SKRIPSI

STUDI PERBANDINGAN METODE EKSTRAKSI FITUR, SELEKSI FITUR, DAN KLASIFIKASI PADA ANALISIS SENTIMEN SELAMA PEMILIHAN UMUM INDONESIA 2019

A COMPARATIVE STUDY OF FEATURES EXTRACTION, FEATURES SELECTION, AND CLASSIFIER METHODS ON SENTIMENT ANALYSIS DURING INDONESIAN ELECTION 2019

GAMA CANDRA TRI KARTIKA
15/378060/PA/16535

PROGRAM STUDI ILMU KOMPUTER
DEPARTEMEN ILMU KOMPUTER DAN ELEKTRONIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS GADJAH MADA
YOGYAKARTA

2019

SKRIPSI

STUDI PERBANDINGAN METODE EKSTRAKSI FITUR, SELEKSI FITUR, DAN KLASIFIKASI PADA ANALISIS SENTIMEN SELAMA PEMILIHAN UMUM INDONESIA 2019

A COMPARATIVE STUDY OF FEATURES EXTRACTION, FEATURES SELECTION, AND CLASSIFIER METHODS ON SENTIMENT ANALYSIS DURING INDONESIAN ELECTION 2019

Diajukan untuk memenuhi salah satu syarat memperoleh derajat
Sarjana Komputer

GAMA CANDRA TRI KARTIKA
15/378060/PA/16535

PROGRAM STUDI ILMU KOMPUTER
DEPARTEMEN ILMU KOMPUTER DAN ELEKTRONIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS GADJAH MADA
YOGYAKARTA

2019

HALAMAN PENGESAHAN

SKRIPSI

STUDI PERBANDINGAN METODE EKSTRAKSI FITUR, SELEKSI FITUR, DAN KLASIFIKASI PADA ANALISIS SENTIMEN SELAMA PEMILIHAN UMUM INDONESIA 2019

Telah dipersiapkan dan disusun oleh

GAMA CANDRA TRI KARTIKA
15/378060/PA/16535

Telah disetujui
pada tanggal 8 Agusstus 2019

Sri Mulyana, Drs., M Kom.
Pembimbing

PERNYATAAN

Dengan ini kami menyatakan bahwa dalam skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar Sarjana di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 8 Agusstus 2019

GAMA CANDRA TRI KARTIKA
15/378060/PA/16535

Pembimbing

Sri Mulyana, Drs., M Kom.

Karya ini kupersembahkan kepada
Ibu Sri Wahyuni dan Bapak Sugiono,
Mas Alfa Adib Ashidiqie dan Mbak Uthie Alawiyah,
Mas Betha Purba Praj Rahmatika dan Mbak Alinda Bela Fazrina,
Seluruh teman-teman dari SD 1 Purwosari, SMP 2 Kendal, SMA 1 Semarang, dan
Program Studi Ilmu Komputer UGM.

"Hidup itu seperti memanah. Semakin kamu tarik anak panahmu, semakin kuat anak panahmu melaju. Begitu pula hidup, semakin kamu belajar dari masa lalu, semakin kuat kamu menyongsong masa depan. Namun jika kamu terlalu kuat saat menarik anak panahmu, maka tali pada busur akan putus dan anak panahmu tidak akan bisa melaju. Begitu pula juga hidup, jika kamu terlalu berkutat pada masa lalu, maka kamu tidak akan pernah bergerak ke masa depan"

(Gama Candra Tri K)

"When do you think people die? When they are shot through the heart by the bullet of a pistol? No. When they are ravaged by an incurable disease? No. When they drink a soup made from a poisonous mushroom!? No! It's when they are forgotten."

(Dr. Hiluluk)

"We don't have time to waste asking for things we don't have. We can only look for the best way to fight with the things we have for our whole life."

(Hiruma Yoichi)

"The problem is not the problem. The problem is your attitude about the problem."

(Jack Sparrow)

"Worry is a misuse of imagination"

(Dan Zadra)

PRAKATA

Segala puji dan syukur semata-mata hanya untuk Allah SWT, karena atas segala rahmat, taufik, hidayah dan bantuan-Nya maka skripsi dengan judul "Studi Perbandingan Metode Ekstraksi Fitur, Seleksi Fitur, dan Klasifikasi Pada Analisis Sentimen Selama Pemilihan Umum Indonesia 2019" ini dapat selesai disusun. Sholawat serta salam tak lupa senantiasa tercurahkan kepada Nabi Muhammad SAW beserta keluarga, para sahabat, dan pengikut ajarannya yang telah membawa umat ini dari alam kegelapan menuju alam terang benerang seperti sekarang ini.

Telah banyak bantuan yang diperoleh selama dalam penulisan skripsi ini. Untuk itu tak lupa penulis ucapkan terima kasih yang sebesar-besarnya kepada

1. Ibu Sri Wahyuni, Bapak Sugiono, Kakak saya Alfa Adib Ashiddiqie dan Betha Purba Praj Rahmatika atas bantuan dan doanya,
2. Bapak Sri Mulyana, Drs., M Kom. selaku sebagai Dosen Pembimbing Skripsi yang telah sabar, meluangkan waktu, tenaga dan pikiran dalam membimbing penulis untuk menyelesaikan Skripsi,
3. Bapak Edi Winarko, Drs., M.Sc.,Ph.D. dan Bapak Agus Sihabudin, S.Si., M.Kom. selaku sebagai Dosen Penguji yang telah memberi banyak masukan pada Skripsi ini,
4. Mas Ryan, Mas Felix, Ejak, dan Bima atas referensi dan bantuannya atas tema analisis sentimen,
5. Tata, Sholihin, dan Wulan sebagai teman-teman satu bimbingan yang ikut membantu bertukar pendapat dan pengalaman baik dalam ide maupun penulisan penelitian ini
6. Teman-teman lab Sistem Cerdas yang berbagi ilmu, pendapat, informasi dan berdiskusi mengenai Skripsi masing-masing,
7. Tim penguji yang memberikan kritik dan saran yang sangat membangun dalam penyempurnaan skripsi ini,
8. Seluruh dosen Fakultas Matematika dan Ilmu Pengetahuan Alam, khususnya program studi Ilmu Komputer yang telah memberikan banyak sekali ilmu yang bermanfaat,

9. Keluarga besar Himakom dan OmahTI terutama divisi **Knowledge Discovery Community** yang telah menjadi keluarga kedua dan memberikan pembelajaran yang sangat berarti bagi penulis selama masa kuliah, baik bekerja sama dalam acara, mengasah keterampilan yang sangat mendukung perkuliahan, dan tak lupa canda tawa yang telah dilewati bersama-sama,

10. Teman-teman Ilmu Komputer 2015 yang banyak memberikan pengalaman dalam melakukan banyak hal baik suka maupun duka selama perkuliahan,

11. Josua, Lantang, Ucup, Bagas, Ozi, Devni, Eza, Muammar, Ijun, Alex, Riza, Syamil, Randy, Ruqi, Kego, Sora, dan Pram sebagai teman-teman bermain yang bermain bersama saat mengisi waktu luang dalam melakukan banyak hal baik suka maupun duka selama mengisi waktu luang,

12. Pihak-pihak lain yang tidak dapat disebutkan satu per satu.

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan, maka saran dan kritik yang konstruktif dari semua pihak sangat diharapkan demi penyempurnaan selanjutnya. Akhirnya hanya kepada Allah SWT kita kembalikan semua urusan dan semoga skripsi ini dapat bermanfaat bagi semua pihak, khususnya bagi penulis dan para pembaca pada umumnya, semoga Allah SWT meridhoi dan dicatat sebagai ibadah disisi-Nya. Apabila terdapat saran, kritik, atau pertanyaan mengenai penelitian ini mohon untuk menghubungi penulis melalui surat elektronik `candragctk17@gmail.com`. Semoga skripsi ini dapat bermanfaat bagi kita semua dan lebih khusus lagi bagi pengembangan dan penerapan ilmu komputer dalam berbagai bidang.

Yogyakarta, 8 Agusstus 2019

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

ABSTRAKSI

STUDI PERBANDINGAN METODE EKSTRAKSI FITUR, SELEKSI FITUR, DAN KLASIFIKASI PADA ANALISIS SENTIMEN SELAMA PEMILIHAN UMUM INDONESIA 2019

Oleh

GAMA CANDRA TRI KARTIKA
15/378060/PA/16535

Data dari media sosial seperti Twitter dan situs portal berita seperti Line-Today merupakan data yang objektif untuk diolah dikarenakan penggunanya yang sangat aktif dalam menyampaikan pendapatnya di kedua media sosial dan portal berita tersebut. Namun perbandingan metode dalam analisis sentimen belum banyak diterapkan dalam domain Bahasa Indonesia dan dataset Twitter dan Line-Today.

Tujuan dari penelitian ini adalah untuk mengetahui nilai terbaik dari metode ekstraksi fitur, seleksi fitur, dan metode pengklasifikasian serta pemilihan kombinasi yang terbaik dari proses seleksi fitur dan ekstraksi fitur, terhadap nilai analisis sentimen dengan berbagai metode pengklasifikasian.

Hasil penelitian ini adalah pada dataset Line-Today, pengklasifikasi terbaik pada masing-masing rata-rata semua nilai adalah Extra Tree Classifier (akurasi = 84.01 %), Support Vector Machine (presisi = 82.6 %), Decision Tree (*recall* = 74.69 %), Extra Tree Classifier (*f1 score* = 77.76 %) dan Multinomial Naive Bayes (waktu *running* = 6 detik).

Kemudian pada dataset Twitter, pengklasifikasi terbaik pada masing-masing rata-rata semua nilai adalah Support Vector Machine (akurasi = 81.04 %), Support Vector Machine (presisi = 84.4 %), Extra Tree Classifier (*recall* = 85.13 %), Extra Tree Classifier (*f1 score* = 81.71 %) dan Multinomial Naive Bayes (waktu *running* = 12.69 detik).

Kata-kata kunci : analisis sentimen, seleksi fitur, ekstraksi fitur, count vectorizer, term frequency–inverse document frequency, chi square, analysis of variance, mutual information, random forest, support vector machine, decision tree, extra tree classifier, k-nearest neighbour, multinomial naive bayes, gaussian naive bayes, logistic regression, neural network, adaboost, gradient boosting, *cross-validation*.

ABSTRACT

A COMPARATIVE STUDY OF FEATURES EXTRACTION, FEATURES SELECTION, AND CLASSIFIER METHODS ON SENTIMENT ANALYSIS DURING INDONESIAN ELECTION 2019

By

GAMA CANDRA TRI KARTIKA
15/378060/PA/16535

Data from social media such as Twitter and online news portal such as Line-Today are objective data to be processed because users are very active in expressing their opinions on both social media and online news portal. But the comparison of methods in sentiment analysis has not been widely applied in the Indonesian language, Twitter and Line-Today dataset .

The purpose of this study was to determine the best value of feature extraction methods, feature selection methods, and classification methods and the selection of the best combination of feature selection and feature extraction processes, to the value of sentiment analysis with various classification methods.

The results of this study are in the Line-Today dataset, the best classifiers in each average of all values are Extra Tree Classifier (accuracy = 84.01 %), Support Vector Machine (precision = 82.6 %), Decision Tree (recall = 74.69 %), Extra Tree Classifier (f1 score = 77.76 %) and Multinomial Naive Bayes (running time = 6 seconds).

Then on the Twitter dataset, the best classifiers in each average of all values are Support Vector Machine (accuracy = 81.04 %), Support Vector Machine (precision = 84.4 %), Extra Tree Classifier (recall = 85.13 %), Extra Tree Classifier (f1 score = 81.71 %) and Multinomial Naive Bayes (running time = 12.69 seconds).

Keywords: sentiment analysis, feature selection, feature extraction, count vectorizer, term frequency–inverse document frequency, chi square, analysis of variance, mutual information, random forest, support vector machine, decision tree, extra tree classifier, k-nearest neighbour, multinomial naive bayes, gaussian naive bayes, logistic regression, neural network, adaboost, gradient boosting, cross-validation.

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Media sosial adalah teknologi yang menggunakan komputer secara interaktif yang dapat memfasilitasi untuk membuat serta berbagi informasi, ide, opini, dan bentuk ekspresi lain melalui komunitas dan jaringan virtual seperti internet. Menurut survei yang dilakukan oleh Asosiasi Penyelenggara Jasa Internet Indonesia (APJII), Pada tahun 2017, jumlah pengguna internet di Indonesia mencapai 143,26 juta jiwa. Angka tersebut meningkat 7.95 persen dibandingkan pada tahun sebelumnya, yang mencapai 132,7 juta jiwa (Yuniarni, 2018).

Indonesia merupakan negara demokrasi yang melaksanakan pemilihan umum (Pemilu) setiap 5 tahun sekali untuk memilih pemimpin beserta perangkat legislatif lainnya. Sampai saat ini, pemilu masih digunakan sebagai inti dari proses demokrasi di Indonesia. Pada pemilu 2019 yang telah dilakukan, sudah ditentukan kedua calon presiden dan wakil presiden yaitu calon pertama Ir. H. Joko Widodo dengan wakilnya Prof. Dr. K. H. Ma'ruf Amin dan calon kedua Letnan Jenderal (Purn.) H. Prabowo Subianto Djojohadikusumo dengan pasangannya H. Sandiaga Salahuddin Uno, B.B.A., M.B.A.

Pengaruh media sosial sangat besar dalam dunia politik. Sejak tahun 2008, para pasangan calon presiden Amerika Serikat telah menggunakan media sosial seperti Facebook dan Twitter untuk menggalang dukungan. Di Indonesia, pada pemilu presiden RI tahun 2014 lalu, tim kampanye pasangan calon dan juga para pendukungnya dengan gencar menggunakan media sosial dengan mengunggah beragam video, foto atau pun status seputar pilpres melalui media sosial. Salah satu calon presiden, Joko Widodo, bahkan sudah menggunakan media sosial sebagai media kampanye ketika mencalonkan diri sebagai gubernur DKI Jakarta pada 2012 lalu. Hal ini pun tetap diikuti pada masa kampanye pemilu 2019 dengan tagar-tagar unik di Twitter yang menunjukkan dukungan pada masing-masing calon.

Analisis sentimen merupakan ilmu yang berguna untuk menganalisis pendapat seseorang, sentimen seseorang, evaluasi seseorang, sikap seseorang dan emosi seseorang ke dalam bahasa tertulis. Teknik sentimen dapat mendukung beberapa skenario seperti sentimen yang diskrit yang terdiri 1 dan 0 atau sentimen yang bernilai

kontinyu.

Telah banyak metode statistik yang dilakukan untuk memprediksi elektabilitas tokoh ataupun partai politik. Misalnya dengan survey, namun hasil survei yang dihasilkan oleh lembaga survey terkadang tidak sesuai dengan kenyataan dan seringkali digunakan untuk mengarahkan opini sehingga tidak netral oleh salah satu kandidat. Terdapat juga *quick count*, namun quick count membutuhkan waktu yang lama dan dilaksanakan setelah pemilihan umum. Pada penelitian yang lain, metode analisis sentimen dengan berbagai algoritme telah diusulkan untuk mengetahui dan mengevaluasi elektabilitas tokoh atau partai politik yang dilaksanakan sebelum pemilu (Lestari et al., 2017).

Preprocessing perlu dilakukan untuk mengetahui didalam penelitian apakah memang baik dilakukan atau tidak dalam penelitian analisis sentimen, sehingga terlihat perbandingan akurasi dari data yang sudah dinormalisasi dan *stemming* (Saputra et al., 2015). Dengan melihat hasil akurasi, dapat disimpulkan bahwa penggunaan *preprocessing* perlu dilakukan untuk mendapatkan hasil akurasi yang lebih optimal.

Performa model klasifikasi menjadi bagian penting dalam proses klasifikasi. Hal ini menunjukan seberapa akurat sistem dapat mengklasifikasikan data dengan benar. Salah satu metode untuk meningkatkan akurasi dengan seleksi fitur. Seleksi fitur adalah proses mereduksi fitur-fitur yang dianggap tidak relevan dalam proses klasifikasi yang akan menimbulkan *overfitting*. Jika seleksi fitur TF-IDF memperhitungkan jumlah kemunculan fitur saja, seleksi fitur *chi square* menggunakan metode statistika untuk mengukur independensi sebuah *term* dengan kategorinya, tidak sebatas kemunculan fitur saja. (Lestari et al., 2017)

Metode rekayasa fitur untuk klasifikasi sentimen *tweet* sering menghasilkan sejumlah besar fitur. Sejumlah besar contoh dataset yang dihasilkan dapat memiliki dimensi yang sangat tinggi. Selain itu, pengklasifikasi pada dataset berukuran besar secara komputasi adalah mahal. Seleksi fitur yang mendapat sedikit perhatian dalam riset klasifikasi sentimen *tweet*, memilih *subset* fitur yang optimal, yang mengurangi dimensi dataset, membantu mengurangi biaya komputasi, dan meningkatkan kinerja klasifikasi (Prusa et al., 2015).

Perbandingan metode pengklasifikasian telah dilakukan pada algoritme klasifikasi seperti Naïve Bayes (NB), Support Vector Machine (SVM), dan Artificial Neural Network (ANN) diusulkan oleh banyak peneliti untuk digunakan pada analisis sentimen *review* film (Chandani dan Wahono, 2015). Berbagai metode seleksi fitur dan metode ekstraksi fitur serta berbagai metode pengklasifikasian seperti De-

cision Tree (DT), K-Nearest Neighbour (KNN), Naive Bayes (NB), Support Vector Machine (SVM) pada dataset Twitter juga dilakukan (Shah dan Patel, 2016). Sebuah studi komprehensif membandingkan algoritme pengklasifikasian (Naïve Bayes, Support Vector Machine, Logistik Regression dan Random Forest) dilakukan oleh (Onan et al., 2016).

Pemilih yang menggunakan Twitter dan portal berita memiliki karakteristik kritis, mandiri, independen, rasional dan mendukung perubahan (Sukendar et al., 2017). 738 jumlah sampel responden sampel adalah 56 % per perempuan dan 44 % laki-laki dengan rentang usia 20-45 tahun. Semua responden yang berpartisipasi dalam survei adalah pengguna Line yang menginstall aplikasi di perangkat pengguna Line (*Indonesian LINE User 2016 - Survey Report* 2016). Oleh sebab itu penelitian ini mengusulkan topik analisis sentimen pada data Twitter dan berita dari Line-Today untuk mengetahui sentimen tokoh politik dalam periode waktu tertentu berdasarkan sentimen (positif dan negatif) masing masing tokoh politik.

1.2 Rumusan Masalah

Data dari media sosial seperti Twitter dan situs portal berita seperti Line-Today merupakan data yang objektif untuk diolah dikarenakan penggunanya yang sangat aktif dalam menyampaikan pendapatnya di kedua media sosial dan portal berita tersebut. Namun perbandingan metode dalam analisis sentimen belum banyak diterapkan dalam domain Bahasa Indonesia dan dataset Twitter dan Line-Today. Penelitian dalam perbandingan berbagai macam metode analisis sentimen diperlukan sebagai acuan untuk penelitian lain kedepannya. Berdasarkan hal tersebut, penelitian ini akan melakukan studi perbandingan ekstraksi fitur, seleksi fitur, dan metode klasifikasi pada *sentiment analysis* pada pemilihan umum Indonesia 2019.

1.3 Batasan Masalah

Dari rumusan masalah yang telah diuraikan. Beberapa batasan masalah dalam penelitian ini adalah sebagai berikut.

1. Penelitian ini hanya membahas tentang Pemilihan Umum 2019 dan tokoh-tokoh seperti Joko Widodo, Ma'ruf Amin, Prabowo Subianto, dan Sandiaga Uno.

2. Dataset yang digunakan berasal dari Twitter dan Line-Today.

3. Metode ekstraksi fitur menggunakan Term Frequency–Inverse Document Frequency (TF-IDF) dan Count Vectorizer.

4. Metode pengklasifikasian menggunakan Random Forest, Support Vector Machine, Decision Tree, Extra tree, K-Nearest Neighbour, Multinomial Naive Bayes, Gaussian Naive Bayes, Logistic Regression, Neural Network dengan Multi-Layer Perceptron, Ada Boost, dan Gradient Boosting.

5. Metode seleksi fitur menggunakan Chi Square, Analysis of Variance (ANOVA), dan Mutual Information

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah di atas, tujuan dari penelitian ini adalah untuk mengetahui nilai terbaik dari metode ekstraksi fitur, seleksi fitur, dan metode pengklasifikasian serta pemilihan kombinasi yang terbaik dari proses seleksi fitur dan ekstraksi fitur, terhadap nilai analisis sentimen dengan berbagai metode pengklasifikasian.

1.5 Manfaat Penelitian

Berdasarkan pada tujuan penelitian yang telah diuraikan di atas, maka manfaat penelitian adalah sebagai berikut.

1. Mengetahui perbandingan performa dari metode ekstrasi fitur seperti Count Vectorizer dan TFIDF

2. Mengetahui perbandingan performa dari metode seleksi fitur seperti Chi Square, ANOVA, dan Mutual Information

3. Mengetahui perbandingan performa dari metode pengklasifikasi seperti Random Forest, Support Vector Machine, Decision Tree, Extra tree, K-Nearest Neighbour, Multinomial Naive Bayes, Gaussian Naive Bayes, Logistic Regression, Neural Network dengan Multi-Layer Perceptron, Ada Boost, dan Gradient Boosting.

4. Mengetahui dampak pada metode pengklasifikasi pada penggunaan ekstraksi fitur dan seleksi fitur

5. Referensi baru dalam penelitian berdasarkan data dari Twitter dan Line-Today mengenai pemilihan umum 2019 di Indonesia.

6. Dapat digunakan sebagai acuan dalam penelitian berikutnya dan referensi penelitian berikutnya.

1.6 Metodologi Penelitian

Tahapan yang dilakukan dalam penelitian ini secara umum adalah

1.6.1 Studi Literatur

Pada tahap ini dilakukan pengumpulan informasi informasi yang mendukung penelitian ini. Informasi yang telah dikumpulkan mencakup penjelasan mengenai analisis sentimen, s*crapping data*, metode ekstraksi fitur, metode seleksi fitur, metode pengklasifikasian, dan metode evaluasi k-fold cross validation untuk menguji akurasi, presisi, *recall*, dan *f-measure* model. Informasi tersebut didapatkan dari berbagai sumber seperti jurnal ilmiah, paper, website, buku, skripsi dan sebagainya.

1.6.2 Pengumpulan Data

Penelitian ini menggunakan data dari Twitter dan Line-Today yang diambil menggunakan beberapa alat untuk *scrapping* dan alat untuk mendapatkan komentar dengan *request* komentar dari artikel berita. Setiap data kemudian dilabeli secara manual untuk sentimen positif atau negatifnya.

1.6.3 Perancangan Sistem

Proses *preprocessing* mengubah data yang tidak terstruktur menjadi data terstruktur agar lebih mudah diolah. Hal ini dilakukan untuk agar data yang memiliki banyak *noise* dapat diproses dan dianalis oleh sistem. Proses yang dilakukan antara lain *case folding* untuk merubah menjadi huruf kecil, *punctuation removal* untuk menghilangkan simbol yang tidak diperlukan, *whitespace removal* untuk menghapus menghapus spasi awal dan akhir, *stopword removal* untuk menghapus kata-kata tidak memiliki makna, dan *stemming* untuk mengubah semua kata menjadi bentuk kata dasarnya. Lalu dilakukan proses ekstraksi fitur menggunakan Count Vectorizer dan TFIDF. Dilanjutkan dengan proses seleksi fitur menggunakan Chi Square, ANOVA,

dan MI. Kemudian melakukan klasifikasi semua jenis data hasil ekstraksi fitur dan seleksi fitur dengan semua masing-masing model pengklasifikasi Random Forest, Support Vector Machine, Decision Tree, Extra tree, K-Nearest Neighbour, Multinomial Naive Bayes, Gaussian Naive Bayes, Logistic Regression, Neural Network dengan Multi-Layer Perceptron, Ada Boost, dan Gradient Boosting.

1.6.4 Implementasi Sistem

Pada tahap ini dilakukan implementasi algoritma dan skenario yang sudah dirancang dalam bentuk program komputer. Program tersebut akan dibuat dengan menggunakan bahasa pemrograman Python3 dengan menggunakan Jupyter Notebook atau Google Colab.

1.6.5 Pengujian

Pengujian akan dilakukan dengan menggunakan metode *10-fold cross validation*. Secara *stratified* data akan dibagi menjadi 10 bagian. Kemudian satu bagian akan diujikan sebagai data uji, sedangkan bagian lainya akan digunakan sebagai data latih. Pengujian dilakukan sebanyak sepuluh kali dengan digunakan bagian yang berbeda-beda sebagai data ujinya. Hasil tersebut kemudian di rata-rata dari setiap pengujian yang dilakukan. Hasil pengujian tersebut kemudian dimasukan akurasi, presisi, *recall*, dan *f-measure*nya.

1.6.6 Penulisan Laporan

Tahapan penulisan laporan dilakukan sejalan dengan berjalannya penelitian.

1.7 Sistematika Penulisan

I BAB 1 : PENDAHULUAN
Bab ini memuat latar belakang penelitian, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

II BAB 2 : TINJAUAN PUSTAKA
Bab ini memuat penelitian-penelitian terdahulu yang berkaitan dengan topik permasalahan, metode yang digunakan, dan referensi yang diacu oleh penelitian ini.

III BAB 3 : LANDASAN TEORI

Bab ini memuat dasar teori-teori yang digunakan dalam penelitian ini.

IV BAB 4 : ANALISIS DAN PERANCANGAN SISTEM

Bab ini memuat analisis pengambilan data, *preprocessing*, ekstraksi fitir, seleksi fitur, klasifikasi, dan validasi.

V BAB 5 : IMPLEMENTASI

Bab ini membahas implementasi rancangan sistem, yaitu memuat kode implementasi pengambilan data, *preprocessing*, ekstraksi fitir, seleksi fitur, klasifikasi, dan validasi beserta penjelasannya.

VI BAB 6 : HASIL DAN PEMBAHASAN

Bab ini membahas eksperimen perbandingan data, ekstraksi fitur, seleksi fitur dan klasifikasi. Bab ini juga membahas perbandingan performa model terbaik yang diperoleh dari hasil validasi.

VII BAB 7 : PENUTUP

Bab ini memuat kesimpulan dari penelitian yang telah dilakukan beserta saran untuk penelitian selanjutnya.

BAB II

TINJAUAN PUSTAKA

Komparasi algoritme klasifikasi antara Support Vector Machine (SVM), Naïve Bayes (NB) dan Artificial Neural Network (ANN) didapatkan SVM dengan hasil terbaik dengan nilai accuracy = 81.10% dan nilai Area Under Curve (AUC) = 0.904. Hasil dari komparasi algoritme seleksi fitur antara Information Gain, Chi Square, Forward Selection, Backward Elimination didapatkan information gain pada parameter *top* k dengan nilai k = 200 sebagai hasil terbaik, dengan nilai rata-rata akurasi adalah 84.57% dan nilai AUC = 0.899 (Chandani dan Wahono, 2015).

Analisis sentimen dan klasifikasi tokoh publik pada Twitter dilakukan menggunakan metode Naïve Bayes. Tokoh publik yang dipilih adalah tokoh publik yang dianggap layak dan memiliki kemampuan untuk menjadi pemimpin. Naïve Bayes dikombinasikan dengan fitur sehingga dapat mendeteksi negasi dan menggunakan pembobotan Term Frequency dan Term Frequency–Inverse Document Frequency (TF-IDF). Selain metode Naïve Bayes juga digunakan metode Support Vector Machine (SVM) . Hasil klasifikasi berupa sentimen positif dan negatif dengan kategori tokoh politik berdasarkan kapabilitas, integritas, dan akseptabilitas tokoh tersebut. Dari proses pengklasifikasian sentimen dan kategori tokoh politik SVM lebih unggul dari Naïve Bayes (Hidayatullah dan Azhari, 2015).

Sepuluh teknik seleksi fitur berbasis lter dilakukan dan membandingkannya dengan tidak digunakannya atau digunakannya seleksi fitur di empat pengklasifikasi yang berbeda. Teknik-teknik ini digunakan untuk memilih sepuluh bagian fitur yang berbeda dari dataset yang terdiri dari 3000 *tweet* dari sentimen 140 *corpus*. Eksperimen menunjukkan bahwa seleksi fitur dapat secara signifikan meningkatkan kinerja klasifikasi untuk semua pengklasifikasi. Menggunakan 200 fitur terbaik, tetapi 100 dan 150 fitur terbaik juga dilakukan dengan cara yang sama. 75 fitur terbaik memang mengungguli yang lain, tetapi performa yang dilakukan lebih buruk dari 100 hingga 200 fitur terbaik. Menggunakan seleksi fitur untuk memilih 50 atau lebih sedikit fitur umumnya menghasilkan kinerja yang buruk, lebih rendah daripada tidak menggunakan seleksi fitur. Signifikansi statistik dari temuan penelitian tersebut diuji dengan melakukan Analysis of Variance (ANOVA) (Prusa et al., 2015).

Analisis sentimen dari Presiden Indonesia periode 2014-2019 Joko Widodo dilakukan dengan metode yang digunakan dalam pengambilan data menggunakan Se-

arch Techniques. Setelah itu proses *preprocessing* dilakukan dengan cara *stemming* menggunakan pustaka Sastrawi, tokenisasi N-gram, *stopword removal* dan mempertahankan emoticon. Metode klasifikasi yang digunakan hanya menggunakan SVM dengan akurasi yang terbaik dalam penelitian ini dengan dilakukan normalisasi dan *stemming* pada data sebesar 89,2655% menggunakan metode SVM, dan kemudian data yang dinormalisasi saja sebesar 88,7006% menggunakan metode SVM (Saputra et al., 2015).

Penelitian untuk menguji performa prediksi dari lima metode ekstraksi kata statistik (ekstraksi kata berbasis frekuensi, TF-IDF, ekstraksi kata berbasis informasi kejadian, ekstraksi kata kunci berbasis eksentrisitas dan algoritme Text Rank) pada klasifikasi algoritme dan metode *ensemble* untuk klasifikasi teks dokumen ilmiah. Dalam penelitian ini, sebuah studi komprehensif membandingkan algoritme pengklasifikasi dasar (Naïve Bayes, Support Vector Machine, Logistic Regression dan Random Forest) dengan lima metode *ensemble* yang sering digunakan (AdaBoost, Bagging, Dagging, Random Subspace dan Majority Voting) dilakukan. Sejauh pengetahuan penelitian ini adalah analisis empiris pertama, yang mengevaluasi efektivitas metode ekstraksi kata kunci statistik dalam hubungannya dengan algoritme pengklasifikasi *ensemble*. Skema klasifikasi dibandingkan dalam hal akurasi klasifikasi, F-Measure atau *F1 Score* dan AUC. Untuk memvalidasi analisis empiris, uji Analysis of Variance (ANOVA) dua arah digunakan. Analisis eksperimental menunjukkan bahwa *ensemble* Bagging dari Random Forest dengan metode ekstraksi kata kunci berbasis frekuensi menghasilkan hasil yang menjanjikan untuk klasifikasi teks. Untuk pengumpulan dokumen Association for Computing Machinery (ACM), kinerja prediksi rata-rata tertinggi (93,80%) diperoleh dengan pemanfaatan metode ekstraksi kata kunci berdasarkan frekuensi dilakukan dengan *ensemble* Bagging dari algoritme Random Forest. Secara umum, hasil SVM dan Random Forest menjanjikan. Analisis empiris menunjukkan bahwa pemanfaatan representasi kata kunci berbasis dokumen teks dalam hubungannya dengan pengklasifikasi *ensemble* dapat meningkatkan kinerja prediktif dan skalabilitas skema klasifikasi teks, yang praktis penting dalam bidang aplikasi klasifikasi teks (Onan et al., 2016).

Komparasi seleksi fitur seperti Document Frequency, Information Gain, Gini Index, Chi Square Statistic, Best term, Ambiguity measure, dan Distinguished feature selector, serta berbagai metode pengklasifikasian seperti Decision Tree (DT), K-Nearest Neighbour (KNN), Naive Bayes (NB), Support Vector Machine (SVM) pada dataset Twitter juga telah dilakukan. Hasil dari peneltian tersebut menghasilk-

an tabel perbandingan kelebihan dan kekurangan dari masing-masing metode seleksi fitur, ekstraksi fitur dan metode pengklasifikasian (Shah dan Patel, 2016).

Metode untuk memprediksi hasil pemilu Presiden Amerika Serikat 2016 dengan melakukan analisis sentimen pada media sosial juga telah dilakukan. Media sosial yang digunakan adalah Twitter dengan *tweet* berbahasa Inggris. Ada tiga metode klasifikasi yang digunakan dalam penelitian ini, Binary Multinomial Naive Bayes, SentiWordNet, dan AFINN-11. Setelah itu, ketiga metode tersebut dibandingkan akurasinya. AFINN-111 dan Binary Multinomial Naive Bayes keduanya memberikan skor yang bagus. Namun, Binary Multinomial Naive Bayes dipilih untuk digunakan karena menghasilkan nilai F1-score yang lebih tinggi daripada AFINN-111 (Wicaksono, 2016).

Teks mining dilakukan diantaranya menggunakan metode pengklasifikasian Support Vector Machine (SVM), Naïve Bayessian (NB) dan K-Nearest Neighbor (KNN). Ketiga algoritme ini akan dibandingkan untuk mengetahui performa yang baik dalam hal akurasi untuk dua dataset yang berbeda yaitu Internet Movie Database (IMDB) *review* film dan sentimen Twitter. Hasil dari komparasi menunjukkan SVM memperoleh hasil yang baik dalam akurasi pada dataset IMDB review film 78,55% dan pada dataset Twitter 72%. Sama halnya dengan NB yang memperoleh akurasi pada data Twitter 78.55% tetapi berbeda pada data Twitter 67,33%. Hasil F-Measure *review* film menunjukan SVM dan NB memperoleh hasil yang sama yaitu 0,785 dan untuk hasil Area Under Curve (AUC), NB mengungguli hasil dengan nilai 0,869. SVM memperoleh hasil 0,786 sedangkan KNN memperoleh hasil 0,572 pada AUC. Pada nilai f-measure untuk Twitter, SVM lebih unggul dengan memperoleh hasil 0,720 dan NB memperoleh hasil 0,673 sedangkan KNN 0,545. Untuk hasil AUC, sama seperti dataset IMDB, pada dataset Twitter, NB juga mengungguli SVM dan K-NN. AUC untuk NBC memperoleh hasil 0,735, SVM memperoleh hasil 0,658 dan KNN memperoleh hasil 0,618 (Ipmawati, 2017).

Penelitian untuk membandingkan kinerja empat algoritme pemilihan fitur populer (Document Frequency, Chi Statistic, Information Gain dan Ratio Gain) dan lima algoritme pengklasifikasi mesin populer (Decision Tree, Naïve Bayes, Support Vector Machine, Radial Basis Function Neural Network dan K-Nearest Neighbour) dalam klasifikasi sentimen multi-kelas. Eksperimen dilakukan pada tiga set data publik yang mencakup dua belas *subset* data, dan Cross-Validation 10 kali lipat digunakan untuk memperoleh akurasi klasifikasi mengenai setiap kombinasi algoritme seleksi fitur, algoritme pengklasifikasian, ukuran set fitur dan *subset* data. Berdasarkan

3600 data yang diperoleh, akurasi klasifikasi (4 algoritme seleksi fitur x 5 algoritme pengklasifikasi × 15 fitur set ukuran × 12 *subset* data), rata-rata akurasi klasifikasi masing-masing algoritme dihitung, dan uji Wilcoxon digunakan untuk memverifikasi perbedaan antara algoritme yang berbeda dalam klasifikasi sentimen multi-kelas. Hasil penelitian menunjukkan bahwa, dalam hal akurasi klasifikasi, Ratio Gain mendapatkan performa terbaik di antara empat algoritme seleksi fitur dan Support Vector Machine mendapatkan performa terbaik di antara lima algoritme pengklasifikasian. Dalam hal waktu eksekusi, perbandingan serupa juga dilakukan. Hasil yang diperoleh akan sangat penting untuk lebih meningkatkan klasifikasi sentimen multi-kelas yang ada dan mengembangkan klasifikasi sentimen multi-kelas baru (Liu et al., 2017).

Penelitian dengan melakukan analisis sentimen menggunakan data *tweet* dan berita dari masing-masing tokoh politik untuk mengetahui elektabilitasnya menggunakan metode Multinomial Naive Bayes. Tokoh politik yang digunakan dalam penelitian adalah 10 tokoh politik yang dianggap populer di Indonesia. Dataset yang digunakan berjumlah 16.523 data latih dan 6.550 data uji. Data *tweet* didapatkan menggunakan alat Tweet Catcher dan berita didapatkan dari 3 situs berita di Indonesia yaitu tribunnews.com, tempo.co, dan viva.co.id menggunakan alat *scrapper* dalam kurun waktu 17 November 2016 sampai 1 November 2017. Setelah data terkumpul, dilakukan tahap *preprocessing* dan *filtering*. Lalu dilakukan seleksi top-n kata fitur menggunakan metode Chi Square dan TF-IDF. Selanjutnya adalah pembentukan model klasifikasi dan proses testing dengan membandingkan hasil elektabilitas tiap tokoh politik tanpa seleksi fitur dan dengan seleksi fitur Chi Square dan TF-IDF. Hasil penelitian ini menunjukan bahwa nilai performa model menggunakan metode seleksi fitur Chi Square lebih tinggi dengan rata-rata nilai akurasi 85,24%, presisi 88,84%, *recall* 91,65% dan f-measure 90,17% dibandingkan dengan menggunakan metode seleksi fitur TF-IDF dengan rata-rata nilai akurasi 78,11%, presisi 87,41%, recall 87,79% dan f-measure 87,54% serta jika dibandingkan tanpa seleksi fitur dengan nilai rata-rata akurasi 74,69% , presisi 87,40%, recall 84,88% dan f-measure 84,72% (Suryotomo, 2018). Untuk perbandingan penelitian ditampilkan pada Tabel 2.1.

Tabel 2.1: Perbandingan dengan Penelitian Sebelumnya

No.	Peneliti	Penelitian	Metode Penelitian	Perbedaan
01	Chandani dan Wahono, 2015	Komparasi algoritme klasifikasi Machine Learning dan feature selection pada analisis sentimen review film	Metode pengklasifikasian menggunakan SVM, NB dan ANN	Sumber dataset, metode seleksi fitur dan metode pengklasifikasian
02	Hidayatullah dan Azhari, 2015	Analisis sentiment dan klasfikasi tokoh publik berdasarkan data di twitter	Klasifikasi menggunakan SVM dan Naive Bayes serta ekastraksi fitur menggunakan TFIDF	Sumber dataset, metode seleksi fitur dan metode pengklasifikasian
03	Prusa et al., 2015	Impact of feature selection techniques for tweet sentiment classification	Seleksi fitur dengan Chi-Squared, Threshold-Based Feature Selection, Gini-Index, Kolmogorov-Smirnov, Mutual Information, Probability Ratio, Precision-Recall Curve, dan Receiver Operating Characteristic curve serta metode klasifikasi KNN, DT, LR, dan NN	Sumber dataset, metode seleksi fitur dan metode pengklasifikasian

04	Saputra et al., 2015	Analisis sentimen data presiden Jokowi dengan preprocessing normalisasi dan stemming menggunakan metode naive bayes dan SVM	Metode preprocessing menggunakan Sastrawi dan metode pengklasifikasian menggunakan Naive Bayes dan Support Vector Machine	Sumber dataset, metode seleksi fitur dan metode pengklasifikasian
05	Onan et al., 2016	Ensemble of keyword extraction methods and classifiers in text classification	Metode pengklasifikasian Naïve Bayes, Support Vector Machine, Logistic Regression dan Random Forest dengan lima metode ensemble (AdaBoost, Bagging, Dagging, Random Subspace dan Majority Voting	Sumber dataset dan metode pengklasifikasian

06	Shah dan Patel, 2016	A review on feature selection and feature extraction for text classification	seleksi fitur (Document Frequency, Information Gain, Gini Index, Chi Square Statistic, Best term, Ambiguity measure, dan Distinguished feature selector), serta metode pengklasifikasian (Decision Tree (DT), K-Nearest Neighbour (KNN), Naive Bayes (NB), Support Vector Machine (SVM))	Metode pengklasifikasian
07	Wicaksono, 2016	A proposed method for predicting US presidential election by analyzing sentiment in social media	Metode pengklasifikasian Binary Multinomial Naive Bayes, SentiWordNet, dan AFINN-11	Sumber dataset dan metode pengklasifikasian
08	Ipmawati, 2017	Komparasi Teknik Klasifikasi Teks Mining Pada Analisis Sentimen	Metode pengklasifikasian Support Vector Machine (SVM), Naïve Bayessian (NB) dan K-Nearest Neighbor (KNN)	Sumber dataset dan metode pengklasifikasian

09	Liu et al., 2017	Multi-class sentiment classification: The experimental comparisons of feature selection and machine learning algorithms	Metode pengklasifikasian Decision Tree, Naïve Bayes, Support Vector Machine, Radial Basis Function Neural Network dan K-Nearest Neighbour	Sumber dataset dan metode pengklasifikasian
10	Suryotomo, 2018	Analisis sentimen untuk mengetahui elektabilitas tokoh politik menggunakan metode multinomial naive bayes	Metode seleksi fitur TFIDF dan Chi Square serta metode pengklasifikasian Support Vector Machine dan Multinomial Naive Bayes	Sumber dataset, metode ekstraksi fitur dan metode pengklasifikasian

BAB III
LANDASAN TEORI

3.1 Information Retrieval

Information Retrieval (IR) merupakan ilmu yang mempelajari metode dan prosedur untuk menemukan kembali informasi yang tersimpan dari berbagai sumber yang relevan atau koleksi sumber informasi yang dicari atau dibutuhkan. Proses-proses dalam IR dapat berupa pembuatan index (*indexing*), panggilan (*searching*), dan pemanggilan data kembali (*recalling*) (Manning et al., 2010).

Karena ketersediaan beragam sumber daya di internet, IR sangat subjektif pada jenis media, format data yang didukung oleh media, dan jenis analisis yang diperlukan. Beberapa situs *microblogging* seperti Twitter, Sina-Wiebo menyediakan Application Programming Interface (API) mereka untuk mengumpulkan data publik dari situs mereka. Twitter telah menyediakan API REST Twitter untuk mendapatkan data statis seperti informasi profil pengguna, dan Streaming API2 untuk mendapatkan data *streaming* seperti *tweets*. Demikian pula dengan Facebook telah menyediakan Facebook Graph API4. API ini membantu dalam mengekstrak *post* dan informasi lain dari Facebook (Ravi dan Ravi, 2015).

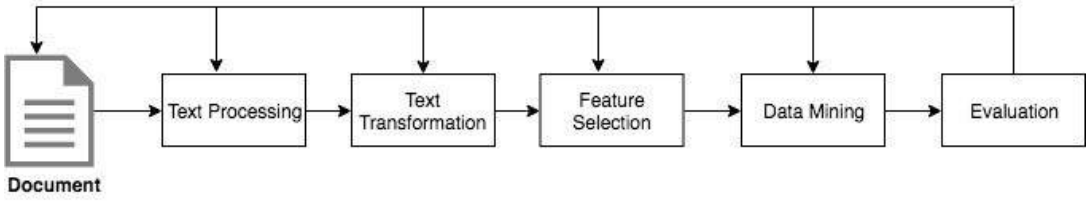
3.2 Text Mining

Text mining adalah suatu bidang didalam *data mining* dimana penggalian informasi dan pengelolaan sekumpulan dokumen menggunakan alat analisis. Gagasan utama dari *text mining* adalah mengetahui cakupan atau topik dari permasalahan dalam teks. Pengambilan informasi dari teks (text mining) antara lain dapat meliputi kategorisasi teks atau dokumen, analisis sentimen, pencarian topik yang lebih spesifik, serta *spam filtering*. *Text mining* penting dalam analisis sentimen sebagai pengidentifikasi emosional pada suatu pernyataan, sehingga banyak studi tentang analisis sentimen dilakukan (Manning et al., 2010).

Berbeda dengan data mining yang biasanya memproses data terstruktur, *text mining* biasanya digunakan untuk memproses *unstructured* atau minimal *semi-structured data*. Akibatnya *text mining* mempunyai tantangan tambahan yang tidak ditemui di *data mining* seperti struktur data yang kompleks dan tidak lengkap, arti yang tidak jelas, tidak baku, bahasa yang berbeda serta terjemahan yang tidak aku-

rat. Oleh karena itu biasanya Natural Language Processing (NLP) digunakan untuk memproses *unstructured data text* tersebut (Adiwijaya, 2006).

Gambar (3.1) menjelaskan bagaimana proses yang terjadi didalam *text mining*.



Gambar 3.1: Proses Dalam Text Mining Ravi dan Ravi, 2015

Proses *text mining* dimulai dengan *text preprocessing* yaitu pengubahan bentuk dokumen menjadi data terstruktur dengan cara seperti *tokenization*, *stopword removal*, *stemming*, normalisasi dan lain sebagainya. Dari hasil *text processing* selanjutnya dilakukan *text transformation* untuk menemukan fitur-fitur yang tersimpan didalam data sesuai kebutuhan yang diperlukan. Setelah diketahui fitur-fiturnya selanjutnya dilakukan seleksi fitur untuk menentukan fitur yang berpengaruh atau tidak dalam pemodelan data menggunakan pemeringkatan atau pembobotan fitur. Setelah itu, digunakan berbagai algoritme *data mining* untuk menemukan informasi atau pola yang menarik dari data yang terpilih. Selanjutnya dilakukan evaluasi model apakah pola atau informasi yang dihasilkan bertentangan dengan fakta atau hipotesa sebelumnya (Adiwijaya, 2006).

3.3 Sentiment Analysis

Sentiment Analysis (SA) adalah bidang dalam penelitian yang menggunakan berbagai teknik seperti Natural Language Processing (NLP), Information Retrieval (IR), dan Data Mining (DM) yang digunakan untuk scara sistematis mengidentifikasi, mengekstrak, mengukur, dan mempelajari keadaan afektif dan informasi yang subjektif. Untuk menghadapi data teks yang tidak terstruktur, metode tradisional NLP sepert *information retrieval* dan *information extraction* muncul. Untuk mendapatkan pengertian dari teks yang diekstraksi, banyak upaya penelitian telah dilakukan dalam beberapa tahun terakhir yang mengarah ke *Automated* SA, area penelitian NLP yang lain (Ravi dan Ravi, 2015).

3.4 Preprocessing

Preprocessing adalah tahap penting yang dilakukan untuk membersihkan data atau mengubah data menjadi bentuk data yang terstruktur. Proses membersihkan data meliputi pengecekan data yang tidak konsisten, menghapus data yang terduplikat dan mengkoreksi kesalahan yang terjadi saat penulisan teks (Wikarsa dan Thahir, 2015).

Data mentah yang diperoleh dari berbagai sumber yang sering kali perlu diproses terlebih dahulu sebelum meluncurkan analisis sepenuhnya secara matang. Beberapa langkah preprocessing yang umum adalah : *tokenization, stop word removal, stemming, part of speech (POS) tagging*, dan *feature extraction and representation* (Ravi dan Ravi, 2015).

3.4.1 Tokenization

Tokenization adalah perintah untuk memotong menjadi bagian-bagian kecil, yang disebut *Token*, disaat yang sama membuang karakter tertentu, seperti tanda baca. *Token* ini sering disebut sebagai term atau kata, tetapi kadang-kadang penting untuk membuat perbedaan tipe atau *token*. *Token* adalah turunan dari urutan karakter dalam beberapa dokumen tertentu yang dikelompokkan bersama sebagai unit semantik yang berguna untuk diproses. Tipe adalah kelas dari semua *token* yang berisi urutan karakter yang sama. Suatu term adalah tipe (yang mungkin dinormalisasi) yang termasuk dalam kamus sistem IR. Himpunan term indeks dapat sepenuhnya berbeda dari *token*, misalnya, mereka bisa menjadi pengidentifikasi semantik dalam taksonomi, tetapi dalam praktiknya dalam sistem IR modern sangat terkait dengan *token* dalam dokumen. Namun, alih-alih menjadi *token* yang muncul dalam dokumen, mereka biasanya diturunkan dari mereka dengan berbagai proses normalisasi. *Token* yang tidak diindeks (*Stopwords*) bukan term, dan jika beberapa *token* dihancurkan bersama melalui normalisasi, mereka diindeks sebagai satu term, di bawah formulir dinormalisasi. Secara konseptual, pemisahan pada ruang kosong juga dapat membagi apa yang seharusnya dianggap sebagai *token* tunggal (Manning et al., 2010).

3.4.2 Stopwords Removal

Stopword removal adalah tahap pemilihan kata kata penting dari hasil *token*, yaitu kata apa saja yang akan digunakan untuk mewakili dokumen. *Stopword* adalah kata-kata yang tidak deskriptif (tidak bermakna) yang dapat dibuang dengan pende-

katan *bag of words* (database kumpulan kata kata yang tidak deskriptif/tidak bermakna), kemudian jika hasil *tokenization* itu terdapat kata tidak bermakna dalam data tersebut, maka hasil *tokenisasi* itu dibuang. Biasanya performa *text mining* ataupun IR dapat ditingkatkan dengan *stopword removal* (Vijayarani et al., 2015).

3.4.3 Stemming

Stemming adalah proses pengubahan bentuk kata menjadi kata dasar atau tahap mencari akar kata dari tiap hasil. Dengan dilakukannya proses *stemming* setiap kata berimbuhan akan berubah menjadi kata dasar, dengan demikian dapat lebih mengoptimalkan proses *text mining*. Terdapat 2 poin penting yang dipertimbangkan dalam proses *stemming*.

1. Kata yang tidak memiliki makna yang sama lebih baik disimpan terpisah.
2. Bentuk morfologi dari suatu kata yang memiliki makna dasar yang sama lebih baik dipetakan kedalam akar yang sama.

Dua aturan ini cukup baik digunakan dalam *teks mining* atau *language processing*. *Stemming* biasanya dipertimbangkan sebagai *recall-enhancing device*. Untuk bahasa yang relatif simpel morfologinya, *stemming* tidak dapat bekerja optimal dibandingkan untuk bahasa yang kompleks morfologinya. Kebanyakan eksperimen *stemming* ini diaplikasikan untuk bahasa inggris (Vijayarani et al., 2015).

3.5 Term Frequency-Inverse Document Frequency

Term frequency adalah total frekuensi munculnya sebuah kata *term* dalam *corpus*. Untuk menghitung *term frequency*, melibatkan jumlah semua kejadian dari kata dalam semua dokumen dalam corpus. Untuk lebih jelasnya, rumus untuk mencari nilai term frequency dapat dilihat pada persamaan (3.1).

$$tf(t_i d_j) = \frac{f_{ij}}{max(f(w,d) : \ w \in \ d} \tag{3.1}$$

dimana :

f_{ij} = frekuensi kemunculan kata t_i pada dokumen d_j

w = nilai maksimum yang dihitung menggunakan frekuensi dari seluruh term yang muncul pada dokumen d_j

Inverse document frequency (IDF) adalah nilai yang menyatakan bahwa semakin jarang sebuah *term* muncul dalam dokumen-dokumen yang ada didalam *corpus*, maka semakin relevan *term* tersebut. Metode IDF ditambahkan karena **term frequency** dinilai terlalu sederhana dalam mengukur tingkat pentingnya sebuah term karena tidak melibatkan informasi secara global dalam **corpus**. IDF dapat membantu dalam membedakan satu dokumen dengan dokumen-dokumen lainnya (Siddiqi dan Sharan, 2015).

$$idf(t_i d_j) = \log(\frac{|N|}{1 + |d \in \ D : t_i \in \ d|}) \tag{3.2}$$

dimana :
$|N|$ = jumlah total seluruh dokumen
$|d \in D : t_i \in d|$ = banyaknya dokumen dimana suatu kata (t_i) muncul

Untuk menghitung TF-IDF, maka hal yang dilakukan adalah mengalikan nilai dari term frequency dengan nilai IDF dari suatu term tersebut. Rumus dari TF-IDF dapat dilihat pada persamaan (3.3)

$$tf - idf(t_i d_j) = tf(t_i d_j) \times idf(t_i d_j) \tag{3.3}$$

dimana :
$tf - idf(t_i d_j)$ = bobot TF-IDF kata ke-i dalam dokumen d_j
$tf(t_i d_j)$ = term frequency kata ke-i dalam dokumen d_j
$idf(t_i d_j)$ = inverse document frequency kata ke-i dalam dokumen d_j

3.6 Chi Square

Seleksi fitur digunakan untuk mereduksi fitur yang tidak relevan dalam proses klasifikasi. Seleksi fitur Chi Square menggunakan teori statistika untuk menguji independensi sebuah term dengan kategorinya. Seleksi fitur Chi Square dilakukan dengan cara mengurutkan setiap berdasarkan fitur berdasarkan hasil seleksi fitur Chi Square dari nilai yang terbesar hingga terkecil. Nilai seleksi fitur Chi Square yang lebih besar dari nilai signifikan menunjukan penolakan hipotesis independensi. Sedangkan jika dua peristiwa menunjukan dependen, maka fitur tersebut menyerupai atau sama

dengan label kategori sesuai pada kategori (Ling et al., 2014). Persamaan Chi Square dapat dilihat pada persamaan (3.4)

$$X^2(D,t,c) = \sum_{et=0,1} \sum_{ec=0,1} \frac{(N_{etec} - E_{etec})^2}{E_{etec}} \tag{3.4}$$

dimana :

$X^2(D,t,c)$ = merupakan nilai Chi Square dari dokumen pada term t untuk kelas c

N_{etec} = *observed value* (jumlah term t pada kelas c) E_{etec} = expected value (jumlah *term* t pada kelas c)

Sementara contoh untuk mencari nilai salah satu **expected value** dapat dilihat pada persamaan (3.5)

$$E_{11} = N \times P(t) \times P(c) = N \times \frac{N_{11} + N_{10}}{N} \times \frac{N_{11} + N_{01}}{N} \tag{3.5}$$

dimana :

N = jumlah dokumen

N_{11} = Jumlah kemunculan term t pada kelas c

N_{10} = Jumlah kemunculan term t pada kelas bukan c

N_{11} = Jumlah pada kelas c yang memuat term t

N_{01} = Jumlah pada kelas c yang tidak memuat term t

3.7 F-test Analysis of Variance

F-test adalah uji statistik yang dimana statistik uji memiliki distribusi F di bawah hipotesis nol. Ini paling sering digunakan ketika membandingkan model statistik, untuk mengidentifikasi model yang paling cocok dengan populasi dari mana data sampel. F-test telah diterapkan ke data menggunakan kuadrat terkecil. F-test dalam Analysis of Variance (ANOVA) menilai menilai apakah nilai-nilai yang diharapkan dari variabel kuantitatif dalam beberapa kelompok yang telah ditentukan berbeda satu sama lain. Sebagai contoh, anggaplah data berupa teks. F-test ANOVA dapat digunakan untuk menilai apakah salah satu kata lebih unggul, atau lebih rendah, dibandingkan kata yang lain dengan hipotesis nol bahwa semua kata mengha-

silkan respons rata-rata yang sama. Ini adalah contoh dari tes "omnibus", yang berarti bahwa satu tes dilakukan untuk mendeteksi salah satu dari beberapa kemungkinan perbedaan. Keuntungan dari ANOVA F-test adalah bahwa kita tidak perlu menentukan sebelumnya kata mana yang akan dibandingkan, dan tidak perlu menyesuaikan untuk membuat beberapa perbandingan. Kerugian dari F-test ANOVA adalah bahwa jika kita menolak hipotesis nol, kita tidak tahu kata mana yang dapat dikatakan berbeda secara signifikan dari yang lain, juga, jika F-test dilakukan pada level α, dapatkah kita menyatakan bahwa pasangan perlakuan dengan perbedaan rata-rata terbesar secara signifikan berbeda pada level α (Winter, 2015). Persamaan F-test pada ANOVA dapat dilihat pada persamaan (3.6).

$$F = \frac{\sum_{i=1}^{K} \frac{n_i \ (\bar{Y}_i - \bar{Y})^2}{(K-1)}}{\sum_{i=1}^{K} \sum_{j=1}^{n_i} \frac{(Y_{ij} - \bar{Y}_i)^2}{(N-K)}} \tag{3.6}$$

dimana :
$\bar{Y}_i$ = menunjukkan sampel rata-rata dalam kelompok ke-i
n_i = bilangan observasi dari kelompok ke i
$\bar{Y}$ = menunjukan rata-rata dari data
K = menunjukan jumlah kelompok
Y_{ij} = observasi ke j di dalam kelompok K ke i
N = Keseluruhan ukuran sampel

3.8 Mutual Information

Metode seleksi fitur yang umum adalah untuk menghitung $A(t, c)$ sebagai *expected* Mutual Information (MI) dari term t dan kelas c. MI mengukur seberapa banyak informasi yang ada atau tidaknya suatu term berkontribusi untuk membuat keputusan klasifikasi yang benar pada c (Manning et al., 2010). Lebih jelasnya pada persamaan (3.7).

$$I(U;C) = \sum_{e_t(1,0)} \sum_{e_c(1,0)} P(U = e_t, C = e_c) \log_2 \frac{P(U = e_t, C = e_c)}{P(U = e_t)P(C = e_c)} \tag{3.7}$$

dimana :

U = variabel acak yang mengambil nilai dari e_t = 1 (dokumen yang memuat term t) dan e_t = 0 (dokumen yang tidak memuat t)

C = variabel acak yang mengambil nilai e_c = 1 (dokumen di dalam kelas c) dan e_c = 0 (dokumen yang tidak di dalam kelas c)

MI mengukur seberapa banyak informasi dalam pengertian teoretis informasi yang termuat tentang suatu kelas. Jika distribusi suatu term adalah sama di dalam kelas sebagaimana dalam koleksi secara keseluruhan, maka $I(U;C) = 0$. MI mencapai nilai maksimumnya jika term tersebut merupakan indikator sempurna untuk bagian kelas, terjadi, jika term hadir dalam dokumen jika dan hanya jika dokumen tersebut ada di kelas (Manning et al., 2010).

3.9 Support Vector Machine

Support Vector Machines (SVM) secara khusus mendefinisikan kriterianya untuk mencari permukaan keputusan yang jauh dari titik data. Jarak ini dari permukaan keputusan ke titik data terdekat menentukan margin dari classifier. Metode konstruksi ini tentu saja berarti bahwa fungsi keputusan untuk suatu SVM sepenuhnya ditentukan oleh suatu (biasanya kecil) subset data yang mendefinisikan posisi pemisah. Titik-titik ini disebut sebagai Support Vector (dalam ruang vektor, titik dapat dianggap sebagai vektor antara titik asal dan titik (Manning et al., 2010).

Dalam penelitian ini, implementasinya berdasarkan LIBSVM dengan C-Support Vector Classification (Chang dan Lin, 2011). Diberikan *vector training* $x_i \in R^n, i = 1, ..., l$, di dalam dua kelas dan suatu indikator vektor $y \in r^l$ yang meliputi $y_i \in (1, -1)$. C-SVC menyelesaikan masalah optimasi primal dengan persamaan (3.8).

$$\vec{w} = \sum_{i=1}^{n} c_i y_i \varphi(\vec{x}_i) \tag{3.8}$$

dimana :

w = variabel vektor

n = matriks *semidefinite* positif

$y_i \alpha_i$ = nama label *support vector*

$\varphi(x_i)$ = Pemetaan x_i ke ruang dimensi yang lebih tinggi

Koefisien c_i dapat diselesaikan dengan pemrograman kuadratik. Lalu, kita

dapat menemukan indeks i seperti $0 < c_i < (2n\lambda)^{-}1$. jadi $\varphi(\vec{x}_i)$ terletak pada batasan ruang transformasi.

3.10 Decision Tree

Decision Tree (DT) adalah struktur yang mirip diagarma alur di mana setiap internal node mewakili "tes" pada atribut, masing-masing cabang mewakili hasil pengujian, dan setiap *leaf node* mewakili label kelas (keputusan diambil setelah menghitung semua atribut). DT terdiri dari tiga jenis node yaitu Decision nodes, Chance nodes, dan End nodes (Kamiński et al., 2018).

3.11 Random Forest

Random Forest (RF) adalah penggolong yang terdiri dari kumpulan pengklasifikasi *tree-structured* $h(x, k), k = 1, ...$ di mana k adalah vektor acak independen yang terdistribusi secara identik dan masing-masing *tree* memberikan masukan untuk kelas paling populer di input x (Breiman, 2001).

RF menggunakan algoritme pembelajaran *tree* yang dimodifikasi yang memilih pada setiap kandidat yang terpecah dalam proses pembelajaran, suatu *subset* acak fitur. Proses ini kadang-kadang disebut *feature bagging*. Alasan melakukan ini adalah korelasi *tree* dalam sampel *bootstrap* : jika satu atau beberapa fitur adalah prediktor yang sangat kuat untuk variabel respons (target output), fitur ini akan dipilih di banyak pohon B, menyebabkannya menjadi berkorelasi. Analisis tentang bagaimana *bagging* dan proyeksi sub-ruang acak berkontribusi pada perolehan akurasi dalam kondisi yang berbeda (Ho, 2002).

3.12 Extra Tree Classifier

Extra Tree Classifier (ETC) berbeda dari DT biasanya dalam cara mereka dibangun. Ketika mencari pemisahan terbaik untuk memisahkan sampel sebuah *node* menjadi dua kelompok, pemisahan acak diambil untuk masing-masing fitur $max_features$ yang dipilih secara acak dan pemisahan terbaik di antara yang dipilih. Ketika $max_features$ diatur menjadi 1, ini sama dengan membangun DT yang benar-benar acak (Pedregosa et al., 2011).

ETC Menambahkan satu langkah lebih lanjut dari pengacakan menghasilkan RF. Meskipun mirip dengan RF biasa yang merupakan *ensemble tree* individu,

ada dua perbedaan utama : pertama, setiap *tree* dilatih menggunakan sampel pembelajaran keseluruhan (bukan sampel bootstrap), dan kedua, pemisahan atas-bawah di pembelajaran *tree* diacak. Alih-alih menghitung titik potong optimal secara lokal untuk setiap fitur yang sedang dipertimbangkan, titik potong acak dipilih. Nilai ini dipilih dari distribusi seragam dalam rentang empiris fitur (dalam set pelatihan *tree*). Kemudian, dari semua split yang dihasilkan secara acak, split yang menghasilkan skor tertinggi dipilih untuk membagi *node*. Mirip dengan RF biasa, jumlah fitur yang dipilih secara acak untuk dipertimbangkan pada setiap node dapat ditentukan. Nilai default untuk parameter ini adalah $\sqrt{n}$ untuk klasifikasi dan n untuk regresi, di mana n adalah jumlah fitur dalam model (Geurts et al., 2006).

3.13 Gaussian Naive Bayes

Naive Bayes adalah teknik sederhana untuk membangun pengklasifikasi model yang menetapkan label kelas untuk contoh masalah, direpresentasikan sebagai vektor nilai fitur, di mana label kelas diambil dari beberapa set hingga. Tidak ada algoritme tunggal untuk melatih pengklasifikasi tersebut, tetapi keluarga algoritme berdasarkan pada prinsip umum : semua pengklasifikasi Naive Bayes mengasumsikan bahwa nilai fitur tertentu tidak tergantung pada nilai fitur lain, mengingat variabel kelas. Misalnya, buah dapat dianggap apel jika warnanya merah, bulat, dan berdiameter sekitar 10 cm. Klasifikasi Naive Bayes mempertimbangkan setiap fitur ini untuk berkontribusi secara independen terhadap probabilitas bahwa buah ini adalah apel, terlepas dari kemungkinan korelasi antara warna, kebulatan, dan fitur diameter.

Saat menghadapi data kontinyu, asumsi biasanya adalah bahwa nilai kontinyu yang terkait dengan setiap kelas didistribusikan menurut distribusi normal (atau Gaussian). Misalnya, anggaplah data pelatihan berisi atribut kontinyu, x. Pertama-tama mengelompokkan data berdasarkan kelas dan kemudian menghitung rata-rata dan *variance* dari x di setiap kelas. Biarkan μ_k menjadi nilai rata-rata dari x yang menjadi bagian kelasC_k dan biarkan σ_k^2 menjadi *variance* dari nilai x yang menjadi bagian kelas C_k. Misalkan saat mengumpulkan nilai observasi v. Kemudian, *probabilitas* distribusi dari v pada kelas $C_k, p(x = v|C_K$, dapat dihitung dengan menerapkan v pada persamaan distribusi normal dengan parameter μ_k dan σ_k^2 (Hand dan Yu, 2001) dengan algoritme (3.9).

$$p(x = v|C_k) = \frac{1}{\sqrt{2\phi\sigma_k^2}} e^{-\frac{(v-\mu_k)^2}{2\sigma_k^2}} \tag{3.9}$$

dimana :
x = atribut kontinyu
μ_k = rata-rata nilai x yang terkait dengan kelas C_k
σ_k^2 = variance nilai x yang terkait dengan kelas C_k
v = nilai observasi

3.14 Multinomial Naive Bayes

Dengan model Multinomial, sampel (fitur dari vektor) mewakili frekuensi kejadian yang dihasilkan oleh Multinomial $(p_1, ..., p_n)$ dimana p_i dalam probabilitas i yang muncul (atau K Multinomial dalam kasus multi kelas). Fitur vektor $x = (x_1, ..., x_n)$ adalah histogram dengan x_i menghitung banyak kejadian dalam contoh tertentu. Ini adalh model yang bisanya digunakan untuk klasifikasi dokumen, dengan peristiwa yang mewakili kata dalam satu dokumen (Rennie et al., 2003). Kemungkinan dalam pengamatan observasi histogram x seperti persamaan (3.10).

$$p(x|C_k) = \frac{(\sum_i x_i)!}{\prod_i x_i!} \prod_i p_{ki}^{x_i} \tag{3.10}$$

dimana :
p_i = probabilitas terjadinya peristiwa i muncul
k = nilai multinomial dalam kasus multi-kelas
x_i = menghitung berapa kali peristiwa i yang diamati dalam contoh tertentu

3.15 K-Nearest Neighbour

Klasifikasi K-Nearest Neighbour (KNN), outputnya adalah bagian kelas. Suatu objek diklasifikasikan oleh nilai terdekat tetangganya, dengan objek yang diterapkan ke kelas paling umum di antara tetangganya yang terdekat (k adalah bilangan bulat positif). Jika k = 1, maka objek hanya ditugaskan untuk kelas tetangga terdekat itu. Pada KNN ditetapkan setiap dokumen ke kelas mayoritas k tetangga terdekat di mana k adalah parameter. Dasar dari klasifikasi KNN adalah bahwa, berdasarkan hipotesis

kontiguitas, dokumen uji d diharapkan memiliki label yang sama dengan dokumen pelatihan yang terletak di wilayah lokal di sekitar d (Manning et al., 2010).

Contoh pelatihan adalah vektor dalam ruang fitur multidimensi, masing-masing dengan label kelas. Fase pelatihan algoritme hanya terdiri dari penyimpanan vektor fitur dan label kelas dari sampel pelatihan. Dalam fase klasifikasi, k adalah konstanta yang ditentukan pengguna, dan vektor yang tidak berlabel (perintah atau titik uji) diklasifikasikan dengan menetapkan label yang paling sering di antara sampel pelatihan k yang terdekat dengan titik perintah itu. Perhitungan jarak yang umum digunakan untuk variabel kontinyu adalah *Euclidean distance*. Untuk variabel diskrit, seperti untuk klasifikasi teks, metrik lain dapat digunakan, seperti *overlap metric* (atau *Hamming distance*) (Jaskowiak dan Campello, 2011).

3.16 Logistic Regression

Logistic Regression (LR) adalah model statistik yang dalam bentuk dasarnya menggunakan fungsi logistik untuk memodelkan variabel dependen biner, meskipun ada ekstensi yang lebih kompleks. Konsep dasar matematika yang mendasari LR adalah Logit, logaritma natural dari ratio peluang. Secara umum, LR sangat cocok untuk menggambarkan dan menguji hipotesis tentang hubungan antara variabel hasil kategoris dan satu atau lebih variabel prediktor kategoris atau kontinyu. Dalam kasus paling sederhana dari regresi linear untuk satu prediktor kontinyu X dan satu variabel hasil pembagian Y, tampilan data tersebut menghasilkan dua garis paralel, masing-masing sesuai dengan nilai hasil pembagian. Karena dua garis paralel sulit untuk dijelaskan dengan persamaan regresi kuadrat terkecil biasa karena pembagian hasil, seseorang dapat membuat kategori untuk prediktor dan menghitung rata-rata variabel hasil untuk masing-masing kategori (Peng et al., 2002). LR sederhana memiliki bentuk pada persamaan (3.11) dan (3.12).

$$\text{logit}(Y) = natural\log(odds) = \ln(\frac{\phi}{1-\phi} = \alpha + \beta X \tag{3.11}$$

$$\phi = \text{Probability}(Y|X = x) = \frac{e^{\alpha+\beta x}}{1 + e^{\alpha+\beta x}} \tag{3.12}$$

dimana :

Y = hasil ketertarikan
x = nilai spesifik dari X
ϕ = probabilitas hasil dari kejadian
α = potongan Y
β = koefisien regresi
e = 2.71828

3.17 Multilayer Perceptron

Multilayer Perceptron (MLP) adalah kelas jaringan saraf tiruan *feedforward*. MLP terdiri dari setidaknya tiga lapisan node : *input layer*, *hidden layer*, dan *output layer*. Kecuali untuk input node, setiap node adalah neuron yang menggunakan fungsi aktivasi nonlinear. Jika sebuah MLP memiliki fungsi aktivasi linier di semua neuron, yaitu fungsi linier yang memetakan input tertimbang ke output masing-masing *neuron*, maka aljabar linier menunjukkan bahwa sejumlah lapisan dapat dikurangi menjadi input dua lapis model keluaran. Dalam MLP beberapa *neuron* menggunakan fungsi aktivasi nonlinear yang dikembangkan untuk memodelkan frekuensi potensial aksi, atau menembakkan, *neuron* biologis. Dalam konteks peneltian ini, menggunakan fungsi aktivasi Rectified Linear Unit (ReLU) (Glorot et al., 2011). Dalam konteks jaringan syaraf tiruan, ReLU adalah fungsi aktivasi yang didefinisikan sebagai bagian positif dari argumennya seperti persamaan (3.13).

$$f(x) = x^+ = \max(0, x) \tag{3.13}$$

dimana :
x = input neuron

Pembelajaran terjadi dalam perceptron dengan mengubah bobot koneksi setelah setiap bagian data diproses, berdasarkan jumlah kesalahan dalam output dibandingkan dengan hasil yang diharapkan. Ini adalah contoh *supervised learning*, dan dilakukan melalui *backpropagation*, generalisasi dari algoritme kuadrat terkecil dalam *perceptron linier*. Kita dapat merepresentasikan derajat error dalam *output node* j dalam titik data ke-n (contoh pembelajaran) oleh $e_j(n) = d_j(n) - y_j(n)$, di mana d adalah nilai target dan y adalah nilai yang dihasilkan oleh perceptron. Bobot *node* kemudian dapat disesuaikan berdasarkan koreksi yang meminimalkan kesalahan dalam

seluruh keluaran, yang diberikan oleh persamaan (3.14).

$$\varepsilon(n) = \frac{1}{2}e_j^2(n) \tag{3.14}$$

dimana :

j = node output pada titik data ke-n

Menggunakan *gradient descent*, perubahan bobotnya menjadi persamaan (3.15).

$$\Delta w_{ji}(n) = -\eta\frac{\delta\varepsilon(n)}{\eta v_j(n)}y_i(n) \tag{3.15}$$

dimana :

y_i = keluaran neuron sebelumnya

η = *learning rate*

Turunan yang akan dihitung tergantung pada bidang lokal yang diinduksi v_j, yang itu sendiri bervariasi. Sangat mudah untuk membuktikan bahwa untuk simpul keluaran turunan ini dapat disederhanakan menjadi persamaan (3.16).

$$-\frac{\delta\varepsilon(n)}{\delta v_j(n)} = e_j(n)\phi'(v_j(n)) \tag{3.16}$$

dimana :

ϕ' = turunan dari fungsi aktivasi

Analisis lebih sulit untuk perubahan bobot *hidden node*, tetapi dapat ditunjukkan bahwa turunan yang relevan seperti persamaan (3.17).

$$-\frac{\delta\varepsilon(n)}{\delta v_j(n)} = \phi'(v_j(n))\sigma_k - \frac{\delta\varepsilon(n)}{\delta v_j(n)}w_{kj}(n) \tag{3.17}$$

Ini tergantung pada perubahan bobot dari *node* ke-k, yang mewakili *hidden ouput*. Jadi untuk mengubah bobot *hidden node*, bobot *output node* berubah sesuai dengan turunan dari fungsi aktivasi, dan karenanya algoritme ini mewakili backpropagation dari fungsi aktivasi (Haykin, 1994).

3.18 Gradient Bosting

Gradient Boosting (GB) adalah teknik pembelajaran mesin untuk masalah regresi dan klasifikasi, yang menghasilkan model prediksi dalam bentuk model prediksi *ensemble*. GB membangun model dalam mode per langkah seperti metode *boosting* lainnya lakukan, dan itu menggeneralisasikan dengan optimalisasi *loss function* (Friedman, 2002).

Seperti metode peningkatan lainnya, GB menggabungkan pengklasifikasian yang "lemah" menjadi pengklasifikasi tunggal yang kuat dalam cara yang berulang-ulang. Lebih jelasnya dalam pengaturan regresi kuadrat terkecil, di mana tujuannya adalah untuk melatih model F untuk memprediksi nilai-nilai dari bentuk $\hat{y} = F(x)$ dengan meminimalkan *mean squared error* $\frac{1}{n}\sum_i(\hat{y}_i - y_i)^2$, dimana i mengindeks beberapa set pelatihan ukuran n dari nilai aktual dari variabel keluaran y (Li, 2016).

Pada setiap tahap m, $1 \leq m \leq M$, dari GB, dapat diasumsikan bahwa ada beberapa model yang tidak sempurna F_m. algoritme GB meningkatkan F_m dengan membangun model baru yang menambahkan *estimator* h untuk memberikan model yang lebih baik : $F_{m+1}(x) = F_m(x) + h(x)$. Untuk menemukan $h4$, solusi GB dimulai dengan pengamatan h sempurna seperti $F_{m+1}(x) = F_m(x) + h(x) = y$ atau setara $h(x) = y - F_m(x)$ (Li, 2016)

Oleh karena itu, GB akan cocok dengan h sisa dari $y - F_m(x)$. Seperti pada *boosting* lainnya, setiap F_{m+1} mencoba untuk memperbaiki kesalahan pendahulunya F_m. Inti dari ide ini untuk *loss function* selain dari kuadrat kesalahan, dan untuk masalah klasifikasi dan pemeringkatan, mengikuti dari pengamatan bahwa residual $y - F(x)$ untuk model yang diberikan adalah gradien negatif dari *squared error loss function* $\frac{1}{2}(y - F(x))^2$. Jadi, GB adalah algoritme *gradient descent*, dan intinya berarti memasukkan kerugian yang berbeda dan gradiennya (Li, 2016).

3.19 Adaptive Boosting

Adaptive Boosting (AdaBoost), adalah meta-algoritme pembelajaran mesin yang dirumuskan oleh Yoav Freund dan Robert Schapire. Dapat digunakan bersama dengan banyak jenis algoritme pembelajaran lainnya untuk meningkatkan kinerja. Keluaran dari algoritme pembelajaran lainnya (pengklasifikasi dasar) digabungkan menjadi jumlah terbobot yang mewakili hasil akhir dari pengklasifikasi yang dikuatkan. AdaBoost adaptif dalam arti bahwa pengklasifikasi dasar berikutnya disesuaikan

untuk contoh-contoh yang salah diklasifikasi oleh pengklasifikasi sebelumnya. AdaBoost sensitif terhadap data dan outlier. Dalam beberapa masalah itu rentan terhadap masalah *overfitting* dari algoritme pembelajaran lainnya. Pengklasifikasi secara individu dapat menjadi lemah, tetapi selama kinerja masing-masing sedikit lebih baik daripada menebak secara acak, model akhir dapat terbukti menyatu dengan pengklasifikasi yang kuat (Freund dan Schapire, 1997).

Proses pelatihan AdaBoost hanya memilih fitur-fitur yang diketahui untuk meningkatkan kemampuan prediksi model, mengurangi dimensi dan berpotensi meningkatkan waktu eksekusi karena fitur-fitur yang tidak relevan tidak perlu dihitung. AdaBoost mengacu pada metode khusus pelatihan *boosting*. Pengklasifikasi *boosting* berbentuk seperti persamaan (3.18).

$$F_T(x) = \sum_{t=1}^{T} f_t(x) \tag{3.18}$$

di mana setiap f_t adalah pengklasfikasi yang lemah yang mengambil objek x sebagai input dan mengembalikan nilai yang menunjukkan kelas objek. Sebagai contoh, dalam masalah dua kelas, tanda keluaran pengklasifikasi yang lemah mengidentifikasi kelas objek yang diprediksi dan nilai absolut memberikan kepercayaan pada klasifikasi itu. Demikian pula, pengklasifikasi ke-T positif jika sampel berada dalam kelas positif dan negatif sebaliknya. Setiap pengklasifikasi yang lemah menghasilkan hipotesis keluaran, $h(x_i)$, untuk setiap sampel dalam set pelatihan. Pada setiap iterasi t, pengklasifikasi yang lemah dipilih dan diberi koefisien α_t sedemikian sehingga jumlah kesalahan pelatihan E_t dari hasil pengklasifikasi tingkatkan t diminimalkan. Lebih jelasnya pada persamaan (3.19).

$$E_t = \sum_{i} E[F_{t-1}(x_i) + \alpha_t h(x_i)] \tag{3.19}$$

Di sini $F_{t-1}(x)$ adalah pengklasifikai yang dikuatkan yang telah dibangun hingga tahap pelatihan sebelumnya, $E(F)$ adalah beberapa fungsi *error* dan $f_t(x) = \alpha_t h(x)$ adalah pengklasifikasi yang lemah yang sedang dipertimbangkan untuk penambahan pengklasifikasi akhir.

Pada setiap iterasi dari proses pelatihan, bobot $w_{i,t}$ diterapkan untuk setiap sampel dalam set pelatihan yang sama dengan error saat ini $E(F_{t-1}(x_i))$ pada sampel

itu. Bobot ini dapat digunakan untuk menginformasikan pelatihan pengklasifikasi yang lemah (Freund dan Schapire, 1997).

3.20 Pengujian

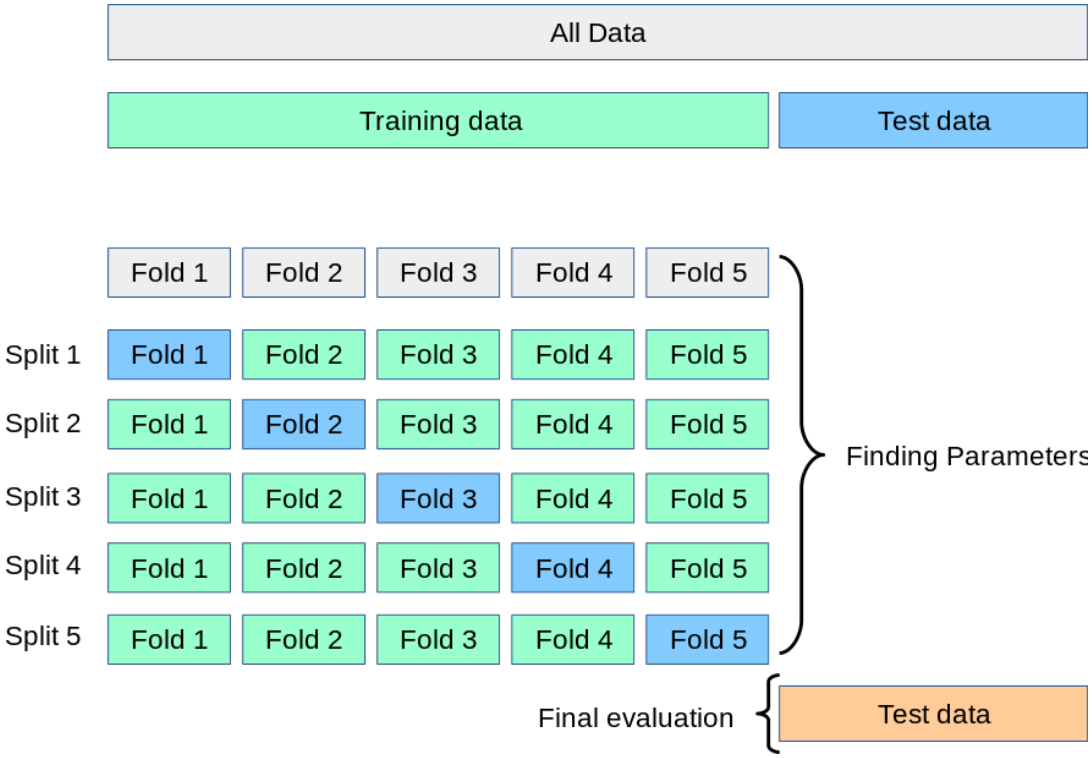
Pengujian dilakukan untuk menghitung performa dari sistem atau dalam penelitian ini performa klasifikasi sentimen. Untuk menghitung performa klasifikasi, data latih digunakan untuk melakukan validasi. Validasi dapat dilakukan dengan menggunakan metode yang dinamakan K-Fold Cross Validation.

3.20.1 K-Fold Cross Validation

Cross Validation merupakan salah satu metode pengujian dalam data mining dan machine learning. Dalam melakukan *cross validation*, dataset dibagi menjadi beberapa bagian atau dalam hal ini disebut *fold* secara acak. Satu bagian dari hasil pembagian tersebut digunakan sebagai data pengujian dan sisanya digunakan sebagai data pelatihan. Pengujian dilakukan sejumlah bagian yang ada dengan bergantian menggunakan bagian yang berbeda sebagai data pengujian. Hasil pengujian yang dihitung adalah hasil keseluruhan yaitu merata-rata hasil dari keseluruhan pengujian yang dilakukan (Witten, 2011).

Cross Validation yang dilakukan dengan pembagian sebanyak k disebut sebagai k-fold cross validation. Jumlah bagian yang umum digunakan untuk pengujian adalah sebanyak sepuluh. Pengujian banyak dilakukan dengan menggunakan 10-fold cross validation dikarenakan dari hasil percobaan menunjukkan 10-fold cross validation memberikan perkiraan kesalahan dari klasifikasi yang terbaik. Pengujian k-fold cross validation yang dilakukan sekali sering tidak memberikan perkiraan kesalahan yang bisa diandalkan. Hal ini disebabkan pengujian yang dilakukan beberapa kali dengan parameter yang sama dapat memberikan hasil yang berbeda yang disebabkan oleh unsur random yang ada pada saat pembagian data. Oleh karena itu untuk mendapatkan hasil yang dapat diandalkan pengujian harus dilakukan berkali-kali dengan hasilnya merupakan rata- rata dari semua hasil pengujian.

Contoh penggunaan k-fold cross validation dengan k=5 bisa dilihat pada ilustrasi gambar (3.2).



Gambar 3.2: Ilustrasi k-fold cross validation dengan k=5 Pedregosa et al., 2011

3.21 Perhitungan Performa

Model klasifikasi adalah pemetaan dari suatu input data menjadi suatu output yang merupakan prediksi kelas. Klasifikasi yang hanya menghasilkan dua kelas sebagai outputnya disebut klasifikasi biner. Kedua kelas tersebut sering kali merupakan kelas positif dan kelas negatif. Terdapat empat kemungkinan yang terjadi dari proses pengklasifikasian biner seperti yang dijabarkan oleh Fawcett, 2006 seperti pada gambar (3.2).

- True Positive (TP) : Prediksi menghasilkan true, dan kejadiannya true. Sebagai contoh jika model memprediksi seseorang mendapat sentimen positif dan pada kenyataannya mereka memiliki sentimen positif.
- False Positive (FP) : Prediksi menghasilkan true, dan kejadiannya false. Sebagai contoh jika model memprediksi seseorang mendapat sentimen positif dan pada kenyataannya mereka memiliki sentimen negatif.
- True Negative (TN) : Prediksi menghasilkan false, dan kejadiannya false. Seba-

gai contoh jika model memprediksi seseorang mendapat sentimen negatif dan pada kenyataannya mereka memiliki sentimen negatif.

- False Negative (FN) : Prediksi menghasilkan false, dan kejadiannya false. Sebagai contoh jika model memprediksi seseorang mendapat sentimen negatif dan pada kenyataannya mereka memiliki sentimen positif.

Hasil klasifikasi biner pada suatu data set dapat direpresentasikan pada suatu matriks 2x2 yang disebut confusion matrix seperti yang ditunjukkan pada gambar (3.3).

		Kelas Data	
		Positive	Negative
Prediksi kelas	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Gambar 3.3: Ilustrasi Confussion Matrix

Terdapat beberapa rumus umum yang digunakan untuk menghitung performa dari klasifikasi seperti berikut.

- Accuracy (Akurasi) : perbandingan kasus yang diidentifikasi benar dengan jumlah semua kasus. Persamaan (3.20) menunjukkan rumus untuk accuracy.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.20}$$

- Precision/PPV : proporsi kasus yang diidentifikasi dengan benar sebagai milik kelas ‘a’ di antara semua kasus yang diklasifikasikan oleh pengklasifikasi bahwa mereka termasuk dalam kelas ‘a’. Persamaan (3.21) menunjukkan rumus untuk menghitung precision.

$$Precision = \frac{TP}{TP + FP} \tag{3.21}$$

- Recall/Sensitivity : proporsi kasus yang diidentifikasi dengan benar sebagai milik kelas ‘a’ di antara semua kasus yang benar-benar termasuk dalam kelas ‘a’. Persamaan (3.22) menunjukkan rumus untuk menghitung recall.

$$Precision = \frac{TP}{TP + FN} \tag{3.22}$$

- F1 score/F-measure : bobot rata-rata antara precision dan recall. Persamaan (3.23) menunjukkan rumus F1 score.

$$F1score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3.23}$$

BAB IV
ANALISIS DAN PERANCANGAN

4.1 Analisis Sistem

Data dari media sosial seperti Twitter dan situs penampung berita seperti Line-Today merupakan data yang objektif untuk diolah dikarenakan penggunanya yang sangat aktif dalam menyampaikan pendapatnya di kedua media sosial tersebut. Namun perbandingan metode dalam analisis sentimen belum banyak diterapkan dalam domain Bahasa Indonesia dan dataset Twitter. Penelitian dalam perbandingan berbagai macam metode analisis sentimen diperlukan sebagai acuan untuk penelitian lain kedepannya. Oleh karena itu, pada penelitian ini dilakukan studi perbandingan ekstraksi fitur, seleksi fitur, dan metode klasifikasi pada *sentiment analysis* selama pemilihan umum Indonesia 2019.

Analisis sentimen yang dilakukan pada penelitian yang dilakukan untuk mengetahui nilai terbaik dari metode ekstraksi fitur, seleksi fitur, dan metode pengklasifikasian serta pemilihan kombinasi yang terbaik dari proses seleksi fitur dan ekstraksi fitur, terhadap nilai analisis sentimen dengan berbagai metode pengklasifikasian dengan data Twitter dan Line-Today yang diambil menggunakan calon presiden dan calon wakil presiden dalam pemilihan umum Indonesia tahun 2019 yaitu Joko Widodo, Ma'ruf Amin, Prabowo Subianto dan Sandiaga Uno sebagai kata kunci.

Metode ekstraksi fitur menggunakan Count Vectorizer dan TFIDF. Lalu, dari masing-masing fitur yang yang telah diekstraksi, akan dilakukan seleksi fitur menggunakan Chi Square, ANOVA, dan Mutual Information.

Berikutnya, data akan diuji menggunakan *cross validation* dengan metode pengklasifikasian yang berbeda seperti Random Forest, Support Vector Machine, Decision Tree, Extra Tree, K-Nearest Neighbour, Multinomial Naive Bayes, Gaussian Naive Bayes, Logistic Regression, Neural Network dengan Multi-layer Perceptron, Ada Boost, dan Gradient Boosting. Lalu yang terakhir akan didapatkan hasil akurasi, presisi, *recall* dan *f-measure* pada masing-masing data, metode ekstraksi fitur, metode seleksi fitur, dan metode pengklasifikasian.

4.2 Tahapan Penelitian

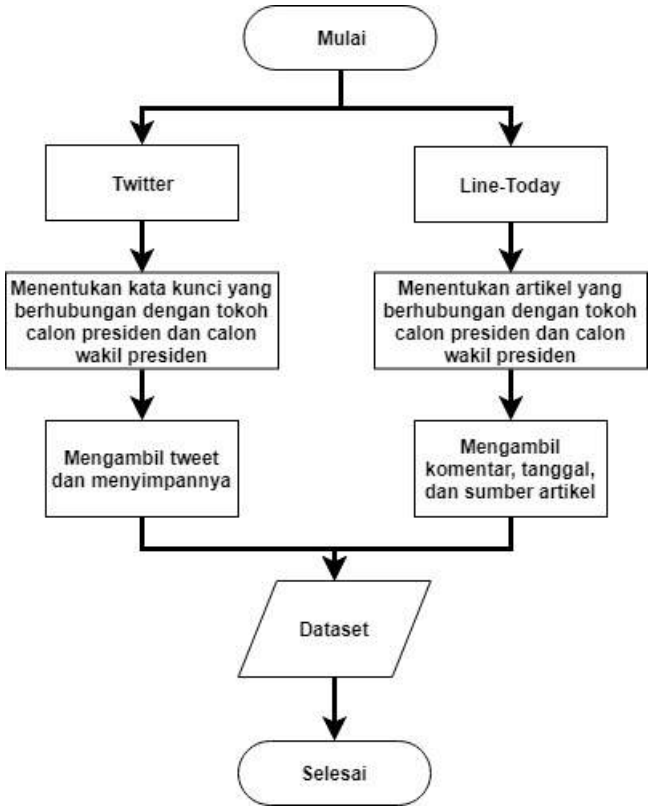
Tahapan yang dilakukan dalam penelitian ini secara umum adalah .

4.2.1 Studi Literatur

Pada tahap ini dilakukan pengumpulan informasi informasi yang mendukung penelitian ini. Informasi yang telah dikumpulkan mencakup penjelasan mengenai analisis sentimen, s*cra*pping *data*, metode ekstraksi fitur, metode seleksi fitur, metode pengklasifikasian, dan metode evaluasi k-fold cross validation untuk menguji akurasi, presisi, *recall*, dan *f-measure* model. Informasi tersebut didapatkan dari berbagai sumber seperti jurnal ilmiah, paper, website, buku, skripsi dan sebagainya.

4.2.2 Pengumpulan Data

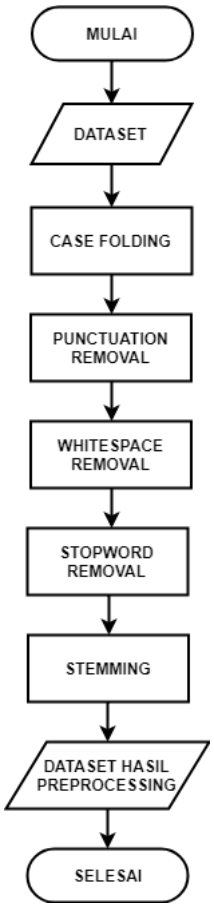
Penelitian ini menggunakan data dari Twitter dan Line-Today yang diambil menggunakan beberapa alat untuk *scrapping* dan alat untuk mendapatkan komentar dengan *request* komentar dari artikel berita. Setiap data kemudian dilabeli secara manual untuk sentimen positif atau negatifnya. Rangkaian dari proses pengumpulan data ditunjukan pada gambar (4.1) .



Gambar 4.1: Diagram Alur Scrapping Data

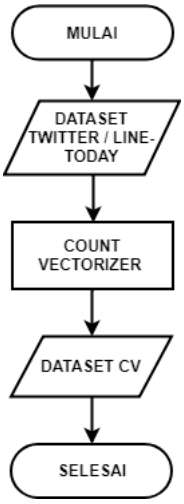
4.2.3 Perancangan Sistem

Proses *preprocessing* mengubah data yang tidak terstruktur menjadi data terstruktur agar lebih mudah diolah. Hal ini dilakukan untuk agar data yang memiliki banyak *noise* dapat diproses dan dianalis oleh sistem. Proses yang dilakukan antara lain *case folding* untuk merubah menjadi huruf kecil, *punctuation removal* untuk menghilangkan simbol yang tidak diperlukan, *whitespace removal* untuk menghapus menghapus spasi awal dan akhir, *stopword removal* untuk menghapus kata-kata tidak memiliki makna, dan *stemming* untuk mengubah semua kata menjadi bentuk kata dasarnya. Rangkaian dari proses preprocessing ditunjukan pada gambar (4.2).

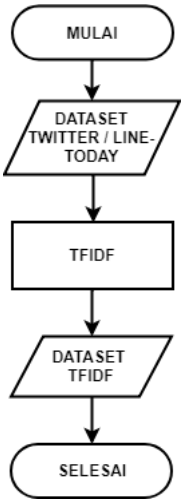


Gambar 4.2: Diagram Alur Preprocessing

Lalu dilakukan proses ekstraksi fitur menggunakan Count Vectorizer dan TFIDF. Rangkaian dari proses ekstraksi fitur ditunjukan pada gambar (4.3) dan (4.4).

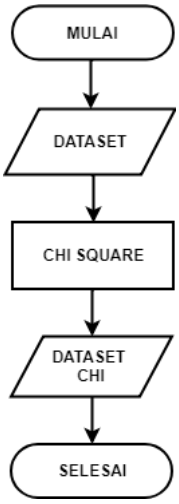


Gambar 4.3: Diagram Alur Ekstraksi Fitur Count Vectorizer

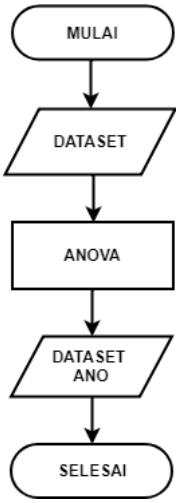


Gambar 4.4: Diagram Alur Ekstraksi Fitur TF-IDF

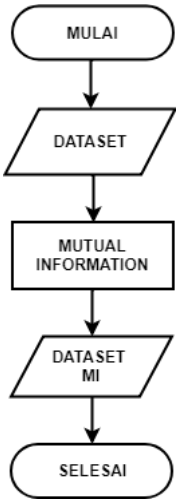
Dilanjutkan dengan proses seleksi fitur menggunakan Chi Square, ANOVA, dan MI. Rangkaian dari proses seleksi fitur ditunjukan pada gambar (4.5), (4.6) dan (4.7) .



Gambar 4.5: Diagram Alur Seleksi Fitur Chi Square

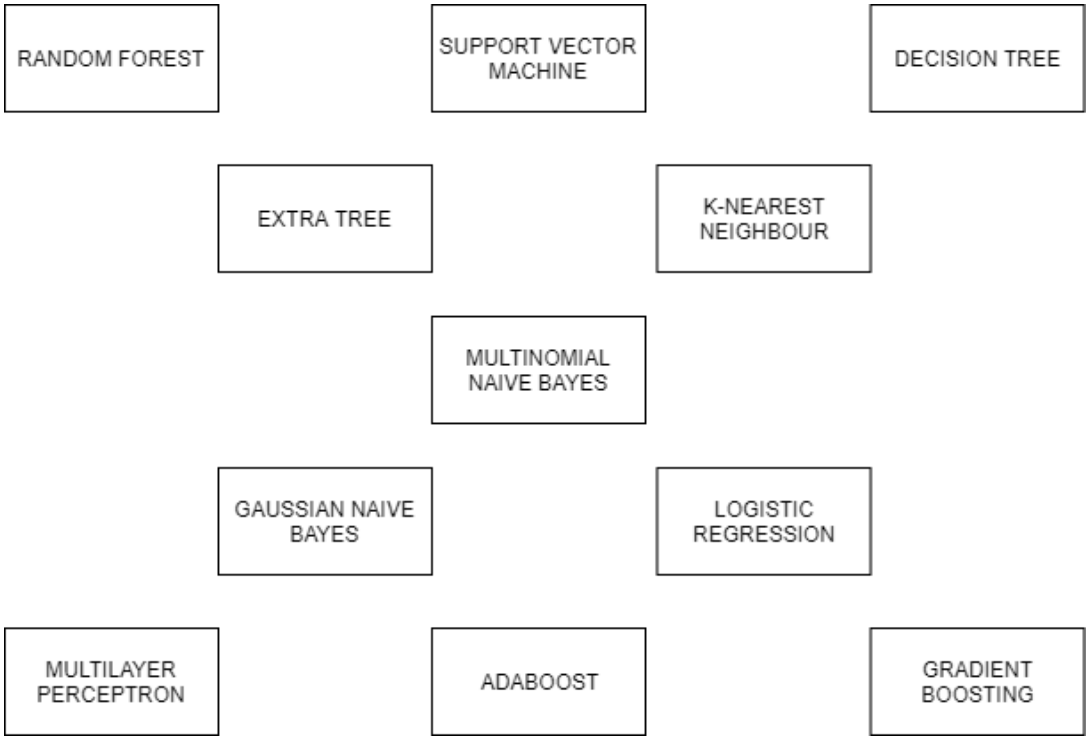


Gambar 4.6: Diagram Alur Seleksi Fitur ANOVA



Gambar 4.7: Diagram Alur Seleksi Fitur Mutual Information

Kemudian melakukan klasifikasi semua jenis data hasil ekstraksi fitur dan seleksi fitur dengan semua masing-masing model pengklasifikasi Random Forest, Support Vector Machine, Decision Tree, Extra tree, K-Nearest Neighbour, Multinomial Naive Bayes, Gaussian Naive Bayes, Logistic Regression, Neural Network dengan Multi-Layer Perceptron, Ada Boost, dan Gradient Boosting. Kumpulan model pengklasifikasi ditunjukan pada gambar (4.8) .



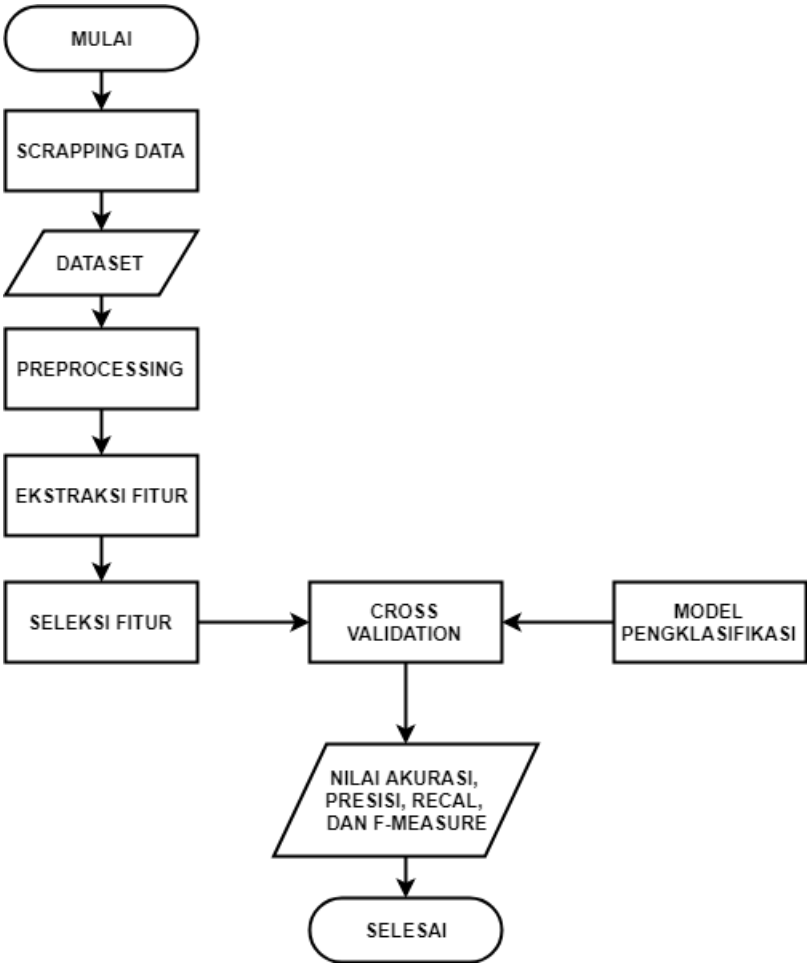
Gambar 4.8: Kumpulan Model Pengklasifikasi

4.2.4 Implementasi Sistem

Pada tahap ini dilakukan implementasi algoritma dan skenario yang sudah dirancang dalam bentuk program komputer. Program tersebut akan dibuat dengan menggunakan bahasa pemrograman Python3 dengan menggunakan Jupyter Notebook atau Google Colab.

4.2.5 Klasifikasi dan Validasi

Pengujian akan dilakukan dengan menggunakan metode *10-fold cross validation*. Secara *stratified* data akan dibagi menjadi 10 bagian. Kemudian satu bagian akan diujikan sebagai data uji, sedangkan bagian lainya akan digunakan sebagai data latih. Pengujian dilakukan sebanyak sepuluh kali dengan digunakan bagian yang berbeda-beda sebagai data ujinya. Hasil tersebut kemudian di rata-rata dari setiap pengujian yang dilakukan. Hasil pengujian tersebut kemudian dimasukan akurasi, presisi, *recall*, dan *f-measure*nya. Diagram alur tahapan keseluruhan dan pengujian ditunjukkan pada gambar (4.9) .



Gambar 4.9: Rangkaian Seluruh Diagram Alur Penelitian Dengan Cross Validation 10 Fold

4.2.6 Penulisan Laporan

Tahapan penulisan laporan dilakukan sejalan dengan berjalannya penelitian.

4.3 Rancangan Pengumpulan Data

4.3.1 Line-Today

Proses pengumpulan data pada Line-Today dilakukan secara manual pada rentang waktu 15-18 April 2019. Hal yang pertama dilakukan adalah melihat 10 artikel terpopuler pada masing-masing hari yang berhubungan dengan topik pemilu 2019. Total jumlah data yang digunakan dalam penelitian ini sebanyak 5477 komentar dan

17 artikel berita. Berikut cuplikan data komentar dan berita dapat dilihat pada tabel (4.1).

Tabel 4.1: Cuplikan data Line-Today yang akan dikumpulkan

Komentar
memang enak kalau di duain????
Bodo amat, ketika masih calon, janjinya muluk2,setelah jadi pemimpin rakyat yang menderita. ak GOLPUT
prabowo pembunuh aktivis mahasiswa, pelanggar penjahat HAM berat, tukang jagal manusia, kroni Orba Biang Korupsi, biangnya Tukang Fitnah hoax ...prabohoax... ...drpd ikutan kena dosa kena karma trus masuk neraka, 01 saja Aaahhhhhh.....
Gaji presiden mah kecil, yang besar mah tunjangan dan lain lain.
kalau quick count yang menang 02,berarti tidak ada kecurangan. Kalau quick count yang menang 01,berarti banyak kecurangan. Mending tidak usa pemilu.. Haha
NGEBET JADI PRESIDEN SAMPAI HILANG AKAL SEHAT yang WARAS NGALAH saja
Di saat ambisi mengalahkan akal sehat

4.3.2 Twitter

Sementara itu, data *tweet* didapatkan dengan teknik *scrapping* dengan menggunakan pustaka Twint dengan memasukan kata kunci berupa nama tokoh, rentang waktu, dan bahasa pengguna dari twitter. Dataset diambil dalam rentang waktu 15-18 April 2019. Setelah data di *scrapping*, masing-masing kata kunci dari tokoh, *tweet* kemudian dimasukan kedalam suatu file berekstensi $.csv$. Kata kunci yang digunakan yaitu "Joko Widodo", "Ma'ruf Amin", "Prabowo Subianto", dan "Sandiaga Uno". Total jumlah data yang digunakan dalam penelitian ini sebanyak 166.834 data *tweet*. Namun dikarenakan keterbatasan waktu dan komputasi maka akan dilakukan *random sampling* dan mengambil sejumlah 5477 *tweet*. Berikut cuplikan data Twitter dapat dilihat pada tabel(4.2).

Tabel 4.2: Cuplikan data Twitter yang akan dikumpulkan

Tweet
Pak prabowo- sandi yg menang Real count dan tak terbnthkn TheVictoryOfPrabowo
VictoryForPrabowo kawal suara kami di cimahi bogor @fadlizon @prabowo @FaldoMaldini @sandiuno pic.twitter.com/wNhUS7Uymc
Pak @Jokowi lahir dalam keluarga muslim, gemar membangun silaturrahmi, serta memiliki kepedulian tinggi terhadap masyarakat muslim. PilihOrangBaik PilihYgJelasIslamnya PilihYgBajuPutih @jokowi Gaspol @saaebunglon @bocahsosmed @sukangetweet
Kl @prabowo dan @sandiuno ingin menang, relawan 02 harus siap menjaga hasil perhitungan TPS minimal sampai tingkat kabupaten. Jangan pernah lepaskan pengawasan walaupun sedetik sebelum sampai tingkat minimal kabupaten kota. Biar pengecekan Situng lebih mudah. Good Luck.
Tunggu hasil real count, jgn trpengaruh quick count pesanan
Tiga tahun di Aceh, karakter keislaman Pak @Jokowi digembleng. PilihOrangBaik PilihYgJelasIslamnya PilihYgBajuPutih https://www.gesuri.id/pemilu/3-tahun-di-aceh-karakter-keislaman-jokowi-digembleng-b1WcIZivOÂ â¦
Mari Bersama Sama Berdoa Team @jokowi Untuk Kemenangan Bangsa Indonesia NO1

4.4 Rancangan Preprocessing

4.4.1 Case Folding

Case folding adalah proses penyamaan ukuran teks pada suatu kalimat. Tidak semua teks konsisten dalam penggunaan huruf kecil dan huruf besar. Oleh karena itu perlunya dilakukan penyamaan *case*, standar dalam penyamaan ini semua teks akan diubah menjadi bentuk standar yang sama yaitu huruf kecil. Contoh proses *case folding* ditampilkan pada tabel (4.3).

Tabel 4.3: Contoh Case Folding

Sebelum Case Folding	Sesudah Case Folding
Nunggu KPU dulu woy, yaelah??	nunggu kpu dulu woy, yaelah??

4.4.2 Punctuation Removal

Tahap *punctuation removal* digunakan memilih teks saja. Hal ini diperlukan untuk membuang fitur non teks yang tidak bermakna. Contoh proses *punctuation removal* ditampilkan pada tabel (4.4).

Tabel 4.4: Contoh Punctuation Removal

Sebelum Case Folding	Sesudah Case Folding
Sujud syukur atas kemenagan pakde.....??????????	Sujud syukur atas kemenagan pakde

4.4.3 Whitespace Removal

Tahap *whitespace removal* digunakan untuk menghapus ruang kosong. Hal ini dilakukan untuk menghilangkan spasi, enter, atau tab yang akan mengganggu proses tokenisasi. Contoh proses *whitespace removal* ditampilkan pada tabel (4.5).

Tabel 4.5: Contoh Whitespace Removal

Sebelum Case Folding	Sesudah Case Folding
Gaji sih tidak diambil Tunjangan" diembat	Gaji sih tidak diambil Tunjangan" diembat

4.4.4 Stopwords Removal

Tahap *stopwords removal* digunakan untuk menghapus kata yang tidak memiliki makna. *Stopwords removal* dilakukan dengan mencari kata yang tidak bermakna dalam kalimat. Hal ini dilakukan dengan mencari isi dari kalimat apakah terdapat kata tidak bermakna berdasarkan perbandingan kamus kata *stopwords* berbahasa Indonesia di pustaka Natural Language Tool Kit NLTK. Contoh proses *stopwords removal* ditampilkan pada tabel (4.6).

Tabel 4.6: Contoh Stopwords Removal

Sebelum Case Folding	Sesudah Case Folding
Sebenernya yang pencintraan siapa sih iya.	Sebenernya pencintraan sih iya.

4.4.5 Stemming

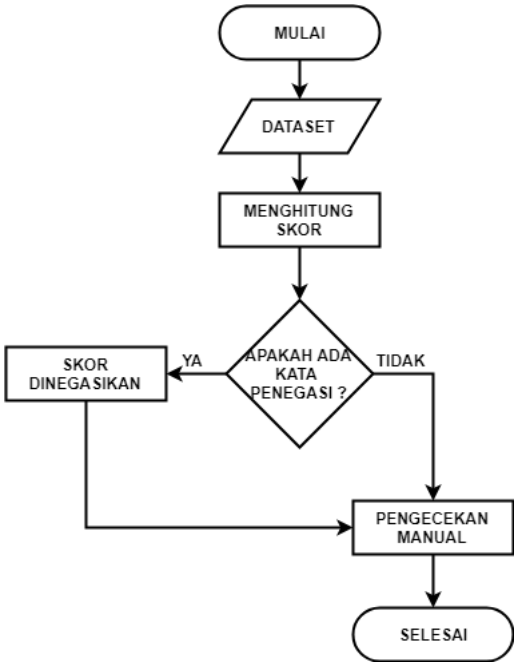
Tahap *stemming* digunakan untuk membuat kata menjadi bentuk dasar. Hal ini dilakukan untuk menyamakan kata yang sebenarnya bermakna sama namun dengan penulisan berbeda. Proses *stemming* dilakukan dengan bantuan pustaka Sastrawi. Contoh proses *stopwords removal* ditampilkan pada tabel (4.7).

Tabel 4.7: Contoh Stemming

Sebelum Case Folding	Sesudah Case Folding
Semoga ini bukan langkah untuk mengundang masa protes	moga ini bukan langkah untuk undang masa protes

4.5 Pelabelan Data

Pelabelan data adalah pemberian sentimen pada masing-masing *tweet* dan komentar data menggunakan polaritas sentimen positif dan negatif. Langkah awal pelabelan dilakukan dengan melihat kekuatan kata berdasarkan penelitian Hidayatullah dan Azhari, 2015 yang memberikan skor dari kata dan daftar kata penegasi yang membalikkan hasil akhir skor. Kemudian secara manual dengan label yang dihasilkan dicek secara sekilas ke dalam kelas ’1’ dan ’0’. Berikut diagram alur pelabelan data pada gambar (4.10).



Gambar 4.10: Diagram alur pelabelan

4.6 Ekstraksi Fitur

4.6.1 Count Vectorizer

Count vectorizer adalah mengkonversi kumpulan dokumen teks ke matriks jumlah dalam bentuk token. Contoh terdapat 3 kalimat "saya suka dukung jokowi", "dia tidak suka prabowo", dan "saya tidak suka jokowi". Maka bentuk contoh hasil ekstrasi *count vectorizer* akan seperti pada tabel (4.8)

Tabel 4.8: Contoh ekstraksi Count Vectorizer

	dia	dukung	jokowi	prabowo	saya	suka	tidak
0	0	1	1	0	1	1	0
1	1	0	0	1	0	1	1
2	0	0	1	0	1	1	1

4.6.2 Term Frequency Inverse-Document Frequency

Dalam Term Frequency dan Invers Document Frequency (TFIDF) untuk menghitung pembobotan diperlukan perhitungan Term Frequency dan Invers Document Frequency. Tahapan yang diperlukan untuk perhitungan bobot TFIDF adalah

1. Menghitung nilai Term Frequency,
2. Menghitung nilai Invers Document Frequency yaitu perhitungan jumlah seluruh dokumen dibagi dengan banyaknya dokumen di mana kata i muncul. Rumus perhitungan Invers Document Frequency ditampilkan pada persamaan (3.2),
3. Menghitung nilai TFIDF dengan membagi Term Frequency dengan Invers Document Frequency. Rumus perhitungan TFIDF ditampilkan pada persamaan (3.3).

Contoh hasil perhitungan TFIDF dengan contoh 3 kalimat yang sama pada tabel (4.8) ditampilkan pada Tabel (4.9).

Tabel 4.9: Contoh ekstraksi TFIDF

	dia	dukung	jokowi	prabowo	saya	suka	tidak
0	0	0.631745	0.480458	0	0.480458	0.373119	0
1	0.584483	0	0	0.584483	0	0.345205	0.444514
2	0	0	0.526820	0	0.526820	0.409123	0.526820

4.7 Seleksi Fitur

Kemudian seleksi fitur akan dilakukan pada hasil ekstraksi sebelumnya. Seleksi fitur dilakukan untuk mengungurangi biaya komputasi dan menghindari overfitting. Metode seleksi fitur yang akan digunakan dan dibandingkan adalah Chi Square, F-test menggunakan Analysis of Variance (ANOVA), dan Mutual Information (MI). Seleksi fitur Chi Square menggunakan teori statistika untuk menguji independensi sebuah *term* dengan kategorinya. Seleksi fitur Chi Square dilakukan dengan cara mengurutkan setiap berdasarkan fitur berdasarkan hasil seleksi fitur Chi Square dari nilai yang terbesar hingga terkecil. Perhitungan nilai Chi Square dijelaskan pada

persamaan (3.4) dan (3.5). Nilai seleksi fitur Chi Square yang lebih besar dari nilai signifikan menunjukan penolakan hipotesis independensi.

F-test adalah uji statistik yang dimana statistik uji memiliki distribusi F di bawah hipotesis nol. Ini paling sering digunakan ketika membandingkan model statistik, untuk mengidentifikasi model yang paling cocok dengan populasi dari mana data sampel. F-test telah diterapkan ke data menggunakan kuadrat terkecil. F-test dalam Analysis of Variance (ANOVA) menilai menilai apakah nilai-nilai yang diharapkan dari variabel kuantitatif dalam beberapa kelompok yang telah ditentukan berbeda satu sama lain. Sebagai contoh, anggaplah data berupa teks. Perhitungan nilai scoring pada F-test ANOVA ditunjukkan pada persamaan (3.6). F-test ANOVA dapat digunakan untuk menilai apakah salah satu kata lebih unggul, atau lebih rendah, dibandingkan kata yang lain dengan hipotesis nol bahwa semua kata menghasilkan respons rata-rata yang sama. Ini adalah contoh dari tes "omnibus", yang berarti bahwa satu tes dilakukan untuk mendeteksi salah satu dari beberapa kemungkinan perbedaan.

Metode seleksi fitur yang umum adalah untuk menghitung $A(t, c)$ sebagai *expected* Mutual Information (MI) dari term t dan kelas c. MI mengukur seberapa banyak informasi yang ada atau tidaknya suatu term berkontribusi untuk membuat keputusan klasifikasi yang benar pada c. Perhitungan nilai scoring pada MI dijelaskan pada persamaan (3.7). Seleksi yang akan dilakukan dalam penelitian ini yaitu mengambil 50 persen dari total fitur terbaik berdasarkan *scoring* metode Chi Square, ANOVA, dan MI.

4.8 Klasifikasi

Pada penelitian ini dilakukan klasifikasi sentimen untuk mengetahui sentimen dari setiap tweet dari dataset Twitter dan komentar dataset Line-Today. Pengklasifikasi yang akan digunakan yaitu :

1. Random Forest : Random Forest (RF) menciptakan beberapa *tree*, dalam penelitian ini 100, dan menghitung model terbaik untuk dataset yang diberikan. Alih-alih mempertimbangkan semua fitur saat memisahkan *node*, algoritme RF memilih fitur terbaik dari *subset* semua fitur. Ini memberikan bias yang lebih tinggi untuk varian yang lebih rendah, yang menghasilkan model yang jauh lebih baik.

2. Support Vector Machine : Support Vector Machine (SVM) adalah algoritme

pembelajaran mesin yang *diawasi* yang dapat digunakan untuk tujuan klasifikasi. SVM didasarkan pada gagasan untuk menemukan hyperplane yang terbaik membagi dataset menjadi dua kelas. Anda dapat menganggap hyperplane sebagai garis yang secara linear memisahkan dan mengklasifikasikan satu set data. Secara intuitif, semakin jauh dari titik data dari *hyperplane*, semakin yakin SVM telah mengklasifikasi dengan benar. Karena itu agar titik data berada sejauh mungkin dari hyperplane, sambil tetap berada di sisi yang benar.

3. Decision Tree : Decision Tree (DT) adalah alat pendukung keputusan yang menggunakan grafik atau model keputusan seperti pohon dan kemungkinan konsekuensinya.

4. Extra Tree : Metode Extra Tree (*Extremely Randomized Tree*) bertujuan untuk mengacak lebih lanjut *tree* dalam konteks fitur masukan secara numerik, di mana pilihan titik potong optimal bertanggung jawab atas sebagian besar dari varian *tree* yang diinduksi. Gagasan ini cukup produktif dalam konteks banyaknya masalah yang ditandai oleh sejumlah besar fitur numerik yang bervariasi lebih atau kurang secara kontinyu. Hal ini sering mengarah pada peningkatan akurasi berkat pemerataannya dan pada saat yang sama secara signifikan mengurangi beban komputasi terkait dengan penentuan optimal titik potong di DT dan RF.

5. K-Nearest Neighbour : K-Nearest Neighbour (KNN) bekerja dengan banyak k dengan posisi acak. Untuk menemukan tetangga terdekat, dihitung jarak antara titik k dengan titik lainnya. Kemudian dipilh sebanyak k tetangga teratas yang jaraknya paling dekat dengan titik hitam. Karena mayoritas titik hijau di sekitar titik hitam ini, k hitam ini menetapkan label hijau untuknya.

6. Multinomial Naive Bayes : Multinomial Naive Bayes (MNB) memperkirakan probabilitas bersyarat dari kata tertentu yang diberikan kelas sebagai frekuensi relatif dari term t dalam dokumen milik kelas c. Variasi memperhitungkan jumlah kemunculan *term* t dalam dokumen pelatihan dari kelas c.

7. Gaussian Naive Bayes : Gaussian Naive Bayes (GNB) bekerja sama seperti MNB hanya berubah saat mengukur semua kemunculan dalam *term* t dalam dokumen. GNB mengukur kejadian hanya sekali.

8. Logistic Regression : Logistic Regression (LR) adalah pengklasifikasi yang menggunakan logit model. Logit model adalah model matematika yang digu-

nakan dalam statistik untuk memperkirakan probabilitas suatu peristiwa yang terjadi telah diberikan beberapa data sebelumnya. LR umumnya digunakan di mana variabel dependen adalah Biner atau Dikotomis. Itu berarti variabel dependen hanya dapat mengambil dua nilai yang mungkin seperti "1 atau 0". Faktor atau variabel independen dapat berupa variabel kategori atau numerik.

9. Multilayer Perceptron : Multilayer Perceptron (MLP) adalah bagian dari Neural Network yang terdiri dari setidaknya 3 node yang merupakan input layer, output layer dan hidden layer. Masing-masing layer MLP, kecuali *input layer* adalah neuron yang menggunakan fungsi aktivasi non-linear. Node MLP disusun berlapis-lapis dalam hidden layer. Dalam MLP, neuron menggunakan fungsi aktivasi non-linear yang dirancang untuk memodelkan perilaku neuron di otak manusia. Sebuah multi-layer perceptron memiliki fungsi aktivasi linier di semua neuronnya dan menggunakan *backpropagation* untuk pelatihannya.

10. Ada Boost : Pertama dipilih pengklasifikasi dasar yang membuat prediksi pada data yang diberikan. Hitung kesalahan klasifikasi. Bobot dari contoh kesalahan klasifikasi meningkat. Klasifikasi kedua dilatih pada set pelatihan dengan bobot yang diperbarui. Secara sederhana, Jalankan pengklasifikasi dan buat prediksi. Jalankan pengklasifikasi lain agar sesuai dengan contoh yang sebelumnya salah klasifikasi dan membuat prediksi. Ulangi sampai semua atau sebagian besar contoh pelatihan diterapkan.

11. Gradient Boosting : Mirip dengan AdaBoost, Gradient Boosting juga bekerja dengan model prediksi berulang-ulang yang ditambahkan ke ensemble. Alih-alih memperbarui bobot contoh pelatihan seperti AdaBoost, Gradient Boosting cocok dengan model baru dengan residual error. Sederhananya, Sesuaikan model dengan set pengklasifikasi yang diberikan. Hitung residual error yang menjadi contoh pelatihan baru. Model baru dilatih dengan residual error dan seterusnya. Tambahkan semua model yang dipilih untuk membuat prediksi.

4.9 Validasi

Pada klasifikasi sentimen dibutuhkan validasi model untuk melihat seberapa bagus model tersebut untuk digunakan. Validasi yang digunakan dalam penelitian ini adalah K-Fold cross-validation dengan k = 10. Evaluasi performa ini didapatkan dengan membandingkan hasil klasifikasi dan prediksi 10 *fold* sentimen dengan data

yang diberi label sebelumnya. Model yang sudah diklasifikasi tadi dihitung dalam perhitungan nilai akurasi, presisi, *recall*, dan *f1 score* atau *f-measure*. Persamaan perhitungan nilai nilai akurasi, presisi, *recall*, dan *f1 score* ditampilkan pada persamaan (3.20), (3.21), (3.22), (3.23).

BAB V

IMPLEMENTASI

5.1 Spesifikasi Sistem

Pada tahapan ini dibangun penelitian sesuai dengan apa yang telah dirancang pada bab analisis dan perancangan sistem. Spesifikasi pada sistem pakar ini terdiri dari perangkat keras dan perangkat lunak.

1. Perangkat keras yang digunakan dalam penelitian sebagai berikut.
   (a) Laptop ASUS A456UR
   (b) Processor Intel(R) Core(TM) i5-7200u 2.5 GHz
   (c) Memory RAM 4,00 GB
   (d) Harddisk 500 GB
   (e) VGA Nvidia GeForce 930mx 4 GB
2. Dan perangkat komputer dari Laboratorium Sistem Cerdas sebagai berikut.
   (a) IMac 21.5-inch, Mid 2011
   (b) Processor Intel(R) Core(TM) i5 2.5 GHz
   (c) Memory RAM 4,00 GB
   (d) Harddisk 500 GB
   (e) AMD Radeon HD 6750 512 MB
3. Perangkat lunak yang digunakan dalam membangun sistem pakar ini adalah sebagai berikut.
   (a) Python 3.7.2
   (b) Jupyter Notebook 1.0.0
   (c) Google Chrome 75.0.3770
   (d) Google Colab
   (e) Library Scikit-Learn 0.21.2
   (f) Library Pandas 0.24.2

(g) Library Numpy 1.16.4

(h) Library NLTK 3.4.3

(i) Library Sastrawi 1.0.1

(j) Library Moses Tokenizer 0.6.2

(k) Library Twint 1.2.0

5.2 Implementasi Pengumpulan Data

5.2.1 Line-Today

Proses pengumpulan data pada Line-Today dilakukan secara manual pada rentang waktu 15-18 April 2019. Hal yang pertama dilakukan adalah melihat 10 artikel terpopuler pada masing-masing hari yang berhubungan dengan topik pemilu 2019. Kemudian *Inspect Element* pada masing-masing artikel dan masuk ke bagian *network*. Kemudian pilih xhr. Lakukan Refresh pada bagian komentar. Pilih *request* GET menuju API Line-Today. Selanjutnya unduh hasil semua komentar yang berupa JSON. Kemudian ubah tipe data JSON ke dalam bentuk CSV. Kombinasikan semua data dari rentang waktu 15-18 April 2019 menjadi satu dataset.

5.2.2 Twitter

Proses pengumpulan tweet pada tahap implementasi menggunakan pustaka Twint. Sebelumnya dikenal sebagai Tweep, Twint adalah alat *scrapping* Twitter yang dibuat dalam Python yang memungkinkan untuk *scrapping* Tweet dari profil Twitter tanpa menggunakan API Twitter. Twint menggunakan operator pencarian Twitter untuk memungkinkan Anda *scrapping* Tweet dari pengguna tertentu, *scrapping* Tweet yang berkaitan dengan topik tertentu, tagar dan *trending*. Implementasi dengan Twint ditunjukan pada gambar (5.1).

```
1   import twint
2
3   # Konfigurasi pencarian
4   c = twint.Config()
5
6   # Kata kunci pencarian (['Joko Widodo', 'Ma'ruf Amin', 'Prabowo Subianto'\
7   # , 'Sandiaga Uno')
8   c.Search = "Joko Widodo"
9
10  #Tanggal jangkauan
11  c.Since = "2019-5-15"
12  c.Until = "2019-5-18"
13
14  # Bahasa Twitter
15  c.Lang = "id"
16
17  #Format penyimpanan output
18  c.Store_csv = True
19
20  #Lokasi penyimpanan output
21  c.Output = "Twitter"
22
23  twint.run.Search(c)
```

Gambar 5.1: Implementasi Program *Scrapping* Twint

Pada baris ke-1 dilakukan pemanggilan pustaka Twint. Selanjutnya pada baris ke-4, dibuat variabel untuk menentukan konfigurasi persyaratan saat melakukan *scrapping*. Kemudian di baris ke-8 disebutkan kata kunci dalam pencarian *scrapping*. Nantinya, di tiap *scrapping* akan diubah dalam masing-masing kata kuncinya. Kemudian di baris ke-11 sampai ke-12 diberikan persyaratan batas waktu dalam *scrapping*. Selanjutnya di baris ke-15 menambahkan persyaratan bahwa pengguna menggunakan twitter berdomain bahasa Indonesia. Selanjutnya di baris ke-18 menunjukan konfigurasi keluaran jenis data yaitu Comma Separator Value (CSV). Di baris ke-21 memberikan konfigurasi penamaan pada keluaran hasil *scrapping*. Kemudian di baris ke-23 akan menjalankan program untuk melakukan*scrapping*.

5.3 Preprocessing

5.3.1 Case Folding

Proses yang dilakukan pertama pada *preprocessing* yaitu *case folding*. *case folding* bertujuan untuk merubah menjadi huruf kecil. Pada penelitian ini, proses *case folding* menggunakan pustaka Pandas. Implementasi Case Folding ditunjukan pada gambar (5.2).

```
1   import pandas as pd
2
3   line = pd.read_csv('line.csv', encoding = "ISO-8859-1")
4   twitter = pd.read_csv('twitter_sampling', encoding = "ISO-8859-1")
5
6   line.Berita = line.Berita.str.lower()
7   twitter.tweet = twitter.tweet.str.lower()
```

Gambar 5.2: Implementasi program untuk *case folding*

Pada baris ke-1 dilakukan pemanggilan pustaka Pandas. Kemudian pada baris ke-3 dan ke-4 dilakukan proses memasukkan data Line-Today dan Twitter ke masing-masing variabel menggunakan *encoding* ISO-8859-1. ISO-8859-1 digunakan karena *encoding* paling standar dari berbagai sistem operasi pada dokumen. Lalu di baris ke-6 dan ke-7 dilakukan proses *Case Folding* pada masing-masing variabel yang menggunakan perintah **lower()**. Alasan menggunakan **str** sebelum *case folding* karena beragamnya tipe data pada komentar Line-Today dan *tweet* pada Twitter. Maka dari itu semua tipe data akan diseragamkan menjadi bentuk *string* yang kemudian diubah menjadi *lowercase* semua.

5.3.2 Punctuation Removal

Proses *preprocessing* selanjutnya dilakukan *puntuation removal*. *punctuation removal* untuk menghilangkan simbol yang tidak diperlukan. Implementasi *punctuation removal* ditunjukan pada gambar (5.3).

```
1   import pandas as pd
2
3   line = pd.read_csv('line.csv', encoding = "ISO-8859-1")
4   twitter = pd.read_csv('twitter_sampling', encoding = "ISO-8859-1")
5
6   line.Berita = line.Berita.str.replace('[^\w\s]','')
7   twitter.tweet = twitter.tweet.str.replace('[^\w\s]','')
```

Gambar 5.3: Implementasi program untuk *punctuation removal*

Untuk baris ke-1 sampai baris ke-4 sama seperti proses sebelumnya, yaitu pemanggilan pustaka dan memasukkan data ke variabel. Lalu di baris ke-6 dan baris ke-7 dilakukan penghapus semua non-karakter. adalah singkatan dari "*word character*". Itu cocok dengan karakter $ASCII[A-Za-z0-9]$. dari "karakter spasi" terdiri dari spasi, tab, penghentian baris, atau *form feed*. Lalu perintah **replace()** akan mengganti semua non-karakter menjadi kosong.

5.3.3 Whitespace Removal

Preprocessing selanjutnya yaitu *whitespace removal*. *whitespace removal* digunakan untuk menghapus menghapus spasi awal dan akhir. Implementasi *whitespace removal* ditunjukan pada gambar (5.4).

```
1   import pandas as pd
2
3   line = pd.read_csv('line.csv', encoding = "ISO-8859-1")
4   twitter = pd.read_csv('twitter_sampling', encoding = "ISO-8859-1")
5
6   line.Berita = line.Berita.str.strip()
7   twitter.tweet = twitter.tweet.str.strip()
```

Gambar 5.4: Implementasi program untuk *whitespace removal*

Untuk baris ke-1 sampai baris ke-4 sama seperti proses sebelumnya, kemudian dilakukan *whitespace removal* di baris ke-6 dan ke-7 menggunakan perintah **strip()**.

5.3.4 Stopwords Removal

Proses selanjutnya akan dilakukan *stopwords removal*. *Stopwords removal* untuk menghapus kata-kata tidak memiliki makna. Untuk mendapatkan kumpulan kata tidak bermakna berbahasa Indonesia dari pustaka NLTK. Implementasi *stopwords removal* ditunjukan pada gambar (5.5).

```
1   import nltk
2   from nltk.corpus import stopwords
3   from nltk.tokenize import word_tokenize
4   from mosestokenizer import MosesDetokenizer
5
6   detokenizer = MosesDetokenizer()
7   stopwords = set(stopwords.words('indonesian'))
8
9   for i in range(len(line.Berita)):
10      tokens = word_tokenize(line.Berita[i])
11      filtered_sentences = []
12
13      for w in tokens:
14        if w not in stopwords:
15          filtered_sentences.append(w)
16
17      sentence = detokenizer(filtered_sentences)
18      line.Berita[i] = sentence
19
20  for i in range(len(twitter.tweet)):
21      tokens = word_tokenize(twitter.tweet[i])
22      filtered_sentences = []
23
24      for w in tokens:
25        if w not in stopwords:
26          filtered_sentences.append(w)
27
28      sentence = detokenizer(filtered_sentences)
29      twitter.tweet[i] = sentence
```

Gambar 5.5: Implementasi program untuk *stopwords removal*

Baris ke-1 sampai ke-4 melakukan pemanggilan pustaka yang diperlukan. Baris ke-6 menjadikan perintah **MosesDetokenizer** ke dalam variabel. Penggunaan **MosesDetokenizer** diperlukan untuk menggabungkan kembali proses *stopwords removal* yang harus di *tokenize*. Di baris ke-7, kata *stopwords* berbahasa Indonesia yang didapatkan dari NLTK dimasukkan ke dalam bentuk *dictionary*. Kemudian di baris ke-9 dilakukan pengulangan dengan batasan panjang fitur. Kemudian di baris ke-10, masing-masing baris i pada fitur diubah dalam bentuk *token*. Di baris ke-11 dibuat variabel baru yang bersifat sementara untuk menyimpan hasil filter pada masing-masing baris fitur. Lalu di aris ke-13 sampai ke baris ke-15 dilakukan pengecekan pada fitur di baris i untuk melihat adakah bagian *stopwords* pada fitur baris ke i. Jika tidak ada, maka hasil akan dimasukkan ke dalam variabel $filtered_sentences$. Di baris ke-17 dilakukan *detokenize* pada fitur yang sudah di *filter*. Kemudian fitur yang sudah di *filter* akan menggantikan fitur pada baris ke i. Untuk baris ke-20 sampai baris ke-29 sama saja prosesnya dengan baris ke-9 sampai baris ke-18, hanya berbeda datasetnya.

5.3.5 Stemming

Proses terakhir pada *preprocessing* yaitu *stemming*. *stemming* digunakan untuk mengubah semua kata menjadi bentuk kata dasarnya. Proses *stemming* dengan data berbahasa Indonesia menggunakan pustaka Sastrawi. Implementasi *stopwords removal* ditunjukan pada gambar (5.6).

```
1  from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
2
3  factory = StemmerFactory()
4  stemmer = factory.create_stemmer()
5
6  for i in range(len(line.Berita)):
7    line.Berita[i] = stemmer.stem(line.Berita[i])
8
9  for i in range(len(twitter.tweet)):
10   twitter.tweet[i] = stemmer.stem(twitter.tweet[i])
```

Gambar 5.6: Implementasi program untuk *stemming*

Baris ke-1 digunakan untuk meamnggil pustaka Sastrawi yang nantinya menggunakan perintah **StemmerFactory()**. Baris ke-3 dan baris ke-4 digunakan untuk membuat *stemmer*. Lalu baris ke-6 dan baris ke-7 dilakukan proses stemming pada masing-masing baris i pada fitur. Kemudian baris ke-9 dan baris ke-10 memiliki tugas yang sama dengan baris ke-6 dan baris ke-7 hanya berbeda datasetnya saja.

5.4 Pelabelan Data

Pelabelan data adalah pemberian sentimen pada masing-masing *tweet* dan komentar data menggunakan polaritas sentimen positif dan negatif. Pelabelan data digunakan untuk pembentukan model klasifikasi dalam melakukan pemodelan data latih. Pelabelan juga digunakan untuk mengukur performa dari klasifikasi dengan membandingkan label yang telah dimasukkan. Langkah awal pelabelan dilakukan dengan melihat kekuatan kata berdasarkan penelitian Hidayatullah dan Azhari, 2015 yang memberikan skor dari kata dan daftar kata penegasi yang membalikkan hasil akhir skor. Kemudian secara manual dengan label yang dihasilkan dicek secara sekilas ke dalam kelas ’1’ dan ’0’. Implementasi pelabelan ditunjukan pada gambar (5.7).

```
1   skor2 = pd.read_csv("skor2.csv")
2   negate = pd.read_csv("negatingword.csv")
3
4   for i in range(len(line.Berita)):
5     score = 0
6     for j in range(len(skor2.kata)):
7       if str(skor2.kata[j]) in str(line.Berita[i]):
8         score += skor2.skor[j]
9     for j in range(len(negate.kata)):
10      if str(negate.kata[j]) in str(line.Berita[i]):
11        score *= -1
12
13    if score >= 1 :
14      line.sentimen[i] = 1
15    else :
16      line.sentimen[i] = 0
17
18  for i in range(len(twitter.tweet)):
19    score = 0
20    for j in range(len(skor2.kata)):
21      if str(skor2.kata[j]) in str(twitter.tweet[i]):
22        score += skor2.skor[j]
23    for j in range(len(negate.kata)):
24      if str(negate.kata[j]) in str(twitter.tweet[i]):
25        score *= -1
26
27    if score >= 1 :
28      twitter.sentimen[i] = 1
29    else :
30      twitter.sentimen[i] = 0
```

Gambar 5.7: Implementasi program untuk pelabelan

Pada baris ke-1 dan ke-2 dimasukkan data kumpulan kata yang memiliki skor masing-masing dan kumpulan kata penegasi yang didapatkan dari penelitian Hidayatullah dan Azhari, 2015. Lalu di baris ke-4 dimulai proses perulangan sepanjang banyaknya fitur teks dari data Line. Di baris ke-5 membuat variabel skor kosong sementara yang nanti akan kembali ke nilai kosong di tiap perulangan. Kemudian di baris ke-6 sampai baris ke-8 dilakukan pengecekan dan skoring berdasarkan jumlah skor kata dataset skor di dalam fitur per baris i. Kemudian di baris ke-9 sampai baris ke-11 dilakukan pengecekan ada atau tidaknya kata penegasi dalam fitur, jika ada maka hasil skor sebelumnya akan dinegasikan. Yang terakhir di baris ke-13 sampai baris ke-16 akan dilakukan pengelompokan berdasark skor akhir fitur per baris i. Jika skor akhir positif maka akan dimasukan kelas '1', jika tidak maka masuk ke kelas '0'.

5.5 Ekstraksi Fitur

5.5.1 Count Vectorizer

Proses selanjutnya adalah ekstraksi fitur dari dataset Line-Today dan Twitter. Terdapat 2 Ekstraksi fitur yang digunakan yaitu Count Vectorizer dan Term Frequency Inverse Document Frequency (TFIDF). Count Vectorizer mengkonversi kumpulan dokumen teks ke matriks jumlah *token*. Implementasi Count Vectorizer ditunjukan pada gambar (5.8).

```
1   from sklearn.feature_extraction.text import CountVectorizer
2
3   xcv = line.Berita
4   ycv = line.Sentimen
5   x2cv = twitter.tweet
6   y2cv = twitter.sentimen
7
8   countvect = CountVectorizer(analyzer = "word", tokenizer = None, lowercase = None)
9
10  xcv = countvect.fit_transform(xcv).toarray()
11  x2cv =countvect.fit_transform(x2cv).toarray()
```

Gambar 5.8: Implementasi program untuk ekstraksi fitur dengan Count Vectorizer

Di baris ke-1 melakukan pemanggilan pustaka Scikit-Learn dan mengambil perintah **CountVectorizer()**. Kemudian di baris ke-3 sampai baris ke-6 melakukan pemisahan data fitur dan data kelas pada dataset Line-Today dan Twitter. Kemudian di baris ke-8 memasukan perintah **CountVectorizer()** ke dalam variabel dengan beberapa parameter. Parameter *analyzer* ke bagian "word" dengan pemotongan bersifat per kata, kemudian parameter *tokenizer* tidak dilakukan karena sudah dilakukan sebelumnya, dan parameter *lowercase* tidak dilakukan karena juga sudah dilakukan pada proses sebelumnya. Kemudian di baris ke-10 dan baris ke-11 melakukan transformasi fitur teks ke bentuk vektor matriks.

5.5.2 Term Frequency Inverse Document Frequency

TFIDF mengkonversi kumpulan fitur teks ke matriks fitur TFIDF. Implementasi TFIDF ditunjukan pada gambar (5.9).

```
1   from sklearn.feature_extraction.text import TfidfVectorizer
2
3   xtf = line.Berita
4   ytf = line.Sentimen
5   x2tf = twitter.tweet
6   y2tf = twitter.sentimen
7
8   tfidf = TfidfVectorizer(analyzer = "word", tokenizer = None, lowercase = None)
9
10  xtf = tfidf.fit_transform(xtf).toarray()
11  x2tf = tfidf.fit_transform(x2tf).toarray()
```

Gambar 5.9: Implementasi program untuk ekstraksi fitur dengan TFIDF

Di baris ke-1 melakukan pemanggilan pustaka Scikit-Learn dan mengambil perintah **TfidfVectorizer()**. Kemudian di baris ke-3 sampai baris ke-6 melakukan pemisahan data fitur dan data kelas pada dataset Line-Today dan Twitter. Kemudian di baris ke-8 memasukan perintah **TfidfVectorizer()** ke dalam variabel dengan beberapa parameter. Parameter *analyzer* ke bagian "word" dengan pemotongan bersifat per kata, kemudian parameter *tokenizer* tidak dilakukan karena sudah dilakukan sebelumnya, dan parameter *lowercase* tidak dilakukan karena juga sudah dilakukan pada proses sebelumnya. Kemudian di baris ke-10 dan baris ke-11 melakukan transformasi fitur teks ke bentuk vektor matriks.

5.6 Seleksi Fitur

5.6.1 Chi Square

Proses berikutnya adalah melakukan seleksi fitur. Salah satu metode untuk seleksi fitur adalah menggunakan *chi square*. Implementasi *chi square* ditunjukan pada gambar (5.10).

```
1   from sklearn.feature_selection import chi2
2   from sklearn.feature_selection import SelectPercentile
3
4   chi = chi2
5
6   xcvchi = SelectPercentile(chi, percentile = 50).fit_transform(xcv, y)
7   xtfchi = SelectPercentile(chi, percentile = 50).fit_transform(xtf, y)
8
9   x2cvchi = SelectPercentile(chi, percentile = 50).fit_transform(x2cv, y2)
10  x2tfchi = SelectPercentile(chi, percentile = 50).fit_transform(x2tf, y2)
```

Gambar 5.10: Implementasi program untuk seleksi fitur dengan Chi Square

Pada baris ke-1 melakukan pemanggilan pada pustaka **Scikit-Learn** dan mendapatkan fungsi *chi square*. Kemudian di baris ke-2 adalah metode pengambilan seberapa banyak fitur yang akan diseleksi. Pada penelitian dilakukan seleksi secara relatif 50 % dari jumlah fitur terbaik. Kemudian di baris ke-4 memasukkan fungsi *chi square* yang didapat dari **Scikit-Learn** sebelumnya. Kemudian di baris ke-6 dan baris ke-7 dilakukan seleksi fitur dari data Line-Today yang di ekstraksi dengan *count vectorizer* dan TFIDF. Begitu juga dengan baris ke-9 dan baris ke-10 dilakukan seleksi fitur dari data Twitter yang di ekstraksi dengan *count vectorizer* dan TFIDF.

5.6.2 Analysis of Variance

Metode seleksi fitur yang lainnya adalah *Analysis of Variance* (ANOVA). Implementasi ANOVA ditunjukan pada gambar (5.11).

```
1  from sklearn.feature_selection import f_classif
2  from sklearn.feature_selection import SelectPercentile
3
4  ano = f_classif
5
6  xcvano = SelectPercentile(anova, percentile = 50).fit_transform(xcv, y)
7  xtfano = SelectPercentile(ano, percentile = 50).fit_transform(xtf, y)
8
9  x2cvano = SelectPercentile(ano, percentile = 50).fit_transform(x2cv, y2)
10 x2tfano = SelectPercentile(ano, percentile = 50).fit_transform(x2tf, y2)
```

Gambar 5.11: Implementasi program untuk seleksi fitur dengan ANOVA

Pada baris ke-1 melakukan pemanggilan pada pustaka **Scikit-Learn** dan mendapatkan fungsi ANOVA dengan penilaian dengan *F-test*. Kemudian di baris ke-2 adalah metode pengambilan seberapa banyak fitur yang akan diseleksi. Pada penelitian dilakukan seleksi secara relatif 50 % dari jumlah fitur terbaik. Kemudian di baris ke-4 memasukkan fungsi ANOVA yang didapat dari **Scikit-Learn** sebelumnya. Kemudian di baris ke-6 dan baris ke-7 dilakukan seleksi fitur dari data Line-Today yang di ekstraksi dengan *count vectorizer* dan TFIDF. Begitu juga dengan baris ke-9 dan baris ke-10 dilakukan seleksi fitur dari data Twitter yang di ekstraksi dengan *count vectorizer* dan TFIDF.

5.6.3 Mutual Information

Metode seleksi fitur yang lainnya adalah *Mutual Information* (MI). Implementasi MI ditunjukan pada gambar (5.12).

```
1  from sklearn.feature_selection import mutual_info_classif
2  from sklearn.feature_selection import SelectPercentile
3
4  mi = mutual_info_classif
5
6  xcvmi = SelectPercentile(mi, percentile= 50).fit_transform(xcv, y)
7  xtfmi = SelectPercentile(mi, percentile= 50).fit_transform(xtf, y)
8
9  x2cvmi = SelectPercentile(mi, percentile= 50).fit_transform(x2cv, y2)
10 x2tfmi = SelectPercentile(mi, percentile= 50).fit_transform(x2tf, y2)
```

Gambar 5.12: Implementasi program untuk seleksi fitur dengan Mutual Information

Pada baris ke-1 melakukan pemanggilan pada pustaka **Scikit-Learn** dan mendapatkan fungsi MI. Kemudian di baris ke-2 adalah metode pengambilan seberapa banyak fitur yang akan diseleksi. Pada penelitian dilakukan seleksi secara relatif 50 % dari jumlah fitur terbaik. Kemudian di baris ke-4 memasukkan fungsi MI yang didapat dari **Scikit-Learn** sebelumnya. Kemudian di baris ke-6 dan baris ke-7 dilakukan seleksi fitur dari data Line-Today yang di ekstraksi dengan *count vectorizer* dan TFIDF. Begitu juga dengan baris ke-9 dan baris ke-10 dilakukan seleksi fitur dari data Twitter yang di ekstraksi dengan *count vectorizer* dan TFIDF.

5.7 Klasifikasi

Proses selanjutnya membuat fungsi pemodelan klasifikasi yang akan digunakan *cross validation* semua jenis data hasil ekstraksi fitur dan seleksi fitur dengan semua masing-masing model pengklasifikasi Random Forest, Support Vector Machine, Decision Tree, Extra tree, K-Nearest Neighbour, Multinomial Naive Bayes, Gaussian Naive Bayes, Logistic Regression, Neural Network dengan Multi-Layer Perceptron, Ada Boost, dan Gradient Boosting. Implementasi fungsi model pengklasifikasi ditunjukan pada gambar (5.13).

```python
seed = 17
import numpy as np
'''#Random Forest Classifier'''
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(n_estimators=100, random_state=seed)

'''Support Vector Machines'''
from sklearn.svm import SVC
svm = SVC(gamma = 'auto', kernel='linear',  random_state=seed)

'''Decision Tree Classifier'''
from sklearn.tree import DecisionTreeClassifier
dt = DecisionTreeClassifier(random_state=seed)

'''Extra Tree Classifier'''
from sklearn.ensemble import ExtraTreesClassifier
etc = ExtraTreesClassifier(n_estimators=100, random_state=seed)

'''K-Nearest Neighbour'''
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier()

'''Multinomial Naive Bayes'''
from sklearn.naive_bayes import MultinomialNB
mnb = MultinomialNB()

'''Gaussian Naive Bayes'''
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()

'''Logistic Regression'''
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression(solver='saga', random_state=seed)

'''Neural Network'''
from sklearn.neural_network import MLPClassifier
nn = MLPClassifier(random_state=seed)

'''ADA Boosting'''
from sklearn.ensemble import AdaBoostClassifier
abc = AdaBoostClassifier(n_estimators=100, random_state=seed)

'''Gradient Boosting'''
from sklearn.ensemble import GradientBoostingClassifier
gbc = GradientBoostingClassifier(n_estimators=100, random_state=seed)
```

Gambar 5.13: Implementasi program untuk fungsi pengklasifikasian

Pada pengklasifikasi *ensemble* seperti Random Forest, Extra Tree, Ada Boost, dan Gradient Boosting menambahkan parameter yang sama seperti **n_estimator** dan **random state**. Alasan penggunaan **n_estimator** sebanyak 100 karena nilai *default* sebelumnya pada pengklasifikasi *ensemble* adalah 100. Kemudian pada pengklasifikasi Support Vector Machine ditambahkan parameter **gamma** menjadi 'auto' karena menunjukkan bahwa tidak ada nilai eksplisit gamma yang diteruskan. Kemudian parameter di SVM berikutnya **kernel** menggunaan 'linear' karena dalam penelitian lebih difokuskan pada Linear SVM di penelitian. Selanjutnya pada pengklasifikasi Logistic Regression ditambahkan parameter **solver** menjadi 'saga' karena lebih cepat pada data yang relatif besar (Pedregosa et al., 2011).

Kemudian dilanjutkan dengan mencari tahu bagaimana dari masing-masing metode pengklasifikasi bekerja beserta contoh hasil prediksi pengklasifikasi. Berikut implementasi untuk fungsi probabilitas prediksi pengklasifikasi ditunjukan pada

gambar (5.14).

```
1   def importances(model, cv, tf, x, y):
2       model.fit(cv,y)
3       probcv = model.predict_proba(cv)
4       model.fit(tf,y)
5       probtf = model.predict_proba(tf)
6       tab_prob = pd.DataFrame({"fitur" : x, "prob positif (cv)" : probcv[:,1], //
7       "prob negatif (cv)" : probcv[:,0], "prob positif (tf)" : probtf[:,1], //
8       "prob negatif (tf)" : probtf[:,0], "fakta" : y}).round(2)
9
10      return tab_prob
```

Gambar 5.14: Implementasi untuk fungsi probabilitas prediksi pengklasifikasi

Pada baris ke-1, dimasukkan variabel yang terdiri dari jenis model pengklasifikasi yang digunakan, fitur yang telah diekstraksi menggunakan Count Vectorizer dan TFIDF, fitur yang belum diekstraksi untuk menampilkan fitur dalan bentuk teks, dan kelas sentimen yang telah dilabeli sebelumnya. Pada baris ke-2 sampai ke-5 dilakukan pelatihan fitur dari ekstraksi Count Vectorizer dan TFIDF, kemudian fitur dicari probabilitas dari prediktor model pengklasifikasi. Kemudian di baris ke-6 sampai baris ke-8 akan dibentuk tabel dengan kolom "fitur" yang berisi nama fitur, "prob positif (cv)" yang berisi probabilitas fitur dari Count Vectorizer bernilai "1", "prob negatif (cv)" yang berisi probabilitas fitur dari Count Vectorizer bernilai "0", "prob positif (tf)" yang berisi probabilitas fitur dari TFIDF bernilai "1", "prob negatif (tf)" yang berisi probabilitas fitur dari TFIDF bernilai "0", dan "fakta" yang berisi nilai sentimen sebenarnya dari dataset.

5.8 Validasi

Proses berikutnya melakukan Pengujian akan dilakukan dengan menggunakan metode *10-fold cross validation*. Secara *stratified* data akan dibagi menjadi 10 bagian. Kemudian satu bagian akan diujikan sebagai data uji, sedangkan bagian lainya akan digunakan sebagai data latih. Hasil pengujian didapatkan yaitu akurasi, presisi, *recall*, *f-measure*nya, dan lama melakukan proses *running*. Implementasi fungsi penilaian *cross validation* ditunjukan pada gambar (5.15).

```
1   def x_val_accuracy(model,x,y):
2       from sklearn.model_selection import cross_val_score
3       x_val_score = cross_val_score(model, x, y, cv = 10, scoring = 'accuracy').mean()
4       x_val_score = np.round(x_val_score*100, 2)
5       return x_val_score
6
7   def x_val_precision(model,x,y):
8       from sklearn.model_selection import cross_val_score
9       x_val_score = cross_val_score(model, x, y, cv = 10, scoring = 'precision').mean()
10      x_val_score = np.round(x_val_score*100, 2)
11      return x_val_score
12
13  def x_val_recall(model,x,y):
14      from sklearn.model_selection import cross_val_score
15      x_val_score = cross_val_score(model, x, y, cv = 10, scoring = 'recall').mean()
16      x_val_score = np.round(x_val_score*100, 2)
17      return x_val_score
18
19  def x_val_f1(model,x,y):
20      from sklearn.model_selection import cross_val_score
21      x_val_score = cross_val_score(model, x, y, cv = 10, scoring = 'f1').mean()
22      x_val_score = np.round(x_val_score*100, 2)
23      return x_val_score
24
25  def runtime(model,x,y):
26      start = time.time()
27      from sklearn.model_selection import cross_val_score
28      x_val_score = cross_val_score(model, x, y, cv = 10, scoring = 'accuracy').mean()
29      x_val_score = np.round(x_val_score*100, 2)
30      runtime = time.time() - start
31      return np.round(runtime, 2)
```

Gambar 5.15: Implementasi program untuk fungsi penilaian *cross-validation*

Pada baris ke-1 adalah pembuatan fungsi **x_val_accuracy** berdasarkan input yang terdiri dari model pengklasifikasi yang dilakukan, x sebagai fitur dari data, dan y nilai sentimen. Kemudian di baris ke-2 melakukan pemanggilan pustaka **Scikit Learn** dan fungsi **cross_val_score**. Kemudian di baris ke-3 melakukan penghitungan skor *cross validation* berdasarkan parameter model pengklasifikasi yang digunakan, x sebagai fitur dari data, y nilai sentimen, cv dengan jumlah 10 sebagai jumlah *fold* yang dilakukan, dan **scoring** menggunakan 'accuracy' untuk memberikan nilai akurasinya. Pada fungsi **x_val_precision, x_val_recall, dan x_val_f1** pada baris ke-7, ke-13, dan ke-19 hanya mengubah parameter **scoring** nya saja. Kemudian dari 10 kali *cross validation* akan di rata-rata nilainya dengan perintah **mean()**. Kemudian di baris ke-4, nilai akhir dari *cross validation* akan dilakukan pembulatan 2 digit menggunakan perintah **round()**. Kemudian di baris ke-25 akan dibuat fungsi **runtime()** untuk menghitung waktu yang dibutuhkan untuk melakukan *running*. Pada bagian

antara baris ke-26 dan baris ke-29 mendapatkan waktu sebelum dimulainya *cross validation* dan selisih waktu akhir dengan waktu sebelum dimulai. Kemudian di baris ke-31 melakukan keluaran dengan pembulatan 2 digit.

BAB VI

HASIL DAN PEMBAHASAN

6.1 Proses Pengumpulan Data

6.1.1 Line-Today

Tahap pengumpulan data dengan cara mengunduh data dari Line-Today menggunakan 10 besar artikel dan artikel yang berkaitan dengan pemilu. Tahap pengumpulan data telah dijelaskan pada bab sebelumnya. Data tweet yang telah dikumpulkan kemudian disimpan dalam file format $.csv$. Tabel (6.1) menunjukkan cuplikan data Line-Today yang berhasil dikumpulkan.

Tabel 6.1: Cuplikan data Line-Today yang dikumpulkan

Komentar
kita jangan salahkan kedua belah pihak karna apa hasil resminyaa itu tanggal 22 jadi kalian sabar aja gausah saling tuduh pendukung kedua belah pihak, gimana indonesia mau lebih maju kalau perpecahan seperti ini ada dimana mana.
Kasihan banget. Tuhan ampuni mereka sebab apa yang mereka perbuat mrk tidak tahu.
I came here for Bobby!
Lebih cinta kalau pemimpin orang Indonesia asli, bukan keturunan etnis tertentu
Siapapun presiden nya tidak bakal bikin gue tidak remedial fisika lagi.
Count saja blm keluar resmi dri kpu ?? lu ngitung apaan? Line tudey pendukung 02 lu iya? Hahaha
KPU selalu salah di mata yang kalah :(

6.1.2 Twitter

Tahap pengumpulan data dengan cara mengunduh data dari Twitter menggunakan kata kunci dan tanggal yang berkaitan dengan pemilu. Tahap pengumpulan data telah dijelaskan pada bab sebelumnya. Data tweet yang telah dikumpulkan kemudian disimpan dalam file format $.csv$. Tabel (6.2) menunjukkan cuplikan data Twitter yang berhasil dikumpulkan.

Tabel 6.2: Cuplikan data Twitter yang dikumpulkan

Tweet
Pak prabowo- sandi yg menang Real count dan tak terbnthkn TheVictoryOfPrabowo
VictoryForPrabowo kawal suara kami di cimahi bogor @fadlizon @prabowo @FaldoMaldini @sandiuno pic.twitter.com/wNhUS7Uymc
Pak @Jokowi lahir dalam keluarga muslim, gemar membangun silaturrahmi, serta memiliki kepedulian tinggi terhadap masyarakat muslim. PilihOrangBaik PilihYgJelasIslamnya PilihYgBajuPutih @jokowi Gaspol @saaebunglon @bocahsosmed @sukangetweet
Kl @prabowo dan @sandiuno ingin menang, relawan 02 harus siap menjaga hasil perhitungan TPS minimal sampai tingkat kabupaten. Jangan pernah lepaskan pengawasan walaupun sedetik sebelum sampai tingkat minimal kabupaten kota. Biar pengecekan Situng lebih mudah. Good Luck.
Tunggu hasil real count, jgn trpengaruh quick count pesanan
Tiga tahun di Aceh, karakter keislaman Pak @Jokowi digembleng. PilihOrangBaik PilihYgJelasIslamnya PilihYgBajuPutih https://www.gesuri.id/pemilu/3-tahun-di-aceh-karakter-keislaman-jokowi-digembleng-b1WcIZivOÂ â¦
Mari Bersama Sama Berdoa Team @jokowi Untuk Kemenangan Bangsa Indonesia NO1

6.2 Preprocessing

6.2.1 Case Folding

Tahap *case folding* digunakan membuat semua teks menjadi ke bentuk *lower-case*. Tabel (6.3) dan (6.4) menunjukkan cuplikan data Line-Today dan Twitter yang di *Case Folding*.

Tabel 6.3: Cuplikan data Line-Today yang di *case folding*

Komentar
kita jangan salahkan kedua belah pihak karna apa hasil resminyaa itu tanggal 22 jadi kalian sabar aja gausah saling tuduh pendukung kedua belah pihak, gimana indonesia mau lebih maju kalau perpecahan seperti ini ada dimana mana.
kasihan banget. tuhan ampuni mereka sebab apa yang mereka perbuat mrk tidak tahu.
i came here for bobby!
lebih cinta kalau pemimpin orang indonesia asli, bukan keturunan etnis tertentu
siapapun presiden nya tidak bakal bikin gue tidak remedial fisika lagi.
count saja blm keluar resmi dri kpu ?? lu ngitung apaan? line tudey pendukung 02 lu iya? hahaha
kpu selalu salah di mata yang kalah :(

Tabel 6.4: Cuplikan data Twitter yang di *case folding*

Tweet
pak prabowo- sandi yg menang real count dan tak terbnthkn thevictoryofprabowo
victoryforprabowo kawal suara kami di cimahi bogor @fadlizon @prabowo @faldomaldini @sandiuno pic.twitter.com/wnhus7uymc
pak @jokowi lahir dalam keluarga muslim, gemar membangun silaturrahmi, serta memiliki kepedulian tinggi terhadap masyarakat muslim. pilihorangbaik pilihygjelasislamnya pilihygbajuputih @jokowi gaspol @saaebunglon @bocahsosmed @sukangetweet
kl @prabowo dan @sandiuno ingin menang, relawan 02 harus siap menjaga hasil perhitungan tps minimal sampai tingkat kabupaten. jangan pernah lepaskan pengawasan walaupun sedetik sebelum sampai tingkat minimal kabupaten kota. biar pengecekan situng lebih mudah. good luck.
tunggu hasil real count, jgn trpengaruh quick count pesanan
tiga tahun di aceh, karakter keislaman pak @jokowi digembleng. pilihorangbaik pilihygjelasislamnya pilihygbajuputih https://www.gesuri.id/pemilu/3-tahun-di-aceh-karakter-keislaman-jokowi-digembleng-b1wcizivoâ â¦
mari bersama sama berdoa team @jokowi untuk kemenangan bangsa indonesia no1

6.2.2 Punctuation Removal

Tahap *punctuation removal* digunakan memilih teks saja. Tabel (6.5) dan (6.6) merupakan cuplikan data Line-Today dan Twitter yang di *punctuation removal*.

Tabel 6.5: Cuplikan data Line-Today yang di *punctuation removal*

Komentar
kita jangan salahkan kedua belah pihak karna apa hasil resminyaa itu tanggal 22 jadi kalian sabar aja gausah saling tuduh pendukung kedua belah pihak gimana indonesia mau lebih maju kalau perpecahan seperti ini ada dimana mana
kasihan banget tuhan ampuni mereka sebab apa yang mereka perbuat mrk tidak tahu
i came here for bobby!
lebih cinta kalau pemimpin orang indonesia asli bukan keturunan etnis tertentu
siapapun presiden nya tidak bakal bikin gue tidak remedial fisika lagi
count saja blm keluar resmi dri kpu lu ngitung apaan line tudey pendukung 02 lu iya hahaha
kpu selalu salah di mata yang kalah

Tabel 6.6: Cuplikan data Twitter yang di *whitespace removal*

Tweet
pak prabowo sandi yg menang real count dan tak terbnthkn thevictoryofprabowo
victoryforprabowo kawal suara kami di cimahi bogor fadlizon prabowo faldomaldini sandiuno pictwittercomwnhus7uymc
pak jokowi lahir dalam keluarga muslim gemar membangun silaturrahmi serta memiliki kepedulian tinggi terhadap masyarakat muslim pilihorangbaik pilihygjelasislamnya pilihygbajuputih jokowi gaspol saaebunglon bocahsosmed sukangetweet
kl prabowo dan sandiuno ingin menang relawan 02 harus siap menjaga hasil perhitungan tps minimal sampai tingkat kabupaten jangan pernah lepaskan pengawasan walaupun sedetik sebelum sampai tingkat minimal kabupaten kota biar pengecekan situng lebih mudah good luck
tunggu hasil real count jgn trpengaruh quick count pesanan
tiga tahun di aceh karakter keislaman pak jokowi digembleng pilihorangbaik pilihygjelasislamnya pilihygbajuputih httpswwwgesuriidpemi-lu3tahundiacehkarakterkeislamanjokowidigemblengb1wcizivoâ â
mari bersama sama berdoa team jokowi untuk kemenangan bangsa indonesia no1

6.2.3 Whitespace Removal

Tahap *whitespace removal* digunakan untuk menghapus ruang kosong. Tabel (6.7) dan (6.8) merupakan cuplikan data Line-Today dan Twitter yang di *whitespace removal*.

Tabel 6.7: Cuplikan data Line-Today yang di *whitespace removal*

Komentar
kita jangan salahkan kedua belah pihak karna apa hasil resminyaa itu tanggal 22 jadi kalian sabar aja gausah saling tuduh pendukung kedua belah pihak gimana indonesia mau lebih maju kalau perpecahan seperti ini ada dimana mana
kasihan banget tuhan ampuni mereka sebab apa yang mereka perbuat mrk tidak tahu
i came here for bobby
lebih cinta kalau pemimpin orang indonesia asli bukan keturunan etnis tertentu
siapapun presiden nya tidak bakal bikin gue tidak remedial fisika lagi
count saja blm keluar resmi dri kpu lu ngitung apaan line tudey pendukung 02 lu iya hahaha
kpu selalu salah di mata yang kalah

Tabel 6.8: Cuplikan data Twitter yang di *whitespace removal*

Tweet
pak prabowo sandi yg menang real count dan tak terbnthkn thevictoryofprabowo
victoryforprabowo kawal suara kami di cimahi bogor fadlizon prabowo faldomaldini sandiuno pictwittercomwnhus7uymc
pak jokowi lahir dalam keluarga muslim gemar membangun silaturrahmi serta memiliki kepedulian tinggi terhadap masyarakat muslim pilihorangbaik pilihygjelasislamnya pilihygbajuputih jokowi gaspol saaebunglon bocahsosmed sukangetweet
kl prabowo dan sandiuno ingin menang relawan 02 harus siap menjaga hasil perhitungan tps minimal sampai tingkat kabupaten jangan pernah lepaskan pengawasan walaupun sedetik sebelum sampai tingkat minimal kabupaten kota biar pengecekan situng lebih mudah good luck
tunggu hasil real count jgn trpengaruh quick count pesanan
tiga tahun di aceh karakter keislaman pak jokowi digembleng pilihorangbaik pilihygjelasislamnya pilihygbajuputih httpswwwgesuriidpemi-lu3tahundiacehkarakterkeislamanjokowidigemblengb1wcizivoâ â
mari bersama sama berdoa team jokowi untuk kemenangan bangsa indonesia no1

6.2.4 Stopwords Removal

Tahap *stopwords removal* digunakan untuk menghapus kata yang tidak memiliki makna. Tabel (6.9) dan (6.10) merupakan cuplikan data Line-Today dan Twitter yang di *stopword removal*.

Tabel 6.9: Cuplikan data Line-Today yang di *stopword removal*

Komentar
salahkan belah karna hasil resminyaa tanggal 22 sabar aja gausah tuduh pendukung belah gimana indonesia maju perpecahan dimana
kasihan banget tuhan ampuni perbuat mrk
i came here for bobby
cinta pemimpin orang indonesia asli keturunan etnis
presiden nya bikin gue remedial fisika
count blm resmi dri kpu lu ngitung line tudey pendukung 02 lu iya hahaha
kpu salah mata kalah

Tabel 6.10: Cuplikan data Twitter yang di *stopword removal*

Tweet
prabowo sandi yg menang real count terbnthkn thevictoryofprabowo
victoryforprabowo kawal suara cimahi bogor fadlizon prabowo faldomaldini sandiuno pictwittercomwnhus7uymc
jokowi lahir keluarga muslim gemar membangun silaturrahmi memiliki kepedulian masyarakat muslim pilihorangbaik pilihygjelasislamnya pilihygbajuputih jokowi gaspol saaebunglon bocahsosmed sukangetweet
kl prabowo sandiuno menang relawan 02 menjaga hasil perhitungan tps minimal tingkat kabupaten lepaskan pengawasan sedetik tingkat minimal kabupaten kota biar pengecekan situng mudah good luck
tunggu hasil real count jgn trpengaruh quick count pesanan
aceh karakter keislaman jokowi digembleng pilihorangbaik pilihygjelasislamnya pilihygbajuputih httpswwwgesuriidpemi-lu3tahundiacehkarakterkeislamanjokowidigemblengb1wcizivoâ â
mari berdoa team jokowi kemenangan bangsa indonesia no1

6.2.5 Stemming

Tahap *stemming* digunakan untuk membuat kata menjadi bentuk dasar. Tabel (6.11) dan (6.12) merupakan cuplikan data Line-Today dan Twitter yang di *stopword*

removal.

Tabel 6.11: Cuplikan data Line-Today yang di *stemming*

Komentar
salah belah karna hasil resminyaa tanggal 22 sabar aja gausah tuduh dukung belah gimana indonesia maju pecah mana
kasihan banget tuhan ampun buat mrk
i came here for bobby
cinta pimpin orang indonesia asli turun etnis
presiden nya bikin gue remedial fisika
count blm resmi dri kpu lu ngitung line tudey dukung 02 lu iya hahaha
kpu salah mata kalah

Tabel 6.12: Cuplikan data Twitter yang di *stemming*

Tweet
prabowo sandi yg menang real count terbnthkn thevictoryofprabowo
victoryforprabowo kawal suara cimahi bogor fadlizon prabowo faldomaldini sandiuno pictwittercomwnhus7uymc
jokowi lahir keluarga muslim gemar bangun silaturrahmi milik peduli masyarakat muslim pilihorangbaik pilihygjelasislamnya pilihygbajuputih jokowi gaspol saaebunglon bocahsosmed sukangetweet
kl prabowo sandiuno menang rawan 02 jaga hasil hitung tps minimal tingkat kabupaten lepas awas detik tingkat minimal kabupaten kota biar kece situng mudah good luck
tunggu hasil real count jgn trpengaruh quick count pesan
aceh karakter islam jokowi gembleng pilihorangbaik pilihygjelasislamnya pilihygbajuputih httpswwwgesuriidpemi-lu3tahundiacehkarakterkeislamanjokowidigemblengb1wcizivo
mari doa team jokowi menang bangsa indonesia no1

6.3 Pelabelan Data

Data kemudian akan diberi label secara otomatis seperti yang sudah dijelaskan pada gambar (4.10) dan akan diperbaiki secara manual apabila terdapat nilai sentimen

yang kurang tepat. Berikut pada tabel (6.13) dan (6.14) data Line-Today dan Twitter yang telah diberi label.

Tabel 6.13: Cuplikan data Line-Today yang di label

Komentar	sentimen
salah belah karna hasil resminyaa tanggal 22 sabar aja gausah tuduh dukung belah gimana indonesia maju pecah mana	1
kasihan banget tuhan ampun buat mrk	1
i came here for bobby	0
cinta pimpin orang indonesia asli turun etnis	1
presiden nya bikin gue remedial fisika	0
count blm resmi dri kpu lu ngitung line tudey dukung 02 lu iya hahaha	1
kpu salah mata kalah	0

Tabel 6.14: Cuplikan data Twitter yang di label

Tweet	sentimen
prabowo sandi yg menang real count terbnthkn thevictoryofprabowo	1
victoryforprabowo kawal suara cimahi bogor fadlizon prabowo faldomaldini sandiuno pictwittercomwnhus7uymc	1
jokowi lahir keluarga muslim gemar bangun silaturrahmi milik peduli masyarakat muslim pilihorangbaik pilihygjelasislamnya pilihygbajuputih jokowi gaspol saaebunglon bocahsosmed sukangetweet	1
kl prabowo sandiuno menang rawan 02 jaga hasil hitung tps minimal tingkat kabupaten lepas awas detik tingkat minimal kabupaten kota biar kece situng mudah good luck	1
tunggu hasil real count jgn trpengaruh quick count pesan	0
aceh karakter islam jokowi gembleng pilihorangbaik pilihygjelasislamnya pilihygbajuputih httpswwwgesuriidpemi-lu3tahundiacehkarakterkeislamanjokowidigemblengb1wcizivo	0
mari doa team jokowi menang bangsa indonesia no1	1

Label 1 menunjukkan bahwa komentar atau *tweet* memiliki sentimen positif,

sedangkan label 0 menunjukkan bahwa komentar atau *tweet* tersebut memiliki sentimen polaritas negatif. Label ini nantinya akan digunakan sebagai tolak ukur komentar atau *tweet* dikatakan positif atau negatif pada saat pelatihan data, dan juga akan digunakan sebagai validasi pada saat pengujian data.

6.4 Ekstraksi Fitur

Setelah selesai tahap *preprocessing*, data lalu diubah menjadi bentuk vektor menggunakan ekstraksi Count Vectorizer dan TFIDF. Berikut pada gambar (6.1) dan (6.2) cuplikan data pada ekstraksi fitur Count Vectorizer pada data Line-Today dan Twitter.

	000	0008	01	0101dirojer	010647	01berarti	01bomat	01lebih	01njgn	...	yuk	yv	ywda	zaman	zholimi	zikir	znnbukan	zonk	zz
0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5455	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5456	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5457	0	0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5458	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5459	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5460	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5461	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0

5462 rows × 6241 columns

Gambar 6.1: Cuplikan ekstraksi fitur Count Vectorizer pada Line-Today

	000	000811	002	003	004	0047	007	008	01	...	zonk	zonkga	zonkk	zonkkk	ztw868992	zuhdi	zul	zulkifli	zumba
0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	1	...	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5722	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5723	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5724	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5725	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5726	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5727	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5728	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0

5729 rows × 11798 columns

Gambar 6.2: Cuplikan ekstraksi fitur Count Vectorizer pada Twitter

Kemudian akan dilakukan juga sebagai perbandingan, melakukan ekstraksi fitur dengan TFIDF. Berikut pada gambar (6.3) dan (6.4) cuplikan data pada ekstraksi fitur TFIDF pada data Line-Today dan Twitter.

	000	0008	01	0101dirojer	010647	01berarti	01bomat	01lebih	01njgn	...	yuk	yv	ywda	zaman	zholimi	zikir	znnbukan	zonk	zz
0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.343082	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5455	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5456	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5457	0.0	0.0	0.238431	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5458	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5459	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5460	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5461	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

5462 rows × 6241 columns

Gambar 6.3: Cuplikan ekstraksi fitur TFIDF pada Line-Today

	000	000811	002	003	004	0047	007	008	01	...	zonk	zonkga	zonkk	zonkkk	ztw868992	zuhdi	zul	zulkifli	zumba
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.153509	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5722	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5723	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5724	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5725	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5726	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5727	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5728	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

5729 rows × 11798 columns

Gambar 6.4: Cuplikan ekstraksi fitur TFIDF pada Twitter

Pada keempat gambar dari gambar (6.1) sampai gambar (6.4) tersebut memiliki judul kolom berupa fitur yang ada pada kelompok data tersebut. Pada bagian samping kiri merupakan total dokumen pada data tersebut. Semakin banyak dokumen yang dimasukkan maka semakin banyak juga fitur dan indeks dokumen yang dihasilkan. Nilai Count Vectorizer dan TFIDF "0" muncul apabila kata yang bersangkutan tidak muncul di dokumen manapun atau fitur muncul di semua dokumen. Perbedaan Count Vectorizer adalah bahwa Count Vectorizer hanya melabeli "1" dan "0" untuk mengetahui kumpulan teks ada atau tidak pada kalimat, sedangkan TFIDF memberikan pembobotan berdasarkan seluruh kalimat pada data. Pada keempat gambar tersebut ditunjukkan bahwa ekstraksi Count Vectorizer dan TFIDF berhasil dijadikan sebagai kelompok data untuk masukan ke dalam pengklasifikasi. Kemudian selanjutnya dilanjutkan ke tahap seleksi data.

6.5 Seleksi Fitur

Setelah selesai tahap preprocessing, data lalu diubah menjadi bentuk vektor menggunakan ekstraksi Chi Square, ANOVA, dan Mutual Information. Berikut pada gambar (6.5), (6.6), (6.7) dan (6.8) cuplikan data pada seleksi fitur Chi Square pada data Line-Today dan Twitter serta masing-masing metode ekstraksi fiturnya.

	000	01	010647	01berarti	01nlanjutkan	01nnegara	02	02berarti	0812	...	youtube	yra	yu	yuhuiiii	yv	ywda	zaman	zholimi	zikir
0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5455	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5456	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5457	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5458	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5459	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5460	0	0	0	0	0	0	4	0	0	...	0	0	0	0	0	0	0	0	0
5461	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0

5462 rows × 3120 columns

Gambar 6.5: Cuplikan seleksi fitur Chi Square pada ekstraksi Count Vectorizer Line-Today

	01	010647	01berarti	01nlanjutkan	01nnegara	01optimis	02	02berarti	0812	...	yok	your	youtube	yra	ysmg	yu	yuhuiiii	yv	ywda
0	0.000000	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.000000	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.343082	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.000000	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.000000	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.000000	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.000000	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5455	0.000000	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5456	0.000000	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5457	0.238431	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5458	0.000000	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5459	0.000000	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5460	0.000000	0.0	0.0	0.0	0.0	0.0	0.356197	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5461	0.000000	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

5462 rows × 3120 columns

Gambar 6.6: Cuplikan seleksi fitur Chi Square pada ekstraksi TFIDF Line-Today

	000	000811	003	0047	007	008	01	0102	01062016	...	yourturnbro	youtube	yudi	yuk	yukafiru	yuks	zarazettirazr	ziarah	zul
0	0	0	0	0	0	0	0	0	0	...	0	1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	1	0	0	...	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5722	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5723	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5724	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5725	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5726	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5727	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5728	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0

5729 rows × 5899 columns

Gambar 6.7: Cuplikan seleksi fitur Chi Square pada ekstraksi Count Vectorizer Twitter

	000	000811	003	0047	007	008	01	0102	016	...	yukafiru	yukcoblosuntukindonesia	yukk	yuks	yuuuuk	zarazettirazr	ziarah	ztw868992	z
0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.153509	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
5722	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
5723	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
5724	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
5725	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
5726	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
5727	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
5728	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0

5729 rows × 5899 columns

Gambar 6.8: Cuplikan seleksi fitur Chi Square pada ekstraksi TFIDF Twitter

Kemudian juga dilakukan seleksi fitur ANOVA sebagai perbandingan. Berikut pada gambar (6.9), (6.10), (6.11) dan (6.12) cuplikan data pada seleksi fitur ANOVA pada data Line-Today dan Twitter beserta masing-masing metode ekstraksi fiturnya.

	000	01	010647	01berarti	01nlanjutkan	01nnegara	02	02berarti	0812	...	youtube	yra	yu	yuhuiiii	yv	ywda	zaman	zholimi	zikir
0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5455	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5456	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5457	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5458	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5459	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5460	0	0	0	0	0	0	4	0	0	...	0	0	0	0	0	0	0	0	0
5461	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0

5462 rows × 3120 columns

Gambar 6.9: Cuplikan seleksi fitur ANOVA pada ekstraksi Count Vectorizer Line-Today

	000	01	010647	01berarti	01nlanjutkan	01nnegara	02	02berarti	0812	...	yra	yth	yu	yuhuiiii	yv	ywda	zaman	zholimi	zikir
0	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.343082	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5455	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5456	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5457	0.0	0.238431	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5458	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5459	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5460	0.0	0.000000	0.0	0.0	0.0	0.0	0.356197	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5461	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

5462 rows × 3120 columns

Gambar 6.10: Cuplikan seleksi fitur ANOVA pada ekstraksi TFIDF Line-Today

	000	000811	003	004	0047	007	008	01	0102	...	zen	ziarah	zmn	zonkga	zonkk	zonkkk	zuhdi	zul	zumba
0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	1	0	...	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5722	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5723	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5724	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5725	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5726	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5727	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5728	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0

5729 rows × 5899 columns

Gambar 6.11: Cuplikan seleksi fitur ANOVA pada ekstraksi Count Vectorizer Twitter

	000811	003	0047	007	008	01	0102	01062016	0137	...	zen	ziarah	zmn	zonkga	zonkk	zonkkk	zuhdi	zul	zumba
0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.153509	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5722	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5723	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5724	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5725	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5726	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5727	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5728	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

5729 rows × 5899 columns

Gambar 6.12: Cuplikan seleksi fitur ANOVA pada ekstraksi TFIDF Twitter

Lalu juga dilakukan seleksi fitur menggunakan metode Mutual Information. Berikut pada gambar (6.13), (6.14), (6.15) dan (6.16) cuplikan data pada seleksi fitur Mutual Information pada data Line-Today dan Twitter beserta masing-masing metode ekstraksi fiturnya.

	01	0101dirojer	01bomat	01lebih	01nlanjutkan	01nnegara	01optimis	02	02berarti	...	yowesh	yth	yu	yuhuiiii	yuhuuuuu	yuhuuuuuuuuu	yv	zon
0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	
5	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	
6	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
5455	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	
5456	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	
5457	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	
5458	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	
5459	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	
5460	0	0	0	0	0	4	0	0	0	...	0	0	0	0	0	0	0	
5461	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	

5462 rows × 3120 columns

Gambar 6.13: Cuplikan seleksi fitur Mutual Information pada ekstraksi Count Vectorizer Line-Today

	000	01	010647	01berarti	01nlanjutkan	01nnegara	02	02berarti	0812	...	yra	yth	yu	yuhuiiii	yv	ywda	zaman	zholimi	zikir
0	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.343082	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5455	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5456	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5457	0.0	0.238431	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5458	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5459	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5460	0.0	0.000000	0.0	0.0	0.0	0.0	0.356197	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5461	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

5462 rows × 3120 columns

Gambar 6.14: Cuplikan seleksi fitur Mutual Information pada ekstraksi TFIDF Line-Today

	000	000811	003	004	0047	007	008	01	0102	...	zen	ziarah	zmn	zonkga	zonkk	zonkkk	zuhdi	zul	zumba
0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	1	0	...	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5722	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5723	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5724	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5725	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5726	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5727	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5728	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0

5729 rows × 5899 columns

Gambar 6.15: Cuplikan seleksi fitur Mutual Information pada ekstraksi Count Vectorizer Twitter

	000811	003	0047	007	008	01	0102	01062016	0137	...	zen	ziarah	zmn	zonkga	zonkk	zonkkk	zuhdi	zul	zumba
0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.153509	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5722	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5723	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5724	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5725	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5726	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5727	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5728	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

5729 rows × 5899 columns

Gambar 6.16: Cuplikan seleksi fitur Mutual Information pada ekstraksi TFIDF Twitter

Pada seluruh gambar dari gambar (6.5) sampai gambar (6.16) tersebut memiliki judul kolom berupa fitur yang ada pada kelompok data tersebut. Pada bagian samping kiri merupakan total dokumen pada data tersebut. Semakin banyak dokumen yang dimasukkan maka semakin banyak juga fitur dan indeks dokumen yang

dihasilkan. Hasil ekstraksi fitur sebelumnya telah berkurang sebanyak 50 persen dari jumlah fitur yang diseleksi menggunakan metode Chi Square, ANOVA, dan Mutual Information.

6.6 Klasifikasi

Tahap berikutnya adalah melakukan klasifikasi dan melihat probabilitas dari pengklasifikasi pada kedua ekstraksi fitur. Berikut akan ditampilkan cuplikan klasifikasi dengan masing-masing probabilitas pada masing-masing pengklasifikasi seperti Random Forest, Support Vector Machine, Decision Tree, Extra tree, K-Nearest Neighbour, Multinomial Naive Bayes, Gaussian Naive Bayes, Logistic Regression, Neural Network dengan Multi-Layer Perceptron, Ada Boost, dan Gradient Boosting, masing-masing ekstraksi fitur seperti Count Vectorizer dan TFIDF, dan masing-masing data seperti Line-Today dan Twitter. Berikut cuplikan klasifikasi dari semua dataset dari Line-Today ke Twitter dan metode Pengklasifikasi dari Random Forest ke Gradient Boosting yang ditampilkan pada tabel (6.15) sampai tabel (6.36).

Tabel 6.15: Cuplikan Klasifikasi Random Forest pada Line-Today

fitur	prob positif (cv)	prob negatif (cv)	prob positif (tf)	prob negatif (tf)	fakta
moga sktx bukn krna kalah yaaaaa	0.02	0.98	0.01	0.99	0
ulang	0	1	0.01	0.99	0
wkwkwk kasi	0	1	0.01	0.99	0
aduh 5 lembaga survei hasil ny jokowi menang	0.04	0.96	0.04	0.96	0
berani situ hukum mati koruptor usul koruptor kasih uang pensiun iya	0.16	0.84	0.07	0.93	0
ngaruh jamin negara hukum ad dibiarin tahan bs liar uda tahan penjara biar rasain gmna hidup bui berani buat berani tanggung jwb soob jaksa berani tolak jamin tahan prabowo	0.83	0.17	0.73	0.27	1
ciyan	0	1	0	1	0

Tabel 6.16: Cuplikan Klasifikasi Random Forest pada Twitter

fitur	prob positif (cv)	prob negatif (cv)	prob positif (tf)	prob negatif (tf)	fakta
pilih 01 11 11 11 dpd nggak cocok satu jokowi psi id pictwittercomywjuemozkn	0.91	0.09	0.89	0.11	1
calon presiden indonesia prabowo subianto harap menang milu april 2019 hadap saing keras indonesia masuk sulawesi utara kampung halaman ibu httpsnewnaratifcomjournalismpolitikidentitasprabowosharexuna9bf3dbf0c2efd551ca832b9b6800749c pilpres2019	0.96	0.04	0.91	0.09	1
rockygerung debat chally liat dungu rocky	0	1	0.03	0.97	0
jarum ya	0	1	0.04	0.96	0
jd yg jual takut nih	0.85	0.15	0.92	0.08	1
dukung yg byk ginihh kalah kecirangan lawann pictwittercomkc5mwtdmmo	0.19	0.81	0.19	0.81	0
hatihati milih presiden jgn kayak milih kucing karung mending pilih aja pilihorangbaik pilihygjelasislamnya pilihygbajuputih jokowi khmarufamin	1	0	0.98	0.02	1

Dapat dilihat pada tabel (6.16) dan tabel (6.17), RF menciptakan beberapa *tree*. Alih-alih mempertimbangkan semua fitur saat memisahkan *node*, algoritme RF memilih fitur terbaik dari *subset* semua fitur. Pada penelitian ini digunakan paramater banyaknya *tree* sebanyak 100. Banyaknya tersebut yang mengakibatkan probabilitas prediksi menjadi tiap kelipatan 0.01.

Tabel 6.17: Cuplikan Klasifikasi Support Vector Machine pada Line-Today

fitur	prob positif (cv)	prob negatif (cv)	prob positif (tf)	prob negatif (tf)	fakta
moga sktx bukn krna kalah yaaaaa	0.01	0.99	0.03	0.97	0
ulang	0.05	0.95	0.01	0.99	0
wkwkwk kasi	0	1	0	1	0
aduh 5 lembaga survei hasil ny jokowi menang	0.06	0.94	0.28	0.72	0
berani situ hukum mati koruptor usul koruptor kasih uang pensiun iya	0.06	0.94	0.03	0.97	0
ngaruh jamin negara hukum ad dibiarin tahan bs liar uda tahan penjara biar rasain gmna hidup bui berani buat berani tanggung jwb soob jaksa berani tolak jamin tahan prabowo	0.93	0.07	0.8	0.2	1
ciyan	0.06	0.94	0.03	0.97	0

Tabel 6.18: Cuplikan Klasifikasi Support Vector Machine pada Twitter

fitur	prob positif (cv)	prob negatif (cv)	prob positif (tf)	prob negatif (tf)	fakta
pilih 01 11 11 11 dpd nggak cocok satu jokowi psi id pictwittercomywjuemozkn	0.88	0.12	0.9	0.1	1
calon presiden indonesia prabowo subianto harap menang milu april 2019 hadap saing keras indonesia masuk sulawesi utara kampung halaman ibu httpsnewnaratifcomjournalismpolitikidentitasprabowosharexuna9bf3dbf0c2efd551ca832b9b6800749c pilpres2019	0.75	0.25	0.84	0.16	1
rockygerung debat chally liat dungu rocky	0.19	0.81	0.06	0.94	0
jarum ya	0.25	0.75	0.12	0.88	0
jd yg jual takut nih	0.75	0.25	0.92	0.08	1
dukung yg byk ginihh kalah kecirangan lawann pictwittercomkc5mwtdmmo	0.25	0.75	0.12	0.88	0
hatihati milih presiden jgn kayak milih kucing karung mending pilih aja pilihorangbaik pilihygjelasislamnya pilihygbajuputih jokowi khmarufamin	0.96	0.04	0.95	0.05	1

Dapat dilihat pada tabel (6.18) dan tabel (6.19), SVM didasarkan pada gagasan untuk menemukan *hyperplane* yang terbaik membagi dataset menjadi dua kelas. Anda dapat menganggap hyperplane sebagai garis yang secara linear memisahkan dan mengklasifikasikan satu set data. Secara intuitif, semakin jauh dari titik data dari *hyperplane*, semakin yakin SVM telah mengklasifikasi dengan benar. Karena itu probabilitas yang didapatkan adalah jarak dari fitur pada hyperplane.

Tabel 6.19: Cuplikan Klasifikasi Decision Tree pada Line-Today

fitur	prob positif (cv)	prob negatif (cv)	prob positif (tf)	prob negatif (tf)	fakta
moga sktx bukn krna kalah yaaaaa	0	1	0	1	0
ulang	0	1	0	1	0
wkwkwk kasi	0	1	0	1	0
aduh 5 lembaga survei hasil ny jokowi menang	0	1	0	1	0
berani situ hukum mati koruptor usul koruptor kasih uang pensiun iya	0	1	0	1	0
ngaruh jamin negara hukum ad dibiarin tahan bs liar uda tahan penjara biar rasain gmna hidup bui berani buat berani tanggung jwb soob jaksa berani tolak jamin tahan prabowo	1	0	1	0	1
ciyan	0	1	0	1	0

Tabel 6.20: Cuplikan Klasifikasi Decision Tree pada Twitter

fitur	prob positif (cv)	prob negatif (cv)	prob positif (tf)	prob negatif (tf)	fakta
pilih 01 11 11 11 dpd nggak cocok satu jokowi psi id pictwittercomywjuemozkn	1	0	1	0	1
calon presiden indonesia prabowo subianto harap menang milu april 2019 hadap saing keras indonesia masuk sulawesi utara kampung halaman ibu httpsnewnaratifcomjournalismpolitikidentitasprabowosharexuna9bf3dbf0c2efd551ca832b9b6800749c pilpres2019	1	0	1	0	1
rockygerung debat chally liat dungu rocky	0	1	0	1	0
jarum ya	0	1	0	1	0
jd yg jual takut nih	1	0	1	0	1
dukung yg byk ginihh kalah kecirangan lawann pictwittercomkc5mwtdmmo	0	1	0	1	0
hatihati milih presiden jgn kayak milih kucing karung mending pilih aja pilihorangbaik pilihygjelasislamnya pilihygbajuputih jokowi khmarufamin	1	0	1	0	1

Dapat dilihat pada tabel (6.19) dan tabel (6.20), DT adalah alat pendukung keputusan yang menggunakan grafik atau model keputusan seperti pohon dan kemungkinan konsekuensinya. Karena didasarkan hanya pada 1 *tree* saja, maka probabilitas yang ditampilkan DT bersifat 1 atau 0 saja.

Tabel 6.21: Cuplikan Klasifikasi Extra Tree pada Line-Today

fitur	prob positif (cv)	prob negatif (cv)	prob positif (tf)	prob negatif (tf)	fakta
moga sktx bukn krna kalah yaaaaa	0	1	0	1	0
ulang	0	1	0	1	0
wkwkwk kasi	0	1	0	1	0
aduh 5 lembaga survei hasil ny jokowi menang	0	1	0	1	0
berani situ hukum mati koruptor usul koruptor kasih uang pensiun iya	0	1	0	1	0
ngaruh jamin negara hukum ad dibiarin tahan bs liar uda tahan penjara biar rasain gmna hidup bui berani buat berani tanggung jwb soob jaksa berani tolak jamin tahan prabowo	1	0	1	0	1
ciyan	0	1	0	1	0

Tabel 6.22: Cuplikan Klasifikasi Extra Tree pada Twitter

fitur	prob positif (cv)	prob negatif (cv)	prob positif (tf)	prob negatif (tf)	fakta
pilih 01 11 11 11 dpd nggak cocok satu jokowi psi id pictwittercomywjuemozkn	1	0	1	0	1
calon presiden indonesia prabowo subianto harap menang milu april 2019 hadap saing keras indonesia masuk sulawesi utara kampung halaman ibu httpsnewnaratifcomjournalismpolitikidentitasprabowosharexuna9bf3dbf0c2efd551ca832b9b6800749c pilpres2019	1	0	1	0	1
rockygerung debat chally liat dungu rocky	0	1	0	1	0
jarum ya	0	1	0	1	0
jd yg jual takut nih	1	0	1	0	1
dukung yg byk ginihh kalah kecirangan lawann pictwittercomkc5mwtdmmo	0	1	0	1	0
hatihati milih presiden jgn kayak milih kucing karung mending pilih aja pilihorangbaik pilihygjelasislamnya pilihygbajuputih jokowi khmarufamin	1	0	1	0	1

Dapat dilihat pada tabel (6.21) dan tabel (6.22), Extra Tree (*Extremely Randomized Tree*) melakukan pengacakan lebih lanjut *tree* dalam konteks fitur masukan secara numerik, di mana pilihan titik potong optimal bertanggung jawab atas sebagian besar dari varian *tree* yang diinduksi. Namun karena sama seperti DT sebelumnya, Extra Tree hanya berlandaskan satu *tree* saja sehingga probabilitas prediksi hanya 1 atau 0 saja.

Tabel 6.23: Cuplikan Klasifikasi K-Nearest Neighbour pada Line-Today

fitur	prob positif (cv)	prob negatif (cv)	prob positif (tf)	prob negatif (tf)	fakta
moga sktx bukn krna kalah yaaaaa	0	1	0	1	0
ulang	0	1	0	1	0
wkwkwk kasi	0	1	0	1	0
aduh 5 lembaga survei hasil ny jokowi menang	0.4	0.6	0	1	0
berani situ hukum mati koruptor usul koruptor kasih uang pensiun iya	0	1	0	1	0
ngaruh jamin negara hukum ad dibiarin tahan bs liar uda tahan penjara biar rasain gmna hidup bui berani buat berani tanggung jwb soob jaksa berani tolak jamin tahan prabowo	0.2	0.8	0.2	0.8	1
ciyan	0	1	0	1	0

Tabel 6.24: Cuplikan Klasifikasi K-Nearest Neighbour pada Twitter

fitur	prob positif (cv)	prob negatif (cv)	prob positif (tf)	prob negatif (tf)	fakta
pilih 01 11 11 11 dpd nggak cocok satu jokowi psi id pictwittercomywjuemozkn	1	0	0.8	0.2	1
calon presiden indonesia prabowo subianto harap menang milu april 2019 hadap saing keras indonesia masuk sulawesi utara kampung halaman ibu httpsnewnaratifcomjournalismpolitikidentitasprabowosharexuna9bf3dbf0c2efd551ca832b9b6800749c pilpres2019	0.2	0.8	0.8	0.2	1
rockygerung debat chally liat dungu rocky	0	1	0	1	0
jarum ya	0.2	0.8	0.2	0.8	0
jd yg jual takut nih	0.4	0.6	0.8	0.2	1
dukung yg byk ginihh kalah kecirangan lawann pictwittercomkc5mwtdmmo	0.2	0.8	0.4	0.6	0
hatihati milih presiden jgn kayak milih kucing karung mending pilih aja pilihorangbaik pilihygjelasislamnya pilihygbajuputih jokowi khmarufamin	0.8	0.2	0.6	0.4	1

Dapat dilihat pada tabel (6.23) dan tabel (6.24), KNN bekerja dengan banyak k dengan posisi acak. Untuk menemukan tetangga terdekat, dihitung jarak antara titik k dengan titik lainnya. Kemudian dipilh sebanyak k tetangga teratas yang jaraknya paling dekat dengan titik k. Karena dalam parameter penelitian ini k yang digunakan adalah 5, maka probabilitas prediksi yang dihasilkan berkelipatan 0.2.

Tabel 6.25: Cuplikan Klasifikasi Multinomial Naive Bayes pada Line-Today

fitur	prob positif (cv)	prob negatif (cv)	prob positif (tf)	prob negatif (tf)	fakta
moga sktx bukn krna kalah yaaaaa	0.01	0.99	0.16	0.84	0
ulang	0.26	0.74	0.23	0.77	0
wkwkwk kasi	0.02	0.98	0.06	0.94	0
aduh 5 lembaga survei hasil ny jokowi menang	0.36	0.64	0.33	0.67	0
berani situ hukum mati koruptor usul koruptor kasih uang pensiun iya	0.04	0.96	0.29	0.71	0
ngaruh jamin negara hukum ad dibiarin tahan bs liar uda tahan penjara biar rasain gmna hidup bui berani buat berani tanggung jwb soob jaksa berani tolak jamin tahan prabowo	0.99	0.01	0.41	0.59	1
ciyan	0.18	0.82	0.19	0.81	0

Tabel 6.26: Cuplikan Klasifikasi Multinomial Naive Bayes pada Twitter

fitur	prob positif (cv)	prob negatif (cv)	prob positif (tf)	prob negatif (tf)	fakta
pilih 01 11 11 11 dpd nggak cocok satu jokowi psi id pictwittercomywjuemozkn	0.99	0.01	0.74	0.26	1
calon presiden indonesia prabowo subianto harap menang milu april 2019 hadap saing keras indonesia masuk sulawesi utara kampung halaman ibu httpsnewnaratifcomjournalismpolitikidentitasprabowosharexuna9bf3dbf0c2efd551ca832b9b6800749c pilpres2019	1	0	0.82	0.18	1
rockygerung debat chally liat dungu rocky	0.01	0.99	0.21	0.79	0
jarum ya	0.22	0.78	0.31	0.69	0
jd yg jual takut nih	0.61	0.39	0.65	0.35	1
dukung yg byk ginihh kalah kecirangan lawann pictwittercomkc5mwtdmmo	0.02	0.98	0.31	0.69	0
hatihati milih presiden jgn kayak milih kucing karung mending pilih aja pilihorangbaik pilihygjelasislamnya pilihygbajuputih jokowi khmarufamin	1	0	0.83	0.17	1

Dapat dilihat pada tabel (6.25) dan tabel (6.26), MNB memperkirakan probabilitas bersyarat dari kata tertentu yang diberikan kelas sebagai frekuensi relatif dari *term* t dalam dokumen milik kelas c. Variasi memperhitungkan jumlah kemunculan *term* t dalam dokumen pelatihan dari kelas c. Karena perhitungan Multinomial hasil probabilitas prediksi bervariasi tidak terbatas pada nilai absolut 1 atau 0 saja.

Tabel 6.27: Cuplikan Klasifikasi Gaussian Naive Bayes pada Line-Today

fitur	prob positif (cv)	prob negatif (cv)	prob positif (tf)	prob negatif (tf)	fakta
moga sktx bukn krna kalah yaaaaa	0	1	0	1	0
ulang	1	0	1	0	0
wkwkwk kasi	1	0	1	0	0
aduh 5 lembaga survei hasil ny jokowi menang	1	0	1	0	0
berani situ hukum mati koruptor usul koruptor kasih uang pensiun iya	0	1	0	1	0
ngaruh jamin negara hukum ad dibiarin tahan bs liar uda tahan penjara biar rasain gmna hidup bui berani buat berani tanggung jwb soob jaksa berani tolak jamin tahan prabowo	1	0	1	0	1
ciyan	0	1	0	1	0

Tabel 6.28: Cuplikan Klasifikasi Gaussian Naive Bayes pada Twitter

fitur	prob positif (cv)	prob negatif (cv)	prob positif (tf)	prob negatif (tf)	fakta
pilih 01 11 11 11 dpd nggak cocok satu jokowi psi id pictwittercomywjuemozkn	1	0	1	0	1
calon presiden indonesia prabowo subianto harap menang milu april 2019 hadap saing keras indonesia masuk sulawesi utara kampung halaman ibu httpsnewnaratifcomjournalismpolitikidentitasprabowosharexuna9bf3dbf0c2efd551ca832b9b6800749c pilpres2019	1	0	1	0	1
rockygerung debat chally liat dungu rocky	0	1	0	1	0
jarum ya	0	1	0	1	0
jd yg jual takut nih	0	1	0	1	1
dukung yg byk ginihh kalah kecirangan lawann pictwittercomkc5mwtdmmo	0	1	0	1	0
hatihati milih presiden jgn kayak milih kucing karung mending pilih aja pilihorangbaik pilihygjelasislamnya pilihygbajuputih jokowi khmarufamin	1	0	1	0	1

Dapat dilihat pada tabel (6.27) dan tabel (6.28), GNB bekerja sama seperti MNB hanya berubah saat mengukur semua kemunculan dalam *term* t dalam dokumen. GNB mengukur kejadian hanya sekali saja dan menghasilkan probabilitas 1 atau 0 saja.

Tabel 6.29: Cuplikan Klasifikasi Logistic Regression pada Line-Today

fitur	prob positif (cv)	prob negatif (cv)	prob positif (tf)	prob negatif (tf)	fakta
moga sktx bukn krna kalah yaaaaa	0.05	0.95	0.15	0.85	0
ulang	0.11	0.89	0.14	0.86	0
wkwkwk kasi	0.02	0.98	0.04	0.96	0
aduh 5 lembaga survei hasil ny jokowi menang	0.44	0.56	0.55	0.45	0
berani situ hukum mati koruptor usul koruptor kasih uang pensiun iya	0.19	0.81	0.3	0.7	0
ngaruh jamin negara hukum ad dibiarin tahan bs liar uda tahan penjara biar rasain gmna hidup bui berani buat berani tanggung jwb soob jaksa berani tolak jamin tahan prabowo	0.88	0.12	0.4	0.6	1
ciyan	0.16	0.84	0.19	0.81	0

Tabel 6.30: Cuplikan Klasifikasi Logistic Regression pada Twitter

fitur	prob positif (cv)	prob negatif (cv)	prob positif (tf)	prob negatif (tf)	fakta
pilih 01 11 11 11 dpd nggak cocok satu jokowi psi id pictwittercomywjuemozkn	0.9	0.1	0.68	0.32	1
calon presiden indonesia prabowo subianto harap menang milu april 2019 hadap saing keras indonesia masuk sulawesi utara kampung halaman ibu httpsnewnaratifcomjournalismpolitikidentitasprabowosharexuna9bf3dbf0c2efd551ca832b9b6800749c pilpres2019	0.64	0.36	0.65	0.35	1
rockygerung debat chally liat dungu rocky	0.13	0.87	0.2	0.8	0
jarum ya	0.24	0.76	0.23	0.77	0
jd yg jual takut nih	0.53	0.47	0.61	0.39	1
dukung yg byk ginihh kalah kecirangan lawann pictwittercomkc5mwtdmmo	0.23	0.77	0.29	0.71	0
hatihati milih presiden jgn kayak milih kucing karung mending pilih aja pilihorangbaik pilihygjelasislamnya pilihygbajuputih jokowi khmarufamin	0.96	0.04	0.86	0.14	1

Dapat dilihat pada tabel (6.29) dan tabel (6.30), LR menggunakan Logit model. Logit model adalah model matematika yang digunakan dalam statistik untuk memperkirakan probabilitas suatu peristiwa yang terjadi telah diberikan beberapa data sebelumnya. Probabilitas yang didapatkan adalah jarak antara fitur dengan garis logistik.

Tabel 6.31: Cuplikan Klasifikasi Neural Network pada Line-Today

fitur	prob positif (cv)	prob negatif (cv)	prob positif (tf)	prob negatif (tf)	fakta
moga sktx bukn krna kalah yaaaaa	0	1	0	1	0
ulang	0	1	0	1	0
wkwkwk kasi	0	1	0	1	0
aduh 5 lembaga survei hasil ny jokowi menang	0.01	0.99	0.01	0.99	0
berani situ hukum mati koruptor usul koruptor kasih uang pensiun iya	0	1	0	1	0
ngaruh jamin negara hukum ad dibiarin tahan bs liar uda tahan penjara biar rasain gmna hidup bui berani buat berani tanggung jwb soob jaksa berani tolak jamin tahan prabowo	1	0	1	0	1
ciyan	0	1	0	1	0

Tabel 6.32: Cuplikan Klasifikasi Neural Network pada Twitter

fitur	prob positif (cv)	prob negatif (cv)	prob positif (tf)	prob negatif (tf)	fakta
pilih 01 11 11 11 dpd nggak cocok satu jokowi psi id pictwittercomywjuemozkn	1	0	1	0	1
calon presiden indonesia prabowo subianto harap menang milu april 2019 hadap saing keras indonesia masuk sulawesi utara kampung halaman ibu httpsnewnaratifcomjournalismpolitikidentitasprabowosharexuna9bf3dbf0c2efd551ca832b9b6800749c pilpres2019	1	0	1	0	1
rockygerung debat chally liat dungu rocky	0	1	0	1	0
jarum ya	0	1	0	1	0
jd yg jual takut nih	1	0	1	0	1
dukung yg byk ginihh kalah kecirangan lawann pictwittercomkc5mwtdmmo	0	1	0	1	0
hatihati milih presiden jgn kayak milih kucing karung mending pilih aja pilihorangbaik pilihygjelasislamnya pilihygbajuputih jokowi khmarufamin	1	0	1	0	1

Dapat dilihat pada tabel (6.31) dan tabel (6.32). NN menggunakan metode Multilayer Perceptron (MLP). MLP adalah bagian dari Neural Network yang terdiri dari setidaknya 3 node yang merupakan input layer, output layer dan hidden layer. Masing-masing layer MLP, kecuali *input layer* adalah neuron yang menggunakan fungsi aktivasi non-linear. Node MLP disusun berlapis-lapis dalam hidden layer. Dalam probabilitas prediksi di MLP, hasil yang didapatkan sangat mendekati 1 atau 0 dikarenakan banyaknya *layer* dan 200 iterasi (nilai 1 dan 0 bisa saja terjadi karena pembulatan 2 desimal).

Tabel 6.33: Cuplikan Klasifikasi ADABoost pada Line-Today

fitur	prob positif (cv)	prob negatif (cv)	prob positif (tf)	prob negatif (tf)	fakta
moga sktx bukn krna kalah yaaaaa	0.49	0.51	0.49	0.51	0
ulang	0.5	0.5	0.5	0.5	0
wkwkwk kasi	0.49	0.51	0.49	0.51	0
aduh 5 lembaga survei hasil ny jokowi menang	0.5	0.5	0.5	0.5	0
berani situ hukum mati koruptor usul koruptor kasih uang pensiun iya	0.5	0.5	0.5	0.5	0
ngaruh jamin negara hukum ad dibiarin tahan bs liar uda tahan penjara biar rasain gmna hidup bui berani buat berani tanggung jwb soob jaksa berani tolak jamin tahan prabowo	0.5	0.5	0.5	0.5	1
ciyan	0.5	0.5	0.5	0.5	0

Tabel 6.34: Cuplikan Klasifikasi ADABoost pada Twitter

fitur	prob positif (cv)	prob negatif (cv)	prob positif (tf)	prob negatif (tf)	fakta
pilih 01 11 11 11 dpd nggak cocok satu jokowi psi id pictwittercomywjuemozkn	0.5	0.5	0.51	0.49	1
calon presiden indonesia prabowo subianto harap menang milu april 2019 hadap saing keras indonesia masuk sulawesi utara kampung halaman ibu httpsnewnaratifcomjournalismpolitikidentitasprabowosharexuna9bf3dbf0c2efd551ca832b9b6800749c pilpres2019	0.5	0.5	0.5	0.5	1
rockygerung debat chally liat dungu rocky	0.5	0.5	0.5	0.5	0
jarum ya	0.5	0.5	0.49	0.51	0
jd yg jual takut nih	0.5	0.5	0.5	0.5	1
dukung yg byk ginihh kalah kecirangan lawann pictwittercomkc5mwtdmmo	0.5	0.5	0.5	0.5	0
hatihati milih presiden jgn kayak milih kucing karung mending pilih aja pilihorangbaik pilihygjelasislamnya pilihygbajuputih jokowi khmarufamin	0.51	0.49	0.51	0.49	1

Dapat dilihat pada tabel (6.33) dan tabel (6.34). Sama seperti RF, Adaboost juga termasuk bagian pengklasifikasi *ensemble* dengan melakukan pelatihan dengan beberapa *tree*. Pada penelitian ini, parameter yang digunakan adalah sebanyak 100 *tree*.

Tabel 6.35: Cuplikan Klasifikasi Gradient Boosting pada Line-Today

fitur	prob positif (cv)	prob negatif (cv)	prob positif (tf)	prob negatif (tf)	fakta
moga sktx bukn krna kalah yaaaaa	0.23	0.77	0.18	0.82	0
ulang	0.27	0.73	0.27	0.73	0
wkwkwk kasi	0.19	0.81	0.22	0.78	0
aduh 5 lembaga survei hasil ny jokowi menang	0.67	0.33	0.56	0.44	0
berani situ hukum mati koruptor usul koruptor kasih uang pensiun iya	0.4	0.6	0.27	0.73	0
ngaruh jamin negara hukum ad dibiarin tahan bs liar uda tahan penjara biar rasain gmna hidup bui berani buat berani tanggung jwb soob jaksa berani tolak jamin tahan prabowo	0.29	0.71	0.27	0.73	1
ciyan	0.27	0.73	0.27	0.73	0

Tabel 6.36: Cuplikan Klasifikasi Gradient Boostong pada Twitter

fitur	prob positif (cv)	prob negatif (cv)	prob positif (tf)	prob negatif (tf)	fakta
pilih 01 11 11 11 dpd nggak cocok satu jokowi psi id pictwittercomywjuemozkn	0.69	0.31	0.68	0.32	1
calon presiden indonesia prabowo subianto harap menang milu april 2019 hadap saing keras indonesia masuk sulawesi utara kampung halaman ibu httpsnewnaratifcomjournalismpolitikidentitasprabowosharexuna9bf3dbf0c2efd551ca832b9b6800749c pilpres2019	0.8	0.2	0.65	0.35	1
rockygerung debat chally liat dungu rocky	0.33	0.67	0.33	0.67	0
jarum ya	0.33	0.67	0.27	0.73	0
jd yg jual takut nih	0.51	0.49	0.52	0.48	1
dukung yg byk ginihh kalah kecirangan lawann pictwittercomkc5mwtdmmo	0.33	0.67	0.31	0.69	0
hatihati milih presiden jgn kayak milih kucing karung mending pilih aja pilihorangbaik pilihygjelasislamnya pilihygbajuputih jokowi khmarufamin	0.9	0.1	0.87	0.13	1

Dapat dilihat pada tabel (6.35) dan tabel (6.36). Sama seperti RF dan Adaboost, Gradietn Boosting juga termasuk bagian pengklasifikasi *ensemble* dengan melakukan pelatihan dengan beberapa *tree*. Pada penelitian ini, parameter yang digunakan juga sama dengan RF dan Adaboost yaitu sebanyak 100 *tree*.

Dari tabel (6.15) sampai tabel (6.36) menampilkan salah satu cuplikan fitur, hasil probabilitas bahwa prediksi melabeli kelas "1" atau "0" pada kedua fitur yang di ekstraksi menggunakan Count Vectorizer dan TFIDF, dan kelas pelabelan yang sebenarnya. Pada Random Forest terdapat banyak parameter *tree* (dalam penelitian ini parameter yang digunakan adalah 100 *tree).* Berbeda dengan Decision Tree dan Extra Tree yang hanya terdapat satu *tree* sehinggan probabilitas prediksinya bernilai absolut. Support Vector Machine mendapatkan probabilitas dengan perkiraan posterior yang dihasilkan secara efektif merupakan versi yang ditingkatkan dari skor pengklasifikasi asli melalui transformasi logistik. Pada Gaussian Naive Bayes mendapatkan probabilitas hanya "1" atau "0" karena Gaussian Naive Bayes adalah model non-linear dan tidak mendukung *sparse* matriks berkebalikan dengan Multinomial Naive Bayes yang mendukung *sparse* matriks. Lalu untuk K-Nearest Neighbour mendapatkan probabilitas hanya berkelipatan 5 karena parameter k yang digunakan pada K-Nearest Neighbour adalah 5. Untuk Logistic Regression mendapatkan probabilitas dengan mendapatkan jarak dari garis logistik yang dibentuk pengklasifikasi. Kemudian pada Neural Network yang menggunakan Multilayer Perceptron mendapatkan probabilitas yang sangat mendekati "1" atau "0" dari hasil pelatihan dengan parameter banyak *hidden layer* sebanyak 100 dan *batch size* sebanyak 200. Kemudian Ada Boost mendapatkan probabilitas sangat mendekati 0.5. Ada Boost adalah algoritme berulang dan T biasanya menunjukkan jumlah iterasi atau "putaran" Sehingga probabilitas Ada Bo-

ost hanya berkutat di kisaran 0.5. Kemudian pengklasfikasi Gradient Boosting sama dengan pengklasifikasi *ensemble* lainnya dengan menggunakan beberapa *tree*.

6.7 Validasi

Tahap terakhir adalah melakukan *cross-validation* dengan *fold* sebanyak 10 pada semua jenis pengklasifikasi seperti Random Forest, Support Vector Machine, Decision Tree, Extra tree, K-Nearest Neighbour, Multinomial Naive Bayes, Gaussian Naive Bayes, Logistic Regression, Neural Network dengan Multi-Layer Perceptron, Ada Boost, dan Gradient Boosting. Validasi akan dibandingkan dengan kedua data Line-Today dan Twitter beserta semua jenis ekstrasi fitur seperti Count Vectrizer dan TFIDF dan semua jenis seleksi fitur seperti Chi Square, ANOVA, dan Mutual Information. Berikut pada tabel (6.37) sampai tabel (6.52) hasil *cross validation* ekstraksi fitur Count Vectorizer pada Line-Today tanpa seleksi fitur.

Tabel 6.37: Hasil Cross Validation ekstraksi fitur Count Vectorizer pada Line-Today tanpa seleksi fitur

	accuracy(%)	precision(%)	recall(%)	f1(%)	time(s)
RF	84.84	82.57	76.68	79.29	395.63
SVM	84	82	74.38	77.77	701.04
DT	84.53	80.23	79.46	79.68	300.58
ETC	85.13	81.93	78.7	80.08	847.27
KNN	70.69	76.27	33.71	46.52	272.4
MNB	76.85	69.56	73.03	70.81	3.89
GNB	59.39	48.5	81	60.56	9.36
LR	82.07	82.05	67.71	73.88	346.6
NN	84.09	81.78	75.62	78.34	756.65
ABC	76.11	74.25	57.31	64.47	699.33
GBC	73.51	75.6	45.51	56.47	1424.24

Hasil *cross-validation* pada ekstraksi fitur Count Vectorizer tanpa seleksi fitur pada data Line-Today mendapatkan rata-rata akurasi 78.29 %, presisi 75.89 %,

recall 67.56 %, *f1 score* 69.81 %, dan waktu *running* 523.36 detik. Random Forest mendapatkan nilai tertinggi di akurasi dan presisi sedangkan Gaussian Naive Bayes mendapatkan nilai terendah pada akurasi dan presisi. Namun Gaussian Naive Bayes mendapatkan nilai tertinggi pada *recall* sedangkan K-Nearest Neighbour mendapatkan nilai *recall* terendah. Extra Tree Classifier mendapatkan nilai tertinggi pada *f1 score* sedangkan Gradient Boosting Classifie mendapatkan nilai terendah pada *f1 score*. Multinomial Naive Bayes mendapatkan waktu tercepat pada waktu *running* sedangkan Gradient Boosting Classifier mendapatkan waktu terlama pada waktu *running*. Sementara itu berikut pada tabel (6.16) hasil *cross validation* ekstraksi fitur TFIDF pada Line-Today tanpa seleksi fitur.

Tabel 6.38: Hasil Cross Validation ekstraksi fitur TFIDF pada Line-Today tanpa seleksi fitur

	accuracy(%)	precision(%)	recall(%)	f1(%)	time(s)
RF	84.11	84.17	71.5	76.84	462.12
SVM	82.82	80.82	71.69	75.76	1018.12
DT	83.13	80.14	74.38	76.91	372.56
ETC	84.34	84.11	72.6	77.6	1095.4
KNN	64.08	73.56	8.63	15.18	287.42
MNB	78.32	78.88	59.65	67.58	2.58
GNB	60.98	49.7	78.36	60.74	8.95
LR	79.25	83.2	56.72	66.9	79.68
NN	82.7	79.15	74.62	76.66	737.13
ABC	76.08	72.49	59.55	65.2	731.83
GBC	74.88	76.79	48.62	58.97	1498.2

Hasil *cross-validation* pada ekstraksi fitur TFIDF tanpa seleksi fitur pada data Line-Today mendapatkan rata-rata akurasi 77.34 %, presisi 76.64 %, *recall* 61.48 %, *f1 score* 65.30 %, dan waktu *running* 572.18 detik. Extra Tree Classifier mendapatkan nilai tertinggi pada akurasi sedangkan Gaussian Naive Bayes mendapatkan nilai terendah pada pada akurasi. Random Forest mendapatkan nilai tertinggi pada presisi sedangkan Gaussian Naive Bayes mendapatkan nilai terendah pada presisi. Gaussian

Naive Bayes mendapatkan nilai tertinggi pada *recall* sedangkan K-Nearest Neighbour mendapatkan nilai terendah pada *recall*. Extra Tree Classifier mendapatkan nilai tertinggi pada *f1 score* sedangkan Gradient Boosting Classifier mendapatkan nilai terendah pada *f1 score*. Multinomial Naive Bayes mendapatkan waktu *running* tercepat sedangkan Gradient Boosting Classifier mendapatkan waktu *running* terlama. Sementara itu berikut pada tabel (6.16) hasil *cross validation* ekstraksi fitur TFIDF pada Line-Today tanpa seleksi fitur. Sementara itu berikut pada tabel (6.17) hasil *cross validation* ekstraksi fitur Count Vectorizer pada Twitter tanpa seleksi fitur.

Tabel 6.39: Hasil Cross Validation ekstraksi fitur Count Vectorizer pada Twitter tanpa seleksi fitur

	accuracy(%)	precision(%)	recall(%)	f1(%)	time(s)
RF	80.64	80.78	81.41	81.05	637.42
SVM	79.65	82.33	76.5	79.26	3029.13
DT	80.48	81.22	80.28	80.71	621.88
ETC	81.86	80.32	85.29	82.71	1463.9
KNN	71.78	80.35	59.01	68.01	585.05
MNB	73.94	74.22	74.82	74.49	7.96
GNB	59.14	68.16	36.98	47.93	18
LR	79.21	83.26	74.1	78.37	688.25
NN	79.14	81.2	76.91	78.95	1782.2
ABC	79.58	83.84	74.24	78.69	1614.83
GBC	76.63	82.7	68.48	74.86	3031.59

Hasil *cross-validation* pada ekstraksi fitur Count Vectorizer tanpa seleksi fitur pada data Twitter mendapatkan rata-rata akurasi 76.55 %, presisi 79.85 %, *recall* 71.64 %, *f1 score* 75 %, dan waktu *running* 1225.47 detik. Extra Tree Classifier mendapatkan nilai tertinggi pada akurasi sedangkan Gaussian Naive Bayes mendapatkan nilai terendah pada pada akurasi. ADA Boost Classifier mendapatkan nilai tertinggi pada presisi sedangkan Gaussian Naive Bayes mendapatkan nilai terendah pada presisi. Extra Tree Classifier mendapatkan nilai tertinggi pada *recall* sedangkan Gaussian Naive Bayes mendapatkan nilai terendah pada *recall*. Extra Tree Classifier

mendapatkan nilai tertinggi pada *f1 score* sedangkan Gaussian Naive Bayes mendapatkan nilai terendah pada *f1 score*. Multinomial Naive Bayes mendapatkan waktu *running* tercepat sedangkan Gradient Boosting Classifier mendapatkan waktu *running* terlama. Sementara itu berikut pada tabel (6.18) hasil *cross validation* ekstraksi fitur TFIDF pada Twitter tanpa seleksi fitur.

Tabel 6.40: Hasil Cross Validation ekstraksi fitur TFIDF pada Twitter tanpa seleksi fitur

	accuracy(%)	precision(%)	recall(%)	f1(%)	time(s)
RF	80.45	82.59	78.12	80.25	674.23
SVM	79.61	83.58	74.72	78.83	2713.14
DT	79.18	79.43	79.8	79.58	715.1
ETC	81.46	81.21	82.78	81.96	1781.99
KNN	67.85	73.43	57.91	64.66	583.2
MNB	73.97	71.5	81.31	76.07	4.81
GNB	59.33	66.48	40.51	50.33	17.54
LR	78.48	83.81	71.63	77.19	145.65
NN	75.96	78.37	72.97	75.54	1743.69
ABC	79.58	82.92	75.51	78.99	1620.43
GBC	76.75	83.06	68.34	74.95	3052.91

Hasil *cross-validation* pada ekstraksi fitur TFIDF tanpa seleksi fitur pada data Twitter mendapatkan rata-rata akurasi 75.69 %, presisi 78.76 %, *recall* 71.24 %, *f1 score* 74.4 %, dan waktu *running* 1186.61 detik. Extra Tree Classifier mendapatkan nilai tertinggi pada akurasi sedangkan Gaussian Naive Bayes mendapatkan nilai terendah pada pada akurasi. Logistic Regression mendapatkan nilai tertinggi pada presisi sedangkan Gaussian Naive Bayes mendapatkan nilai terendah pada presisi. Extra Tree Classifier mendapatkan nilai tertinggi pada *recall* sedangkan Gaussian Naive Bayes mendapatkan nilai terendah pada *recall*. Extra Tree Classifier mendapatkan nilai tertinggi pada *f1 score* sedangkan Gaussian Naive Bayes mendapatkan nilai terendah pada *f1 score*. Multinomial Naive Bayes mendapatkan waktu *running* tercepat sedangkan Gradient Boosting Classifier mendapatkan waktu *running* terla-

ma. Sementara itu berikut pada tabel (6.19) hasil *cross validation* ekstraksi fitur Count Vectorizer pada Line-Today dengan seleksi fitur Chi Square.

Tabel 6.41: Hasil Cross Validation ekstraksi fitur Count Vectorizer pada Line-Today dengan seleksi fitur Chi Square

	accuracy(%)	precision(%)	recall(%)	f1(%)	time(s)
RF	84.16	81.86	75.05	78.14	212.82
SVM	83.78	83.67	71.07	76.6	367.97
DT	84.69	81.01	78.31	79.45	122.84
ETC	85.13	81.6	78.99	80.14	437.7
KNN	71.09	79.55	32.36	45.5	136.22
MNB	83.57	80.41	75.82	77.85	7.17
GNB	78.4	88.02	50.36	63.49	9.27
LR	81.76	83.38	64.93	72.8	179.83
NN	85	83.14	76.25	79.33	1050.09
ABC	75.67	73.82	56.35	63.69	306.52
GBC	73.53	76.02	45.17	56.23	652.9

Hasil *cross-validation* pada ekstraksi fitur Count Vectorizer dengan seleksi fitur Chi Square pada data Line-Today mendapatkan rata-rata akurasi 80.62 %, presisi 81.13 %, *recall* 64.06 %, *f1 score* 70.29 %, dan waktu *running* 316.67 detik. Extra Tree Classifier mendapatkan nilai tertinggi pada akurasi sedangkan K-Nearest Neighbour mendapatkan nilai terendah pada pada akurasi. Gaussian Naive Bayes mendapatkan nilai tertinggi pada presisi sedangkan ADA Boost Classifier mendapatkan nilai terendah pada presisi. Extra Tree Classifier mendapatkan nilai tertinggi pada *recall* sedangkan Gradient Boosting Classifier mendapatkan nilai terendah pada *recall*. Extra Tree Classifier mendapatkan nilai tertinggi pada *f1 score* sedangkan K-Nearest Neighbour mendapatkan nilai terendah pada *f1 score*. Multinomial Naive Bayes mendapatkan waktu *running* tercepat sedangkan Multilayer Perceptron mendapatkan waktu *running* terlama. Sementara itu berikut pada tabel (6.20) hasil *cross validation* ekstraksi fitur TFIDF pada Line-Today dengan seleksi fitur Chi Square.

Tabel 6.42: Hasil Cross Validation ekstraksi fitur TFIDF pada Line-Today dengan seleksi fitur Chi Square

	accuracy(%)	precision(%)	recall(%)	f1(%)	time(s)
RF	84.01	85.4	69.54	76.14	242.37
SVM	84.12	85.92	69.68	76.53	504.64
DT	84.09	81.69	75.19	78.01	154.48
ETC	84.6	86.88	70.02	77.11	557.65
KNN	67.74	88.1	17.54	28.51	135.3
MNB	82.07	87.73	61.47	71.93	6.67
GNB	79.11	88.02	52.66	65.42	8.92
LR	80.13	84.71	58.07	68.35	42.62
NN	85.79	88.07	71.88	78.87	592.08
ABC	76.51	73.66	59.36	65.48	307.5
GBC	75.04	77.14	48.96	59.36	653.58

Hasil *cross-validation* pada ekstraksi fitur TFIDF dengan seleksi fitur Chi Square pada data Line-Today mendapatkan rata-rata akurasi 80.29 %, presisi 84.30 %, *recall* 59.49 %, *f1 score* 67.79 %, dan waktu *running* 291.44 detik. Multilayer Perceptron mendapatkan nilai tertinggi pada akurasi sedangkan K-Nearest Neighbour mendapatkan nilai terendah pada pada akurasi. K-Nearest Neighbour mendapatkan nilai tertinggi pada presisi sedangkan ADA Boost Classifier mendapatkan nilai terendah pada presisi. Decision Tree mendapatkan nilai tertinggi pada *recall* sedangkan K-Nearest Neighbour mendapatkan nilai terendah pada *recall*. Multilayer Perceptron mendapatkan nilai tertinggi pada *f1 score* sedangkan K-Nearest Neighbour mendapatkan nilai terendah pada *f1 score*. Multinomial Naive Bayes mendapatkan waktu *running* tercepat sedangkan Gradient Boosting Classifier mendapatkan waktu *running* terlama. Sementara itu berikut pada tabel (6.21) hasil *cross validation* ekstraksi fitur Count Vectorizer pada Twitter dengan seleksi fitur Chi Square.

Tabel 6.43: Hasil Cross Validation ekstraksi fitur Count Vectorizer pada Twitter dengan seleksi fitur Chi Square

	accuracy(%)	precision(%)	recall(%)	f1(%)	time(s)
RF	81.06	77.88	87.75	82.5	351.96
SVM	83.54	84	83.67	83.8	1145.75
DT	81.03	79.74	84.15	81.86	341.59
ETC	81.37	77.67	89.06	82.96	839.01
KNN	74.25	76.48	71.46	73.85	304.48
MNB	80.47	75.52	91.25	82.62	16.47
GNB	65.04	59.62	97.08	73.87	21.42
LR	81.25	84.68	77.19	80.7	373.18
NN	82.14	79.28	87.93	83.36	2686.79
ABC	79.93	83.85	75.1	79.16	699.99
GBC	76.45	82.56	68.2	74.65	1407.01

Hasil *cross-validation* pada ekstraksi fitur Count Vectorizer dengan seleksi fitur Chi Square pada data Twitter mendapatkan rata-rata akurasi 78.78 %, presisi 78.30 %, *recall* 82.99 %, *f1 score* 79.94 %, dan waktu *running* 744.33 detik. Support Vector Machine mendapatkan nilai tertinggi pada akurasi sedangkan Gaussian Naive Bayes mendapatkan nilai terendah pada pada akurasi. Logistic Regression mendapatkan nilai tertinggi pada presisi sedangkan Gaussian Naive Bayes mendapatkan nilai terendah pada presisi. Gaussian Naive Bayes mendapatkan nilai tertinggi pada *recall* sedangkan Gradient Boosting Classifier mendapatkan nilai terendah pada *recall*. Support Vector Machine mendapatkan nilai tertinggi pada *f1 score* sedangkan K-Nearest Neighbour mendapatkan nilai terendah pada *f1 score*. Multinomial Naive Bayes mendapatkan waktu *running* tercepat sedangkan Multilayer Perceptron mendapatkan waktu *running* terlama. Sementara itu berikut pada tabel (6.22) hasil *cross validation* ekstraksi fitur TFIDF pada Twitter dengan seleksi fitur Chi Square.

Tabel 6.44: Hasil Cross Validation ekstraksi fitur TFIDF pada Twitter dengan seleksi fitur Chi Square

	accuracy(%)	precision(%)	recall(%)	f1(%)	time(s)
RF	81.06	82	80.55	81.22	381.43
SVM	83.14	87.34	78.22	82.5	1342.18
DT	80	80.78	79.7	80.2	391.6
ETC	82.28	81.42	84.53	82.91	1051.06
KNN	65.47	83.95	39.8	53.94	293.09
MNB	79.89	76.33	87.75	81.63	13.93
GNB	70.87	64.54	94.99	76.85	19.56
LR	80.12	85.53	73.42	78.96	80.96
NN	80.28	81.53	79.21	80.33	2297.15
ABC	79.61	83	75.47	79.01	702.68
GBC	76.82	83.02	68.54	75.05	1409.67

Hasil *cross-validation* pada ekstraksi fitur TFIDF dengan seleksi fitur Chi Square pada data Twitter mendapatkan rata-rata akurasi 78.14 %, presisi 80.86 %, *recall* 76.56 %, *f1 score* 77.51 %, dan waktu *running* 725.76 detik. Support Vector Machine mendapatkan nilai tertinggi pada akurasi sedangkan Gaussian Naive Bayes mendapatkan nilai terendah pada pada akurasi. Support Vector Machine mendapatkan nilai tertinggi pada presisi sedangkan Gaussian Naive Bayes mendapatkan nilai terendah pada presisi. Gaussian Naive Bayes mendapatkan nilai tertinggi pada *recall* sedangkan K-Nearest Neighbour mendapatkan nilai terendah pada *recall*. Extra Tree Classifier mendapatkan nilai tertinggi pada *f1 score* sedangkan K-Nearest Neighbour mendapatkan nilai terendah pada *f1 score*. Multinomial Naive Bayes mendapatkan waktu *running* tercepat sedangkan Multilayer Perceptron mendapatkan waktu *running* terlama. Sementara itu berikut pada tabel (6.23) hasil *cross validation* ekstraksi fitur Count Vectorizer pada Line-Today dengan seleksi fitur ANOVA.

Tabel 6.45: Hasil Cross Validation ekstraksi fitur Count Vectorizer pada Line-Today dengan seleksi fitur ANOVA

	accuracy(%)	precision(%)	recall(%)	f1(%)	time(s)
RF	84.03	81.73	74.91	77.96	213.14
SVM	83.96	84.24	71.07	76.79	369.61
DT	84.24	80.36	77.69	78.84	122.82
ETC	84.84	81.54	78.13	79.63	432.92
KNN	70.71	78.89	31.36	44.39	135.96
MNB	83.67	80.16	76.58	78.12	7.06
GNB	78.73	88.16	51.32	64.31	9.45
LR	81.95	83.58	65.26	73.05	180.22
NN	85.09	83.1	76.39	79.38	991.74
ABC	75.47	73.37	56.25	63.44	306.52
GBC	73.82	76.3	45.94	56.95	653.6

Hasil *cross-validation* pada ekstraksi fitur Count Vectorizer dengan seleksi fitur ANOVA pada data Line-Today mendapatkan rata-rata akurasi 80.59 %, presisi 81.04 %, *recall* 64.08 %, *f1 score* 70.26 %, dan waktu *running* 311.19 detik. Multilayer Perceptron mendapatkan nilai tertinggi pada akurasi sedangkan K-Nearest Neighbour mendapatkan nilai terendah pada pada akurasi. Gaussian Naive Bayes mendapatkan nilai tertinggi pada presisi sedangkan ADA Boost Classifier mendapatkan nilai terendah pada presisi. Extra Tree Classifier mendapatkan nilai tertinggi pada *recall* sedangkan K-Nearest Neighbour mendapatkan nilai terendah pada *recall*. Extra Tree Classifier mendapatkan nilai tertinggi pada *f1 score* sedangkan K-Nearest Neighbour mendapatkan nilai terendah pada *f1 score*. Multinomial Naive Bayes mendapatkan waktu *running* tercepat sedangkan Multilayer Perceptron mendapatkan waktu *running* terlama. Sementara itu berikut pada tabel (6.24) hasil *cross validation* ekstraksi fitur TFIDF pada Line-Today dengan seleksi fitur ANOVA.

Tabel 6.46: Hasil Cross Validation ekstraksi fitur TFIDF pada Line-Today dengan seleksi fitur ANOVA

	accuracy(%)	precision(%)	recall(%)	f1(%)	time(s)
RF	84.53	85.09	71.7	77.46	239.8
SVM	83.96	86.47	68.48	76.01	502.37
DT	84.23	81.44	75.86	78.31	145.62
ETC	85.1	84.31	74.58	78.91	530.92
KNN	69.51	80.74	25.41	37.86	133.09
MNB	83.34	88.85	64.3	74.26	6.71
GNB	80.94	88.75	57.5	69.31	8.96
LR	80.24	84.91	58.31	68.69	42.04
NN	85.33	84.55	75.72	79.55	774.4
ABC	76.56	73.79	59.32	65.5	307.14
GBC	75.32	77.71	49.25	59.71	653.64

Hasil *cross-validation* pada ekstraksi fitur TFIDF dengan seleksi fitur ANOVA pada data Line-Today mendapatkan rata-rata akurasi 80.82 %, presisi 83.33 %, *recall* 61.86 %, *f1 score* 69.60 %, dan waktu *running* 304.06 detik. Multilayer Perceptron mendapatkan nilai tertinggi pada akurasi sedangkan K-Nearest Neighbour mendapatkan nilai terendah pada pada akurasi. Multinomial Naive Bayes mendapatkan nilai tertinggi pada presisi sedangkan ADA Boost Classifier mendapatkan nilai terendah pada presisi. Decision Tree mendapatkan nilai tertinggi pada *recall* sedangkan K-Nearest Neighbour mendapatkan nilai terendah pada *recall*. Multilayer Perceptron mendapatkan nilai tertinggi pada *f1 score* sedangkan K-Nearest Neighbour mendapatkan nilai terendah pada *f1 score*. Multinomial Naive Bayes mendapatkan waktu *running* tercepat sedangkan Multilayer Perceptron mendapatkan waktu *running* terlama. Sementara itu berikut pada tabel (6.25) hasil *cross validation* ekstraksi fitur Count Vectorizer pada Tiwtter dengan seleksi fitur ANOVA.

Tabel 6.47: Hasil Cross Validation ekstraksi fitur Count Vectorizer pada Twitter dengan seleksi fitur ANOVA

	accuracy(%)	precision(%)	recall(%)	f1(%)	time(s)
RF	81.5	78.43	87.89	82.87	356.97
SVM	83.61	83.87	84.05	83.93	1150.46
DT	80.73	79.43	83.91	81.58	341.4
ETC	81.36	77.42	89.54	83.02	841.26
KNN	74.83	76.84	72.46	74.54	294
MNB	80.45	75.18	92.04	82.74	15.35
GNB	65.14	59.67	97.29	73.97	20.35
LR	81.43	84.83	77.4	80.89	356.12
NN	82.04	78.83	88.51	83.37	2365.18
ABC	79.91	83.84	75.06	79.14	697.89
GBC	76.84	82.8	68.85	75.14	1402.76

Hasil *cross-validation* pada ekstraksi fitur Count Vectorizer dengan seleksi fitur ANOVA pada data Twitter mendapatkan rata-rata akurasi 78.89 %, presisi 78.29 %, *recall* 83.36 %, *f1 score* 80.11 %, dan waktu *running* 712.89 detik. Support Vector Machine mendapatkan nilai tertinggi pada akurasi sedangkan Gaussian Naive Bayes mendapatkan nilai terendah pada pada akurasi. Support Vector Machine mendapatkan nilai tertinggi pada presisi sedangkan Gaussian Naive Bayes mendapatkan nilai terendah pada presisi. Gaussian Naive Bayes mendapatkan nilai tertinggi pada *recall* sedangkan Gradient Boosting Classifier mendapatkan nilai terendah pada *recall*. Support Vector Machine mendapatkan nilai tertinggi pada *f1 score* sedangkan K-Nearest Neighbour mendapatkan nilai terendah pada *f1 score*. Multinomial Naive Bayes mendapatkan waktu *running* tercepat sedangkan Multilayer Perceptron mendapatkan waktu *running* terlama. Sementara itu berikut pada tabel (6.26) hasil *cross validation* ekstraksi fitur TFIDF pada Twitter dengan seleksi fitur ANOVA.

Tabel 6.48: Hasil Cross Validation ekstraksi fitur TFIDF pada Twitter dengan seleksi fitur ANOVA

	accuracy(%)	precision(%)	recall(%)	f1(%)	time(s)
RF	80.96	78.21	86.83	82.26	375.35
SVM	83.47	87.05	79.35	82.99	1322.09
DT	79.72	79.43	81.27	80.3	396.64
ETC	80.12	75.29	90.74	82.28	1035.77
KNN	68.08	64.74	81.89	72.3	285.97
MNB	75.88	69.44	94.03	79.87	13.88
GNB	65.53	59.94	97.32	74.18	19.76
LR	80.76	85.24	75.3	79.92	80.72
NN	80.87	76.98	89.06	82.57	2298.76
ABC	79.63	82.97	75.58	79.05	700.88
GBC	76.71	82.94	68.41	74.93	1409.16

Hasil *cross-validation* pada ekstraksi fitur TFIDF dengan seleksi fitur ANOVA pada data Twitter mendapatkan rata-rata akurasi 77.43 %, presisi 76.57 %, *recall* 83.62 %, *f1 score* 79.15 %, dan waktu *running* 721.73 detik. Support Vector Machine mendapatkan nilai tertinggi pada akurasi sedangkan Gaussian Naive Bayes mendapatkan nilai terendah pada pada akurasi. Support Vector Machine mendapatkan nilai tertinggi pada presisi sedangkan Gaussian Naive Bayes mendapatkan nilai terendah pada presisi. Gaussian Naive Bayes mendapatkan nilai tertinggi pada *recall* sedangkan Gradient Boosting Classifier mendapatkan nilai terendah pada *recall*. Support Vector Machine mendapatkan nilai tertinggi pada *f1 score* sedangkan K-Nearest Neighbour mendapatkan nilai terendah pada *f1 score*. Multinomial Naive Bayes mendapatkan waktu *running* tercepat sedangkan Multilayer Perceptron mendapatkan waktu *running* terlama. Sementara itu berikut pada tabel (6.27) hasil *cross validation* ekstraksi fitur Count Vectorizer pada Line-Today dengan seleksi fitur Mutual Information.

Tabel 6.49: Hasil Cross Validation ekstraksi fitur Count Vectorizer pada Line-Today dengan seleksi fitur Mutual Information

	accuracy(%)	precision(%)	recall(%)	f1(%)	time(s)
RF	81.07	77.64	71.26	74.03	233.68
SVM	80.19	79.33	64.88	71.11	445.22
DT	79.78	76.31	67.95	71.57	157.2
ETC	81	76.88	71.98	74.16	513.12
KNN	72.01	72.39	43.25	53.69	133.48
MNB	76.54	70.95	66.56	68.41	7.22
GNB	53.97	44.97	87.62	59.37	9.63
LR	78.65	79.17	59.85	67.96	179.5
NN	79.64	75.22	69.58	72.17	1203.33
ABC	75.39	74.31	54.58	62.79	305.19
GBC	73.34	76.57	44.07	55.42	645.97

Hasil *cross-validation* pada ekstraksi fitur Count Vectorizer dengan seleksi fitur Mutual Information pada data Line-Today mendapatkan rata-rata akurasi 75.60 %, presisi 73.07 %, *recall* 63.78 %, *f1 score* 66.43 %, dan waktu *running* 348.5 detik. Random Forest mendapatkan nilai tertinggi pada akurasi sedangkan Gaussian Naive Bayes mendapatkan nilai terendah pada pada akurasi. Support Vector Machine mendapatkan nilai tertinggi pada presisi sedangkan Gaussian Naive Bayes mendapatkan nilai terendah pada presisi. Gaussian Naive Bayes mendapatkan nilai tertinggi pada *recall* sedangkan Gradient Boosting Classifier mendapatkan nilai terendah pada *recall*. Extra Tree Classifier mendapatkan nilai tertinggi pada *f1 score* sedangkan K-Nearest Neighbour mendapatkan nilai terendah pada *f1 score*. Multinomial Naive Bayes mendapatkan waktu *running* tercepat sedangkan Multilayer Perceptron mendapatkan waktu *running* terlama. Sementara itu berikut pada tabel (6.28) hasil *cross validation* ekstraksi fitur TFIDF pada Line-Today dengan seleksi fitur Mutual Information.

Tabel 6.50: Hasil Cross Validation ekstraksi fitur TFIDF pada Line-Today dengan seleksi fitur Mutual Information

	accuracy(%)	precision(%)	recall(%)	f1(%)	time(s)
RF	80.55	78.59	66.56	71.7	236.41
SVM	77.75	78.36	57.59	66.08	590.18
DT	79.69	75.62	68.71	71.7	161.07
ETC	81.96	80.19	70.21	74.45	568.6
KNN	65.05	58.11	33.42	41.84	134.53
MNB	74.2	80.41	42.33	54.92	6.7
GNB	54.52	45.08	83.87	58.58	8.91
LR	76.22	79.65	50.44	61.39	43.16
NN	76.23	70.86	67.56	68.7	1005.96
ABC	74.75	71.88	56.3	62.92	306.97
GBC	74.44	76.94	47.52	58.12	647.63

Hasil *cross-validation* pada ekstraksi fitur TFIDF dengan seleksi fitur Mutual Information pada data Line-Today mendapatkan rata-rata akurasi 74.12 %, presisi 72.34 %, *recall* 58.59 %, *f1 score* 62.76 %, dan waktu *running* 337.28 detik. Extra Tree Classifier mendapatkan nilai tertinggi pada akurasi sedangkan Gaussian Naive Bayes mendapatkan nilai terendah pada pada akurasi. Multinomial Naive Bayes mendapatkan nilai tertinggi pada presisi sedangkan Gaussian Naive Bayes mendapatkan nilai terendah pada presisi. Gaussian Naive Bayes mendapatkan nilai tertinggi pada *recall* sedangkan K-Nearest Neighbour mendapatkan nilai terendah pada *recall*. Extra Tree Classifier mendapatkan nilai tertinggi pada *f1 score* sedangkan K-Nearest Neighbour mendapatkan nilai terendah pada *f1 score*. Multinomial Naive Bayes mendapatkan waktu *running* tercepat sedangkan Multilayer Perceptron mendapatkan waktu *running* terlama. Sementara itu berikut pada tabel (6.29) hasil *cross validation* ekstraksi fitur Count Vectorizer pada Twitter dengan seleksi fitur Mutual Information.

Tabel 6.51: Hasil Cross Validation ekstraksi fitur Count Vectorizer pada Twitter dengan seleksi fitur Mutual Information

	accuracy(%)	precision(%)	recall(%)	f1(%)	time(s)
RF	78.39	79.12	78.25	78.65	392.9
SVM	77.94	82.46	71.97	76.83	1246.13
DT	76.19	77.36	75.27	76.29	411.26
ETC	78.41	78.42	79.45	78.92	941.55
KNN	71.9	78.96	61.1	68.86	282.54
MNB	70	68.74	75.37	71.87	15.28
GNB	57.25	69.65	28.37	40.28	20.54
LR	78.08	82.91	71.73	76.88	356.66
NN	74.39	76.44	72.01	74.11	2610.81
ABC	77.43	82.69	70.43	76.04	694.28
GBC	76.03	82.76	66.9	73.94	1397.22

Hasil *cross-validation* pada ekstraksi fitur Count Vectorizer dengan seleksi fitur Mutual Information pada data Twitter mendapatkan rata-rata akurasi 74.18 %, presisi 78.14 %, *recall* 68.26 %, *f1 score* 72.06 %, dan waktu *running* 760.83 detik. Extra Tree Classifier mendapatkan nilai tertinggi pada akurasi sedangkan Gaussian Naive Bayes mendapatkan nilai terendah pada pada akurasi. Logistic Regression mendapatkan nilai tertinggi pada presisi sedangkan Multinomial Naive Bayes mendapatkan nilai terendah pada presisi. Extra Tree Classifier mendapatkan nilai tertinggi pada *recall* sedangkan Gaussian Naive Bayes mendapatkan nilai terendah pada *recall*. Extra Tree Classifier mendapatkan nilai tertinggi pada *f1 score* sedangkan Gaussian Naive Bayes mendapatkan nilai terendah pada *f1 score*. Multinomial Naive Bayes mendapatkan waktu *running* tercepat sedangkan Multilayer Perceptron mendapatkan waktu *running* terlama. Sementara itu berikut pada tabel (6.30) hasil *cross validation* ekstraksi fitur TFIDF pada Twitter dengan seleksi fitur Mutual Information.

Tabel 6.52: Hasil Cross Validation ekstraksi fitur TFIDF pada Twitter dengan seleksi fitur Mutual Information

	accuracy(%)	precision(%)	recall(%)	f1(%)	time(s)
RF	78.2	79.63	76.88	78.2	467.79
SVM	77.33	84.9	67.52	75.17	1390.74
DT	75.65	76.88	74.68	75.74	510.49
ETC	78.32	78.21	79.66	78.89	1241.97
KNN	69.56	79.56	54.17	64.42	285.09
MNB	70.22	67.17	81.31	73.54	13.85
GNB	57.38	68.6	30.02	41.72	19.55
LR	76.94	83.81	67.89	74.96	77.66
NN	72.72	74.14	71.29	72.63	2225.64
ABC	77.38	82.21	70.95	76.12	696.49
GBC	76.3	83.08	67.24	74.28	1407.04

Hasil *cross-validation* pada ekstraksi fitur TFIDF dengan seleksi fitur Mutual Information pada data Twitter mendapatkan rata-rata akurasi 73.64 %, presisi 78.02 %, *recall* 67.42 %, *f1 score* 71.42 %, dan waktu *running* 757.85 detik. Extra Tree Classifier mendapatkan nilai tertinggi pada akurasi sedangkan Gaussian Naive Bayes mendapatkan nilai terendah pada pada akurasi. Support Vector Machine mendapatkan nilai tertinggi pada presisi sedangkan Multinomial Naive Bayes mendapatkan nilai terendah pada presisi. Multinomial Naive Bayes mendapatkan nilai tertinggi pada *recall* sedangkan Gaussian Naive Bayes mendapatkan nilai terendah pada *recall*. Extra Tree Classifier mendapatkan nilai tertinggi pada *f1 score* sedangkan Gaussian Naive Bayes mendapatkan nilai terendah pada *f1 score*. Multinomial Naive Bayes mendapatkan waktu *running* tercepat sedangkan Multilayer Perceptron mendapatkan waktu *running* terlama.

BAB VII
KESIMPULAN DAN SARAN

7.1 Kesimpulan

Berdasarkan peneltian yang telah dilakukan maka kesimpulan yang didapat sebagai berikut.

1. Pada dataset Line-Today, pengklasifikasi terbaik pada masing-masing rata-rata semua nilai adalah Extra Tree Classifier (akurasi = 84.01 %), Support Vector Machine (presisi = 82.6 %), Decision Tree (*recall* = 74.69 %), Extra Tree Classifier (*f1 score* = 77.76 %) dan Multinomial Naive Bayes (waktu *running* = 6 detik).

2. Pada dataset Twitter, pengklasifikasi terbaik pada masing-masing rata-rata semua nilai adalah Support Vector Machine (akurasi = 81.04 %), Support Vector Machine (presisi = 84.4 %), Extra Tree Classifier (*recall* = 85.13 %), Extra Tree Classifier (*f1 score* = 81.71 %) dan Multinomial Naive Bayes (waktu *running* = 12.69 detik).

3. Pada seluruh data beserta semua jenis seleksi fitur dan pengklasifikasi, ekstraksi fitur Count Vectorizer memberikan peningkatan lebih baik daripada ekstraksi fitur TFIDF pada rata-rata nilai akurasi (0.75 %), rata-rata nilai*recall* (3.18 %), dan rata-rata nilai *f1 score* (2 %).

4. Pada seluruh data beserta semua jenis seleksi fitur dan pengklasifikasi, ekstraksi fitur TFIDF memberikan peningkatan lebih baik daripada ekstraksi fitur Count Vectorizer pada rata-rata nilai presisi (0.64 %) dan rata-rata waktu *running* (5.79 detik).

5. Pada seluruh data beserta semua jenis ekstraksi dan pengklasifikasi, seleksi fitur Chi Square memberikan peningkatan nilai terbaik dibanding ANOVA dan Mutual Information pada rata-rata nilai akurasi (2.49 %), dan presisi (3.36 %).

6. Pada seluruh data beserta semua jenis ekstraksi dan pengklasifikasi, seleksi fitur ANOVA memberikan peningkatan nilai terbaik dibanding Chi Square dan Mutual Information pada rata-rata nilai *reacall* (5.25 %), *f1 score* (5.25 %) dan waktu *running* (364.44 detik).

7.2 Saran

Saran yang dapat diberikan untuk penelitian selanjutnya adalah sebagai berikut.

1. Mendapatkan kumpulan data Kamus Besar Bahasa Indonesia untuk melakukan normalisasi pada teks yang salah tulis.
2. Membandingkan beberapa *percentile* nilai *scoring* terbaik pada masing-masing metode seleksi fitur.
3. Mendapatkan referensi resmi dari para ahli ataupun pakar dalam pelabelan sentimen positif atau negatif sebelum dimulainya pelatihan data.
4. Melakukan *random sampling* dalam jumlah tertentu pada dataset relatif besar untuk mengurangi waktu validasi.
5. Mempertimbangkan fitur yang dapat mengetahui sarkasme.
6. Memberikan parameter-parameter tambahan pada pengklasifikasi untuk mendapatkan nilai paling optimal dari masing-masing pengklasifikasi.
7. Menggunakan teknik normalisasi untuk mengatasi ketidakseimbangan jumlah label pada data latih.
8. Mencoba menggunakan jenis metode validasi lain seperti Permutation Test atau Validation Curve.

DAFTAR PUSTAKA

Adiwijaya, I. 2006, Text Mining dan Knowledge Discovery, *Komunitas Data mining Indonesia & Soft-computing Indonesia*.

Breiman, L. 2001, Random forests, *Machine learning* 45.1, pp. 5–32.

Chandani, V., Wahono, R. S., et al. 2015, Komparasi algoritma klasifikasi Machine Learning dan feature selection pada analisis sentimen review film, *Journal of Intelligent Systems* 1.1, pp. 56–60.

Chang, C.-C. dan Lin, C.-J. 2011, LIBSVM: A library for support vector machines, *ACM transactions on intelligent systems and technology (TIST)* 2.3, p. 27.

Fawcett, T. 2006, An introduction to ROC analysis, *Pattern recognition letters* 27.8, pp. 861–874.

Freund, Y. dan Schapire, R. E. 1997, A decision-theoretic generalization of on-line learning and an application to boosting, *Journal of computer and system sciences* 55.1, pp. 119–139.

Friedman, J. H. 2002, Stochastic gradient boosting, *Computational statistics & data analysis* 38.4, pp. 367–378.

Geurts, P., Ernst, D., dan Wehenkel, L. 2006, Extremely randomized trees, *Machine learning* 63.1, pp. 3–42.

Glorot, X., Bordes, A., dan Bengio, Y. 2011, Deep sparse rectifier neural networks, *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 315–323.

Hand, D. J. dan Yu, K. 2001, Idiot's Bayes—not so stupid after all?, *International statistical review* 69.3, pp. 385–398.

Haykin, S., 1994, *Neural networks: a comprehensive foundation*, Prentice Hall PTR.

Hidayatullah, A. F. dan Azhari, A. S. 2015, Analisis sentimen dan klasifikasi kategori terhadap tokoh publik pada twitter, *Seminar Nasional Informatika (SEMNASIF)*, vol. 1, 1.

Ho, T. K. 2002, A data complexity analysis of comparative advantages of decision forest constructors, *Pattern Analysis & Applications* 5.2, pp. 102–112.

Indonesian LINE User 2016 - Survey Report. 2016.

Ipmawati, J. et al. 2017, Komparasi Teknik Klasifikasi Teks Mining Pada Analisis Sentimen, *IJNS-Indonesian Jornal on Networking and Security* 6.1.

Jaskowiak, P. A. dan Campello, R. 2011, Comparing correlation coefficients as dissimilarity measures for cancer classification in gene expression data, *Proceedings of the Brazilian symposium on bioinformatics*, Brasília, pp. 1–8.

Kamiński, B., Jakubczyk, M., dan Szufel, P. 2018, A framework for sensitivity analysis of decision trees, *Central European journal of operations research* 26.1, pp. 135–159.

Lestari, A. R. T., Perdana, R. S., dan Fauzi, M. A. 2017, Analisis Sentimen Tentang Opini Pilkada Dki 2017 Pada Dokumen Twitter Berbahasa Indonesia Menggunakan Näive Bayes dan Pembobotan Emoji, *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN* 2548, p. 964X.

Li, C. 2016, A Gentle introduction to gradient boosting, *URL: http://www. ccs. neu. edu/home/vip/teach/MLcourse/4_ boosting/slides/gradient_boosting. pdf*.

Ling, J., Kencana, I. P.E. N., dan Oka, T. B. 2014, Analisis Sentimen Menggunakan Metode Naïve Bayes Classifier Dengan Seleksi Fitur Chi Square, *E-Jurnal Matematika* 3.3, pp. 92–99.

Liu, Y., Bi, J.-W., dan Fan, Z.-P. 2017, Multi-class sentiment classification: The experimental comparisons of feature selection and machine learning algorithms, *Expert Systems with Applications* 80, pp. 323–339.

Manning, C., Raghavan, P., dan Schütze, H. 2010, Introduction to information retrieval, *Natural Language Engineering* 16.1, pp. 100–103.

Onan, A., Korukoğlu, S., dan Bulut, H. 2016, Ensemble of keyword extraction methods and classifiers in text classification, *Expert Systems with Applications* 57, pp. 232–247.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. 2011, Scikit-learn: Machine learning in Python, *Journal of machine learning research* 12.Oct, pp. 2825–2830.

Peng, C.-Y. J., Lee, K. L., dan Ingersoll, G. M. 2002, An introduction to logistic regression analysis and reporting, *The journal of educational research* 96.1, pp. 3–14.

Prusa, J. D., Khoshgoftaar, T. M., dan Dittman, D. J. 2015, Impact of feature selection techniques for tweet sentiment classification, *The Twenty-Eighth International Flairs Conference*.

Ravi, K. dan Ravi, V. 2015, A survey on opinion mining and sentiment analysis: tasks, approaches and applications, *Knowledge-Based Systems* 89, pp. 14–46.

Rennie, J, Shih, L, Teevan, J, dan Karger, D. 2003, Tackling the poor assumptions of Naive Bayes classifiers (PDF), ICML.

Saputra, N., Adji, T. B., dan Permanasari, A. E. 2015, Analisis sentimen data presiden Jokowi dengan preprocessing normalisasi dan stemming menggunakan metode naive bayes dan SVM, *Jurnal Dinamika Informatika* 5.1.

Shah, F. P. dan Patel, V. 2016, A review on feature selection and feature extraction for text classification, *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, IEEE, pp. 2264–2268.

Siddiqi, S. dan Sharan, A. 2015, Keyword and keyphrase extraction techniques: a literature review, *International Journal of Computer Applications* 109.2.

Sukendar, M. U., Sos, S, dan Kom, M. 2017, PEMILIHAN PRESIDEN, MEDIA SOSIAL DAN PENDIDIKAN POLITIK BAGI PEMILIH PEMULA, *Jurnal IKON Prodi D3 Komunikasi Massa–Politeknik Indonusa Surakarta Vol* 1.5.

Suryotomo, R. 2018, Analisis sentimen untuk mengetahui elektabilitas tokoh politik menggunakan metode multinomial naive bayes, PhD thesis, Universitas Gadjah Mada.

Vijayarani, S, Ilamathi, M. J., dan Nithya, M. 2015, Preprocessing techniques for text mining-an overview, *International Journal of Computer Science & Communication Networks* 5.1, pp. 7–16.

Wicaksono, A. J. et al. 2016, A proposed method for predicting US presidential election by analyzing sentiment in social media, *2016 2nd international conference on science in information technology (ICSITech)*, IEEE, pp. 276–280.

Wikarsa, L. dan Thahir, S. N. 2015, A text mining application of emotion classifications of Twitter's users using Naive Bayes method, *2015 1st International Conference on Wireless and Telematics (ICWT)*, IEEE, pp. 1–6.

Winter, B. 2015, The F distribution and the basic principle behind ANOVAs, *Author, Birmingham*.

Witten, D. M. et al. 2011, Classification and clustering of sequencing data using a Poisson model, *The Annals of Applied Statistics* 5.4, pp. 2493–2518.

Yuniarni, S. 2018, Indonesia had 143m internet users in 2017: APJII, *Jakarta Post, February* 19.