

**IMPLEMENTASI METODE RANDOM FOREST PADA
KLASIFIKASI DATA ULASAN KONSUMEN PERUSAHAAN
(Studi Kasus: Aplikasi KAI Access)**

Skripsi

HALAMAN SAMPUL



Oleh :

Wildan Abdul Aziz

11160910000095

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH

JAKARTA

1442 H / 2021 M

PERNYATAAN ORISINALITAS

PERNYATAAN ORISINALITAS

Dengan ini saya menyatakan bahwa:

1. Skripsi ini merupakan hasil karya asli saya yang diajukan untuk memenuhi salah satu persyaratan memperoleh gelar strata 1 di UIN Syarif Hidayatullah Jakarta.
2. Semua sumber yang saya gunakan dalam penulisan ini telah saya cantumkan sesuai dengan ketentuan yang berlaku di UIN Syarif Hidayatullah Jakarta.
3. Apabila di kemudian hari terbukti bahwa karya ini bukan hasil karya asli saya atau merupakan hasil jiplakan dari karya orang lain, maka saya bersedia menerima sanksi yang berlaku di UIN Syarif Hidayatullah Jakarta.

Jakarta, 26 Februari 2023


Wildan Abdul Aziz



KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Komputer Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Syarif Hidayatullah Jakarta. Saya menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

1. Bapak Ir. Nasrul Hakiem, S.Si., M.T. Ph.D selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Syarif Hidayatullah Jakarta;
2. Bapak Dr. Imam Marzuki Shofi, MT selaku Ketua Program Studi Teknik Informatika dan Bapak Andrew Fiade, M. Kom selaku Sekretaris Program Studi Teknik Informatika.
3. Bapak Viktor Amrizal, M.Kom dan Bu Viva Arifin, MMSI, Ph.D selaku Dosen Pembimbing I dan II yang telah menyediakan waktu, tenaga dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini memberikan bimbingan bantuan, masukan, semangat dan motivasi dalam menyelesaikan skripsi ini.
4. Bapak/ Ibu Dosen dan staff UIN Jakarta, khususnya Fakultas Sains dan Teknologi Program Studi Teknik Informatika yang telah memberikan ilmu dan pengalaman yang berharga di dalam maupun di luar kelas.
5. Orang tua dan keluarga saya yang telah memberikan bantuan dukungan material dan moral.


6. Emir Akbar S.Kom, Gerald Halim S.Kom, Ikhum Muhammad dll, yang telah banyak membantu dalam penyusunan skripsi dan senantiasa memberikan semangat
7. Teman-teman KPA ARKADIA yang senantiasa memberikan semangat dan dukungannya.
8. Serta semua pihak yang telah membantu penyelesaian skripsi ini yang tidak dapat disebut namanya satu per satu

Akhir kata, saya berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Ciputat, Jakarta 25 Januari 2023

Wildan Abdul Aziz





Nama : Wildan Abdul Aziz
Program Studi : Teknik Informatika
Judul : Klasifikasi Sengketa Komplain Perusahaan Menggunakan Metode Random Forest (Studi Kasus: KAI Access)

ABSTRAK

Seiring dengan berkembangnya kemajuan teknologi, pada saat ini masyarakat dapat menilai atau mengungkapkan pendapat publik terkait kepuasannya sebagai pengguna layanan. Banyak media untuk mengungkapkan atau menilai hal tersebut, salah satunya yaitu melalui Google Playstore. Dengan begitu Google Playstore dapat menjadi sumber data complain konsumen terhadap layanan aplikasi. KAI Access contohnya, yang merupakan salah satu layanan aplikasi yang memudahkan masyarakat dalam mengakses pembelian tiket kereta. Sudah sejak 2014 PT. KAI meluncurkan aplikasi namun ternyata tidak sedikit yang mengeluhkan kurangnya layanan aplikasi di Google Play Store. Dengan tingkat akurasi yang cukup tinggi yang dihasilkan oleh metode Random Forest, maka dari itu metode Random Forest dirasa menjadi metode yang tepat digunakan untuk menganalisis permasalahan sengketa complain konsumen. Tujuan dari penelitian ini adalah untuk mengetahui hasil dari penerapan algoritma Random Forest dalam melakukan klasifikasi sengketa komplain pelanggan KAI Access dan mengetahui akurasi dari algoritma Random Forest dalam melakukan klasifikasi. Metode penggunaan data yang digunakan pada penelitian ini adalah studi literatur dan filtering data. Sedangkan metode implementasi dimulai dari tahap analisis,

implementasi, dan pengujian. Adapun hasil dari penelitian ini menunjukkan bahwa hasil klasifikasi, classification report, AUC-ROC Curve, dan confusion matrix menunjukkan hasil yang baik. Dari 500 data yang diuji, terdapat 442 ulasan yang dapat diprediksi benar, sehingga penggunaan algoritma Random Forest dalam melakukan klasifikasi sengketa komplain pelanggan perusahaan dapat dikatakan sangat baik.

Kata Kunci: Random Forest, klasifikasi, confusion matrix.

Nama : Wildan Abdul Aziz

Program Studi : Teknik Informatika

Judul : Classification of Company Complaint Disputes Using the Random Forest Method (Case Study: KAI Access)

ABSTRACT

Along with the development of technological advances, at this time people can assess or express public opinion regarding their satisfaction as service users. There are many media to express or evaluate this, one of which is through the Google Playstore. That way Google Playstore can become a source of consumer complaint data against application services. KAI Access is an example, which is an application service that makes it easier for the public to access buying train tickets. Since 2014 PT. KAI launched the application, but apparently not a few complained about the lack of application services on the Google Play Store. With a fairly high level of accuracy produced by the Random Forest method, therefore the Random Forest method is felt to be the right method to use to analyze consumer complaint dispute problems. The purpose of this research is to find out the results of applying the Random Forest algorithm in classifying KAI Access customer complaint disputes and knowing the accuracy of the Random Forest algorithm in carrying out classifications. The data

usage method used in this research is literature study and data filtering. While the implementation method starts from the analysis, implementation, and testing stages. The results of this study indicate that the classification results, classification report, AUC-ROC Curve, and confusion matrix show good results. Of the 500 data tested, there were 442 reviews that could be predicted to be correct, so the use of the Random Forest algorithm in classifying company customer complaint disputes can be said to be very good.

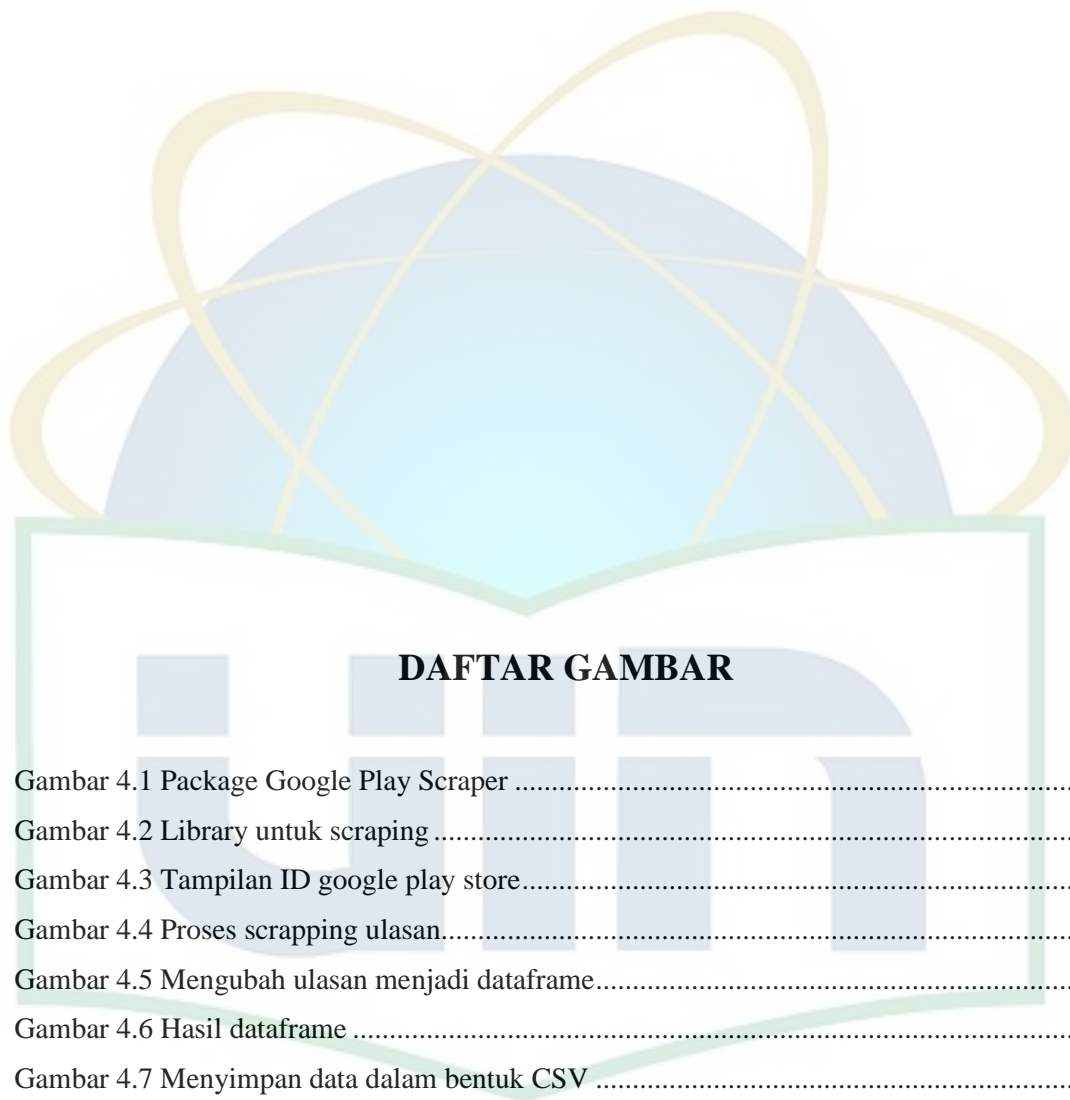
Keywords: Random Forest, classification, confusion matrix.

DAFTAR ISI

LEMBAR PERSETUJUAN.....	ii
PENGESAHAN UJIAN	iii
KATA PENGANTAR	1
ABSTRAK.....	4
ABSTRACT.....	5
DAFTAR ISI.....	6
DAFTAR GAMBAR	9
DAFTAR LAMPIRAN.....	Error! Bookmark not defined.
BAB I.....	11
PENDAHULUAN	11
1.1. Latar Belakang.....	11
1.2. Rumusan Masalah	14
1.3. Batasan Masalah	15
1.4. Tujuan Penelitian.....	15
1.5. Manfaat Penelitian.....	15
1.6. Metode Penelitian.....	15
1.6.1. Metode Pengumpulan Data.....	15
1.6.2. Metode Implementasi	16

1.7. Sistematika Penulisan	16
BAB II.....	18
TINJAUAN PUSTAKA DAN LANDASAN TEORI	18
2.1. TINJAUAN PUSTAKA	18
2.2. LANDASAN TEORI.....	28
2.2.1. Data Mining	28
2.2.2. Text Mining	30
2.2.3. Text Preprocessing.....	30
2.2.4. Pelabelan Data.....	32
2.2.5. TF – IDF.....	32
2.2.6. Klasifikasi	33
2.2.7. Random Forest	33
2.2.8. Confusion Matrix	35
BAB III	37
METODE PENELITIAN.....	37
3.1. Alur Penelitian	37
3.2. Metode Usulan.....	38
3.3. Pengujian	39
3.4. Diagram Alir Penelitian.....	39
BAB IV	43
IMPLEMENTASI.....	43
4.1. Web Scraping	43
4.1.1. Install Google Play Scraper Package.....	43
4.1.2. Install Library yang Dibutuhkan	44
4.1.3. Membuka ID Aplikasi pada Google Playstore	44
4.1.4. Scraping Ulasan	45
4.1.5. Membuat Hasil Scraping Menjadi Dataframe	45
4.1.6. Menyimpan Hasil Scraping dalam Bentuk CSV	46
4.2. Text Preprocessing.....	46
4.2.1. Membuka dataset.....	46
4.2.2. Memilih Variabel yang Penting	47
4.2.3. Casefolding	47
4.2.4. Lemmatisasi.....	48
4.2.5. Stemming	48

4.2.6.	Slang Word Standardization.....	49
4.2.7.	Stopword Removal.....	50
4.2.8.	Unwanted Word Removal.....	50
4.2.9.	Menghapus Kata yang Kurang dari 3 Huruf.....	51
4.2.10.	Split Word.....	51
4.3.	Memberi Label pada Data Ulasan.....	52
4.4.	Visualisasi Word Cloud.....	55
4.5.	Term Frecuency-Inverse Document Frecuency (TF-IDF).....	56
4.6.	Membagi Data Latih dan Data Uji.....	57
4.7.	Klasifikasi Random Forest.....	58
4.7.1.	Model Random Forest.....	58
BAB V	59
HASIL DAN PEMBAHASAN	59
5.1.	Hasil Klasifikasi.....	59
5.3.	AUC-ROC Curve.....	59
5.2.	Classification Report.....	60
5.4.	Confusion Matrix.....	61
BAB VI	64
PENUTUP	64
6.1	Kesimpulan.....	64
DAFTAR PUSTAKA	65



Gambar 4.1 Package Google Play Scraper	44
Gambar 4.2 Library untuk scraping	44
Gambar 4.3 Tampilan ID google play store.....	44
Gambar 4.4 Proses scrapping ulasan.....	45
Gambar 4.5 Mengubah ulasan menjadi dataframe.....	45
Gambar 4.6 Hasil dataframe	45
Gambar 4.7 Menyimpan data dalam bentuk CSV	46
Gambar 4.8 Membuka dataset CSV	46
Gambar 4.9 Memilih feature yang akan di preprocessing	47
Gambar 4.10 Code proses casefolding.....	47
Gambar 4.11 Hasil proses casefolding.....	48
Gambar 4.12 Proses dan hasil Lemmatisasi.....	48

Gambar 4.13 Proses Stemming	48
Gambar 4.14 Membuka dictionary slang word.....	49
Gambar 4.15 Proses Stemming	50
Gambar 4.16 Proses stopwords removal	50
Gambar 4.17 Proses menghilangkan kata yang tidak diinginkan	51
Gambar 4.18 Proses menghapus kata yang kurang dari 3 huruf.....	51
Gambar 4.19 Proses memisahkan kata-kata pada ulasan.....	52
Gambar 4.20 Dictionary kata positif (kiri) dan dictionary kata negatif (kanan).....	53
Gambar 4.21 Proses pelabelan ulasan.....	54
Gambar 4.22 Menyimpan label dalam dataframe	54
Gambar 4.23 Code pie chart label data ulasan.....	54
Gambar 4.24 <i>Pie chart</i> dari label ulasan.....	55
Gambar 4.25 Hasil visualisasi word cloud positif (kiri) dan negatif (kanan)	56
Gambar 4.26 Code Proses TF-IDF	56
Gambar 4.27 Tampilan Bobot Kata Setelah Proses TF-IDF.....	57
Gambar 4.29 Model Random Forest Classifier	58
Gambar 4.30 Akurasi Klasifikasi.....	59
Gambar 4.31 Mengeluarkan Hasil Classification Report	61
Gambar 4.32 Code untuk membuat kurva AUC-ROC	60
Gambar 4.33 Kurva AUC_ROC	60
Gambar 4.34 Code Confusion Matrix.....	62
Gambar 4.35 Confusion Matrix	62

BAB I

PENDAHULUAN

1.1.Latar Belakang

Kepuasan pelanggan adalah salah satu indikator yang dapat diukur dalam pelayanan sebuah perusahaan. Keberadaan pelanggan tentu adalah salah satu yang menjadikan sebuah perusahaan atau penyedia layanan tetap memiliki profit dan dapat semakin berkembang. Pelanggan adalah kunci keberhasilan dari sebuah usaha yang dijalankan perusahaan. Tingkat kesuksesan dan keberhasilan perusahaan diukur dari tingkat profit yang didapatkan. Profit yang tinggi akan sangat dipengaruhi oleh jumlah pelanggan aktif yang melakukan transaksi dengan perusahaan. Contoh nyata dari kondisi ini adalah data perusahaan musik Spotify yang mengalami kenaikan jumlah pelanggan (pelanggan yang loyal) sebanyak 9 juta pelanggan pada kuartal IV tahun 2018, hal ini secara langsung juga ikut mendongkrak pendapatan perusahaan sebesar 11% dari kuartal sebelumnya.

KAI Access juga merupakan salah satu layanan aplikasi untuk memudahkan masyarakat dalam pembelian tiket kereta api. Alfian Willy Saputra menganalisis permasalahan aplikasi KAI Access menggunakan metode Text Mining dan Fishbone Diagram yang menghasilkan adanya beberapa keluhan layanan yang terkait dengan pencarian, pemesanan hingga pembayaran tiket, server, serta masalah terkait akun pengguna aplikasi (Saputra, 2019). Berdasarkan hal tersebut, penulis tertarik untuk menganalisis data yang ada pada aplikasi tersebut.

Data saat ini menjadi salah satu hal yang sangat berpengaruh dalam jalannya sebuah usaha, karena dari data dapat dilihat dan dipotret gambaran



di lapangan tentang layanan yang diberikan. Data saja tidak cukup, jika hanya sekedar data perlu dilakukan pemrosesan data atau pengolahan data, teknik ini biasa disebut data mining. Pengolahan data menggunakan teknik data mining dapat memberikan banyak opsi informasi bagi perusahaan dapat berupa prediksi, klasifikasi maupun klustering. Data yang dilakukan pengolahan menggunakan teknik data mining, selanjutnya dapat dikembangkan kembali ke dalam bentuk penelitian lain yang dapat dituangkan dalam sebuah sistem pakar, sistem pendukung keputusan atau dalam bentuk system yang mampu membantu dalam penentuan kebijakan selanjutnya (Pradana, 2018).

Data juga dapat dilakukan visualisasi dan pengolahan yang menghasilkan informasi baru dan lebih bermanfaat (Pradana, 2020). Selain itu pengolahan data juga dapat dalam bentuk klasifikasi atau klustering dalam pengelompokan data (Musthofa Galih Pradana, Azriel Christian Nurcahyo, 2020). Sedangkan konsep pengembangan ke depan juga dapat dilakukan dalam bentuk Internet of Things yang menerapkan konsep konsep dasar dalam pengolahan data (Faizah et al., 2019) (Lazulfa & Saputro, 2017)

Kaitlin Kirasich dkk yang membandingkan Random Forest dan Logistic Regression. Hasil penelitian adalah tingkat true positif untuk random forest lebih tinggi dari regresi logistik dan menghasilkan tingkat false positif yang lebih tinggi untuk dataset dengan peningkatan variabel noise. Setiap studi kasus terdiri dari 1000 simulasi dan kinerja model secara konsisten menunjukkan tingkat false positive untuk random forest dengan 100 pohon menjadi statistik berbeda dari regresi logistik (Kirasich et al., 2018).

Penilaian Kredit dalam penelitian Akhmad Syukron dan Agus Subekti menerapkan 2 metode yaitu Random Over-Under Sampling dan Random Forest pada dataset German Credit. Metode Random Forest memiliki nilai akurasi yang lebih baik dibandingkan dengan beberapa metode lainnya dengan nilai

akurasi sebesar 0,76 atau 76%. Hasil penelitian menunjukkan bahwa penerapan resampling dengan metode Random Over-Under Sampling pada algoritma Random Forest dapat meningkatkan kinerja akurasi secara efektif pada klasifikasi tidak seimbang untuk penilaian kredit pada dataset German Credit (Syukron & Subekti, 2018).

Penggunaan random forest dan CART pernah dibandingkan oleh Riska Chairunisa dkk yang menghasilkan kesimpulan kurasi terbaik yang didapat pada penelitian ini untuk data breast cancer sebesar 76,92% dengan CART-DWT, Colon Tumor sebesar 90,1% dengan RF-DWT, lung cancer sebesar 100% dengan RF-DWT, prostate tumor sebesar 95,49% dengan RF-DWT, dan ovarian cancer sebesar 100% dengan RF-DWT. Dari hasil tersebut maka dapat disimpulkan bahwa RF-DWT lebih baik dibandingkan CART-DWT (Riska Chairunisa et al., 2020)

Dari ketiga penelitian tersebut, akurasi yang dihasilkan oleh metode Random Forest cukup baik dengan menghasilkan akurasi yang cukup tinggi. Di sisi lain juga keterbaruan metode Random Forest membuat peneliti akan menggunakan metode Random Forest dalam penelitian ini dengan tujuan akhir mendapatkan akurasi dari klasifikasi menggunakan metode random forest.

1.2.Rumusan Masalah

Berdasarkan latar belakang diatas, maka peneliti mengambil rumusan masalah sebagai berikut :

- a. Bagaimana menerapkan algoritma Random Forest dalam melakukan klasifikasi sengketa komplain pelanggan perusahaan? dengan studi kasus aplikasi KAI Access?
- b. Berapa akurasi yang dihasilkan oleh metode Random Forest dalam melakukan klasifikasi komplain?

1.3. Batasan Masalah

Klasifikasi ini memiliki cakupan yang luas, untuk itu, agar penelitian lebih fokus, maka peneliti membuat batasan masalah yaitu :

- a. Menggunakan algoritma Random Forest untuk klasifikasi sengketa komplain pelanggan.
- b. Data yang digunakan dalam penelitian diambil dari dataset public (ulasan Google Playstore KAI Access).
- c. Data yang digunakan adalah berjumlah 5000 data.
- d. Tools yang digunakan yaitu google colaboratory.

1.4. Tujuan Penelitian

Mengetahui hasil dari penerapan algoritma Random Forest dalam melakukan klasifikasi sengketa komplain pelanggan perusahaan KAI Acces dan mengetahui akurasi dari algoritma Random Forest dalam melakukan klasifikasi.

1.5. Manfaat Penelitian

1. Bagi Penulis

Menerapkan ilmu yang didapatkan, dalam hal ini ilmu data mining atau data science untuk melakukan klasifikasi dengan topik sengketa komplain dan mengklasifikasikannya menggunakan algoritma Random Forest.

2. Bagi Pembaca

Memberikan wawasan dan referensi tentang ilmu data mining dan mendapatkan gambaran langkah proses klasifikasi menggunakan algoritma Random Forest.

1.6. Metode Penelitian

1.6.1. Metode Pengumpulan Data

Dalam melakukan pengumpulan data, penulis menggunakan metode berikut ini :

1. Studi Literatur

Penulis mengumpulkan data dari jurnal atau karya tulis yang relevan, hal tersebut dapat membantu penulis untuk menambah referensi sesuai dengan permasalahan.

2. Filtering Data

Penulis menggunakan filtering data dari dataset untuk mendapatkan gambaran variable dan data yang akan digunakan.

1.6.2. Metode Implementasi

Dalam mengimplementasikan model yang penulis ajukan, maka penulis melakukan dengan beberapa tahapan :

1. Analisis

Analisis dilakukan setelah data berhasil dikumpulkan melalui metode scraping. Data akan dianalisis apakah layak dilakukan pengolahan data, atau perlu melakukan pengumpulan data kembali.

2. Implementasi

Proses implementasi dilakukan dengan berbagai tahapan, seperti pre processing, TF-IDF, melakukan klasifikasi ke metode random forest.

3. Pengujian

Pengujian dilakukan untuk mendapatkan hasil akurasi dari algoritma dalam melakukan klasifikasi.

1.7. Sistematika Penulisan

Sistematika penulisan yang dilakukan dalam penelitian ini terdiri dari beberapa bagian :

BAB I PENDAHULUAN

Bab ini membahas tentang hal umum dalam penelitian, seperti latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metode penelitian dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini menjelaskan tentang pengertian dan teori-teori yang dibutuhkan dalam melaksanakan penelitian ini.

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan uraian secara rinci mengenai metode yang digunakan pada saat penelitian yaitu metode pengumpulan data, metode pengimplementasian dan lain sebagainya.

BAB IV IMPLEMENTASI

Bab ini membahas tentang analisis kebutuhan, perancangan dan implementasi sistem sesuai dengan metode yang digunakan dalam penelitian.

BAB V HASIL DAN PEMBAHASAN

Bab ini membahas tentang output dari analisis kebutuhan, perancangan dan implementasi yang dilakukan oleh peneliti sesuai dengan algoritma yang dipakai.

BAB VI PENUTUP

Bab ini berisikan kesimpulan dari apa yang telah dilakukan oleh peneliti dan saran dari pembaca untuk pengembangan penelitian menjadi lebih baik

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1. TINJAUAN PUSTAKA

Penelitian dari Nugraha Listiana Hanun dan Achmad Udin Zailani Penentuan Kelayakan Pemberian Kredit menghasilkan kesimpulan hasil pengujian dengan algoritma klasifikasi Random Forest mampu menganalisis kredit yang bermasalah dan yang debitur yang tidak bermasalah dengan nilai akurasi sebesar 87,88%. Di samping itu, model pohon keputusan ternyata mampu meningkatkan akurasi dalam menganalisis kelayakan kredit yang diajukan calon debitur (Zailani & Hanun, 2020).

Siska Devella dkk menuliskan Klasifikasi Motif Songket Palembang menggunakan Random Forest. Kesimpulan yang dapat diambil adalah metode SIFT dan Random Forest untuk klasifikasi citra motif kain Songket Palembang dapat memberikan akurasi yang cukup baik, dimana metode SIFT dan Random Forest mampu menghasilkan rata-rata overall accuracy 92,98%, per class accuracy 94,07%, presision 92,98%, dan recall 89,74% (Devella et al., 2020).

Penggunaan random forest dan CART pernah dibandingkan oleh Riska Chairunisa dkk yang menghasilkan kesimpulan kurasi terbaik yang didapat pada penelitian ini untuk data breast cancer sebesar 76,92% dengan CART-DWT, Colon Tumor sebesar 90,1% dengan RF-DWT, lung cancer sebesar 100% dengan RF-DWT, prostate tumor sebesar 95,49% dengan RF-DWT, dan ovarian cancer sebesar 100% dengan RF-DWT. Dari hasil tersebut maka dapat disimpulkan bahwa RF-DWT lebih baik dibandingkan CART-DWT (Riska Chairunisa et al., 2020)

Debby Alita dan Auliya Rahman juga menerapkan algoritma random forest dalam klasifikasi sarkasme dalam konteks analisis sentiment. Hasil penelitian ini didapatkan peningkatan nilai rata-rata akurasi sebesar 16,61%, nilai presisi sebesar 5,45%, nilai recall sebesar 9,64% dan kenaikan nilai F1score sebesar 11,27% dengan jumlah data sebanyak 2.027 (Alita & Isnain, 2020).

Penelitian dari Yoga Religia dkk dalam melakukan Klasifikasi Data Bank Marketing menghasilkan kesimpulan algoritma RF dengan akurasi sebesar 88,30%, AUC (+) sebesar 0,500 dan AUC(-) sebesar 0,000. Adapun penggunaan optimasi Bagging dan Genetic Algorithm ternyata belum mampu meningkatkan performa dari algoritma RF untuk klasifikasi data Bank Marketing (Yoga Religia, Agung Nugroho, 2021).

Gde Agung Brahmana Suryanegara dkk melakukan Klasifikasi pada Algoritma Random Forest untuk Deteksi Pasien Penderita Diabetes. Berdasarkan hasil penelitian, akurasi terbaik dihasilkan model 1 (Min-max normalization-RF) sebesar 95.45%, model 2 (Z-score normalization-RF) sebesar 95%, dan model 3 (Tanpa normalisasi data-RF) sebesar 92%. Dari hasil tersebut disimpulkan bahwa model 1 (Min-max normalization-RF) lebih baik dibandingkan dua model normalisasi data lainnya dan mampu meningkatkan performansi klasifikasi Random Forest sebesar 95.45% (Suryanegara et al., 2021).

Penilaian Kredit dalam penelitian Akhmad Syukron dan Agus Subekti menerapkan 2 metode yaitu Random Over-Under Sampling dan Random Forest pada dataset German Credit. Metode Random Forest memiliki nilai akurasi yang lebih baik dibandingkan dengan beberapa metode lainnya dengan nilai akurasi sebesar 0,76 atau 76%. Hasil penelitian menunjukkan bahwa penerapan resampling dengan metode Random Over-Under Sampling pada algoritma Random Forest dapat meningkatkan kinerja akurasi secara efektif pada

klasifikasi tidak seimbang untuk penilaian kredit pada dataset German Credit (Syukron & Subekti, 2018).

Yusuf Sulistyo Nugroho dan Nova Emiliyawati menuliskan tentang Klasifikasi Variabel Tingkat Penerimaan Konsumen yang menghasilkan kesimpulan variabel yang menempati sebagai root node dalam pohon keputusan merupakan variabel yang paling signifikan dalam mempengaruhi tingkat penerimaan mobil pada konsumen. Hasil dari sistem klasifikasi yang dibangun dapat dijadikan pertimbangan bagi produsen mobil di masa mendatang, sehingga produksi mobil menjadi lebih efektif, dapat meminimalisir kerugian, dan meningkatkan profitabilitas produsen (Yusuf Sulistyo Nugroho, 2017).

Jieyu Li dkk menuliskan penerapan Klasifikasi random forest dalam upaya pengendalian banjir. Hasil penelitian menunjukkan bahwa model RFC memiliki karakteristik akurasi klasifikasi tinggi, sensitivitas rendah terhadap sampel banjir dan stabilitas (Li et al., 2020).

Kaitlin Kirasich dkk yang membandingkan Random Forest dan Logistic Regression. Hasil penelitian adalah tingkat true positif untuk random forest lebih tinggi dari regresi logistik dan menghasilkan tingkat false positif yang lebih tinggi untuk dataset dengan peningkatan variabel noise. Setiap studi kasus terdiri dari 1000 simulasi dan kinerja model secara konsisten menunjukkan tingkat false positive untuk random forest dengan 100 pohon menjadi statistik berbeda dari regresi logistik (Kirasich et al., 2018).

Adapun detail dari rujukan penelitian ditunjukkan pada

Tabel 1.

Tabel 1. Review Penelitian Relevan

No	Judul	Objek	Peneliti, Publikasi, Tahun	Metode	Hasil
1	Penerapan Algoritma Klasifikasi Random Forest Untuk Penentuan Kelayakan Pemberian Kredit Di Koperasi Mitra Sejahtera	Koperasi Mitra Sejahtera	Nugraha Listiana Hanun , Achmad Udin Zailani, Journal Of Technology Information, 2020.	Random Forest	Algoritma Random Forest mampu menganalisis kredit yang bermasalah dan yang debitur yang tidak bermasalah yang menghasilkan nilai akurasi sebesar 87,88%.
2	Implementasi Random Forest Untuk Klasifikasi Motif Songket Palembang Berdasarkan SIFT	Motif Songket Palembang	Siska Devella, Yohannes, Firda Novia Rahmawati, Jurnal Teknik Informatika dan Sistem Informasi, 2020.	Random Forest	Hasil pengujian menunjukkan bahwa metode SIFT dan Random Forest untuk klasifikasi citra motif kain Songket Palembang dapat memberikan akurasi yang cukup baik, dimana metode SIFT dan Random Forest mampu menghasilkan rata-rata overall accuracy 92,98%, per class accuracy 94,07%,

					presision 92,98%, dan recall 89,74%.
3	Perbandingan CART dan Random Forest untuk Deteksi Kanker berbasis Klasifikasi Data Microarray	Data Microarray	Riska Chairunisa, Adiwijaya, Widi Astuti, Rekayasa Sistem dan Teknologi Informasi, 2020	CART dan Random Forest	Akurasi terbaik yang didapat pada penelitian ini untuk data breast cancer sebesar 76,92% dengan CART-DWT, Colon Tumor sebesar 90,1% dengan RF-DWT, lung cancer sebesar 100% dengan RF-DWT, prostate tumor sebesar 95,49% dengan RF-DWT, dan ovarian cancer sebesar 100% dengan RF-DWT. Dari hasil tersebut maka dapat disimpulkan bahwa RF-DWT lebih baik dibandingkan CART-DWT.
4	Pendeteksian Sarkasme pada Proses Analisis Sentimen Menggunakan Random Forest Classifier	Sosial Media	Debby Alita, Auliya Rahman, Jurnal Komputasi, 2020	Random Forest	Hasil penelitian ini didapatkan peningkatan nilai rata-rata Akurasi sebesar 16,61%, nilai presisi sebesar

					5,45%, nilai recall sebesar 9,64% dan kenaikan nilai F1score sebesar 11,27% dengan jumlah data sebanyak 2.027 dengan rincian data dengan label positif berjumlah 1023.
5	Analisis Perbandingan Algoritma Optimasi pada Random Forest untuk Klasifikasi Data Bank Marketing	Data Bank Marketing	Yoga Religia , Agung Nugroho , Wahyu Hadikristanto , Rekayasa Sistem dan Teknologi Informasi, 2021	Random Forest	Performa paling optimal dari klasifikasi data Bank Marketing adalah dengan menggunakan algoritma RF dengan akurasi sebesar 88,30%, AUC (+) sebesar 0,500 dan AUC(-) sebesar 0,000. Adapun penggunaan

					optimasi Bagging dan Genetic Algorithm ternyata belum mampu meningkatkan performa dari algoritma RF untuk klasifikasi data Bank Marketing.
6	Peningkatan Hasil Klasifikasi pada Algoritma Random Forest untuk Deteksi Pasien Penderita Diabetes Menggunakan Metode Normalisasi	Pasien Penderita Diabetes	Gde Agung Brahmana Suryanegara , Adiwijaya , Mahendra Dwifebri Purbolaksono , Rekayasa Sistem dan Teknologi Informasi, 2021	Random Forest	Dari hasil tersebut disimpulkan bahwa model 1 (Min-max normalization-RF) lebih baik dibandingkan dua model normalisasi data lainnya dan mampu meningkatkan performansi klasifikasi

					Random Forest sebesar 95.45%.
7	Penerapan Metode Random Over-Under Sampling dan Random Forest untuk Klasifikasi Penilaian Kredit	Penilaian Kredit	Akhmad Syukron, Agus Subekti, Jurnal Informatika, 2018	Random Forest	Hasil penelitian menunjukkan bahwa penerapan resampling dengan metode Random Over-Under Sampling pada algoritma Random Forest dapat meningkatkan kinerja akurasi secara efektif pada klasifikasi tidak seimbang untuk penilaian kredit pada dataset German Credit.
8	Sistem Klasifikasi Variabel Tingkat Penerimaan Konsumen Terhadap Mobil Menggunakan Metode Random Forest	Data Konsumen Mobil	Yusuf Sulistyo Nugroho dan Nova Emiliyawati, Jurnal Teknik Elektro Vol. 9 No. 1, 2017	Random Forest	Adapun variabel yang menempati sebagai root node dalam pohon keputusan merupakan variabel yang paling signifikan dalam mempengaruhi tingkat penerimaan mobil pada konsumen.

9	Intelligent identification of effective reservoirs based on the random forest classification model	Sistem multi-reservoir di lembah Sungai Huaihe di Cina.	Jieyu Li , Ping-an Zhong, Minzhi Yang, Feilin Zhu , Juan Chen, Weifeng Liu, Sunyu Xu, Journal of Hydrology, 2020	Random Forest	Metode yang diusulkan diterapkan pada sistem multi-reservoir di lembah Sungai Huaihe di Cina. Hasil penelitian menunjukkan bahwa model RFC memiliki karakteristik akurasi klasifikasi tinggi, sensitivitas rendah terhadap sampel banjir dan stabilitas. Ini menampilkan lebih banyak dominasi dalam identifikasi dinamis efektif reservoir dibandingkan
---	--	---	--	---------------	--

					dengan model lainnya.
10	Random Forest vs Logistic Regression: Binary Classification for Heterogeneous Datasets	Heterogeneous Datasets	Kaitlin Kirasich, Trace Smith, and Bivin Sadler, PhD, SMU Data Science, 2018	Random Forest dan Logistic Regression	Hasil penelitian adalah tingkat true positif untuk random forest lebih tinggi dari regresi logistik dan menghasilkan tingkat false positif yang lebih tinggi untuk dataset dengan peningkatan variabel noise. Setiap studi kasus terdiri dari 1000 simulasi dan kinerja model secara konsisten menunjukkan tingkat false positive untuk random forest dengan 100

					pohon menjadi statistic berbeda dari regresi logistic
--	--	--	--	--	--

2.2. LANDASAN TEORI

2.2.1. Data Mining

Data Mining merupakan proses untuk menemukan hubungan atau gambaran dari ratusan atau ribuan *field* dari sebuah relasional basis data yang besar. *Data Mining* juga sering disebut sebagai serangkaian proses untuk menggali nilai tambah berupa informasi yang selama ini belum diketahui. Informasi yang dihasilkan diperoleh dengan cara mengekstraksi dan mengenali pola yang penting atau menarik dari data yang terdapat pada basis data. *Data Mining* terutama digunakan untuk mencari pengetahuan yang terdapat dalam basis data yang besar sehingga sering disebut *Knowledge Discovery Databases*. (Hasan, 2017). Fungsi-fungsi dalam *data mining* terdapat 6 fungsi yaitu : (Masripah, 2015)

1. Fungsi deskripsi (*description*)

Fungsi deskripsi merupakan langkah menggambarkan sekumpulan data secara ringkas. Banyak cara yang dapat digunakan untuk memberikan gambaran secara ringkas bagi sekumpulan data yang besar jumlahnya dan banyak macamnya. Contoh dari penggambaran fungsi deksripsi yaitu Deskripsi Grafis, Deskripsi Lokasi, dan Deskripsi Keragaman.

2. Fungsi estimasi (*estimation*)

Fungsi estimasi diidentifikasi untuk membuat perkiraan suatu hal yang sudah memiliki data. Fungsi estimasi terdiri dari dua cara yaitu Estimasi Titik dan Estimasi Selang Kepercayaan.

3. Fungsi prediksi (*prediction*)

Fungsi prediksi adalah memperkirakan hasil dari hal yang belum diketahui, dan digunakan untuk mendapatkan hal baru yang akan muncul selanjutnya. Contoh prediksi seperti Regresi Linier.

4. Fungsi klasifikasi (*classification*)

Fungsi klasifikasi merupakan proses menggolongkan suatu data. Contoh algoritma klasifikasi : Algoritma *Mean Vector*, Algoritma *K-nearest Neighbor*, Algoritma ID3, Algoritma C4.5, dan Algoritma C5.0

5. Fungsi pengelompokan (*cluster*)

Fungsi pengelompokan merupakan sebuah proses pengelompokan data. Data yang dikelompokkan disebut objek atau catatan yang memiliki kemiripan atribut kemudian dikelompokkan pada kelompok yang berbeda. Contoh algoritma yang digunakan seperti : Algoritma *Hierarchical Clustering*, Algoritma *Partitional Clustering*, Algoritma *Single Linkage*, Algoritma *Complete Linkage*, Algoritma *Average Linkage*, Algoritma *K- Means*.

6. Fungsi asosiasi (*association*)

Fungsi asosiasi difungsikan untuk menemukan aturan asosiasi (*association rule*) yang berguna dalam mengidentifikasi item-item yang menjadi objek. Algoritma yang digunakan seperti algoritma *Generalized Association Rules*, *Quantitative Association Rule*, *Asynchronous Parallel Mining*.

2.2.2. Text Mining

Text mining merupakan bidang yang terdapat pada sistem temu balik informasi (information retrieval), data mining dan machine learning. Text mining merupakan proses penambangan data teks dimana sumber data biasanya didapatkan dari dokumen dan tujuannya adalah mencari kata - kata yang dapat mewakili isi dari dokumen sehingga dapat dilakukan analisa keterhubungan antar dokumen. Perbedaan mendasar antara text mining dan data mining terletak pada sumber data yang digunakan. Pada data mining, pola-pola diekstrak dari basis data yang terstruktur, sedangkan di text mining, pola-pola diekstrak dari data tekstual (natural language)(Soepomo, 2014).

2.2.3. Text Preprocessing

Text preprocessing merupakan salah satu komponen yang terdapat didalam text mining. Text preprocessing dilakukan untuk mengubah data tekstual yang tidak terstruktur ke dalam data yang terstruktur dan disimpan kedalam basis data. Tujuan dari preprocessing yaitu menghasilkan sebuah set term index yang dapat mewakili isi dokumen(Sholeh et al., 2018). Komponen text preprocessing yang akan digunakan pada penelitian ini dibagi menjadi beberapa tahapan, yaitu:

1. Case Folding

Case Folding merupakan proses pengubahan data menjadi format yang sesuai. Hal ini bertujuan mengurangi redudansi (duplikasi) data yang akan digunakan dalam proses pengklusteran sehingga proses perhitungan menjadi lebih optimal. Dengan kata lain case folding bisa diartikan sebagai proses mengubah seluruh huruf dari 'a' sampai dengan 'z' dalam dokumen menjadi huruf kecil.

2. Lemmatisasi

Lemmatisasi adalah teknik pada natural language processing yang digunakan untuk mengembalikan kata kepada kata dasarnya yang disesuaikan dengan kamus Bahasa Indonesia. Lemmatisasi digunakan pada kebutuhan yang berhubungan dengan text mining seperti information retrieval yang dilakukan pada tahap preprocessing. Metode lemmatisasi pada Bahasa Indonesia lebih dikenal dengan istilah stemming. Tahun 1996, Nazief sudah pernah membangun system stemming, namun masih terdapat beberapa kesalahan.

3. Stemming

Stemming adalah tahap mencari kata dasar dari tiap kata hasil filtering. Proses stemming secara luas sudah digunakan di dalam Information retrieval (pencarian informasi) untuk meningkatkan kualitas informasi yang akan didapatkan. Dengan dilakukannya proses stemming ini, setiap kata yang berimbuhan akan berubah menjadi kata dasar.

4. Slang Word Standardization

Slang adalah suatu bentuk bahasa dalam pemakaian umum, yang dibuat dengan adaptasi yang populer dan perluasan makna dari kata-kata yang ada dan dengan menyusun kata-kata baru tanpa memperhatikan standar-standar skolastik dan kaidah-kaidah linguisitik dalam pembentukan kata-kata yang pada umumnya terbatas pada kelompok-kelompok social atau kelompok usia tertentu. Slang Word Standardization adalah menyelaraskan kata slang yang telah berbaur pada masyarakat.

5. Stopword Removal

Stopword Removal adalah tahap pemilihan kata-kata penting dari hasil token, yaitu kata-kata yang bisa digunakan untuk mewakili isi dari sebuah dokumen. Proses filtering juga biasa disebut sebagai

stopword removal. Biasanya kata yang termasuk dalam stopwords contohnya adalah adalah, yang, dan, di, itu, dengan, untuk, tidak, dari, dalam, akan, pada, ini, juga, saya, serta, adalah, bahwa, lain, kamu, dan lain lain. Pada proses ini, terdapat dua teknik yang bisa dilakukan yaitu stop list dan word list. Stop list yaitu membuang kata yang tidak deskriptif atau tidak penting. Sedangkan word list yaitu menyimpang kata yang dianggap penting.

2.2.4. Pelabelan Data

Pelabelan data adalah tindakan memberi label atau memberi anotasi pada kumpulan data tidak terstruktur atau terstruktur yang berbeda untuk mengajari komputer mengidentifikasi perbedaan dan pola di antara mereka. Pada dasarnya pelabelan data dalam machine learning mengacu pada proses mendeteksi data yang tidak berlabel (seperti foto, file teks, video, dll.) dan menambahkan satu atau beberapa label yang relevan untuk menawarkan konteks sehingga model pembelajaran mesin dapat belajar dari dia. Pelabelan data sangat penting untuk sejumlah kasus penggunaan, termasuk pengenalan suara, visi komputer, dan pemrosesan bahasa alami.. (Jay, 2022)

2.2.5. TF – IDF

Term Frequency (TF) adalah frekuensi dari kemunculan sebuah term (kata/frasa) dalam dokumen yang bersangkutan. IDF atau Inverse Document Frequency merupakan dokumen yang mengandung term atau token atau kata (Sholeh et al., 2018). IDF berfungsi mengurangi bobot suatu term jika kemunculannya banyak terdapat di seluruh koleksi dokumen yang akan diolah. IDF merupakan sebuah pengurangan dominasi term yang sering muncul di berbagai dokumen yang diolah. Hal ini diperlukan karena term yang banyak muncul di berbagai dokumen dianggap sebagai term umum sehingga tidak penting nilainya.

Didalam Term Frequency, semakin sering frekuensi term yang muncul maka nilai semakin besar, berbeda dengan IDF yaitu semakin sedikit frekuensi kata muncul dalam dokumen, maka semakin besar nilai yang diperolehnya. Perhitungan TF-IDF yaitu mengkalikan term frequency (TF) sebagai penghitung frequency term dalam sebuah dokumen dengan inverse document frequency (IDF) sebagai nilai keinformatifan sebuah term.

Metode TF – IDF ini diperoleh dari penggabungan 2 konsep untuk perhitungan nilai bobot. Frekuensi kemunculan kata di dalam dokumen yang diberikan menunjukkan seberapa penting kata itu di dalam dokumen tersebut. Frekuensi dokumen yang mengandung kata tersebut menunjukkan seberapa umum kata tersebut. Sehingga bobot hubungan antara sebuah kata dan sebuah dokumen akan tinggi apabila frekuensi kata tersebut tinggi di dalam dokumen dan frekuensi keseluruhan dokumen yang mengandung kata tersebut yang rendah pada kumpulan dokumen.

2.2.6. Klasifikasi

Klasifikasi adalah sebuah proses menemukan model atau fungsi yang membedakan konsep atau kelas data dengan tujuan untuk memperkirakan kelas yang tidak diketahui dari suatu objek. Dalam klasifikasi terdapat dua proses, yakni proses *training* dan proses *testing*. Proses *training* menggunakan *training set* yang telah diketahui label-labelnya yang berfungsi untuk membangun model. *Testing* digunakan untuk menguji keakuratan model yang telah dibangun saat proses *training*. (Dicky Nofriansyah, S.Kom., 2017)

2.2.7. Random Forest

Machine Learning adalah salah satu dari metodologi ilmiah modern yang dapat melakukan prosedur otomatis untuk menghasilkan

prediksi pada suatu fenomena dengan melakukan pengamatan dari kejadian yang terjadi sebelumnya yaitu mencari pola pada suatu kumpulan data yang diberikan. Saat ini machine learning telah menjadi metode yang umum digunakan untuk menyelesaikan suatu tugas atau masalah dalam kehidupan sehari-hari yang membutuhkan proses ekstraksi sekumpulan data yang besar. Secara garis besar ada dua tipe machine learning, yaitu Supervised Learning dan Unsupervised Learning. Supervised learning mengacu pada machine learning dimana data yang digunakan untuk belajar sudah diberi label output yang harus dikeluarkan mesin, sedangkan Unsupervised learning sebaliknya mengacu pada machine learning yang belajar dari data yang tidak diberi label output (Shalev-Shwartz & Ben-David, 2013)

Random Forest adalah algoritma supervised learning yang dikeluarkan oleh Breiman pada tahun 2001 (Louppe, 2014). Random Forest biasa digunakan untuk menyelesaikan masalah yang berhubungan dengan klasifikasi, regresi, dan sebagainya. Ada dua hal yang membuat algoritma ini disebut random, yaitu:

1. Setiap pohon tumbuh pada sampel bootstrap yang berbeda diambil dari data latih secara acak.
2. Dalam setiap node split selama pembentukan decision tree, sebagian sampel dari m variabel dipilih dari kumpulan data yang asli dan kemudian yang terbaik akan digunakan dalam node tersebut.

Algoritma ini berupa kombinasi dari beberapa tree predictors atau bisa disebut decision trees dimana setiap tree bergantung pada nilai random vector yang dijadikan sampel secara bebas dan merata pada semua tree dalam forest tersebut. Hasil prediksi dari Random Forest didapatkan melalui hasil terbanyak dari setiap individual decision tree

(voting untuk klasifikasi dan rata-rata untuk regresi). Untuk RF yang terdiri dari N trees dirumuskan sebagai:

$$l(y) = \underset{c}{\operatorname{argmax}} \left(\sum_{n=1}^N l h_n(y) = c \right)$$

Dimana I adalah fungsi indikator dan h_n adalah tree ke- n dari RF (Liparas, 2014). Random Forest memiliki mekanisme internal yang menyediakan estimasi dari generalization error-nya sendiri yang disebut out-of-bag (OOB) error estimate. Dalam pembentukan tree hanya $2/3$ dari data asli yang digunakan dalam pengambilan sampel bootstrap. Sedangkan $1/3$ sisanya diklasifikasikan oleh tree yang terbentuk dan digunakan untuk menguji performanya. OOB error estimation adalah rata-rata dari kesalahan prediksi untuk setiap kasus training y menggunakan tree yang tidak mengikutsertakan y dalam sampel bootstrap-nya. Kemudian, saat RF dibuat, semua training cases menyusuri setiap pohon dan matriks kedekatan setiap kasus dihitung berdasarkan pasangan kasus yang sampai di terminal node yang sama (Liparas, 2014).

2.2.8. Confusion Matrix

Confusion matrix adalah tabel yang menyatakan klasifikasi jumlah data uji yang benar dan jumlah data uji yang salah. Contoh confusion matrix untuk klasifikasi biner.

		Kelas Prediksi	
		1	0
Kelas Sebenarnya	1	TP	FN
	0	FP	TN

Keterangan

TP (*True Positive*) = Jumlah dokumen dari kelas 1 yang benar
Diklasifikasikan sebagai kelas 1

TN (*True Negative*) = Jumlah dokumen dari kelas 0 yang benar
Diklasifikasikan sebagai kelas 0

FP (*False Positive*) = Jumlah dokumen dari kelas 0 yang salah
Diklasifikasikan sebagai kelas 1

FN (*False Negative*) = Jumlah dokumen dari kelas 1 yang salah
Diklasifikasikan sebagai kelas 0

Rumus confusion matrix untuk menghitung accuracy,
precision, dan recall seperti berikut.

$$Accuracy = \frac{TP + TN}{Total}$$

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

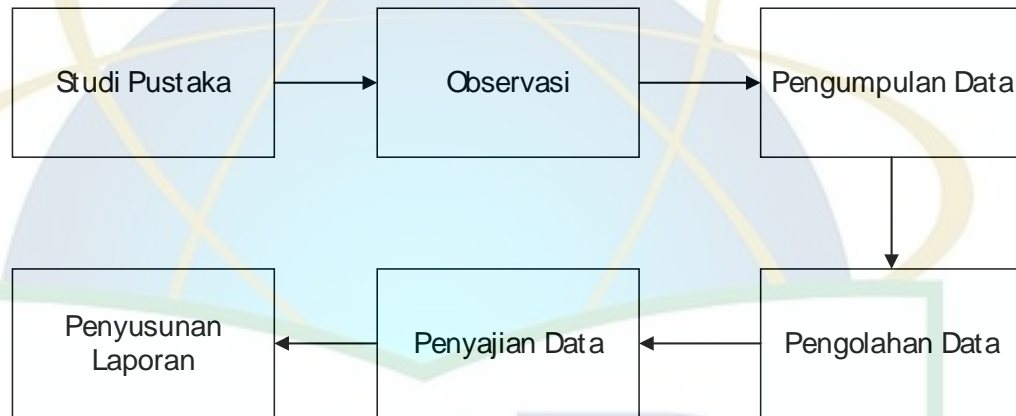
$$Recall = \frac{TP}{TP + FN}$$

BAB III

METODE PENELITIAN

3.1. Alur Penelitian

Adapun alur dalam penelitian ini ditunjukkan pada Gambar 1.



Gambar 1. Alur Penelitian

Penjelasan masing-masing tahapan :

1. Studi Pustaka

Proses studi pustaka dilakukan untuk mencari referensi terkait penelitian yang pernah dilakukan sebelumnya, dan juga rujukan-rujukan dalam bentuk buku dan informasi-informasi mengenai kasus terkait. Hal ini dijadikan landasan untuk menemukan permasalahan yang akan diselesaikan dalam penelitian ini.

2. Observasi

Observasi dilakukan dengan mengamati dari hasil dataset yang digunakan. Pengamatan dilakukan dengan memperhatikan intensitas interaksi yang

terjadi dan akan dilakukan penelitian lebih jauh dengan melakukan klasifikasi.

3. Pengumpulan Data

Pengumpulan data yang dilakukan adalah dengan mengambil *dataset* komplain perusahaan,. Pada studi kasus kali ini, diambil dari dataset ulasan review Google Playstore.

4. Pengolahan Data

Pengolahan data dilakukan setelah proses pengambilan data di sosial media berhasil dilakukan. Data yang digunakan, selanjutnya perlu dilakukan pengolahan data, agar hasil data mentah yang dimiliki dapat di proses ke langkah berikutnya yaitu interpretasi data. Pada proses pengolahan data sendiri dilakukan proses pembersihan data, agar data yang diolah benar-benar data bersih bukan data mentah (data kotor).

5. Penyajian Data

Penyajian data dilakukan untuk menampilkan hasil pengolahan data. Data yang sudah diolah dengan data *cleaning*, *preprocessing* bisa dilakukan penyajian dalam bentuk informasi yang baru dan berguna. Penyajian data membutuhkan bantuan algoritma untuk melakukan klasifikasi sentiment yang terbentuk.

6. Penyusunan Laporan

Setelah semua tahapan penelitian dilakukan, yang terakhir adalah menyusunnya ke dalam laporan penelitian. Laporan ini berisikan tahapan dari awal pengumpulan, pengolahan, penyajian data hingga penarikan kesimpulan dari penelitian yang dilakukan.

3.2. Metode Usulan

Metodologi penelitian didefinisikan sebagai kerangka atau model yang mengandung prinsip-prinsip teoritis mengenai petunjuk atau bagaimana memilih metode yang tepat dalam melaksanakan penelitian. Adapun metode yang dipilih

penulis dalam proses penelitian ini yaitu menggunakan metode KDD. Pemilihan KDD dibandingkan dengan metode pendekatan yang lain adalah keseusaian dengan tahapan yang akan dijalani pada proses asosiasi ini. Dibandingkan dengan Crisp DM dan SEMMA, pendekatan KDD lebih tepat dan sesuai dengan studi kasus. Proses Data Mining berdasarkan KDD terdiri dari lima tahap dalam proses ini, yaitu:

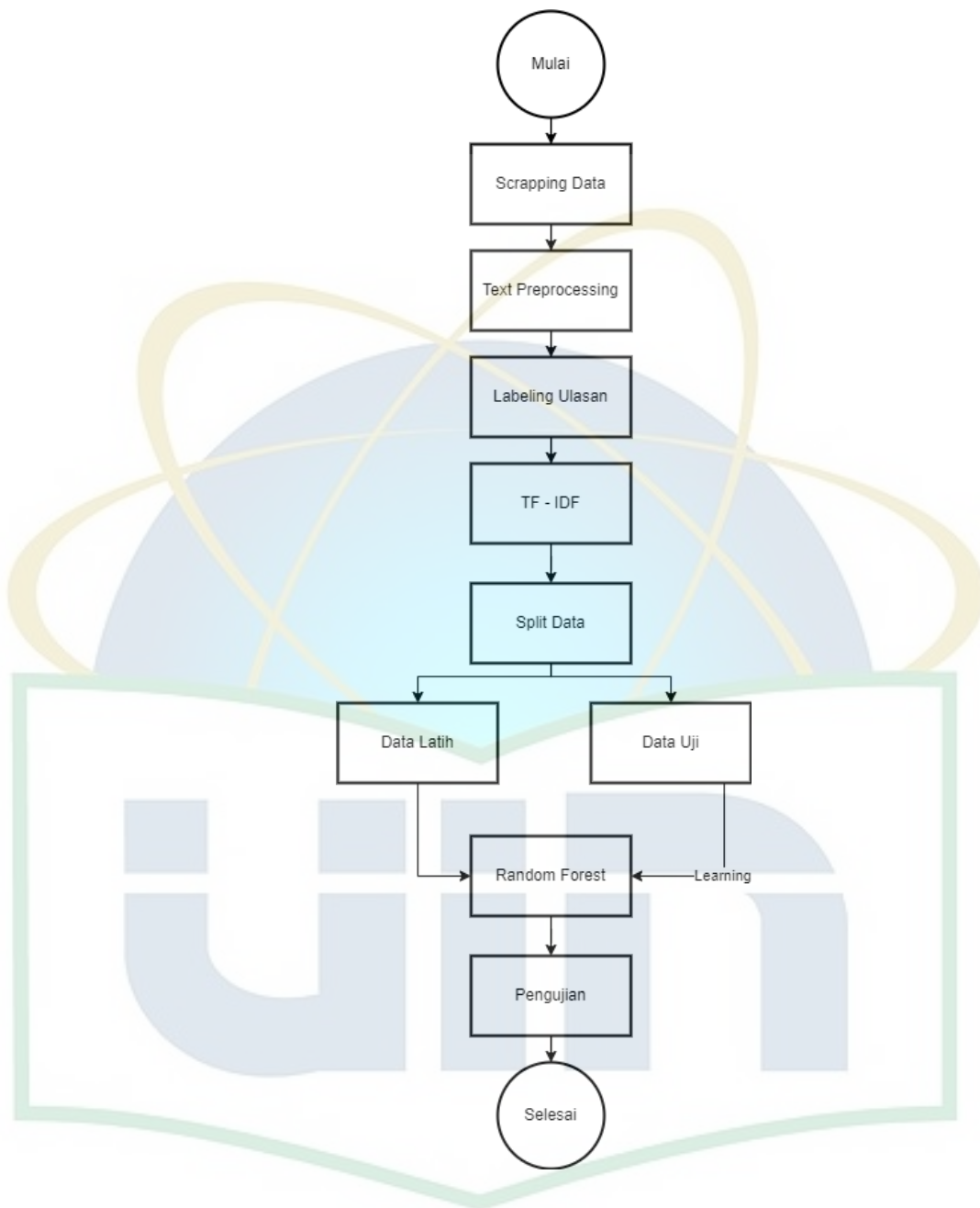
1. Selection: Membuat sebuah target data, fokus dalam bagian dari variabel atau sampel data yang mana discovery akan dilakukan.
2. Preprocessing: Cleaning target data dengan tujuan mendapatkan data yang konsisten
3. Transformation: Transformasi data menggunakan reduksi dimensional atau metode transformasi
4. Data Mining: Mencari pola menarik di dalam sebuah bentuk tertentu, bergantung dari tujuan data mining (biasanya prediksi)
5. Interpretation/Evaluation: Interpretasi dan evaluasi dari pola yang sudah dimining.

3.3. Pengujian

Data yang sudah diolah dan dilakukan klasifikasi, pada tahap selanjutnya akan di lakukan pengujian. Pengujian merupakan tahapan dimana data akan diuji. Tahap pengujian diperlukan sebagai ukuran bahwa algoritma dapat dijalankan sesuai dengan tujuan. Pengujian dilakukan dengan model pengujian akurasi dimana akan dicari nilai dari pengujian klasifikasi mengenai nilai F-Measure, Precision, dan Recall.

3.4. Diagram Alir Penelitian

Tahapan/alur penelitian dalam melakukan klasifikasi data dengan metode Random Forest adalah sebagai berikut:



Adapun penjelasan lanjutan pada Gambar 3.2 dalam penelitian ini dapat dilihat pada point berikut ini.

a. Scrapping Data

Data scraping atau yang juga sering disebut data extraction merupakan teknik atau metode otomatisasi yang memungkinkan seseorang untuk mengekstrak data dari sebuah website, database, aplikasi enterprise, atau sistem legacy yang kemudian dapat menyimpannya ke dalam sebuah file dengan format tabular atau spreadsheet.

b. Text Preprocessing

Text preprocessing merupakan salah satu komponen yang terdapat didalam text mining. Text preprocessing dilakukan untuk mengubah data tekstual yang tidak terstruktur ke dalam data yang terstruktur dan disimpan kedalam basis data. Tujuan dari preprocessing yaitu menghasilkan sebuah set term index yang dapat mewakili isi dokumen. Pada penelitian ini, terdapat beberapa tahapan text preprocessing yang digunakan yaitu casefolding, lematisasi, stemming, slang word standardization, stopwords removal, unwanted word removal.

c. Labeling Ulasan

Labeling dataset pada penelitian ini dilakukan pada tahapan labeling ulasan. Dimana pada dataset belum terdapat label positif dan negatif pada ulasan, sehingga perlu dilakukan proses pelabelan. Pada tahap ini penulis melakukan pelabelan dengan membuat dictionary kosa kata yang berkonotasi positif dan dictionary kosa kata yang berkonotasi negatif

d. TF – IDF

Pembobotan kata pada penelitian ini menggunakan TF – IDF (Term Frequency – Inverse Document Frequency). Dimana bobot yang diperoleh berdasarkan frekuensi yang muncul pada tiap ulasan.

e. Split Data

Split data pada penelitian ini berfungsi untuk membagi data menjadi kepada 2 bagian data yaitu data training dan data testing.

f. Random Forest

Tahapan-tahapan dalam melakukan klasifikasi dengan random forest adalah sebagai berikut:

- 1) Buat suatu bootstrap sample atau pengambilan sampel dengan replacement (pengembalian) dari suatu ukuran dari gugus data.
- 2) Pilih m (m_{try}) variabel secara random dari p variable $m \leq p$
- 3) Setelah dilakukan pemilihan m secara random, maka pohon ditumbuhkan tanpa pruning (pemangkasan).
- 4) Langkah 1-3 dilakukan sebanyak kali hingga terbentuk suatu forest sebanyak pohon.
- 5) Proses penentuan suatu kelas dilakukan dengan majority vote.
- 6) Setelah terbentuk forest, kemudian dicari parameter m_{try} yang optimal sehingga diperoleh nilai misklasifikasi error (Out of Bag Error) yang stabil dan tingkat kepentingan variabel (Variable Importance).
- 7) Setelah diperoleh nilai m_{try} optimal kemudian dilakukan prediksi dengan data testing.

g. Pengujian

Tahapan pengujian pada penelitian ini yaitu menggunakan pengujian confusion matrix.

h. Selesai

BAB IV

IMPLEMENTASI

Pada bab ini dijelaskan hasil dan pembahasan sistem yang telah dibangun. Untuk itu, dilakukan pengujian terhadap sistem, dengan cara membuat proses *scrapping*, proses *preprocessing*, proses klasifikasi dengan menggunakan metode Random Forest, proses perhitungan akurasi sistem, dan proses visualisasi data. Subjek pada penelitian Analisis Sentimen ini adalah ulasan penggunaan aplikasi Kereta Api Indonesia (KAI) pada Google Playstore. Semua proses pada penelitian ini menggunakan Bahasa pemrograman Python pada executable document Google Colaboratory.

Implementasi metode ini dimulai dengan melakukan pengumpulan data ulasan yang berupa komentar pada Google Playstore, melakukan proses sentiment dengan erdasarkan word dictionary yang telah kita bentuk pada Github, menghitung bobot kata dengan metode TF-IDF, melakukan klasifikasi dan kemudian menguji model klasifikasi pada data uji.

4.1. Web Scraping

Proses Web Scraping dilakukan dengan menggunakan Bahasa Pemrograman Python dengan melalui 5 tahapan, yaitu:

4.1.1. Install Google Play Scraper Package

Sebelum melakukan proses scraping, dilakukan penginstalan package google play scraper.

```
[ ] !pip install google-play-scraper
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/  
Collecting google-play-scraper  
  Downloading google_play_scraper-1.2.2-py3-none-any.whl (28 kB)  
Installing collected packages: google-play-scraper  
Successfully installed google-play-scraper-1.2.2
```


Gambar 4.1 Package Google Play Scraper

4.1.2. Install Library yang Dibutuhkan

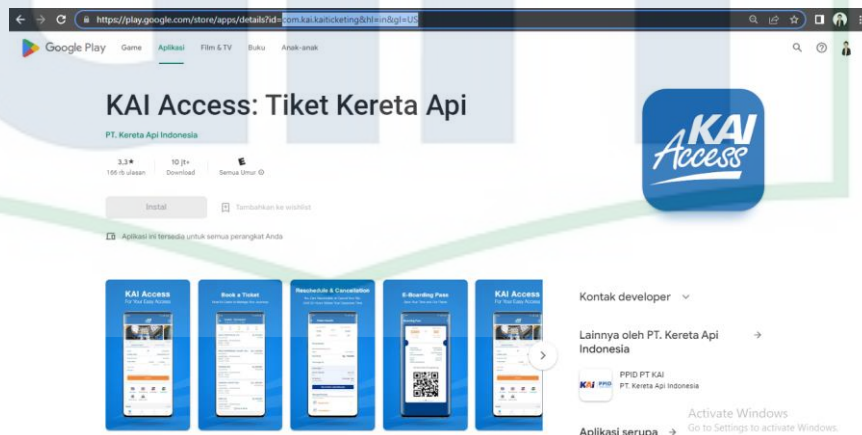
Melakukan import 3 library yang dibutuhkan untuk proses scrapping. Seperti library app untuk mengambil beberapa feature, yaitu “reviewId”, “username”, “userimage”, “content/ulasan”, “score”, dan masih banyak lainnya.

```
[ ] from google_play_scraper import app  
  
import pandas as pd  
  
import numpy as np
```

Gambar 4.2 Library untuk scraping

4.1.3. Membuka ID Aplikasi pada Google Playstore

Kemudian buka aplikasi yang ingin di scrapping pada web google play store. Kemudian ID aplikasi dari google playstore diambil dan dimasukkan pada code yang telah dibuat.



Gambar 4.3 Tampilan ID google play store

4.1.4. Scraping Ulasan

Setelah ID aplikasi pada google playstore di copy, kemudian ID dan jumlah ulasan yang diinginkan dimasukkan ke dalam code dan di jalankan untuk mengeksekusi perintah scrapping.

```
[ ] #Scrape dengan jumlah yang diinginkan
#Run kode ini jika ingin scrape data dengan jumlah tertentu.
#Ganti (misal, ingin scrape sejumlah 1000, maka ganti kode , count = 1000 )

from google_play_scraper import Sort, reviews

result, continuation_token = reviews(
    'com.kai.kaiticketing',
    lang='id', # defaults to 'en'
    country='id', # defaults to 'us'
    sort=Sort.MOST_RELEVANT, # defaults to Sort.MOST_RELEVANT you can use Sort.NEWEST to get newst reviews
    count=5000, # defaults to 100
    filter_score_with=None # defaults to None(means all score) Use 1 or 2 or 3 or 4 or 5 to select certain score
)
```

Gambar 4.4 Proses scrapping ulasan

4.1.5. Membuat Hasil Scraping Menjadi Dataframe

Hasil scrapping kemudian diubah menjadi dataframe dengan menggunakan library numpy dan pandas.

```
df = pd.DataFrame(np.array(result),columns=['review'])
df = df.join(pd.DataFrame(df.pop('review').tolist()))
df.head()
```

Gambar 4.5 Mengubah ulasan menjadi dataframe

Berikut hasil data yang telah diubah dalam bentuk dataframe

	reviewId	userName	userImage	content	score	thumbsUpCount	reviewCreatedVersion	at	replyContent	repliedAt
0	7d3ebf79-9336-48ad-8c9f-91f98e21c045	Riswana Adje Prasetyo	https://play-https://play-	Kecwa dengan aplikasi, Aplikasinya menjadi le...	1	141	2022-08-20	11:08:36	None	NaT
1	dc7b86b5-63d9-4c07-9962-30ce9f1388b	alvina prima san	https://play-https://play-	KAI sekarang sangat amat mengecewakan, Lemot, ...	2	152	2022-08-05	09:28:24	None	NaT
2	d52f726d-ce4e-4343-8459-879610654b70	Nadya wulandari	https://play-https://play-	Tolong apk nya di perbaiki. Entah kenapa seta...	2	147	2022-08-07	15:53:23	None	NaT
3	9ca78181-e70f-40b5-45ad-af6a20190203	Muh. Abdurrohman Sutrisno	https://play-https://play-	Gak jelas, setiap tahapnya pasti macet dan kel...	1	149	2022-08-07	01:00:58	None	NaT
4	63291a57-630e-4893-a30c-f9eb99d610c5	Abdan Syakura	https://play-https://play-	Kepada pihak IT PT KAI, mohon aplikasinya dipe...	1	3	2022-09-09	22:40:28	None	NaT

Gambar 4.6 Hasil dataframe

4.1.6. Menyimpan Hasil Scraping dalam Bentuk CSV

Setelah diubah menjadi dataframe kemudian data di ubah dan simpan dalam bentuk CSV agar dapat kita gunakan sebagai data dalam proses klasifikasi menggunakan metode Random Forest.

```
[ ] my_df.to_csv("scrapped_data.csv", index = False) #Save the file as CSV , to download: click the folder icon on the left.
```

Gambar 4.7 Menyimpan data dalam bentuk CSV

4.2. Text Preprocessing

Proses preprocessing pada NLP adalah proses yang paling penting, dikarenakan pada data text terdapat sangat banyak noise yang dapat menyebabkan proses klasifikasi menghasilkan akurasi yang buruk. Salah satu contoh noise yang banyak kita temuka adalah penggunaan kata singkatan atau tidak terdapat dalam Kamus Besar Bahasa Indonesia (KBBI). Maka dari itu proses preprocessing memiliki tahapan yang cukup panjang.

4.2.1. Membuka dataset

Dataset sebelumnya yang telah di dapatkan dari hasil scrapping kemudian di baca pada code baru

```
df = pd.read_csv('/content/scrapped_data_BBI.csv')
```

	userName	score	at	content
0	Fahmi Ulum	1	2022-08-29 13:14:55	Masa Filur pindah tempat duduk aja gak ada ...
1	Fariz Ridwan	1	2022-08-29 12:16:04	Perlu perbaikan software karena sering nge bug...
2	Dian Karunia putri	1	2022-08-29 11:38:17	Kenapa apk nya tiba' sering log out sendiri? U...
3	Mega Revangga	2	2022-08-29 11:25:11	Kenapa hanya pemesanan jarak jauh saja yang su...
4	Muhamad Yanto	1	2022-08-29 09:59:59	Ini kenapa ga bisa ubah/edt penumpang.??! Def...
...
4995	Pengguna Google	1	2018-09-23 23:19:47	Dari dulu sampai sekarang masih belum bisa nge...
4996	Pengguna Google	1	2018-09-23 23:04:08	mohon jika terjadi update aplikasi versi terba...
4997	Pengguna Google	1	2018-09-22 21:54:33	Makin parah aja setelah di update. Mau ilat ja...
4998	Pengguna Google	2	2018-09-16 10:49:44	Akhir2 ini saya booking lewat aplikasi ini ker...
4999	Pengguna Google	3	2018-09-16 05:45:00	PT. KAI yg terhormat, tolong aplikasi ini di...

5000 rows x 4 columns

Gambar 4.8 Membuka dataset CSV

4.2.2. Memilih Variabel yang Penting

Feature paling penting dalam dataset ini adalah feature “content” yang merupakan ulasan atau komentar para pengguna aplikasi KAI. Feature inilah yang akan kita ambil dan melakukan preprocessing pada feature ini.

```
[ ] df['content']
0      Masa Fitur pindah tempat duduk aja gak ada .. ...
1      Perlu perbaikan software karena sering nge bug...
2      Kenapa apk nya tiba-tiba sering log out sendiri? U...
3      Kenapa hanya pemesanan jarak jauh saja yang su...
4      Ini kenapa ga bisa ubah/edit penumpang.??? Def...
...
4995   Dari dulu sampai sekarang masih belum bisa nge...
4996   mohon jika terjadi update aplikasi versi terba...
4997   Makin parah aja setelah di update. Mau liat ja...
4998   Akhir2 ini saya booking lewat aplikasi ini ker...
4999   PT. KAI yg terhormat, tolong aplikasi ini diti...
Name: content, Length: 5000, dtype: object
```

Gambar 4.9 Memilih feature yang akan di preprocessing

4.2.3. Casefolding

Pada proses casefolding ini seluruh ulasan yang berjumlah 5000 ulasan akan di normalisasi yaitu dengan mengubah seluruh huruf besar yang terdapat pada ulasan menjadi huruf kecil.

```
[ ] import re, string, unicodedata
def CaseFolding(text):
    # Hapus non-ascii
    text = unicodedata.normalize('NFKD', text).encode('ascii', 'ignore').decode('utf-8', 'ignore')

    # Menghapus Tanda Baca
    text = re.sub(r'[^\w]', ' ', text)

    # Menghapus Angka
    text = re.sub(r'[0-9]', '', text).strip()
    text = re.sub(r'[^\d]', ' ', text)

    # Mengubah text menjadi lowercase
    text = text.lower()

    # Menghapus white space
    text = re.sub('[\s]+', ' ', text)

    return text
```

Gambar 4.10 Code proses casefolding

Dari proses casefolding didapatkan hasil sebagai berikut. Dibawah ini ditampilkan 5 ulasan teratas yang telah diterapkan proses casefolding. Seperti kalimat “**M**asa fitur pindah tempat duduk aja gak ada...” akan berubah menjadi “**m**asa fitur tempat aja nggak ada...”

```
[ ] df['Text_Clean'] = df['Text_Clean'].apply(Case_Folding)
df['Text_Clean'].head()

0    masa fitur pindah tempat duduk aja gak ada anehh
1    perlu perbaikan software karena sering nge bug...
2    kenapa apk nya sering log out sendiri udah mas...
3    kenapa hanya pemesanan jarak jauh saja yang su...
4    ini kenapa ga bisa ubah edit penumpang default...
Name: Text_Clean, dtype: object
```

Gambar 4.11 Hasil proses casefolding

4.2.4. Lemmatisasi

Pada tahap ini seluruh ulasan yang menggunakan kata dengan tambahan imbuhan akan diubah menjadi kata dasar pada Kamus Besar Bahasa Indonesia. Seperti pada hasil lemmatisasi yang ditunjukkan pada gambar dibawah, yaitu kalimat “perlu **perbaikan** software karena sering nge bug...” akan di lemmatisasi sehingga menjadi perlu **baik** software karena sering nge bug...”.

```
[ ] from nlp_id.lemmatizer import Lemmatizer
lemmatizer = Lemmatizer()

[ ] df['Text_Clean'] = df['Text_Clean'].apply(lemmatizer.lemmatize)
df['Text_Clean'].head()

0    masa fitur pindah tempat duduk aja gak ada anehh
1    perlu baik software karena sering nge bug perl...
2    kenapa apk nya sering log out sendiri udah mas...
3    kenapa hanya pesan jarak jauh saja yang super ...
4    ini kenapa ga bisa ubah edit tumpang default n...
Name: Text_Clean, dtype: object
```

Gambar 4.12 Proses dan hasil Lemmatisasi

4.2.5. Stemming

Proses stemming hampir sama dengan proses lemmatisasi sebelumnya, hanya saja proses stemming tidak memperhatikan analisis morfologis kata dari suatu kalimat.

```
[ ] from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

# Membuat stemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()

[ ] df['Text_Clean'] = df['Text_Clean'].apply(stemmer.stem)
df['Text_Clean'].head()

0    masa fitur pindah tempat duduk aja gak ada anehh
1    perlu baik software karena sering nge bug perl...
2    kenapa apk nya sering log out sendiri udah mas...
3    kenapa hanya pesan jarak jauh saja yang super ...
4    ini kenapa ga bisa ubah edit tumpang default n...
Name: Text_Clean, dtype: object
```

Gambar 4.13 Proses Stemming

4.2.6. Slang Word Standardization

Pada proses ini seluruh ulasan yang menggunakan kosa kata tidak baku atau kata gaul ‘slang word’. Sebelumnya penulis atau peneliti perlu membuat dictionary yang berisi kosa kata tidak baku yang banyak digunakan oleh banyak generasi sekarang. Kata-kata yang tidak baku ini kemudian akan di berikan kata baku atau kata aslinya. Sehingga ketika kata baku ditemukan pada ulasan, kata tersebut akan langsung kita ganti dengan kata asli yang sesuai dengan KBBI. Berikut dictionary slang word yang kita jalankan pada code.

```
[ ] slang_dictionary = pd.read_csv("https://raw.githubusercontent.com/nasalsabila/kamus-slang/master/colloquial-indonesian-lexicon.csv")
slang_dict = pd.Series(slang_dictionary['formal'].values, index=slang_dictionary['slang']).to_dict()
```

slang_dictionary

	slang	formal	In-dictionary	context	category1	category2	category3
0	woww	wow	1	wow	elongasi	0	0
1	aminn	amin	1	Selamat ulang tahun kakak tulus semoga panjang...	elongasi	0	0
2	met	selamat	1	Met hari netaas kaki? Wish you all the best @t...	abreviasi	0	0
3	netaas	menetas	1	Met hari netaas kaki? Wish you all the best @t...	afiksasi	elongasi	0
4	keberpa	keberapa	0	Birthday yg keberpa kak?	abreviasi	0	0
...
15001	gataunya	enggak taunya	0	Ini kaya nenek2 ya beb gataunya agnezz @yugime...	akronim	0	0
15002	gtau	enggak tau	0	Stidaknya mrka may berkarya Dan berusaha yg tr...	akronim	abreviasi	0
15003	gatau	enggak tau	0	Ih gatau malu	akronim	0	0
15004	fans2	fan-fan	0	Jkt48 adalah tempat di mana sesama fans saling...	reduplikasi	naturalisasi	0
15005	gaharus	enggak harus	0	belajar tuh bisa dimana aja gaharus belajar di...	akronim	0	0

15006 rows x 7 columns

Gambar 4.14 Membuka dictionary slang word

Kemudian data tiap ulasan akan di cek dan akan diubah berdasarkan dictionary yang telah diterapkan. Pada gambar 4.15 dibawah dapat dilihat perubahan kalimat “masa fitur pindah tempat duduk aja **gak** ada...” menjadi “masa fitur pindah tempat duduk aja **enggak** ada...”


```
[ ] def Slangwords(text):
    for word in text.split():
        if word in slang_dict.keys():
            text = text.replace(word, slang_dict[word])
    return text

[ ] df['Text_Clean'] = df['Text_Clean'].apply(Slangwords)
df['Text_Clean'] = df['Text_Clean'].str.replace('mhs', 'mahasiswa')
df['Text_Clean'].head()

0    masa fitur pindah tempat duduk saja enggak ada...
1    perlu baik software karena sering nge bug perl...
2    kenapa apk nya sering log out sendiri sudah me...
3    kenapa hanya pesan jarak jauh saja yang super ...
4    ini kenapa enggakk bisa ubah edit tumpang d...
Name: Text_Clean, dtype: object
```

Gambar 4.15 Proses Stemming

4.2.7. Stopword Removal

Tahap stopwords removal ini menghapus seluruh kata yang dianggap tidak penting, yang mana tidak akan mempengaruhi sentiment pada kalimat. Kata yang dianggap tidak penting disini adalah kata sambung, seperti di, ke, ini, dan, dari. Kata-kata sambung ini akan dihapus sehingga kalimat ulasan akan menjadi padat. Contoh perubahan kalimat dapat dilihat pada Gambar 4.16 di bawah, salah satu contohnya adalah kalimat “masa fitur pindah tempat duduk aja enggak ada...” akan berubah menjadi “fitur pindah duduk enggak anehh”

```
[ ] from nltk.corpus import stopwords
stopword = stopwords

df['Text_Clean'] = df['Text_Clean'].apply(stopword.remove_stopword)
df['Text_Clean'].head()

0    fitur pindah duduk enggak anehh
1    software nge bug cepat nyaman
2    apk log out masuki akun enggak masuk susah
3    pesan jarak super super lot lokal lancar tol...
4    enggakk ubah edit tumpang default data prof...
Name: Text_Clean, dtype: object
```

Gambar 4.16 Proses stopwords removal

4.2.8. Unwanted Word Removal

Tahap ini penulis membuat dictionary yang berisikan kata-kata yang terdapat pada ulasan seperti nama bulan, dan tanda baca. Kemudian ulasan satu persatu akan di periksa jika terdapat kata yang terdapat dalam dictionary yang telah dibuat, maka kata dalam ulasan tersebut akan

dihilangkan. Library yang dibutuhkan pada proses ini adalah library nltk yang merupakan rangkaian perpustakaan dan program untuk pemrosesan bahasa alami simbolis dan statistik untuk bahasa Inggris yang ditulis dalam bahasa pemrograman Python.

```
1 unwanted_words = ['jan', 'feb', 'mar', 'apr', 'may', 'jun', 'jul', 'aug', 'sep', 'oct', 'nov', 'dec', 'uaddown', 'wearquad', 'lam', 'https', 'igshid']
import nltk
from nltk import word_tokenize, sent_tokenize
nltk.download('punkt')

def RemoveUnwantedWords(text):
    word_tokens = word_tokenize(text)
    filtered_sentence = [word for word in word_tokens if not word in unwanted_words]
    return ' '.join(filtered_sentence)

[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Unzipping tokenizers/punkt.zip.

[ ] df['Text_Clean'] = df['Text_Clean'].apply(RemoveUnwantedWords)
df['Text_Clean'].head()

0      fitur pindah duduk enggak anehh
1      software nge bug cepat nyaman
2      apk log out memasuki akun enggak masuk susah
3      pesan jarak super super lot lokal lancar tol...
4      engenggakk ubah edit tumpang default data prof...
Name: Text_Clean, dtype: object
```

Gambar 4.17 Proses menghilangkan kata yang tidak diinginkan

4.2.9. Menghapus Kata yang Kurang dari 3 Huruf

Selanjutnya kata-kata yang membingungkan karena berupa singkatan yang kurang dari tiga huruf dihapus juga dihapus agar kalimat ulasan semakin padat dan menyisakan kata-kata yang penting dalam proses analisis sentiment.

```
df['Text_Clean'] = df['Text_Clean'].str.findall('\w{3,}').str.join(' ')
df['Text_Clean'].head()

0      fitur pindah duduk enggak anehh
1      software nge bug cepat nyaman
2      apk log out memasuki akun enggak masuk susah
3      pesan jarak super super lot lokal lancar tol...
4      engenggakk ubah edit tumpang default data prof...
Name: Text_Clean, dtype: object
```

Gambar 4.18 Proses menghapus kata yang kurang dari 3 huruf

4.2.10. Split Word

Setelah kalimat ulasan bersih dan padat, selanjutnya kalimat ini akan di split atau dipisah-pisah per kata untuk masing-masing ulasan.

Proses ini juga akan memudahkan dalam proses perhitungan TF-IDF tiap kata pada proses selanjutnya.

```
def split_word(teks):
    list_teks = []
    for txt in teks.split(" "):
        list_teks.append(txt)
    return list_teks

[ ] df['Text_Clean_split'] = df['Text_Clean'].apply(split_word)
df['Text_Clean_split'].head()

0      [fitur, pindah, duduk, enggak, anehh]
1      [software, nge, bug, cepat, nyaman]
2      [apk, log, out, memasuki, akun, enggak, masuk,...]
3      [pesan, jarak, super, super, lot, lokal, lanca...]
4      [enggakk, ubah, edit, tumpang, default, dat...]
Name: Text_Clean_split, dtype: object
```

Gambar 4.19 Proses memisahkan kata-kata pada ulasan

4.3. Memberi Label pada Data Ulasan

Pada dataset belum terdapat label positif dan negatif pada ulasan, sehingga perlu dilakukan proses pelabelan. Pada tahap ini penulis melakukan pelabelan dengan membuat dictionary kosa kata yang berkonotasi positif dan dictionary kosa kata yang berkonotasi negatif.

- Dictionary kata positif :
<https://raw.githubusercontent.com/masdevid/ID-OpinionWords/master/positive.txt>
- Dictionary kata negatif :
<https://raw.githubusercontent.com/masdevid/ID-OpinionWords/master/negative.txt>

Adapun isi dari dictionary kata positif dan negatif dapat dilihat pada gambar dibawah ini



0	acungan jempol	0	absurd
1	adaptif	1	acak
2	adil	2	acak-acakan
3	afinitas	3	acuh
4	afirmasi	4	acuh tak acuh
5	agilely	5	adiktif
6	agung	6	adil
7	ahli	7	agresi
8	ahlinya	8	agresif
9	ajaib	9	agresor
10	aklamasi	10	aib

Gambar 4.20 Dictionary kata positif (kiri) dan dictionary kata negatif (kanan)

Dictionary yang telah disiapkan kemudian digunakan untuk melabeli setiap ulasan. Proses yang dilakukan adalah ketika terdapat kata positif yang lebih banyak dibanding kata negatif dalam suatu ulasan maka ulasan tersebut dilabeli dengan label positif, dan begitu juga sebaliknya jika ulasan berisikan kata negatif yang lebih banyak dibandingkan kata positif, maka ulasan akan di beri label negatif. Proses ini dilakukan untuk semua ulasan sejumlah 5000 ulasan, yang kemudian label ini diterapkan pada dataset dengan memberikan feature baru.

```
[ ] # Daftar Kata-kata Positif Bahasa Indonesia
df_positive = pd.read_csv('https://raw.githubusercontent.com/masdevid/ID-OpinionWords/master/positive.txt', sep='\t')
list_positive = list(df_positive.iloc[:,0])
# Daftar Kata-kata Negatif Bahasa Indonesia
df_negative = pd.read_csv('https://raw.githubusercontent.com/masdevid/ID-OpinionWords/master/negative.txt', sep='\t')
list_negative = list(df_negative.iloc[:,0])

# Menghitung kata-kata positif/negatif pada data teks dan menentukan sentimennya
def sentiment_analysis_lexicon_indonesia(text):
    score = 0
    for word in text:
        if (word in list_positive):
            score += 1
    for word in text:
        if (word in list_negative):
            score -= 1
    polarity=''
    if (score > 0):
        polarity = 'positive'
    elif (score < 0):
        polarity = 'negative'
    else:
        polarity = 'neutral'
    return score, polarity
```

Gambar 4.21 Proses pelabelan ulasan

Hasil pelabelan kemudian simpan dan digabungkan dalam dataframe dengan nama feature “polarity”. Dapat dilihat pada proses pelabelan ini didapatkan ulasan positif sebanyak 2970 dan ulasan negatif sebanyak 2030.

```
hasil = df['Text_Clean_split'].apply(sentiment_analysis_lexicon_indonesia)
hasil = list(zip(*hasil))
df['polarity_score'] = hasil[0]
df['polarity'] = hasil[1]
df.polarity.value_counts()
```

positive	2970
negative	2030

Name: polarity, dtype: int64

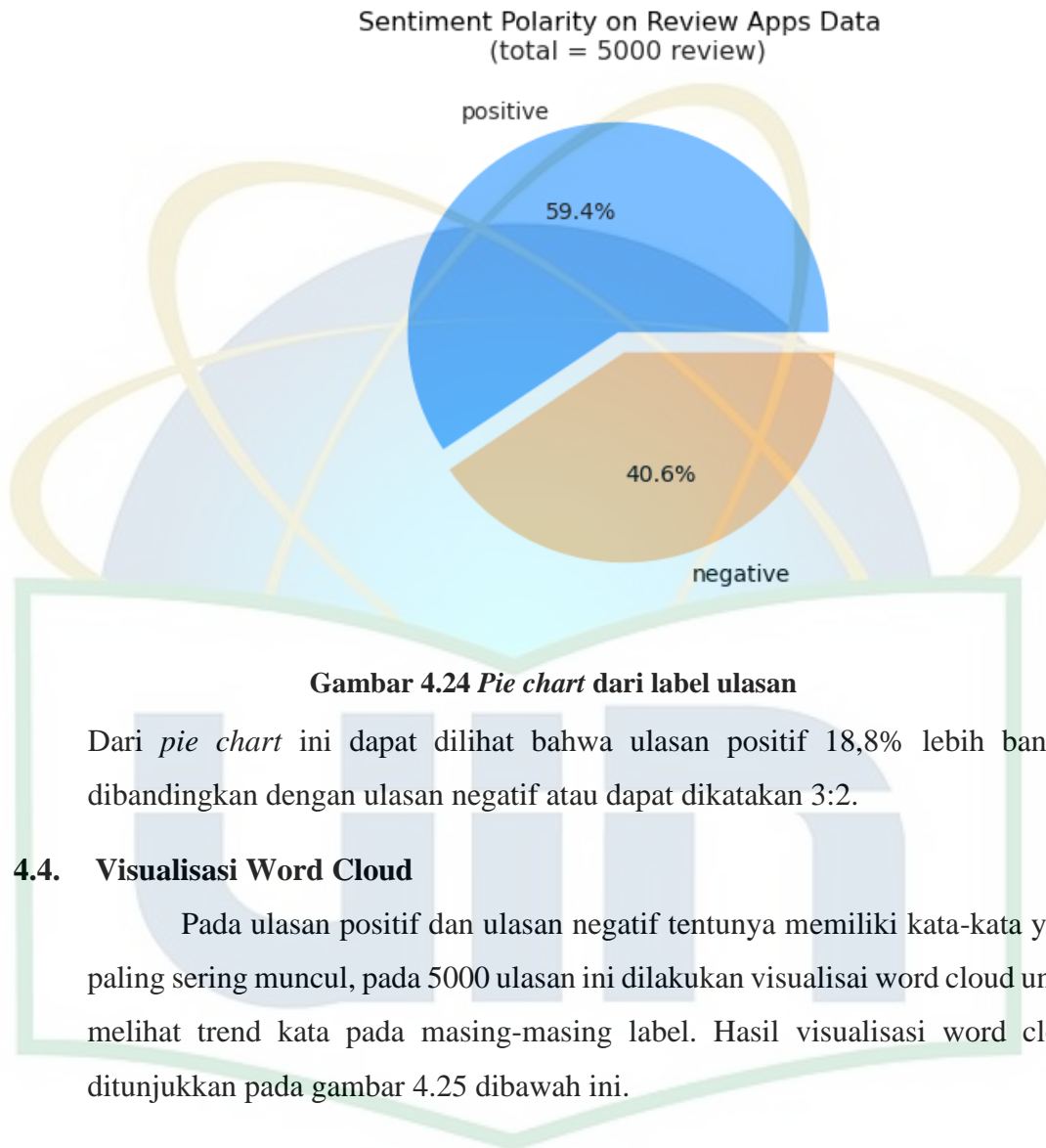
Gambar 4.22 Menyimpan label dalam dataframe

Selanjutnya untuk memperjelas perbandingan antara jumlah label positif dan label negatif maka dapat divisualisasikan dalam bentuk *pie chart*. Visualisasi perbandingan label positif dan negatif dapat dilihat pada gambar 4.23.

```
fig, ax = plt.subplots(figsize = (6, 6))
sizes = [count for count in df['polarity'].value_counts()]
labels = list(df['polarity'].value_counts().index)
explode = (0.1, 0, 0)
colors = ['#66b3ff', '#ffcc99', '#ff9999']
ax.pie(x = sizes, labels = labels, colors=colors, autopct = '%1.1f%%', explode = explode, textprops={'fontsize': 14})
ax.set_title('Sentiment Polarity on Review Apps Data \n (total = 5000 review)', fontsize = 16, pad = 20)
plt.show()
```

Gambar 4.23 Code pie chart label data ulasan

Setelah code dijalankan maka akan di hasilkan gambar diagram pie seperti pada gambar 4.24 dibawah ini.





Gambar 4.25 Hasil visualisasi word cloud positif (kiri) dan negatif (kanan)

4.5. Term Frequency-Inverse Document Frequency (TF-IDF)

Penulisan menggunakan metode TF-IDF untuk mengubah data teks menjadi vektor namun dengan memperhatikan apakah sebuah kata tersebut cukup informatif atau tidak. Proses TF-IDF ini membuat kata yang sering muncul memiliki nilai yang cenderung kecil, sedangkan untuk kata yang semakin jarang muncul akan memiliki nilai yang cenderung besar. Library yang digunakan untuk metode TF-IDF ini dapat diimport pada *Sklearn*. Feature yang dipilih untuk diterapkan metode TF-IDF ini adalah feature “Text_Clean” yang berisi ulasan yang telah di preprocessing sebelumnya. Kemudian pada tahap ini juga kita memilih variable independen dan variable dependen, dimana variable independen adalah **X** (ulasan) dan variable dependennya adalah **y** (label).

```
from sklearn.feature_extraction.text import TfidfVectorizer
df_new['Text_Clean'] = df_new['Text_Clean'].astype(str)
tfidf = TfidfVectorizer()
ulasan = df_new['Text_Clean'].values.tolist()
tfidf_vect = tfidf.fit(ulasan)
X = tfidf_vect.transform(ulasan)
y = df_new['polarity']
print(X[0:2])
```

Gambar 4.26 Code Proses TF-IDF

Setelah seluruh ulasan di terapkan metode TF-IDF maka didapatkan nilai setiap kata dari setiap ulasan. Pada gambar 4.27 dapat dilihat nilai bobot kata dari data ulasan satu dan ulasan dua.

```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy

(0, 4938)    0.5265608844970339
(0, 2009)    0.3808287910198466
(0, 1867)    0.18880090931696192
(0, 1660)    0.3879926583851103
(0, 238)     0.6257147756992851
(1, 5955)    0.6149556728350231
(1, 4484)    0.38552755644060793
(1, 4294)    0.45526803292496115
(1, 1038)    0.35928817793551615
(1, 913)     0.36992032529366525

```

Gambar 4.27 Tampilan Bobot Kata Setelah Proses TF-IDF

4.6. Membagi Data Latih dan Data Uji

Tahapan selanjutnya adalah membagi dataset menjadi data latih dan data uji (data split). Rasio yang penulis gunakan dalam pembagian dataset ini adalah 9:1 yaitu 4500 ulasan sebagai data latih dan 500 ulasan sebagai data uji. Rasio pembagian ini dipilih dikarenakan jumlah data yang hanya 5000 ulasan sehingga perlu data yang cukup banyak pada proses latih (*training*) agar model yang dibuat dapat memiliki cukup banyak informasi untuk dipelajari. Sehingga model dapat mengklasifikasi data uji dengan baik dan menghasilkan akurasi yang tinggi.

```

from pandas.core.common import random_state
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=0)
data_latih = len(y_train)
data_test = len(y_test)
all_data = len(y)
print("Total Keseluruhan Data : ", all_data)
print("Total Data Latih : ", data_latih)
print("Total Data Test : ", data_test)

Total Keseluruhan Data : 5000
Total Data Latih : 4500
Total Data Test : 500

```

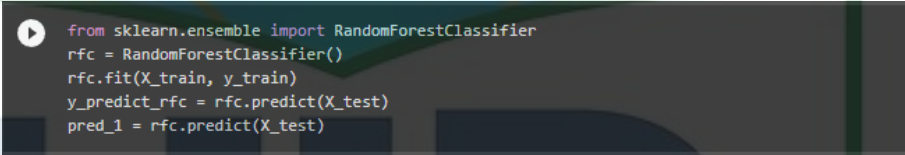
4.28 Pembagian data latih dan data uji

4.7. Klasifikasi Random Forest

Tahapan selanjutnya setelah preprocessing adalah tahap klasifikasi.

4.7.1. Model Random Forest

Tahap klasifikasi menggunakan metode Random Forest dimulai dengan membangun arsitektur Random Forest yang dapat kita import pada **Sklearn**. Setelah arsitektur Random Forest di import kemudian di simpan dengan nama yang ditentukan oleh penulis, dimana penulis menyimpannya dengan nama 'rfc'. Tahapan selanjutnya adalah membuat model dari arsitektur tersebut dengan melatih model dengan menggunakan data latih. Model yang telah dilatih dengan data latih kemudian dapat digunakan untuk mengklasifikasi data uji yang telah dipisahkan dengan labelnya sebelumnya. Hasil prediksi dari data uji yang berjumlah 500 kemudian disimpan dengan nama 'pred_1'.



```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier()
rfc.fit(X_train, y_train)
y_predict_rfc = rfc.predict(X_test)
pred_1 = rfc.predict(X_test)
```

Gambar 4.29 Model Random Forest Classifier

Hasil prediksi ini kemudian akan dibandingkan dengan label asli dari dataset untuk melihat tingkat kebenaran yang dihaikan dari model yang telah dibuat.

BAB V

HASIL DAN PEMBAHASAN

5.1. Hasil Klasifikasi

Pada tahap ini penulis memperlihatkan akurasi dari hasil prediksi data uji, dimana nilai kebenaran dari hasil prediksi yang dihasilkan oleh model akan dibandingkan dengan nilai actual dari data uji tersebut. Sebelumnya library yang dibutuhkan pada proses ini di import dari Sklearn. Library yang digunakan adalah 'accuracy_score'

```
[140] from sklearn.metrics import accuracy_score
      print("Random Forest Classifier Accuracy Score : ", accuracy_score(y_predict_rfc, y_test)*100,"%")

Random Forest Classifier Accuracy Score : 88.4 %
```

Gambar 4.30 Akurasi Klasifikasi

Dapat dilihat bahwa akurasi yang dihasilkan dari model yang telah dibuat adalah 88.4%, yang artinya model yang telah dibuat sudah sangat baik.

5.3. AUC-ROC Curve

Melihat performa dari model dapat juga dilakukan dengan melihat kurva AUC_ROC. Code dari proses plot kurva dibuat dengan menggunakan library metrics dari Sklearn. Dimana sumbu X adalah nilai False Positive dan sumbu Y adalah nilai True Positive. Model dikatakan sangat baik dikarenakan dapat terlihat kurva naik ke atas dan mendekati satu.

```

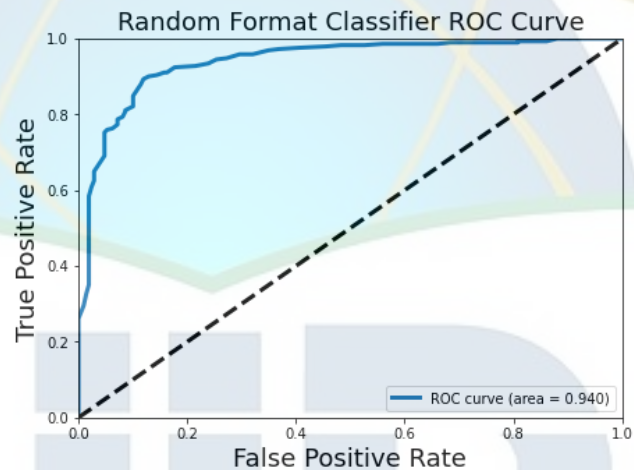
from sklearn import metrics
rfc_FPR = dict()
rfc_TPR = dict()
rfc_ROC_AUC = dict()

rfc_FPR[1], rfc_TPR[1], _ = roc_curve(y_test, rfc_y_score)
rfc_ROC_AUC[1] = metrics.auc(rfc_FPR[1], rfc_TPR[1])

plt.figure(figsize=[7,5])
plt.plot(rfc_FPR[1], rfc_TPR[1], label='ROC curve (area = %0.3f)' % rfc_ROC_AUC[1], linewidth=3)
plt.plot([0, 1], [0, 1], 'k--', linewidth=3)
plt.xlim([0.0, 1])
plt.ylim([0.0, 1])
plt.xlabel('False Positive Rate', fontsize=18)
plt.ylabel('True Positive Rate', fontsize=18)
plt.title('Random Forest Classifier ROC Curve', fontsize=18)
plt.legend(loc='lower right')
plt.show()

```

Gambar 4.32 Code untuk membuat kurva AUC-ROC



Gambar 4.33 Kurva AUC_ROC

5.2. Classification Report

Classification report dari hasil klasifikasi data uji dapat didapatkan dengan mengimport library `classification_report` terlebih dahulu dan menerapkannya pada hasil prediksi.


```
from sklearn.metrics import classification_report
print("RFC Classification Report : \n", classification_report(y_test, prediction1))
```

RFC Classification Report :				
	precision	recall	f1-score	support
0	0.86	0.86	0.86	209
1	0.90	0.90	0.90	291
accuracy			0.88	500
macro avg	0.88	0.88	0.88	500
weighted avg	0.88	0.88	0.88	500

Gambar 4.31 Mengeluarkan Hasil Classification Report

Dari hasil classification report pada gambar 4.31 dapat di ambil informasi bahwa kemampuan pengklasifikasi sudah cukup baik karena untuk kelas 0 dan kelas 1 keduanya lebih dari 80%, hal ini menunjukkan kemampuan model random forest yang telah dilatih untuk tidak melabeli label positif pada data ulasan yang sebenarnya berlabel negatif. Recall pada hasil klasifikasi juga menunjukkan nilai yang cukup tinggi, begitu juga dengan nilai f1-score yang artinya model ini sudah cukup seimbang dan baik.

5.4. Confusion Matrix

Hasil klasifikasi data uji dapat di lihat dengan mudah menggunakan confusion matriks. Library yang digunakan untuk menampilkan confusion matrix dari hasil klasifikasi data uji adalah library confusion_matrix. Input dari confusion matrix yaitu label actual dan label hasil prediksi. Perintah dalam cm_df merupakan perintah untuk membuat label pada gambar confusion matrix yang sesuai dengan nama label asli pada pada dataset. Kemudian setelah itu jumlah label actual dan predict akan di hitung dan kemudian jumlah True Positive, True Negative, False Positive dan False Negative akan plot pada gambar.

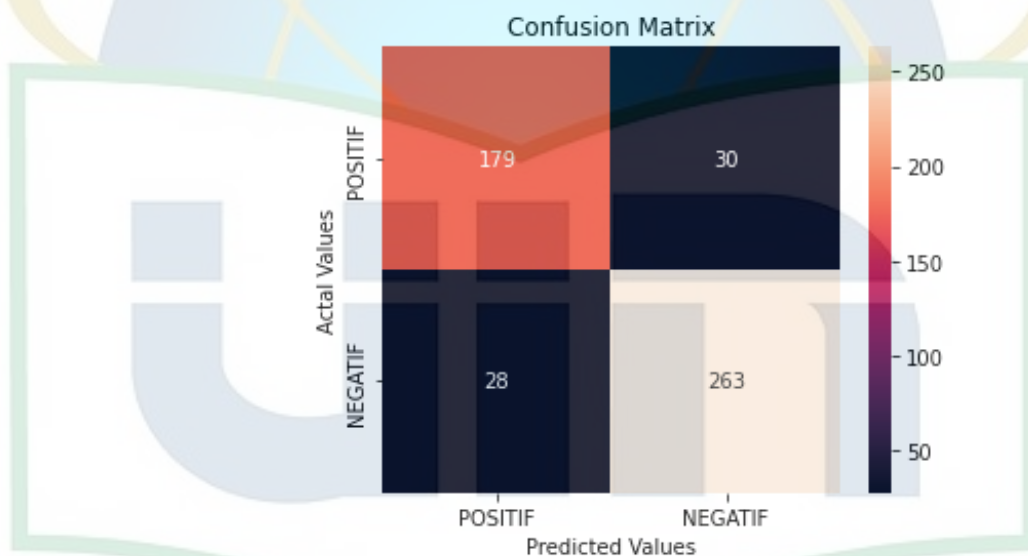

```
[ ] from sklearn.metrics import confusion_matrix
    cm = confusion_matrix(y_test, prediction1)

[ ] cm_df = pd.DataFrame(cm,
                          index = ['POSITIF','NEGATIF'],
                          columns = ['POSITIF','NEGATIF'])

[ ] #Plotting the confusion matrix
    plt.figure(figsize=(5,4))
    sns.heatmap(cm_df,annot=True, fmt='g')
    plt.title('Confusion Matrix')
    plt.ylabel('Actal Values')
    plt.xlabel('Predicted Values')
    plt.show()
```

Gambar 4.34 Code Confusion Matrix

Hasil plot ditampilkan pada gambar 4.35 di bawah ini



Gambar 4.35 Confusion Matrix

True Positif : 179
 True Negatif : 263
 False Positif : 30
 False Negatif : 28

Dapat diketahui bahwa dari hasil klasifikasi data uji sebanyak 500, didapatkan 442 ulasan dapat diprediksi dengan benar dan 58 ulasan diprediksi salah. Hasil ini sudah bisa dikatakan sangat baik.



BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan hasil pembahasan dari penelitian yang penulis lakukan, berikut merupakan kesimpulan pada penelitian ini:

1. Penerapan algoritma Random Forest dalam melakukan klasifikasi sengketa komplain pelanggan perusahaan KAI Access memperoleh hasil bahwa berdasar pada sentiment yang diperoleh dari 5000 ulasan google playstore yang melalui tahap preprocessing dan seleksi fitur TF-IDF didapatkan jumlah positif lebih dominan.
2. Tingkat akurasi dari algoritma Random Forest dalam melakukan klasifikasi yang dihasilkan dari model yang telah dibuat adalah 88.4%, yang artinya model yang telah dibuat sudah sangat baik.

6.2 Saran

Adapun saran yang dapat diberikan berdasarkan hasil dari penelitian ini antara lain:

1. Bagi Peneliti Selanjutnya
Disarankan untuk peneliti selanjutnya dengan topik penelitian serupa untuk menerapkan algoritma lain dalam mengklasifikasikan suatu ulasan, sehingga diperoleh pembaharuan penelitian dengan topik pengklasifikasian.
2. Bagi Instansi KAI Access
Disarankan untuk pihak instansi memperhatikan hasil ulasan dan mengimplementasikan tindakan dengan segera terhadap kritik dan saran yang telah disampaikan oleh klien, sehingga perusahaan dapat berkembang lebih baik lagi.

DAFTAR PUSTAKA

- Alita, D., & Isnain, A. R. (2020). Pendeteksian Sarkasme pada Proses Analisis Sentimen Menggunakan Random Forest Classifier. *Jurnal Komputasi*, 8(2), 50–58. <https://doi.org/10.23960/komputasi.v8i2.2615>
- Devella, S., Yohannes, Y., & Rahmawati, F. N. (2020). Implementasi Random Forest Untuk Klasifikasi Motif Songket Palembang Berdasarkan SIFT. *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, 7(2), 310–320. <https://doi.org/10.35957/jatisi.v7i2.289>
- Dicky Nofriansyah, S.Kom., M. K. D. I. G. W. N. M. S. 2017. (2017). *ALGORITMA DATA MINING DAN PENGUJIAN*. DEEPUBLISH.
- Faizah, A., Saputro, P. H., & Firdaus, R. A. J. (2019). Pemanfaatan Microcontroller Arduino Uno Untuk Sistem Monitoring Suhu Dan Kelembaban Kumbung Jamur Tiram. *Inovate*, 04, 1–8.
- Hasan, M. (2017). *Menggunakan Algoritma Naive Bayes Berbasis*. 9, 317–324.
- Jay. (2022). *Pelabelan Data-Penting untuk model AI*, 2.
- Kirasich, K., Smith, T., & Sadler, B. (2018). Random Forest vs Logistic Regression: Binary Classification for Heterogeneous Datasets. *Data Science Review*, 1(3), 9. <https://scholar.smu.edu/datasciencereviewhttp://digitalrepository.smu.edu.Avala> bleat:<https://scholar.smu.edu/datasciencereview/vol1/iss3/9>
- Lazulfa, I., & Saputro, P. H. (2017). Portfolio Optimization With Buy-in Thresholds Constraint Using Simulated Annealing Algorithm. *Prosiding SI MaNIs (Seminar Nasional Integrasi Matematika Dan Nilai Islami)*, 1(1), 370–377.
- Li, J., Zhong, P. an, Yang, M., Zhu, F., Chen, J., Liu, W., & Xu, S. (2020). Intelligent identification of effective reservoirs based on the random forest classification model. *Journal of Hydrology*, 591, 125324. <https://doi.org/10.1016/j.jhydrol.2020.125324>
- Masripah, S. (2015). Evaluasi Penentuan Kelayakan Pemberian Kredit Koperasi Syariah Menggunakan Algoritma Klasifikasi C4.5. *Jurnal Pilar Nusa Mandiri*, XI(1), 1–10.
- Musthofa Galih Pradana, Azriel Christian Nurcahyo, P. H. S. (2020). PENGARUH SENTIMEN DI SOSIAL MEDIA DENGAN HARGA SAHAM PERUSAHAAN. *Jurnal Ilmiah Edutic*, 6(2).
- Pradana, M. G. (2018). *PENYAKIT DIABETES MELLITUS MENGGUNAKAN METODE CERTAINTY FACTOR DESIGN EXPERT SYSTEM FOR DIAGNOSING DIABETES*. 11(2), 182–191.

Pradana, M. G. (2020). *PENGUNAAN FITUR WORDCLOUD DAN DOCUMENT TERM MATRIX DALAM TEXT MINING*.

Riska Chairunisa, Adiwijaya, & Widi Astuti. (2020). Perbandingan CART dan Random Forest untuk Deteksi Kanker berbasis Klasifikasi Data Microarray. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 4(5), 805–812. <https://doi.org/10.29207/resti.v4i5.2083>

Saputra, A. W. (2019). *ANALISIS PERMASALAHAN PADA APLIKASI SMARTPHONE KAI ACCESS BEDASARKAN END-USER REVIEW MENGGUNAKAN METODE TEXT-MINING DAN FISHBONE DIAGRAM*, 102.

Shalev-Shwartz, S., & Ben-David, S. (2013). Understanding machine learning: From theory to algorithms. In *Understanding Machine Learning: From Theory to Algorithms* (Vol. 9781107057135). <https://doi.org/10.1017/CBO9781107298019>

Suryanegara, G. A. B., Adiwijaya, & Purbolaksono, M. D. (2021). Peningkatan Hasil Klasifikasi pada Algoritma Random Forest untuk Deteksi. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 1(10), 114–122.

Syukron, A., & Subekti, A. (2018). Penerapan Metode Random Over-Under Sampling dan Random Forest Untuk Klasifikasi Penilaian Kredit. *Jurnal Informatika*, 5(2), 175–185. <https://doi.org/10.31311/ji.v5i2.4158>

Yoga Religia, Agung Nugroho, W. H. (2021). JURNAL RESTI Analisis Perbandingan Algoritma Optimasi pada Random Forest untuk. *Rekayasa Sistem Dan Teknologi Informasi*, 1(10), 187–192.

Yusuf Sulistyo Nugroho, N. E. (2017). Sistem Klasifikasi Variabel Tingkat Penerimaan Konsumen Terhadap Mobil Menggunakan Metode Random Forest. *Jurnal Teknik Elektro*, 9(1), 24–29. <https://doi.org/10.15294/jte.v9i1.10452>

Zailani, A. U., & Hanun, N. L. (2020). Penerapan Algoritma Klasifikasi Random Forest Untuk Penentuan Kelayakan Pemberian Kredit Di Koperasi Mitra Sejahtera. *Infotech: Journal of Technology Information*, 6(1), 7–14. <https://doi.org/10.37365/jti.v6i1.61>