ANALISIS SENTIMEN BERBASIS ASPEK PADA ULASAN APLIKASI TANGERANG LIVE MENGGUNAKAN *LATENT DIRICHLET ALLOCATION* DAN *NAÏVE BAYES*

SKRIPSI

Diajukan Sebagai Salah Satu Syarat untuk Memperoleh Gelar Sarjana Strata Satu

Program Studi Sistem Informasi

Disusun Oleh:

M. RIZQI ARIEL GIFFARI
11170930000078

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH JAKARTA

2022 M / 1444 H

ANALISIS SENTIMEN BERBASIS ASPEK PADA ULASAN APLIKASI TANGERANG LIVE MENGGUNAKAN *LATENT DIRICHLET ALLOCATION* DAN *NAÏVE BAYES*

SKRIPSI

Diajukan Sebagai Salah Satu Syarat untuk Memperoleh Gelar Sarjana Strata Satu

Program Studi Sistem Informasi

Disusun Oleh:

M. RIZQI ARIEL GIFFARI
11170930000078

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH JAKARTA

2022 M / 1444 H

LEMBAR PERNYATAAN

DENGAN INI SAYA MENYATAKAN BAHWA SKRIPSI INI BENAR-BENAR HASIL KARYA SAYA SENDIRI DAN BELUM PERNAH DIAJUKAN SEBAGAI SKRIPSI ATAU KARYA ILMIAH PADA PERGURUAN TINGGI ATAU LEMBAGA MANAPUN.

Tangerang, 17 Agustus 2022

M. RIZQI ARIEL GIFFARI
NIM. 11170930000078

ABSTRAK

M. Rizqi Ariel Giffari – 11170930000078. Analisis Sentimen Berbasis Aspek pada Ulasan Aplikasi Tangerang Live Menggunakan *Latent Dirichlet Allocation* dan *Naïve Bayes*. Di bawah bimbingan **Dr. Qurrotul Aini, M.T. dan Ir. Eri Rustamaji, MBA.**

Tangerang LIVE merupakan aplikasi layanan publik terintegrasi milik Pemerintah Kota Tangerang untuk memudahkan masyarakat mendapatkan pelayanan pemerintahan. Walaupun aplikasi Tangerang LIVE mendapatkan *rating* yang cukup baik di Play Store, layanan yang diberikan masih belum maksimal, berdasarkan keluhan masyarakat yang dapat dilihat pada kolom ulasan aplikasi Tangerang LIVE. Penelitian Analisis Sentimen Berbasis Aspek pada Ulasan Aplikasi Tangerang Live Menggunakan *Latent Dirichlet Allocation* dan *Naïve Bayes* bertujuan untuk mengetahui aspek dan klasifikasi sentimen yang ada pada aplikasi Tangerang LIVE dengan menggunakan algoritma *Latent Dirichlet Allocation* dari *library* Gensim untuk pemodelan topik dan *Naïve Bayes* dari *library* Scikit Learn untuk klasifikasi sentimen, lalu hal apa saja yang menjadi kelebihan dan kekurangan dari tiap aspek yang ada pada aplikasi Tangerang LIVE dan nilai kinerja algoritma LDA dan *Naïve Bayes*. Yang nantinya hasil penelitian ini bisa menjadi bahan pertimbangan untuk Pemerintah Kota Tangerang meningkatkan layanan aplikasi Tangerang LIVE. *Dataset* yang digunakan adalah ulasan *user* aplikasi Tangerang LIVE di Play Store yang didapat dengan cara *scraping*. Penelitian ini menggunakan *Latent Dirichlet Allocation* untuk pemodelan topik dan *Naïve Bayes* untuk pengklasifikasian sentimen. Pemodelan topik menggunakan LDA menghasilkan 4 aspek, yaitu: *User Interface, User Experience, Functionality and Performance, Support and Updates*. Untuk klasifikasi sentimen menggunakan *Naïve Bayes* hasil akurasi yang didapat adalah 87,80% lalu dilakukan evaluasi kinerja algoritma menggunakan Kurva ROC menghasilkan nilai AUC 0,94.

Kata Kunci: Analisis Sentimen Berbasis Aspek, Tangerang LIVE, *Latent Dirichlet Allocation, Naïve Bayes, Lexicon Based*, Kuva ROC.

Bab 1-5 + xv Halaman + 89 Halaman + 31 Gambar + 21 Tabel + 73 Daftar Pustaka + Lampiran

Pustaka Acuan (73, 2003 – 2022)

KATA PENGANTAR

Assalamu'alaikum Warahmatullaahi Wabarakaatuh.

Puji dan syukur kehadiran Allah SWT yang telah memberikan rahmat serta hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi ini. *Shalawat* serta salam semoga selalu tercurah kepada junjungan kita Rasulullah Muhammad SAW sebagai *rahmatan lil 'alamin*, beserta para keluarga dan sahabat.

Penulis menyadari skripsi ini masih terdapat kekurangan. Namun demikian penulis berharap skripsi ini dapat memenuhi syarat dalam memperoleh gelar sarjana komputer Program Studi Sistem Informasi di Fakultas Sains dan Teknologi Universitas Islam Negeri Syarif Hidayatullah Jakarta. Skripsi yang berjudul "Analisis Sentimen Berbasis Aspek pada Ulasan Aplikasi Tangerang Live Menggunakan *Latent Dirichlet Allocation* dan *Naïve Bayes*", akhirnya dapat diselesaikan sesuai dengan harapan penulis.

Selama penyusunan skripsi ini tentunya penulis menghadapi kesulitan dan tantangan, namun berkat bantuan dari berbagai pihak yang telah membantu, memberikan dukungan dan bimbingan sehingga kesulitan dapat diatasi dan dapat menyelesaikan skripsi ini. Penulis ingin mengucapkan terima kasih kepada:

1. Bapak Ir. Nashrul Hakiem, M.T., Ph.D. selaku Dekan Fakultas Sains dan Teknologi.
2. Bapak A'ang Subiyakto, M. Kom., Ph. D. selaku Ketua Program Studi Sistem Informasi dan Bapak Nuryasin, M.Kom. selaku Sekretaris Program Studi Sistem Informasi Fakultas Sains dan Teknologi.

3. Ibu Dr. Qurrotul Aini, M.T. dan bapak Ir. Eri Rustamaji, MBA selaku dosen pembimbing yang telah memberikan waktu, tenaga, dan ilmunya selama saya bimbingan skripsi sehingga saya dapat menyelesaikan skripsi ini.
4. Dosen-dosen Program Studi Sistem Informasi yang telah memberikan ilmu selama saya berkuliah di UIN Jakarta
5. Orang tua penulis yang selalu memberikan doa, semangat, dan dukungan yang terus mengalir kepada penulis.
6. Ibu Fitsa Dwi Putri salah satu Kasi di Diskominfo Kota Tangerang.
7. Keluarga besar Program Studi Sistem Informasi UIN Jakarta, khususnya Angkatan 2017 dan seluruh teman-teman kelas C 2017 yang saling mendukung dan memberikan semangat
8. Serta semua pihak yang telah membantu hingga skripsi ini terselesaikan.

Terima kasih atas segala bantuan dari semua pihak, semoga Allah SWT membalas semua kebaikan yang telah diberikan oleh semua pihak dan semoga skripsi ini dapat bermanfaat bagi para pembaca dan penelitian selanjutnya.

Wassalamu'alaikum Warahmatullaahi Wabarakaatuh.

Tangerang, Agustus 2022

M. RIZQI ARIEL GIFFARI
NIM. 11170930000078

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Teknologi informasi yang kian berkembang memberikan perubahan, mulai dari aspek keuangan, kesehatan, pembelajaran juga pada aspek pemerintahan. Salah satu bentuk perkembangan teknologi pada pemerintahan ialah *smart city*. *Smart city* merupakan konsep yang menyusun suatu tatanan kota supaya mampu berperan dalam memudahkan masyarakat untuk mendapatkan informasi dengan cepat, tepat, dan *real time* (Hasibuan *et al.*, 2019). Guna mewujudkan Gerakan Menuju 100 *Smart City* yang berorientasi membimbing pemerintah kota/kabupaten dalam menyusun konsep *smart city* oleh pemerintah pusat, konsep *smart city* dirasa sebagai salah satu solusi untuk mengatasi masalah perkotaan seperti keamanan masyarakat, kemacetan, pembangunan infrastruktur, sampai memberikan pelayanan yang baik bagi masyarakat (Amelia, 2020).

Pemerintah Kota Tangerang memulai konsep *smart city* sejak tahun 2012 dengan dibuatnya aplikasi layanan publik. Pada 2016 Pemerintah Kota Tangerang membuat Tangerang Live Room yang berfungsi sebagai pusat kontrol dalam mengobservasi, memperoleh informasi serta mengevaluasi laporan seputar layanan masyarakat yang kini telah menjadi aplikasi layanan publik terintegrasi dengan nama Tangerang LIVE. Aplikasi yang diintegrasikan yaitu; Sobat Dukcapil, PPDB *Online* Kota Tangerang, Pangkas, Sigacor, Plesiran, dan lainnya. LIVE merupakan akronim dari *Liveable, Investable, Visitable,* dan *E-City* berbasis TIK yang membuat sistem pelayanan masyarakat menjadi lebih efektif dan efisien.

Aplikasi Tangerang LIVE dapat diunduh melalui *platform* pendistribusian aplikasi seperti Google Play Store. Google Play Store merupakan sebuah aplikasi untuk pendistribusian aplikasi pada android dimana *user* mendapatkan aplikasi langsung dari *developer*. Di Google Play Store terdapat penilaian yang bisa dilakukan oleh *user*, berupa bintang 1 sampai 5, juga dapat memberikan ulasan terhadap aplikasi tersebut pada kolom komentar. Aplikasi Tangerang LIVE pada Google Play Store telah diunduh lebih dari 500 ribu kali dengan *rating* bintang di atas 4 dan sebanyak 13 ribu lebih orang telah mengulas aplikasi ini. Pemerintah Kota Tangerang, pada saat ini menghimbau masyarakatnya menggunakan aplikasi Tangerang LIVE untuk mendapatkan pelayanan publik pemerintahan seperti, perizinan *online*, pencatatan sipil dan lainnya. Namun aplikasi Tangerang LIVE masih belum maksimal pelayanannya, dan masyarakat mengeluhkannya di kolom ulasan halaman Tangerang LIVE pada Play Store, hal ini membuat penilaian berupa bintang pada aplikasi belum cukup untuk menggambarkan kualitas dari aplikasi tersebut. Ulasan yang diberikan oleh *user* berbentuk kalimat lebih menggambarkan apa yang dirasakan *user* saat menggunakan aplikasi (Potharaju *et al.*, 2017). Pada Tabel 1.1 berikut merupakan contoh keluhan pada aplikasi Tangerang LIVE.

Tabel 1.1 Keluhan Pada Aplikasi Tangerang LIVE

Tanggal	***Rating***	***User***	**Ulasan**
23-07-2019	1	Indra L	aplikasi nya si bermanfaat. tapi server nya kurang kuat buat penguna yang terlalu banyak. mohon ditingkatkan lagi server nya supaya lebih cepat. kasian orang2 yang cari kerja lewat JOB FAIR. udah datang ke lokasi. udh pada bawa lamaran.

			tpi pendaftaran nya lewat online. pas mau daftar online lewat aplikasi ni TNG LIVE. sistem nya eror terus.. banyak yang pada pulang lagi. dan pada kecewa. mohon di tingkatkan lagi donk....!!!!!!!!!
27-11-2021	3	Jnw Imoo	Aplikasi ini inovatif, saya kasih rating 9/10, kalau pelayanan offline Kelurahan dan kecamatan saya kasih rating 2/10 karena pegawai tidak ada jiwa melayaninya sama sekali, tidak ramah, tidak membantu, tolong para Pekerja loket di Kecamatan dan Kelurahan diajarkan cara bekerja. Berbeda sekali dengan PTSP Jakarta, ramah, mengayomi, lembut, sopan. Mohon dipantau Pemkot Tangerang.
19-08-2021	3	Y0GI T0Y0G	Ini aplikasi susah di mengert , pas mau memperbarui kk yg rusak, kita harus ngisi surat pengantar RT , terus saya isi mau bertemu jam 09,00 di situ yg bikin bingung sudah ketemu RT nya terus di aplikasi GK ada jawaban
18-06-2021	5	Raska Ook	Sy agak bingung ama apk ini apa memang begini sy daptar bantuan umkm cuma di tolak. Tapi kok nggak bisa diupdate lagi y nggak bisa daftar lagi apa udah di black list
21-06-2020	2	Aria Zakaria	Daftar pembuatan akte udah hampir 3 bulan status masih "Pendaftaran" aja. Sebenernya ini ada yang ngerjain gak

			sih?? Kan tujuan dari aplikasi ini itu untuk mempermudah bukan membingungkan. Jadi gak jelas gini ya tangerang live. Masih lebih bagus versi sebelumnya akte anak saya langsung jadi 1 bulan...

Developer perlu mengetahui kelebihan dan kekurangan yang ada pada aplikasinya menurut *user*, agar dapat menjadi acuan untuk evaluasi guna meningkatkan kualitas aplikasinya. Salah satu cara untuk mengetahui hal tersebut dengan analisis sentimen berbasis aspek (Ailiyya, 2020). Penelitian sejenis pernah dilakukan oleh Ailiyya pada tahun 2020 melakukan analisis sentimen berbasis aspek pada ulasan aplikasi Tokopedia. Pada penelitian tersebut aspek yang digunakan yaitu: layanan, sistem, dan kemanfaatan serta menggunakan model SVM. Data yang digunakan adalah ulasan pengguna Tokopedia di Google Play Store dari bulan April sampai bulan Juli 2019 yang didapat dengan proses *scraping* serta menggunakan model SVM untuk klasifikasi sentimen dan aspek.

Penelitian sejenis lainnya dilakukan oleh Astuti (2020), ia melakukan analisis sentimen berbasis aspek menggunakan data ulasan pengguna aplikasi Tokopedia pada Google Play Store dari bulan Oktober 2018 sampai Mei 2019. Astuti melakukan proses *scraping* untuk mengambil data, menggunakan LDA (*Latent Dirichlet Allocation*) untuk penentuan jumlah aspek yang didapat dari *clustering topic* yaitu: aspek kemanfaatan, pelayanan, pengalaman belanja, dan tampilan. lalu menggunakan *Naive Bayes* untuk klasifikasi sentimen. menghasilkan nilai akurasi sebesar 92.5% serta nilai AUC (*Area Under Curve*) mendapat nilai yang baik di angka 0,95.

Penelitian yang juga menggunakan LDA dan *Naive Bayes* dilakukan oleh Putu & Amrullah (2021). Mereka melakukan analisis sentimen menggunakan data *tweet* pada Twitter dari tahun 2014 sampai 2019 yang diambil dengan *crawling.* Hasilnya menunjukkan bahwa algoritma *Naive Bayes* mampu mengklasifikasi sentimen dengan baik yang ditunjukkan melalui hasil pengukuran kinerja algoritma sebesar 92% pada nilai akurasi, 100% pada presisi, 83,84% pada *recall* dan 100% pada *specificity*. Pada pemodelan topik menggunakan LDA dihasilkan topik terbaik pada kelas positif berjumlah 8 topik yang nilai koherensinya 0,613, sedangkan untuk topik terbaik pada kelas negatif berjumlah 12 topik yang nilai koherensinya 0,528.

Naive Bayes merupakan model yang paling sederhana dan bisa bekerja dengan baik pada klasifikasi teks (Aggarwal, 2015). Pendapat lain menyebutkan *Naive Bayes* adalah metode klasifikasi teks yang memiliki kecepatan pemrosesan serta akurasi yang cukup tinggi jika digunakan pada kasus yang memiliki data banyak, besar, dan beragam (Fitriyyah *et al.*, 2019). Metode *Naive Bayes* merupakan algoritma yang sederhana serta bagus dalam pengklasifikasian teks (Wisnu *et al.*, 2020). Prinsip dasar dari LDA adalah setiap dokumen direpresentasikan sebagai campuran topik-topik yang tersembunyi serta belum diketahui dan setiap topiknya terdiri atas penyebaran banyak kata (Cendana *et al.*, 2019). LDA digunakan untuk menentukan beberapa topik yang muncul dari masing-masing opini pada setiap kelas (Putu *et al.*, 2021). Kelebihan metode LDA adalah dapat mengekstrak topik secara akurat pada kumpulan data yang cukup besar (Alamsyah *et al.*, 2018).

Berlandaskan pada uraian yang telah dipaparkan, penulis memilih metode LDA karena LDA dapat meringkas, mengklusterkan topik, mengidentifikasi jumlah topik yang optimal, dan memberi label topik pada *dataset.* untuk *Naïve Bayes* karena NB merupakan metode yang sederhana, namun memiliki akurasi dan performansi yang tinggi dalam pengklasifikasian teks. Karena itu penulis melakukan penelitian yang berjudul "**ANALISIS SENTIMEN BERBASIS ASPEK PADA ULASAN APLIKASI TANGERANG LIVE MENGGUNAKAN *LATENT DIRICHLET ALLOCATION* DAN *NAIVE BAYES***".

1.2. Identifikasi Masalah

Berdasarkan latar belakang yang sudah dijabarkan, maka identifikasi masalahnya adalah:

a. Walaupun aplikasi Tangerang LIVE mendapatkan penilaian yang cukup baik di Google Play Store, terdapat ketidaksesuaian antara bintang penilaian dengan ulasan yang diberikan, yang nantinya jika tidak ada perbaikan pada layanan yang masih kurang membuat pengguna menjadi tidak nyaman dan pengguna baru enggan menggunakan aplikasi ini.
b. Masih mendapat keluhan dari *user* karena layanan yang diberikan belum maksimal seperti: belum semua fitur berjalan dengan baik karena masih terdapat *error* pada beberapa fitur, aplikasi sulit dimengerti oleh beberapa *user*, verifikasi memakan waktu yang lama, dan lain-lain.
c. Belum ada hasil penelitian opini masyarakat tentang aplikasi Tangerang LIVE menggunakan metode *Latent Dirichlet Allocation* dan *Naïve Bayes*.

1.3. Rumusan Masalah

Berdasarkan latar belakang dan identifikasi masalah yang telah dijelaskan, maka bisa dirumuskan masalah dalam penelitian ini adalah:

a. Apa saja aspek dan klasifikasi sentimen yang ada pada aplikasi Tangerang LIVE dengan menggunakan algoritma *Latent Dirichlet Allocation* dan *Naïve Bayes Classifier*?
b. Apa saja kelebihan dan kekurangan dari tiap aspek yang ada pada aplikasi Tangerang LIVE?
c. Bagaimana nilai kinerja algoritma *Latent Dirichlet Allocation* pada pemodelan topik dan *Naïve Bayes Classifier* pada klasifikasi sentimen ulasan aplikasi Tangerang LIVE?

1.4. Batasan Masalah

Batasan masalah pada penelitian ini yaitu:

a. Data yang akan digunakan adalah data ulasan pengguna aplikasi Tangerang LIVE dari aplikasi Google Play Store versi 6.1.0 sampai 6.1.31 yang diambil dengan *Google Play Scraper* dari *library* JoMingyu.
b. Ulasan yang akan diklasifikasi adalah yang menggunakan Bahasa Indonesia.
c. Pemodelan topik dengan *Latent Dirichlet Allocation* dari *library* Gensim.
d. Kelas yang digunakan untuk klasifikasi sentimen hanya positif dan negatif.
e. Menggunakan *Lexicon* untuk pelabelan sentimen, yang selanjutnya data diolah menggunakan metode klasifikasi *Naïve Bayes* dari *library* Scikit Learn.
f. Menggunakan bahasa pemrograman Python serta Google Colab untuk pengolahan data.

1.5. Tujuan Penelitian

Tujuan penelitian yang ingin dicapai dalam penelitian ini adalah:

a. Mengetahui apa saja aspek dan klasifikasi sentimen yang ada pada aplikasi Tangerang LIVE dengan menggunakan algoritma *Latent Dirichlet Allocation* dan *Naïve Bayes Classifier*.
b. Mengetahui hal apa saja yang menjadi kelebihan dan kekurangan dari tiap aspek yang ada pada aplikasi Tangerang LIVE.
c. Mengetahui nilai kinerja algoritma *Latent Dirichlet Allocation* pada pemodelan topik dan *Naïve Bayes Classifier* pada klasifikasi sentimen ulasan aplikasi Tangerang LIVE.

1.6. Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah:

a. Secara teoritis, penelitian ini bisa sebagai referensi dan alternatif sumber untuk penelitian sejenis yang nantinya dilakukan.
b. Secara praktis, hasil penelitian diharapkan bisa sebagai bahan pertimbangan bagi Pemerintah Kota Tangerang untuk perbaikan aplikasi Tangerang LIVE agar kualitas dan layanannya semakin baik lagi.

1.7. Metodologi Penelitian

Penelitian menggunakan data ulasan aplikasi Tangerang LIVE di Google Play Store yang diambil dengan proses *scraping* menggunakan Google Colab dan *browser* Google Chrome. Selanjutnya ulasan dipisahkan berdasarkan simbol titik, karena ulasan yang diberikan *user* terdapat aspek yang berbeda pada kalimat pertama dan selanjutnya. Kemudian dilakukan *pre-processing* karena data yang

didapat belum siap diolah. Selanjutnya kata diubah ke angka agar dapat diolah oleh komputer yang biasa disebut VSM (*Vector Space Model*).

Setelah data bersih dan diberi bobot, dilakukan pemodelan topik dengan LDA, untuk mendapatkan berapa banyak aspek yang digunakan untuk *dataset* ini. Pada sentimen hanya digunakan label negatif dan label positif dengan proses pelabelan sentimen menggunakan *Lexicon*. Kemudian data diklasifikasi menggunakan *Naive Bayes*. Jika ada ketidakseimbangan jumlah data antara kelas yang satu dengan yang lainnya, maka diperlukan *resampling* data untuk mengatasi data yang tidak seimbang, dengan cara *undersampling*, *oversampling* dan gabungan keduanya. Selanjutnya dilakukan perbandingan hasil kinerja algoritma data asli dengan hasil kinerja algoritma setelah di-*resampling*. Setelahnya dilakukan evaluasi kinerja algoritma menggunakan Kurva *Receiver Operating Characteristic*, lalu diinterpretasikan data menggunakan *wordcloud* dari sentimen ditiap aspeknya.

1.8. Sistematika Penulisan

Dalam penyusunan laporan penelitian, pembahasan terbagi dalam lima bab yang secara singkat akan diuraikan sebagai berikut:

BAB 1 PENDAHULUAN

Bab ini berisi penjabaran latar belakang masalah, identifikasi masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

BAB 2 TINJAUAN PUSTAKA

Dalam bab ini dijelaskan teori-teori yang dipergunakan dalam mendukung penelitian ini yang berasal dari buku, jurnal, *e-book*, dan penelitian sejenis.

BAB 3 METODOLOGI PENELITIAN

Bab ini membahas tentang metode-metode yang penulis gunakan dalam penelitian ini seperti metode pengumpulan, pengolahan, pelabelan, dan pengklasifikasian data, yang dilakukan dalam penelitian ini.

BAB 4 HASIL DAN PEMBAHASAN

Bab ini membahas hasil analisis sentimen berbasis aspek pada ulasan aplikasi Tangerang LIVE menggunakan LDA dan *Naive Bayes*. Serta diskusi hasil dengan penelitian sebelumnya.

BAB 5 PENUTUP

Bab ini berisi kesimpulan dari uraian yang sudah dijelaskan pada bab-bab sebelumnya, dan saran-saran untuk pengembang lebih lanjut di masa yang akan datang.

BAB 2

TINJAUAN PUSTAKA

2.1. Perkataan yang Baik

Dalam menyampaikan pendapatnya, seseorang bisa melakukannya secara lisan maupun tulisan. Di zaman yang sudah serba digital, makin banyak tempat seseorang dapat mengutarakan pendapatnya, bisa di media sosial, kolom komentar dll. Sebagai seorang muslim kita harus mampu bertutur kata yang baik, karena Allah SWT sudah mengingatkan kita dalam *al-Qur'an* sebagai berikut:

قَوْلٌ مَّعْرُوْفٌ وَّمَغْفِرَةٌ خَيْرٌ مِّنْ صَدَقَةٍ يَّتْبَعُهَآ اَذًىۗ وَاللّٰهُ غَنِيٌّ حَلِيْم

"Perkataan yang baik dan pemberian maaf lebih baik daripada sedekah yang diiringi tindakan yang menyakiti. Allah Maha kaya, Maha Penyantun." (Q.S Al-Baqarah:263)

Tafsir menurut Kemenag (2022), menyebutkan ayat ini menekankan pentingnya ucapan yang menyenangkan dan pemberian maaf. Perkataan yang baik yang sesuai dengan budaya terpuji dalam suatu masyarakat, yaitu: menolak dengan cara yang baik, tidak dengan cara menyakiti, dan pemberian maaf, yaitu memaafkan tingkah laku yang kurang sopan dari peminta, lebih baik daripada sedekah yang diiringi tindakan yang menyakiti dari pemberi.

Rasulullah SAW pun juga sudah mengingatkan kita umatnya untuk berkata yang baik dalam *hadits* sebagai berikut:

...مَنْ كَانَ يُؤْمِنُ بِاللَّهِ وَالْيَوْمِ الْآخِرِ فَلْيَقُلْ خَيْرًا أَوْ لِيَصْمُتْ ...

"Barangsiapa beriman kepada Allah dan hari akhir hendaknya ia berkata baik atau diam," (HR. al-Bukhari No. 5994).

2.2. Tangerang LIVE

Tangerang LIVE merupakan portal aplikasi yang diluncurkan Pemerintah Kota Tangerang pada 17 Agustus 2016, untuk memberikan kemudahan bagi masyarakat dalam mendapatkan pelayanan publik dan juga salah satu upaya untuk mewujudkan visi Kota Tangerang yang dapat ditemui di Google Play maupun App Store. Aplikasi Tangerang LIVE merupakan aplikasi yang mengintegrasi aplikasi yang sudah di-*launching* pemerintah Kota Tangerang, diantaranya; Sobat Dukcapil, PPDB Online Kota Tangerang, Pangkas, Sigacor, Plesiran dan lainnya.

2.3. Analisis Sentimen

Analisis sentimen atau bisa disebut juga *opinion mining* adalah bidang studi yang menganalisis pendapat, sentimen, evaluasi, penilaian, sikap, dan emosi orang terhadap entitas seperti produk, layanan, organisasi, individu, masalah, peristiwa, topik, dan atributnya. analisis sentimen berfokus pada opini yang mengungkapkan atau menyatakan sentimen positif atau negatif (Liu, 2012).

Analisis sentimen atau *opinion mining* adalah teknik *Natural Language Processing* (NLP) yang digunakan untuk menentukan apakah data positif, negatif, atau netral. Analisis sentimen sering dilakukan pada data tekstual untuk membantu bisnis memantau sentimen *brand* dan produk dalam *feedback* pelanggan, dan memahami kebutuhan pelanggan (MonkeyLearn Inc, 2022). Analisis sentimen umumnya memproses sebuah teks (kalimat, paragraf, buku, dll) dan menghasilkan skor kuantitatif atau klasifikasi untuk menunjukkan apakah algoritma menganggap teks tersebut menyampaikan sentimen positif atau negatif (Saldaña, 2018).

Beberapa tipe analisis sentimen sebagai berikut (MonkeyLearn Inc, 2022):

a) Analisis Sentimen Bertingkat (*Graded Sentiment Analysis*)

Analisis sentimen bertingkat merupakan tipe analisis yang memiliki penilaian spesifik. Tipe ini memungkinkan untuk memperluas kategori polaritas untuk menyertakan tingkat positif dan negatif yang berbeda yaitu: sangat positif, positif, netral, negatif, sangat negatif. Analisis sentimen bertingkat (*fine-grained*) dapat digunakan untuk menafsirkan bintang 5 dalam sebuah ulasan. Bintang 5 untuk menyatakan sangat positif, sedangkan bintang 1 untuk menyatakan sangat negatif.

b) Deteksi Emosi (*Emotion Detection*)

Analisis sentimen pendeteksi emosi digunakan untuk mendeteksi emosi, seperti kebahagiaan, frustrasi, kemarahan, dan kesedihan. Banyak dari sistem pendeteksi emosi menggunakan *Lexicon* (sebuah daftar kata dan emosi yang disampaikan) atau algoritma *machine learning* yang kompleks.

c) Analisis Sentimen Berbasis Aspek (*Aspect-based Sentiment Analysis*)

Analisis sentimen berbasis aspek digunakan saat menganalisis sentimen teks untuk mengetahui aspek atau fitur tertentu yang disampaikan seseorang apakah itu positif, netral, atau negatif. Misalnya pada ulasan sebuah restoran, “Makanannya enak dan murah. Tapi pelayanan dan tempat masih sangat kurang nyaman”. Dari contoh ulasan tersebut dapat dikatakan dari aspek makanan mendapat penilaian yang positif, namun pada aspek pelayanan dan tempat mendapat penilaian negatif.

d) Analisis Sentimen Multibahasa (*Multilingual Sentiment Analysis*)

Analisis sentimen multibahasa digunakan untuk menganalisis teks dalam multibahasa. Tipe analisis sentimen ini dikatakan cukup sulit, karena melibatkan

banyak *preprocessing* dan sumber daya. Sebagian besar sumber dayanya tersedia secara *online* (misalnya *Lexicon* sentimen), sementara yang lain perlu dibuat (misalnya corpora yang diterjemahkan atau algoritma *noise detection*), tetapi harus tahu dahulu cara membuat kode untuk menggunakannya.

Alasan pentingnya analisis sentimen karena manusia mulai mengekspresikan pikiran dan perasaan mereka lebih terbuka dari sebelumnya, analisis sentimen dengan cepat menjadi salah satu alat penting untuk memantau dan memahami sentimen di semua jenis data. Secara otomatis menganalisis *feedback* pelanggan, seperti pendapat mereka dalam tanggapan survei dan percakapan di media sosial, memungkinkan *brand* untuk mempelajari apa yang membuat pelanggan senang atau tidak, sehingga *brand* dapat menyesuaikan produk dan layanan mereka untuk memenuhi kebutuhan pelanggan mereka (MonkeyLearn Inc, 2022).

Menurut Tom (2021), mengenai mengapa analisis itu penting, yang pertama dan terpenting karena emosi dan sikap terhadap suatu topik dapat menjadi informasi yang dapat ditindaklanjuti dan berguna di berbagai bidang bisnis dan penelitian. Kedua, menghemat waktu dan tenaga, karena dalam proses ekstraksi sentimen sepenuhnya otomatis. Ketiga, analisis sentimen menjadi topik yang makin populer seiring dengan berkembangnya *artificial intelligence*, *deep learning*, *machine learning techniques*, dan *natural language processing technologies*. Keempat, dengan berkembangnya teknologi, analisis sentimen akan lebih mudah diakses dan terjangkau oleh publik dan juga perusahaan kecil. Dan terakhir, *tools* yang ada menjadi lebih pintar setiap harinya. Karena semakin banyak *tools* diberikan data, maka semakin pintar dan akurat dalam mengekstraksi sentimen.

Beberapa manfaat analisis sentimen sebagai berikut (MonkeyLearn Inc, 2022):

a) Menyortir Data dalam Skala

Analisis sentimen membantu memproses sejumlah besar data tidak terstruktur dengan cara yang efisien dan hemat biaya.

b) Analisis *Real-Time*

Analisis sentimen dapat mengidentifikasi isu-isu kritis secara *real-time*, sehingga dapat dengan segera mengambil tindakan selanjutnya.

c) Kriteria yang Konsisten

Perusahaan dapat menerapkan kriteria yang sama pada data mereka, sehingga dapat membantu meningkatkan akurasi dan mendapatkan wawasan yang lebih baik.

Analisis sentimen menggunakan berbagai tipe algoritma, yaitu (Tom, 2021):

a) *Rule-based*, Algoritma ini didasarkan pada *Lexicon* yang dibuat secara manual yang mendefinisikan rangkaian kata positif dan negatif. kemudian menganalisis jumlah kata positif dan negatif untuk melihat yang mendominasi.

b) *Automatic*, pada algoritma ini bergantung secara eksklusif pada teknik *machine learning* dan belajar dengan data yang diterima.

c) *Hybrid Systems* menggabungkan elemen yang diinginkan dari teknik *rule-based* dan *automatic* ke dalam satu sistem. Salah satu manfaat besar dari sistem ini adalah bahwa hasilnya sering kali lebih akurat.

Algoritma Pengklasifikasi yang digunakan dalam analisis sentimen biasanya melibatkan model statistik, seperti:

a) *Naïve Bayes* merupakan keluarga algoritma probabilistik yang menggunakan *Teorema Bayes* untuk memprediksi kategori teks.

b) *Linear Regression* merupakan algoritma dalam statistik yang digunakan untuk memprediksi beberapa nilai (*Y*) yang diberikan sekumpulan fitur (*X*)

c) *Support Vector Machines* adalah model non-probabilistik yang menggunakan representasi contoh teks sebagai titik dalam ruang multidimensi.

d) *Deep Learning* merupakan serangkaian algoritma yang mencoba meniru otak manusia, dengan menggunakan jaringan saraf tiruan untuk memproses data.

Tantangan analisis sentimen adalah pada umumnya ulasan yang diberikan pelanggan ditulis dengan bahasa informal, pesan singkat yang menunjukkan isyarat yang terbatas tentang sentimen, serta akronim dan singkatan yang sering digunakan. Tantangan utama dari analisis sentimen *machine-based* seperti: Subjektivitas dan Nada, Konteks dan Polaritas, Ironi dan Sarkasme, Perbandingan, *Emoji*, Mendefinisikan Netral, dan Akurasi *Annotator* Manusia (MonkeyLearn Inc, 2022).

Contoh penggunaan analisis sentimen yaitu (Arviana, 2021):

a) *Social Media Monitoring*

Analisis sentimen digunakan dalam *social media monitoring* untuk memungkinkan bisnis mendapatkan wawasan tentang bagaimana perasaan pelanggan tentang topik tertentu, dan mendeteksi masalah mendesak secara *real time* sebelum masalah tersebut lepas kendali.

b) *Brand Monitoring*

Brand tidak hanya mempunyai informasi yang tersedia pada media sosial, tetapi juga terdapat pada internet, situs berita, blog, forum, ulasan produk, dan masih banyak lagi. Jadi bukan hanya dapat melihat jumlah penyebutan, tetapi juga kualitas individu dan keseluruhan penyebutan.

c) *Customer Feedback*

Customer feedback bertujuan untuk mengumpulkan pendapat dari pelanggan, bisa dengan metode survei dan menganalisisnya dengan teknik analisis sentimen.

d) *Customer Service*

Dengan analisis sentimen dan klasifikasi teks dapat mengatur permintaan dukungan yang masuk berdasarkan topik dan urgensi untuk mengarahkannya ke departemen yang benar dan yang paling mendesak segera ditangani.

e) *Market Research*

Digunakan untuk menganalisis ulasan *online* produk dan membandingkannya dengan produk pesaing. hal tersebut bisa dicari tahu aspek apa dari produk pesaing yang negatif dan gunakan hal tersebut sebagai keuntungan.

Analisis Sentimen Berbasis Aspek (*Aspect-based Sentiment Analysis*)

Mengapa memilih analisis sentimen berbasis aspek, karena pada ulasan aplikasi Tangerang LIVE terdapat ulasan yang di tiap kalimatnya membahas hal yang berbeda. Karena menurut Liu (2010), Saat sebuah kalimat mengandung opini dengan polaritas yang berbeda, maka hal tersebut dapat diidentifikasi dengan analisis sentimen berbasis aspek.

Analisis sentimen berbasis aspek memungkinkan bisnis melakukan analisis mendetail terhadap data *feedback* dari pelanggan mereka, sehingga mereka dapat mempelajari lebih lanjut tentang pelanggan mereka, membuat produk dan layanan yang memenuhi kebutuhan pelanggan. Aspek adalah atribut atau komponen dari sebuah produk atau layanan, misalnya: pengalaman pengguna, waktu respons untuk pertanyaan atau keluhan, atau desain dan harga dari sebuah produk (Pascual, 2019).

Analisis sentimen berbasis aspek penting, karena dapat membantu perusahaan secara otomatis menyortir dan menganalisis data pelanggan, dan mendapatkan wawasan yang kuat dengan praktis. Analisis sentimen berbasis aspek memungkinkan perusahaan untuk memperoleh pemahaman yang lebih dalam tentang produk dan layanan tertentu dengan cepat dan mudah, dan benar-benar fokus pada kebutuhan dan harapan pelanggan mereka (Pascual, 2019).

Berdasarkan sumbernya, dalam mengumpulkan data pada analisis sentimen berbasis aspek dibagi menjadi 2, yaitu (Pascual, 2019):

a) Data Internal, pada data internal terbagi lagi menjadi 3, yakni:
    1) Survei, mengambil data dengan survei dapat mengumpulkan wawasan yang besar untuk perusahaan manapun.
    2) NPS (*Net Promoter Score*), adalah skor yang hasilnya mencerminkan loyalitas, kepuasan, dan antusiasme pelanggan terhadap perusahaan
    3) *Customer Service* dan *Customer Relationship Management* (CRM) *Software*, ini merupakan *software* yang biasanya digunakan dalam bisnis untuk berkomunikasi dengan pelanggan.
b) Data Eksternal, internet memiliki informasi eksternal mulai dari media sosial, artikel berita, ulasan produk, dll. Berikut ini beberapa cara untuk menemukan dan mengumpulkan data yang relevan dari situs web yang berbeda:
    1) *Web Scraping Tools*, ada 2 jenis *web scraping tools*, yaitu: *visual web scraping tools* (*for non-coders*) dan *web scraping frameworks* (*for coders*).
    2) APIs, perusahaan seperti Facebook, Twitter, dan Instagram memiliki API sendiri dan memungkinkan untuk mengekstrak data dari platform mereka.

2.4. *Dataset*

Menurut Khalimi (2020), *dataset* adalah sebuah kumpulan data yang bersifat sebagai himpunan data yang berasal dari informasi-informasi pada masa sebelumnya, dan siap untuk dikelola menjadi sebuah informasi baru. Tujuan dari *dataset* adalah untuk menguji suatu metode penelitian yang dikembangkan oleh para pakar peneliti dengan *public dataset* ataupun *private dataset*.

Dataset dibagi menjadi dua jenis, yaitu:

a) *Private Dataset*

Private dataset adalah *dataset* yang diambil dari sebuah organisasi yang akan dilakukan objek penelitian, seperti data bank, rumah sakit, perusahaan dll.

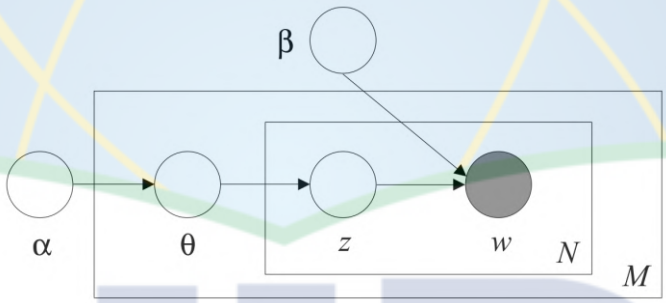
b) *Public Dataset*

Public Dataset adalah *dataset* yang bisa diambil dari repositori publik yang disepakati oleh pakar peneliti *data mining*.

2.5. *Latent Dirichlet Allocation*

Latent Dirichlet Allocation (LDA) adalah salah satu metode pemodelan topik yang paling populer. Pemodelan topik merupakan sebuah metode untuk *unsupervised classification* dokumen, mirip dengan pengelompokan pada data numerik, yang menemukan beberapa kelompok topik bahkan ketika kita tidak yakin dengan apa yang kita cari. Pemodelan topik menyediakan metode untuk mengatur, memahami, mencari, dan meringkas arsip elektronik besar secara otomatis. Pemodelan topik digunakan untuk membantu menemukan tema tersembunyi dalam koleksi, mengklasifikasikan dokumen ke dalam tema yang ditemukan, menggunakan klasifikasi untuk mengatur mencari dokumen (Kulshrestha, 2019).

Metode LDA merupakan salat satu metode pemodelan topik yang paling popular. Selain dapat meringkas, menghubungkan, mengklusterkan topik, LDA mampu untuk memproses data yang sangat besar (Alfanzar, 2019). Metode LDA dapat mengidentifikasi jumlah topik yang optimal, memberi label pada topik, dan menganalisis perbedaan dan kepentingan relatif topik untuk produk yang berbeda (Wang *et al.*, 2018). LDA bertujuan untuk menentukan berapa jumlah topik dari suatu *corpus* dan sebaran kata di tiap topiknya. Pada Gambar 2.1 memperlihatkan representasi model grafis LDA.



Gambar 2.1 Representasi Model Grafis LDA (Blei et al., 2003*)*

LDA didefinisikan notasi berikut (Astuti, 2020):

- Kata merupakan bentuk dasar dari data diskrit.
- Sebuah dokumen merupakan barisan kata-kata *N* yang dinotasikan dengan $\boldsymbol{w} = (w_1, w_2,...,w_N)$ dimana w_N merupakan barisan kata ke-*n*.
- Sebuah *corpus* merupakan koleksi dari *M* dokumen dinotasikan dengan $D = (\boldsymbol{w}_1, \boldsymbol{w}_2,...,\boldsymbol{w}_M)$.

LDA mengasumsikan proses generatif berikut untuk setiap dokumen $\boldsymbol{w}$ dalam korpus *D* (Blei *et al.*, 2003):

- Pilih $N \sim Poisson\ (\xi)$.

- Pilih $\theta \sim Dir(\alpha)$.
- Untuk setiap N kata w_n:
    i. Pilih topik $z_n \sim Multinomial\ (\theta)$.
    ii. Pilih kata w_n dari $p(w_n \mid z_n, \beta)$, probabilitas *multinomial* yang dikondisikan pada topik z_n.

Digunakannya LDA untuk analisis teks karena LDA menyediakan presentasi singkat dari data teks besar untuk mengeksplorasi berbagai topik dan intensitasnya, dapat memanfaatkan sepenuhnya semua ulasan pelanggan dan menghasilkan informasi yang lengkap (Wang *et al.*, 2018).

Penggunaan metode LDA pada analisis topik menghasilkan beberapa implikasi manajerial yang berharga, seperti: memungkinkan desainer produk untuk memahami aspek produk lebih mudah daripada pasar tradisional dan riset produk, perusahaan dalam persaingan produk mampu mengidentifikasi aspek unik, keunggulan kompetitif dan kelemahan produknya dengan para pesaingnya. Oleh karena itu, penentuan posisi pasar yang lebih akurat dan strategi pasar yang tepat dapat dilakukan untuk membantu perusahaan memenangkan persaingan (Wang *et al.*, 2018). Namun LDA memiliki kelemahan dalam mengklasifikasi dokumen ke dalam satu aspek secara langsung (Miller *et al.*, 2016). Ketidakmampuan untuk melabeli topik-topik yang sudah dibentuk juga kelemahan dari metode LDA (Adhitama *et al.*, 2017). Karena pemodelan topik tidak memberi jaminan atas interpretasi keluarannya, maka diperlukannya evaluasi dengan koherensi topik, koherensi topik akan menangkap jumlah topik yang optimal dengan memberikan interpretasi topik ini dengan angka yang disebut nilai koherensi (Kumar, 2018).

Hasil dari proses LDA berupa *keyword* yang nantinya diinterpretasikan menjadi topik atau aspek, maka dari itu diperlukan *list* kata yang merepresentasikan suatu aspek, pada Tabel 2.1 merupakan kata yang merepresentasikan suatu aspek.

Tabel 2.1 Kata yang Merepresentasikan Suatu Aspek
(Areed, 2020; Agarina, 2019; Prajarini, 2018)

Kata	Aspek
Mudah, jelas, konsisten, komunikatif, interaktif	*User Interface*
Kesesuaian, efisien, bagus, membantu, visibilitas, puas	*User Experience*
Proses, layanan, kinerja, mendukung, cepat, berfungsi	*Functionality and Performance*
Update, kritik, saran, harapan, tanya	*Support and Updates*

2.6. *Naïve Bayes*

Naïve Bayes adalah model probabilitas berdasarkan Teorema *Bayes* yang berguna untuk proses klasifikasi. Model klasifikasi *Naïve Bayes* dikenal sangat efisien dan sederhana, serta mempunyai asumsi terhadap independensi dari setiap kondisi sangat kuat, terutama ketika jumlah data *train* yang dimiliki sedikit. Berikut rumus yang dipakai dalam perhitungan *Naïve Bayes* (Raschka, 2014):

$$P(X) = \frac{P(c)P(c)}{P(x)} \quad (2.1)$$

Keterangan:

$P(C|X)$: *Posterior probability*.
$P(c)$: *Prior Probability*.
$P(x|c)$: Probabilitas berdasarkan kondisi hipotesis.
x : Data dengan *class* yang belum diketahui.
c : Hipotesis data yang merupakan suatu *specific class*.

Dalam penerapannya *dataset* dibagi menjadi *data training* dan *data testing*, *data training* merupakan bagian dari *dataset* yang dilatih guna memprediksi atau menjalankan fungsi sebuah algoritma *machine learning*, dan untuk *data testing* merupakan bagian *dataset* yang dites guna melihat keakuratan atau performanya.

Kelebihan yang dimiliki *Naïve Bayes*: pada proses klasifikasi data dapat disesuaikan dengan sifat dan kebutuhan, performansi *Naïve Bayes* memiliki waktu klasifikasi yang singkat sehingga mempercepat proses analisis sentimen (Gunawan *et al.*, 2018). *Naïve Bayes* merupakan metode yang sederhana, namun memiliki akurasi dan performansi yang tinggi dalam pengklasifikasian teks (Routray *et al.*, 2013). Algoritma *Naïve Bayes* bila dibandingkan dengan algoritma klasifikasi yang lain tingkat kesalahan yang dimilikinya sangat minimum (Anggraini *et al.*, 2020).

Naïve Bayes memiliki akurasi dan kecepatan yang baik sehingga dapat diterapkan pada *database Big Data* (Febriansyah *et al.*, 2013). *Naïve Bayes* lebih tepat diaplikasikan pada data yang besar, dapat mengatasi data yang tidak lengkap serta kuat terhadap atribut yang tidak relevan dan *noise* (Muhamad *et al.*, 2017).

Kelemahan pada *Naïve Bayes classifier* dimana sebuah probabilitas tidak bisa mengukur seberapa besar tingkat keakuratan sebuah prediksi, juga memiliki kelemahan pada seleksi atribut yang dapat memengaruhi nilai akurasi. Karena alasan tersebut *Naïve Bayes classifier* perlu dimaksimalkan dengan cara memberikan bobot pada atribut agar kinerjanya lebih efektif (Muhamad *et al.*, 2017). Kelemahan *Naïve Bayes* pada atribut independen kerap terjadi kesalahan dan hasil estimasi probabilitas tidak berjalan dengan optimal (Zaidi *et al.*, 2013).

Kelemahan lainnya *Naïve Bayes* sangat sensitif terhadap pemilihan fitur seleksi, maka diperlukan sebuah pemilihan fitur yang tepat dengan model yang diusulkan (Somantri & Khambali, 2017). Diperlukannya atribut dengan nilai diskrit sehingga membutuhkan proses diskritisasi untuk mengubah atribut kontinu ke dalam bentuk diskrit (Wirawan & Eksistyanto, 2015).

2.7. *Web Scraping*

Web Scraping adalah sebuah proses pengambilan dokumen semi-terstruktur dari internet, yang biasanya berupa halaman-halaman *web* dalam bahasa XHTML atau HTML, lalu dokumen tersebut dianalisis untuk diambil data tertentu untuk digunakan sesuai kebutuhan. *Web scraping* berfokus pada bagaimana memperoleh data serta mengekstrak data dalam ukuran yang bervariasi (Turland, 2010). Teknik *web scraping* merupakan suatu teknik yang sangat bermanfaat untuk mendapatkan artikel ilmiah secara cepat dari halaman web (Josi & Abdillah, 2014). Hasil dari *Web Scraping* biasanya disimpan dalam bentuk data tabular/tabel dengan *array* dua dimensi, yaitu baris dan kolom yang disebut *Data Frame*. Setiap baris pada *data frame* terdiri dari elemen, dan kolom adalah objek dari *Series* (Rezkia, 2021).

Teknik-teknik dalam *web scraping* sebagai berikut (Pradana, 2021):

a) Menyalin Data Manual
b) *Parsing* HTML
c) *Parsing* DOM
d) Menggunakan XPath
e) Menggunakan *Regular Expression*
f) Menggunakan *Text Pattern Matching*

Beberapa manfaat *web scraping* yang lain menurut Pradana (2021), yaitu:

a) Mencari Informasi
b) Mendapatkan *Leads*
c) Memantau Berita dan Konten
d) Optimasi Harga Produk maupun Layanan
e) Mendalami kebutuhan Konsumen dari Kompetitor

Kendala dalam melakukan *web scraping* sebagai berikut (Pradana, 2021):

a) Tidak ada teknik *web scraping* yang 100% efektif.
b) Data yang diperoleh tidak selalu rapi, masih terdapat teks yang tidak diinginkan dan perlu merapikan data hasil web *scraping*.

c) Pemahaman tentang struktur halaman *website* tetap penting.

d) Akses ke suatu laman bisa diblokir, jika terlalu sering melakukan *web scraping* di suatu *website*, admin *website* tersebut dapat memblokir IP.

e) Tidak semua laman *website* mudah di ekstrak datanya, karena adanya pembaruan *website* untuk alasan keamanan salah satunya.

2.8. *Preprocessing*

Sebelum data diolah *machine learning*, diperlukan langkah *preprocessing* agar format data sesuai, yang nantinya dapat diproses dengan algoritma klasifikasi yang akan digunakan (Budi, 2017). Berikut tahapan *preprocessing* penelitian ini:

2.8.1. *Cleansing*

Proses ini merupakan tahapan dimana *dataset* akan dilakukan pembersihan dari *emoticons*, angka, *whitespace*, dan tanda baca. Tanda baca atau simbol dihapus karena tidak memiliki pengaruh dalam hasil analisis sentimen (Sidiq, 2019).

2.8.2. *Lowercase Conversion*

Dilakukannya proses ini karena bentuk kata dengan huruf besar atau kecil dianggap tidak memiliki perbedaan, semua karakter huruf dalam teks diubah menjadi huruf kecil (Uysal & Gunal, 2014).

2.8.3. *Tokenization*

Tokenization merupakan prosedur pemecahan teks menjadi kata, frasa, atau bagian lain yang bermakna yaitu *token*. Dengan kata lain, *tokenization* adalah bentuk segmentasi teks. Umumnya, segmentasi dilakukan hanya dengan mempertimbangkan karakter alfabet atau numerik yang dibatasi oleh karakter non-alfanumerik (misal: tanda baca & spasi) (Uysal & Gunal, 2014).

2.8.4. *Normalize*

Normalisasi adalah proses mengubah singkatan dan kata *slang* menjadi kata yang maknanya sama. Tahap ini diperlukan untuk *dataset* yang berasal dari sosial media yang menggunakan kata singkatan dan kata tidak baku (Ganesan, 2019).

2.8.5. *Stopword Removal*

Stopword adalah kata-kata yang biasa ditemui dalam teks tanpa ketergantungan pada topik tertentu (misal: konjungsi & preposisi) (Uysal & Gunal, 2014). Oleh karena itu dilakukan penghapusan kata-kata yang dianggap tidak sesuai atau sering muncul seperti: ‘di’, ‘yang’, ‘dan’, dan lainnya (Yutika *et al.*, 2021).

2.9. *Lexicon Based*

Lexicon adalah sebuah metode pendekatan berbasis kamus atau *Dictionary Based Approach*. Metode *Lexicon* tekniknya membuat daftar kata opini yang umum dipakai seseorang guna menandakan bahwa kalimat yang dikemukakan adalah kalimat opini. Daftar kata pada *Lexicon* umumnya kata sifat yang dipakai sebagai indikator pada kalimat opini, seperti indah, jelek, bagus, dll (Widyanto, 2017).

Metode *Lexicon* memiliki kelebihan karena data yang berupa kata dari suatu kalimat dibandingkan langsung dengan kamus kata yang ada pada *Lexicon*. Apabila dalam suatu kalimat terdapat kata opini, kalimat tersebut dinyatakan sebagai kalimat opini (Widyanto, 2017). keuntungan *Lexicon* adalah tenaga kerja manual hanya diperlukan ketika membangun kamus sentimen saja (Kannan *et al.*, 2016).

Namun *Lexicon* memiliki kekurangan, apabila kata pada suatu kalimat tidak ada pada *Lexicon*, maka kalimat tersebut dinyatakan bukan kalimat opini, padahal mungkin saja kalimat tersebut kalimat opini (Widyanto, 2017).

Tahapan dalam metode *Lexicon* (Widyanto, 2017):

a. Mengumpulkan kata-kata yang dianggap mewakili kalimat opini.
b. Memproses *dataset* dengan *text preprocessing*.
c. Memecah kalimat menjadi per kata dan dibandingkan dengan tiap kata yang ada pada daftar kata opini.
d. Apabila kata yang ada pada kalimat tersebut sama dengan kata yang ada pada daftar kata opini, kalimat tersebut dinyatakan kalimat opini.

2.10. *Vector Space Model* (VSM)

Data dibagi menjadi data terstruktur dan data tidak terstruktur. Data terstruktur umumnya berbentuk tabular (tabel, matriks, baris-kolom). Sedangkan data tidak terstruktur berbentuk seperti data gambar, audio atau data teks. Data yang seperti itu tidak dapat langsung diolah, karena komputer hanya dapat mengolah data angka. Maka perlu yang namanya *Vector Space Model* (VSM) untuk mengonversikan dokumen menjadi nilai vektor (Astuti, 2020).

Setiap nilai dalam vektor merepresentasikan bobot dari kata dalam dokumen. TF-IDF (*Term Frequency-Inverse Document Frequency*) merupakan cara yang banyak digunakan dalam memberikan bobot pada kata. TF-IDF bertujuan untuk memberikan bobot pada kata *t* dalam dokumen *d* sesuai dengan rumus berikut:

$$weight\ (t,d) = tf\ (t,d) \times\ idf\ (t,D) \qquad (2.2)$$

Keterangan:

t = Kata
d = Dokumen
D = *Corpus*
tf (*t*,*d*) = *Frequency t* di *d*
idf (*t*,*D*) = *Inverse Document Frequency* dari *t* di *D*

Nilai TF-IDF mendapati angka tertinggi pada saat suatu kata *t* muncul berkali-kali dalam jumlah dokumen yang sedikit. Jadi lebih rendah saat suatu kata *t* muncul lebih sedikit dalam satu dokumen atau banyak dokumen. Nilai TF-IDF paling rendah pada saat kata muncul hampir di semua dokumen (Schütze *et al.*, 2008).

Term frequency (*tf*) akan memberitahukan berapa banyak kata yang muncul dalam tiap dokumen. Yang menunjukkan pentingnya kata tersebut dalam suatu dokumen. Makin tinggi bobot *tf* maka semakin banyak kemunculan suatu kata dalam dokumen. Dengan rumus *tf* sebagai berikut:

$$tf = \{1 + (f_{t,d}) \, , \, f_{t,d} > 0 \; 0, \, f_{t,d} = 0 \tag{2.3}$$

Inverse document frequency (*idf*) memberitahukan jarangnya suatu kata yang muncul pada suatu dokumen. Kata yang jarang muncul tersebut berguna untuk membedakan satu dokumen dengan dokumen yang lainnya. penghitungan *idf* merupakan kebalikan dari *df* (*document frequency*) (Wang *et al.*, 2014). Dengan rumus *idf* sebagai berikut:

$$idf = \left(\frac{N}{df_t}\right) \tag{2.4}$$

Keterangan:
N = Banyaknya Dokumen
df_t = Banyaknya dokumen dalam *corpus* yang memuat kata *t*

Bila nilai *idf* tinggi maka kemunculan kata tersebut jarang, sedangkan bila nilai *idf* rendah maka kemunculan kata sering muncul (Schütze *et al.*, 2008).

2.11. *Dataset* Tidak Seimbang

Dataset yang tidak seimbang adalah kondisi dimana *dataset* memiliki kemiringan parah dalam distribusi kelas, seperti contoh 1:100 atau 1:1000 di kelas

minoritas hingga kelas mayoritas. Bias dalam *dataset* dapat memengaruhi banyak algoritma *machine learning*, sehingga sebagian mengabaikan kelas minoritas sepenuhnya. Ini merupakan masalah karena biasanya kelas minoritas dimana prediksi paling penting.

Salah satu pendekatan untuk mengatasi masalah data tidak seimbang dengan *randomly resample dataset* yang tidak seimbang. Ada beberapa pendekatan untuk melakukan *randomly resample dataset* yang tidak seimbang (Brownlee, 2020):

2.11.1. *Undersampling*

Random Undersampling melibatkan pemilihan contoh secara acak dari kelas mayoritas untuk dihapus dari *training dataset.* Ini memiliki efek mengurangi jumlah contoh di kelas mayoritas dalam versi *training dataset* yang diubah.

Pendekatan ini mungkin lebih cocok untuk *dataset* yang memiliki ketidakseimbangan kelas meskipun cukup banyak contoh di kelas minoritas, model yang berguna seperti itu bisa cocok.

Keterbatasan *undersampling* adalah bahwa contoh dari kelas mayoritas dihapus yang mungkin berguna, penting, atau mungkin kritis untuk menyesuaikan batas keputusan yang kuat. Mengingat bahwa contoh dihapus secara acak, tidak ada cara untuk mendeteksi atau mempertahankan contoh yang "baik" atau lebih kaya informasi dari kelas mayoritas.

2.11.2. *Oversampling*

Random oversampling melibatkan duplikasi contoh secara acak dari kelas minoritas dan menambahkannya ke *training dataset.* Contoh dari *training dataset* dipilih secara acak dengan penggantian. Ini berarti bahwa contoh dari kelas

minoritas dapat dipilih dan ditambahkan ke kumpulan data pelatihan baru yang "lebih seimbang" beberapa kali; mereka dipilih dari kumpulan data pelatihan asli, ditambahkan ke kumpulan data pelatihan baru, dan kemudian dikembalikan atau "diganti" dalam kumpulan data asli, memungkinkan mereka untuk dipilih kembali.

Teknik ini bisa efektif untuk algoritma *machine learning* yang dipengaruhi oleh distribusi miring dan dimana beberapa contoh duplikat untuk kelas tertentu dapat memengaruhi kecocokan model. Mungkin berguna untuk menyetel distribusi kelas target. Dalam beberapa kasus, mencari distribusi yang seimbang untuk *dataset* yang sangat tidak seimbang dapat menyebabkan algoritma yang terpengaruh menjadi *overfit* pada kelas minoritas, yang menyebabkan peningkatan kesalahan generalisasi. Efeknya bisa berupa kinerja yang lebih baik pada set data pelatihan, tetapi kinerja yang lebih buruk pada *dataset* uji atau *holdout.*

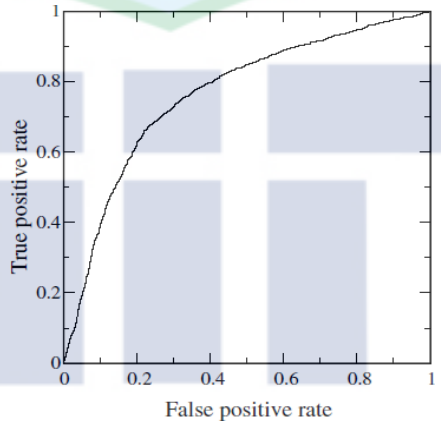
2.11.3. *SMOTE*

Hasil yang menarik dapat dicapai dengan menggabungkan *random oversampling* dan *undersampling*. Misalnya, sejumlah kecil sampel berlebih dapat diterapkan pada kelas minoritas untuk meningkatkan bias, dan juga menerapkan sejumlah kecil sampel kecil pada kelas mayoritas untuk mengurangi bias pada kelas tersebut. Hal ini dapat menghasilkan peningkatan kinerja secara keseluruhan dibandingkan dengan melakukan satu atau teknik lain secara terpisah.

Misalnya, kita memiliki *dataset* dengan distribusi kelas 1:100, pertama kita menerapkan *oversampling* untuk meningkatkan rasio menjadi 1:10 dengan menduplikasi contoh dari kelas minoritas, kemudian menerapkan *undersampling* untuk meningkatkan rasio ke 1:2 dengan menghapus contoh dari kelas mayoritas.

2.12. Kurva *Receiver Operating Characteristic* (ROC)

Keunggulan dari grafik ROC, pada grafik tersebut memungkinkan untuk memvisualisasi serta mengatur kinerja *classifier* tanpa memperhatikan distribusi kelas atau *error cost* (biaya kesalahan). Kurva ROC merupakan visualisasi grafik dari *precision* dan *recall*, yang diinterpretasikan dengan plot FPR (*False Positive Rate*) pada sumbu *x* dan TPR (*True Positive Rate*) pada sumbu *y*. TPR yang bisa disebut juga *recall* merupakan rasio prediksi benar positif dibandingkan keseluruhan data yang benar positif. FPR merupakan persentase kesalahan memprediksi positif yang data aslinya negatif. Model akan dikatakan baik apabila nilai TPR tinggi dan FPR rendah. Nilai TPR dan FPR mengacu pada *confusion matrix* (Astuti, 2020). Pada Gambar 2.2 berikut merupakan ilustrasi Kurva ROC.



Gambar 2.2 Ilustrasi Kurva ROC (Astuti, 2020)

Pada tabel *confusion matrix* baris merupakan nilai prediksi dan kolom merupakan nilai aktual dari nilai positif dan negatif. Angka pada diagonal utama menggambarkan keputusan yang benar, sedangkan diagonal yang lain menggambarkan *error* antar kelas. Nilai prediksi merupakan klasifikasi yang dilakukan program setelah data dimasukkan ke dalam model. Sedangkan nilai aktual merupakan klasifikasi yang sudah dilakukan pada tahap *preprocessing*.

Tabel 2.2 berikut merupakan contoh dari tabel *confusion matrix*.

Tabel 2.2 Tabel *Confusion Matrix* (Fawcett, 2006)

		Nilai Aktual	
		positif (1)	negatif (0)
Nilai Prediksi	positif (1)	TP	FP
	negatif (0)	FN	TN

Keterangan pada tabel *confusion matrix* sebagai berikut:

TP (*True Positive*) : Secara benar memprediksi data positif

TN (*True Negative*) : Secara benar memprediksi data negatif

FP (*False Positive*) : Secara salah memprediksi bahwa data positif

FN (*False Negative*) : Secara salah memprediksi bahwa data negatif

Nilai TPR dihitung menggunakan rumus berikut (Fawcett, 2006):

$$TPR = recall \approx \frac{positive\ correctly\ classified}{total\ positive} = \frac{TP}{TP+FN} \quad (2.5)$$

Rumus untuk menghitung nilai FPR sebagai berikut:

$$FPR \approx \frac{negative\ incorrectly\ classified}{total\ negative} \quad (2.6)$$

FPR dinyatakan sebagai 1 – *specificity* dengan rumus *specificity* sebagai berikut:

$$specificity = \frac{TN}{FP+FN} \quad (2.7)$$

Kumpulan pasangan koordinat TPR dengan FPR akan membuat sebuah kurva ROC, berdasar pada *threshold* yang telah ditentukan. Nilai TP didapat dengan menghitung peluang positif (benar) yang melebihi *threshold*, sedangkan nilai FP didapat dengan menghitung peluang negatif (salah) yang melebihi *threshold*. Selanjutnya nilai FP dan TP dibagi dengan total jumlah data, hasilnya merupakan pasangan koordinat (FPR, TPR). *Range* dari nilai FPR dan TPR mulai 0 sampai 1.

Nilai Klasifikasi ROC didapat dengan menghitung luas area di bawah kurva ROC, atau disebut dengan *Area Under Curve* (AUC). Tabel 2.3 berikut adalah kriteria nilai AUC.

Tabel 2.3 Tabel Nilai AUC dan Interpretasinya (Suwarno & Abdillah, 2016).

Nilai AUC	Interpretasi
0,9 – 1	Klasifikasi sangat baik
0,8 – 0,9	Klasifikasi baik
0,7 – 0,8	Klasifikasi cukup
0,6 – 0,7	Klasifikasi lemah
0,5 – 0,6	Gagal

2.13. Penelitian Sejenis

Penelitian sejenis berisi penelitian yang telah dilakukan sebelumnya yang digunakan penulis sebagai referensi dalam penelitian karena memiliki keterkaitan dengan tema penelitian ini. Sumber-sumber penelitian sejenis diambil dari jurnal dan skripsi yang di dalamnya membahas mengenai analisis sentimen berbasis aspek yang menggunakan metode *Naïve Bayes, Latent Dirichlet Allocation, Lexicon*.

Mengacu pada Tabel 2.4, penelitian ini merujuk pada penelitian yang dilakukan oleh Astuti (2020) sebagai acuan utama. Penelitian yang dilakukan oleh Astuti (2020), menggunakan ulasan aplikasi dengan Bahasa Inggris sebagai *dataset* serta melakukan pelabelan sentimen dan aspek secara manual. Sedangkan pada penelitian ini menggunakan ulasan aplikasi dengan Bahasa Indonesia serta menggunakan metode *Lexicon* untuk pelabelan sentimen. Pelabelan sentimen tidak dilakukan secara manual agar dalam pemberian label sentimen terhindar dari bias opini pribadi dan dapat mengefisiensi waktu (Azhar, 2018).

Tabel 2.4 Penelitian Terdahulu

No	Judul	Penulis	Metode dan *Tools*	*Dataset*	Hasil
1	Analisis Sentimen Berbasis Aspek pada Aplikasi Tokopedia Menggunakan LDA dan *Naïve Bayes*	Astuti (2020)	▪ *Latent Dirichlet Allocation* dan *Naïve Bayes* ▪ *WinPyhton*	Data ulasan pengguna Tokopedia pada Google Play Store dari Oktober 2018 sampai Mei 2019 berbahasa Inggris berjumlah 4.425 kalimat.	Analisis sentimen berbasis aspek menggunakan LDA untuk *clustering topic* mendapat 4 aspek: tampilan, pengalaman belanja, pelayanan dan kemanfaatan. Klasifikasi sentimen menggunakan *Naïve Bayes* menghasilkan nilai akurasi 92,5% serta nilai AUC sebesar 0,95.
2	Analisis Sentimen dan Pemodelan Topik Pariwisata Lombok Menggunakan Algoritma *Naïve Bayes* dan *Latent Dirichlet Allocation*	Putu & Amrullah (2021)	▪ *Latent Dirichlet Allocation* dan *Naïve Bayes* ▪ Tidak menyebutkan *software*	Data *tweet* dari Twitter dari tahun 2014 sampai 2019 sebanyak 9.396 *tweet.* Pelabelan *dataset* secara manual.	Klasifikasi sentimen menggunakan *Naïve Bayes* menghasilkan nilai akurasi 92%, presisi 100%, *recall* 83,84, dan *specificity* 100%. Pemodelan topik menggunakan LDA menghasilkan 8 topik untuk kelas positif nilai koherensi 0,613 dan 12 topik untuk kelas negatif nilai koherensi 0,528.

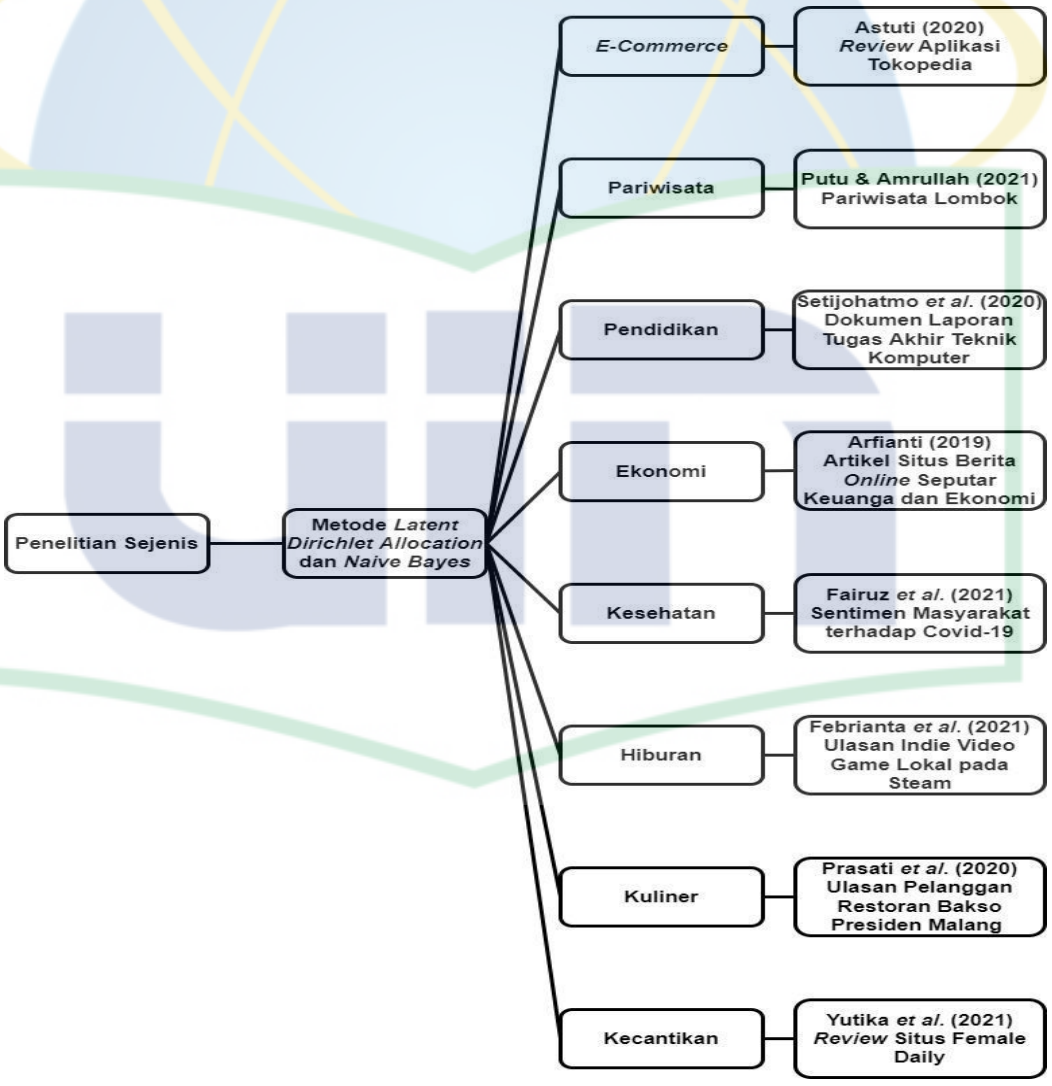
3	Analisis Metoda *Latent Dirichlet Allocation* untuk Klasifikasi Dokumen Laporan Tugas Akhir Berdasarkan Pemodelan Topik	Setijohatmo *et al*. (2020)	▪ *Latent Dirichlet Allocation* ▪ Tidak menyebutkan *software*	Data kata dari abstrak tugas akhir Jurusan Teknik Komputer berbahasa Indonesia sebanyak 144 dokumen.	Metode LDA sensitif pada komposisi kata ketika data yang dipakai mengandung banyak kata umum hasilnya akan mengurangi tingkat presisi. LDA dapat mengelompokkan dokumen dengan topik tertentu tetapi tidak berlabel. Penggunaan metode LDA tingkat relevansinya 75%.
4	Identifikasi Topik Artikel Berita Menggunakan *Topic Modelling* dengan Metode *Latent Dirichlet Allocation*	Arfianti (2019)	▪ *Latent Dirichlet Allocation* ▪ Tidak menyebutkan *software*	Data adalah artikel dari situs berita *online* seputar keuangan dan ekonomi berbahasa Indonesia sebanyak 339 artikel	Hasil dari penelitian ini mendapat tingkat akurasi 89,1% pada pengujian artikel terhadap topik. Parameter yang tepat untuk menemukan hasil distribusi topik yang paling mewakili persebaran topik pada *dataset* adalah 14 topik, 4 kata pada topik, 5 interval dan 10 *burn-in*/iterasi.

5	Analisis Sentimen Masyarakat Terhadap COVID-19 pada Media Sosial Twitter	Fairuz *et al.* (2021)	▪ *Naïve Bayes* dan *K-Nearest Neighbor* ▪ Tidak menyebutkan *software*	Data *tweet* dari Twitter sebanyak 1000 *tweet* dalam kurun waktu 4 September sampai 12 November 2020.	Penggunaan algoritma *Naïve Bayes* mendapatkan hasil akurasi 85%, sedangkan algoritma *K-Nearest Neighbor* mendapat 82% pada saat nilai *k* 6, 8 dan 14.
6	Analisis Ulasan Indie *Video Game* Lokal pada Steam Menggunakan *Sentiment Analysis* dengan Algoritma *Naïve Bayes Classifier* dan LDA-*Based Topic Modelling*	Febrianta *et al.* (2021)	▪ *Latent Dirichlet Allocation* dan *Naïve Bayes* ▪ RStudio	Data ulasan 24 judul *indie video game* lokal berbahasa Inggris dari *web Steam* sebanyak 3497 ulasan.	Klasifikasi sentimen menggunakan *Naïve Bayes* menghasilkan nilai akurasi 83,69%. Pada pemodelan topik menggunakan LDA menghasilkan 5 kata pada sentimen positif dan 5 kata pada sentimen negatif.

7	Analisis Sentimen Berbasis Aspek pada Ulasan Pelanggan Restoran Bakso Presiden Malang dengan Metode *Naïve Bayes Classifier*	Parasati *et al.* (2020)	▪ *Naïve Bayes* dan TF-IDF ▪ Tidak menyebutkan *software*	Data ulasan dari *TripAdvisor* dan *Google Review* sejumlah 2.152 dari tahun 2012 sampai 2019 berbahasa Indonesia.	Penggunaan *Naïve Bayes Classifier* menghasilkan nilai akurasi 88% pada aspek Makanan, 76% pada aspek Layanan dan 84% pada aspek Atmosfir.
8	Analisis Sentimen Berbasis Aspek pada *Review* Female Daily Menggunakan TF-IDF dan *Naïve Bayes*	Yutika *et al.* (2021)	▪ *Naïve Bayes* dan TF-IDF ▪ Tidak menyebutkan *software*	Data *review* Female Daily sebanyak 5054 multilingual.	Performansi tertinggi jika *dataset* di terjemahkan ke dalam Bahasa Inggris lalu diterjemahkan ke Bahasa Indonesia serta tidak menggunakan *stopword removal* dengan parameter *smoothing* atau alpha sebesar 1, *min_df* 0,01, *max_df* 0,7, dan *max_features* 2000 mendapat hasil *F1-Score* 62,81%.

2.14. Ranah Penelitian

Pada ranah penelitian digambarkan ranah penelitian sejenis yang dilakukan penulis berdasarkan penelitian sejenis yang menggunakan metode *Latent Dirichlet Allocation* dan *Naïve Bayes* pada bidang tertentu. Penelitian yang dilakukan penulis berkaitan dengan bidang pemerintahan mengenai klasifikasi topik dan sentimen ulasan aplikasi Tangerang LIVE menggunakan metode *Latent Dirichlet Allocation* dan *Naïve Bayes*. Ilustrasi ranah penelitian dapat dilihat pada Gambar 2.3 berikut.



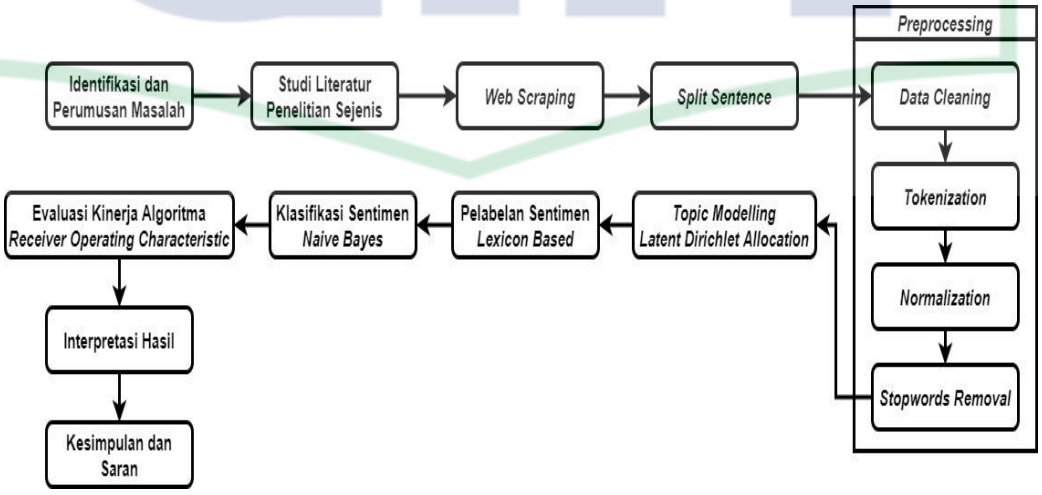
Gambar 2.3 Ranah Penelitian

BAB 3

METODE PENELITIAN

3.1. Alur Penelitian

Penelitian ini dimulai dengan identifikasi dan perumusan masalah, selanjutnya mencari studi literatur penelitian sejenis. Untuk data penelitian menggunakan data ulasan aplikasi yang didapat dengan teknik *web scraping*. Setelahnya data ulasan dipotong pada tiap simbol pemisah titik dan dilanjutkan dengan tahap *preprocessing*. Data yang telah dibersihkan masuk ke tahap *topic modelling* menggunakan metode *Latent Dirichlet Allocation*. Tahap berikutnya data diberikan sentimen menggunakan metode *Lexicon based*. Setelah data diberi label sentimen, data akan diklasifikasi dengan metode *Naïve Bayes classifier.* Hasil dari klasifikasi sentimen dievaluasi kinerja algoritmanya dengan *Receiver Operating Characteristic.* Lalu hasil sentimen tiap aspek divisualisasikan dengan *wordcloud* dan diinterpretasikan kekurangan dan juga kelebihannya. Alur penelitian ini diperlihatkan pada Gambar 3.1 berikut.



Gambar 3.1 Alur Penelitian

3.2. Sumber Data

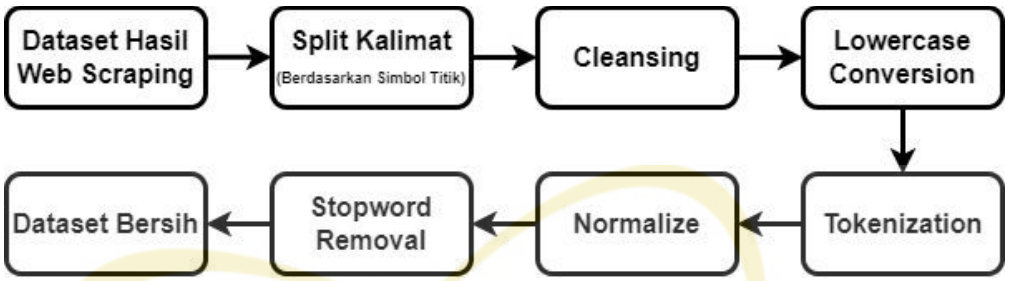
Sumber data yang digunakan merupakan data sekunder yang berupa ulasan berbahasa Indonesia pada aplikasi Tangerang LIVE yang diberikan oleh *user* pada aplikasi Google Play Store. Pengambilan data dilakukan pada 8 April 2022 dan data yang didapatkan sebanyak 2.419 data ulasan dari versi 6.1.0 sampai 6.1.31 dengan menggunakan teknik *web scraping*.

Pada hasil *scraping* hanya diambil beberapa kolom, yaitu: *username*, *content*, *review created version*, *score*, dan *at*. Kolom *username* berisi nama dari pemberi ulasan, kolom *content* berisi ulasan aplikasi yang diberikan oleh *user*, kolom *review created version* merupakan versi aplikasi berapa yang digunakan *user* pada saat memberi ulasan, kolom *score* merupakan penilaian yang diberikan *user* untuk aplikasi, dan kolom *at* merupakan waktu ulasan tersebut diberikan.

3.3. *Text Preprocessing*

Data dari proses *scraping* akan dipisahkan berdasarkan simbol titik. Selanjutnya data melewati langkah *preprocessing* agar dapat diproses dengan algoritma yang akan digunakan (Budi, 2017). Untuk alur *preprocessing* dapat dilihat pada Gambar 3.2. Berikut langkah *text preprocessing* pada penelitian ini:

1. Memisahkan data ulasan berdasarkan simbol titik.
2. Membersihkan data dari *emoticon*, karakter tunggal, tanda baca, angka, spasi berlebih dan membuat data menjadi huruf kecil semua.
3. Membuat data menjadi bentuk *token*.
4. Mengubah kata singkatan dan kata *slang* menjadi kata yang ada pada *Lexicon*
5. Menghilangkan kata yang sering muncul serta tidak memiliki kesesuaian.



Gambar 3.2 Alur *Preprocessing*

3.4. Metode Analisis Sentimen

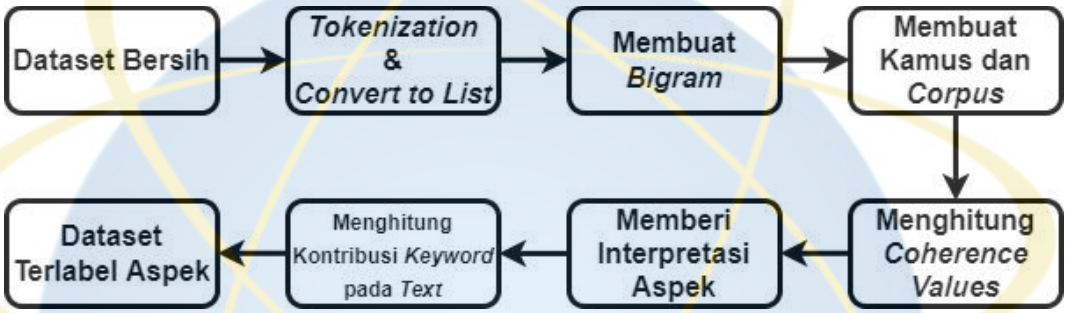
Metode yang digunakan pada penelitian ini, untuk *topic modelling* menggunakan metode *Latent Dirichlet Allocation* (LDA), untuk pelabelan sentimen tidak dilakukan secara manual, melainkan menggunakan metode *Lexicon Based*, dan untuk pengklasifikasian sentimen menggunakan metode *Naïve Bayes*.

3.4.1. *Latent Dirichlet Allocation* (LDA)

Digunakannya LDA karena kelebihannya yaitu, LDA dapat meringkas, mengklusterkan topik, mengidentifikasi jumlah topik yang optimal, dan memberi label topik pada *dataset*. Hasil dari LDA untuk menentukan jumlah aspek terbaik berdasarkan *coherence value* tertinggi, dan selanjutnya dijadikan acuan untuk pelabelan otomatis pada data ulasan. Untuk alur LDA dapat dilihat pada Gambar 3.3. Berikut ini langkah-langkah penggunaan LDA pada penelitian ini:

1. Membuat bentuk dokumen menjadi *token* atau kata dan di *convert* ke bentuk *list*.
2. Membuat *bigram*, *bigram* merupakan dua kata yang sering muncul bersama dalam dokumen.
3. Membuat kamus untuk mendapatkan kata unik, lalu dibuat *corpus*nya.
4. Menghitung *coherence values* di setiap jumlah topik lalu pilih topik dengan *coherence values* tertinggi.

5. Memberi interpretasi aspek dari hasil *topic modelling clustering* dengan nama yang mewakili aspek dominan pada tiap kalimat berdasarkan *keyword*.
6. Tiap baris ulasan akan dihitung kontribusi persentase topik berdasarkan kata kunci, dan hasilnya tiap baris ulasan dapat dilabeli aspek berdasarkan kontribusi persentase topik.

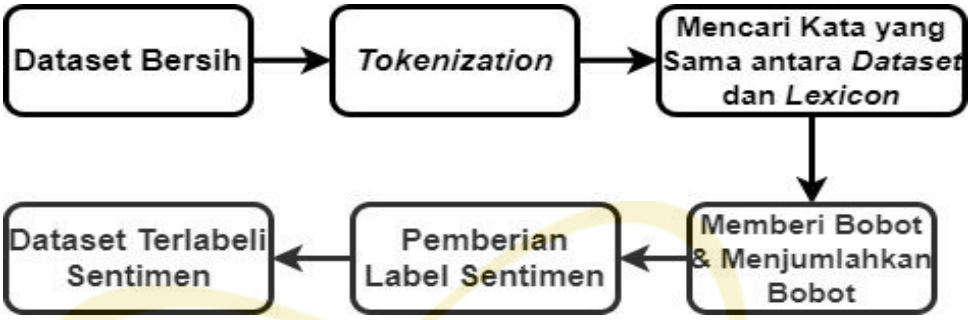


Gambar 3.3 Alur *Latent Dirichlet Allocation* (LDA)

3.4.2. *Lexicon Based*

Digunakannya *Lexicon* pada penelitian ini agar dalam pelabelan sentimen terhindar dari bias opini pribadi. Untuk alur *Lexicon Based* dapat dilihat pada Gambar 3.4. Berikut ini langkah-langkah penggunaan *Lexicon* pada penelitian ini:

1. Memecah kalimat menjadi bentuk *token* atau kata.
2. Memeriksa setiap kata, jika kata muncul dalam *Lexicon*, kemudian beri bobot.
3. Menjumlahkan bobot setiap baris ulasan.
4. Tiap baris ulasan akan dilabeli berdasarkan nilai penjumlahan, jika hasil penjumlahan bobot sentimen dari sebuah kalimat lebih dari 0 dinyatakan positif, jika kurang dari 0 dinyatakan negatif.



Gambar 3.4 Alur *Lexicon Based*

3.4.3. *Naïve Bayes*

Dalam penelitian ini pengklasifikasian sentimen menggunakan metode *Naïve Bayes* karena *Naïve Bayes* merupakan metode yang sederhana, namun memiliki akurasi dan performansi yang tinggi dalam pengklasifikasian teks (Routray *et al.*, 2013). Untuk alur *Naïve Bayes* dapat dilihat pada Gambar 3.5. Berikut ini langkah-langkah penggunaan *Naïve Bayes* pada penelitian ini:

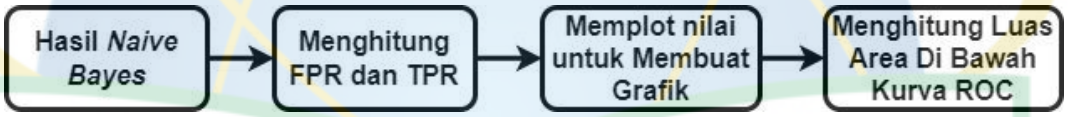
1. Membagi *dataset* menjadi data *train* dan data *test.*
2. Melakukan percobaan menghitung nilai akurasi beberapa perbandingan rasio data *train* dan data *test,* dan ambil rasio dengan nilai akurasi paling tinggi.
3. Karena data tidak seimbang, gunakan teknik *undersampling*, *oversampling* dan gabungan keduanya, lalu pilih teknik yang memiliki nilai akurasi paling tinggi.

Gambar 3.5 Alur *Naïve Bayes*

3.5. Evaluasi Kinerja Algoritma

Untuk mengetahui seberapa baik kinerja klasifikasi diperlukannya evaluasi kinerja algoritma klasifikasi yang pada penelitian ini menggunakan Kurva *Receiver Operating Characteristic* (ROC) dibuat dengan berdasarkan nilai perhitungan pada *confusion matrix*. Untuk alur evaluasi kinerja algoritma dapat dilihat pada Gambar 3.6. Berikut ini langkah-langkah penggunaan Kurva ROC pada penelitian ini:

1. Menghitung *false positive rate* dan *true positive rate*.
2. Mem-*plot* nilai yang telah dihitung untuk membuat grafik.
3. Menghitung luas area di bawah Kurva ROC.



Gambar 3.6 Alur Evaluasi Kinerja Algoritma dengan ROC

BAB 4

HASIL DAN PEMBAHASAN

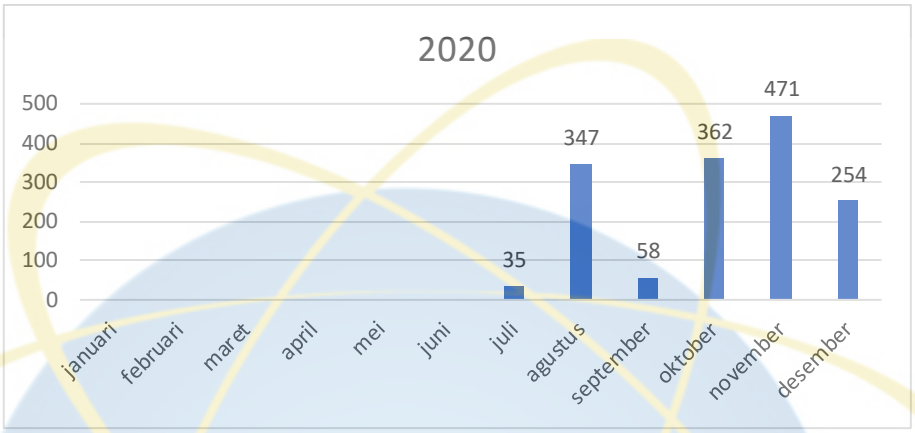
4.1 Pengumpulan Data

Data ulasan yang didapat dengan proses *web scraping* pada tanggal 8 April 2022 sebanyak 6622 baris serta 10 kolom. Namun penulis hanya mengambil 5 kolom yang dibutuhkan serta membatasi hanya ulasan versi 6.1.0 sampai 6.1.31 saja yang diambil sebanyak 2419 baris. Tabel 4.1 menunjukkan beberapa sampel hasil dari *web scraping*.

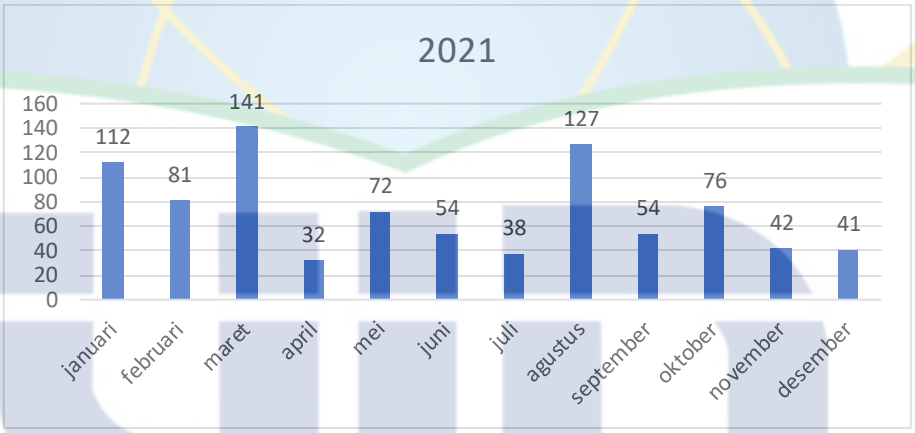
Tabel 4.1 Hasil *Web Scraping* Aplikasi Tangerang LIVE

Version	***Rating***	***User***	***Review***	***Time***
6.1.31	5	Widianta Arifin	Perlu diperbaiki sistemnya, yg lebih cepat dan tidak banyak persyaratan utk ngurus.	2022-02-27 11:58:52
6.1.31	4	Kosiem kosiem	Kok up date pembaharuan nya tidak bisa di instal???	2022-03-03 00:19:38
6.1.31	3	Antoni Hidayat	Kurang Singkat padat dan mudah dipahami	2021-12-29 6:50:01
6.1.29	2	Elly Samson	apikasi sia²	2021-10-30 00:16:50
6.1.31	1	Gila mancing sam	Kelamaan nunggu verifikasi knapa harus 1x24 jam Udah nunggu lama ditolak. Tolong dong saya butuh cepat buat persyaratan anak	2022-02-26 04:16:17

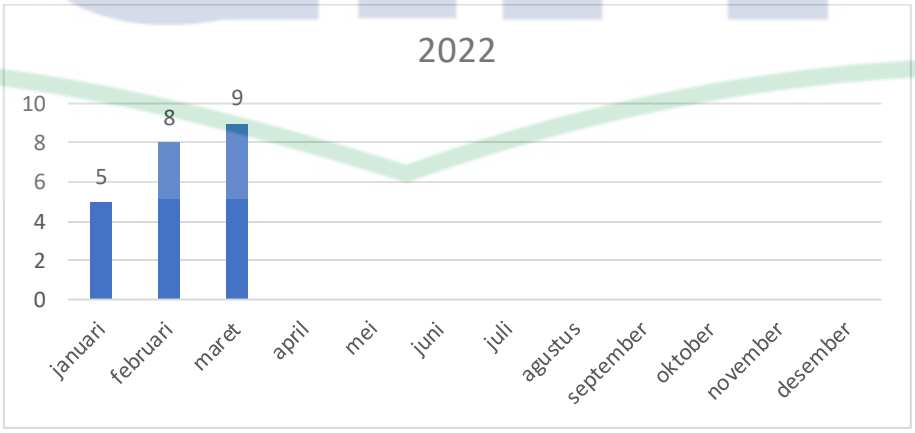
Dari *dataset* ulasan aplikasi Tangerang LIVE dibuat grafik untuk melihat arah *trend* ulasan masyarakat.



(a)



(b)



(c)

Gambar 4.1 Grafik Ulasan Berdasarkan Waktu (a) 2020, (b) 2021, (c) 2022

Gambar 4.1 menunjukkan *trend* jumlah ulasan dari bulan Juli 2020 sampai Maret 2022 terlihat menurun jumlah ulasan yang diberikan masyarakat tiap tahunnya. Pada bulan November 2020 mendapatkan ulasan dari *user* paling banyak dibandingkan dengan bulan yang lain, sebanyak 471 data ulasan.

Program python yang digunakan untuk proses *Web Scraping* dapat dilihat sebagai berikut:

```
# import packcage
!pip install google-play-scraper
from google_play_scraper import Sort, reviews_all
import pandas as pd

# scrapig data
data = reviews_all('id.go.tangerangkota.tangeranglive',
lang='id',
country='id',
sort=Sort.NEWEST,
filter_score_with=None)
df = pd.DataFrame(data)

# mengambil data dengan versi dan colom tertentu
df = df[(df.reviewCreatedVersion >= ('6.1.0')) &
     (df.reviewCreatedVersion <= ('6.1.31'))]
df = df [['userName', 'content', 'reviewCreatedVersion',
     'score','at']]
```

4.2 Hasil *Text Preprocessing*

Tahap *preprocessing* diperlukan agar *dataset* dapat diproses dengan metode yang digunakan pada penelitian ini. *Preprocessing* pada penelitian ini mulai dari *split sentence*, *cleansing*, *lowercase conversion*, *tokenization*, *normalize*, dan *stopword removal.*

4.2.1 *Split Sentence*

Diperlukannya tahap ini karena pada beberapa ulasan mengandung aspek atau sentimen yang berbeda di tiap kalimatnya, oleh karena itu data ulasan perlu dipisah tiap kalimatnya untuk mendapat aspek atau sentimen yang mungkin berbeda di tiap kalimatnya.

Memisahkan kalimat pada data ulasan berdasarkan simbol titik membuat *dataset* yang awalnya sebanyak 2419 menjadi 2676 yang selanjutnya masuk ke tahap *cleansing* data. Dari *dataset* penelitian ini, satu data ulasan dapat di-*split* paling sedikit 2 kalimat dan paling banyak menjadi 4 kalimat. Tabel 4.2 merupakan contoh *Split Sentence* salah satu ulasan.

Tabel 4.2 Hasil *Split Sentence*

Sebelum	Sesudah
Aplikasi sangat membantu. Cuma untuk pengaduannya sangat kurang. Ditingkatkan lgi untuk pelayanan pengaduan,,🙏🙏	• Aplikasi sangat membantu • Cuma untuk pengaduannya sangat kurang • Ditingkatkan lgi untuk pelayanan pengaduan,,🙏🙏

Program python yang digunakan untuk proses *Split Sentence* dapat dilihat sebagai berikut:

```
# import packcage
import pandas as pd

# import data
df = pd.read_excel("/content/drive/MyDrive/hasil ulasan.xlsx")
df = df.filter(['content'])
df = df.astype({'content' : 'string'})

#Split Kalimat
df = df.str.split('.', expand=True)
#Menggabungkan semua colom menjadi satu colom
df = pd.Series(df.values.ravel('F'))
#Mengganti cell kosong dengan NaN
df = df.copy()
df = df.replace(r'^s*$', float('NaN'), regex = True)
#Menghapus cell yang mengandung NaN
df = df.copy()
df.dropna(inplace = True)
#Merubah nama kolom
df = pd.DataFrame(df)
df = df.rename(columns={0:"ulasan"})
```

4.2.2 *Cleansing*

Pada tahap ini dilakukan pembersihan *dataset* dari *emoticons*, angka, *whitespace*, dan tanda baca. Tabel 4.3 berikut merupakan contoh *Cleansing* pada salah satu ulasan.

Tabel 4.3 Hasil *Cleansing*

Sebelum	Sesudah
• Aplikasi sangat membantu • Cuma untuk pengaduannya sangat kurang • Ditingkatkan lgi untuk pelayanan pengaduan,,🙏🙏	• Aplikasi sangat membantu • Cuma untuk pengaduannya sangat kurang • Ditingkatkan lgi untuk pelayanan pengaduan

Program python untuk proses *Cleansing* dapat dilihat sebagai berikut:

```
# import packcage
import re, unicodedata
# cleaning data
def cleaning(str):
    #remove non-ascii
    str = unicodedata.normalize('NFKD', str).encode('ascii',
          'ignore').decode('utf-8', 'ignore')
    #remove single char
    str = re.sub(r"\b[a-z]\b", "",str)
    #remove punctuations
    str = re.sub(r'[^\w]|_',' ',str)
    #remove digit or numbers
    str = re.sub(r"[0-9]", " ", str)
    #Remove additional white spaces
    str = re.sub('[\s]+', ' ', str)
    return str
df['data_clean'] = df.ulasan.apply(lambda x: cleaning(x))
```

4.2.3 *Lowercase Conversion*

Setelah tahap *Cleansing* data ulasan yang memiliki huruf kapital diubah menjadi huruf kecil, tujuannya agar tidak ada duplikat kata dikarenakan adanya perbedaan huruf kapital dan huruf kecil. Tabel 4.4 berikut merupakan contoh *Lowercase conversion* pada salah satu ulasan.

Tabel 4.4 Hasil *Lowercase Conversion*

Sebelum	Sesudah
• Aplikasi sangat membantu • Cuma untuk pengaduannya sangat kurang • Ditingkatkan lgi untuk pelayanan pengaduan	• aplikasi sangat membantu • cuma untuk pengaduannya sangat kurang • ditingkatkan lgi untuk pelayanan pengaduan

Program python untuk proses *Lowercase Conversion* dapat dilihat sebagai berikut:

```
# lowercase data
def lowcase(str):
    str = str.lower()
    return str
df['lowcase'] = df.['data_clean'].apply(lowcase)
```

4.2.4 *Tokenization*

Tahap *Tokenization* merupakan proses memecah suatu teks menjadi frasa atau kata yang bermakna yaitu *token*. Proses ini dilakukan karena pada proses *preprocessing* selanjutnya kalimat harus berbentuk *token*. Tabel 4.5 berikut merupakan contoh *Tokenization* pada salah satu ulasan.

Tabel 4.5 Hasil *Tokenization*

Sebelum	Sesudah
• aplikasi sangat membantu • cuma untuk pengaduannya sangat kurang • ditingkatkan lgi untuk pelayanan pengaduan	• [‘aplikasi’, ‘sangat’, ‘membantu’] • [‘cuma’, ‘untuk’, ‘pengaduannya’, ‘sangat’, ‘kurang’] • [‘ditingkatkan’, ‘lgi’, ‘untuk’, ‘pelayanan’, ‘pengaduan’]

Program python untuk proses *Tokenization* dapat dilihat sebagai berikut:

```
#import packcage
import nltk
from nltk.tokenize import word_tokenize, nltk.download('punkt')
```

```
# tokenisasi data
def word_tokenize_wrapper(text):
    return word_tokenize(text)
df['data_tokens'] = df['lowcase'].apply(word_tokenize_wrapper)
```

4.2.5 *Normalize*

Tahap *Normalize* diperlukan untuk mengubah kata *slang*, singkatan dan *typo* menjadi bentuk yang mempunyai persamaan makna agar memiliki keseragaman makna. Tabel 4.6 berikut merupakan contoh proses *Normalize*.

Tabel 4.6 Hasil *Normalize*

Sebelum	Sesudah
• [‘aplikasi’, ‘sangat’, ‘membantu’] • [‘cuma’, ‘untuk’, ‘pengaduannya’, ‘sangat’, ‘kurang’] • [‘ditingkatkan’, ‘lgi’, ‘untuk’, ‘pelayanan’, ‘pengaduan’]	• [‘aplikasi’, ‘amat’, ‘membantu’] • [‘cuma’, ‘untuk’, ‘pengaduan’, ‘amat’, ‘kurang’] • [‘tingkatkan’, ‘lagi’, ‘untuk’, ‘pelayanan’, ‘pengaduan’]
• [‘sy’, ‘seneng’, ‘dngn’, ‘aplikasi’, ‘ini’]	• [‘saya’, ‘senang’, ‘dengan’, ‘aplikasi’, ‘ini’]

Program python yang digunakan untuk proses *Normalize* dapat dilihat sebagai berikut:

```
# normalisasi data
normalized_word = pd.read_excel("/content/normalisasi s3.xlsx")
normalized_word_dict = {}
for index, row in normalized_word.iterrows():
    if row[0] not in normalized_word_dict:
        normalized_word_dict[row[0]] = row[1]
def normalized_term(document):
    return [normalized_word_dict[term] if term in
            normalized_word_dict else term for term in document]
df['data_normalisasi'] = df['data_tokens'].apply(normalized_term)
```

4.2.6 *Stopword Removal*

Tahap *Stopword Removal* dilakukan untuk menghapus kata umum yang digunakan dalam teks namun tidak memiliki pengaruh sentimen pada suatu kalimat.

Pada penelitian ini menggunakan *library* dari NLTK yang memiliki *corpus stopwords* Bahasa Indonesia. Tabel 4.7 berikut merupakan contoh proses *Stopword Removal.*

Tabel 4.7 Hasil *Stopword Removal*

Sebelum	Sesudah
• [‘aplikasi’, ‘amat, ‘membantu’] • [‘cuma’, ‘untuk’, ‘pengaduan’, ‘amat, ‘kurang’] • [‘tingkatkan’, ‘lagi’, ‘untuk’, ‘pelayanan’, ‘pengaduan’]	• [‘aplikasi’, ‘amat, ‘membantu’] • [‘cuma’, ‘pengaduan’, ‘amat, ‘kurang’] • [‘tingkatkan’, ‘pelayanan’, ‘pengaduan’]
• [‘kenapa’, ‘setiap’, ‘buka’, ‘harus’, ‘update’]	• [‘buka’, ‘harus’, ‘update’]

Program python untuk proses *Stopword Removal* dapat dilihat sebagai berikut:

```
# import packcage
import nltk
nltk.download('stopwords')

# stopword removal data
list_stopwords = stopwords.words('indonesian')
sw = set(list_stopwords)
def stopwords_removal(words):
    return [word for word in words if word not in sw]
df['data_stopwords'] = df['data_normalisasi'].apply
                       (stopwords_removal)
```

4.3 Metode Analisis Sentimen

Metode yang digunakan dalam analisis sentimen ini meliputi: *Latent Dirichlet Allocation* untuk pemodelan topik, *Lexicon Based* untuk melabeli sentimen pada *dataset* dan *Naïve Bayes Classifier* untuk klasifikasi sentimen.

4.3.1 Pemodelan Topik

Pada metode LDA dibutuhkan yang namanya *corpus* yang berisikan kata unik atau *token* dari *dataset.* Tabel 4.8 merupakan contoh isi dari *corpus.*

Tabel 4.8 Contoh Isi *Corpus*

‘kerja’, ‘bagus’, ‘informasi’, ‘cepat’, ‘respon’, ‘update’, ‘pembaruan’, ‘install’, ‘ide’, ‘puas’, ‘diperbaiki’, ‘sistem’, ‘persyaratan’, ‘mengurus’, ‘mudah’, ‘membantu’, ‘buka’, ‘bantuan’, ‘diganti’, ‘sembako’, ‘mengganti’, ‘alamat’,

Proses selanjutnya menghitung TF-IDF berdasarkan kamus atau *dictionary* yang dibuat berdasarkan *dataset* menggunakan *library* Gensim. Tabel 4.9 merupakan contoh hasil TF-IDF.

Tabel 4.9 Contoh Hasil TF-IDF

[[(0, 1), (1, 1)], [(0, 1)], [(0, 1)] , [(0, 1)] , [(2, 1)]]

Setiap kata pada *dataset* diubah menjadi angka dan tidak akan berulang atau *token*. Pada Tabel 4.9 angka sebelum tanda koma merupakan kata dan angka setelah tanda koma merupakan jumlah munculnya kata tersebut dalam sebuah ulasan.

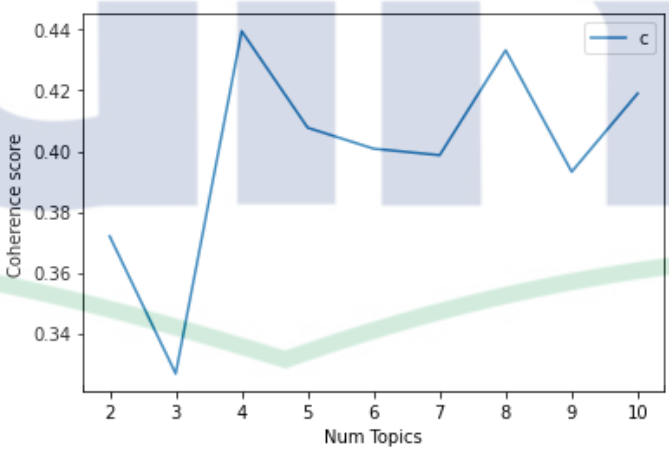
Program python untuk proses persiapan penentuan jumlah aspek sebagai berikut:

```
# import packcage
import pandas as pd
import gensim
import gensim.corpora as corpora
from gensim.utils import simple_preprocess

#import data
df_ori = pd.read_excel("/content/drive/MyDrive/prepro.xlsx")
df_ori = df_ori.filter(['ulasan'])
df = pd.read_excel("/content/drive/MyDrive/prepro.xlsx")
df = df.filter(['data_prepro'])

# Convert to list
data = df.data_prepro.values.tolist()
def sent_to_words(sentences):
    for sentence in sentences:yield(gensim.utils.simple_preprocess
                                    (str(sentence),deacc=True))
data_words = list(sent_to_words(data))

# Build the bigram and trigram models
bigram = gensim.models.Phrases(data_words, min_count=50,
         threshold=1) # higher threshold fewer phrases.
# Faster way to get a sentence clubbed as a trigram/bigram
bigram_mod = gensim.models.phrases.Phraser(bigram)
```

```
# Define functions bigrams
def make_bigrams(texts):
    return [bigram_mod[doc] for doc in texts]
# Form Bigrams
data_words_bigrams = make_bigrams(data_words)

# Create Dictionary
id2word = corpora.Dictionary(data_words_bigrams)
# Create Corpus
texts = data_words_bigrams
# Term Document Frequency
corpus = [id2word.doc2bow(text) for text in texts]
```

4.3.1.1 Penentuan Jumlah Aspek

Untuk mengetahui berapa banyaknya aspek terbaik dalam *dataset* yang digunakan pada penelitian ini maka perlu menghitung nilai koherensi. Hasil dari TF-IDF selanjutnya digunakan dalam proses uji *clustering* aspek, pada sumbu *X* merupakan *num topics* lalu untuk sumbu *Y* merupakan *coherence score*. *Num topics* yang memiliki *coherence score* paling tinggi merupakan *num topics* terbaik untuk dipilih. Gambar 4.2 menunjukkan bahwa *coherence score* tertinggi didapat oleh *num topics* 4 dengan *coherence score* 0,44, maka *num topics* terbaik adalah 4.



Gambar 4.2 Grafik Hasil Nilai Koherensi

Program python untuk proses penentuan jumlah aspek dapat dilihat sebagai berikut:

```
#import packcage
!pip install pyLDAvis
import gensim
from gensim.models import CoherenceModel
```

```
from gensim.models.ldamodel import LdaModel
# Plotting tools
import pyLDAvis
import pyLDAvis.gensim_models  # don't skip this
import matplotlib.pyplot as plt
%matplotlib inline

#function to compute coherence values
def compute_coherence_values(dictionary, corpus, texts, limit,
    start, step):
    coherence_values = []
    model_list = []
    for num_topics in range(start, limit, step):
        model = LdaModel(corpus=corpus, num_topics=num_topics,
                id2word=id2word)
        model_list.append(model)
        coherencemodel = CoherenceModel(model=model, texts=texts,
                        dictionary=id2word, coherence='c_v')
        coherence_values.append(coherencemodel.get_coherence())
    return model_list, coherence_values

# Show graph
limit=11; start=2; step=1;
model_list, coherence_values = compute_coherence_values
                               (dictionary=id2word, corpus=corpus,
                                texts=texts, start=start,
                                limit=limit, step=step)

x = range(start, limit, step)
plt.plot(x, coherence_values)
plt.xlabel("Num Topics")
plt.ylabel("Coherence score")
plt.legend(("coherence_values"), loc='best')
plt.show()

# Print the coherence scores
for m, cv in zip(x, coherence_values):
    print("Num Topics =", m, " has Coherence Value of", round(cv,
    3))
```

4.3.1.2 Penentuan Nama Aspek

Berdasarkan hasil penghitungan nilai koherensi didapatkan *num topics* 4 sebagai yang terbaik untuk dijadikan aspek, dan nama label aspek akan diinterpretasikan berdasarkan penelitian sebelumnya yang dilakukan oleh Alqaryouti *et al*., (2018) menyebutkan kategori aspek menurut standar tertulis oleh Android, Apple, dan Smart Dubai Office. Tabel 4.10 berikut menampilkan hasil dari *clustering topics*.

Tabel 4.10 Interpretasi Aspek

Topic	Kata Kunci	Interpretasi Aspek
1	0,0146 * “mudah” + 0,0099 * “berhenti” + 0,0075 * “susah” + 0,0059 * “kasih” + 0,0059 * “data” + 0,0059 * “ribet” + 0,0059 * “jelas” + 0,0059 * “bikin” + 0,0055 * “daftar” + 0,0055 * “download”	*User Interface*
2	0,2632 * “bagus” + 0,105 * “membantu” + 0,0265 * “kasih” + 0,0258 * “bantuan” + 0,0222 * “terima” + 0,0088 * “pemerintah” + 0,0085 * “kerja” + 0,0082 * “memudahkan” + 0,0062 * “betul” + 0,0054 * “semoga”	*User Experience*
3	0,0483 * “bantuan” + 0,0365 * “proses” + 0,0286 * “informasi” + 0,0265 * “daftar” + 0,0265 * “verifikasi” + 0,0255 * “bermanfaat” + 0,0252 * “membantu” + 0,0217 * “bagus” + 0,0211 * “respon” + 0,0211 * “kasih”	*Functionality and Performance*
4	0,0612 * “update” + 0,0326 * “semoga” + 0,0126 * “tolong” + 0,0097 * “membantu” + 0,0097 * “bagus” + 0,0088 * “buka” + 0,0085 * “kasih” + 0,0082 * “bantuan” + 0,0079 * “terima” + 0,0075 * “maju”	*Support and Updates*

Sehingga aspek yang didapat pada penelitian ini adalah, *User Interface, User Experience Functionality and Performance*, dan *Support and Updates*. Berikut contoh data ulasan terlabeli aspek pada Tabel 4.11.

Tabel 4.11 Contoh Data Ulasan Berlabel Aspek

Aspek	Kata	Interpretasi Aspek
1	Lumayan membantu dan aplikasi mudah digunakan	*User Interface*

2	Bagus, Sangat membantu	*User Experience*
3	Hm bantuan msh dalam proses mulu	*Functionality and Performance*
4	tapi terlalu sering untuk diupdate, tolonglah ya diperbaiki lagi	*Support and Updates*

Program python untuk proses penentuan nama aspek dapat dilihat sebagai berikut:

```
# import packcage
from pprint import pprint

# Select the model and print the topics
optimal_model = model_list[2]
model_topics = optimal_model.show_topics(formatted=False)
pprint(optimal_model.print_topics(num_words=10))

# melabeli data berdasarkan keywords
def format_topics_sentences(ldamodel=optimal_model, corpus=corpus,
    texts=data):
    # Init output
    sent_topics_df = pd.DataFrame()
    # Get main topic in each document
    for i, row in enumerate(ldamodel[corpus]):
        row = sorted(row, key=lambda x: (x[1]), reverse=True)
        # Get the Dominant topic, Perc Contribution and Keywords
          for each document
        for j, (topic_num, prop_topic) in enumerate(row):
            if j == 0:  # => dominant topic
               wp = ldamodel.show_topic(topic_num)
               topic_keywords = ", ".join([word for word, prop in
               wp])
               sent_topics_df = sent_topics_df.append(pd.Series([
                                int(topic_num), round(prop_topic,4),
                                topic_keywords]), ignore_index=True)
            else:
                break
    sent_topics_df.columns = ['Dominant_Topic', 'Perc_Contribution',
                              'Topic_Keywords']
    # Add original text to the end of the output
    contents = pd.Series(texts)
    sent_topics_df = pd.concat([sent_topics_df, contents], axis=1)
    return(sent_topics_df)
df_topic_sents_keywords = format_topics_sentences(ldamodel=
                          optimal_model, corpus=corpus, texts=data)
# Format
df_dominant_topic = df_topic_sents_keywords.reset_index()
df_dominant_topic.columns = ['Document_No', 'Dominant_Topic',
                             'Topic_Perc_Contrib','Keywords','Text']

# melihat sentimen dari text original
cek_df = pd.DataFrame([])
cek_df = df_dominant_topic.copy()
cek_df['Ulasan'] = df_ori['ulasan'].copy()
cek_df.head()
```

4.3.2 Pelabelan Sentimen

Digunakannya *Lexicon Based* pada penelitian ini untuk melabeli *dataset* apakah kalimat ulasan tersebut memiliki sentimen positif atau negatif. *Lexicon* yang dipakai berasal dari *library* Github evanmartua34 (2020) yang memodifikasi dan menambahkan kata dari kamus InSet (Indonesia *Sentiment Lexicon*) yang ada pada *library* Github fajri91 (2019). *Lexicon* memiliki banyak data 10.251 kata yang memiliki bobot tiap katanya dari +5 sampai -5. Gambar 4.3 Merupakan beberapa contoh kata yang terdapat dalam *Lexicon*.

	A	B	C
1	word	weight	number_of_words
2	hai	3	1
3	merekam	2	1
4	ekstensif	3	1
5	paripuma	1	1
6	detail	2	1
7	pemik	3	1
8	belas	2	1
9	welas	4	1
10	kabung	1	1

	A	B	C
4236	mbeling	-2	1
4237	rame	-2	1
4238	irl	-4	1
4239	ngeselin	-5	1
4240	ngatain	-5	1
4241	ganjaran	-5	1
4242	junjungan	-2	1
4243	imut-imut	-1	1
4244	soruh	-2	1
4245	lurus hati	-1	2
4246			

Gambar 4.3 *Lexicon*

text	sentiment	label
kerja bagus	4	positif
Good	2	positif
Ok	4	positif
Mantap banget	6	positif
Informasi tentang apa di kota Tangerang	1	positif
Bagus	2	positif
Tanglive sangat cepat merespon	10	positif
Kok update pembaruannya tidak bisa diinstall?	-7	negatif

Gambar 4.4 Contoh Beberapa Ulasan Hasil Sentimen dengan Metode *Lexicon*

Tabel 4.12 Hasil Sentimen dengan Metode *Lexicon*

Sentimen	Jumlah
Positif	1735
Negatif	915

Gambar 4.4 dan Tabel 4.12 merupakan hasil proses melabeli *dataset* ulasan *user* pada versi aplikasi Tangerang LIVE 6.1.0 sampai 6.1.31 menggunakan metode *Lexicon.* Untuk kalimat ulasan sentimen positif berjumlah 1735 data dan sentimen negatif berjumlah 915 data.

Program python untuk proses Pelabelan Sentimen dapat dilihat sebagai berikut:

```
# import packcage
!pip install Sastrawi
import pandas as pd
import numpy as np
import nltk
from nltk.tokenize import word_tokenize
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
nltk.download('punkt')

# Import Dataset
df_ori = pd.read_excel("/content/drive/MyDrive/prepro.xlsx")
df_ori = df_ori.filter(['ulasan'])
df_ori.tail()
df = pd.read_excel("/content/drive/MyDrive/prepro.xlsx")
df = df.filter(['data_prepro'])
df.tail()

# import lexicon dan menghapus kata negasi
negasi = ['bukan','tidak','kurang','belom','slow']
lexicon = pd.read_csv('/content/drive/modified_full_lexicon.csv')
lexicon = lexicon.reset_index(drop=True)
len(lexicon)
lexicon_word = lexicon['word'].to_list()
lexicon_num_words = lexicon['number_of_words']

# proses lexicon
sencol =[]
senrow = np.array([])
nsen = 0
factory = StemmerFactory()
stemmer = factory.create_stemmer()
sentiment_list = []
# function to write the word's sentiment if it is founded
def found_word(ind,words,word,sen,sencol,sentiment,add):
    # if it is already included in the bag of words matrix, then
      just increase the value
    if word in sencol:
        sen[sencol.index(word)] += 1
    else:
    # if not, than add new word
        sencol.append(word)
        sen.append(1)
        add += 1
    # if there is a negation word before it, the sentiment would
```

```
      be the negation of it's sentiment
    if (words[ind-1] in negasi):
        sentiment += -lexicon['weight'][lexicon_word.index(word)]
    else:
        sentiment += lexicon['weight'][lexicon_word.index(word)]
    return sen,sencol,sentiment,add
# checking every words, if they are appear in the lexicon, and
  then calculate their sentiment if they do
for i in range(len(df)):
    nsen = senrow.shape[0]
    words = word_tokenize(df['data_prepro'][i])
    sentiment = 0
    add = 0
    prev = [0 for ii in range(len(words))]
    n_words = len(words)
    if len(sencol)>0:
        sen =[0 for j in range(len(sencol))]
    else:
        sen =[]

    for word in words:
        ind = words.index(word)
        # check whether they are included in the lexicon
        if word in lexicon_word :
           sen,sencol,sentiment,add = found_word(ind,words,word,
                                      sen,sencol,sentiment,add)
        else:
        # if not, then check the root word
            kata_dasar = stemmer.stem(word)
            if kata_dasar in lexicon_word:
               sen,sencol,sentiment,add = found_word(ind,words,word,
                                          sen,sencol,sentiment,add)
        # if still negative, try to match the combination of words
          with the adjacent words
            elif(n_words>1):
                if ind-1>-1:
                    back_1 = words[ind-1]+' '+word
                    if (back_1 in lexicon_word):
                       sen,sencol,sentiment,add= found_word(ind,
                                words,back_1,sen,sencol,sentiment,add)
                    elif(ind-2>-1):
                        back_2    = words[ind-2]+' '+back_1
                        if back_2 in lexicon_word:
                            sen,sencol,sentiment,add= found_word
                        (ind,words, back_2,sen,sencol,sentiment,add)
    # if there is new word founded, then expand the matrix
    if add>0:
        if i>0:
            if (nsen==0):
                senrow = np.zeros([i,add],dtype=int)
            elif(i!=nsen):
                padding_h = np.zeros([nsen,add],dtype=int)
                senrow = np.hstack((senrow,padding_h))
                padding_v = np.zeros([(i-nsen),senrow.shape[1]],
                            dtype=int)
                senrow = np.vstack((senrow,padding_v))
            else:
```

```
                padding =np.zeros([nsen,add],dtype=int)
                senrow = np.hstack((senrow,padding))
            senrow = np.vstack((senrow,sen))
        if i==0:
            senrow = np.array(sen).reshape(1,len(sen))
    # if there isn't then just update the old matrix
    elif(nsen>0):
        senrow = np.vstack((senrow,sen))
    sentiment_list.append(sentiment)
len(sentiment_list)
print(senrow.shape[0])

# membangun data frame yang berisi bag of words dan sentimen yang
  dihitung sebelumnya
sencol.append('sentiment')
sentiment_array = np.array(sentiment_list).reshape(senrow.shape
                  [0],1)
sentiment_data = np.hstack((senrow,sentiment_array))
df_sen = pd.DataFrame(sentiment_data,columns = sencol)
df_sen

# melihat sentimen dari text original
cek_df = pd.DataFrame([])
cek_df['text'] = df_ori['ulasan'].copy()
cek_df['prepro'] = df['data_prepro'].copy()
cek_df['sentiment']  = df_sen['sentiment'].copy()

# create a list of our conditions
conditions = [(cek_df['sentiment'] < 0),(cek_df['sentiment'] > 0),]
values = ['negatif', 'positif']
cek_df['label'] = np.select(conditions, values)
```

4.3.3 Klasifikasi Sentimen

Dalam tahap ini *dataset* dibagi menjadi data latih/*train* dan data uji/*test* dengan beberapa skenario percobaan rasio yaitu 90% data *train* dan 10% data *test*, 80% data *train* dan 20% data *test*, dan 70% data *train* dan 30% data *test* berdasarkan penelitian sebelumnya oleh Gormantara (2020). Pada Tabel 4.13 merupakan hasil akurasi dari skenario rasio *dataset.*

Tabel 4.13 Hasil Akurasi Skenario Rasio *Dataset*

Data *Train* : Data *Test*	Akurasi
90% : 10%	84,53%
80% : 20%	85,84%
70% : 30%	86,04%

Berdasarkan Tabel 4.13 diambil rasio yang memiliki nilai akurasi paling tinggi yaitu 70% : 30% untuk digunakan dalam mengatasi masalah *Imbalance Ratio* (IR) *dataset* sebesar 1,896 dengan beberapa teknik yaitu *Undersampling*, *Oversampling*, dan SMOTE. Pada Tabel 4.14 berikut merupakan hasil kinerja dari setiap teknik dan sebelum di-*resampling*.

Tabel 4.14 Hasil Kinerja Algoritma

Sebelum *Resampling*			
Label	***Precision***	***Recall***	**F1-*Score***
Negatif	0,84	0,76	0,80
Positif	0,87	0,92	0,89
Akurasi	86,04%		

Undersampling			
Label	***Precision***	***Recall***	**F1-*Score***
Negatif	0,84	0,81	0,82
Positif	0,89	0,91	0,90
Akurasi	87,42%		

Oversampling			
Label	***Precision***	***Recall***	**F1-*Score***
Negatif	0,84	0,82	0,83
Positif	0,90	0,91	0,90
Akurasi	87,80%		

SMOTE			
Label	***Precision***	***Recall***	**F1-*Score***
Negatif	0,84	0,81	0,82
Positif	0,89	0,91	0,90
Akurasi	87,42%		

Berdasarkan Tabel 4.14, nilai akurasi paling tinggi didapat dengan teknik *oversampling*, dan untuk tabel *confusion matrix* dapat dilihat pada Tabel 4.15 berikut.

Tabel 4.15 *Confusion Matrix*

		Nilai Aktual	
		positif (1)	negatif (0)
Nilai Prediksi	positif (1)	462	52
	negatif (0)	45	236

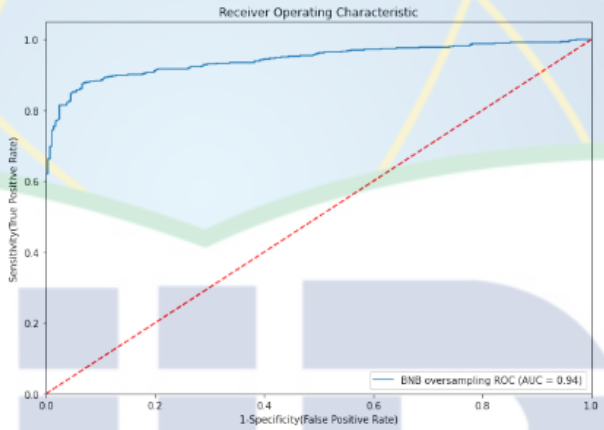
Program python untuk proses algoritma *Naïve Bayes* dapat dilihat sebagai berikut:

```
# import packcage
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn import metrics
from sklearn.metrics import roc_curve, auc
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score, precision_score,
     recall_score
from sklearn.naive_bayes import GaussianNB, MultinomialNB,
     BernoulliNB
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import
     TfidfVectorizer,CountVectorizer
from imblearn.combine import SMOTEENN
from imblearn.metrics import classification_report_imbalanced
from imblearn.over_sampling import RandomOverSampler
from imblearn.under_sampling import RandomUnderSampler

#import data
data = pd.read_excel('/content/drive/MyDrive/lexicon.xlsx')
cleanreview = data['prepro']
# tf-idf
Tfidf_vectorizer = TfidfVectorizer(max_df=0.75, min_df=5)
listdf=cleanreview.values.astype('U')
listdf = [d for d in listdf]
tfidf = Tfidf_vectorizer.fit_transform(listdf)
tfidf_term = Tfidf_vectorizer.get_feature_names()
print(tfidf.shape)
tfidf_vectorizer=TfidfVectorizer(use_idf=True)
tfidf_vectorizer_vectors=tfidf_vectorizer.fit_transform(listdf)
first_vector_tfidfvectorizer=tfidf_vectorizer_vectors[1]
df = pd.DataFrame(first_vector_tfidfvectorizer.T.todense(),
index=tfidf_vectorizer.get_feature_names(), columns=["tfidf"])
df.sort_values(by=["tfidf"],ascending=False)
X = tfidf, y = data['label']
```

```
#menghitung keseimbangan data sentimen print
("DATA SHAPE: ", data.shape)
data['label'].value_counts()
sns.countplot(data['label'])
plt.show()
pos = [i for i,x in enumerate(y) if x == '1' ]
neg = [i for i,x in enumerate(y) if x == '-1' ]
#split data training dan testing
X_train, X_test, y_train, y_test = train_test_split(X, y,
random_state=1, test_size=0.3)
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)

# hasil original
bnb = BernoulliNB()
bnb.fit(X_train, y_train)
y_bnb = bnb.predict(X_test); del bnb
print('Original Results:',classification_report(y_test, y_bnb))
print(confusion_matrix(y_test, y_bnb))
print('Akurasi original test = ', accuracy_score(y_test, y_bnb))
#undersampling
rm = RandomUnderSampler(random_state=1)
X_rm, y_rm = rm.fit_resample(X_train, y_train)
bnb = BernoulliNB()
bnb.fit(X_rm, y_rm)
y_bnb = bnb.predict(X_test); del bnb
print('UnderSampling
Results:\n',classification_report_imbalanced(y_test, y_bnb))
print(confusion_matrix(y_test, y_bnb))
print('Akurasi undersampling test= ', accuracy_score(y_test,
y_bnb))
#oversampling
ros = RandomOverSampler(random_state=1)
X_ros, y_ros = ros.fit_resample(X_train, y_train)
bnb = BernoulliNB()
bnb.fit(X_ros, y_ros)
y_bnb = bnb.predict(X_test); del bnb
print('OverSampling
Results:\n',classification_report_imbalanced(y_test, y_bnb))
print(confusion_matrix(y_test, y_bnb))
print('Akurasi oversampling test = ', accuracy_score(y_test,
y_bnb))
#both
smt = SMOTEENN (sampling_strategy='auto')
X_smt, y_smt = smt.fit_resample(X_train, y_train)
bnb = BernoulliNB()
bnb.fit(X_rm, y_rm)
y_bnb = bnb.predict(X_test); del bnb
print('Combination
Results:\n',classification_report_imbalanced(y_test, y_bnb))
print(confusion_matrix(y_test, y_bnb))
print('Akurasi combination = ', accuracy_score(y_test, y_bnb))
```

4.3.4 Evaluasi Kinerja Algoritma

Untuk mengevaluasi kinerja dari algoritma *Naïve Bayes* akan menggunakan Kurva ROC yang dibuat berdasarkan *confusion matrix* pada Tabel 4.15 dan menghitung luas area di bawah Kurva ROC atau disebut juga *Area Under Curve* (AUC). *Confusion matrix* memberikan rincian dari kesalahan klasifikasi, dan Kurva ROC menggambarkan hubungan antara *observed class* dan *predicted class* (Suwarno & Abdillah, 2016). Gambar 4.5 berikut merupakan Kurva ROC berdasarkan *confusion matrix* teknik *oversampling.*



Gambar 4.5 Kurva ROC dengan Nilai AUC

Berdasarkan Gambar 4.5, didapatkan nilai AUC sebesar 0,94. Maka dari itu menurut Tabel 2.3, karena nilai AUC mendapat 0,94, maka dinyatakan bahwa klasifikasi sangat baik.

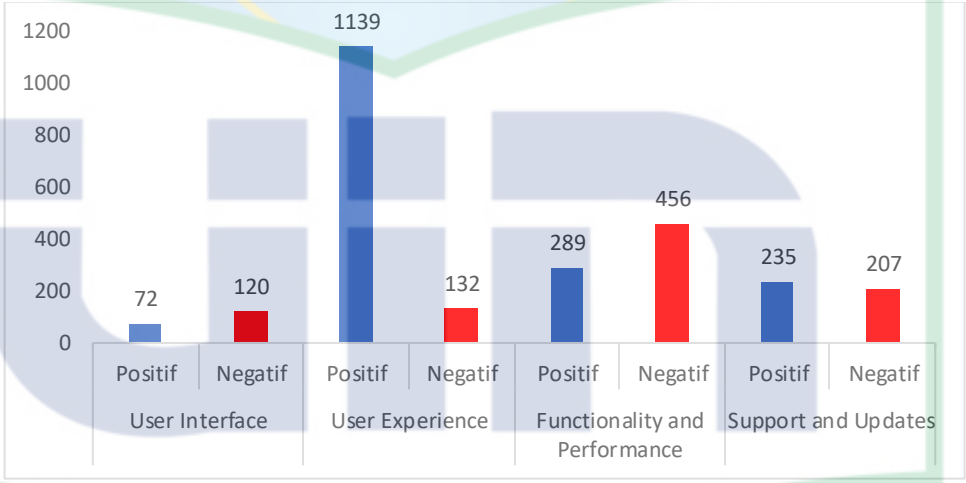
Program python untuk proses Evaluasi Kinerja Algoritma sebagai berikut:

```
# kurva ROC
bnb.fit(X_ros, y_ros)
probs_ros_bnb = bnb.predict_proba(X_test)
probs_ros_bnb = probs_ros_bnb[:, 1]
fpr, tpr, thresholds = roc_curve(y_test, probs_ros_bnb)
roc_auc = auc(fpr, tpr)
probsNB=[probs_ros_bnb]
models=["BNB oversampling"]
plt.figure(figsize=(10,7))
for idx,m in enumerate(models):
```

```
# Compute False postive rate, and True positive rate
  fpr, tpr, thresholds = metrics.roc_curve(y_test, probsNB[idx])
# Calculate Area under the curve to display on the plot
  roc_auc = auc(fpr, tpr)
# Now, plot the computed values
  plt.plot(fpr, tpr, label='%s ROC (AUC = %0.2f)' % (m, roc_auc))
# Custom settings for the plot
plt.plot([0, 1], [0, 1],'r--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('1-Specificity(False Positive Rate)')
plt.ylabel('Sensitivity(True Positive Rate)')
plt.title('Receiver Operating Characteristic')
plt.legend(loc="lower right")
plt.savefig(fname="auc sampling bernoulliNB predicting.jpg")
plt.show()  # Display
```

4.4 Interpretasi Hasil

Setelah *dataset* dilabeli aspek dan sentimen, hasilnya dapat dilihat pada Gambar 4.6 yang menunjukkan jumlah sentimen tiap kelas pada tiap aspek.

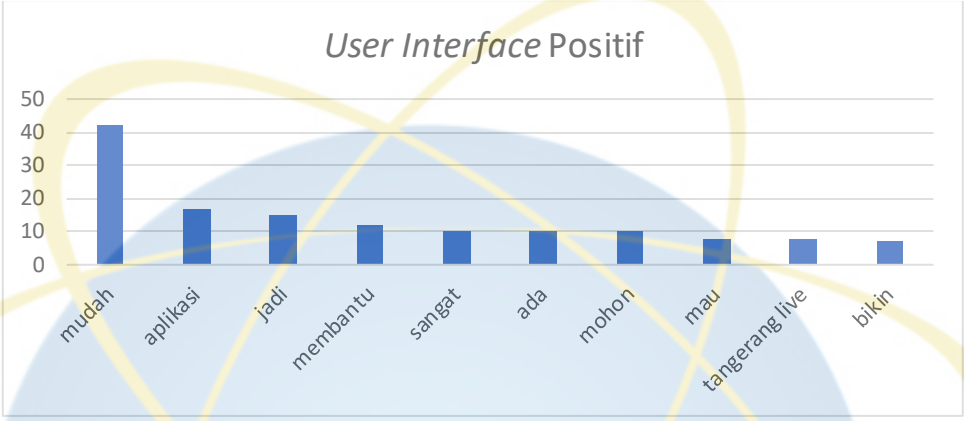


Gambar 4.6 Grafik Jumlah Sentimen Tiap Kelas pada Tiap Aspek

Berdasarkan Gambar 4.6, pada aspek *User Interface* sentimen pada kelas negatif lebih banyak dengan 120 ulasan, untuk aspek *User Experience* sentimen positif jauh lebih banyak dengan 1139 ulasan, untuk aspek *Functionality and Performance* sentimen negatif lebih banyak dengan 456 ulasan, dan untuk aspek *Support and Updates* sentimen positif sedikit lebih banyak dengan 235 ulasan.

4.4.1 Aspek *User Interface* Sentimen Positif

Pada Gambar 4.7 berikut menunjukkan kata yang paling sering muncul pada aspek *user interface* sentimen positif.



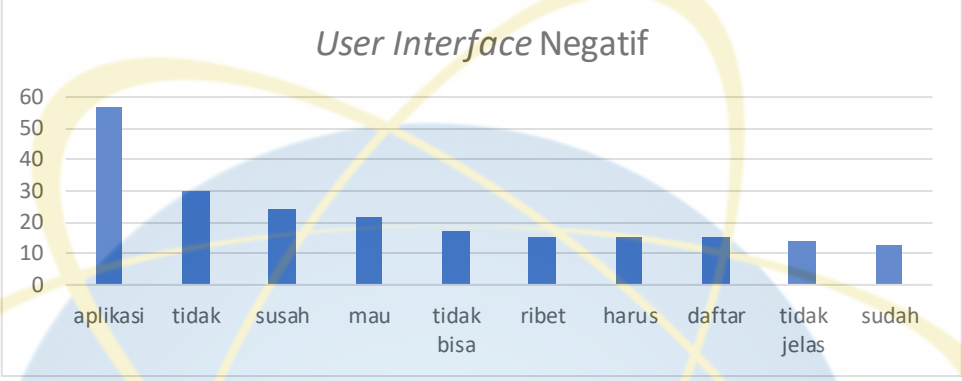
Gambar 4.7 Grafik Kata Paling Sering Muncul pada Aspek *User Interface* Positif

Berdasarkan Gambar 4.7, untuk aspek *user interface* bersentimen positif pada aplikasi Tangerang LIVE adalah mudahnya aplikasi Tangerang LIVE digunakan dan dipahami, seperti untuk mengurus administrasi pembuatan akta, dan layanan lengkap dalam satu aplikasi. Berikut ini Gambar 4.8 merupakan *wordcloud* dari kata yang paling sering muncul pada aspek *user interface* bersentimen positif.

Gambar 4.8 *Wordcloud* Aspek *User Interface* Sentimen Positif

4.4.2 Aspek *User Interface* Sentimen Negatif

Pada Gambar 4.9 berikut menunjukkan kata yang paling sering muncul pada aspek *user interface* sentimen negatif.



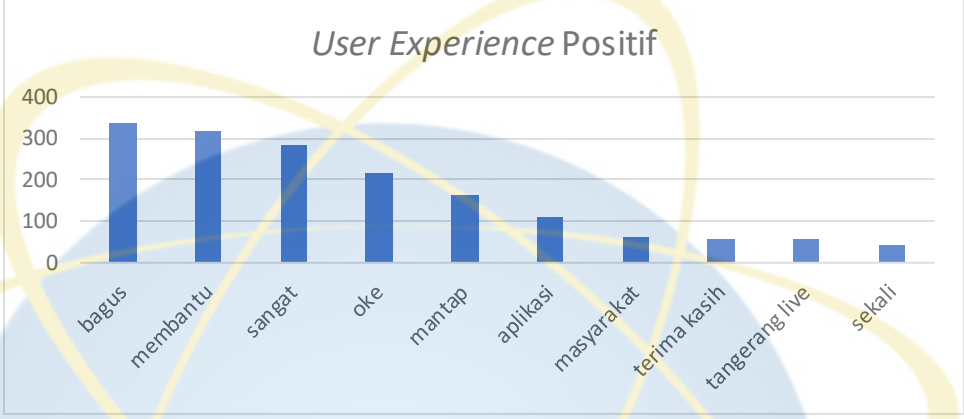
Gambar 4.9 Grafik Kata Paling Sering Muncul pada Aspek *User Interface* Negatif

Mengacu dari Gambar 4.9, untuk aspek *user interface* bersentimen negatif pada aplikasi Tangerang LIVE adalah layanan yang ada di dalam aplikasi tidak terintegrasi, sehingga *user* masih perlu keluar dari aplikasi Tangerang LIVE dan men-*download* aplikasi lain untuk layanan tertentu. Selanjutnya masalah *user* kesulitan dalam proses *login* dan masalah aplikasi berhenti sendiri (*force close*). Gambar 4.10 berikut adalah *wordcloud* dari kata yang paling sering muncul pada aspek *user interface* bersentimen negatif.

Gambar 4.10 *Wordcloud* Aspek *User Interface* Sentimen Negatif

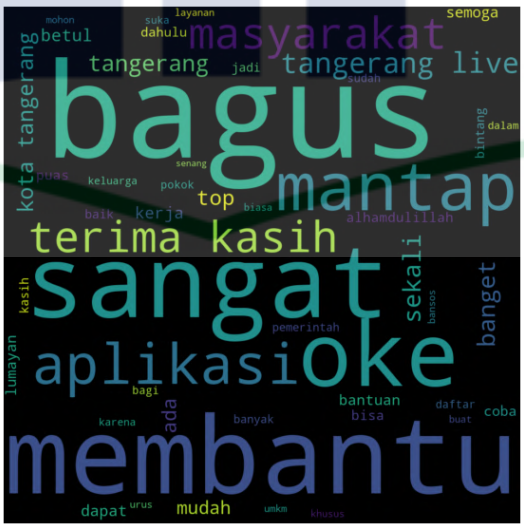
4.4.3 Aspek *User Experience* Sentimen Positif

Berikut ini Gambar 4.11 merupakan grafik yang menunjukkan kata paling sering muncul pada aspek *user experience* sentimen positif.



Gambar 4.11 Grafik Kata Paling Sering Muncul pada Aspek *User Experience* Positif

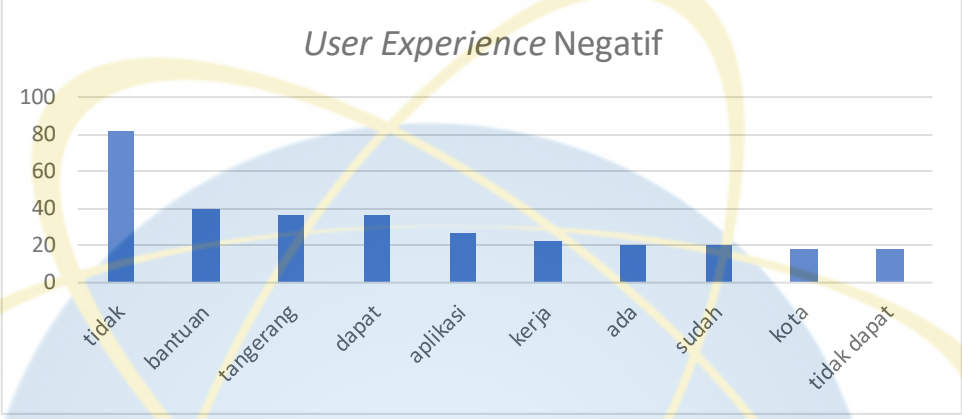
Bersumber dari Gambar 4.11, untuk aspek *user experience* bersentimen positif pada aplikasi Tangerang LIVE adalah aplikasinya bagus dan sangat membantu masyarakat, *user* merasa puas dengan layanan yang ada pada aplikasi Tangerang LIVE. Gambar 4.12 berikut adalah *wordcloud* dari kata yang paling sering muncul pada aspek *user experience* bersentimen positif.



Gambar 4.12 *Wordcloud* Aspek *User Experience* Sentimen Positif

4.4.4 Aspek *User Experience* Sentimen Negatif

Gambar 4.13 berikut merupakan grafik yang menunjukkan kata paling sering muncul pada aspek *user experience* sentimen negatif.



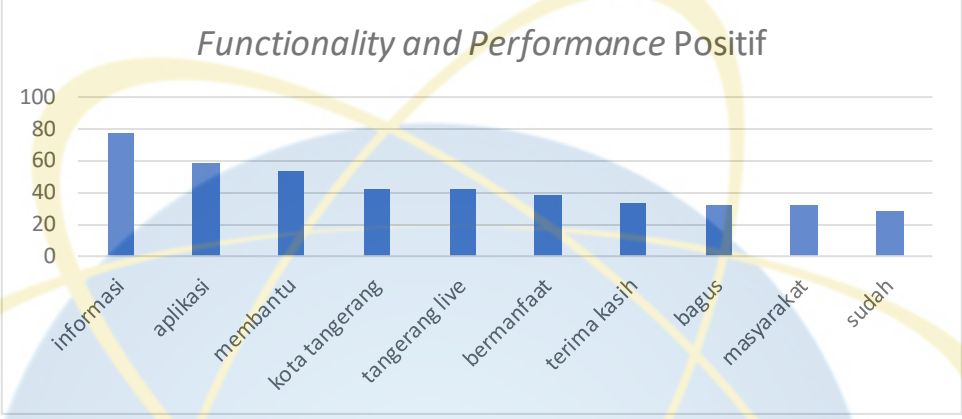
Gambar 4.13 Grafik Kata Paling Sering Muncul pada Aspek *User Experience* Negatif

Menurut Gambar 4.13, untuk aspek *user experience* bersentimen negatif pada aplikasi Tangerang LIVE adalah masyarakat tidak mendapatkan bantuan dari layanan bantuan yang disediakan pemerintah pada aplikasi Tangerang LIVE, dan kekecewaan masyarakat terhadap layanan yang diberikan aplikasi Tangerang LIVE belum maksimal. Pada Gambar 4.14 berikut merupakan *wordcloud* dari kata yang paling sering muncul pada aspek *user experience* bersentimen negatif.

Gambar 4.14 *Wordcloud* Aspek *User Experience* Sentimen Negatif

4.4.5 Aspek *Functionality and Performance* Sentimen Positif

Pada Gambar 4.15 berikut menunjukkan kata yang paling sering muncul pada aspek *functionality and performance* sentimen positif.



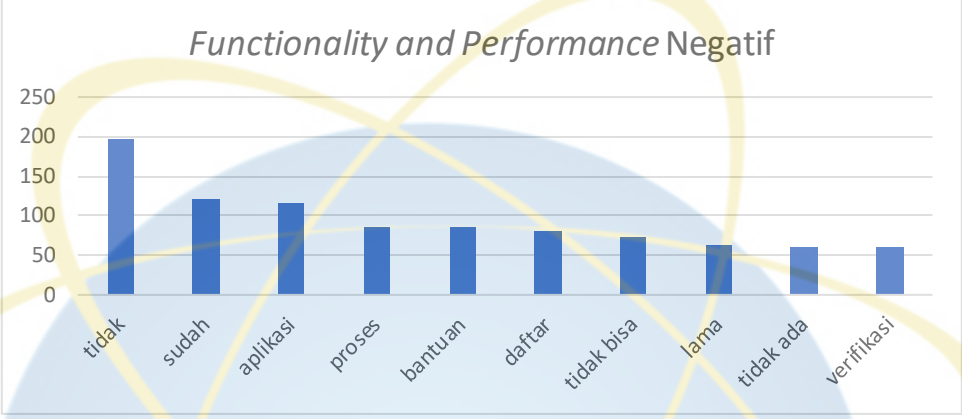
Gambar 4.15 Grafik Kata Paling Sering Muncul pada Aspek *Functionality and Performance* Positif

Berdasarkan Gambar 4.15, untuk aspek *functionality and performance* bersentimen positif pada aplikasi Tangerang LIVE adalah di dalam aplikasi Tangerang LIVE terdapat informasi yang bermanfaat dan layanan untuk membantu masyarakat. Berikut ini Gambar 4.16 merupakan *wordcloud* dari kata yang paling sering muncul pada aspek *functionality and performance* bersentimen positif.

Gambar 4.16 *Wordcloud* Aspek *Functionality and Performance* Sentimen Positif

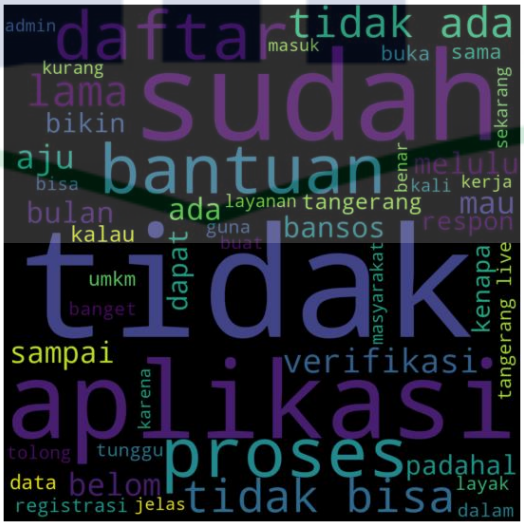
4.4.6 Aspek *Functionality and Performance* Sentimen Negatif

Pada Gambar 4.17 berikut menunjukkan kata yang paling sering muncul pada aspek *functionality and performance* sentimen negatif.



Gambar 4.17 Grafik Kata Paling Sering Muncul pada Aspek *Functionality and Performance* Negatif

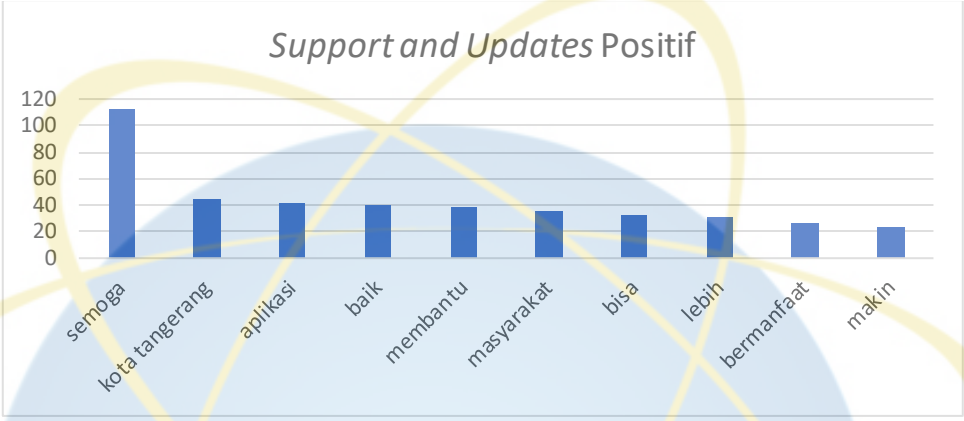
Mengacu dari Gambar 4.17, untuk aspek *functionality and performance* bersentimen negatif pada aplikasi Tangerang LIVE adalah proses layanan yang ada pada aplikasi Tangerang LIVE lama, seperti proses verifikasi dan pengajuan bantuan. Berikut ini Gambar 4.18 merupakan *wordcloud* dari kata yang paling sering muncul pada aspek *functionality and performance* bersentimen negatif.



Gambar 4.18 *Wordcloud* Aspek *Functionality and Performance* Sentimen Negatif

4.4.7 Aspek *Support and Updates* Sentimen Positif

Berikut ini Gambar 4.19 merupakan grafik yang menunjukkan kata paling sering muncul pada aspek *support and updates* sentimen positif.



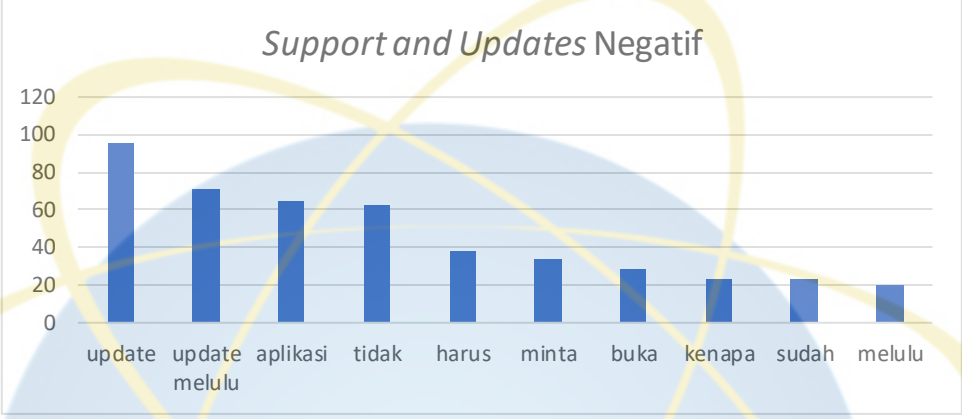
Gambar 4.19 Grafik Kata Paling Sering Muncul pada Aspek *Support and Updates* Positif

Bersumber dari Gambar 4.19, untuk aspek *support and updates* bersentimen positif pada aplikasi Tangerang LIVE adalah harapan masyarakat untuk aplikasi Tangerang LIVE semakin baik lagi layanannya agar lebih bermanfaat dan membantu masyarakat. Gambar 4.20 berikut adalah *wordcloud* dari kata yang paling sering muncul pada aspek *support and updates* bersentimen positif.

Gambar 4.20 *Wordcloud* Aspek *Support and Updates* Sentimen Positif

4.4.8 Aspek *Support and Updates* Sentimen Negatif

Gambar 4.21 berikut merupakan grafik yang menunjukkan kata paling sering muncul pada aspek *support and updates* sentimen negatif.



Gambar 4.21 Grafik Kata Paling Sering Muncul pada Aspek *Support and Updates* Negatif

Menurut Gambar 4.21 untuk aspek *support and updates* bersentimen negatif pada aplikasi Tangerang LIVE adalah aplikasi Tangerang LIVE terlalu sering *update* aplikasi dan juga masih terdapat *bug* pada *update* versi yang barunya. Pada Gambar 4.22 berikut merupakan *wordcloud* dari kata yang paling sering muncul pada aspek *support and updates* bersentimen negatif.

Gambar 4.22 *Wordcloud* Aspek *Support and Updates* Sentimen Negatif

4.5 Perbandingan dengan Penelitian Sebelumnya

Pada Tabel 4.16 berikut menampilkan hasil kinerja pada penelitian sebelumnya yang sejenis.

Tabel 4.16 Kinerja Algoritma pada Penelitian Sebelumnya

Peneliti	*Dataset*	Rasio	Kinerja Algoritma
Astuti (2020)	Ulasan aplikasi Tokopedia di aplikasi Play Store berjumlah 4.425 baris	70:30	92.5%
Parasati *et al* (2020)	Ulasan restoran Bakso President Malang di situs TripAdvisor dan Google Review berjumlah 2.152 baris	50:50	82,67%
Permana *et al* (2021)	Ulasan aplikasi *mobile baking* di aplikasi Play Store berjumlah 6194 baris	-	86,76%
Giffari (2022)	Ulasan aplikasi Tangerang LIVE di aplikasi Play Store berjumlah 2676 baris	70:30	87,80%

Berdasarkan Tabel 4.16 penelitian milik Astuti (2020), dengan *dataset* ulasan aplikasi Tokopedia sebanyak 4.425 baris, menggunakan rasio 70:30 data *train* dan data *test* menghasilkan nilai akurasi 92,5%. Lalu untuk pemodelan topik menggunakan metode LDA mendapat 4 topik terbaik dengan nilai koherensi 0,537. Penelitian lain oleh Parasati *et al* (2020), dengan *dataset* ulasan restoran Bakso President Malang sebanyak 2.152 baris, dengan rasio 50:50 menghasilkan nilai akurasi 82,67%. Lalu pada pemodelan topik secara manual mengambil 3 topik.

Penelitian selanjutnya oleh Permana *et al* (2021), dengan *dataset* ulasan aplikasi *mobile banking* sebayak 6.194 baris menghasilkan nilai akurasi 86,76%. Lalu untuk pemodelan topik dengan metode LDA mendapat 20 topik kelas positif

dengan nilai koherensi 0,48 dan 20 topik kelas negatif dengan nilai koherensi 0,36. Pada penelitian ini menggunakan *dataset* ulasan aplikasi Tangerang LIVE sebanyak 2.676 baris dengan rasio data *train* dan data *test* sebesar 70:30, kinerja algoritmanya mendapat nilai akurasi 87,80% serta pada evaluasi kinerja algoritma dengan Kurva ROC mendapat nilai AUC sebesar 0,94. Lalu untuk pemodelan topik menggunakan metode LDA mendapat 4 topik terbaik dengan nilai koherensinya 0,44. Namun *dataset* yang digunakan cukup sulit karena terdapat beberapa kata yang muncul sangat dominan dan membuat pendistribusian kata pada *clustetring* topik menjadi *similar*/serupa antara topik yang satu dengan yang lainnya.

BAB 5

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil pembahasan klasifikasi teks data ulasan aplikasi Tangerang LIVE menggunakan metode *Latent Dirichlet Allocation, Lexicon Based,* dan *Naïve Bayes* berikut ini merupakan kesimpulan yang diambil.

a. Aspek yang didapat dari penelitian ini dengan menggunakan metode LDA sebanyak 4 aspek, yaitu: *User Interface, User Experience, Functionality and Performance, Support and Updates*. Dan untuk klasifikasi sentimennya pada kelas positif 1735 data dan kelas negatif 915 data, dan untuk tiap aspeknya, pada *User Interface* sentimen pada kelas negatif lebih banyak dengan 120 ulasan, untuk aspek *User Experience* sentimen positif jauh lebih banyak dengan 1139 ulasan, untuk aspek *Functionality and Performance* sentimen negatif lebih banyak dengan 456 ulasan, dan untuk aspek *Support and Updates* sentimen positif sedikit lebih banyak dengan 235 ulasan.

b. Kelebihan Aplikasi Tangerang LIVE untuk aspek *User Interface* adalah mudahnya aplikasi digunakan dan dipahami, dan layanan lengkap dalam satu aplikasi. Untuk aspek *User Experience* adalah aplikasinya bagus dan sangat membantu masyarakat, *user* merasa puas dengan layanan yang ada pada aplikasi. Untuk aspek *Functionality and Performance* adalah di dalam aplikasi terdapat informasi yang bermanfaat dan layanan untuk membantu masyarakat. Untuk aspek *Support and Updates* adalah harapan masyarakat untuk aplikasi Tangerang LIVE semakin baik lagi layanannya agar lebih

bermanfaat dan membantu masyarakat. Selanjutnya Kekurangan aplikasi Tangerang LIVE untuk aspek *User Interface* adalah layanan yang ada di dalam aplikasi tidak terintegrasi, masalah *user* kesulitan dalam proses *login* dan masalah aplikasi berhenti sendiri (*force close*). Untuk aspek *User Experience* adalah masyarakat tidak mendapatkan bantuan dari layanan bantuan yang disediakan pemerintah pada aplikasi, dan kekecewaan masyarakat terhadap layanan yang diberikan aplikasi belum maksimal. Untuk aspek *Functionality and Performance* adalah proses layanan yang ada pada aplikasi lama, seperti proses verifikasi dan pengajuan bantuan. Untuk aspek *Support and Updates* adalah aplikasi terlalu sering *update* aplikasi dan juga masih terdapat *bug* pada *update* versi yang barunya.

c. Pada pemodelan topik menggunakan LDA mendapatkan 4 topik terbaik untuk dijadikan aspek dengan nilai koherensi 0,44 dan untuk pengklasifikasian sentimen menggunakan metode *Naïve Bayes* untuk kinerja algortima mendapat nilai 87,80% pada akurasi, 88% pada presisi, 88% pada *recall*, 85% pada *specificity* dan 88% pada F1-*Score*, serta pada evaluasi kinerja algoritma dengan Kurva ROC mendapat nilai AUC sebesar 0,94.

5.2 Kendala Penelitian

Kendala yang dialami penulis dalam penelitian ini adalah saat proses normalisasi data, karena pada *dataset* terdapat banyak kata-kata yang digunakan tidak baku, salah ketik, singkatan, dan kata *slang*/gaul, seperti kata ‘senang’ menjadi ‘seneng’, ‘sng’, ‘seneg’ dan lainnya, yang menyebabkan kata dengan makna yang sama tidak terhitung dan data tidak termuat pada *lexicon*.

Lalu pada proses pemodelan topik, dimana pada *dataset* yang digunakan terdapat beberapa kata yang dominan muncul seperti kata ‘bagus’, ‘membantu’ dan ‘bantuan’ yang membuat pendistribusian kata pada *clustering* topik menjadi *similar*/serupa antara topik yang satu dengan yang lainnya.

5.3 Saran

Berdasarkan hasil penelitian yang telah dilakukan, berikut ini adalah saran yang dapat dijadikan bahan pertimbangan untuk penelitian selanjutnya:

1. *Dataset* pada penelitian ini hanya menggunakan data ulasan untuk versi 6.1.0 sampai 6.1.31, oleh karena itu, untuk penelitian selanjutnya bisa mengambil ulasan untuk semua versi aplikasi.
2. Pemodelan topik dan juga pengklasifikasian sentimen bisa menggunakan metode yang lainnya, seperti *Latent Semantic Analysis* dan *Hierarchical Dirichlet Process* untuk pemodelan topik sedangkan untuk pengklasifikasian sentimen seperti *K-Nearest Neighbor* dan *Support Vector Machine*.
3. Pada penelitian ini *Lexicon* yang digunakan adalah InSet. untuk penelitian selanjutnya bisa dengan *Lexicon* lain yang lebih lengkap dan cocok untuk *dataset* yang digunakan seperti *Vader Sentiment.*

DAFTAR PUSTAKA

Adhitama, R., Kusumaningrum, R., & Gernowo, R. (2017). Pelabelan Topik Otomatis pada Artikel Berita Menggunakan *Latent Dirichlet Allocation* dan Ontologi (*Doctoral dissertation, School of Postgraduate*).

Agarina, M., Sutedi, S., & Karim, A. S. (2019). Evaluasi *User Interface* Desain Menggunakan Metode Heuristics Pada Website Sistem Informasi Manajemen Seminar Institut Bisnis dan Informatika (IBI) Darmajaya. In *Prosiding Seminar Nasional Darmajaya* (Vol. 1, pp. 192-200).

Aggarwal, C. C. (2015). *Data Mining: The Textbook*. Springer.

Ailiyya, S. (2020). Analisis Sentimen Berbasis Aspek pada Ulasan Aplikasi Tokopedia Menggunakan *Support Vector Machine* (*Bachelor's thesis*, Fakultas Sains dan Teknologi Universitas Islam Negeri Syarif Hidayatullah Jakarta).

Alamsyah, A., Rizkika, W., Nugroho, D. D. A., Renaldi, F., & Saadah, S. (2018). *Dynamic Large Scale Data on Twitter Using Sentiment Analysis and Topic Modeling*. *In 2018 6th International Conference on Information and Communication Technology (ICoICT)* (pp. 254-258). IEEE.

Alfanzar, A. I. (2019). *Topic Modelling* Skripsi menggunakan Metode *Latent Dirichlet Allocation* (Doctoral dissertation, UIN Sunan Ampel Surabaya).

Alghunaim, A. (2015). *A Vector Space Approach for Aspect-Based Sentiment Analysis* (*Doctoral Dissertation, Massachusetts Institute of Technology*).

Alqaryouti, O. H. A. (2017). *Aspect-Based Sentiment Analysis for Government Smart Applications Customers' Reviews* (*Doctoral dissertation*, The British University in Dubai (BUiD)).

Alqaryouti, O., Siyam, N., & Shaalan, K. (2018). *A Sentiment Analysis Lexical Resource and Dataset for Government Smart Apps Domain. In International Conference on Advanced Intelligent Systems and Informatics*, *845*, 230-240. Springer, Cham.

Alqaryouti, O., Siyam, N., Monem, A. A., & Shaalan, K. (2020). *Aspect-Based Sentiment Analysis Using Smart Government Review Data. Applied Computing and Informatics.*

Amelia, D. (2020). Kajian Kesiapan Kota Tangerang dalam Menerapkan Konsep *Smart City* (*Doctoral dissertation*, Institut Teknologi Indonesia).

Anggraini, M., Tyas, R. A., Saulasiyah. I. A., Aini, Q. (2020). Implementasi Algoritma *Naïve Bayes* dalam Penentuan *Rating* Buku. *SISTEMASI: Jurnal Sistem Informasi*, *9*(3), 557-566.

Areed, S., Alqaryouti, O., Siyam, B., & Shaalan, K. (2020). *Aspect-Based Sentiment Analysis for Arabic Government Reviews. In Recent Advances in NLP: The Case Of Arabic Language*, *874*, 143-162. Springer, Cham.

Arfianti, K. (2019). Identifikasi Topik Artikel Berita Menggunakan *Topic Modelling* dengan Metode *Latent Dirichlet Allocation*.

Arviana, G. N. (2021). *Sentiment Analysis*: Pengertian, Teknik, dan Penggunaannya. Diambil dari https://glints.com/id/lowongan/sentiment-analysis/.

Astuti, S. P. (2020). Analisis Sentimen Berbasis Aspek pada Aplikasi Tokopedia Menggunakan LDA dan *Naïve Bayes* (*Bachelor's Thesis*, Fakultas Sains dan Teknologi Universitas Islam Negeri Syarif Hidayatullah Jakarta).

Azhar, Y. (2018). Metode *Lexicon-Learning Based* untuk Identifikasi Tweet Opini Berbahasa Indonesia. *Jurnal Nasional Pendidikan Teknik Informatika: JANAPATI*, *6*(3), 237-243.

Bansal, H. (2020). *Latent Dirichlet Allocation*. Diambil dari https://medium.com/analytics-vidhya/latent-dirichelt-allocation-1ec8729589d4.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). *Latent Dirichlet Allocation*. *Journal of Machine Learning Research*, *3*, 993-1022.

Brownlee, J (2020). *Random Oversampling and Undersampling for Imbalanced Classification*. Diambil dari https://machinelearningmastery.com/random-oversampling-and-undersampling-for-imbalanced-classification/.

Budi, S. (2017). *Text Mining* untuk Analisis Sentimen *Review* Film Menggunakan Algoritma *K-Means*. *Techno.Com: Jurnal Teknologi Informasi*, *16*(1), 1-8.

Cendana, M., & Permana, S. D. H. (2019). Pra-Pemrosesan Teks pada Grup Whatsapp untuk Pemodelan Topik. *Jurnal Mantik Penusa*, *3*(3).

Fairuz, A. L., Ramadhani, R. D., & Tanjung, N. A. F. (2021). Analisis Sentimen Masyarakat Terhadap COVID-19 pada Media Sosial Twitter. *Indonesian Journal of Data Science, IoT, Machine Learning and Artificial Intelligence*, *1*(1), 41-150.

Fawcett, T. (2006). *An Introduction to ROC Analysis*. *Pattern Recognition Letters*, *27*(8), 861-874.

Febriansyah, R. F., Sukardi, F. R., & Aini, Q. (2019). Rekomendasi Moda Transportasi Mahasiswa dengan Algoritma *Naïve Bayes* (Studi Kasus: Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta). *Simetris: Jurnal Teknik Mesin, Elektro dan Ilmu Komputer*, *10*(2), 733-740.

Febrianta, M. Y., Widiyanesti, S., & Ramadhan, S. R. (2021). Analisis Ulasan *Indie Video Game* Lokal pada *Steam* Menggunakan *Sentiment Analysis* dengan Algoritma *Naive Bayes Classifier* dan LDA-*Based Topic Modeling*. *eProceedings of Management*, *8*(4), 3102-3109.

Fitriyyah, S. N. J., *et al.* (2019). Analisis Sentimen Calon Presiden Indonesia 2019 dari Media Sosial Twitter Menggunakan Metode *Naive Bayes*. *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, *5*(3), 279-285.

Ganesan, A. L., Schwietzke, S., Poulter, B., Arnold, T., Lan, X., Rigby, M., ... & Manning, M. R. (2019). Advancing scientific understanding of the global methane budget in support of the Paris Agreement. *Global Biogeochemical Cycles*, *33*(12), 1475-1512.

Gormantara, A. (2020). Analisis Sentimen Terhadap *New Normal* Era di Indonesia pada Twitter Analisis Sentimen Terhadap *New Normal* Era di Indonesia pada Twitter Menggunakan Metode *Support Vector Machine*.

Gunawan, B., Sastypratiwi, H., & Pratama, E. E. (2018). Sistem Analisis Sentimen pada Ulasan Produk Menggunakan Metode *Naive Bayes*. *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, *4*(2), 113-118.

Hamzah, A. (2012). Klasifikasi Teks Dengan *Naïve Bayes Classifier* (NBC) untuk Pengelompokan Teks Berita dan *Abstract* Akademis. *In Prosiding Seminar Nasional Aplikasi Sains & Teknologi (SNAST) Periode III ISSN* (p. 911X).

Hasibuan, A., & Sulaiman, O. K. (2019). *Smart City*, Konsep Kota Cerdas Sebagai Alternatif Penyelesaian Masalah Perkotaan Kabupaten/Kota, Di Kota-Kota Besar Provinsi Sumatera Utara. *Buletin Utama Teknik*, *14*(2), 127-135.

Josi, A., & Abdillah, L. A. (2014). Penerapan Teknik *Web Scraping* pada Mesin Pencari Artikel Ilmiah. *Jurnal Sistem Informasi (SISFO)*, *5*, 159-164.

Juliasari, N., & Sitompul, J. C. (2012). Aplikasi *Search Engine* dengan Metode *Depth First Search* (DFS). *Jurnal Mahasiswa TI SI BIT*, *9*(1), 9-12.

Kannan, S., Karuppusamy, S., Nedunchezhian, A., Venkateshan, P., Wang, P., Bojja, N., & Kejariwa, A. (2016). *Big Data analytics for social media*. *Big data: principles and paradigms, Cambridge-MA, Morgan Kaufmann-Elsevier*, *3*, 63-94.

Kementerian Agama, (2022). Qur'an Kemenag. Diambil dari https://quran.kemenag.go.id/sura/2/263.

Khalimi, A. M., (2020). Contoh *Dataset* dan Pengertian *Dataset*. Diambil dari https://www.pengalaman-edukasi.com/2020/11/apa-itu-dataset.html

Kulshrestha, R. (2019). *A beginner's guide to latent dirichlet allocation* (LDA). Diambil dari https://towardsdatascience.com/latent-dirichlet-allocation-lda-9d1cd064ffa2.

Kumar, K. (2018). *Evaluation of Topic Modeling: Topic Coherence*. Diambil dari https://datascienceplus.com/evaluation-of-topic-modeling-topic-coherence/

Liu, B. (2010). *Sentiment Analysis and Subjectivity*. *Handbook of natural language processing*, *2*, 627-666.

Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. *Synthesis lectures on human language technologies*, *5*(1), 1-167.

Miller, T., Dligach, D., & Savova, G. (2016). *Unsupervised document classification with informed topic models. In Proceedings of the 15th Workshop on Biomedical Natural Language Processing* (pp. 83-91).

MonkeyLearn Inc, (2022). *Sentiment Analysis Guide*. Diambil dari https://monkeylearn.com/sentiment-analysis/.

Muhamad, H., Prasojo, C. A., Sugianto, N. A., Surtiningsih, L., (2017). Optimasi *Naïve Bayes Classifier* dengan Menggunakan *Particle Swarm Optimization* pada Data Iris. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK), 4*(3), 180-184.Nuri, A. (2021). Implementasi Naïve Bayes dan Support Vector Machine dengan Lexicon Based untuk Analisis Sentimen pada Twitter.

Nuri, A. (2021). Implementasi *Naïve Bayes* dan *Support Vector Machine* dengan *Lexicon Based* untuk Analisis Sentimen pada Twitter.

Nurjaman, M., Mubarok, M., & Adiwijaya, A. (2017). Analisis Sentimen pada Ulasan Buku Berbahasa Inggris Menggunakan Information Gain dan Support Vector Machine. eProceedings of Engineering, 4(3), 4900-4906.

Parasati, W., Bachtiar, F. A., & Setiawan, N. Y. (2020). Analisis Sentimen Berbasis Aspek pada Ulasan Pelanggan Restoran Bakso President Malang dengan Metode *Naïve Bayes Classifier*. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, *4*(4), 1090-1099.

Pascual, F. (2019). Guide to Aspect-Based Sentiment Analysis. Diambil dari https://monkeylearn.com/blog/aspect-based-sentiment-analysis/.

Permana, M. E., Ramadhan, H., Budi, I., Santoso, A. B., & Putra, P. K. (2020). Sentiment Analysis and Topic Detection of Mobile Banking Application Review. In 2020 Fifth International Conference on Informatics and Computing (ICIC) (pp. 1-6). IEEE.

Potharaju, R., Rahman, M., & Carbunar, B. (2017). *A Longitudinal Study of Google Play*. *IEEE Transactions on computational social systems*, *4*(3), 135-149.

Pradana, G. (2021). *Web Scraping* Pengertian, Teknik, Manfaat, dan Kendala. Diambil dari https://ngalup.co/articles/pengertian-teknik-manfaat-kendala-web-scraping/.

Prajarini, D. (2018). Perancangan *Prototype Web Profile* Desa Wisata Dan Kerajinan Gamplong Sleman Dengan Metode Desain *User Experience*. *Aksa: Jurnal Desain Komunikasi Visual*, *2*(1), 249-259.

Putu, N. L. P. M., & Amrullah, A. Z. (2021). Analisis Sentimen dan Pemodelan Topik Pariwisata Lombok Menggunakan Algoritma *Naive Bayes* dan *Latent Dirichlet Allocation*. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *5*(1), 123-131.

Raschka, S. (2014). *Naive Bayes and Text Classification I - Introduction and Theory*. *arXiv preprint arXiv:1410.5329*.

Rezkia, S. M. (2021). Belajar Python Mengenal Pandas dan Series untuk Meningkatkan Kompetensi Data. Diambil dari https://www.dqlab.id/belajar-python-mengenal-pandas-dan-series-untuk-meningkatkan-kompetensi-data.

Ridwan, M., Suyono, H., & Sarosa, M. (2013). Penerapan *Data Mining* untuk Evaluasi Kinerja Akademik Mahasiswa Menggunakan Algoritma *Naive Bayes Classifier*. J*urnal EECCIS*, *7*(1), 59-64.

Routray, P., Swain, C. K., & Mishra, S. P. (2013). *A Survey on Sentiment Analysis*. *International Journal of Computer Applications*, *76*(10), 1-8.

Saldaña, Z. W. (2018). *Sentiment Analysis for Exploratory Data Analysis*. *Programming Historian*.

Schütze, H., Manning, C. D., & Raghavan, P. (2008). *Introduction to Information Retrieval. Cambridge: Cambridge University Press, 39*, 234-265.

Setijohatmo, U. T., Rachmat, S., Susilawati, T., & Rahman, Y. (2020). Analisis Metoda *Latent Dirichlet Allocation* untuk Klasifikasi Dokumen Laporan Tugas Akhir Berdasarkan Pemodelan Topik. *In Prosiding Industrial Research Workshop and National Seminar, 11*(1), 402-408.

Sidiq, M. (2019). Pengaruh Pra Proses pada Analisis Sentimen Dalam Teks Berbahasa Indonesia (*Doctoral dissertation*, Universitas Komputer Indonesia).

Somantri, O., & Khambali, M. (2017). *Feature Selection* Klasifikasi Kategori Cerita Pendek Menggunakan *Naïve Bayes* dan Algoritme Genetika. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*, *6*(3), 301-306.

Suwarno, S., & Abdillah, A. A. (2016). Penerapan Algoritma *Bayesian Regularization Backpropagation* untuk Memprediksi Penyakit Diabetes. *Indonesian Journal of Mathematics and Natural Sciences*, *39*(2), 150-158.

Tom (2021), *What Is Sentiment Analysis and How to Do It Yourself?*. Diambil dari https://brand24.com/blog/sentiment-analysis/

Turland, M. (2010). *Php Architect's Guide to Web Scraping with PHP*. *Marco Tabini & Associates, Incorporated*.

Uysal, A. K., & Gunal, S. (2014). *The Impact of Preprocessing on Text Classification*. *Information Processing & Management*, *50*(1), 104-112.

Wang, S., Lo, D., & Lawall, J. (2014). *Compositional Vector Space Models for Improved Bug Localization. In 2014 IEEE International Conference on Software Maintenance and Evolution* (pp. 171-180).

Wang, W., Feng, Y., & Dai, W. (2018). *Topic Analysis of Online Reviews for Two Competitive Products Using Latent Dirichlet Allocation*. *Electronic Commerce Research and Applications*, *29*, 142-156.

Widyanto, L. (2017). Deteksi Subjektifitas Teks Berbahasa Indonesia Menggunakan Metode *Lexicon-Rule Based* (*Doctoral Dissertation*, *University of* Muhammadiyah Malang).

Wirawan, I. N. T., & Eksistyanto, I. (2015). Penerapan *Naive Bayes* pada *Intrusion Detection System* dengan Diskritisasi Variabel. *Jurnal Ilmiah Teknologi Informasi*, *13*(2), 182-189.

Wisnu, G. R. G., Muttaqi, A. R., Santoso, A. B., Putra, P. K., & Budi, I. (2020). *Sentiment Analysis and Topic Modelling of 2018 Central Java Gubernatorial Election using Twitter Data. In 2020 International Workshop on Big Data and Information Security* (IWBIS) (pp. 35-40). IEEE.

Yutika, C. H., Adiwijaya, A., & Al Faraby, S. (2021). Analisis Sentimen Berbasis Aspek pada *Review Female Daily* Menggunakan TF-IDF dan *Naïve Bayes*. Jurnal Media Informatika Budidarma, *5*(2), 422-430.

Zaidi, N. A., *et al*. (2013). *Alleviating Naive Bayes Attribute Independence Assumption by Attribute Weighting*, *14*, 1947-1988.

LAMPIRAN

KEMENTERIAN AGAMA
UIN SYARIF HIDAYATULLAH JAKARTA
FAKULTAS SAINS DAN TEKNOLOGI

Jl. Ir. H. Juanda No. 95 Ciputat 15412 Indonesia
Telp. (62-21) 7493606, 7493547 Fax. (62-21) 7493315

Website : fst.uinjkt.ac.id
Email : fst@uinjkt.ac.id

Nomor : B - 1868E/F9/ KM.01 /04/2022 Jakarta, 22 April 2022
Lampiran : -
Perihal : Pembimbing Skripsi

Kepada Yth.
1. Dr. Qurrotul Aini M.T.
2. Ir. Eri Rustamaji MBA

Assalamualaikum, Wr Wb

Dengan ini diharapkan kesediaan Saudara untuk menjadi pembimbing I/II/ (Materi/Teknis)* penulisan skripsi mahasiswa:

Nama	: M RIZQI ARIEL GIFFARI
NIM	: 11170930000078
Program Studi	: Sistem Informasi
Judul Skripsi	: Analisis Sentimen Berbasis Aspek pada Ulasan Aplikasi Tangerang LIVE Menggunakan Latent Dirichlet Allocation dan Naive Bayes

Judul tersebut telah disetujui oleh Program Studi bersangkutan pada tanggal dengan outline, abstraksi dan daftar pustaka terlampir. Bimbingan skripsi ini diharapkan selesai dalam waktu 6 (enam) bulan setelah ditandatanganinya surat penunjukan pembimbing skripsi

Apabila terjadi perubahan terkait dengan skripsi tersebut selama proses pembimbingan, harap segera melaporkan kepada Program Studi bersangkutan.

Demikian atas kesediaan Saudara, kami ucapkan terima kasih.

Wassalamu'alaikum Wr.Wb

Jakarta, 22 April 2022
a.n Dekan
Wakil Dekan Bid. Akademik

Dr. Iim Rochaeni, M.Si.
NIP. 196203081989032001

Cek Keaslian Surat di https://e-letter.fst.uinjkt.ac.id/scan

KEMENTERIAN AGAMA
UIN SYARIF HIDAYATULLAH JAKARTA
FAKULTAS SAINS DAN TEKNOLOGI

Jl. Ir. H. Juanda No. 95 Ciputat 15412 Indonesia
Telp. (62-21) 7493606, 7493547 Fax. (62-21) 7493315

Website : fst.uinjkt.ac.id
Email : fst@uinjkt.ac.id

Nomor : B - 2338/F9 / KM. 01 /06/2022 Jakarta, 16 Juni 2022
Lampiran : -
Hal : Permohonan Riset

Kepada Yth.
Kepala Dinas Komunikasi dan Informatika Kota Tangerang
Dinas Komunikasi dan Informatika Kota Tangerang
Di
Tempat

Assalamualaikum, Wr Wb

Dengan hormat kami sampaikan bahwa:

Nama	: M RIZQI ARIEL GIFFARI
Tempat/Tanggal Lahir	: Tangerang / 28 April 1999
NIM	: 11170930000078
Semester	: 10
Program Studi	: Sistem Informasi
Alamat	: Jl Ir Sutami No 20
Telp/HP	: 085155433783

adalah benar yang bersangkutan mahasiswa Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta dan bermaksud melakukan penelitian/riset di instansi yang Bapak/Ibu pimpin, yang sedang dalam penyelesaian skripsi dengan judul skripsi:

"Analisis Sentimen Berbasis Aspek pada Ulasan Aplikasi Tangerang LIVE Menggunakan Latent Dirichlet Allocation dan Naive Bayes"

Untuk itu, kami mohon kesediaannya untuk memberikan kesempatan kepada mahasiswa tersebut dalam melaksanakan penelitian Data/riset di instansi/perusahaan yang Bapak/Ibu pimpin.

Demikian, atas perhatian dan kerjasamanya kami ucapkan terima kasih.

Wassalamu'alaikum, Wr Wb

Jakarta, 16 Juni 2022
a.n. Dekan
Wakil Dekan Bid. Akademik

Dr. Iin Ichaeni, M.Si.
NIP. 196203081989032001

Cek Keaslian Surat di https://e-letter.fst.uinjkt.ac.id/scan

SURAT PERYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : M Rizqi Ariel Giffari

NIK : 3603122804990003

Tempat, Tanggal Lahir : Tangerang, 28 April 1999

Jenis Kelamin : Laki - Laki

Alamat : Jl. Ir. Sutami No. 20 03/10, Sukasari, Tangerang

No. HP : 085155433783

Menyatakan bahwa telah mengambil data ulasan aplikasi Tangerang LIVE dari Google Play Store pada tanggal 8 April 2022 untuk keperluan skripsi saya yang berjudul "Analisis Sentimen Berbasis Aspek pada Ulasan Aplikasi Tangerang LIVE Menggunakan *Latent Dirichlet Allocation* dan *Naive Bayes*". Data yang diambil adalah ulasan dari versi 6.1.0 sampai versi 6.1.31 dengan rincian: *username*, ulasan, versi aplikasi yang diulas, *rating*, dan kapan ulasan tersebut diberikan.

Apabila saya menggunakan data yang diambil untuk hal komersil, merugikan pihak Pemerintah Kota Tangerang, atau hal lain yang melanggar norma dan ketentuan lainnya, maka saya bersedia bertanggung jawab sesuai dengan hukum yang berlaku. Demikian surat pernyataan ini saya buat dengan sebenar-benarnya.

Tangerang, 24 Juni 2022

Mengetahui,	Yang Membuat Pernyataan,
Dinas Komunikasi dan Informatika	
Kota Tangerang	
(FITSA DWI PUTRI)	(M Rizqi Ariel Giffari)