PERANCANGAN SISTEM ANALISIS SENTIMEN PADA ULASAN APLIKASI MOBILE MENGGUNAKAN RANDOM FOREST CLASSIFIER DAN ASOSIASI TEKS (STUDI KASUS: APLIKASI MYTELKOMSEL)

LAPORAN KERJA PRAKTIK

Disusun oleh

MUHAMAD RENDI

5190411597

PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS & TEKNOLOGI
UNIVERSITAS TEKNOLOGI YOGYAKARTA
YOGYAKARTA
2022

LAPORAN KERJA PRAKTIK

PERANCANGAN SISTEM ANALISIS SENTIMEN PADA ULASAN APLIKASI MOBILE MENGGUNAKAN RANDOM FOREST CLASSIFIER DAN ASOSIASI TEKS (STUDI KASUS: APLIKASI MYTELKOMSEL)

Disusun oleh

MUHAMAD RENDI

5190411597

Telah diseminarkan

Pada tanggal 2022

Dosen Pembimbing

Anita Fira Waluyo, S.Si., M.Sc.
NIK. 110717132

Yogyakarta,
Ketua Program Studi Informatika

Commented [MR1]: Ttd dan tanggal ttd

Dr. Enny Itje Sela, S.Si., M.Kom.
NIK 111116086

LEMBAR PERNYATAAN

Commented [MR2]: Program sarjana dan fakultas
Dan di cek template

Saya yang bertanda tangan di bawah ini

Nama : Muhamad Rendi

NPM : 5190411597

Program Studi : Informatika

Program : Sarjana

Fakultas : Teknologi Informasi dan Elektro

Menyatakan bahwa Kerja Praktik dengan judul "Perancangan Sistem Analisis Sentimen Pada Ulasan Aplikasi Mobile Menggunakan Random Forest Classifier Dan Asosiasi Teks (Studi Kasus: Aplikasi Mytelkomsel)" ini adalah karya ilmiah asli saya dan belum pernah dipublikasikan oleh orang lain, kecuali yang tertulis sebagai acuan dalam naskah ini dan disebutkan dalam daftar pustaka. Apabila di kemudian hari, karya saya disinyalir bukan merupakan ka rya asli saya, maka saya bersedia menerima konsekuensi apa yang diberikan Program Studi Informatika Fakultas Sains & Teknologi Universitas Teknologi Yogyakarta.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Dibuat : Yogyakarta

Pada tanggal :

Yang menyatakan

Muhamad Rendi

ABSTRAK

Pada satu dekade terakhir, telah terjadi gelombang digitalisasi ke kehidupan masyarakat yang mengubah perilaku masyarakat sehari-hari. Salah satu bagian yang terlihat jelas berkembang ialah penggunaan aplikasi smartphone. Salah satu aplikasi yang memudahkan pengguna adalah aplikasi MyTelkomsel. Aplikasi MyTelkomsel adalah layanan berbentuk aplikasi yang diluncurkan Telkomsel untuk memberikan kemudahan mengelola akun SIM card dan mengakses layanan pelanggan dengan menggunakan smartphone. Hingga saat ini, aplikasi MyTelkomsel telah diunduh lebih dari 50 juta pengguna dan memiliki lebih dari 6 juta ulasan pengguna. Ulasan yang diberikan oleh pengguna mencakup bukan hanya mengenai fitur pada aplikasi, namun juga pelayanan Telkomsel secara keseluruhan. Tidak bisa dipungkiri bahwa ulasan yang ditulis oleh pengguna dapat mempengaruhi citra sebuah perusahaan. Namun, memantau dan mengelola ulasan dari pengguna juga bukanlah hal yang mudah. Apalagi jika ulasan yang dimuat jumlahnya terlalu banyak untuk diproses secara manual. Oleh sebab itu, penulis melakukan penelitian menggunakan data ulasan pengguna aplikasi MyTelkomsel dari situs Google Play Store. Jumlah ulasan pengguna pada penelitian ini berjumlah 5.000 data, yang kemudian dilakukan normalisasi, lalu dilabelisasi menjadi kelas sentimen positif dan negatif. Dari hasil labelisasi data, selanjutnya dilakukan proses klasifikasi menggunakan algoritma *Random Forest* dan memperoleh tingkat akurasi tertinggi sebesar 97,44%. Pada penelitian ini juga dilakukan asosiasi teks untuk mengetahui informasi yang dianggap penting dan berguna dalam pengambilan keputusan. Asosiasi teks pada kelas sentimen positif adalah terkait aplikasi, transaksi, update,fitur dan harga. Sedangkan pada kelas sentimen negatif adalah terkait mati, ganggu, padam, susah, dan *error*.

Kata Kunci: *Web Scraping*, Analisis Sentimen, *Random Forest*, Asosiasi Teks, Google Play Store, MyTelkomsel

ABSTRACT

In the last decade, there has been a late wave of digitalization in people's lives that has changed people's daily behavior. One part that is clearly growing is the use of smartphone applications. One application that makes it easy for users is the MyTelkomsel application. The MyTelkomsel application is a service in the form of an application launched by Telkomsel to provide convenience in managing SIM card accounts and accessing customer service using a smartphone. To date, the MyTelkomsel application has been downloaded by more than 50 million users and has more than 6 million user reviews. Reviews given by users include not only the features of the application, but also Telkomsel's services as a whole. It is undeniable that reviews written by users can affect the image of a company. However, monitoring and managing reviews from users is also not easy. Especially if the reviews that are loaded are too many to be processed manually. Therefore, the authors conducted research using user review data for the MyTelkomsel application from the Google Play Store site. The number of user reviews in this study amounted to 5,000 data, which were then normalized, then labeled into positive and negative sentiment classes. From the results of data labeling, the classification process was carried out using the Random Forest algorithm and obtained the highest level of accuracy of 97.44%. In this study, text associations were also conducted to find out information that was considered important and useful in decision making. Text associations in the positive sentiment class are related to applications, transactions, updates, features and prices. While in the negative sentiment class, it is related to death, disturbance, extinction, difficulty, and error.

KeyWord **:** *Web Scraping*, sentiment analysis , *Random Forest*, *text association*, Google Play Store, MyTelkomsel

Commented [MR3]: Sebelum bab 1, nomor halaman harus ada di tengah bawah

KATA PENGANTAR

Commented [MR4]: Nama dosbim

Segala puji dan syukur tak lupa dipanjatkan kepada Allah SWT yang telah memberikan rahmat dan anugerah-Nya, sehingga dapat menyelesaikan proposal kerja praktik yang berjudul "Rancang Bangun Sistem Analisis Sentimen Pada Ulasan Aplikasi Mobile Menggunakan Random Forest Classifier Dan Asosiasi Teks (Studi Kasus: Aplikasi MyTelkomsel)".

Penyusunan Kerja Praktik diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana pada Program Studi Informatika Fakultas Sains & Teknologi Universitas Teknologi Yogyakarta.

Pada kesempatan ini, tak lupa terimakasih kepada semua pihak yang telah memberikan dukungan moril maupun materil sehingga proposal kerja praktik ini dapat selesai. Ucapan terimakasih ini penulis tujukan kepada :

1. Dr. Bambang Moertono Setiawan, M.M., C.A., Akt. Selaku RektorUniversitas Teknologi Yogyakarta.
2. Dr. Endy Marlina, MT. selaku Dekan Fakultas Sains & Teknologi,Universitas Teknologi Yogyakarta.
3. Dr. Enny Itje Sela, S.Si., M.Kom. selaku Ketua Program Studi Informatika,Universitas Teknologi Yogyakarta.
4. Anita Fira Waluyo, S.Si., M.Sc. selaku Dosen Pembimbing Kerja Praktik yang telah memberi arahan dan nasihat kepada penulis.
5. Kedua orang tua yang telah memberikan doa dan dukungan dalammenyelesaikan proposal penelitian ini.
6. Teman-taman saya yang telah membantu saya dalam menyusun proposal ini

Akhir kata, semoga proposal kerja praktik ini dapat berguna dan bermanfaat bagi para pembaca serta pihak-pihak lain yang berkepentingan.

Yogyakarta, …….. 2022

Penulis

DAFTAR ISI

Commented [MR5]: Jangan di Italic, untuk sub bab huruf awal yg kapital

Commented [AFWMS6]: -Penomoran anak subbab 3 diperbaiki Kembali
-6.1 ny mana?

Commented [MR7]: Sub bab capital hanya awalan huruf dan romawi kecil, halaman pendahulu 1

DAFTAR TABEL

Commented [MR8]: Spasi 1, hururf awalan capital

DAFTAR GAMBAR

BAB I
PENDAHULUAN

Commented [MR9]: Tiap2 awal bab penomoran tengah bawah, setelah itu pojok kanan atas

1.1 Latar Belakang

Pada satu dekade terakhir, telat terjadi gelombang digitalisasi ke kehidupan masyarakat yang mengubah perilaku masyarakat sehari-hari. salah satu bagian yang terlihat jelas berkembang ialah penggunaan aplikasi smartphone. Dengan kebutuhan pengguna yang luas hal ini menjadi peluar developer untuk mengembangkan aplikasi.

Menurut data dari SensorTower jumlah aplikasi siap unduh pada tahun 2021 pada platform Google Play Store sebesar 28,2 miliar aplikasi. Dengan jumlah aplikasi yang banyak semakin memudahkan pengguna dalam berbagai hal. Salah satu aplikasi yang memudahkan pengguna adalah aplikasi MyTelkomsel. Aplikasi MyTelkomsel adalah layanan berbentuk aplikasi yang diluncurkan Telkomsel untuk memberikan kemudahan mengelola akun SIM card dan mengakses layanan pelanggan dengan menggunakan smartphone. Hingga saat ini, aplikasi MyTelkomsel telah diunduh lebih dari 50 juta pengguna dan memiliki lebih dari 6 juta ulasan pengguna. Ulasan yang diberikan oleh pengguna mencakup bukan hanya mengenai fitur pada aplikasi, namun juga pelayanan Telkomsel secara keseluruhan. Ulasan ini mencakup keluhan yang bersifat negatif dan saran yang bersifat positif. Ulasan pengguna merupakan salah satu media yang efektif dan efesien untuk menemukan informasi terhadap citra dari suatu perusahaan. Hal ini dikarenakan harapan pelanggan dipengaruhi oleh pengalaman pembelian mereka sebelumnya, nasihat teman atau kolega serta hasil ulasan pembeli sebelumnya (Tjiptono, 208).

Google Play Store merupakan layanan digital milik Google berupa toko untuk memasarkan produk berupa aplikasi, permainan, hingga buku dan film. Google Play Store dapat diakses melalui aplikasi Android, situs web, dan Google TV (Marziah, L., 2020). Google Play memiliki beragam fitur, salah satunya adalah penggunaan dapat memberikan ulasan terhadap aplikasi. Berdasarkan ulasan tersebut dapat diketahui sentimen apa aja yang

mempengaruhi dan perlu diperbaiki oleh perusahaan. Ulasan dapat dianalisis menggunakan metode *text mining. Text mining* adalah proses penemuan pengetahuan dengan mengekstrak pola atau pengetahuan yang dianggap penting dari data teks (Bookhamer, P. dan Zhang, Z.J., 2016). Salah satu teknik dari *text mining* adalah analisis sentimen yang mana dapat digunakan untuk mengklasifikasi opini masyarakat ke dalam kelas positif, negatif, maupun netral. Dan Analisis sentimen merupakan riset komputasional dari opini, sentimen, dan emosi yang diekspresikan secara tekstual (Liu, 2010). Apabila ditemukan adanya ulasan pada kelas sentimen negatif, maka Pihak Telkomsel dapat dengan cepat mengambil tindakan untuk mengatasi dan menyelesaikan keluhan pengguna.

Pada penelitian ini, klasifikasi sentiment akan dilakukan menggunakan metode *Random Forest Classifier*. *Random Forest Classifier* merupakan salah satu metode dalam Decision Tree. Decision Tree atau pohon pengambil keputusan adalah sebuah diagram alir yang berbentuk seperti pohon yang memiliki sebuah root node yang digunakan untuk mengumpulkan data, Sebuah inner node yang berada pada root node yang berisi tentang pertanyaan tentang data dan sebuah leaf node yang digunakan untuk memecahkan masalah serta membuat keputusan. *Random Forest* pertama kali dikenalkan oleh Breiman pada Tahun 2001. Dalam penelitiannya menunjukkan kelebihan *Random Forest* antara lain dapat menghasilkan error yang lebih rendah, memberikan hasil yang bagus dalam klasifikasi, dapat mengatasi data training dalam jumlah sangat besar secara efisien, dan metode yang efektif untuk mengestimasi missing data (Breiman, 2001). Setelah melakukan klasifikasi, dilakukan proses eksplorasi informasi seluas-luasnya dari masing-masing kelas sentimen positif dan negatif. Proses eksplorasi yang dilakukan menggunakan asosiasi teks untuk menemukan topik-topik yang umumnya dibahas oleh pengguna beserta keterkaitan antar topik tersebut.

Berdasarkan latar belakang tersebut, penelitian ini akan menganalisa sentimen ulasan aplikasi *mobile* dengan objek penelitian adalah MyTelkomsel. Data ulasan diambil dari kolom ulasan di Google Play Store. Melalui penelitian ini, diharapkan nantinya dapat memberikan informasi

yang berguna untuk pihak Telkomsel secara khusus maupun pihak lain yang membutuhkan.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah diuraikan diatas, maka rumusan masalah dalam penelitian ini meliputi:

1. Apakah algoritma *Random Forest Classifier* mampu dalam mengklasifikasikan ulasan aplikasi MyTelkomsel menjadi kelas sentimen positif dan negatif?
2. Apakah tingkat akurasi model yang menggunakan *Random Forest Classifier* mendekati keadaan realita?
3. Apakah ada informasi penting yang diperoleh dari hasil asosiasi teks?

1.3 Batasan Masalah

Batasan masalah yang ditentukan untuk menghindari perluasan pembahasan dalam penelitian ini adalah sebagai berikut:

1. Data ulasan yang digunakan berasal dari Google Play Store.
2. Data ulasan yang digunakan adalah ulasan (teks) berbahasa Indonesia.
3. Data ulasan yang tidak dapat dinormalisasi menggunakan perangkat lunak, dilakukan normalisasi secara manual oleh penulis berdasarkan acuan dari Kamus Besar Bahasa Indonesia (KBBI) dan Pedoman Umum Ejaan Bahasa Indonesia (PUEBI).
4. Setelah dinormalisasi, data ulasan akan diklasifikasi dan diasosiasi menjadi dua kelas sentimen, yakni positif dan negatif.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah mengetahui sentimen dari ulasan pengguna aplikasi MyTelkomsel di Google Play Store dan memperoleh informasi penting yang dapat dimanfaatkan dalam

pengambilan keputusan perusahaan untuk mengevaluasi pengembangan aplikasi MyTelkomsel.

1.5 Manfaat Penelitian

Pada penelitian ini, terdapat beberapa manfaat yang akan diperoleh berdasarkan tujuan penelitian, antara lain:

1. Bagi penulis, menerapkan ilmu pengetahuan yang telah diperoleh selaa perkuliahan dan mengetahui implementasi metode *Random Forest Classifier* dalam melakukan kalsifikasi data ulasan aplikasi MyTelkomsel
2. Bagi lingkup akademis, dapat dijadikan contoh studi kasus, acuan, serta referensi untuk melakukan penelitian terkait di masa mendatang.
3. Bagi instansi, dapat dijadikan sebagai wawasan dan acuan dalam mengambil keputusan bagi pihak Telkomsel maupun pihak lain yang mebutuhkan
4. Bagi pengguna aplikasi, mendapatkan feedback perbaikan aplikasi MyTelkomsel yang di harapkan

Commented [MR10]: 1.6 sistematika penulisan

1.6 Sistematika Penelitian

Sistematika penulisan yang dipergunakan dalam penulisan Penelitian ini dapat diuraikan sebagai berikut :

BAB 1 PENDAHULUAN

Pada bab ini akan dibahas tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

BAB 1 TINJAUAN PUSTAKA

Bab ini memaparkan penelitian-penelitian terdahulu yang berhubungan dengan permasalahan yang diteliti dan menjadi acuan konseptual

BAB 3 LANDASAN TEORI

Pada bab ini akan dibahas tentang teori-teori dan konsep yang berhubungan dengan penelitian yang dilakukan dan mendukung dalam pemecahan masalahnya.

BAB 4 METODOLOGI PENELITIAN

Bab ini memaparkan populasi dan sampel, variabel penelitian, jenis dan sumber data, metode analisis data, dan tahapan penelitian.

BAB 5 ANALISIS DAN PEMBAHASAN

Pada bab ini akan dibahas mengenai analisa yang dilakukan terhadap hasil pengumpulan, pengolahan dan analisa data yang diperoleh dari hasil penelitian.

BAB 6 PENUTUP

Pada bab ini akan dibahas mengenai kesimpulan yang diperoleh dari hasil penelitian dan analisa data yang telah dilakukan serta saran-saran yang dapat diterapkan dari hasil pengolahan data yang dapat menjadi masukan yang berguna kedepannya.

BAB II
KAJIAN HASIL PENELITIAN DAN LANDASAN TEORI

2.1 Kajian Hasil Penelitian

Dalam melakukan penelitian ini, peneliti mengacu pada beberapa penelitian yang telah dilakukan sebelumnya, antara lain penelitian oleh Fitri, E. dkk.(2020). Penelitian tersebut membahas analisis sentimen terhadap aplikasi Ruangguru menggunakan *Random Forest, Random Forest,* dan *Support Vectors Machine* berdasarkan ulasan pengguna aplikasi RuangGuru dari Google Play Store. Penelitian ini memberikan hasil bahwa dari model klasifikasi *Random Forest* 97,16% dengan menggunakan *Cross Validation* dan skor AUC 0,996. Kemudian akurasi dengan model support klasifikasi *Support Vector Machine* menghasilkan tingkat akurasi sebesar 96,01% dengan nilai AUC sebesar 0,543 dan akurasi pada pengujian model klasifikasi *Naive Bayes* sebesar 94,16% dari nilai AUC 0,999. Penelitian ini menunjukkan peningkatan akurasi dari penelitian sebelumnya sebesar 7,16% dengan final cut *Random Forest* sebagai model klasifikasi *Random Forest* dengan performansi terbaik.

Dalam penelitian yang dilakukan oleh Alita, Debby. dkk (2020). melakukan kombinasi antara proses sentimen analisis dengan deteksi sarkasme untuk pengklasifikasian opini yang terdapat pada Twitter. Proses analisis sentimen dilakukan dengan tahapan preprocessing dan ekstraksi fitur dan diklasifikan dengan menggunakan metode Support Vector Machine dilanjutkan dengan proses pendeteksian sarkasme yang dilakukan tahapan ekstraksi fitur dengan 4 set fitur yaitu sentiment related, punctuation-relate, lexical and syntactic, dan pattern-relate dan diklasifikasikan dengan menggunakan metode Random Forest Classifier. Hasil penelitian ini didapatkan peningkatan nilai ratarata 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, data dengan label negatif berjumlah 587 dan data dengan label netral berjumlah 462. Data sarkasme didapatkan dari tweet dengan label positif yang kemudian diberikan label sarkasme atau tidak sarkasme dan didapat hasil label dengan

Commented [AFWMS11]: Ini dicek lagi ukuran batas penulisan kanan kirinya apakah sama dengan yg di BAB I?

jumlah keseluruhan berlabel sarkasme berjumlah 354 dan tidak sarkasme berjumlah 669.

Penelitian lainnya yang menjadi acuan adalah penelitian yang dilakukan oleh Adiwijaya, dkk. (2021). Penelitian tersebut menggunakan teknik pengekstrakan fitur menggunakan *Word2Vec*, serta menggunakan metode *Random Forest* untuk melakukan pengklasifikasian sentimen. Dari penelitian yang telah dilakukan, didapatkan penelitian yang menggunakan skip-gram *Word2Vec* 300 dimensi, kemudian menerapkan *Adaptive Boosting* kepada base model, memiliki nilai akurasi terbaik sebesar 75.76%

Mengacu pada penelitian lain yang dilakukan oleh Susanto, Irwan. Dkk. (2021). Penelitian tersebut mengusulkan model analisis sentimen pelanggan hotel menggunakan metode *Random Forest Classifier* dan *Term Frequency–Inverse Document Frequency* (TF–IDF). Dataset yang digunakan untuk membangun model sentimen analisis adalah data komentar-komentar pelanggan hotel di Purwokerto yang diunduh dari situs tripadvisor.co.id. Pada preprocessing melibatkan proses konversi *slangword* menjadi kata baku sesuai KBBI, *stemming*, dan menambahkan kata-kata *stopword* baru selain *stopword* dalam library sastrawi. Hasil penelitian menunjukkan akurasi model mencapai akurasi 87,23%. Akan tetapi jika tanpa proses stemming, akurasi model hanya 87,01%.

Penelitian lainnya yang menjadi acuan tambahan adalah penelitian yang dilakukan oleh Fauziah. dkk. (2021). Penelitian tersebut ada untuk mengetahui opini pada pengguna Twitter yang ada berkaitan dengan event Flash Sale yang diadakan e-commerce. Dengan menggunakan metodelogi tiga algoritma klasifikasi yaitu *Naive Bayes, K-Nearest Neighbour* dan *Random Forest* dalam pengklasifikasian data untuk mengetahui akurasi tingkat nilai sentimen pengguna Twitter pada event Flash Sale. Penelitian ini mengambil dua sampel data dari kata kunci “flash sale” dan “flash sale shopee” ,hasil dari implementasi ketiga algoritma klasifikasi tersebut yaitu akurasi sebesar 83.53% *Naïve Bayes*, 82.94% *K-NN*, 80.59% *Random Forest* untuk kata kunci “flash sale” dan 81.48% *Naïve Bayes*, 77,78% K-NN, 74.07% *Random Forest* untuk kata kunci “flash sale shopee”. Dengan ini Algoritma *Naïve Bayes* menjadi rekomendasi untuk penggklasifikasian

data Analisis Sentimen dengan akurasi lebih besar dan lebih stabil digunakan untuk data yang besar maupun kecil

Tabel 2.1 Perbandingan Kajian Hasil Penelitian

No	Judul	Penulis	Metode	Kesimpulan
1	Analisis Sentimen Terhadap Aplikasi Ruangguru Menggunakan Algoritma Naive Bayes, Random Forest Dan Support Vector Machine	Evita Fitri, yuri Yuliani, Susy Rosyida, Windu gata (2020)	Random Forest(NBC), Random Forest (RF), Support Vector Machine (SVM)	Model klasifikasi Random Forest 97,16% dengan menggunakan Cross Validation dan skor AUC 0,996. Kemudian akurasi dengan model support klasifikasi Support Vector Machine menghasilkan tingkat akurasi sebesar 96,01% dengan nilai AUC sebesar 0,543 dan akurasi pada pengujian model klasifikasi Naive Bayes sebesar 94,16% dari nilai AUC 0,999. Penelitian ini menunjukkan peningkatan akurasi dari penelitian sebelumnya sebesar 7,16% dengan final cut Random Forest sebagai model klasifikasi Random Forest dengan performansi terbaik
2	Pendeteksian Sarkasme pada Proses Analisis sentimen Menggunakan Random Forest Classifier	Debby Alita, Auliya Rahman (2020)	Random Forest Classifier (RFC)	Didapatkan peningkatan nilai ratarata 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.

Commented [AFWMS12]: Ini kenapa semua Pustaka malah tdk ada di dftar Pustaka??

3	Analisis Sentimen Terhadap Ulasan Film Menggunakan algoritma Random Forest	Muhammad Asjad Adna Jihad, Adiwijaya, Widi Astuti (2021)	Random Forest dan (TF–IDF)	Didapatkan penelitian yang menggunakan skip-gram Word2Vec 300 dimensi, kemudian menerapkan Adaptive Boosting kepada base model, memiliki nilai akurasi terbaik sebesar 75,76%

Tabel 2.1 Perbandingan Kajian Hasil Penelitian Lanjutan

No	Judul	Penulis	Metode	Kesimpulan
4	Analisis Sentimen Pelanggan Hotel di Purwokerto menggunakan Metode Random Forest dan TF-IDF (Studi Kasus: Ulasan Pelanggan Pada Situs TRIPADVISOR)	Bamo Bayu Baskoro, Irwan Susanto, Siti Khomsah (2021)	Random Forest dan	Hasil penelitian menunjukkan akurasi model mencapai akurasi 87,23%. Akan tetapi jika tanpa proses stemming, akurasi model hanya 87,01%.
5	Sentimen Analisis pengguna Twitter pada Event Flash Sale Menggunakan Algoritma K-NN, Random Forest, dan Naive Bayes	Aprilia Wandani, Fauziah, Andrian ningsih (2021)	K-NN, Random Forest, Naïve Bayes	Hasil dari implementasi ketiga algoritma klasifikasi tersebut yaitu akurasi sebesar 83.53% Naïve Bayes, 82.94% K-NN, 80.59% Random Forest untuk kata kunci "flash sale" dan 81.48% Naïve Bayes, 77,78% K-NN, 74.07% Random Forest untuk kata kunci "flash sale shopee". Dengan ini Algoritma Naïve Bayes menjadi rekomendasi untuk penggklasifikasian data Analisis Sentimen dengan akurasi lebih besar dan lebih stabil digunakan untuk data yang besar maupun kecil.

Kaitan antara penelitian penulis dengan kajian penelitian terdahulu adalah sebagai acuan untuk membuat penelitian dan membuat logika program untuk penyelesaian masalah studi kasus yang ditiliti. Adapun hal yang di tingkatkan dari penelitian terdahulu adalah penggunaan kamus

slang word atau bahasa alay dalam pembersihan data, sehingga data menjadi lebih siap untuk digunakan bahkan jika data mengandung kata alay indonesia.

Commented [MR13]: Keterangan kaitan dengan penilitan kita

2.2 Landasan Teori

2.2.1 Machine Learning

Machine learning merujuk pada sebuah metode yang membuat komputer memiliki kemampuan dalam mempelajari dan melakukan sebuah pekerjaan secara otomatis. Proses machine learning dilakukan melalui algoritma tertentu, sehingga pekerjaan yang diperintahkan kepada komputer dapat dilakukan secara otomatis (Hairani, 2018).

Commented [AFWMS14]: Nama ini sy liat tdk ada di Pustaka.
Semua acuan harus ada di daftar pustaka

Machine learning dilakukan melalui 2 fase, yaitu fase training dan fase application. Fase training adalah proses pemodelan dari algoritma yang digunakan akan dipelajari oleh sistem melalui training data, sedangkan fase application adalah proses pemodelan yang telah dipelajari sistem melalui fase training akan digunakan untuk menghasilkan sebuah keputusan tertentu, dengan menggunakan testing data. Machine learning dapat dilakukan dengan dua cara, yaitu supervised learning dan unsupervised learning. Unsupervised learning adalah pemrosesan sample data dilakukan tanpa mewajibkan hasil akhir memiliki bentuk yang sesuai dengan bentuk tertentu, dengan menggunakan beberapa sample data sekaligus. Penerapan unsupervised learning dapat ditemukan pada proses visualisasi, atau eksplorasi data. Supervised learning adalah pemrosesan sample data x akan diproses sedemikian rupa, sehingga menghasilkan output yang sesuai dengan hasil akhir y. Supervised learning dapat diterapkan pada proses klasifikasi (Hairani, 2018).

2.2.2 Natural Language Processing

Natural Language Processing (NLP) adalah cabang dari kecerdasan buatan yang berhubungan dengan interaksi antara komputer dan manusia menggunakan bahasa alami. Artinya pada sistem, bahasa alami mencakup percakapan informasidari basis data komputer ke dalam bahasa yang dapat dibaca oleh manusia. Sebuahsistem NLP harus memperhatikan pengetahuan terhadap bahasa itu sendiri, baik dari segi kata yang digunakan, bagaimana

kata-kata tersebut digabung untuk menghasilkan suatu kalimat, apa arti dari sebuah kata, apa fungsi sebuah kata dalamsebuah kalimat dan sebagainya. Pengolahan bahasa alami mengenal beberapatingkat pengolahan, antara lain:

a) Fonetik dan Fonologi

   Fonetik dan fonologi berhubungan dengan suara yang menghasilkan kata yang dapat dikenali. Proses pengolahan ini menjadi penting dalam aplikasiyang memakai metode *speech-based system*

b) Morfologi

   Morfologi merupakan pengetahuan tentang kata dan bentuknya yang dimanfaatkan untuk membedakan satu kata dengan kata lainnya. Padatingkat ini juga dapat dipisahkan antara kata dan elemen lainnya. Misalnya kata "mengerjakan" yang dibagi menjadi "kerja" (kata dasar); me- (imbuhan depan); dan -kan (imbuhan belakang).

c) Sintaksi

   Sintaksis merupakan pemahaman tentang urutan kata dalam pembentukan kalimat dan hubungan antarkata tersebut dalam proses perubahan bentuk dari kalimat menjadi bentuk yang sistematis. Pada tingkat ini dilakukan proses pengaturan tata letak suatu kata untuk membentuk kalimat yang dapat dikenali.

d) Semantik

   Semantik merupakan pemetaan bentuk struktur sintaks dengan memanfaatkan tiap kata ke dalam bentuk yang lebih mendasar dan tidak tergantung struktur kalimat. Semantik mempelajari arti suatu kata dan bagaimana arti dari kata-kata tersebut membentuk suatu arti kalimat yang utuh. Dalam tingkatan ini belum tercakup konteks dari kalimat tersebut.

e) Pragmatik

   Pengetahuan pada tingkatan pragmatik berkaitan dengan masing-masing konteks yang berbeda tergantung pada situasi dan tujuan

pembuatan sistem.

f) Discourge knowledge

Discourge knowledge melakukan pengenalan apakah suatu kalimat yang sudah dibaca dan dikenali sebelumya sehingga dapat mempengaruhi arti dari kalimat selanjutnya. Informasi ini penting diketahui untuk melakukan pengolahan arti terhadap kata ganti orang dan untuk mengartikan aspek sementara dari informasi.

g) Word knowledge

Word knowledge mencakup arti sebuah kata secara umum dan apakah ada arti khusus bagi suatu kata dalam suatu percakapan dalam konteks tertentu.

2.2.3 Web Scrapping

Website merupakan aplikasi yang di dalamnya terdapat berbagai dokumen multimedia (teks, gambar, animasi, maupun video) yang menggunakan protokol HTTP (*HyperText Transfer Protocol*), dan untuk mengaksesnya digunakan perangkat lunak yang disebut *browser*. Sedangkan *web scraping* adalah proses pengambilan sebuah dokumen semiterstruktur dari internet, umumnya berupa halaman web dalam bahasa markup seperti HTML (*HyperText Markup Language*) atau XHTML (*Extensible HyperText Markup Language*), dan menganalisis dokumen tersebut untuk diambil data tertentu dari halaman tersebut yang mana digunakan dalam kepentingan tertentu. Terdapat serangkaian langkah yang perlu dilakukan untuk melakukan *web scraping* (Josi, A. dkk., 2014) sebagai berikut:

a) *Create scraping template*: Pembuatan kode program yang akan mempelajari dokumen HTML dari *website* yang akan diambil informasinya.

b) *Explore site navigation*: Pembuatan kode program yang akan mempelajari teknik navigasi pada *website* yang akan diambil informasinya untuk ditirukan pada aplikasi *web scraper*.

c) *Automate navigation and extraction*: Berdasarkan informasi yang didapatkan dari langkah 1 dan 2 diatas, aplikasi *web scraper*

dibuat agar pengambilan informasi pada *website* dapat dilakukan secara otomatis.

d) *Extract data and package history*: Informasi yang telah diperoleh dari langkah 3 akan disimpan dalam format *file* tertentu, dan untuk selanjutnya dikonversi ke dalam bentuk format data yang siap diolah.

2.2.4 Text Mining

Text mining merupakan proses penggalian informasi dari sekumpulan dokumen data berupa teks yang mengandung informasi yang tidak terstruktur dengan menggunakan analisis tertentu (Feldman, R. dan Sanger, J., 2007). Pekerjaan yang dilakukan dalam konsep *text mining* secara garis besar adalah penggalian deskriptif (*descriptive mining*) dan penggalian prediktif (*predictive mining*). Pekerjaan *predictive mining* meliputi klasifikasi dokumen ke dalam kategori-kategori, lalu menggunakan informasi tersebut untuk membuat keputusan. Misalnya, kepuasan pelanggan terhadap suatu produk dapat diketahui melalui dokumen komentar pembelian, sehingga komentar pelanggan di pembelian yang akan datang dapat diprediksi. Lalu dilakukan *descriptive mining* yang membantu perusahaan untuk melakukan pengelompokkan dokumen berdasarkan konsep yangtelah ditentukan . Untuk memperoleh informasi akhir yang berguna bagi pemilik data, *text mining* harus melalui tiga tahap(Miner, G. dkk., 2012) sebagai berikut:

a) Preprocessing

Tahap ini mengkonversi informasi dari format yang belum terstrukturmenjadi format yang dapat diproses.

b) Penyusun vektor

Untuk dapat dipahami oleh sistem operasi *text mining*, sebuah vektorrepresentasi dari token-token kata perlu dibuat berdasarkan kemunculan kata tersebut dalam dokumen.

c) Ekstraksi informasi

Metode ekstraksi informasi yang digunakan dalam penelitian ini adalah klasifikasi yang membagi objek ke dalam kategori yang

telah ditentukan (*supervised method*). Pendekatan model klasifikasi yang digunakan adalah teknik statistik atau *machine learning*. Pendekatan *machine learning* akan dipakai dalam penelitian ini di mana mesin akan mengoperasikan model yang mampu belajar dari contoh dokumen yang terklasifikasi.

2.2.5 Analisis Sentimen

Analisis Sentimen adalah sebuah proses untuk menentukan atau mengukur nilai sentimen atau opini yang ada terhadap suatu objek atau kejadian yang berupa teks dan dapat dikategorikan sebagai sentimen positif, negatif, serta netral. Pengguna internet saat ini banyak yang menuliskan pendapat atau opini, pengalaman dan berbagai hal yang terjadi atau sekiranya menarik untuk mereka. Analisis sentimen dianggap sebagai masalah pengelompokan. Sama seperti dalam laporan besar, nilai sentimen pada tweet dapat dikomunikasikan dalam berbagai cara dan ditandai dengan adanya sentimen didalamnya. jika ada sentimen dalam tweet, mengandung polar word atau kata-kata berlawanan maka itu ditetapkan positif atau negatif, jika tidak dianggap Netral. Langkah analisis sentimen sebagai berikut:

a) *Level 1*

Mencari sentimen negatif dan positif pada setiap baris

b) *Level 2*

Analisa sentimen seluruh dokumen sebagai negatif atau positif

c) *Level 3*

Menerapkan pengelompokan dimana mengumpulkan semua atribut yang ada dengan hasil sentimen yang sama

d) *Level 4*

Memanfaatkan visualisasi data dari analisis sentiment untuk interaksi antar *user*

2.2.6 Text Preprocessing

Dalam proses *text mining*, dokumen yang digunakan harus dipersiapkan terlebih dahulu sebelum dapat digunakan dalam proses utama. Proses mempersiapkan dokumen atau dataset mentah disebut juga

dengan proses *text preprocessing*. *Text preprocessing* berfungsi untuk mengubah data teks yang tidakterstruktur atau sembarang menjadi data yang terstruktur. Adapun serangkaian proses yang dilakukan dalam tahapan *text preprocessing* (Krouska, A. dkk., 2016) adalah sebagai berikut:

a) Spelling normalization

Spelling normalization adalah proses subtitusi atau perbaikan kata-kata singkatan atau salah ejaan. Subtitusi kata dilakukan untuk menghindari jumlah perhitungan dimensi kata yang melebar. Perhitungan dimensi kata akan melebar jika kata yang salah eja tidak diubah karena kata tersebut sebenarnya mempunyai maksud dan arti yang sama, tapi dianggap sebagai entitas yang berbeda pada saat proses penyusunan matriks.

b) Case folding

Case folding adalah proses penyamaan *case* (format huruf) dalam sebuah dokumen. Hal ini dilakukan untuk mempermudah pencarian. Tidak semua dokumen teks konsisten dalam penggunaan huruf kapital. Oleh karena itu, peran *case folding* dibutuhkan dalam mengkonversi keseluruhan teks dalam dokumen menjadi suatu bentuk standar.

c) Tokenizing

Tokenizing atau tokenisasi adalah proses penguraian teks yang semula berupa kalimat-kalimat yang berisi kata-kata. Proses tokenisasi diawali dengan menghilangkan beberapa *delimiter* yaitu simbol dan tanda baca yangada pada teks tersebut seperti @, $, /, &, #, tanda titik (.), koma (,) tanda tanya (?), dan tanda seru (!). Proses pemotongan string berdasarkan tiap kata yang menyusunnya, umumnya setiap kata akan terpisahkan dengan karakter spasi. Proses tokenisasi mengandalkan karakter spasi pada dokumen teks untuk melakukan pemisahan. Hasil dari proses ini adalah kumpulan kata.

d) Stemming

Stemming dilakukan dalam pengolahan data teks untuk mendapatkan kata dasar dari sebuah kata yang telah mengalami

imbuhan dengan asumsi bahwa kata-kata tersebut sebenarnya memiliki makna dan arti yang sama. Algoritma ini bekerja berdasarkan morfologi struktural dalam kalimat bahasa Indonesia, yang terdiri atas awalan, akhiran, sisipan, dan awalan+akhiran. Inti dari tahap ini memiliki tujuan meliputi: 1) dalam perkara efisiensi, pada *stemming* dilakukan pengurangan jumlah kata dalamdokumen agar mengurangi kebutuhan dalam ruang penyimpanan dan mempercepat dalam melakukan pencarian. 2) dalam perkara efektivitas, *stemming* dilakukan untuk mengurangi *recall* dengan pengurangan bentuk-bentuk kata ke dalam bentuk dasarnya. Sebagai contoh adalah kata "men- duduk-i", "minum-lah", "per-banding-an", dan sebagainya.

e) Filtering

Filtering adalah proses mengambil kata-kata penting dari hasil token. Algoritma *stopword* (membuang kata yang kurang penting) atau *wordlist* (menyimpan kata yang penting) dapat digunakan pada tahap ini. *Stopword* adalah kata-kata yang tidak deskriptif dan bukan merupakan kata penting dari suatu dokumen sehinggga dapat dibuang (Putranti, N. D. dan Winarko,E., 2014). Contoh *stopword* adalah "yang", "dan", "di", "dari", dan seterusnya. Dalam filtrasi ini menggunakan *stopword* agar kata-kata yang kurang penting dan sering muncul dalam suatu dokumen dibuang sehingga hanya menyisakan kata-kata yang penting dan mempunyai arti untuk diproses ke tahap selanjutnya.

2.2.7 Data Splitting

Seperti namanya merujuk, tahap Data Splitting akan membagi dataset yang telah melalui preprocessing & cleaning menjadi dua bagian yaitu data train serta data test dengan proporsi tertentu. Data train akan digunakan untuk mencari nilai fitness dari model sedangkan data test digunakan untuk mengevaluasi hasil nilai fitness model tersebut. Seperti yang umunya diketahui, train/test split membagi dataset menjadi train set dan test set, dimana setelah dilakukan pelatihan menggunakan train set, model hasil pembelajaran akan dievaluasi dengan test set . Pada penelitian

yang dilakukan, peneliti menetapkan rasio 8:2 sebagai rasio pembagian dalam data splitting, didasari dengan Pareto Principle, dimana dikutip dalam aturan tersebut, *"..., that 20% of the factors in most situations account for 80% of what happens.*.

2.2.8 Pembobotan Kata

Pembobotan kata atau *term weighting* merupakan salah satu tahapan yang perlu diperhatikan dalam mencari informasi dari koleksi dokumen yang heterogen.Dalam dokumen umunya terdapat kata, frasa, atau unit indeks lainnya yang menunjukan konteks dari dokumen tersebut, hal inilah yang disebut *term*. *Term weighting* digunakan untuk memberikan indikator dari setiap kata sesuai dengan tingkat kepentingan masing-masing kata dalam dokumen (Zafikri, A., 2008). Salahsatu metode pembobotan kata terbaru yang paling banyak digunakan adalah metode *Term Frequency – Inverse Document Frequency* (TF-IDF). Dalam TF-IDF, perhitungan bobot *term* dari sebuah dokumen dilakukan dengan menghitung masing-masing nilai *Term Frequency* dan *Inverse Document Frequency*.

2.2.8.1 Term Frequency (TF)

Term Frequency merupakan faktor yang menentukan perhitungan bobot *term* berdasarkan jumlah dan bentuk kemunculan kata pada dokumen. Pada dasarnya, dapat dikatakan bahwa semakin besar nilai jumlah kemunculan suatu *term*, maka semakin besar juga nilai bobot *term* tersebut dalam dokumen. Menurut Zafikri (2008), perhitungan dalam pembobotan nilai *Term Frequency* dapat dilakukan dengan beberapa cara sebagai berikut.

a) TF biner, pemberian bobot *term* dilihat berdasarkan ada tidaknya suatu katadalam dokumen. Jika terdapat kata tersebut maka diberi nilai satu, jika tidakdiberi nilai nol.
b) TF murni (*raw* TF), pemberian bobot *term* dilihat berdasarkan jumlah kemunculan suatu kata dalam dokumen. Misal, jika kata tersebut muncul tiga kali maka diberi bobot tiga.
c) TF logaritmit, pemberian bobot *term* pada dokumen yang

memiliki sedikit kata dalam *query*, namun mempunyai frekuensi yang tinggi.

$$\mathrm{tf} = 1 + \log(tf) \quad (2.1)$$

d) TF normalisasi, pemberian bobot *term* diperoleh dengan membandingkan frekuensi sebuah kata dengan jumlah seluruh kata dalam dokumen

$$tf = 0{,}5 + 0{,}05x \left(\frac{tf}{\max tf}\right) \quad (2.2)$$

2.2.8.2 Term Frequency (TF)

Inverse Document Frequency merupakan proses mengurangi dominasi *term*umum yang sering muncul dalam dokumen. *Term* umum perlu dihilangkan karena sering menyebabkan analisis menjadi kurang maksimal. Selain itu, IDF juga bertujuan untuk menjaga faktor kejarangmunculan kata (*term scarcity*). Pembobotan dalam IDF dilakukan dengan menghitung nilai faktor kebalikan dari frekuensi dokumen yang mempunyai suatu kata. Adapun perhitungan nilai *Inverse Document Frequency* dapat dilakukan dengan persamaan (2.3) berikut:

$$df_j = \log\left(\frac{D}{df_j}\right) \quad (2.3)$$

di mana

D = jumlah keseluruhan dokumen

df_j = jumlah dokumen yang memiliki *term* t_j

Adapun nilai TF-IDF diperoleh dari perkalian nilai *Term Frequency* dengan nilai *Inverse Document Frequency*. Maka pada perhitungan TF-IDF untuk *raw* TFmenggunakan persamaan (2.4) berikut.

$$w_{ij} = tf_{ij} \times df_{ij} \quad (2.4)$$

di mana

w_{ij} = bobot *term* t_j terhadap d_i

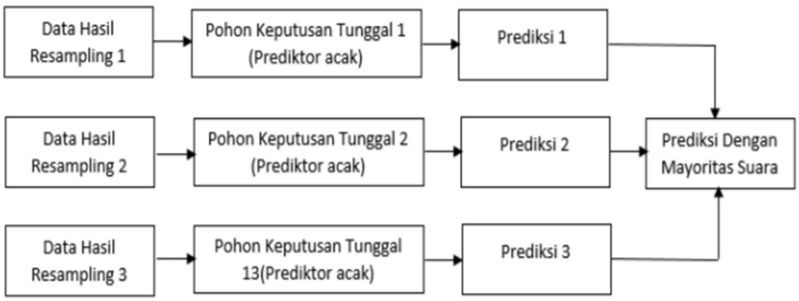
tf_{ij} = jumlah kemunculan *term* t_j dalam dokumen d_i

2.2.9 Metode Random Forest

Salah satu metode yang digunakan untuk pengklasiifikasian dan regresi adalah Random Forest. Metode Random Forset merupakan sebuah esamble (kumpulan) metode pembelajaran menggunakan pohon keputusan sebagai base classifier yang dibangun dan dikombinasikan, adapun beberapa aspek penting dalam metode Random Forest diantaranya melakukan bootstrap sampling untuk membangun pohon prediksi, masing-masing pohon keputusan memprediksi dengan prediktor acak dan Random Forest sendiri melakukan prediksi dengan mengkombinasikan hasil dari setiap pohon keputusan dengan cara majorty vote untuk klasifikasi dan juga rata-rata untuk regresi.

Adapun Random Forest dapat dibangun menggunakan bagging dengan pemilihan atribut acak. Metode CART (Clasification and Regresion Tree) sendiri dapat digunakan untuk menumbuhkan pohon keputusan, pohon keputusan tersebut tumbuh hingga ukuran maksimum dan tidak akan dipangkas sehingga dihasilkan kumpulan pohon yang kemudian disebut forest. Adapun kelebihan dari metode Forest sebagai berikut:

a) Hasil Akurasi bagus
b) Relatif kuat terhadap outliers dan noise
c) Lebih cepat dibandingkan dengan bagging dan boosting
d) Sifatnya yang sederhana dan mudah dipararelkan Selanjutnya gambaran algoritma sederhana Random Forest dapat dilihat sebagai berikut:



Gambar 2.1 Algoritma Random Forest

Commented [AFWMS15]: Semua keterangan gambar maupun table hilangkan titik terakhirnya
Missal : 2.1

2.2.10 Confusion Matrix

Confusion matrix adalah sebuah tabel yang menyatakan jumlah data uji yang benar diklasifikasikan dan jumlah data uji yang salah diklasifikasikan. Contoh *confusion matrix* untuk klasifikasi biner ditunjukkan pada Tabel 2.2 berikut:

Tabel 2.2 Confusion Matrix untuk klasifikasi biner

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

Keterangan untuk Tabel 2.2 dinyatakan sebagai berikut:

a) *True Posstive* (TP), yaitu jumlah dokumen dari kelas 1 yang benar dandiklasifikan sebagai kelas 1.
b) *True Negatif* (TN), yaitu jumlah dokumen dari kelas 0 yang benardiklasifikasikan sebagai kelas 0.
c) *False Positive* (FP), yaitu jumlah dokumen dari kelas 0 yang salahdiklasifikasikan sebagai kelas 1.
d) *False Negatif* (FN) yaitu jumlah dokumen dari kelas 1 yang salahdiklasifikasikan sebagai kelas 0.

Perhitungan akurasi dinyatakan dalam persamaan (2.5) berikut.

$$Akurasi = \frac{TP+TN}{TP+FN+FP+T} \times 100\%$$

(2.5)

$$Presisi = \frac{TP}{TP+} \times 100\%$$

(2.6)

$$Recall = \frac{TP}{TP+FN} \times 100\%$$

(2.7)

$$F-1\ Score = \frac{2 \times Recall \times Presisi}{Recall+Presis}$$

(2.8)

Kemudian dapat juga menghitung APER (*Apparent Error Rate*) atau disebut juga sebagai laju *error*, yang mana merupakan ukuran evaluasi yangdigunakan untuk melihat peluang kesalahan klasifikasi yang dihasilkan oleh suatu fungsi klasifikasi. Semakin kecil nilai APER, maka hasil pengklasifikasian semakin baik Formulasi untuk menghitung APER dituliskan dalam persamaan (2.6) berikut.

$$APER = \frac{TP+TN}{TP+FN+FP+T} \times 100\%$$

(2.9)

2.2.11 Asosiasi Teks

Asosiasi teks diperoleh dengan melakuan pendekatan pada perhitungan nilaikorelasi. Pada umumnya, nilai korelasi digunakan dalam menyatakan hubungan duaatau lebih veriabel kuantitatif. Namun pada asosiasi teks, nilai korelasi dimaknai sebagai keeratan hubungan antar dua atau lebih variabel kualitatif . Korelasi bertujuan untuk menemukan tingkat hubungan antara variabel bebas (X) dan variabel bebas (Y), dalam

ketentuan data memiliki syarat-syarat tertentu . Perhitungan nilai korelasi pada asosiasi teks menggunakan persamaan (2.7) berikut.

$$r = \frac{n\sum \omega_i y_i - (\sum \omega_i)\cdot(\sum y_i)}{\sqrt{\{n\sum {\omega_i}^2 - (\sum \omega_i)^2\}\cdot\{n\sum {y_i}^2 - (\sum y_i)^2\}}}$$

(2.10)

dengan

r = nilai korelasi antara variabel ω dan variabel y

n = banyaknya pasangan data ω dan y

$\sum\omega_i$ = jumlah nilai pada variabel ω; i = 1, 2, 3, …, n

$\sum y_i$ = jumlah nilai pada variabel y

$\sum\omega_i^2$ = kuadrat dari jumlah nilai pada variabel ω

$\sum y_i^2$ = kuadrat dari jumlah nilai pada variabel y

$\sum\omega_i \cdot \sum y_i$ = total dari hasil perkalian antara nilai variabel ω dan variabel y

Dalam perhitungan asosiasi teks, pertama-tama data teks ditransformasikan ke dalam *document-term matrix* (dtm). Adapaun simulasi perhitungan dilakukan pada enam data sebagai berikut.

kata1	kata2	kata3	kata4	kata5
kata1	kata2	kata3	kata4	kata5
kata1	kata2	kata3	kata4	kata5
kata1	kata2	kata3	kata4	kata5
kata1	kata2	kata3	kata4	kata5

Kemudian kelima kata tersebut diubah menjadi *document term matrix* (dtm).

Docs	kata1	kata2	kata3	kata4	kata5
1	1	2	3	4	5
2	2	3	4	5	6
3	3	4	5	6	7

4	4	5	6	7	8
5	5	6	7	8	9

Setelah diperoleh nilai *document term matrix,* selanjutnya dilakukan perhitungan nilai asosiasi. Nilai asosiasi diperoleh dengan menghitung rumus korelasi seperti pada simulasi kata 2 dan kata 4 berikut.

	ω_i	y_i	${\omega_i}^2$	${y_i}^2$	$\omega_i y_i$
Docs	kata2	kata4	kata2^2	kata4^2	kata2Xkata4
1	2	4	4	16	8
2	3	5	9	25	15
3	4	6	16	36	24
4	5	7	25	49	35
5	6	8	36	64	48
Total	20	30	90	190	130

$$r = \frac{n\sum \omega_i\, y_i - (\sum \omega_i)\cdot(\sum y_i)}{\sqrt{\{n\sum {\omega_i}^2 - (\sum \omega_i)^2\}\cdot\{n\sum {y_i}^2 - (\sum y_i)^2\}}}$$

$$r = \frac{(6\times 130) - (20\times 30)}{\sqrt{\{(60\times 90) - 20^2\}\cdot\{(6\times 190) - 30^2\}}}$$

$$r = \frac{180}{\sqrt{140\cdot 240}} = \frac{180}{33.600} = \frac{180}{183{,}3} = 0{,}98$$

Jadi, nilai korelasi kata 2 dan kata 4 sebesar 0,98. Hal ini menunjukkan bahwabesarnya asosiasi atau hubungan antara kata 2 dan kata 4 sebesar 98%.

BAB III
GAMBARAN UMUM INSTANSI

Commented [MR16]: Gambar instansi adalah logo instansi

3.1 Profil Instansi

Gambar 3.1 Logo Telkomsel

Telkomsel adalah sebuah perusahaan yang bergerak dalam bidang jasa pelayanan telekomuikasi selular berbasis GSM. Telkomsel merupakan singkatan dari "Telekomunikasi Selular" dengan produk-produknya adalah kartu *HALO,* sim*PATI* dan kartuAS.

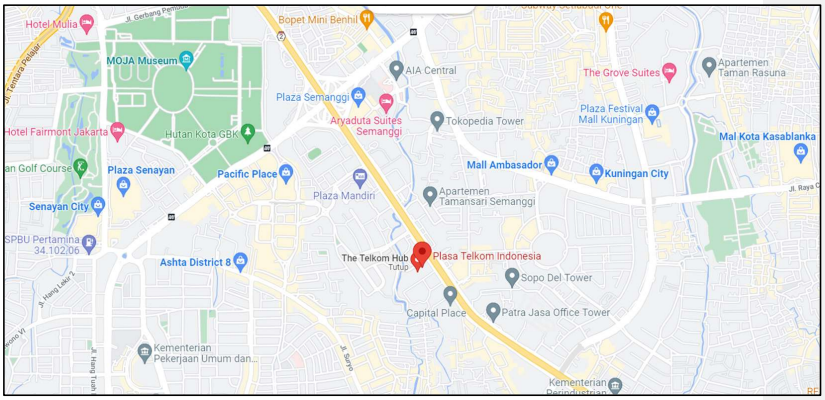
Telekomunikasi selular GSM di Indonesia berawal dari pemerintah yang meminta PT Telkom untuk melakukan *pilot project* di Batam dan Bintan pada bulan November 1993. Pada tanggal 31 Desember 1993, proyek tersebut dapat beroperasi.

Pada tanggal 26 Mei 1995, atas keputusan Menteri Pariwisata, pos, dan Telekomunikasi (Menparpostel) dan Menteri Keuangan (Menkeu), berdirilah PT Telekomunikasi Selular (Telkomsel) sebagai perusahaan jasa penyedia layanan telekomunikasi selular GSM kedua di Indonesia setelah PT. Satelit Indonesia (Satelindo) yang berdiri pada tanggal 29 Januari 1993. Pada awal berdirinya, kepemilikan saham Telkomsel dimiliki oleh Telkom sebesar 51,0% dan Indosat sebesar 49%.

Dengan semakin berkembanganya bisnis telekomunikasi khususnya telekomunikasi selular dan keinginan yang kuat untuk menjadikan PT Telkomsel sebagai operator telepon seluler yang bertaraf internasional dengan produk yang mempunyai standar internasional, maka melalui seleksi dan proses tender telekomunikasi yang ketat dan

transparan, akhirnya terpilih dua perusahaan telekomunikasi untuk diajak bekerjasama, yaitu KPN Royal Dutch Telecom (KPN) yang merupakan perusahaan telekomunikasi dari Belanda sebagai mitra asing dan PT Sedco Megacell Asia sebagai mitra lokal. KPN membeli 17,28% sedangkan PT. Sedco Megacell Asia membeli 5% saham. KPN dan Sedco masuk ke Telkomsel sehingga komposisi kepemilikan saham Telkomsel adalah Telkom 42,72%, Indosat 35%, KPN 17,28% dan Secdo 5%. Dengan masuknya dua mitra tersebut, maka status Telkomsel berubah dari perusahaan Penanaman Modal Dalam Negeri (PMDN) menjadi Penanaman Modal Asing (PMA). Pada tahun 1999 diterbitkannya Undang-Undang No. 36/1999 tentang Telekomunikasi yang berlaku efektif sejak tanggal 8 September 2008 dan antara lain berisi penghapusan monopoli penyelenggaraan telekomunikasi. Pada tahun 2001 Telkom membeli 35% saham Telkomsel dari Indosat sebagai bagian implementasi restukturisasi industri jasa telekomunikasi di Indonesia, yang ditandai dengan penghapusan kepemilikan bersama dan kepemilikan silang antara Telkom dengan Indosat. Setelah transaksi ini, Telkom menguasai 77,72% saham Telkomsel. Telkom membeli 90,32% saham Dayamitra dan mengkonsolidasikan laporan keuangan Dayamitra ke dalam laporan keuangan Telkom. Telkom membeli seluruh saham Pramindo melalui 3 tahap, yaitu 30% saham pada saat ditandatanganinya perjanjian jual beli tanggal 15 Agustus 2002, 15% pada tanggal30 September 2003 dan sisa 55% saham pada tanggal 31 Desember 2004. Telkom menjual saham Telkomsel kepada Singapore Telecom Mobile Pte. Ltd (SingTel) sehingga setelah penjualan saham ini Telkom memiliki 65% saham Telkomsel dan 35% sisanya dimiliki oleh SingTel.

Adapun alamat dari Telkomsel adalah Telkom Landmark Tower, Jl. Jendral Gatot Subroto Kav. 52 RT.6/RW.1, Kuningan Barat, Mampang Prapatan Jakarta Selatan, DKI Jakarta, 12710 Indonesia



Gambar 3.2 Alamat Telkomsel terlihat dari Maps

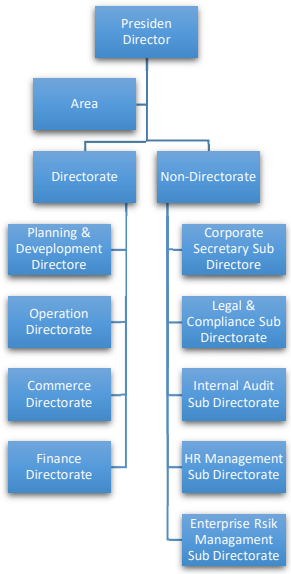
3.1.1 Visi, Misi dan Struktur Organisasi

Visi PT Telkomsel adalah menjadi penyedia layanan *mobile lifestyle* terbaik di Indonesia (*The best mobile lifestyle provider in region*. Sedangkan Misi PT Telkomsel adalah Memberikan pelayanan dan solusi komunikasi yang sesuai dengan harapan *customer,* memberikan nilai tambah kepada para *stakeholder* dan memberikan kontribusi tehadap pembangunan ekonomi bangsa. (*Deliver mobile life style – service and solution in excellent way that exceed customer expectation, create value for all stakeholders, and the economic development for nation*.

Secara struktural, Telkomsel terbagi menjadi 3 unit kerja yaitu *Directorate, Non-Directorate* dan Area. Ketiga unit kerja ini kemudian bertanggungjawab langsung kepada Direksi. Area merupakan unit kerja yangberada di wilayah regional. Telkomsel terbagi menjadi 4 Area yaitu Area I atau disebut dengan Area Sumatera yang meliputi seluruh wilayah Pulau Sumatera. Area 2 atau Area Jawa Barat dan Banten yang wilayah kerjanya meliputi Jawa Barat dan Banten. Area 3 adalah Area Jawa-Bali Nusra yang meliputi wilayah Jawa Tengah, Jawa Timur dan Kepulauan Nusa Tenggara. Dan yang terakhir adalah area 4 yaitu Area Pamasuka yang meliputi Papua, Maluku, Sulawesi dan Kalimantan.

Sedangkan *Directorate* merupakan unit kerja yang dikepalai oleh direktur. Direktur bertanggungjawab kepada Direktur Utama. *Directorate*

membawahi unit kerja yang terdiri dari *Planning & Development Directorate, Operation Directorate, Commerce Directorate, Finance Directorate*. Untuk *Sub- Directorate,* merupakan unit kerja yang tanggungjawabnya dipegang oleh *Vice President* bukan oleh Direktur. *Vice Precident* ini yang bertanggungjawab langsung kepada Direktur Utama. Unit kerja *Sub-Directorate* terdiri dari*Corporate Secretary Sub Directorate, Sub Directorate, Legal & Compliance Sub Directorate, Internal Audit Sub Directorate, HR Management Sub Directorate, Enterprise Risk Management Sub Directorate.* Untuk lebih jelasnya, struktural Telkomsel dapat dilihat pada tabel di bawah ini:



Gambar 3.3 Struktur Organisasi

3.1.2 Layanan dan Produk

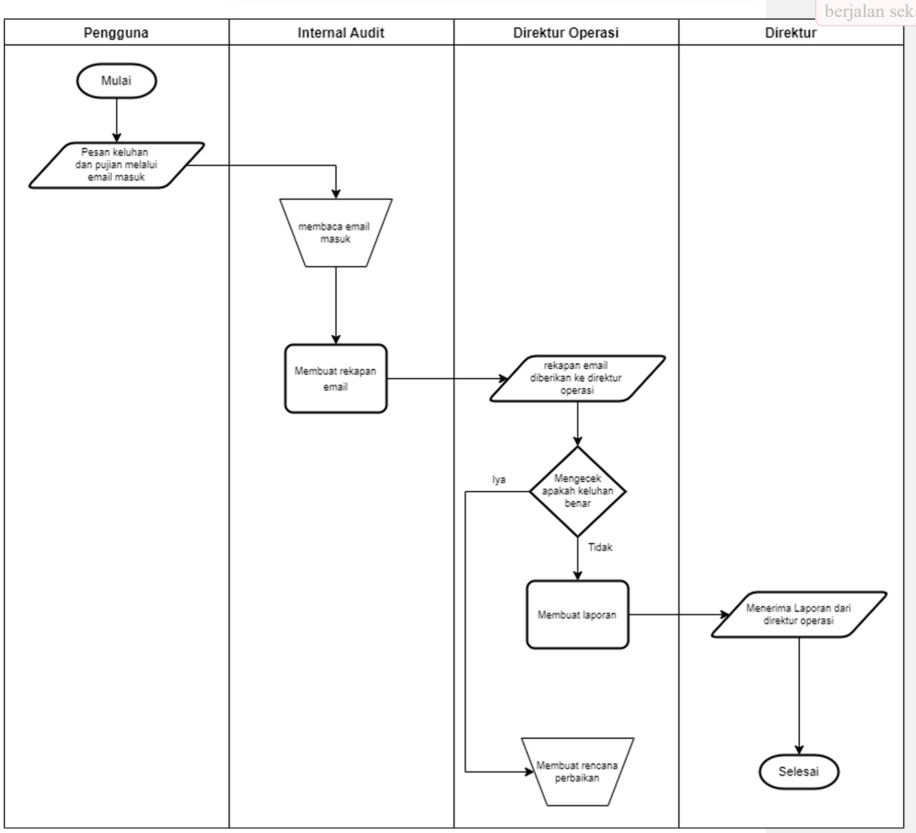
Telkomsel adalah perusahaan yang bergerak dalam bidang penyedia (*provider*) jasa layanan telekomunikasi selular berbasis teknologi GSM (*Global System for Mobile Communications*) yang menggunakan atau mengaplikasikan teknologi *GSM 900Mhz* dan *GSM 1800 Mhz (Dual Band)* yang pertama di Indonesia.

Layanan telepon selular bersistem GSM dipilih oleh Telkomsel karena sistem telekomunikasi seluler sebelumnya bersistem analog (seperti

AMPS dan NMT) yang tidak memberikan jaminan keamanan pembicaraan dan wilayah cakupannya yang terbatas. Teknologi seluler GSM pun jauh lebih unggul karena mampu menghasilkan kualitas suara jernih dan ditunjang dengan semakin bertambah luasnya jangkauan (*coverage area*) dari tahun ke tahun.

3.1.3 Aturan Bisnis

Commented [MR17]: Flowchart bagan aturan bisnis yang berjalan sekarang



Gambar 3.4 Flowchart bagan aturan bisnis yang berjalan sekarang

PT Telkomsel (Persero) sebagai Badan Usaha Milik Negara (BUMN) yang bergerak di bidang jasa layanan teknologi informasi dan komunikasi (TIK) dan jaringan telekomunikasi di Indonesia, bertekad untuk memberikan pelayanan jasa jaringan telekomunikasi yang terbaik dan memenuhi standar jaringan telekomunikasi yang dapat diterima dunia international dan mewujudkan hal itu dengan bertumpu pada kapasitas

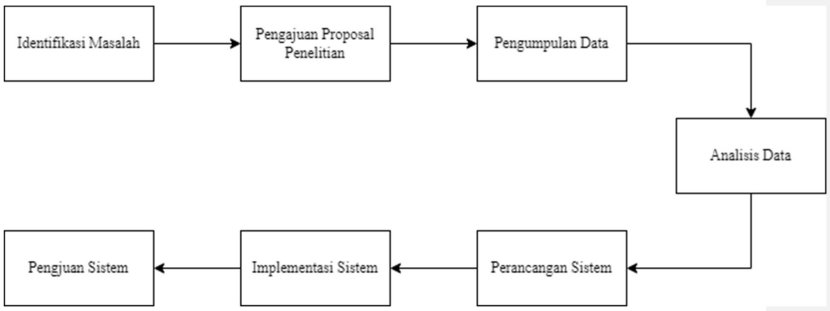
seluruh warganya. Dalam menjalankan bisnisnya, Telkomsel bertekad bekerja dengan semangat untuk selalu menghasilkan produk dan pelayanan yang terbaik serta memperlakukan pelanggan, mitra usaha, dan pemasok dengan adil tanpa membeda-bedakannya.

Dalam rangka menjaga agar tetap ada konsistensi dalam penyelenggaraan perusahaan yang baik (Good Corporate Governance), manajemen Telkomsel bertekad untuk menumbuh kembangkan kebiasaan dan tata pergaulan profesional yang baik dan sekaligus mencerminkan jati diri Telkomsel yang dapat kita banggakan bersama. Usaha ini juga merupakan perwujudan dari kesungguhan hati warga Telkomsel untuk bekerja dan berusaha selaras dengan falsafah, visi, misi, dan tata nilai perusahaan yang sudah disepakati bersama. Semua ini akan dijalankan dengan tetap mengacu pada aspirasi untuk menciptakan nilai yang maksimal bagi bangsa dan negara Indonesia.

Manajemen Telkomsel juga bertekad untuk menyelenggarakan perusahaan dengan mengajak seluruh anggota Telkomsel dan semua pihak yang peduli dengan kemajuan perusahaan ini, dapat menjaga perusahaan ini agar tetap berkiprah secara bertanggung jawab. Keterbukaan dan partisipasi ini akan dijalankan dengan prinsip bahwa informasi perusahaan dapat diakses dan diperoleh dengan mudah oleh masyarakat dan semua pihak yang berhak, tanpa mengabaikan prinsip kerahasiaan informasi tersebut. Sebaliknya, manajemen perusahaan juga senantiasa membuka diri bagi semua masukan dan saran dari lingkunan internal dan eksternal perusahaan.

3.2 Tahap Penyelesaian Masalah

Tahapan penyelesaian masalah didefinisikan sebagai pedoman penulis dalam melaksanakan penelitian. Hal ini bertujuan agar hasil yang dicapai tidak menyimpang dan sesuai dengan tujuan yang telah ditetapkan. Adapun diagram alirpenelitian dapat dilihat pada Gambar 3.5 berikut.



Gambar 3.5 Tahap Penyelesaian Masalah

Berdasarkan Gambar 3.5, dapat dijelaskan deskripsi atau uraian dari setiap tahap penelitian sebagai berikut.

a) Indetifikasi Masalah

Commented [MR18]: Ini bukan sub-bab dari prosedur

Pada tahap ini, penulis melakukan persiapan seperti menentukan judul penelitian, rumusan masalah, batasan masalah, tujuan, serta studi pustaka untuk mencari teori-teori yang dapat membantu dalam penyelesaian masalah yang diteliti.

b) Pengajuan Proposal Penelitian

Pada tahap ini, penulis membuat proposal penelitian untuk persetujuan topik yang diangkat kepada universitas sebagai syarat administratif penelitian

c) Pengumpulan Data

Pada tahap pengumpulan data, komponen yang diperlukan yaitu data ulasan pengguna aplikasi MyTelkomsel. Data ulasan diperoleh dari situs Google Play Store menggunakan teknik *web scraping* dengan format *file* CSV. Setelah memperoleh data, selanjutnya data tersebut akan diolah dalam tahap analisis data.

d) Analisis Data

Tahap analisis data yaitu proses menerjemahkan tujuan ke dalam spesifikasikebutuhan sistem. Analisis dilakukan agar dapat mengetahui permasalahan yang muncul pada sistem yang sedang berjalan. Sehingga dapat dibangun sistem yang lebih baik dengan menerapkan solusi dari permasalahan yang terjadi.

e) Perancangan Sistem

Pada tahap ini, dilakukan perancangan proses atau alur sistem dan perancangan sistem yang akan dibangun. Tahap ini bertujuan untuk menjelaskan proses-proses yang akan dilakukan dan urutan proses yang akan dilakukan. Perancangan sistem meliputi blok diagram alur sistem dan *flowchart* sistem yang akan dibangun.

f) Impelementasi Sistem

Pada tahap implementasi sistem, dilakukan penerapan terhadap rancangan-rancangan yang telah dibuat pada tahap sebelumnya, sehingga dihasilkan suatu luaran berupa sistem yang dapat melakukan klasifikasi data ulasan pengguna an aplikasi MyTelkomsel. Sistem ini diimplementasikan menggunakan bahasa pemrograman Python dan Jupyter Notebook.

g) Pengujian Sistem

Pengujian sistem dilakukan untuk menjelaskan algoritma *Random Forest* yang mana mengitung probabilitas data dari hasil klasifikasi data, sehingga menghasilkan luaran berupa klasifikasi yang akurat. Pengujian dilakukan untuk mengetahui tingkat akurasi dengan menggunakan *confusion matrix*, sebuah matrik dari prediksi yang akan dibandingkan dengan kelas yang asli dari data input.*Confusion matrix* menghasilkan data angka dari hasil pengujian sistem. Hasilpengujian digunakan untuk melihat nilai akurasi dari sistem yang dibangun.

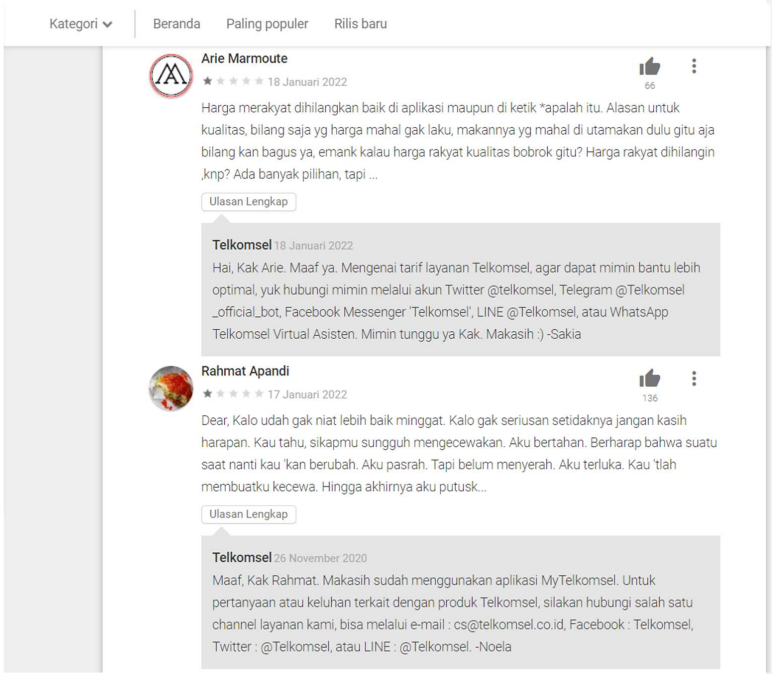
3.3 Bahan dan Data

3.3.1 Data yang Diperoleh

Commented [AFWMS19]: Seharusnya sub bab 3.3.1

Data ulasan pengguna aplikasi MyTelkomsel yang digunakan pada penelitianini berupa data sekunder yang diperoleh dari situs Google Play Store. *Dataset* ini terdiri atas lima kategori meliputi: bintang 1 (sangat buruk), bintang 2 (buruk), bintang 3 (netral), bintang 4 (baik), dan bintang 5 (sangat baik). Total ulasan yang sebenarnya ialah lebih dari 6.000.000 data, namun pada penelitian ini hanya menggunakan 5.000 data dengan tujuan efisiensi sistem. Selain itu, dataset yang digunakan juga tidak menyertakan kategori bintang 3 (netral) agar sistem dapat menghindari

bias pada saat melakukan pembobotan kata. Contoh data ulasan aplikasi MyTelkomsel di situs Google Play Store terlihat pada Gambar 3.6 berikut.



Gambar 3.6 Data Ulasan Aplikasi MyTelkomsel

3.3.2 Prosedur Pengumpulan Data

Commented [AFWMS20]: 3.3.2

Dataset yang digunakan pada penelitian ini adalah data ulasan aplikasi MyTelkomsel yang diambil dari situs Google Play Store dengan alamat URL https://play.google.com/store/apps/details?id=com.telkomsel.telkomselcm. *Dataset* diperoleh menggunakan teknik *web scraping* dengan bantuan *library* Google Play Scraper dan *cloud environment* Google Colaboratory.

Tabel 3.1 Contoh review pengguna Aplikasi MyTelkomsel

No.	Ulasan
1.	Najis, maling banget sih ini Telkomsel, saya tidak menyambungkan data Telkomsel saya tapi kenapa pulsa saya kesedot terus..!! Balikin pulsa saya parah nih, lain kali saya akan screenshot buat bukti kesedot nya pulsa saya, alasan nya dikirim lewat SMS, "anda memakai data Telkomsel", padahal kaga..!! Najis tinggal di Indonesia, pulsa aja disedot terus sama Telkom :(
2.	Alasannya; 1. Harga paket mahal dan terbagi-bagi. 2. Sekalinya listrik padam, signal 4G langsung hilang. 3. Kalau main game, ping nya di atas 80. 4. Sejak mengalami transformasi mulai dari logo dan lain2, harga paket melonjak tidak wajar. (Saya pernah beli paketan saat promo seharga 450 ribu mendapatkan 3000 GB. Sekarang dengan harga yg sama, dapet nya di bawah 1000 GB)
3.	Buat pihak Telkomsel inisiatif dong kaya operator lain,ada fitur lock pulsa,jadi kalo paketan abis nge dadak pulsa gak ke ambil,yang lebih parah nya lagi,ini saya data internet pake katru lain,dan saya isi pulsa Telkomsel selang beberapa jam belum juga kepake pulsa udah ke ambil lagi tanpa sepengetahuan saya,saya gak berlangganan apapun, NSP,dll,gak pernah langganan sama sekali,kok langsung keambil gitu aja? Ya saya geram dong, Mohon tindakannya,untuk saya bisa lebih enak pake kartu tsel.

Commented [MR21]: Keterangan tabel, spasi paragraf 1

BAB IV
ANALISIS DAN DESIGN SISTEM

4.1 Analisis Sistem

Dalam membangun sebuah sistem, diperlukan adanya analisis sistem. Analisis sistem dilakukan bertujuan untuk mengidentifikasi kebutuhan-kebutuhan pada sistem, kebutuhan ini meliputi kebutuhan fungsional maupun kebutuhannonfungsional. Berikut adalah uraian analisis sistem dalam membangun sistem klasifikasi sentimen pada ulasan pengguna aplikasi MyTelkomsel.

4.1.1 Analisis Kebutuhan Fungsional

Analisis sistem digunakan untuk menganalisis dan mengidentifikasi permasalahan untuk menghasilkan sistem sesuai dengan yang diinginkan. Sistem yang dibangun akan digunakan untuk mengklasifikasi sentimen pada ulasan pengguna aplikasi MyTelkomsel menggunakan algoritma Random Forest Classifier. Data ulasan yang akan diproses diperoleh dari situs Google Play Store karena situs tersebut menyediakan data khusus untuk ulasan-ulasan dari berbagai aplikasi Android. Klasifikasi sentimen dilakukan berdasarkan nilai bobot kata positif dan negatif terhadap data ulasan yang telah dibersihkan melalui proses preprocessing.

4.1.2 Analisis Kebutuhan Nonfungsional

Commented [AFWMS22]: 4.1.2

Analisis kebutuhan sistem secara nonfungsional adalah analisis mengenai kebutuhan pendukung sistem yang akan dibuat, bertujuan untuk memenuhi kebutuhan nonfungsional pada sistem. Kebutuhan secara nonfungsional tersebut meliputi kebutuhan *hardware* (perangkat keras) dan *software* (perangkat lunak) yang digunakan dalam penelitian sebagai berikut.

1. Perangkat keras (*Hardware*)

Perangkat keras yang digunakan untuk merancang dan mengimplementasikan sistem yang dibuat yaitu laptop dengan spesifikasi:

a) LENOVO IP3-14ALC6 GQID/G5ID

b) AMD Ryzen™ 3 5300U
c) AMD Radeon Graphics
d) RAM 8GB DDR4 3200MHz
e) 512GB SSD NVMe

2. Perangkat Lunak (Soft*ware*)

Perangkat lunak yang digunakan untuk merancang dan mengimplementasikan sistem yang dibuat yaitu laptop dengan spesifikasi:

a) Jupyter Notebook

Jupyter Notebook adalah *tools* untuk bahasa pemrograman Julia, Python, dan R yang mana berfungsi dalam menuliskan *script* analisis sentimen pada ulasan pengguna aplikasi MyTelkomsel.

b) Bahasa pemrograman Python versi 3.6 64-bit

Bahasa pemrograman Python digunakan sebagai basis bahasa pemrograman untuk membangun sistem klasifikasi sentimen pada ulasan pengguna aplikasi MyTelkomsel. Bahasa Python dipilih karena mendukung komputasi statistik dan memiliki banyak *library* yang dibutuhkan dalam membangun sistem klasifikasi sentimen pada ulasan pengguna aplikasi MyTelkomsel.

c) Windows 10 Education versi 10.0 64-bit

Windows 10 Education adalah sistem operasi yang diinstal dalam laptop dan digunakan pada penelitian ini.

4.2 Desain Sistem

Pada tahap ini, penulis melakukan desain atau perancangan sistem meliputiblok diagram alur, *flowchart*, dan *wireframe* sistem. Tujuan dari perancangan ini yaitu dapat memberikan gambaran kepada pengguna tentang sistem yang diusulkandan memberikan ilustrasi dalam pembuatan sistem nantinya.

4.2.1 Desain Logik

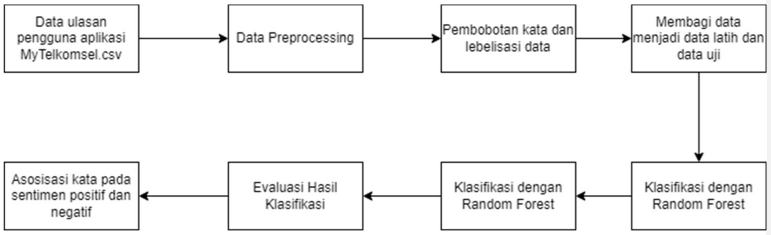
Perancangan sistem dapat diartikan sebagai gambaran atau sketsa

Commented [AFWMS23]: 4.2.1
Cek kembali penomoran sub bab setelah ini

dari alur proses sistem pengolahan *dataset*. Dalam perancangan suatu sistem terdapat proses aliran sistem dan *flowchart* sistem. Perancangan sistem ini dilakukan sebelum proses implementasi sistem. Berikut adalah rancangan blok diagram alur dan*flowchart* yang dibuat sesuai dengan sistem yang akan dibangun.

a) Blok Diagram Alur

Perangcangan sistem berguna untuk menjelaskan bagaimana alur sistem yang akan dibangun. Perancangan blok diagram alur sistem dapat dilihat pada Gambar 4.1.

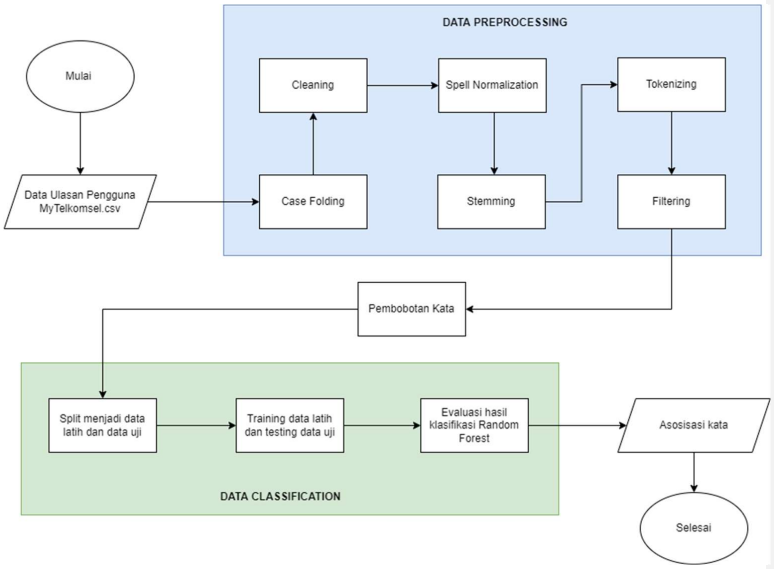


Gambar 4.1 Blog Diagram Alur Sistem

b) Flowchart

Flowchart akan menjabarkan algoritma yang digunakan pada pembuatan sistem pengklasifikasi sentimen pada ulasan pengguna aplikasi MyTelkomsel Mobilemenggunakan *Random Forest*. Algoritma dimulai dari memasukkan data ulasan pengguna aplikasi MyTelkomsel dengan format *file* CSV, kemudian sistem mulai melakukan *preprocessing* untuk membersihkan danmenormalisasi data. Hasil dari proses *processing* akan dilanjutkan ke proseslabelisasi data. Dalam tahap ini, data ulasan diberi nilai bobot pada tiap kataberdasarkan sentimen positif dan negatif. Lalu dari nilai bobot tersebut akanditentukan apakah sebuah ulasan masuk ke sentimen positif atau sentimen negatif. Setelah itu, data hasil labelisasi akan dibagi menjadi data latih dan data uji. Data latih menjadi model atau referensi untuk mengklasifikasikan sentimen ulasan menjadi positif atau negatif. Kemudian dilakukan pengujian terhadap data uji untuk mengetahui seberapa efektif *Random Forest* bekerja. Setelah selesai, sistem akan menampilkan hasil klasifikasi besera nilai akurasi yang didapat. Terakhir, dilakukan asosiasi teks untuk mengetahui informasi penting yang berguna bagi berbagai pihak. *Flowchart* algortima sistem

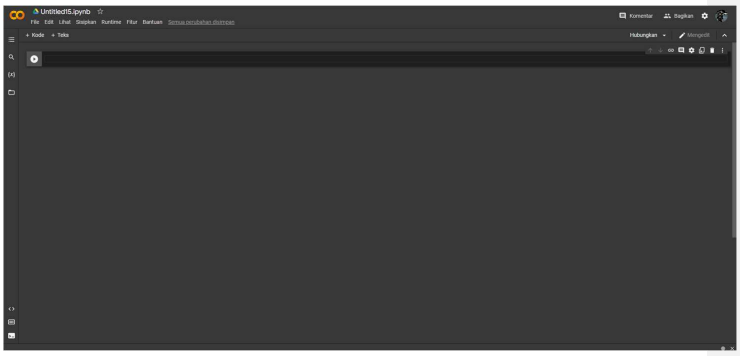
yang dibangun dapat dilihat pada Gambar 4.2.



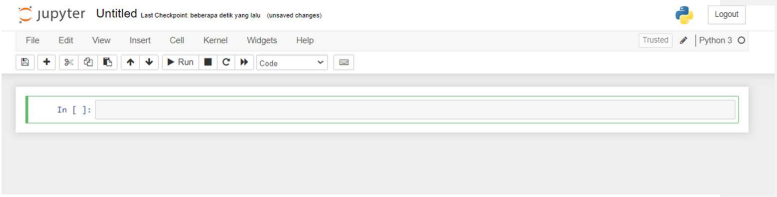
Gambar 4.2 Flowchart Sistem

4.2.2 Desain Fisik

Pada penelitian ini, tidak dilakukan perancangan tampilan antarmuka sistem yang akan dibuat karena sistem berupa analisis yang memuat kode *script* pada Bahasa pemrograman Python. Dalam implementasinya, penulis menggunakan *software* Google Colaboratory dan *software* Jupyter Notebook dengan bantuan *package* sehingga sistem mampu berjalan dengan baik. Pada Gambar 4.3 dan 4.4 berikut merupakan antarmuka dari *Google Colaboratory* dan *Jupyter Notebook*.



Gambar 4.3 Antarmuka Google Colaboratory



Gambar 4.4 Antarmuka Jupyter Notebook

BAB V
IMPLEMENTASI DAN HASIL

Commented [MR24]: Judul bab spasi 1 dan enter

5.1 Implementasi

Implementasi merupakan penerapan terhadap cara kerja sistem yang telah dirancang berdasarkan analisis dan desain sistem yang sudah dibuat. Pada implementasi ini membahas tentang hasil implementasi sistem dan hasil pengujiansistem. Implementasi sistem adalah tahap di mana sistem telah diterapkan dan disiap untuk digunakan. Bab ini akan membahas kode program maupun penjelasandari proses *web scraping* dalam mengumpulkan data, proses *data preprocessing* dalam menormalisasi data, proses labelisasi data untuk pembobotan kata berdasarkan sentimen positif dan negatif, proses *machine learning* atau klasifikasi sentimen, proses evaluasi terhadap hasil klasifikasi, dan asosiasi teks.

5.1.1 Proses Web Scraping

Sebelum melakukan klasifikasi sentimen data ulasan pengguna aplikasi MyTelkomsel, tahap pertama yang perlu dilakukan ialah mengumpulkan data melaluiteknik *web scraping*. Pada tahap ini, bahasa pemrograman Python digunakan karena lebih efektif dalam mengumpulkan data secara cepat. Selain itu, *cloud environment* Google Colaboratory dan *library* Google Play Scraper digunakan dalam proses *web scraping* agar tidak membebani sumber daya komputer penulis. Adapun luaran yang diharapkan pada tahap ini berupa dataset ulasan dengan format*file* CSV karena mudah diolah dan dianalisis. Gambar 5.1 berikut merupakan *source code* dari proses *web scraping* data ulasan aplikasi MyTelkomsel Mobile.

```
!pip install google-play-scraper
import pandas as pd
from tqdm import tqdm
from google_play_scraper import Sort, reviews, app
app_id = ['com.telkomsel.telkomselcm']
app_reviews = []
for app in tqdm(app_id):
    for score in list(range(1, 6)):
        for sort_order in [Sort.MOST_RELEVANT, Sort.NEWEST]:
          rvs, _ = reviews(
            app,
            lang = 'id',
            country = 'id',
            sort = sort_order,
            count= 5000 if score == 3 else 5000,
            filter_score_with = score
          )
          for r in rvs:
            r['sortOrder'] = 'most_relevant' if sort_order == Sort.MOST_RELEVANT else 'newest'
            r['appId'] = app
    app_reviews.extend(rvs)
len(app_reviews)
df = pd.DataFrame(app_reviews)
df
df['score'].value_counts()
df.drop(df.loc[df['score']==3].index, inplace=True)
len(df)
df.drop(['reviewId', 'userImage', 'userName', 'thumbsUpCount', 'reviewCreatedVersion', 'at', 'replyContent', 'repliedAt',
         'sortOrder','appId', 'score'], axis=1, inplace=True)
pd.options.display.max_colwidth=1000
df
df.to_csv("../dataset/raw_dataset.csv", index=None)
```

Gambar 5.1 Source Code Porses Web Scrapping

Commented [MR25]: Gambar digabung saja

Baris *script* ke-1 adalah instruksi untuk menginstal *library* Google Play Scraper di *cloud environment* Google Colaboratory. Baris *script* ke-2 sampai 4 adalah memanggil *library* yang dibutuhkan system antara lain Pandas, TQDM, danGoogle Play Scraper. Baris *script* ke-5 dan 6 adalah mendefinisikan variabel ID aplikasi MyTelkomsel Mobile dan variabel *array* untuk menampung data ulasan dari GooglePlay Store. Baris *script* ke-7 adalah instruksi perulangan untuk menampilkan progress *web scraping* dalam bentuk *progress bar*. Baris *script ke-8* adalah instruksi perulangan agar mampu menampilkan data ulasan dengan *rating* bintang satu sampai lima. Baris *script* ke-9 sampai 17 adalah instruksi perulangan untuk proses*web scraping* yang mana didasarkan pada beberapa parameter seperti ID aplikasi, Bahasa, negara, penyortiran, jumlah ulasan, dan *rating*. Pada awalnya, data hasil *web scraping* dalam tipe data *list of character* yang mana terpisah, sehingga baris *script* ke-18 sampai 21 adalah instruksi perulangan untuk mengumpulkan setiap iterasi data hasil *web scraping* menjadi satu kesatuan *list of character*. Baris *script* ke-22 adalah instruksi untuk menampilkan jumlah data hasil *web scraping*. Baris *script* ke-23 dan 24 adalah mengkonversi *list of character* menjadi

dataframe dan menampilkan *dataframe* tersebut. Baris *script* ke-25 adalah menampilkan jumlah data pada masing-masing *value* dalam kolom *score*. Baris *script* ke-26 adalah instruksi untuk menghapus data dengan nilai *score* sama dengantiga. Hal ini perlu dilakukan untuk menghindari ulasan atau komentar yang netral. Baris *script* ke-27 adalah instruksi untuk menampilkan jumlah data setelah dilakukan penghapusan beberapa data. Baris *script* ke-28 sampai 30 adalah menghapus beberapa kolom yang tidak diperlukan dalam pengolahan data. Baris *script* ke-30 dan ke-31 adalah menampilkan data lima teratas dari *dataframe*. Baris *script* ke-32 adalah instruksi untuk mengkonversi *dataframe* menjadi *file* dengan format CSV.

5.1.2 Proses Data Preprocessing

Sebelum melakukan labelisasi data pada ulasan pengguna aplikasi MyTelkomsel, perlu dilakukan *data preprocessing*. Data ulasan yang diperoleh melalui teknik *web scraping* belum sepenuhnya siap digunakan untuk proses labelisasi datakarena data masih belum terstruktur dengan baik dan terdapat banyak *noise*. Data masih memuat angka, tanda baca, *emoticon*, serta kata-kata yang kurang bermaknauntuk dijadikan fitur. Maka dari itu, perlu dilakukan *data preprocessing* yang bertujuan untuk menyeragamkan bentuk kata, menghilangkan karakter-karakter selain huruf, dan mengurangi jumlah kosakata sehingga data menjadi mudah diolah.Gambar 5.2 sampai 5.4 berikut merupakan *source code* dari proses *data preprocessing* data ulasan aplikasi MyTelkomsel

```python
import pandas as pd
import re
import string
import nltk
nltk.download('punkt')
nltk.download('stopwords')
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory
df = pd.read_csv("../dataset/raw_dataset.csv")
df
case_folding = []
for review in df['content']:
    # Ubah menjadi Lowercase
    case_folding.append(review.lower())
case_folding[0]
proses_cleaning1 = []
for clean in case_folding:
    proses = clean.encode("ASCII", "ignore")
    proses_cleaning1.append(proses.decode())
proses_cleaning1[11]
from string import digits
proses_cleaning2 = []
for clean in proses_cleaning1:
    proses = str.maketrans('', '', digits)
    proses_cleaning2.append(clean.translate(proses))
proses_cleaning2[20]
proses_cleaning3 = []
for clean in proses_cleaning2:
    proses_cleaning3.append(clean.translate(str.maketrans('', '', string.punctuation)))
proses_cleaning3[30]
proses_cleaning4 = []
for clean in proses_cleaning3:
    proses_cleaning4.append(re.sub(r'\s+', ' ', clean, flags=re.I))
proses_cleaning4[40]
proses_cleaning5 = []
for clean in proses_cleaning4:
    proses_cleaning5.append(re.sub(r'\s+[a-zA-Z]\s+', ' ', clean))
proses_cleaning5[40]
proses_cleaning6 = []
for clean in proses_cleaning5:
    proses_cleaning6.append(re.sub(r'\^[a-zA-Z]\s+', ' ', clean))
proses_cleaning6
data_clean = pd.DataFrame(proses_cleaning6, columns=['text'])
data_clean[50:60]
data_clean.to_csv("../dataset/proses_cleaning6.csv", index=None)
```

Gambar 5.2 Source Code Preprocessing data Python

Baris *script* ke-1 sampai 10 adalah instruksi untuk memanggil *library* atau *package* yang dibutuhkan dalam membangun sistem meliputi Pandas, Regular Expression, string, NLTK, Stopwords, punkt, Sastrawi. Baris *script* ke-11 dan 12 adalah membaca *dataset* dan menampilkan data. Baris *script* ke-13 adalah intruksi untuk membuat variabel untuk menampung hasil *case folding*. Baris *script* ke-14 dan 17 adalah instruksi untuk melakukan proses *case folding* dan menampilkan hasilnya. Baris *script* ke-18 adalah intruksi untuk membuat variabel untuk menampung hasil penghapus karakter non-ASCII. Baris *script* ke-19 dan 22 adalah instruksi untuk melakukan proses penghapus karakter non-ASCII dan menampilkan hasilnya. Baris *script* ke-23 adalah intruksi untuk membuat variabel untuk menampung hasil menghapus angka. Baris *script* ke-24 dan

28 adalah instruksi untuk melakukan proses menghapus angka dan menampilkan hasilnya. Baris *script* ke-29 adalah intruksi untuk membuat variabel untuk menampung hasil menghapus tanda baca. Baris *script* ke-30 dan 34 adalah instruksi untuk melakukan proses menghapus tanda baca dan menampilkan hasilnya. Baris *script* ke-35 adalah intruksi untuk membuat variabel untuk menampung hasil menghapus spasi yang berlebihan. Baris *script* ke-36 dan 40 adalah instruksi untuk melakukan proses menghapus spasi yang berlebihan dan menampilkan hasilnya. Baris *script* ke-41 adalah intruksi untuk membuat variabel untuk menampung hasil menghapus 1 karakter. Baris *script* ke-42 dan 46 adalah instruksi untuk melakukan proses menghapus 1 karakter dan menampilkan hasilnya. Baris *script* ke-47 adalah intruksi untuk membuat variabel untuk menampung hasil menghapus 1 karakter dari awal. Baris *script* ke-48 dan 52 adalah instruksi untuk melakukan proses menghapus 1 karakter dari awal dan menampilkan hasilnya. Baris *script* ke-53 dan 54 adalah intruksi untuk mengubah data list menjadi dataframe dan menampilkannya. Baris *script* ke-53 dan 54 adalah intruksi untuk menyimpan dataframe pada proses sebelumnya menjadi csv.

```
1  library(tm)
2  library(textclean)
3  proses_normalisasi <- readLines("../dataset/proses_cleaning6.csv")
4  kamus_slang <- read.csv("../dictonary/colloquial-indonesian-lexicon.csv", sep=",", header=TRUE)
5  hasil_normalisasi <- replace_internet_slang(proses_normalisasi, slang=paste0("\\b",
6                                              kamus_slang$slang, "\\b"),
7                                              replacement=kamus_slang$formal, ignore.case=TRUE)
8  write.csv(hasil_normalisasi, "../dataset/hasil_normalisasi.csv")
```

Gambar 5.3 Source Code Preprocessing data R

Baris *script* ke-1 dan 2 adalah instruksi untuk memanggil *library* atau *package* yang dibutuhkan dalam membangun sistem meliputi TM, TextClean. Baris *script* ke-3 adalah instruksi untuk membaca *dataset*. Baris *script* ke-4 sampai 7 adalah instruksi untuk melakukan proses *spell normalization.* Baris *script* ke-8 adalah intruksi untuk menyimpan hasil proses sebelumnya.

```
hasil_normalisasi = pd.read_csv("../dataset/hasil_normalisasi.csv")
hasil_normalisasi
hasil_normalisasi.drop(['Unnamed: 0'], axis=1, inplace=True)
hasil_normalisasi.columns = ['text']
hasil_normalisasi = hasil_normalisasi.iloc[1:5000]
hasil_normalisasi
factory = StemmerFactory()
stemmer = factory.create_stemmer()
stemming_proses = []
for stem_ind in hasil_normalisasi['text']:
    stemming_proses.append(stemmer.stem(stem_ind))
stemming_proses[30]
factory = StopWordRemoverFactory()
stopword = factory.create_stop_word_remover()
stopword_proses = []
for kalimat in stemming_proses:
    stopword_proses.append(stopword.remove(kalimat))
stopword_proses
tokenize_proses = []
for kalimat in stopword_proses:
    tokenize_proses.append(nltk.tokenize.word_tokenize(kalimat))
tokenize_proses
gabung_tokenize = []
for kata in tokenize_proses:
    gabung_tokenize.append(' '.join(kata))
gabung_tokenize
dataset_bersih = pd.DataFrame(gabung_tokenize, columns=['text'])
dataset_bersih
dataset_bersih = dataset_bersih[dataset_bersih.text != '']
dataset_bersih
dataset_bersih.sort_values(by='text', ascending=True)
dataset_bersih.drop_duplicates(subset ="text",
                               keep = False)
dataset_bersih.to_csv("../dataset/dataset_bersih.csv", index=None)
```

Gambar 5.4 Source Code Preprocessing data Python

Baris *script* ke-1 dan 2 adalah membaca *dataset* dan menampilkan data. Baris *script* ke-3 adalah intruksi untuk menghapus kolom yang tidak diperlukan. Baris *script* ke-4 adalah intruksi untuk mengganti nama kolom menjadi text. Baris *script* ke-5 dan 6 adalah intruksi untuk mengganti isi dari variabel hasil normalisasi menjadi nilai dari range 1 – 5000 dan menampilkan data. Baris *script* ke-6 dan 9 adalah intruksi untuk menampung library dan membuat list kosong untuk menampung hasil stemming. Baris *script* ke-10 dan 12 adalah intruksi untuk melakukan proses stemming dan menampilkan hasilnya. Baris *script* ke-13 dan 18 adalah intruksi untuk melakukan proses stopword dan menampilkan hasilnya. Baris *script* ke-19 dan 22 adalah intruksi untuk melakukan proses tokenize dan menampilkan hasilnya. Baris *script* ke-23 dan 26 adalah intruksi untuk melakukan proses penggabungan tokenize. Baris *script* ke-27 dan 28 adalah intruksi untuk mengubah data proses

sebelumnya menjadi dataframe dan menampilkannya. Baris *script* ke-29 dan 30 adalah intruksi untuk melakukan menghapus data kosong dan menampilkan hasilnya. Baris *script* ke-31 dan 32 adalah intruksi untuk melakukan pengurutan dan penghapusan duplikat data. Baris *script* ke-29 adalah intruksi untuk menyimpan hasil proses pembersihan data.

5.1.3 Proses Labelisasi Data

Klasifikasi merupakan teknik *machine learning* yang termasuk ke dalam pembelajaran terawasi (*supervised learning*), sehingga dibutuhkan pelabelan kelassentimen terhadap data ulasan. Pada penelitian ini, pelabelan klasifikasi yang dilakukan pada analisis sentimen dibagi menjadi dua kelas sentimen yakni sentimenpositif dan sentimen negatif. Berdasarkan jumlah kata yang terkandung dalam kalimat yang diberikan pengguna dapat diketahui gambaran umum dari penilaian pengguna terhadap aplikasi MyTelkomsel. Maka dari itu, ulasan dengan jumlah kata negatif lebih banyak dari kata positif dapat dilabelkan ke dalam kelas negatif, sedangkan ulasan dengan jumlah kata positif lebih banyak dari kata negatif dapat dilabelkan ke dalam positif. Gambar 5.5 berikut merupakan *source code* dari proses labelisasi data ulasan aplikasi MyTelkosemsel.

Commented [AFWMS26]: Ini bener nmr acuannya?

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
data_siap = pd.read_csv('../dataset/dataset_bersih.csv')
data_siap
def hitung_skor_sentiment(kalimat, positif, negatif):
    list_kata = kalimat.split()
    cocok_positif = 0
    cocok_negatif = 0
    for kata in list_kata:
        if any(item.lower() == kata.lower() for item in positif):
            cocok_positif += 1
    for kata in list_kata:
        if any(item.lower() == kata.lower() for item in negatif):
            cocok_negatif += 1
    score = cocok_positif - cocok_negatif
    return score
with open('../dictonary/list-of-positive-words.txt') as f:
    lines_pos = f.readlines()
list_positif = [x.replace('\n', '') for x in lines_pos]

with open('../dictonary/list-of-negative-words.txt') as f:
    lines_neg = f.readlines()
list_negatif = [x.replace('\n', '') for x in lines_neg]
scores = []
for kalimat in data_siap['text']:
    scores.append(hitung_skor_sentiment(kalimat, list_positif, list_negatif))
scores
data = {
    'text' : data_siap['text'],
    'score' : scores
}
df = pd.DataFrame(data)
df
label = []
for score in df['score']:
    if score < 0 :
        label.append('negatif')
    else:
        label.append('positif')
data = {
    'text' : data_siap['text'],
    'score' : scores,
    'label' : label
}
df = pd.DataFrame(data)
df

 sns.countplot(x='label', data=df)
 plt.title('Tabel Distribusi Kolom Sentiment', fontsize=16)
 plt.ylabel('Persentase', fontsize=16)
 plt.xlabel('Sentiment', fontsize=16)
 plt.xticks(rotation='vertical')
 dataset_positif = df[df['label'] == 'positif']
 dataset_negatif = df[df['label'] == 'negatif']
 dataset_positif.to_csv("../dataset/dataset_positif.csv", index=None)
 dataset_negatif.to_csv("../dataset/dataset_negatif.csv", index=None)
 df.to_csv("../dataset/dataset_jadi.csv", index=None)

```

Gambar 5.5 Source code untuk melebelkan data

Baris *script* ke-1 sampai 3 adalah instruksi untuk memanggil *library* atau *package* yang dibutuhkan dalam membangun sistem meliputi Pandas, matplotlib dan seaborn. Baris *script* ke-4 dan 5 adalah membaca *dataset* dan menampilkan data. Baris *script* ke-6 sampai 17 adalah intruksi untuk membuat fungsi penghitungan score. Baris *script* ke-18 sampai 24. adalah intruksi untuk membuka file list kata negatif dan positif dan menampungnya. Baris *script* ke-25 sampai 28 adalah intruksi untuk menghitung score semua data dan menampungnya. Baris *script* ke-29 sampai 34 adalah intruksi untuk membuat list menjadi dataframe utuh dan menampilkan hasilnya. Baris *script* ke-35 sampai 47 adalah intruksi untuk membuat melebelkan data berdasarkan score yang dimiliki dan

menampilkannya. Baris *script* ke-49 sampai 53 adalah intruksi untuk membuat plot grafik perbandingan label positif dan negatif. Baris *script* ke-54 sampai 58 adalah intruksi untuk menyimpan hasil setiap proses kedalam file csv.

5.1.4 Proses Klasifikasi Sentimen

Proses klasifikasi dilakukan dengan menggunakan data latih dan data uji dalam proses pembuatan *machine learning*. Model *machine learning* dibuat dengan melakukan pelatihan pada data latih yang selanjutnya dapat diujikan menggunakan data uji. Pengujian model dilakukan untuk mengetahui tingkat keakurasian model dan sejauh mana model dapat melakukan klasifikasi dengan benar. Dalam membangun sistem dengan *machine learning*, terdapat beberapa metode algoritmayang dapat digunakan. Metode algoritma yang digunakan dalam penelitian ini adalah *Random Forest*. Adapun *source code* dari proses klasifikasi sentimendata ulasan aplikasi MyTelkomsel terdapat pada Gambar 5.6 berikut :

Commented [AFWMS27]: ??

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.feature_extraction.text import TfidfVectorizer
from imblearn.over_sampling import SMOTE
from collections import Counter
from sklearn.model_selection import train_test_split
import time
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score, recall_score, precision_score, f1_score, roc_auc_score
df = pd.read_csv('../dataset/dataset_jadi.csv')
df
x = df['text']
y = df['label']
le = LabelEncoder()
le.fit(["positif", "negatif"])
y = le.transform(df['label'].values)
tfidfconverter = TfidfVectorizer(max_features=2000, min_df=5, max_df=0.7, ngram_range=(1,3))
X1 = tfidfconverter.fit_transform(x).toarray()
oversample = SMOTE(k_neighbors=5)
X_smote, Y_smote = oversample.fit_resample(X1, y)
counter = Counter(Y_smote)
print(counter)
data = {'label' : Y_smote}
y_plot = pd.DataFrame(data)
plt.figure(figsize=(12,5))
sns.countplot(x='label', data=y_plot)
plt.title('Tabel Distribusi Kolom Sentiment', fontsize=16)
plt.ylabel('Persentase', fontsize=16)
plt.xlabel('Sentiment', fontsize=16)
plt.xticks(rotation='vertical')
X_train, X_test, y_train, y_test = train_test_split(X_smote, Y_smote, test_size=0.2, random_state=0)
text_classifier_en = RandomForestClassifier(n_estimators=100, random_state=0)
t0_en = time.time()
text_classifier_en.fit(X_train, y_train)
t1_en = time.time()
predictions_en = text_classifier_en.predict(X_test)
t2_en = time.time()
time_linear_train_en = t1_en-t0_en
time_linear_predict_en = t2_en-t1_en
print("EN Training time: %fs; Prediction time: %fs" % (time_linear_train_en, time_linear_predict_en))
print("Random Forest")
print('Accuracy  = ', round(accuracy_score(y_test, predictions_en)*100,2),'%')
print('Recall    = ', round(recall_score(y_test, predictions_en)*100,2),'%')
print('Precision = ', round(precision_score(y_test, predictions_en)*100,2),'%')
print('F1-Score  = ', round(f1_score(y_test, predictions_en)*100,2),'%')
print('ROC AUC   = ', roc_auc_score(y_test, predictions_en))
```

```
print("")
review = "jelek banget sih ini, apaan ini aplikasi, tidak berguna sama sekali"
review_vector = tfidfconverter.transform([review]).toarray() # vectorizing
pred_text = text_classifier_en.predict(review_vector)
pred_text = le.inverse_transform(pred_text)
print(pred_text)
```

Gambar 5.6 *Source code* proses klasifikasi sentimen

Commented [AFWMS28]: Untuk gambar yg terpisah halaman juga ini jd Gambar Lanjutan
Source code sebelumnya diberi ketrangan gambar dengan nmr gambar yg sama

Baris *script* ke-1 sampai 11 adalah instruksi untuk memanggil *library* atau *package* yang dibutuhkan dalam membangun sistem meliputi Pandas, matplotlib dan seaborn, lebel encoderm train test split, counter, dan sklearn. Baris *script* ke-12 sampai18 adalah membaca *dataset* dan menampilkan data serta mengubah label menjadi bentuk angka. Baris *script* ke-19 sampai24 adalah intruksi untuk mengatasi permasalah oversampling. Baris *script* ke-28 sampai 32 adalah intruksi untuk membuat grafik perbandingan hasil SMOTE. Baris *script* ke-33 sampai 41 adalah intruksi untuk membuat model random forest dan mengklasifikasikan datanya. Baris *script* ke-42 sampai 48 adalah intruksi untuk menampilkan hasil akurasi dll dari model random forest. Baris *script* ke-50 sampai 54 adalah intruksi untuk mencoba model dengan data baru.

5.1.5 Proses Asosiasi kata

Asosiasi teks diperoleh dengan menggunakan pendekatan nilai korelasi terhadap masing-masing kata terhadap kemungkinan suatu kata dibahas bersamaan dengan kata lainnya. Berdasarkan kata-kata yang paling banyak muncul, dapat diperoleh asosiasi antarkata pada masing-masing kelas sentimen secara bersamaan guna memperoleh informasi. Adapun *source code* dari proses asosiasi teks untuk data ulasan positif aplikasi MyTelkomsel terdapat pada Gambar 5.7 berikut:

```
library(SnowballC)
library(RColorBrewer)
library(wordcloud)
library(tm)
dataset_visual <- read.csv("../dataset/dataset_positif.csv", header=TRUE)
corpus_visual <- Corpus(VectorSource(dataset_visual$text))
corpus_visual2 <- tm_map(corpus_visual, removeWords, c("kasih","mohon","ayan","terima","kali","tolong",
                                                       "selesai","sekal","pakai","terimakasih","bintang",
                                                       "coba","moga","kalo"))
corpus_visual3 <- tm_map(corpus_visual2, gsub, pattern="langgan",replacement="langganan")
corpus_visual3 <- tm_map(corpus_visual2, gsub, pattern="layan",replacement="layanan")
corpus_visual3 <- tm_map(corpus_visual2, gsub, pattern="rusaksudah",replacement="sudah")
corpus_visual3 <- tm_map(corpus_visual2, gsub, pattern="rusakmohon",replacement="mohon")
corpus_visual3 <- tm_map(corpus_visual2, gsub, pattern="nya",replacement="terbaik")
tdm <- TermDocumentMatrix(corpus_visual3)
matrix <- as.matrix(tdm)
vector <- sort(rowSums(matrix), decreasing=TRUE)
data_frame <- data.frame(word = names(vector), freq=vector)
head(data_frame, 50)
set.seed(1234)
wordcloud(words = data_frame$word, freq = data_frame$freq, min.freq = 1,
          max.words=50, random.order=FALSE, rot.per=0.35,
          colors=brewer.pal(8, "Set2"))
findFreqTerms(tdm, lowfreq = 10)
myvector <- as.list(findAssocs(tdm, terms=c("aplikasi","transaksi","pelayanan",
                                            "update","harga","fitur"),
                    corlimit = c(0.15,0.15,0.15,0.15,0.15,0.15)))
myvector
kk-barplot(data_frame[1:15,]$freq, las = 2, names.arg = data_frame[1:15,]$word,
           cex.axis=1.2,cex.names=1.2,
           main ="Most frequent words",
           ylab = "Word frequencies",col = terrain.colors(4))
termFrequency <- rowSums(as.matrix(tdm))
termFrequency <- subset(termFrequency, termFrequency>=115)
text(k, sort(termFrequency, decreasing = T)-2,
     labels=sort(termFrequency, decreasing = T),pch=6, cex=1)
```

Gambar 5.7 *Source code* proses asosiasi teks

Baris *script* ke-1 dan 4 adalah instruksi untuk memanggil *library* atau *package* yang dibutuhkan dalam membangun sistem meliputi RcolorBrewer dan wordcloud. Baris *script* ke-5 adalah membaca *dataset* hasil proses labelisasi data. Baris *script* ke-6 adalah mengkonversi *dataset* menjadi *corpus*. Baris *script* ke-7 sampai 9 adalah instruksi untuk melakukan proses penghapusan beberapa kata (*noise*) yang tak diperlukan. Baris *script* ke-10 sampai 14 adalah instruksi untuk melakukan pergantian bentuk kata dari yang awalnya *slang word* menjadi ternormalisasi. Baris *script* ke-15 adalah untuk mengkonversi *corpus* menjadi *Term Document Matrix* yang berisikumpulan kata pada sentimen positif. Baris *script* ke-16 dan 17 adalah untuk mengkonversi *Term Document Matrix* menjadi bentuk matriks dan memvektorisasi matriks tersebut. Baris *script* ke-18 dan 19 adalah instruksi untuk mengkonversi vektor menjadi *dataframe* dan menampilkan hasilnya. Baris *script* ke-20 adalah mendefinisikan angka random agar hasil dari proses distribusi persebaran data tetap sama meskipun dilakukan berulang-ulang. Baris

script ke-21 sampai 23 adalah intruksi untuk membuat visualisasi data berupa *wordcloud* yang berisi kumpulan kata positif dengan frekuensi kemunculan kata tertinggi. Baris *script* ke-24 adalah instruksi untuk menampilkan kata-kata dalam dokumen matriks yang memiliki frekuensi kemunculan tinggi, yakni dengan frekuensi minimal 10 kali kemunculan. Baris *script* ke-25 sampai 28 adalah instruksi untuk mencari korelasi antarkata (yang telah didefinisikan) dengan nilai korelasi minimal 0,15 dan kemudian menampilkan hasilnya. Baris *script* ke-29 sampai 31 adalah untuk mencari korelasi pada satu kata (yang telah didefinisikan) dan menampilkan hasilnya. Baris *script* ke-32 sampai 36 adalah untuk memvisualisasikan data dalam bentuk *barplot* untuk kata-kata positif dengan frekuensi tertinggi.

5.2 Hasil

5.2.1 Hasil Proses Web Scraping

Pada penelitian ini, dilakukan pengambilan data berupa ulasan pengguna aplikasi MyTelkomsel pada situs Google Play Store. Pengambilan data dilakukan dengan teknik *web scraping* menggunakan *library* dalam bahasa pemrograman Python. Sebenarnya terdapat beragam *library* yang dapat digunakan untuk mengambil data pada situs *online*. Namun penulis menggunakan *library* Google Play Scraper karena dapat digunakan untuk mengekstraksi data pada situs *online* secara gratis. Adapun pada penelitian ini, digunakan *library* Google Play Scraper versi 1.0.0. Data yang diekstraksi kemudian dapat diekspor menjadi *spreadsheet* Microsoft Excel (CSV) dan diunduh ke dalam komputer lokal. Contoh hasil proses*web scraping* menggunakan *library* Google Play Scraper seperti pada Gambar 5.8.

	content
0	Keren dan sangat mudah liat pulsa tau kuota
1	Mantap
2	BismiLLah Baik ...
3	Makasih Telkomsel, hari ini gw dapet promo kuota 27 GB. Cuma tuker poin sama pulsa 10 dapet kuota gede banget. Mantep cuy.
4	Mantap

Gambar 5.8 Hasil Proses Web Scrapping

5.2.2 Hasil Proses Data Preprocessing

Pada tahap *data preprocessing*, akan dilakukan pembersihan data menggunakan metode *text mining*. Beberapa tahap yang akan dilakukan di antaranya meliputi *case folding*, *cleaning*, *spelling normalization*, *stemming*, *tokenizing*, dan *filtering* yang akan dijelaskan sebagai berikut.

a) Case Folding

Case folding adalah proses penyeragaman bentuk huruf di mana dalam proses ini hanya menerima huruf latin antara "a" sampai "z". Karakter lain selain huruf dianggap sebagai *delimiter* sehingga karakter tersebut akan dihapus dari *dataset*. Kemudian penyeragaman dilakukan dengan mengubah isi dokumen menjadi huruf kecil secara keseluruhan (dari "a" sampai dengan "z"). Hal ini bertujuan agar kata yang ditulis dengan huruf awal kapital dan huruf nonkapital tidak terdeteksi memiliki arti yang berbeda. Implementasi *case folding* dapat dilihat salah satu contoh data ulasan aplikasi MyTelkomsel pada Tabel 5.1 berikut.

Tabel 5.1 Hasil proses *case folding*

Data *input*	Data *output*
Tak kasih bintang 5,tp untuk saat ini Telkomsel kok kayak gini, kok sering gangguan terus,gmn ini untuk kelanjutannya sebagai jaringan no 1.....?	tak kasih bintang 5,tp untuk saat ini telkomsel kok kayak gini, kok sering gangguan terus,gmn ini untuk kelanjutannya sebagai jaringan no 1.....?

b) Cleaning

Pada tahap ini, dilakukan pembersihan ulasan dari simbol ataupun karakter yang tidak diperlukan seperti angka, singkatan, kata yang tidak diperlukan,tanda baca, dan lain sebagainya. Tabel 5.2 sampai 5.5 berikut merupakan salah satu contoh ulasan hasil dari proses *cleaning*.

- Menghapus karakter non-ASCII

Tabel 5.2 Hasil Proses Cleaning pertama

Data *input*	Data *output*
Kenapa pas beli kuota ga bisa **ðÿ™\u008f**	Kenapa pas beli kuota ga bisa

- Menghapus angka

Tabel 5.3 Hasil Proses Cleaning

Data *input*	Data *output*
tak kasih bintang 5,tp untuk saat ini telkomsel kok kayak gini, kok sering gangguan terus,gmn ini untuk kelanjutannya sebagai jaringan no 1.....?	tak kasih bintang ,tp untuk saat ini telkomsel kok kayak gini, kok sering gangguan terus,gmn ini untuk kelanjutannya sebagai jaringan no .....?

- Menghapus tanda baca

Tabel 5.4 Hasil proses *cleaning* ketiga

Data *input*	Data *output*

tak kasih bintang ,tp untuk saat ini telkomsel kok kayak gini, kok sering gangguan terus, gmn ini untuk kelanjutannya sebagai jaringan no .....?	tak kasih bintang tp untuk saat ini telkomsel kok kayak gini kok sering gangguan terus gmn ini untuk kelanjutannya sebagai jaringan no

- Menghapus spasi yang berlebihan

Tabel 5.5 Hasil proses *cleaning* keempat

Data *input*	Data *output*
gak bisa masuk terus katanya nomor ponsel sudah terdaftar gunakan nomor ponsel lain kan aneh.	gak bisa masuk terus katanya nomor ponsel sudah terdaftar gunakan nomor ponsel lain kan aneh.

c) *Spelling Normalization*

Pada bagian *spelling normalization* berguna untuk melakukan perbaikan kata-kata yang disingkat maupun salah ejaan dengan bentuk tertentu yang memiliki maksud yang sama. Sebagai contoh pada kata "tidak", memiliki banyak bentuk penulisan yaitu "tak", "enggak", "tdak", "tdk", dan sebagainya. Dalam penelitian ini, proses *spelling normalization* dilakukan dengan bantuan kamus *slang word* yang telah dikurasi oleh Nikmatun Aliyah Salsabila dan timnya. Adapun salah satu contoh perbaikan kata gaulatau *slang word* pada ulasan seperti terlihat pada Tabel 5.6 berikut.

Tabel 5.6 Hasil proses *spelling normalization*

Data *input*	Data *output*
tak kasih bintang tp untuk saat ini telkomsel kok kayak gini kok sering gangguan terus gmn ini untuk kelanjutannya sebagai jaringan no	tak kasih bintang tapi untuk saat ini telkomsel kok kayak gini kok sering gangguan terus gimana ini untuk kelanjutannya sebagai jaringan nomor

d) *Stemming*

Stemming merupakan tahap yang dilakukan untuk mengubah kata berimbuhan pada ulasan menjadi kata dasar. Pada tahap ini, dilakukan penghapusan imbuhan pada kata hasil dari proses *spelling normalization*, baik yang mengandung imbuhan induktif maupun deduktif. *Stemming* dapatdilakukan pada bahasa pemrograman R menggunakan *package* bernama katadasaR. Adapun Tabel 5.7 berikut merupakan salah satu contoh ulasan dari hasil proses

stemming.

Tabel 5.7 Hasil proses *stemming*

Data *input*	Data *output*
tak kasih bintang tapi untuk saat ini telkomsel kok kayak gini kok sering gangguan terus gimana ini untuk kelanjutannya sebagai jaringan nomor	tak kasih bintang tapi untuk saat ini telkomsel kok kayak gini kok sering ganggu terus bagaimana ini untuk lanjut sebagai jaringan nomor

e) Tokenizing

Tokenizing adalah proses memisahkan kata pada sebuah kalimat menjadi kata-kata yang independen. *Tokenizing* dilakukan untuk mendapatkan *token* atau potongan kata yang akan menjadi entitas yang memiliki nilai dalam penyusunan matriks pada proses selanjutnya. *Tokenizing* dapat memudahkan proses perhitungan keberadaan kata tersebut dalam kalimat ataupun untuk menghitung frekuensi kemunculan kata tersebut dalam *corpus*. Contoh proses *tokenizing* ditunjukkan pada Gambar 5.9

Gambar 5.9 Hasil proses *tokenizing*

tak kasih bintang tapi untuk saat ini telkomsel kok kayak gini kok sering ganggu terus bagaimana ini untuk lanjut sebagai jaringan nomor

tak	kasih	bintang	tapi	untuk
saat	ini	telkomsel	kok	kayak
gini	kok	sering	ganggu	terus
bagaimana	ini	untuk	lanjut	sebagai
jaringan	nomor			

f) Filtering

Tahap *filtering* merupakan tahap dilakukannya pemilihan kata pada ulasan atau pengurangan dimensi kata di dalam corpus yang disebut *stopword*. Pada penelitian ini, daftar *stopword* yang digunakan adalah yang telah disusun oleh Fadilah Z. Tala yang berisi sebanyak 758 kata. *Stopword removal* merupakan tahap untuk menghilangkan kata-kata yang tidak berpengaruh atau tidak informatif namun seringkali muncul dalam ulasan. Kata-kata tersebut seperti kata penghubung, kata ganti orang, kata seruan, dan kata lainnya yang tidak begitu memiliki arti dalam penentuan kelas sentimen suatu ulasan. Kumpulan *stopword* akan disimpan di dalam *stoplist*dan contoh isi *stoplist* antara lain "aku", "kamu", "dan", "ke", "itu", "selain", "adalah", dan masih banyak lagi. Implementasi proses *filtering* dapat dilihat pada Tabel 5.8 berikut

Tabel 5.8 Hasil proses *filtering*

Data *input*	Data *output*
tak kasih bintang tapi untuk saat ini telkomsel kok kayak gini kok sering ganggu terus bagaimana ini untuk lanjut sebagai jaringan nomor	kasih bintang telkomsel kok kayak sering ganggu bagaimana lanjut sebagai jaringan nomor

5.2.3 Hasil Proses lebelisasi data

Pada umumnya, analisis sentimen digunakan untuk melakukan klasifikasi (pelabelan) dokumen teks ke dalam tiga kelas sentimen, yaitu sentimen positif, negatif dan netral. Cara menentukan kelas sentimen adalah dengan menghitung skor jumlah kata positif dikurangi skor jumlah kata negatif dalam setiap kalimat ulasan.Tapi pada penelitian ini akan digunakan dua pelabelan kelas sentimen, yaitu sentimen positif dan

sentimen negatif. Hal ini dilakukan karena kelas sentimen

Netral dianggap kurang memberikan manfaat bagi pihak PT PLN (Persero). Adapun perhitungan skor sentimen ulasan dapat dilakukan dengan persamaan (5.1) berikut.

$$skor = jumlah\ kata\ positif - jumlah\ kata\ negatif \quad (5.1)$$

Klasifikasi yang akan digunakan pada penelitian ini menggunakan data dengan sentimen positif dan negatif. Suatu ulasan diklasifikasikan sebagai sentimen positif bila mengandung pernyataan positif seperti pujian, ungkapan terima kasih, atau testimoni positif tentang aplikasi MyTelkomsel. Suatu ulasan diklasifikasikan sebagai sentimen negatif bila mengandung pernyataan-pernyataan negatif seperti ketidakpuasan, penghinaan, laporan kegagalan layanan, dan sebagainya. Simulasi perhitungan skor sentimen ulasan dapat dilihat pada Tabel 5.4 berikut

Commented [AFWMS29]: ??

Tabel 5.4 Simulasi perhitungan skor sentimen

Ulasan	Jumlah Kata Positif	Jumlah Kata Negatif	Skor Sentimen	Label Kelas
aplikasi **eror** menu **langgan** pilih titik koordinat lokasi **muncul**tampil pilih provinsi jawa timur **muncul** **tolong** aplikasi **mudah**	4	2	2	Positif
proses bayar bayar aplikasi **kecewa** aplikasi **batal** rubah **batal** aplikasi icon **batal** sekali **kesan** kayak **paksa** **coba**	2	5	−3	Negatif

Kalimat yang memiliki skor >= 0 akan diklasifikasikan ke dalam kelas positif, sedangkan kalimat yang memiliki skor < 0 diklasifikasikan ke dalam kelas negatif. Adapun hasil proses labelisasi kelas sentimen diperoleh jumlah data sepertipada Tabel 5.5 berikut.

Tabel 5.5 Distribusi kelas sentimen hasil lebelisasi data

Kelas Sentimen	Jumlah Data	Persentase Data
Positif	4596	95%
Negatif	404	15%
Total	5000	100%

Berdasarkan Tabel 5.5 diketahui bahwa *dataset* hasil proses labelissimengalami ketidakseimbangan kelas (*imbalanced class*). Hal ini perlu untuk dihindari karena dapat berpengaruh terhadap performa atau akurasi proses klasifikasi data. Untuk itu, pada penelitian ini digunakan teknik *oversampling* yangmana akan menyeimbangkan *dataset* dengan meningkatkan ukuran sampel pada kelas minoritas. Adapun teknik *oversampling* yang diterapkan dalam sistem dilakukan dengan bantuan *library* SMOTE (*Synthetic Minority Over-sampling Technique*) versi0.0-3. Setelah dilakukan proses *oversampling*, diperoleh hasil distribusi kelas sentimen seperti pada Tabel 5.6 berikut.

Tabel 5.6 Hasil Proses Oversampling denan SMOTE

Kelas Sentimen	Jumlah Data	Persentase Data
Positif	4417	50%
Negatif	4417	50%
Total	8834	100%

5.2.4 Hasil Proses klasifikasi sentimen

Data latih digunakan oleh algoritma *Random Forest* untuk membentuk sebuah model pengklasifikasi. Model ini merupakan representasi pengetahuan yang akan digunakan untuk prediksi kelas data baru yang belum pernah diketahui. Semakin besar data latih yang digunakan, maka akan semakin baik sistem dalam memahami pola data. Sedangkan data uji merupakan *dataset* yang digunakan untuk mengetahui

tingkat akurasi dari model yang dihasilkan. Data uji digunakan untuk mengukur sejauh mana algoritma *Random Forest* berhasil melakukan klasifikasi dengan benar. Data yang digunakan untuk data latihdan data uji adalah data yang telah memiliki label kelas. Adapun pada penelitian inidilakukan lima kali percobaan dengan perbandingan antara data latih dan data ujiyang berbeda, antara lain terlihat pada Tabel 5.7 sampai Tabel 5.11 berikut.

Commented [AFWMS30]: Nmr table acuan dicek lg

b) Perbandingan data latih 70% dan data uji 30%

Tabel 5.7 Data latih 70% dan data uji 30%

Kelas Sentimen	Jumlah Data	Data Latih	Data Uji
Positif	4417	3092	1325
Negatif	4417	3092	1325
Total	8834	6184	2650

c) Perbandingan data latih 75% dan data uji 25%

Tabel 5.8 Data latih 75% dan data uji 25%

Kelas Sentimen	Jumlah Data	Data Latih	Data Uji
Positif	4417	3313	1104
Negatif	4417	3313	1104
Total	8834	6626	2208

d) Perbandingan data latih 80% dan data uji 20%

Tabel 5.9 Data latih 80% dan data uji 20%

Kelas Sentimen	Jumlah Data	Data Latih	Data Uji
Positif	4417	3534	883
Negatif	4417	3534	883
Total	8834	7068	1766

e) Perbandingan data latih 85% dan data uji 15%

Tabel 5.10 Data latih 85% dan data uji 15%

Kelas Sentimen	Jumlah Data	Data Latih	Data Uji
Positif	4417	3754	663
Negatif	4417	3754	663
Total	8834	7408	1326

f) Perbandingan data latih 90% dan data uji 10%

Tabel 5.11 Data latih 90% dan data uji 10%

Kelas Sentimen	Jumlah Data	Data Latih	Data Uji
Positif	4417	3975	442
Negatif	4417	3975	442
Total	8834	7951	884

Simulasi penerapan algoritma *Random Forest* dijelaskan melalui uraian berikut ini. Peneliti ingin menganalisis sentimen terhadap ulasan-ulasan pada aplikasi My Telkomsel di Google Play Store. Dalam menganalisis sentimen, data ulasan yang telah dikumpulkan selanjutnya dikategorikan pada setiap kelas sentimen apakah kata tersebut bernilai positif atau negatif. Kemudian hasil labelisasi tersebut akan dibuat menjadi sebuah data latih. Adapun data latih tersebutdapat dilihat Tabel 5.12 di bawah ini.

Tabel 5.12 Data latih untuk simulasi

No.	Ulasan	Label Kelas
1.	**bantu mudah ganggu langgan murah**	Positif
2.	**ganggu ribet langgan ganggu mahal murah**	Negatif
3.	**murah bantu langgan mudah**	Positif
4.	**ribet mahal mudah ganggu eror**	Negatif

Dari data latih tersebut, kemudian dipilih kata-kata unik dari semua ulasan tersebut sehingga menjadi dokumen matriks pada Tabel 5.13

Tabel 5.13 Dokumen matriks untuk simulasi

	bantu	mudah	ganggu	langgan	ribet	mahal	murah	eror
Ulasan1	1	1	1	1			1	
Ulasan2			2	1	1	1	1	
Ulasan3	1	1		1			1	
Ulasan4		1	1		1	1		1

Commented [MR31]: perbaiki

Dari matriks tersebut kemudian dihitung dengan menggunakan persamaan (2.4) sehingga setiap kemunculan kata dalam ulasan diubah menjadi nilai TF-IDF.Berikut adalah contoh perhitungan TF-IDF untuk kata "bantu" pada ulasan 1 dengan jumlah ulasan = 2, TF("bantu") = 1, dan IDF("bantu) = 1.

TF-IDF(ulasan1, "bantu") = x log 3/ 1 = 0.47

Apabila proses perhitungan di atas dilakukan untuk semua dokumen dan semua kata, maka akan dihasilkan matriks seperti Tabel 5.14 berikut.

Tabel 5.14 Hasil perhitungan TF-IDF untuk dokumen matriks

Commented [MR32]: Tabel jangan lebih dari batas

	bantu	mudah	ganggu	langgan	ribet	mahal	murah	eror
Ulasan1	0,477	0,477	0,477	0,477			0,477	
Ulasan2			0,176	0,477	0,477	0,477	0,477	
Ulasan3	0,477	0,477		0,477			0,477	
Ulasan4		0,477	0,477		0,477	0,477		0,477

Kemudian hasil perhitungan TF-IDF ini akan diolah dengan menggunakan algoritma *Random Forest* sehingga menghasilkan model probabilitas. Misalkan ulasan 1 dan 3 merupakan ulasan dengan kelas positif sedangkan ulasan 2 merupakan ulasan dengan kelas negatif. Sehingga matriks TF-IDF tersebut akan berubah menjadi seperti pada Tabel 5.15.

Commented [AFWMS33]: ???

Tabel 5.15 Hasil perhitungan TF-IDF untuk dokumen matriks

	bantu	mudah	ganggu	langgan	ribet	mahal	murah	eror	**label**
U1	0,477	0,477	0,477	0,477			0,477		Pos
U2			0,176	0,477	0,477	0,477	0,477		Neg
U3	0,477	0,477		0,477			0,477		Pos
U4		0,477	0,477		0,477	0,477		0,477	Neg

Berdasarkan Tabel 5.15 diketahui kelas positif memiliki dua data ulasan dengan jumlah kata sebanyak 6 kata dari 8 kosakata, sedangkan kelas negatif memiliki dua data ulasan dengan jumlah kata sebanyak 6 kata dari 8 kosakata. Dari data tersebut, maka proses membangun probabilitasnya adalah sebagai berikut.

1. Kelas Positif

Tahap ini dilakukan dengan mengumpulkan seluruh kosakata yang munculpada semua dokumen, kemudian dihitung nilai probabilitas dari setiap kosakata. Berikut adalah contoh perhitungan nilai probabilitas kata "bantu" pada kelas positif.

Berdasarkan perhitungan tersebut, diperoleh nilai probabilitas kata "bantu" pada kelas positif adalah sebesar 0,214. Hasil perhitungan terhadap kata-kata lain pada kelas positif disajikan pada Tabel 5.16.

Tabel 5.16 Hasil perhitungan probabilitas kosakata kelas positif

Kosakata	**Nilai Probabilitas**
bantu	0,214
mudah	0,214
ganggu	0,142
langgan	0,214
ribet	0,071
mahal	0,071
murah	0,214
eror	0,071

2. Kelas Negatif

Dengan menggunakan teknik yang sama seperti pehitungan pada kelas positif di atas, maka hasil dari perhitungan probabilitas kata-kata kelas negatif disajikan dalam Tabel 5.17 berikut.

Tabel 5.117 Hasil perhitungan probabilitas kosakata kelas negatif

Commented [AFWMS34]: Pada halaman berikutny jg diberi keterangan table lanjutan dan nama kolom yang sama

Kosakata	Nilai Probabilitas
bantu	0,071
mudah	0,142
ganggu	0,214
langgan	0,142
ribet	0,214

Tabel 5.17 Hasil Perhitungan probabilitas kosakata kelas negatif lanjutan

mahal	0,214
murah	0,142
eror	0,142

Dengan demikian, data perhitungan di atas membentuk model probabilitas yang dihasilkan oleh algoritma *Random Forest* pada Tabel 5.18.

Tabel 5.18 Model probabilitas tiap kelas sentimen

Kosakata	Prob. Positif	Prob. Negatif	Klasifikasi
bantu	**0,214**	0,071	Positif
mudah	**0,214**	0,142	Positif
ganggu	0,142	**0,214**	Negatif
langgan	**0,214**	0,142	Positif
ribet	0,071	**0,214**	Negatif
mahal	0,071	**0,214**	Negatif
murah	**0,214**	0,142	Positif
eror	0,071	**0,142**	Negatif

Setelah model probabilitas yang ada pada Tabel 5.18 terbentuk,

kemudian model tersebut disimpan dalam *database* sistem pengklasifikasi. Setelah itu, modeltersebut akan diuji akurasinya dengan menggunakan data baru yang belum diketahui kelasnya. Sebagai contoh, akan menguji data ulasan baru sebagai data ujiyang isinya "eror ganggu bantu mahal" dan diperoleh hasil berupa prediksi kelas sentimen seperti terlihat pada Tabel 5.19 berikut.

Tabel 5.19 Prediksi sentimen pada data uji

Kelas	eror	ganggu	bantu	mahal	Probabilitas
Positif (P=0,5)	0,071	0,142	0,214	0,071	0,0000765
Negatif (P=0,5)	0,142	0,214	0,071	0,214	**0,0002308**

Pada Tabel 5.19 terdapat kolom kelas dengan masing-masing P = 0,5. Nilai P = 0,5 merupakan peluang masing-masing kelas sebelum dilakukan proses prediksi. Sementara itu, pada kolom nilai probabilitas merupakan nilai probabilitasulasan tersebut terhadap masing-masing kelas di mana nilai yang terbesar dari data tersebut ialah hasil prediksinya. Dengan memperhatikan Tabel 5.19, maka dapat disimpulkan bahwa ulasan "eror ganggu bantu mahal" merupakan ulasan yang termasuk ke dalam kelas negatif karena nilai probabilitas kelas negatif lebih tinggi dibandingkan dengan nilai probabilitas kelas positif.

Setelah dilakukan simulasi klasifikasi sentimen pada data latih dan data uji, selanjutnya dilakukan analisis terhadap hasil klasifikasi dari algoritma *random forest*. Analisis hasil klasifikasi *random forest* dilakukan dengan membuat *confusion matrix* agar dapat diketahui nilai akurasi, *recall*, dan presisi. *Confusion matrix* merupakan salah satu informasi penting yang digunakan pada *machine learning* untuk mengetahui performa sebuah sistem pengklasifikasi. Padapenelitian ini, *confusion matrix* dibuat dengan menggunakan aplikasi RStudio. Berdasarkan pembagian data latih dan data uji yang berbeda, didapatkan hasil *confusion matrix* dengan nilai akurasi, presisi, *recall*, dan *F1 score* yang berbeda pula yang dapat dilihat pada Tabel 5.20 berikut.

Tabel 5.20 Hasil Confusion matrix

Pembagian	Akurasi	Presisi	Recall	F1 Score
Data latih 70% dan data uji 30%	96,91%	96,58%	97,31%	96,94%
Data latih 75% dan data uji 25%	97,1%	96,95%	97,3%	91,12%
Data latih 80% dan data uji 20%	96,83%	96,75%	95,97%	96,86%
Data latih 85% dan data uji 15%	97,44%	96,9%	98,06%	97,48%
Data latih 90% dan data uji 10%	97,29%	96,4%	98,48%	97,43%

Commented [MR35]: Tambahkan rumus presisi dll

Berdasarkan hasil akurasi pada Tabel 5.20 diketahui bahwa persentase akurasi klasifikasi dengan algoritma *random forest* akan cenderung naik seiring dengan berkurangnya jumlah data yang digunakan dalam pengujian (data uji). Adapun hasil maksimal diperoleh pada perbandingan data latih 85% dan data uji 15%, dengan akurasi tertinggi yaitu 97,44% dan presisi lebih dari 96% untuk kelas positif maupun negatif. Adapun secara keseluruhan, nilai akurasi, presisi, *recall*, maupun *F1 score* mengalami peningkatan seiring berkurangnya jumlah data uji yang digunakan dalam sistem klasifikasi sentimen.

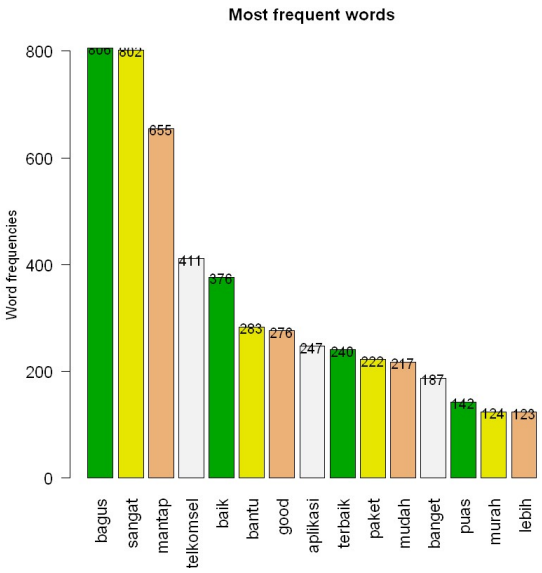
5.2.5 Hasil Proses asosiasi teks

Dalam penelitian ini, dilakukan visualisasi pada semua kelas data yakni sentimen negatif maupun positif. Hal ini bertujuan untuk mengekstrasi informasi secara keseluruhan tentang topik atau bahasan yang sering diulas para pengguna aplikasi MyTelkomsel. Selain itu, juga dilakukan pencarian korelasi antarkata pada masing-masing kelas data yang paling sering muncul secara bersamaan gunamemperoleh informasi penting dan berguna bagi pihak yang membutuhkan.

1. Kelas Positif

Data ulasan positif yang digunakan adalah data hasil proses labelisasi yangdilakukan menggunakan kamus *lexicon* berbahasa Indonesia. Ekstraksi informasi pada ulasan positif dilakukan secara berulang-ulang

hingga mendapatkan informasi tentang ulasan positif pengguna aplikasi MyTelkomsel yang paling sering dibicarakan. Ulasan positif tersebut di identifikasi berdasarkan frekuensi kata dalamulasan, Gambar 5.10 berikut adalah visualisasi hasil ekstraksi informasi yang didapatkan dari ulasan dengan klasifikasi ulasan positif.



Gambar 5.10 Kata yang paling banyak muncul pada kelas positif

Berdasarkan hasil klasifikasi ulasan positif, dari jumlah ulasan positif sebanyak 4711 ulasan, diperoleh beberapa kata yang paling banyak muncul diantaranya adalah kata "aplikasi" dengan frekuensi sebanyak 600 kali, "sangat" sebanyak 602 kali, "mantao" 655 kali, dan seterusnya. Kata-kata yang muncul sepertipada Gambar 5.10 merupakan kata yang memiliki sentimen positif dan merupakantopik pembicaraan yang paling banyak diulas oleh pengguna aplikasi MyTelkomsel. Kata-kata tersebut selanjutnya digunakan sebagai dasar untuk menemukan asosiasidengan kata lainnya, sehingga dapat diperoleh informasi yang lebih baik. Kumpulan kata-kata yang sering muncul tersebut juga dapat ditampilkan dalam bentuk *wordcloud* seperti terlihat pada Gambar 5.11.

Gambar 5.11 Wordcloud ulasan positif

Berdasarkan visualisasi *wordcloud* dapat dilihat dengan lebih jelas topik dan kata-kata positif yang sering digunakan pengguna aplikasi MyTelkomsel dalam memberikan ulasan. Semakin besar ukuran kata pada *wordcloud* menggambarkan semakin tinggi pula frekuensi kata tersebut, dan artinya semakin sering pengunjungmenggunakan kata tersebut sebagai topik pembicaraan atau penilaian positif dalam ulasan. Selanjutnya, dilakukan pencarian asosiasi antarkata yang sering muncul secara bersamaan dan diperoleh hasil pada Tabel 5.21 sebagai berikut.

Tabel 5.21 Asosiasi kata pada kelas sentimen positif

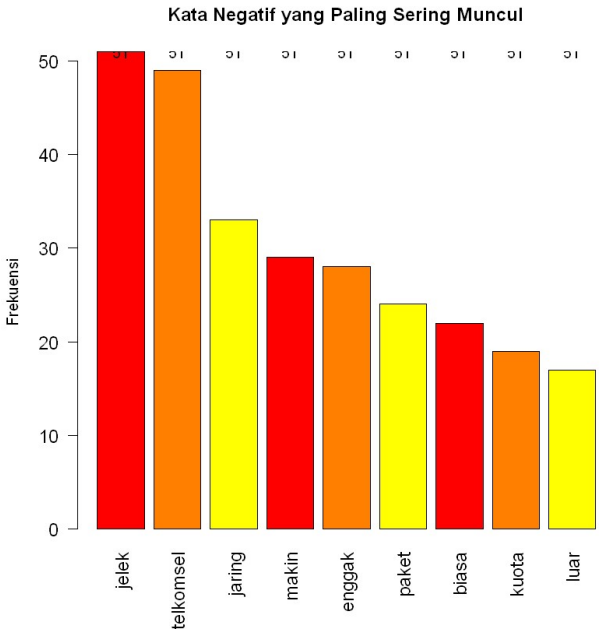
aplikasi		**transaksi**		**layanan**		**cepat**		**harga**	
bagus	0,18	manual	0,43	gunung	0,18	respon	0,27	percaya	0,58
mutasi	0,18	putar	0,43	posko	0,18	tanggap	0,19	tingkat	0,58
tolong	0,18	hasil	0,22	ajak	0,17	*rating*	0,17	stabil	0,36
upgrade	0,17	status	0,22	bicara	0,17	ramah	0,16	admin	0,29
rekening	0,17	transfer	0,21	ekstra	0,17	posisi	0,16	*mobile*	0,29
		beli	0,20	jasa	0,17				
		tombol	0,17						
		riwayat	0,17						
		tampil	0,17						

Berdasarkan Tabel 5.21, diperoleh beberapa asosiasi kata pada klasifikasi kelas positif. Proses ekstraksi informasi dengan asosiasi dilakukan secara berulang-ulang dengan cara menyaring kata-kata yang memiliki hubungan dengan kata lain dan didasarkan pada relevansi kata pada topik yang diulas. Jika dilihat asosiasi katayang berkaitan dengan kata "aplikasi", dapat diperoleh informasi tentang aplikasi yang bagus, adanya fitur mutasi, tolong tambahkan fitur baru, *upgrade* atau *update* aplikasi, dan fitur rekening. Kata-kata yang berasosiasi dengan kata "transaksi" juga memberikan informasi mengenai perlunya fitur transaksi manual, transaksi yang tidak memutar atau membingungkan, adanya hasil dan status setelah melakukan transaksi, adanya fitur transaksi melalui transfer sehingga memudahkan pembelian, tombol pada aplikasi, adanya fitur riwayat transaksi, serta tampilan atauUI (*User Interface*) aplikasi.

Kata-kata yang berasosiasi dengan kata "layanan" memberikan informasi bahwa layanan MyTelkomsel sudah beroperasi hingga di dataran tinggi atau pegunungan, adanya posko apabila pelanggan memerlukan bantuan, adanya ajakan dari para pengguna untuk memakai aplikasi MyTelkomsel, adanya layanan *customer service* agar pelanggan dapat membicarakan keluhan, adanya layanan ekstra, dan aplikasi MyTelkomsel sangat berjasa atau berkesan bagi pengguna. Kata-kata yang berasosiasi dengan kata "cepat" antara lain respon yang tanggap hingga pengguna memberikanrating yang baik, pelayanan yang ramah, dan adanya fitur *tracking* untuk mengetahui posisi pengguna yang mengalami gangguan pada layanan MyTelkomsel. Terakhir dapat dilihat bahwa kata "harga" memberikan informasi bahwa biaya layanan MyTelkomsel sangat terpercaya dan sesuai dengan tingkat kebutuhan pengguna, diharapkan juga agar harga selalu stabil, serta tidak adanya biaya admin pada aplikasi MyTelkomsel sangat membantu dan meringankan bagi para pelanggan.

2. Kelas Negatif

Ekstraksi informasi pada ulasan negatif dilakukan secara berulang-ulang hingga mendapatkan informasi tentang ulasan negatif pengguna aplikasi MyTelkomsel yang paling sering dibicirakan. Berdasarkan hasil labelisasi, ulasan negatifpengguna terhadap aplikasi sedikit lebih banyak bila dibandingkan dengan jumlah ulasan positif. Hal tersebut menunjukkan bahwa mayoritas pengguna apliksi MyTelkomsel menginginkan adanya peningkatan layanan ke arah yang lebih baik. Hasil ekstraksi informasi berupa ulasan negatif diidentifikasi berdasarkan frekuensi kata dalam ulasan. Selain itu, ekstraksi juga didasarkan pada relevansi kata dengan topikyang mengacu pada sentimen negatif. Gambar 5.12 berikut adalah visualisasi hasilekstraksi informasi yang didapatkan dari ulasan dengan klasifikasi ulasan negatif.



Gambar 5.12 Kata yang paling banyak muncul pada kelas negatif

Berdasarkan hasil klasifikasi ulasan negatif, diperoleh beberapa kata yang paling banyak muncul dengan topik dan dianggap relevan sebagai sentimen negatifdi antaranya adalah kata "jelek" dengan frekuensi sebanyak 50 kali, "telkomsel" sebanyak 48 kali, "jaring" sebanyak 37 kali, "enggak" sebanyak 35 kali, "paket"sebanyak 26 kali, dan seterusnya. Kata-kata yang muncul seperti pada Gambar 5.12 merupakan kata yang memiliki sentimen negatif berbahasa Indonesia dan merupakan topik pembicaraan yang paling banyak diulas. Kata-kata tersebut selanjutnya digunakan sebagai dasar untuk menemukan asosiasi dengan kata lainnya, sehingga dapat diperoleh informasi berupa sentimen negatif yang lebih akurat. Kumpulan kata-kata yang sering muncul tersebut dapat ditampilkan dalam bentuk *wordcloud* seperti terlihat pada Gambar 5.13.

Gambar 5.13 Wordcloud ulasan negatif

Visualisasi wordcloud pada Gambar 5.13 memberikan gambaran yang lebihjelas tentang topik dan kata-kata negatif yang sering digunakan pengunjung dalam memberikan ulasan. Beberapa topik yang sering dibahas pengunjung di antaranya adalah tentang listrik, padam, ganggu, rumah, dan sebagainya. Selanjutnya dilakukan pencarian asosiasi antar kata yang sering muncul secara bersamaan dan diperoleh hasil pada Tabel 5.22 sebagai berikut

Tabel 5.22 Asosiasi kata pada kelas sentimen negatif

mati		**jelek**		**jaringan**		**susah**		**error**	
lampu	0,53	lapor	0,35	konstruksi	0,60	jarak	0,31	*server*	0,38
siang	0,38	*force*	0,33	lokal	0,60	naik	0,31	cek	0,23
bayar	0,30	sering	0,33	pohon	0,60	nyala	0,31	*string*	0,23
hidup	0,30	*close*	0,28	rencana	0,60	saklar	0,31	*subtype*	0,23
hujan	0,30	mobile	0,26	roboh	0,60	setrika	0,31	sibuk	0,19
ampas	0,29	listrik	0,25	sengat	0,60				
bosan	0,29	maaf	0,20						

Tabel 5.22 menunjukkan asosiasi antarkata pada ulasan negatif, kata-kata tersebut merupakan topik yang paling sering dibicarakan pengguna dalam ulasannya. Berdasarkan tabel tersebut dapat diperoleh beberapa informasi berikut. Kata-kata yang berasosiasi dengan kata "mati" pada ulasan negatif memberikan informasi tentang lampu mati atau listrik padam, sering terjadi pemadaman pada waktu siang dan hujan, pelanggan merasa kecewa karena mereka sudah membayar,serta mereka juga meminta untuk segera hidup agar tidak ampas dan bosan.

Kata-kata yang berasosiasi dengan kata "jelek" pada ulasan negatif memberikan informasi tentang banyaknya laporan listrik yang terganggu, aplikasi MyTelkomsel yang sering *force close*, dan pihak MyTelkomsel hanya sering minta maaf saat terjadi permasalahan tersebut. Kata-kata yang berasosiasi dengan kata "padam" pada ulasan negatif memberikan informasi tentang pemadaman di wilayah kontruksi, pemadaman listrik lokal, listrik yang padam saat ada pohon roboh, rencana pemadaman listrik yang kurang tersosialisasikan, dan banyaknya masyarakat yang tersengat listrik karena kurangnya edukasi dan sosialisasi mengenai pengelolaan listrik yang bijak untuk rumah tangga.

Kata-kata yang berasosiasi dengan kata "susah" memberikan informasi tentang jarak rumah yang cukup jauh dengan teknisi MyTelkomsel sehingga pelanggan susahmemperoleh bantuan jika terjadi masalah, susahnya pelanggan untuk meminta naikdaya listrik, listrik yang padam susah (lama) untuk nyala kembali, susah dalam pemasangan saklar, dan susah untuk menyetrika karena listrik padam. Kata-kata yang berasosiasi dengan kata "*error*" memberikan informasi tentang keluhan pelanggan mengenai server aplikasi MyTelkomsel yang sering sibuk, banyak pelanggan yang meminta tim IT MyTelkomsel untuk segera cek *error* tersebut, dan masalah*string subtype error* yang terjadi pada kolom input aplikasi MyTelkomsel.

BAB VI
PENUTUP

6.1 Kesimpulan

Berdasarkan pengujian yang telah dilakukan pada analisis sentimen data ulasan pengguna aplikasi MyTelkomsel menggunakan *Random Forest*, maka simpulan yang didapatkan oleh penulis adalah sebagai berikut.

1) Algoritma *Random Forest* mampu dalam mengklasifikasikan data ulasan aplikasi MyTelkomsel menjadi kelas sentimen positif dan negatif dengan baik setelah dilakukan pembersihan, normalisasi, dan labelisasi.
2) Pengujian sistem dilakukan menggunakan lima jenis perbandingan data latih dan data uji. Dengan menggunakan perbandingan data latih dan data uji sebesar 85%:15%, diperoleh tingkat akurasi tertinggi yaitu 97,44%. Artinya dari 884 data ulasan yang diujikan, terdapat 861 ulasan yang benar diklasifikasikan oleh sistem.
3) Berdasarkan hasil asosiasi teks yang dilakukan, secara umum dapat diketahui bahwa pengguna aplikasi MyTelkomsel mayoritas membicarakan mengenai aplikasi, listrik, rumah, dan token karena selalu muncul baik pada kelas sentimen positif maupun negatif. Secara umum, metode asosiasi teks yang digunakan menunjukkan hasil ekstraksi informasi pada kelas positif di antaranya terkait aplikasi, transaksi, layanan, cepat, dan harga. Sedangkan pada kelas negatif yang sering dikeluhkan meliputi jelek, ganggu, padam, susah, dan *error*.

6.2 Saran

Adapun saran yang dapat diberikan oleh penulis setelah melakukan penelitian ini terkait perbaikan dan pengembangan untuk penelitian selanjutnya adalah sebagai berikut.

1) Bagi pihak PT Telkomsel (Persero), hasil ekstraksi informasi dari ulasan-ulasan yang telah diberikan oleh pengguna, khususnya ulasan yang berbentuk negatif dapat dijadikan bahan evaluasi dalam peningkatan kepuasan pengguna, memberikan pelayanan yang lebih baik untuk masa mendatang, serta dapat digunakan untuk pengembangan fitur dan pembaruan aplikasi pada versi selanjutnya.
2) Sistem labelisasi kelas sentimen yang digunakan dalam penelitian ini hanyaterbatas pada pendeteksian sentimen tiap kata menggunakan kamus *lexicon*,sehingga kata-kata dalam bentuk frasa dan klausa belum dapat teridentifikasi dengan baik. Untuk penelitian selanjutnya, sebaiknya dapat menggunakan teknik labelisasi data yang mampu mendeteksi sentimenhingga dalam bentuk frasa dan klausa.
3) Dalam penelitian ini, data ulasan yang digunakan masih dibatasi untuk ulasan yang berbahasa Indonesia saja, sehingga pada penelitian selanjutnya perlu dikembangkan menggunakan ulasan pada bahasa asing ataupun bahasa daerah.
4) Bagi peneliti selanjutnya, dapat menggunakan pendekatan atau algoritma *machine learning* yang lain sebagai perbandingan terhadap performa algoritma *Random Forest* dalam mengklasifikasikan sentimen ulasan pengguna aplikasi MyTelkomsel.

DAFTAR PUSTAKA

Abdullah, D. dan Fadlisyah, F. (2014), *Statistika Terapannya Pada Bidang Informatika*, Yogyakarta: Graha Ilmu.

Alita, D., dan Rahman, A. (2020), *Pendeteksian Sarkasme pada Proses Analisis Sentimen Menggunakan Random Forest Classifier* , Universitas Teknokrat Indonesia.

Breiman, L. (2021), *Random Forest,* University of California, Berkeley. Vol. 45, No. 5–32.

Baskoro, B., Susanto, I., Khomsah, S., (2021), *Analisis Sentimen Pelanggan Hotel di Purwokerto Menggunakan Metode Random Forest dan TF-IDF (Studi Kasus: Ulasan Pelanggan Pada Situs TRIPADVISOR)*, Institut Teknologi Telkom Purwokerto.

Bookhamer, P. dan Zhang, Z. J. (2016), *Knowledge Management In A Global Context: A Case Study*, Information Resources Management Journal, 29(1), 57-74.

Evita, F., Yulianti, Y., Rosyida, S., Gata, W., (2020), *Analisis Sentimen Terhadap Aplikasi Ruangguru Menggunakan Algoritma Naive Bayes, Random Forest Dan Support Vector Machine*, STMIK Nusa Mandiri.

Fanani, F. (2017), *Klasifikasi Review Software Pada Google Play Menggunakan Pendekatan Analisis Sentimen*, Skripsi, S.Kom., Universitas Gadjah Mada, Yogyakarta.

Feldman, R. dan Sanger, J. (2007), *The Text Mining Handbook: Advanced Approaches In Analyzing Unstructured Data*, New York: CambridgeUniversity Press.

Hairani. (2018), *Aplikasi Pemetaan Kualitas Pendidikan di Indonesia Menggunakan Metode K-Means*, J. MATRIK, vol. 17, no. 2, pp. 13–23.

Jihad, M., Adiwijaya, Astuti, W. (2021), *Analisis Sentimen Terhadap Ulasan Film Menggunakan Algoritma Random Forest*, Universitas Teknokrat Indonesia.

Josi, A. dan Abdillah, L.A. (2014), *Penerapan Teknik Web Scraping Pada Mesin Pencari Artikel Ilmiah*, Jurnal Sistem Informasi, 5(2), 259-264.

Krouska, A., Troussas, C. dan Virvou, M. (2016), *The Effect Of Preprocessing Techniques On Twitter Sentimen Analysis*, 7th International Conference on Information, Intelligence, Systems & Applications (IISA), 2016, 1–5.

Liu, B. (2010), *Sentiment Analysis And Subjectivity*, Handbook of Natural Language Processing, ed. 2, 627-666.

Liu, B. (2012), *Sentiment Analysis And Opinion Mining*, Synthesis Lectures on Human Language Technologies, 5(1), 1-167.

Manning, C.D., Raghavan, P. dan Schütze, H. (2008), *An Introduction to Information Retrieval*, New York: Cambridge University.

Marziah, K. (2020), *What is Google Play*, (https://www.lifewire.com/what-is- google-play-1616720), Diakses 24 April 2021.

Miner, G., Elder IV, J., Fast, A., Hill, T., Nisbet, R. dan Delen, D. (2012), *PracticalText Mining and Statistical Analysis For Non-Structured Text DataApplications*, Cambridge: Academic Press.

Pang, B. dan Lee, L. (2008), *Opinion Mining and Sentiment Analysis*, Foundationsand Trends in Information Retrieval, 2(1-2), 1-135.

Putranti, N.D. dan Winarko, E. (2014), *Analisis Sentimen Twitter Untuk Teks Berbahasa Indonesia Dengan Maximum Entropy dan Support Vector Machine*, IJCCS (Indonesian Journal of Computing and Cybernetics Systems),8(1), 91-100.

Siraj, F. dan Abdoulha, M.A. (2007), *Mining Enrolment Data Using Predictive and Descriptive Approaches*, Knowledge-Oriented Applications in Data Mining, 53–72.

Tan, Pang-Ning, Steinbach, M., Adeyeye Oshin, M., Kumar, V. dan Vipin (2006), *introduction To Data Mining*, New York: Pearson Addison Wesley.

Ulwan, M.N. (2016), P*attern Recognition Pada Unstructured Data Teks Menggunakan Support Vector Machine dan Association*, Skripsi, S.Kom., Program Studi Statistika Universitas Islam Indonesia.

Zafikri, A. (2008), *Implementasi Vector Space Model Metode Term Frequency Inverse Document Frequency (TF-IDF) Pada Sistem Temu Kembali Informasi*, Skripsi, S.Kom., Universitas Udayana.

Wandani, A., Fauziah, Andrianingsih. (2021), *Sentimen Analisis Pengguna Twitter pada Event Flash Sale Menggunakan Algoritma K-NN, Random Forest, dan Naive Bayes*, Universitas Nasional.