasikan, memprediksi data teks, menampilkan hasil prediksi data teks.*	*teks yang ingin diklasifikasikan dan hasil akan ditampilkan.*	*prediksi teks berdasarkan teks yang di input.*

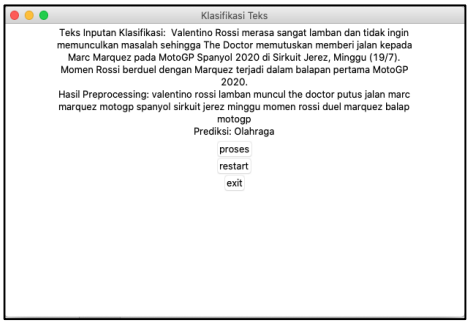
Pengujian *whitebox* pada inkremental dua memiliki satu proses yang dilakukan. Proses pengujian yakni memprediksi data teks berdasarkan teks yang telah diinput oleh pengguna aplikasi. Seluruh proses pengujian dengan metode *whitebox* berhasil dicapai berdasarkan skenario yang telah diharapkan dengan menggunakan kode yang melakukan proses tersebut.

Pada tahap implementasi akan dijelaskan kode-kode dan perhitungan serta akan ditampilkannya output tahap tersebut. Penjelasan kode akan dilakukan dengan cara membahas setiap fungsi-fungsi yang akan digunakan didalam proses pembuatan aplikasi.



Gambar 6 Tampilan aplikasi

Gambar 6 menunjukkan tampilan aplikasi ketika *code* di jalankan dengan ada tiga *button* di tampilannya.



Gambar 7 Tombol proses ditekan

Tombol proses ditekan dan akan menampilkan tampilan hasil akurasi dan prediksi seperti yang ada pada Gambar 7.

IV. SIMPULAN

Setelah melakukan penelitian tentang klasifikasi teks dapat ditarik kesimpulan sebagai berikut**:**

1. Pengembangan model untuk pengklasifikasian teks berbahasa Indonesia berhasil dilakukan dan pengimplementasian metode SVM bisa digunakan untuk mengklasifikasikan teks berita.
2. Akurasi model yang dibuat untuk klasifikasi teks berita olahraga dan bukan olahraga bernilai 92%.

DAFTAR RUJUKAN

[1] Jasiliu A.Kadiri And Niran A. Adetoro, "Information Explosion And The Challenges Of Information Andcommunication Technology Utilization In Nigerian Libraries Andinformation Centres," *Ozean Journal Of Social Sciences 5*, 2012.

[2] C. C. Aggarwal and C. X. Zhai, “A survey of text classification algorithms,” n *Mining Text Data*, vol. 9781461432234, 2012, pp. 163–222.

[3] D. Ramdass and S. Seshasai, "Document Classification for Newspaper Articles," 2009.

[4] T. Kanan and E. A. Fox, "Automated arabic text classification with P-Stemmer, machine learning, and a tailored news article taxonomy," J. Assoc. Inf. Sci.Technol., 2016.

[5] Motaz K. Saad, "*The Impact Of Text Preprocessing And Term Weighting On Arabic Text Classification*," 2010.

[6] Arni Darliani Asy'arie And Adi Wahyu Pribadi, "Automatic News Articles Classification In Indonesian Language By Using Naive Bayes Classifier Method," *Iiwas*, 2009.

[7] Thiago Salles And Leonardo Rocha, "Automatic Document Classification Temporally Robust,".

[8] Shrikanth Shankar And George Karypis, "A Feature Weight Adjustment Algorithm For Document Categorization,".

[9] S. Kannan *et al.*, "Preprocessing Techniques for Text Mining," *Int. J. Comput. Sci. Commun. Networks*, 2015.

[10] A. T. H. Harjanta, "Preprocessing Text untuk Meminimalisir Kata yang Tidak Berarti dalam Proses Text Mining," Jurnal Informatika Upgris, vol. 1, 2015.

[11] Tala, Fadillah Z. 2003. A Study of Stemming Efects on Information Retrieval in Bahasa Indonesia. Institute for Logic, Language and ComputationUniversiteit van Amsterdam The Netherlands.http://www.illc.uva.nl/Research/Reports /MoL-2003-02.text.pdf. Diakses tanggal 19 Maret 2020.

[12] M. Rouse, "Stemming," *searchenerpriseai.techtarget.com*. [Online]. Available: https://searchenterpriseai.techtarget.com/definition/stemming. [Accessed: 06-Jun-2020].

[13] Samsudiney, 2019. Penjelasan Sederhana Tentang apa itu SVM?. [online] medium. Available at: https://medium.com/@samsudiney/penjelasan-sederhana-tentang-apa-itu-svm-149fec72bd02 [Accessed 2 April 2020].

[14] S. Patel, "Chapter 2 : SVM (Support Vector Machine) — Theory," *medium.com*, 2017. [Online]. Available: https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72. [Accessed: 19-Jun-2020].

[15] A. Wibowo, "10 FOLD-CROSS VALIDATION," *binus*. [Online]. Available: https://mti.binus.ac.id/2017/11/24/10-fold-cross-validation/. [Accessed: 08-Jun-2020].

[16] A. A. Maarif, "Penerapan Algoritma TF-IDF untuk Pencarian Karya Ilmiah," *Dok. Karya Ilm. | Tugas Akhir | Progr. Stud. Tek. Inform. - S1 | Fak. Ilmu Komput. | Univ. Dian Nuswantoro Semarang*, 2015.

[17] Putra, Agung Auliaguntary Arif. 2016. Implementasi Text Summarization Menggunakan Metode Vector Space Model pada Artikel Berita Bahasa Indonesia. Skripsi. Jurusan Teknik Informatika. Fakultas Teknik dan Ilmu Komputer. Universitas Komputer Indonesia.

[18] Zuhri, Muhammad. 2011. Hadis Nabi Telaah Historis dan Metodologis. Yogyakarta: Tiara Wacana Yogya.