ANALISIS SENTIMEN BERBASIS ASPEK PADA APLIKASI TOKOPEDIA MENGGUNAKAN LDA DAN NAÏVE BAYES

SKRIPSI

Shinta Prima Astuti

11150940000029

PROGRAM STUDI MATEMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UIN SYARIF HIDAYATULLAH JAKARTA

2020 M / 1441 H

ANALISIS SENTIMEN BERBASIS ASPEK PADA APLIKASI TOKOPEDIA MENGGUNAKAN LDA DAN NAÏVE BAYES

Skripsi

Diajukan kepada
Universitas Islam Negeri Syarif Hidayatullah Jakarta
Fakultas Sains dan Teknologi
Untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Matematika (S.Mat)

Oleh:
Shinta Prima Astuti
11150940000029

PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UIN SYARIF HIDAYATULLAH JAKARTA
2020 M / 1441 H

PERNYATAAN

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

Jakarta, 10 Januari 2020

Shinta Prima Astuti

11150940000029

LEMBAR PENGESAHAN

Skripsi berjudul **"Analisis Sentimen Berbasis Aspek pada Aplikasi Tokopedia Menggunakan LDA dan Naïve Bayes"** yang ditulis oleh **Shinta Prima Astuti, NIM 11150940000029** telah diuji dan dinyatakan lulus dalam sidang Munaqosyah Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta pada hari Jumat, 10 Januari 2020. Skripsi ini telah diterima sebagai salah satu syarat untuk memperoleh gelar sarjana strata satu (S1) Program Studi Matematika.

Menyetujui,

Pembimbing I	Pembimbing II
Muhaza Liebenlito, M.Si	**Irma Fauziah, M.Sc**
NIDN. 2003098802	**NIP. 198007032011012005**
Penguji I	Penguji II
Dr. Taufik Edy Sutanto, MScTech	**Madona Wijaya, M.Sc**
NIP. 19790530 200604 1 002	**NIP. 198506242019032007**

Mengetahui,

Dekan Fakultas Sains dan Teknologi	Ketua program Studi matematika
Prof. Dr. Lily Surayya Eka P., M.Env. Stud	**Dr. Suma'inna, M.Si**
NIP. 196904042005012005	**NIP. 197912082007012015**

LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Yang bertanda tangan di bawah ini:

Nama : Shinta Prima Astuti

NIM : 11150940000029

Program Studi : Matematika Fakultas Sains dan Teknologi

Demi pengembangan ilmu pengetahuan, saya menyetujui untuk memberikan **Hak Bebas Royalti Non-Eksklusif** (*Non-Exclusive-Free Right*) kepada Program Studi Matematika Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta atas karya ilmiah saya yang berjudul:

"Analisis Sentimen Berbasis Aspek pada Aplikasi Tokopedia Menggunakan LDA Naïve Bayes"

beserta perangkat yang diperlukan (bila ada). Dengan Hak Bebas Royalti Non-Eksklusif ini, Program Studi Matematika Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta berhak menyimpan, mengalihmedia/formatkan, mengelolanya dalam bentuk pangkalan data (*database*), mendistribusikannya, dan menampilkan/mempublikasikannya di internet dan media lain untuk kepentingan akademis tanpa pelu meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta. Segala bentuk tuntutan hokum yang timbul atas pelanggaran Hak Cipta karya ilmiah ini menjadi tanggungjawab saya sebagai penulis.

Demikian pernyataan ini yang saya buat dengan sebenarnya.

Dibuat di Tangerang Selatan

Pada tanggal: 10 Januari 2020

Pernyataan

(Shinta Prima Astuti)

PERSEMBAHAN DAN MOTTO

Segala puji dan syukur saya persembahkan untuk Allah SWT serta shalawat kepada Nabi Muhammad SAW sehingga saya mampu melewati rintangan dalam menyelesaikan skripsi ini.

Persembahan juga saya berikan kepada kedua orang tua yang selalu mendukung dan mendoakan saya.

Skripsi ini juga dipersembahkan untuk teman-teman terdekat yang selalu memberikan semangat dan doanya untuk saya.

“Allah tidak akan membebani seseorang di luar batas kemampuannya.”

“Selalu bersyukur dan jangan cepat puas.”

ABSTRAK

Shinta Prima Astuti, Analisis Sentimen Berbasis Aspek pada Aplikasi Tokopedia Menggunakan LDA dan Naïve Bayes, di bawah bimbingan **Muhaza Liebenlito, M.Si** dan **Irma Fauziah, M.Sc.**

Tokopedia menempati posisi pertama sebagai aplikasi dengan pengguna aktif bulanan terbanyak di platform ponsel Android dan iPhone. Walaupun Tokopedia banyak diminati orang, kemungkinan ada hal yang tidak disukai oleh penggunanya. Karenanya peneliti mengusulkan penelitian *Aspect-Based Sentiment Analysis*, yaitu mengekstrak sentimen dan aspek dari aplikasi tersebut. Data yang digunakan adalah ulasan pengguna Tokopedia di Play Store dengan cara *scraping*. Pada penelitian ini, dilakukan klasifikasi sentimen menggunakan Naïve Bayes. Penentuan jumlah aspek dilakukan dengan *clustering* topik menggunakan LDA, dan menghasilkan 4 topik, yaitu kebermanfaatan, pelayanan, pengalaman belanja, dan tampilan. Hasil clustering menjadi acuan untuk proses anotasi aspek secara manual. Karena data sentimen negatif dan positif tidak seimbang, dilakukan *resampling data* menggunakan teknik *RandomUnderSampler*, *RandomOverSampler*, dan SMOTEENN. Evaluasi yang digunakan untuk klasifikasi sentimen adalah kurva ROC dan AUC. Akurasi tertinggi didapat setelah dilakukan oversampling yaitu sebesar 92,5%. Nilai AUC untuk *oversampling* sudah cukup baik, yaitu sebesar 0,95.

Kata Kunci: Oversampling, ROC, AUC, Aspect-Based Sentiment Analysis, Data Tidak Seimbang.

ABSTRACT

Shinta Prima Astuti, Aspect Based Sentiment Analysis of Tokopedia Application Using LDA and Naïve Bayes, under the guidance of **Muhaza Liebenlito, M.Sc** and **Irma Fauziah, M.Sc**.

Tokopedia takes the first position as the application with the most active monthly users on Android and iPhone mobile platforms. Although Tokopedia is much in demand, there are some things that are disliked by the users. Therefore the researcher proposes an Aspect-Based Sentiment Analysis study, which extracts sentiments and aspects of the application. The data used is Tokopedia user reviews on the Play Store by scraping. In this study, sentiment classification was conducted using Naïve Bayes. Number of aspects is done by clustering topics using LDA, and gets 4 topics, there are helpful, service, shopping experience, and display. The results of the clustering are used for manual aspect annotation process. Because negative and positive sentiment data are imbalance, resampling data is performed using the RandomUnderSampler, RandomOverSampler, and SMOTEENN techniques. The evaluations used for sentiment classification are the ROC and AUC curves. The highest accuracy is obtained after oversampling which is equal to 92.5%. AUC value for oversampling is good enough, which is equal to 0.95.

Keywords: Oversampling, ROC, AUC, Aspect-Based Sentiment Analysis, Imbalanced Data.

KATA PENGANTAR

Assalamu'alaikum Wr. Wb

Alhamdulillah puji syukur kehadirat Allah SWT yang telah memberikan rahmat dan hidayat-Nya, serta nikmat sehat dan panjang umur sehingga penulis mampu menyelesaikan penelitian ini. Shalawat serta salam saya panjatkan kepada Nabi Muhammad SAW dan para sahabat-sahabatnya. Penulis menyelesaikan penelitian ini untuk memperoleh gelar sarjana Matematika. Skripsi ini dapat terselesaikan karena bantuan dan dukungan dari berbagai pihak. Maka itu, penulis ingin menyampaikan rasa terima kasih kepada:

1. Prof. Dr. Lily Surayya Eka Putri, M.Env.Stud selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Syarif Hidayatullah Jakarta.
2. Ibu Dr. Suma'inna, M.Si, selaku Ketua Program Studi Matematika Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta dan Ibu Irma Fauziah M.Sc, selaku Sekretaris program studi Matematika Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta.
3. Bapak Muhaza Liebenlito, M.Si selaku pembimbing I dan Ibu Irma Fauziah, M.Sc selaku pembimbing II atas ilmu dan arahannya selama penyusunan skripsi ini hingga akhirnya bisa terselesaikan.
4. Bapak Dr. Taufik Edy Sutanto, M.Sc.Tech selaku penguji I dan Ibu Madona Wijaya, M.Sc selaku penguji II, terima kasih atas kritik dan sarannya kepada penulis, serta bersedia meluangkan waktunya untuk menguji seminar hasil dan sidang skripsi.
5. Kedua orang tua, adik, sepupu, dan kerabat keluarga lainnya yang tiada hentinya memberikan dukungan dalam bentuk apapun, baik itu semangat ataupun doa hingga penulis mampu menyelesaikan skripsi ini.
6. Teman – teman terdekat matematika yaitu Ery, Ayu, Intan, Fitria, Khusnul, Auli, Hamid, Aldo, dan Vika yang selalu memberikan semangat dan membantu banyak dalam berdiskusi selama pembuatan skripsi.
7. Teman seperjuangan skripsi Afifah, Rahil, dan Wina yang selalu menemani proses pembuatan skripsi, berdiskusi, dan memberikan saran terkait skripsi.

8. Teman – teman matematika angkatan 2015 yang tidak bisa disebutkan satu – persatu.
9. Seluruh pihak yang telah membantu dan mendukung penulis dalam penyelesaian skripsi ini.

Penulis menyadari bahwa masih ada kesalahan dalam penyusunan skripsi ini. Maka dari itu penulis mengharapkan kritik dan saran yang membangun supaya menjadi bahan perbaikan bagi peneliti selanjutnya. Penulis juga berharap penelitian ini bermanfaat bagi siapapun yang membacanya.

Wassalamu'alaikum Wr. Wb.

Ciputat, 10 Januari 2020

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I
PENDAHULUAN

1.1 Latar Belakang

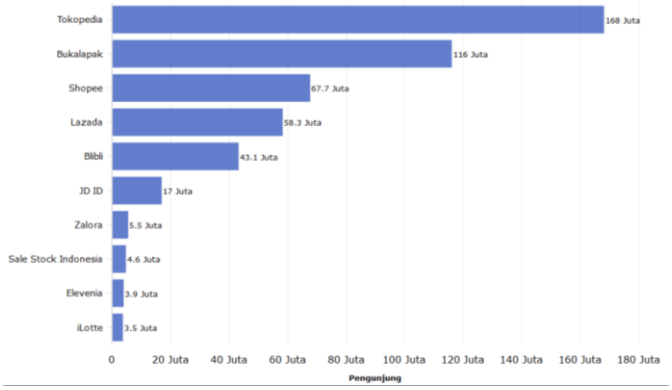
Dalam ayat Al-Qur'an disebutkan bahwa:

وَاَنْ لَّيْسَ لِلْاِنْسَانِ اِلَّا مَا سَعٰى

"dan bahwasannya seorang manusia tiada memperoleh selain apa yang telah diusahakannya." (QS An-Najm:39). Jika dilihat dari ayat tersebut, manusia memperoleh hasil dalam pekerjaan sesuai dengan apa yang telah diusahakannya. Hasil yang diperoleh akan sebanding dengan kerja keras yang dilakukannya. Kita juga perlu meneladani Rasulullah SAW untuk berusaha sesuai batas kemampuan diri. Setiap manusia tidak ada yang sempurna karena kesempurnaan hanya milik Allah SWT. Maka, semua yang dibuat oleh manusia pasti memiliki kekurangan dan kelebihan walaupun sudah mengerjakannya semaksimal mungkin.

Sebagai contoh adanya toko *online* yang sudah banyak tersedia di Indonesia. Majunya perkembangan teknologi memunculkan adanya toko online yang bisa diakses melalui *website* atau aplikasi, sehingga kita hanya perlu berbelanja melalui *gadget* tanpa harus keluar rumah. Selain efisien waktu dan tenaga, banyaknya promosi dan harganya yang lebih murah daripada belanja di toko biasa membuat masyarakat senang berbelanja *online*[1]. Namun, kelemahannya adalah barang juga mungkin tidak sesuai dengan yang diberikan pada foto dan tidak bisa memeriksa kondisi barang baik atau buruk.

Banyak perusahaan rintisan (*start up*) di bidang *e-commerce* yang bersaing membuat aplikasi toko online. Berikut adalah urutan e-commerce berdasarkan banyaknya pengunjung terbanyak selama triwulan IV 2018:



Gambar 1.1. *E-commerce* dengan Pengunjung Terbanyak di Indonesia per Bulan[2]

Gambar 1.1. E-commerce *dengan Pengunjung Terbanyak di Indonesia per Bulan* menunjukkan bahwa Tokopedia menjadi toko online dengan pengunjung terbanyak, yaitu lebih dari 100 juta pengunjung per bulan selama triwulan IV 2018, mengalahkan beberapa toko online terkenal lainnya. Tokopedia juga menempati posisi pertama sebagai aplikasi dengan pengguna aktif bulanan terbanyak di platform ponsel Android dan iPhone, disusul dengan Shopee, Bukalapak, dan Lazada pada kuartal 1 2019[3]. Namun, aplikasi toko online tersebut pasti memiliki keunggulan dan kelemahan, juga Tokopedia sekali pun. Pengalaman pengguna Tokopedia sebagai aplikasi dengan pengguna aktif terbanyak kerap kali dituangkan pada kolom komentar yang ada di *Play Store*, baik itu kritik atau penyampaian kepuasannya. Sentimen negatif atau positif pengguna terhadap suatu produk merujuk ke beberapa aspek produk. Misalnya, pengguna senang menggunakan aplikasi Tokopedia karena fiturnya yang mudah digunakan walaupun tampilannya kurang bagus.

Permasalahan seperti itu disebut *Aspect-Based Sentiment Analysis (ABSA).* Penelitian sebelumnya pernah dilakukan pada data ulasan film[4]. Teks ulasan tersebut tidak hanya menjelaskan sentimen keseluruhan film tapi juga menampilkan berbagai aspek seperti vokal, lirik, kualitas rekaman, kreatifitas, dan lain-lain. Contoh kalimatnya adalah "*aku suka ceritanya tapi tidak dengan musiknya*". Kalimat tersebut mengandung dua sentimen berlawanan dari dua aspek yaitu cerita dan musik. Penelitian tersebut melakukan klasifikasi sentimen dengan

menggolongkan setiap kata sebagai positif atau negatif, dengan mengambil skor rata – rata dari semua kata di SentiWordNet. Jika rata – rata lebih besar dari 0 maka kalimat dianggap bersentimen positif, sebaliknya negatif. Proses klasifikasi aspek menggunakan tag semantik yang dianotasikan dengan proses semantik. Penelitian lain juga dilakukan pada data blog berbahasa Cina[5], mereka mencari *global topic* dengan model LDA (*Latent Dirichlet Allocation*), kemudian mencari topik yang lebih spesifik yaitu *local topic* dengan metode *sliding window*. Proses sentimen dari local topic tersebut menggunakan model usulan yaitu BWM+DASM (*Benchmark Weighted Method + Direct Average Sum Method*). Penelitian ini menghasilkan akurasi klasifikasi sebesar 92.15%. Pada penelitian skripsi[6], ia melakukan tiga tahap untuk pekerjaan ABSA yaitu ekstraksi aspek, mengkategorikan aspek, dan klasifikasi sentimen. Hasil F1-Score dari ketiga tahap itu adalah 0.793, 0.823, dan 0.642. Ekstraksi aspek menggunakan algoritma CRF, sedangkan pengkategorian aspek dan klasifikasi sentimen menggunakan algoritma MaxEnt.

Pekerjaan yang akan dilakukan peneliti ada dua tahap. Pertama, melakukan clustering topik dengan LDA seperti pada [5] untuk penentuan jumlah aspek yang akan menjadi acuan untuk pelabelan manual aspek. Kedua, pekerjaan klasifikasi sentimen menggunakan Naïve Bayes. Namun, awalnya kalimat dipotong berdasarkan titik terlebih dahulu. Pemberian anotasi/label sentimen dan aspek dilakukan manual. Karena jumlah sentimen negatif dan positif tidak seimbang, maka dilakukan resampling data dengan teknik undersampling, oversampling, dan gabungan keduanya. Evaluasi model klasifikasi sentimen menggunakan kurva ROC dan daerah AUC. Lalu, hasil sentimen di tiap aspek direpresentasikan menggunakan *wordcloud*.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang di atas, rumusan masalahnya yaitu:

1. Aspek apa saja yang diulas pengguna aplikasi Tokopedia?
2. Apa saja yang menjadi kelebihan dan kelemahan dari tiap aspek pada aplikasi Tokopedia?

3. Bagaimana nilai evaluasi AUC pada klasifikasi sentimen pada ulasan Tokopedia?

1.3 Tujuan Penelitian

Tujuan dari penelitian yang dilakukan adalah sebagai berikut:

1. Mengetahui aspek apa saja yang diulas oleh pengguna aplikasi Tokopedia.
2. Mengetahui hal apa saja yang menjadi kelebihan dan kekurangan dari aplikasi Tokopedia.
3. Mengetahui nilai evaluasi AUC untuk melihat seberapa baik model klasifikasi sentimen.

1.4 Batasan Masalah

Batasan masalah yang ditentukan peneliti adalah sebagai berikut:

1. Hanya mengambil ulasan aplikasi Tokopedia pada sistem operasi Android yaitu pada Play Store.
2. Pilihan urutan ulasan pada Play store dipilih berdasarkan yang paling bermanfaat.
3. Bahasa ulasan yang didapat adalah Bahasa Inggris.
4. Pada ulasan yang panjang, tidak diambil keseluruhan kalimat ulasan tapi hanya diambil kalimat yang muncul pada halaman.
5. Memotong kalimat ulasan berdasarkan simbol pemisah titik.

1.5 Manfaat Penelitian

Dengan adanya penelitian ini diharapkan masyarakat mengetahui baik dan buruknya aplikasi tersebut berdasarkan pengalaman orang lain sehingga bisa lebih selektif sebelum mengunduh aplikasi, juga menjadi informasi untuk pihak Tokopedia tentang hal apa saja yang menjadi keluhan dan kepuasan penggunanya, yang bisa dijadikan evaluasi supaya bisa meningkatkan kualitas.

BAB II
TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1. Web Scraping

Web scraping merupakan proses pengambilan dokumen semi-terstruktur dari internet, biasanya terdiri dari halaman-halaman web dalam bahasa HTML atau XHTML, kemudian menganalisa dokumen tersebut untuk diambil data yang dibutuhkan sesuai kepentingan. *Web scraping* hanya fokus pada bagaimana memperoleh data dan mengekstrak data dalam ukuran yang bervariasi[7].

2.2. Data Tidak Seimbang

Data yang sudah diperoleh memiliki kemungkinan jumlahnya tidak seimbang antar kelasnya. Maka itu, bisa diatasi dengan melakukan *resampling.* Ada tiga cara untuk menyeimbangkan kembali data, dengan cara *undersampling* (membuang sampel dari kelas mayoritas) atau *oversampling* (menduplikasi beberapa sampel dari kelas minoritas), atau menggabungkan kedua cara tersebut[8].

2.3. Analisis Sentimen Berbasis Aspek (*Aspect-Based Sentiment Analysis*)

Analisis sentimen adalah analisis yang bertujuan untuk mengidentifikasi opini dan emosi yang diekspresikan oleh seseorang. ABSA melakukan analisis sentimen yang lebih dalam dari teks ulasan. Contohnya saat melihat ulasan dari sebuah lagu, opini yang dihasilkan tidak hanya keseluruhan sentimen tapi juga aspek spesifik seperti vokal, lirik, kualitas rekaman, dan lain-lain[4]. Tujuan dari ABSA adalah mengidentifikasi aspek dari entitas yang diberikan dan mengindentifikasi sentimen yang diekspresikan pada tiap aspek[9]. Pekerjaan ini dipecah menjadi sub tahapan yaitu pendeteksian topik dan analisis sentimen pada topiktersebut.

2.4. *Preprocessing*

Berikut tahapan yang peneliti lakukan pada preprocessing:

a. Mengubah kata Slang

Slang adalah kata-kata tidak baku yang sifatnya musiman, biasa digunakan oleh remaja atau kelompok sosial tertentu agar komunikasi bisa dimengerti oleh sekelompok itu saja[10]. Kata-kata yang ada di kolom komentar seringkali memuat kata slang ataupun singkatan, maka itu kita perlu mengganti kata slang dan singkatan itu ke kata yang mudah kita pahami. Sebelumnya diperlukan kamus kata berisi perubahan kata slang menjadi kata baku.

b. Case-folding

Case-folding adalah mengubah semua karakter menjadi sama, misalnya mengubah kata dari huruf besar menjadi huruf kecil[11]. Untuk data *string*, bisa menggunakan fungsi `string.lower()` untuk membuat semua huruf kecil atau fungsi `string.upper()` untuk membuat semua huruf besar, bisa juga dengan membuat definisi sendiri untuk case-folding.

c. Menghapus Stopword

Stopword adalah kata-kata yang tidak memiliki semantik dan tidak berhubungan dengan informasi yang relevan dengan kasus yang diteliti, biasanya kata depan dan subjek[11]. Perlu dibuat kamus khusus juga untuk kata-kata stopword yang bisa di-*update* menyesuaikan data yang akan diteliti. Contoh stopword bahasa inggris adalah "*the*", "*a*", "*when*", "*is*", dan lain-lain.

d. Menghapus simbol

Karena setiap kalimat komentar baik dalam web atau sosial media sering mengandung karakter selain huruf, maka itu harus dihapus karena untuk memudahkan melakukan analisa data, juga tidak penting untuk diolah.

2.5. Vector Space Model (VSM)

Data dibagi menjadi dua yaitu data terstruktur dan tidak terstruktur. Data terstruktur biasanya sudah dalam bentuk tabular (tabel, matriks, baris-kolom). Data seperti ini bisa langsung diolah dengan mudah. Sedangkan data tidak

terstruktur di antaranya seperti data teks, gambar, atau audio. Data seperti itu tidak bisa langsung diolah. Komputer tidak bisa mengolah data yang bukan angka. Maka, data seperti itu harus dikonversi ke angka.

VSM digunakan untuk mengubah dokumen menjadi nilai vektor. Setiap nilai dalam vektor merepresentasikan bobot dari kata dalam dokumen. Cara yang paling banyak digunakan adalah tfidf (*term frequency and inverse document frequency*). Tf-idf memberikan bobot pada kata *t* dalam dokumen *d* sesuai dengan rumus berikut:

$$weight(t,d) = tf(t,d) \times idf(t,D) \quad (2.1)$$

Dimana *t, d, D, tf (t,d), idf (t,D)* adalah kata, dokumen, corpus (kumpulan dokumen), *frequency t* di *d*, dan *inverse document frequency* dari *t* di *D*. Nilai tf-idf mencapai tertinggi saat suatu kata *t* muncul berkali-kali dalam jumlah dokumen yang sedikit. Nilai tf-idf akan lebih rendah jika suatu kata *t* muncul lebih sedikit dalam satu dokumen, atau dalam banyak dokumen. Nilai tf-idf terendah jika kata muncul hampir di semua dokumen[12].

Frekuensi kata atau tf (*term frequency*) menunjukkan seberapa banyak kata muncul dalam satu dokumen. Ini menunjukkan seberapa penting kata itu dalam suatu dokumen. Semakin tinggi bobot tf, artinya semakin besar kemunculan suatu kata dalam dokumen. Rumus tf diberikan seperti berikut:

$$tf = \begin{cases} 1 + log_{10}(f_{t,d}), & f_{t,d} > 0 \\ 0, & f_{t,d} = 0 \end{cases} \quad (2.2)$$

Invers frekuensi dokumen atau idf (*inverse document frequency*) menunjukkan kelangkaan dari suatu kata. Kata yang jarang digunakan akan lebih berguna untuk membedakan dokumen dengan yang lain. Idf dihitung sebagai kebalikan dari frekuensi dokumen atau *df* (*document frequency*)[13]. Rumus idf diberikan seperti berikut:

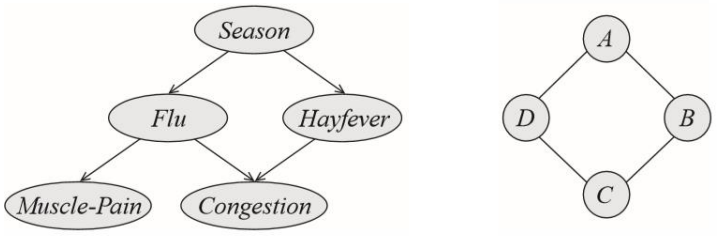
$$idf = log_{10}\left(\frac{N}{df_t}\right) \quad (2.3)$$

Dimana *N* adalah banyaknya dokumen, df_t menunjukkan banyaknya dokumen dalam corpus yang memuat kata *t*. Nilai idf tinggi artinya kemunculan dari kata

tersebut jarang, sedangkan nilai idf rendah menunjukkan seringnya kata tersebut muncul[12].

2.5. Probabilistic Graphical Model

Probabilistic graphical model merupakan mekanisme untuk menjelaskan distribusi yang kompleks ke bentuk yang lebih singkat dan padat. Graphical model ini menggunakan representasi berbasis graf. Node atau lingkaran menunjukkan variabel dari domain, dan sisi menunjukkan interaksi peluang antar node/variabel. Ada dua kelompok representasi grafis, yaitu Bayesian network dan markov network. Bayesian network menggunakan graf berarah (dimana sisi memiliki awalan dan target), sedangkan markov network menggunakan graf tidak berarah[14]. Sebagai contoh pada Gambar 2.1. *Representasi graf (a) Bayesian network dan (b) Markov network*, kita lihat tidak ada interaksi langsung antara *muscle pain* dengan *season*, tapi keduanya berinteraksi langsung dengan *Flu.*



(a) (b)

Gambar 2.1. Representasi graf (a) Bayesian network dan (b) Markov network

2.6. Teorema Bayes

Teorema Bayes mengukur probabilitas bersyarat dari variabel acak (variabel kelas), jika diberikan pengamatan yang diketahui tentang nilai variabel acak lainnya (variabel fitur). Teorema Bayes digunakan secara luas dalam probabilitas dan statistik. Bayes merupakan teknik prediksi berbasis probabilistik sederhana yang berdasar pada teorema bayes, dengan asumsi independensi (ketidaktergantungan) yang naif (kuat). Asumsi tersebut maksudnya adalah sebuah fitur pada data tidak

berkaitan dengan ada atau tidaknya fitur lain dalam data yang sama. Teorema bayes dirumuskan seperti berikut[15]:

$$P(D|E) = \frac{P(E|D) \times P(D)}{P(E)} \quad (2.4)$$

Keterangan:

$P(D|E)$: peluang bersyarat (*conditional probability*) dari hipotesis D terjadi jika diberikan bukti (*evidence*) E terjadi.

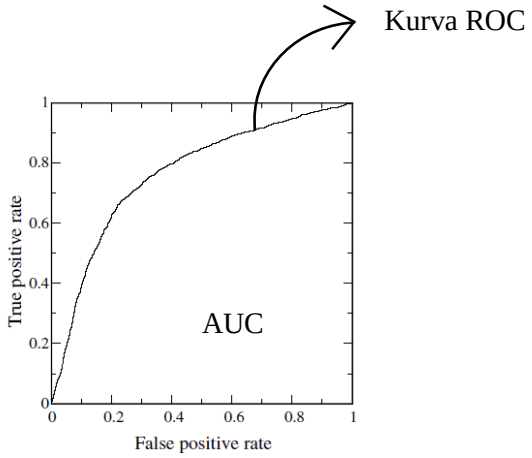
$P(E|D)$: peluang bukti E terjadi akan memengaruhi hipotesis D.

$P(D)$: peluang awal hipotesis/kejadian D terjadi tanpa memandang bukti apapun.

$P(E)$: peluang awal bukti E tanpa memandang hipotesis/bukti yang lain.

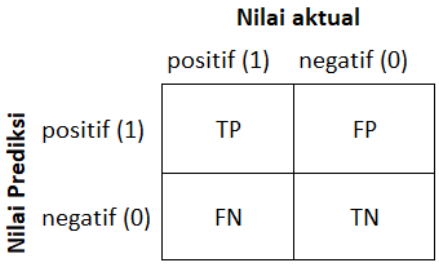
2.7. Kurva Receiver Operating Characteristic (ROC)

Kurva ROC adalah visualisasi grafik dari *precision* dan *recall*, yang didefinisikan dengan plot *False Positive Rate (FPR)* di sumbu x dan *True Positive Rate (TPR)* di sumbu y. TPR atau disebut *recall* adalah rasio prediksi benar positif dibandingkan keseluruhan data yang benar positif. FPR (1-specificity) adalah persentase kesalahan memprediksi positif dimana data aslinya adalah negatif. Contoh kurva ROC ditunjukkan seperti pada gambar berikut:



Gambar 2.2. Ilustrasi Gambar Kurva ROC dan daerah AUC

Salah satu keunggulan grafik ROC adalah grafik tersebut memungkinkan untuk memvisualisasi dan mengatur kinerja classifier tanpa memperhatikan distribusi kelas atau biaya kesalahan (*error cost*). TPR adalah seberapa banyak model memprediksi kelas positif saat klasifikasi aktual juga positif. FPR adalah seberapa banyak model memprediksi kelas positif secara salah, saat klasifikasi aktualnya negatif. Maka, model akan dianggap baik jika pada saat nilai TPR tinggi dan FPR rendah. Nilai TPR dan FPR didasarkan pada confusion matrix seperti berikut:



Gambar 2.3. Tabel Confusion Matrix

Pada Gambar 2.3. *Tabel Confusion Matrix* menunjukkan bahwa baris adalah nilai prediksi, dan kolom adalah nilai aktual dari positif dan negatif. Angka di diagonal utama merepresentasikan keputusan yang benar, dan angka di diagonal lainnya merepresentasikan eror antar berbagai kelas. Nilai aktual adalah klasifikasi yang telah dilakukan saat preprocessing. Nilai prediksi adalah klasifikasi yang dilakukan oleh program setelah memasukkan data ke model. Keterangan dalam tabel diberikan 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 diestimasi seperti rumus berikut[16]:

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

Nilai FPR (atau disebut juga *false alarm rate*) diestimasi seperti rumus berikut:

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

FPR dinyatakan sebagai $1 - specificity$, rumusnya diberikan seperti berikut:

$$specificity = \frac{TN}{FP+TN} \tag{2.7}$$
$$= 1 - FPR.$$

Model klasifikasi memprediksi data berdasarkan peluang dari tiap kelasnya. Kumpulan pasangan koordinat FPR dan TPR akan menghasilkan kurva ROC, berdasarkan *threshold* (batas) yang ditentukan. Nilai FP didapat dari menghitung peluang salahnya (negatif) yang lebih dari threshold, dan nilai TP didapat dari menghitung peluang benarnya (positif) yang lebih dari threshold. Kemudian kedua nilai masing – masing dibagi total jumlah data, lalu didapatkan pasangan koordinat (FPR, TPR). Setiap koordinat dihubungkan untuk membentuk kurva ROC. Range nilai FPR dan TPR adalah 0 sampai 1. Setiap nilai threshold menghasilkan titik yang berbeda pada ruang ROC.

Nilai klasifikasi ROC bisa ditentukan dengan cara menghitung luas area di bawah kurva ROC, atau disebut AUC (*Area Under Curve*). Ada beberapa nilai kriteria AUC seperti berikut[17]:

Tabel 2.1. Tabel Nilai AUC dan Interpretasinya

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

BAB III
METODE PENELITIAN

3.1. Sumber Data

Data yang digunakan adalah data sekunder yang berasal dari ulasan di Play Store. Proses pengambilan data menggunakan cara *scraping* yang menghasilkan 3077 data ulasan dari 17 Oktober 2018 sampai 10 Mei 2019 dari versi 3.1.1 sampai 3.30.

Data yang diperoleh berbahasa Inggris dan lokasi pengulas adalah Indonesia. Ulasan yang muncul terurut berdasarkan ulasan yang paling bermanfaat (*most helpful*). Parameter yang digunakan adalah x, yaitu batasan angka untuk banyaknya jumlah klik dan *scroll* saat membuka ulasan di halaman berikutnya. Semakin banyak nilai x yang digunakan, semakin banyak pula data ulasan yang akan diperoleh. Peneliti menggunakan nilai $x = 1000$, dan proses scraping selama kurang lebih 3 jam. Setelah proses scraping selesai, data disimpan dalam bentuk *CSV* (*Comma Separated Value*).

Peneliti menggunakan *package Selenium* untuk proses scraping dan browser Chrome. Ulasan yang didapat dari hasil scraping tidak urut waktu. Berikut adalah beberapa data awal dari hasil scraping:

Tabel 3.1. Hasil scraping Tokopedia

Date	*Rating*	*User*	*Review*
27-Apr-19	4	Adi Prasetio Raharjo	A great market place! I have never found issues with their seller partner. Almost all the stuff I bought arrive within 3 hours. need some improvements in the menu though, i would like to see my most frequently purchases in a glance for repeat orders. And wish to edit the categories to suit my needs....Full Review

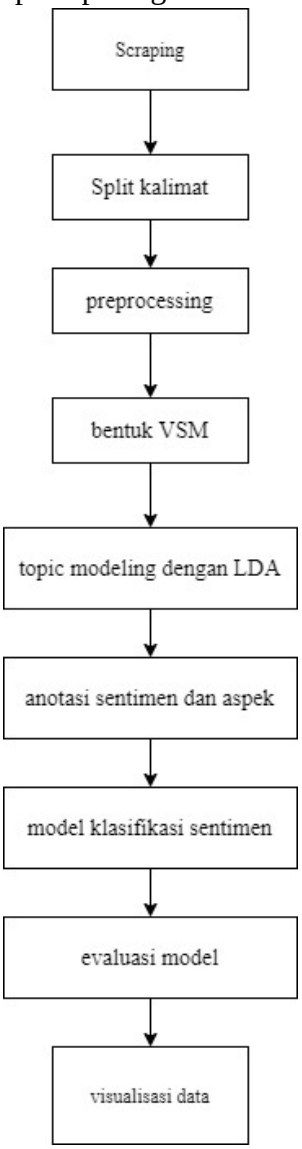
28-Apr-19	5	Dewi Rosalin	[UPDATE] The CS is more responsive and helpful in their Twitter account. Just contact them through their twitter and I got my cashback. Hopefully there are no more system failure that cause problem anymore
4-May-19	1	Johan Cahyadi	Your promotions are useless, everytime I ent. it, it's always 'Terjadi kesalahan, silahkan tutup bla bla'. even if I have met the requirement.. what a big let down. You should care for an update that give less burden for users, esp. with heavy animations (breaking eggs take longer than before which ...Full Review
1-May-19	1	Markus MR	Very bad experiance. Ordered and paid. After a few days we got a call from Tokopedia that there was a error in the system and they would like to help to correct the mistake. Asking us confirming the booking number which we did. Unfortunately only minutes after we realized that could be a possible sc...Full Review
22-Apr-19	1	Albert Sutanto	@ Pinda hape. YES.the CS are very lauzy not solving a problem only give an excuse. I have a same problems with you none of them been solved. any way just 4get the "KUPON MANTAP or DISCOUNT VAUCHER". It is very useless vaucher...@Shanty S'aid. yes you are

Dari hasil scraping, fitur yang diambil adalah date, rating, user, dan review. Berdasarkan Tabel 3.1. *Hasil scraping Tokopedia*, kolom date berisi tanggal ulasan itu dibuat. Pada saat proses scraping, tanggal tidak sesuai urutan waktu, maka kemudian diurutkan di *Microsoft Excel* berdasarkan waktu terlama hingga terbaru untuk bisa dilihat bagaimana perkembangan sentimen tiap waktu. Kolom rating berisi rata-rata nilai yang diberikan oleh pengulas terhadap aplikasi tersebut[18].

Kolom user berisi nama dari pengulas, dan kolom review berisi ulasan aplikasi tersebut.

3.2. Alur Penelitian

Alur penelitian diberikan seperti pada gambar berikut:



Gambar 3.1. Alur Penelitian

Penulis menggunakan *software WinPython* dengan bantuan beberapa modul yaitu *sklearn, nltk, pandas, bs4 (beautifulsoup), imblearn, seaborn, matplotlib, numpy,* dan *pyLDAvis.* Berdasarkan alur penelitian pada Gambar 3.1. *Alur Penelitian*, langkah awal yang dilakukan adalah mengambil data dengan cara scraping web Play Store. Setelah data didapat dan disimpan ke CSV, ulasan dipotong berdasarkan simbol pemisah yaitu titik, karena ada beberapa ulasan yang mengandung sentimen dan aspek berbeda di tiap ulasannya, sehingga tidak bisa diberi hanya satu label sentimen dan aspek saja dalam satu ulasan. Kemudian melakukan pembersihan data atau preprocessing. Langkah preprocessing yang dilakukan di antaranya mengubah kata slang/singkatan, case-folding, dan menghapus stopword dan simbol. Data harus dibersihkan terlebih dahulu karena data yang didapat merupakan data mentah yang belum siap olah, dan mengandung *noise*, artinya masih ada karakter selain huruf yang harus dihapus, dan banyak kata-kata singkatan yang perlu diubah. Kata yang sering muncul dan tidak penting juga perlu dihapus supaya mudah untuk mengolah data dengan hanya melihat kata yang penting saja. Selanjutnya, kata diubah ke angka supaya bisa diolah oleh komputer. Kata – kata diberi bobot sesuai seringnya kemunculan pada seluruh teks. pengubahan kata ke angka disebut VSM (*Vector Space Model*).

Setelah data bersih dan diberi bobot, kemudian melakukan *clustering topic* dengan LDA untuk menemukan berapa aspek yang harus digunakan untuk pemberian anotasi aspek. Pada sentimen hanya digunakan dua label yaitu negatif dan positif. Proses pelabelan sentimen dan aspek dilakukan manual. Kemudian, data dimasukkan ke model klasifikasi biner yaitu Naïve Bayes. Karena data klasifikasi sentimen tidak seimbang, yaitu lebih banyak data sentimen positif dibanding sentimen negatif, maka dilakukan resampling data untuk mengatasi data tidak seimbang, dengan cara undersampling, oversampling, atau gabungan keduanya. Kemudian, evaluasi klasifikasi sentimen dilakukan menggunakan kurva ROC (*Receiver Operating Characteristic*) dan AUC (*Area Under Curve*), kemudian dilihat nilai AUC tertinggi dari data sebelum dan setelah resampling. Lalu interpretasi data menggunakan wordcloud dari sentimen di setiap aspek.

3.3. Latent Dirichlet Allocation (LDA)

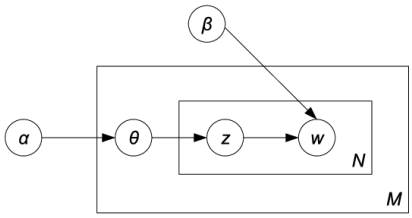
Ide dasar dari LDA adalah dokumen direpresentasikan sebagai campuran acak topik yang tersembunyi, dimana setiap topik dicirikan sebagai distribusi dari kata – kata[19]. Tujuan dari LDA adalah menentukan berapa jumlah topik dari suatu corpus dan sebaran kata di setiap topiknya. Secara formal, didefinisikan notasi berikut:

- Kata adalah bentuk dasar dari data diskrit
- Sebuah dokumen adalah barisan kata – kata *N* yang dinotasikan dengan **w** = (w_1, $w_{2, \ldots,}$ w_N), dimana w_N adalah barisan kata ke-*n*
- Sebuah *corpus* adalah koleksi dari *M* dokumen dinotasikan dengan *D* = ($\mathbf{w}_1$, $\mathbf{w}_2$, …, $\mathbf{w}_M$).

LDA mengasumsikan proses untuk setiap dokumen **w** di corpus *D*:

1. Pilih $N \sim$ Poisson (ξ)
2. Pilih $\theta \sim$ Dir (α)
3. Untuk setiap kata *N* yaitu w_n:
   a) Pilih topik $z_n \sim$ Multinomial (θ)
   b) Pilih kata w_n dari $P(w_n|z_n,\beta)$, peluang multinomial pada topik.

Proses kerja LDA diberikan seperti pada graphical model berikut[19]:



Gambar 3.2. Graphical Model LDA

Berdasarkan *graphical model* pada Gambar 3.2. *Graphical Model LDA*, parameter α dan β diberikan untuk corpus. Parameter α (*per-document topic distribution*) adalah parameter untuk distribusi topik dari dokumen, dan β (*per-topic word distribution*) adalah parameter untuk distribusi kata dari topik. Semakin besar nilai α, maka setiap dokumen mengandung sebagian besar topik, artinya tidak hanya ada satu topik spesifik. Sedangan semakin besar nilai β, maka setiap topik

mengandung sebagian besar kata, tidak hanya ada beberapa kata spesifik yang membedakan topik satu dengan lainnya. Untuk setiap dokumen *N*, terdapat distribusi topik yaitu θ. Karena LDA adalah *soft clustering*, maka setiap dokumen bisa terdiri dari beberapa topik yang berbeda. Kemudian, kita bisa menentukan topik dari setiap kata dalam setiap dokumen yang dinotasikan dengan *z*, untuk nantinya dikumpulkan menjadi cluster - cluster. Maka hasilnya adalah campuran kata – kata di tiap topik/cluster yang sudah ditentukan sebelumnya kemudian diinterpretasi hasil tiap cluster tersebut membahas topik apa.

Langkah pengerjaan LDA diberikan seperti contoh berikut[20]:

1. Menentukan nilai α, β, dan jumlah topik k.
2. Memecah dokumen menjadi bentuk kata per kata (tokenisasi), kemudian memberi nilai pada tiap kata. Contoh seperti pada tabel berikut:

Tabel 3.2. Dokumen dan Kata Unik untuk Contoh LDA

No.	Dokumen	Hasil token	Kata unik	Nilai tiap kata
1.	adik pergi ke bogor melihat kijang	“adik” “pergi” “ke” “bogor” “melihat” “kijang”	“adik” “pergi” “ke” “bogor” “melihat” “kijang”	1 2 3 4 5 6
2.	mobil kijang dibawa ayah pergi ke kantor	“mobil” “kijang” “dibawa” “ayah” “pergi” “ke” “kantor”	“mobil” “dibawa” “ayah” “kantor”	7 8 9 10

Dari dua dokumen yang diberikan seperti pada Tabel 3.2. *Dokumen dan Kata Unik untuk Contoh LDA*, setiap kalimat ditokenisasi menjadi bentuk kata per kata. Dari hasil token itu, diambil kata uniknya saja, dan kemudian memberi angka inisialisasi untuk setiap kata uniknya.

3. Kemudian menetapkan topik untuk setiap kata uniknya, pada contoh ini digunakan jumlah topik = 2, ditunjukkan pada tabel berikut:

Tabel 3.3. Hasil Sebaran Kata Unik di Tiap Topik

		Kata Unik									
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Topik	**1**	0	1	0	0	1	1	1	1	0	1
	2	1	1	2	1	0	1	0	0	1	0

Setelah menentukan jumlah topik, dibuat matriks topik dan kata unik seperti pada Tabel 3.3. *Hasil Sebaran Kata Unik di Tiap Topik*. Jumlah setiap kolom adalah total banyaknya kata unik muncul di seluruh dokumen. Misal pada kata unik 1 (kata "adik"), hanya ada 1 kata muncul di seluruh dokumen dan kata tersebut masuk ke topik 2, sedangkan pada kata unik 2 (kata "pergi") terdapat 2 kata muncul di seluruh dokumen, dan ada sejumlah 1 kata yang masuk topik 1 dan 2, dan seterusnya.

4. Menentukan topik untuk tiap kata dalam setiap dokumen.

Tabel 3.4. Topik Tiap Kata Unik dalam Masing – Masing Kalimat

Kalimat 1		**Kalimat 2**	
Kata Unik	**Topik**	**Kata Unik**	**Topik**
"adik"	2	"mobil"	1
"pergi"	1	"kijang"	2
"ke"	2	"dibawa"	1
"bogor"	2	"ayah"	2
"melihat"	1	"pergi"	2
"kijang"	1	"ke"	2
		"kantor"	1

Setelah menetapkan kata unik ke tiap topik, kemudian hasil tersebut dicocokkan ke tiap dokumen, untuk nantinya dihitung berapa jumlah kata yang masuk ke tiap topik.

5. Menghitung jumlah kata tiap topik pada masing – masing dokumen.

Tabel 3.5. Jumlah Kata Unik Tiap Topik pada Masing – Masing Dokumen

Dokumen	Topik	
	1	**2**
1	3	3
2	3	4

Nilai pada Tabel 3.5. *Jumlah Kata Unik Tiap Topik pada Masing – Masing Dokumen* adalah jumlah kata unik yang masuk ke topik 1 dan 2 di tiap dokumen. Pada dokumen 1, terdapat 3 kata yang masuk topik 1 dan topik 2. Pada dokumen 2, terdapat 3 kata yang masuk topik 1 dan 4 kata yang masuk topik 2.

6. Kemudian menghitung peluang topik di masing – masing dokumen, dengan persamaan berikut:

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} {\theta_k}^{\alpha_k - 1} \qquad (3.1)$$

Hasilnya seperti pada matriks berikut:

Tabel 3.6. Peluang Topik Tiap Dokumen

Dokumen	Topik	
	1	**2**
1	0.5000000	0.5000000
2	0.1111111	0.8888889

Pada Tabel 3.6. *Peluang Topik Tiap Dokumen*, peluang kemunculan topik 1 dan 2 pada dokumen 1 sebesar 0.5, sedangkan peluang kemunculan topik 1 pada dokumen 2 sebesar 0.11 dan topik 2 sebesar 0.89. Sehingga, proporsi topik pada dokumen 1 adalah seimbang, dan pada dokumen 2 lebih banyak mengandung topik 2 dibanding topik 1.

7. Kemudian peluang distribusi kata di tiap topik diberikan seperti berikut:

Tabel 3.7. Peluang Kata Tiap Topik

Topik	Kata Unik									
	adik	pergi	ke	bogor	melihat	kijang	mobil	dibawa	ayah	kantor
1	0.33256	0.00033	0.00033	0.33256	0.33256	0.00033	0.00033	0.00033	0.00033	0.00033
2	0.00010	0.19990	0.19990	0.00010	0.00010	0.19990	0.10000	0.10000	0.100000	0.100000

Nilai – nilai pada Tabel 3.7. *Peluang Kata Tiap Topik* merupakan peluang kemunculan kata pada tiap topik, berdasarkan pada tabel di langkah 3. Setiap kata mungkin bisa masuk ke beberapa topik, hanya proporsinya saja yang berbeda. Contoh pada kata adik, peluang munculnya pada topik 1 lebih besar daripada di topik 2.

Hasil dari model tiap dokumen memunculkan kata – kata dari cluster topik yang telah ditentukan. Kumpulan kata di tiap cluster topik selanjutnya diinterpretasi untuk mendapatkan informasi topik apa yang dibicarakan di tiap cluster. Maka, total peluang dari model LDA berdasarkan graphical model bisa ditulis dalam rumus berikut:

$$p(\theta, w, z|\alpha, \beta) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta) p(w|z_n, \beta) \quad (3.2)$$

3.4. Naïve Bayes

Model klasifikasi yang digunakan adalah Naïve Bayes. Klasifikasi Bayes didasarkan pada teorema Bayes untuk peluang bersyarat. Naïve Bayes adalah model peluang yang paling sederhana dan bisa bekerja dengan baik pada klasifikasi teks[15].

3.4.1. Naïve Bayes untuk Klasifikasi

Klasifikasi Naïve Bayes didasarkan pada teorema Bayes untuk probabilitas bersyarat. Kaitan antara Naïve Bayes dengan klasifikasi, korelasi hipotesis, dan bukti adalah bahwa hipotesis dalam teorema bayes merupakan label kelas yang menjadi target pemetaan dalam klasifikasi, sedangkan bukti merupakan fitur-fitur yang menjadi masukan dalam model klasifikasi. Jika X

adalah vektor masukan yang berisi fitur dan Y adalah label kelas, maka ditulis P(Y|X). Notasi tersebut disebut peluang akhir (*posterior probability*) untuk Y dan P(Y) disebut peluang awal (*prior probability*) dari Y. Sesuai dengan teorema bayes yang sudah dipaparkan pada bab II, maka formulasi Naïve Bayes untuk klasifikasi adalah:

$$P(Y|X) = \frac{P(Y)\ \prod_{i=1}^{q} P(X_i|Y)}{P(X)}. \tag{3.3}$$

$P(Y|X)$ adalah peluang data dengan fitur vektor X pada kelas Y, dimana $X = (x_1, x_2, x_3, \dots, x_n)$. X bisa kita anggap sebagai satu dokumen, dan $(x_1, x_2, x_3, \dots, x_n)$ adalah kata – kata dalam satu dokumen X. $P(Y)$ adalah peluang awal kelas Y. $\prod_{i=1}^{q} P(X_i|Y)$ adalah peluang independen kelas Y dari semua fitur dalam vektor X. Nilai $P(X)$ selalu tetap/konstan sehingga dalam perhitungan prediksi nantinya kita tinggal menghitung bagian $P(Y)\ \prod_{i=1}^{q} P(X_i|Y)$ [21].

Fitur – fitur dalam bayes independen. Maka bisa kita tulis:

$$P(y|x_1, \dots, x_n) = \frac{P(x_1|y)P(x_2|y) \dots P(x_n|y)P(y)}{P(x_1)P(x_1) \dots P(x_n)}$$

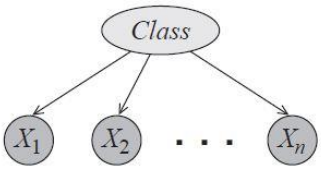
yang bisa diekspresikan sebagai:

$$P(y|x_1, \dots, x_n) = \frac{P(y) \prod_{i=1}^{n} P(x_i|y)}{P(x_1)P(x_1) \dots P(x_n)}$$

karena penyebut konstan, maka bisa diabaikan:

$$P(y|x_1, \dots, x_n) \propto\ P(y) \prod_{i=1}^{n} P(x_i|y)$$

atau bisa dibentuk grafik model (*graphical model*) seperti berikut:



Gambar 3.3. Graphical Model Naïve Bayes[14]

Kemudian, kita mencari peluang dari input yang diberikan untuk semua nilai kemungkinan dari variable kelas y dan menentukan kelas dari nilai peluang terbesar dari peluang posterior. Pernyataan ini bisa ditulis secara matematis seperti berikut:

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y). \qquad (3.4)$$

Di bawah ini adalah contoh dari naïve bayes *classifier*, yaitu keputusan bermain berdasarkan ramalan cuaca dan suhu:

	ya	**tidak**	**P(ya)**	**P(tidak)**
panas	2	2	2/9	2/5
sejuk	4	2	4/9	2/5
Dingin	3	1	3/9	1/5
Total	9	5	100%	100%

(a)

	ya	**tidak**	**P(ya)**	**P(tidak)**
cerah	2	3	2/9	3/5
mendung	4	0	4/9	0/5
hujan	3	2	3/9	2/5
Total	9	5	100%	100%

(b)

Bermain		**P(ya) / P(tidak)**
Ya	9	9/14
Tidak	5	5/14
Total	14	100%

(c)

Tabel 3.8. Peluang (a) ramalan cuaca, (b) suhu, dan (c) hasil akhir keputusan.

Contoh kita ingin menghitung peluang bermain jika dingin dan hujan, maka:

$$P(dingin|ya) = \frac{P(dingin) \times P(ya|dingin)}{P(bermain = ya)}$$

$$= \frac{\left(\frac{4}{14}\right)\left(\frac{3}{4}\right)}{\frac{9}{14}}$$

$$= \frac{3}{9}.$$

$$P(hujan|ya) = \frac{P(hujan) \times P(ya|hujan)}{P(bermain = ya)}$$

$$= \frac{\left(\frac{5}{14}\right)\left(\frac{3}{9}\right)}{\frac{9}{14}}$$

$$= \frac{15}{23}.$$

Misal kita punya fitur $hari\ ini = (hujan, dingin)$, maka peluang bermain hari ini diberikan rumus seperti:

$$P(ya|hari\ ini) = \frac{P(hujan|ya)P(dingin|ya)P(ya)}{P(hari\ ini)}$$

$$P(tidak|hari\ ini) = \frac{P(hujan|tidak)P(dingin|tidak)P(tidak)}{P(hari\ ini)}$$

karena $P(hari\ ini)$ sama di kedua peluang, maka bisa kita abaikan, sehingga peluang menjadi:

$$P(ya|hari\ ini) \propto \frac{3}{9} . \frac{3}{9} . \frac{9}{14} \approx 0.0714$$

$$P(tidak|hari\ ini) \propto \frac{2}{5} . \frac{1}{5} . \frac{5}{14} \approx 0.0286$$

karena $P(ya|hari\ ini) + P(tidak|hari\ ini) = 1$, maka angka berikut bisa dikonversi ke peluang menjadi:

$$P(ya|hari\ ini) = \frac{0.0714}{0.0714 + 0.0286} = 0.714$$

$$P(tidak|hari\ ini) = \frac{0.0286}{0.0714 + 0.0286} = 0.286.$$

Dari hasil di atas, karena $P(ya|hari\ ini) > P(tidak|hari\ ini)$, maka keputusan bermain adalah ‘ya’.

BAB IV
Hasil dan Pembahasan

4.1. Kondisi Data

Dari total ulasan yang didapat melalui proses sraping scraping, terdapat 10 ulasan yang tidak bisa dibaca, sehingga data ulasan seperti itu dihapus dan yang digunakan berjumlah 3067. Ada beberapa ulasan dengan rating baik seperti 4 dan 5 yang isi ulasannya justru negatif dan mengandung kritikan. Jadi untuk pelabelan klasifikasi sentimen, tidak bisa hanya dilihat dari rating saja, diperlukan pelabelan manual. Kondisi data ulasan berdasarkan rating ditunjukkan pada tabel-tabel berikut:

Tabel 4.1. Contoh Beberapa Ulasan

No.	Rating	Ulasan
1.	5	Instant delivery is not shown in application , pls do some adjustment to become easier to understand. Thank you
2.	4	Its nice but too complicated to use.
3.	1	t ff3 hpdp bnmÃ± a p I'm no
4.	4	J. J obviously not b j. Ibj I bb bbin I bb I bb. Ni i

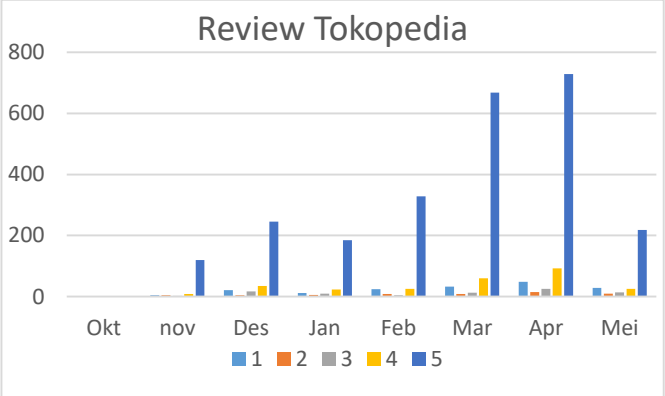
Seperti pada kalimat ulasan nomor 1 di Tabel 4.1. *Contoh Beberapa Ulasan*, kalimat ulasan mengandung kritikan dan penulis ulasan memberikan saran untuk Tokopedia. Ia menulis bahwa fitur pengiriman instan tidak muncul di aplikasi, dan ia menyarankan untuk melakukan pengaturan baru supaya aplikasi lebih mudah dimengerti. Namun ia memberikan rating 5 yang artinya rating baik. Pada kalimat ulasan nomor 2, ulasan mengandung pujian dan kritikan, bahwa aplikasinya bagus tetapi rumit digunakan, dan ia memberikan rating baik yaitu 4. Karena itu, kalimat perlu dipisah untuk mendapatkan aspek yang mungkin berbeda di tiap kalimat. Penulis memisahkan kalimat berdasarkan pemisah titik. Maka, rating tidak bisa dijadikan acuan untuk memberikan label sentimen positif atau negatif. Pada kalimat nomor 3 dan 4

mengandung ulasan yang tidak bisa dipahami, maka ulasan seperti itu perlu dihapus.

Tabel 4.2. Jumlah Ulasan Tiap Rating

Rating	Jumlah ulasan
1	169
2	56
3	84
4	266
5	2492
Total	**3067**

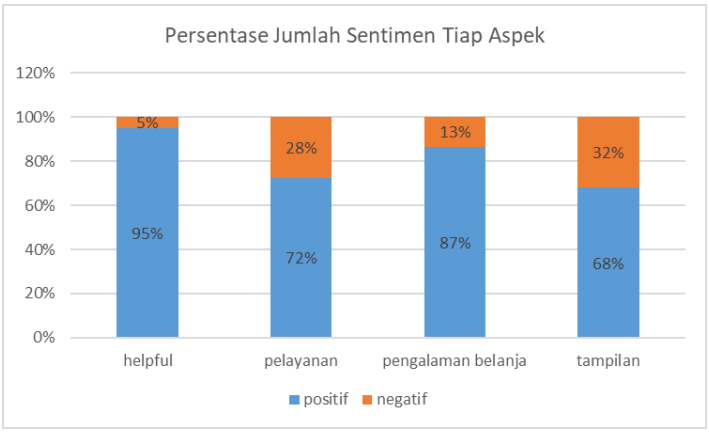
Berdasarkan Tabel 4.2. *Jumlah Ulasan Tiap Rating*, rating 5 memiliki jumlah terbanyak dari rating lainnya yaitu sebanyak 2492 ulasan, dan rating 3 memiliki jumlah terendah yaitu sebanyak 84 ulasan.



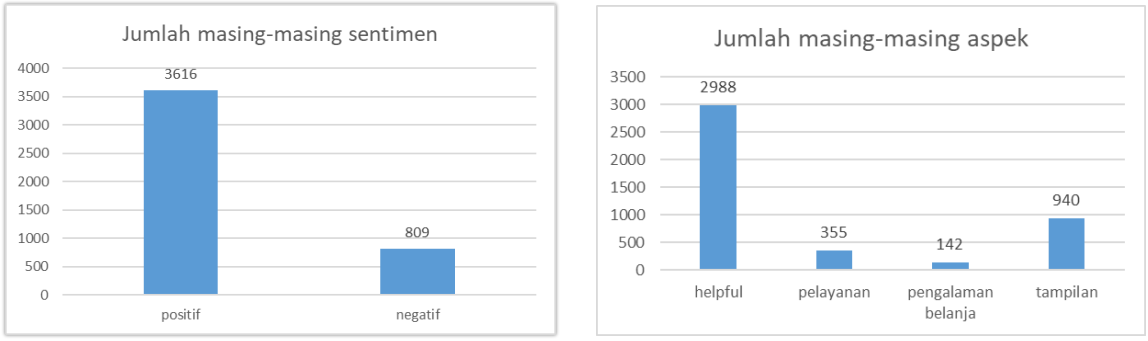
Gambar 4.1. Grafik Rating Tiap Bulan

Gambar 4.1. *Grafik Rating Tiap Bulan* menunjukkan perkembangan jumlah ulasan berdasarkan rating, selama bulan Oktober 2018 hingga Mei 2019. Terlihat bahwa ulasan dengan rating 5 mendominasi di tiap bulan. Jumlah ulasan pada rating lainnya tidak terlalu meningkat pesat antar bulan.

Label yang digunakan untuk klasifikasi sentimen hanya positif dan negatif, dan untuk klasifikasi aspek ada 4. Karena setiap ulasan dipotong per-kalimat dengan pemisah titik, maka jumlah data menjadi 4425 kalimat.



Gambar 4.2. Persentase Jumlah Sentimen Tiap Aspek



(a) (b)

Gambar 4.3. (a) Jumlah Ulasan Label Sentimen, (b) Jumlah Ulasan Label Aspek

Gambar 4.2. *Persentase Jumlah Sentimen Tiap Aspek* menunjukkan persentase dari jumlah sentimen setiap aspek setelah ulasan dipotong menjadi kalimat-kalimat. Sentimen positif mendominasi di tiap aspek. Sedangkan hasil klasifikasi masing – masing sentimen dan aspek seperti pada Gambar 4.3. *(a) Jumlah Ulasan Label Sentimen, (b) Jumlah Ulasan Label Aspek*. Karena jumlah sentimen tidak seimbang, maka perlu dilakukan resampling data untuk mengatasinya, dengan undersampling, oversampling, dan juga gabungan keduanya.

4.2. Hasil Preprocessing

Data teks harus dibersihkan terlebih dahulu supaya bisa diolah dan diubah ke bentuk angka, agar bisa diolah oleh komputer. Preprocessing yang dilakukan

meliputi menghapus simbol dan stopword, case-folding, tokenisasi, dan mengubah kata slang. Penulis tidak menghapus kata negasi seperti “not” untuk memudahkan interpretasi topik, juga penghapusan negasi akan mengubah arti dari ulasannya. Hasilnya disimpan dalam bentuk CSV. Berikut contoh beberapa data awal dari hasil preprocessing:

Tabel 4.3. Beberapa Data Awal Hasil Preprocessing

No.	Date	Rating	Review	Cleaned_review
1	10/17/2018	5	Recomended app	recommended
2	11/12/2018	5	I have relied on Tokopedia to obtain from cheap and mundane items to pricy gadgets that need extra care in handling and shipping	relied cheap mundane item pricy gadget extra care handling shipping
3	11/12/2018	5	Everything were delivered as expected and I am satisfied (so far) with its choice of stores, couriers, user-friendly design, and comprehensive system	delivered expected satisfied choice store courier user friendly design comprehensive
4	11/12/2018	5	Sometimes, Tokopedi	
5	11/12/2018	5	Full Review	

Pada Tabel 4.3. *Beberapa Data Awal Hasil Preprocessing* hasil kalimat ulasan yang sudah di-preprocessing dimasukkan ke kolom cleaned_review, pada kalimat nomor 4 dan 5 kolom cleaned_review kosong karena kalimat di kolom review merupakan kata yang masuk dalam list stopword, sehingga kata – kata itu dihapus.

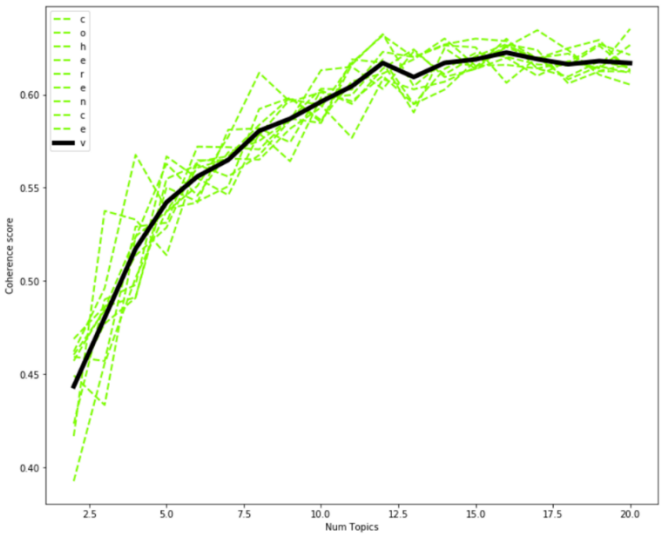
4.3. Pembobotan Kata

Pembobotan kata menggunakan tf. Parameter pada fungsi Tf_vectorizer yaitu max_df dan min_df. Nilai yang digunakan adalah max_df = 0.75 dan min_df = 5. Nilai max_df tersebut artinya kita mengabaikan kata yang muncul lebih dari 75% dokumen, dan nilai min_df tersebut artinya kita mengabaikan kata yang muncul

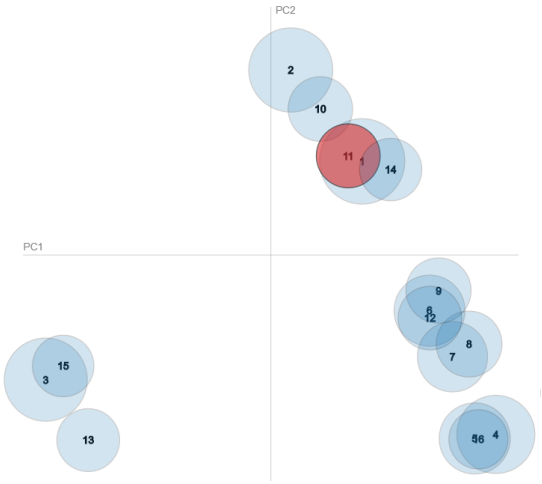
kurang dari 5 dokumen. Bentuk tf yang didapat adalah (4425, 327), artinya terdapat 4425 baris (kalimat) dan 327 kata unik.

4.4. Penentuan Jumlah Aspek

Jumlah aspek ditentukan dengan melakukan clustering dari seluruh data. Proses clustering akan menghasilkan jumlah topik (aspek) dan interpretasi topik apa saja yang didapat dari seluruh data tersebut. Hal ini dilakukan agar diketahui jumlah aspek yang harus diberikan dalam proses anotasi klasifikasi aspek. Dalam penentuan jumlah topik dilakukan iterasi dari 2 sampai 20 topik, dan dilihat nilai coherence dari masing – masing topik.



Gambar 4.4. Grafik Nilai Coherence 20 Topik



Gambar 4.5. Visualisasi LDA 16 Topik

Pada Gambar 4.4. *Grafik Nilai Coherence 20 Topik* menunjukkan bahwa grafik rata – rata nilai coherence (garis hitam) dari seluruh iterasi meningkat seiring bertambahnya jumlah topik. Pemilihan banyak topik bisa dilakukan dengan melihat nilai coherence score tertinggi. Dari 20 iterasi topik, jumlah 16 topik memiliki nilai coherence tertinggi. Tetapi dari Gambar 4.5. *Visualisasi LDA 16 Topik* visualisasi dengan LDAvis justru banyak cluster yang beririsan. Hal itu mengindikasikan bahwa cluster yang beririsan atau berdekatan bisa dijadikan satu cluster, maka cluster topik bisa dibatasi lebih kecil dari 20.

Gambar 4.6. Visualisasi LDA 4 Topik

Jika dilihat pada jumlah topik = 4 pada Gambar 4.6. *Visualisasi LDA 4 Topik*, bentuk LDAvis menunjukkan cluster yang terpisah, juga kata – kata tiap cluster bisa diinterpretasi topiknya, walaupun nilai coherence pada topik yang lebih dari 4 akan meningkat. Nilai coherence pada 4 topik adalah 0.53707. Hasil *top words* dari tiap cluster topik ditunjukkan seperti pada tabel berikut:

Tabel 4.4. Sebaran Kata Tiap Jumlah Topik

topik	Kata tiap topik	Interpretasi topik
1	online apps shop **user nice friendly** best **useful** great recommended **interface** helpful trusted store buy	Tampilan
2	**good** price helpful **service product** like thing **help** job **work response** buy buying **fast**	Pelayanan
3	**easy use** best **place market** indonesia **helpful** simple commerce great fast useful buy **payment discount**	Kebermanfaatan
4	not great time love seller **customer** update much better **product service cashback** get **experience** bad	Pengalaman belanja

Dari kumpulan kata pada tiap topik seperti pada Tabel 4.4. *Sebaran Kata Tiap Jumlah Topik*, bisa diinterpretasi tiap topik membahas tentang apa. Kata yang dicetak tebal merupakan kata yang lebih menentukan interpretasi topik dibanding kata lainnya di tiap cluster topik.

4.5. Klasifikasi Sentimen

Penulis menggunakan model klasifikasi Naïve Bayes. Karena data tidak seimbang, maka diperlukan beberapa cara untuk mengatasinya. Teknik yang digunakan adalah *RandomUnderSampler* untuk undersampling, *RandomOverSampler* untuk oversampling, dan SMOTEENN untuk gabungan keduanya. Dengan beberapa teknik tersebut, dilihat evaluasi *precision, recall*, dan *F1-Score* dari masing-masing teknik, dibandingkan dengan evaluasi pada data asli.

Tabel 4.5. Hasil Evaluasi Klasifikasi Sentimen

Tanpa resampling

label	Precision	Recall	F1-Score
-1	0.78	0.62	0.69
1	0.94	0.97	0.95

Akurasi tanpa resampling: 91.9 %

Undersampling

label	Precision	Recall	F1-Score
-1	0.74	0.75	0.74
1	0.96	0.95	0.96

Akurasi undersampling: 92.4%

Oversampling

label	Precision	Recall	F1-Score
-1	0.71	0.81	0.76
1	0.97	0.94	0.96

akurasi oversampling: 92.5 %

SMOTEENN

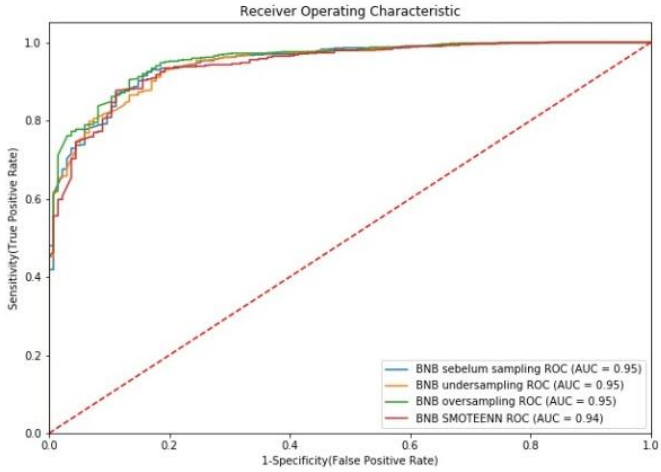
Label	Precision	Recall	F1-Score
-1	0.74	0.75	0.74
1	0.97	0.95	0.96

akurasi SMOTEENN: 92.4 %

Tabel 4.6. Tabel Confusion Matrix

		Nilai Prediksi	
		negatif	positif
Nilai aktual	negatif	110	25
	positif	44	742

Jika dilihat pada Tabel 4.5. *Hasil Evaluasi Klasifikasi Sentimen*, akurasi tertinggi dihasilkan dengan teknik oversampling. Pada confusion matrix Tabel 4.6. *Tabel Confusion Matrix* menunjukkan bahwa banyak data yang benar terklasifikasikan salah (negatif) sebanyak 110 data, banyak data yang terklasifikasi benar (positif) sebanyak 742 data, data yang salah memprediksi data positif sebanyak 25 data, dan data yang salah memprediksi data negatif sebanyak 44 data. Selain itu, evaluasi juga bisa dilihat menggunakan kurva ROC dan AUC, karena ukuran performanya lebih cocok digunakan pada data yang tidak seimbang[22]. Gambar 4.7. *Grafik Kurva ROC dan Nilai AUC dengan Naïve Bayes* adalah grafik kurva ROC dan daerah AUC dari data testing, menunjukkan bahwa nilai AUC pada oversampling juga sudah baik.



Gambar 4.7. Grafik Kurva ROC dan Nilai AUC dengan Naïve Bayes

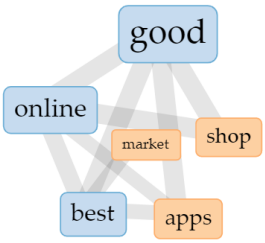
4.6. Visualisasi dan Interpetasi Data

Visualisasi data menggunakan wordcloud dan wordlink pada sentimen di setiap aspek, seperti ditunjukkan pada gambar berikut:

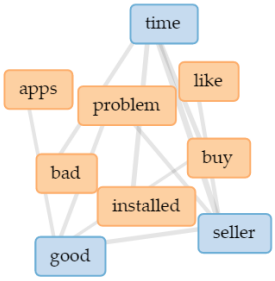
(a)

(b)

Gambar 4.8. Wordcloud Aspek Helpful Sentimen (a) Positif dan (b) Negatif



(a)



(b)

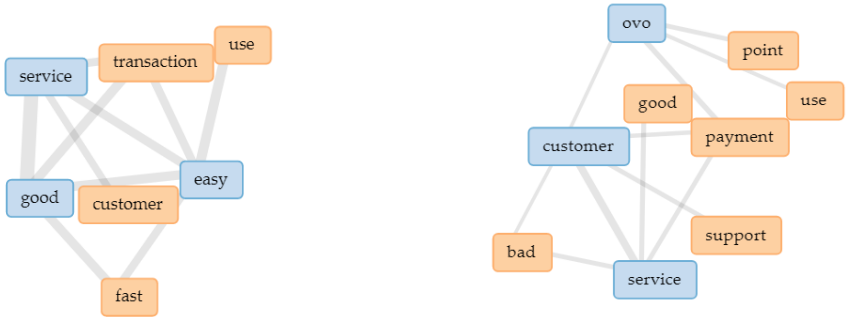
Gambar 4.9. Wordlink Aspek Helpful Sentimen (a) Positif dan (b) Negatif

Berdasarkan Gambar 4.8. *Wordcloud Aspek Helpful Sentimen (a) Positif dan (b) Negatif*, sentimen positif pada aspek kebermanfaatan (helpful) menunjukkan bahwa Tokopedia menjadi aplikasi yang mudah digunakan untuk berbelanja online, tempat yang direkomendasikan untuk berbelanja, dan merupakan aplikasi yang terpercaya. Sedangkan hal yang dinilai negatif oleh masyarakat adalah para penjual yang memberikan foto tidak asli, cashback yang justru masuk ke ovopoint, masalah kupon yang tidak bisa dipakai, dan harapan dari pengguna untuk menghapus akun yang sudah tidak terpakai.

(a)

(b)

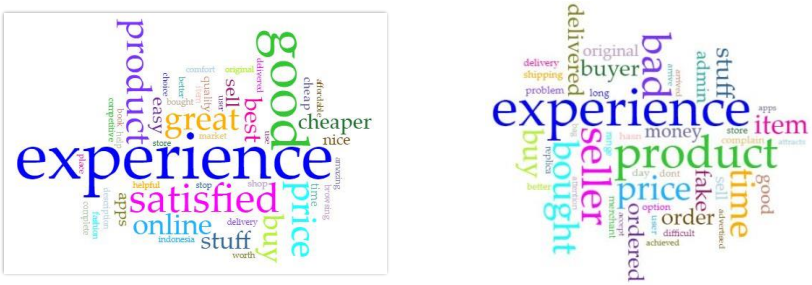
Gambar 4.10. Wordcloud Aspek Pelayanan Sentimen (a) Positif dan (b) Negatif



(a) (b)

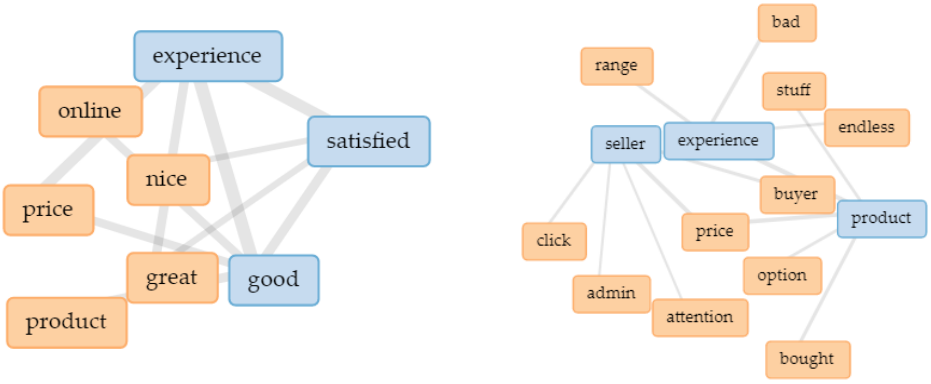
Gambar 4.11. Wordlink Aspek Pelayanan Sentimen (a) Positif dan (b) Negatif.

Wordcloud aspek pelayanan pada Gambar 4.10. *Wordcloud Aspek Pelayanan Sentimen (a) Positif dan (b) Negatif* menunjukkan bahwa Tokopedia memberikan pelayanan *customer service* yang baik, responnya cepat, dan transaksi yang aman. Namun, customer service juga menjadi faktor kritikan dari pengguna. Pada wordcloud negatif juga muncul kata customer service, karena tidak responsif, tidak interaktif, juga tidak adanya telepon yang bisa dihubungi untuk menyampaikan keluhan. Selain itu, penjual juga mengirimkan barang yang tidak sesuai permintaan pembeli. Terkait pembayaran ovo, masyarakat mengeluhkan perubahan ovo menjadi tokocash, juga mereka tidak bisa mengaktivasi ovo-nya.



(a) (b)

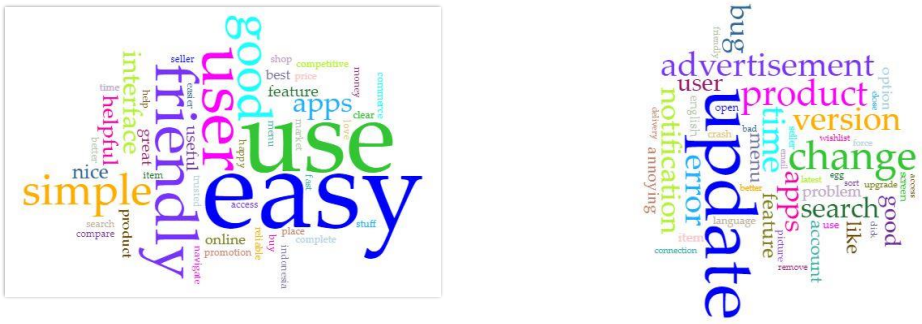
Gambar 4.12. Wordcloud Aspek Pengalaman Belanja Sentimen (a) Positif dan (b) Negatif



(a) (b)

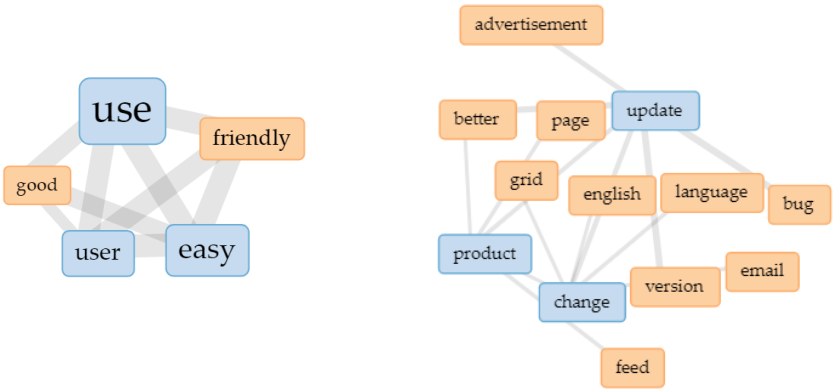
Gambar 4.13. Wordlink Aspek Pengalaman Belanja Sentimen (a) Positif dan (b) Negatif

Pada Gambar 4.12. *Wordcloud Aspek Pengalaman Belanja Sentimen (a) Positif dan (b) Negatif*, aspek pengalaman belanja menunjukkan bahwa pengguna merasa puas menggunakannya, juga harga produk yang murah, banyak barang yang mudah dicari, dan aplikasi yang baik untuk penjual dan pembeli. Namun, pengalaman buruk yang pengguna alami diantaranya penjual yang tidak mengembalikan barang yang dikomplain, serta uang yang tidak dikembalikan. Juga penjual yang lama dalam memroses transaksi, dan barang yang dibeli tidak dikirim sesuai aslinya.



(a) (b)

Gambar 4.14. Wordcloud Aspek Tampilan Sentimen (a) Positif dan (b) Negatif



(a) (b)

Gambar 4.15. Wordlink Aspek Tampilan Sentimen (a) Positif dan (b) Negatif

Berdasarkan Gambar 4.14. *Wordcloud Aspek Tampilan Sentimen (a) Positif dan (b) Negatif*, para pengguna senang dengan aplikasi Tokopedia karena mudah digunakan, user friendly, dan juga simpel. Sedangkan pada sentimen negatif, pengguna mengeluhkan terkait update. Ada fitur yang hilang, tidak bisa login, dan beberapa fitur yang tidak penting setelah dilakukan update. Selain itu juga perubahan setelah update lebih buruk dan adanya iklan yang mengganggu tampilan.

BAB V
Penutup

5.1. Kesimpulan

Dari hasil penelitian dan pemaparan yang sudah dijelaskandisimpulkan bahwa ada beberapa aspek yang merangkum opini pengguna Tokopedia berdasarkan ulasan yang ditulis pada kolom komentar di Play Store, dimana masing – masing memiliki sentimen positif dan negatif. Hasil tersebut berdasarkan data komentar selama bulan Oktober 2018 hingga 10 Mei 2019, sebanyak 4425 kalimat ulasan. Penentuan jumlah aspek didapat dari clustering topic menggunakan LDA, didapat 4 aspek yaitu kebermanfaatan (helpful), pelayanan, pengalaman belanja, dan tampilan. Hasil topik dari LDA ini digunakan untuk acuan dalam proses anotasi aspek. Lalu, klasifikasi sentimen menggunakan Naïve Bayes dan teknik oversampling karena data antara sentimen negatif dan positif tidak seimbang, didapat akurasi sebesar 92.5%. Juga nilai AUC menunjukkan nilai yang baik yaitu 0,95.

Kelebihan Tokopedia dari aspek kebermanfaatan adalah aplikasinya mudah digunakan dan direkomendasikan untuk digunakan, kekurangannya adalah penjual tidak memberikan foto asli, cashback yang masuk ke ovo point, dan harapan untuk menghapus akun yang tidak terpakai. Dari aspek pelayanan, pengguna senang karena customer service baik dan responnya cepat, tetapi ada pula yang merasakan bahwa customer service sulit dihubungi, juga dari pelayanan penjual yang salah mengirim barang. Pada aspek pengalaman belanja, sentimen positif yang diberikan tentang produk yang murah dan mudah dicari, sedangkan sentimen negatifnya adalah penjual tidak mengembalikan barang yang dikomplain pembeli. Dari aspek tampilan, pengguna senang karena user friendly dan mudah digunakan, tetapi adanya iklan yang muncul setelah proses update mengganggu pengguna dalam proses belanja pada aplikasi.

5.2. Saran

Ada beberapa saran untuk peneliti lainnya yang tertarik untuk melanjutkan penelitian terkait ini:

1. Klasifikasi bisa menggunakan model *machine learning* lainnya, atau *deep learning*.
2. Bisa menggunakan teknik lainnya untuk mengatasi data tidak seimbang.
3. Bisa menggunakan model clustering lainnya.
4. Menggunakan data bahasa Indonesia dan data diambil diurutkan berdasarkan komentar terbaru.

REFERENSI

[1] Wartaekonomi, "alasan belanja online," 2018. [Online]. Available: https://www.wartaekonomi.co.id/read203120/ternyata-ini-5-alasan-millenial-gemar-belanja-online-enggak-heran-deh.html. [Accessed: 03-Jul-2019].

[2] Katadata, "e-commerce paling diminati," 2019. [Online]. Available: https://databoks.katadata.co.id/datapublish/2019/01/31/ini-e-commerce-indonesia-paling-diminati-pada-triwulan-iv-2018.

[3] Okezone, "11 Temuan Penting Peta E-Commerce Indonesia di Q1 2019," 2019. [Online]. Available: https://techno.okezone.com/read/2019/05/10/207/2054228/11-temuan-penting-peta-e-commerce-indonesia-di-q1-2019.

[4] Tun Thura, J. Na, dan C. S. G. Khoo, "Aspect-based sentiment analysis of movie reviews on discussion boards," vol. 36, no. 6, pp. 823–848, 2010.

[5] F. Xianghua, L. Guo, G. Yanyan, dan W. Zhiqiang, "Knowledge-Based Systems Multi-aspect sentiment analysis for Chinese online social reviews based on topic modeling and HowNet lexicon," *Knowledge-Based Syst.*, vol. 37, pp. 186–195, 2013.

[6] D. Ekawati dan M. L. Khodra, "Aspect-based Sentiment Analysis for Indonesian Restaurant Reviews," 2017.

[7] A. Josi, L. A. Abdillah, dan Suryayusra, "PENERAPAN TEKNIK WEB SCRAPING PADA MESIN PENCARI ARTIKEL ILMIAH."

[8] Q. Wu, "an efficient Random forest algorithm for imbalanced text," *elsevier*, 2014.

[9] A. Alghunaim, "A Vector Space Approach for Aspect-Based Sentiment Analysis," 2015.

[10] "slang." [Online]. Available: https://kbbi.web.id/slang. [Accessed: 16-Sep-2019].

[11] A. Kyoomarsi, F., Khosravi, H., Eslami, E., Dehkordy, P. K., dan Tajoddin, "Optimizing Text Summarization Based on Fuzzy Logic," in *IEEE*, 2008.

[12] C. Manning, P. Raghavan, dan H. Schutze, *An Introduction to Information Retrieval*, no. c. 2009.

[13] J. Wang, S., Lo, D., dan Lawall, "Compositional Vector Space Models for Improved Bug Localization," *IEEE*, 2014.

[14] D. Koller dan N. Friedman, *probabilstic Graphical models*. London, England.

[15] C. C. Aggarwal, *Data Mining*. NY, USA: Springer, 2015.

[16] T. Fawcett, "An introduction to ROC analysis," vol. 27, pp. 861–874, 2006.

[17] Suwarno dan A. Abdillah, "PENERAPAN ALGORITMA BAYESIAN REGULARIZATION BACKPROPAGATION UNTUK MEMPREDIKSI PENYAKIT DIABETES," *MIPA*, vol. 39, no. 45, pp. 150–158, 2016.

[18] Y. Li, B. Jia, Y. A. O. Guo, dan X. Chen, "Mining User Reviews for Mobile App Comparisons," 2017, vol. 1, no. 3.

[19] D. M. Blei, A. Y. Ng, dan M. I. Jordan, "Latent Dirichlet Allocation," vol. 3, pp. 993–1022, 2003.

[20] A. Arlina dan M. Liebenlito, "Sequential Topic Modeling: A Case Study on Indonesian LGBT Conversation on Twitter," *InPrime*, vol. 1, no. 1, pp. 17–31, 2019.

[21] E. Prasetyo, *Data Mining - Konsep dan Aplikasi Menggunakan Matlab*. Yogyakarta: ANDI, 2012.

[22] N. V. Chawla, "Data Mining for Imbalanced Datasets: An Overview", *Springer*, 2009.

LAMPIRAN

A. Scraping data

```
from selenium import webdriver
from time import sleep
import bs4
import pandas as pd
import requests
from selenium.webdriver.common.keys import Keys
import time

# Change this number to get more or less reviews
# Current set of x=100 yielded 11,312 reviews
x = 1000

link =
"https://play.google.com/store/apps/details?id=com.tokopedia.tkpd&
showAllReviews=true"

driver = webdriver.Chrome('C:/WinPython_64bit/notebooks/Google-
Play-Store-Review-Extractor-master/chromedriver.exe')
driver.get(link + '&showAllReviews=true')

num_clicks = 0
num_scrolls = 0
while num_clicks <= x and num_scrolls <= x*5:
    try:
        show_more =
driver.find_element_by_xpath('//*[@id="fcxH9b"]/div[4]/c-
wiz/div/div[2]/div/div[1]/div/div/div[1]/div[2]/div[2]/div/content
/span')
        show_more.click()
        num_clicks += 1

    except:
        html = driver.find_element_by_tag_name('html')
        html.send_keys(Keys.END)
        num_scrolls +=1
        time.sleep(2)

soup = bs4.BeautifulSoup(driver.page_source, 'html.parser')
h2 = soup.find_all('h2')
```

```
results_df = pd.DataFrame()
for ele in h2:
    if ele.text == 'Reviews':
        c_wiz = ele.parent.parent.find_all('c-wiz')
        for sibling in c_wiz[0].next_siblings:
            try:
                #print (sibling)
                comment_shift = 0
                spans = sibling.find_all('span')
                for user_block in range(0,len(spans)):
                    i = user_block *10
                    name = spans[i+0+comment_shift].text
                    try:
                        rating =
spans[i+1+comment_shift].div.next_element['aria-label']
                        rating = str(''.join(filter(str.isdigit,
rating)))
                    except:
                        comment_shift += 2
                        continue
                    date = spans[i+2+comment_shift].text
                    review = spans[i+8+comment_shift].text
                    print ('Name: %s\nRating: %s\nDate:
%s\nReview: %s\n' %(name, rating, date, review))
                    temp_df = pd.DataFrame([[date, rating, name,
review]], columns = ['Date','Rating','User','Review'])

                    results_df = results_df.append(temp_df)
            except:
                continue

results_df = results_df.reset_index(drop=True)
results_df.to_csv('C:/WinPython_64bit/notebooks/Google-Play-Store-
Review-Extractor-master/review_tokped.csv', index=False)

driver.close()
```

B. Import modul

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
#import warnings; warnings.simplefilter('ignore')
import time
import pyLDAvis, pyLDAvis.sklearn; pyLDAvis.enable_notebook()
import pickle, TSutanto_lib as ittc
import imblearn
import numpy as np
import sklearn

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import label_binarize
from sklearn.multiclass import OneVsRestClassifier
from scipy import interp
from sklearn.metrics import roc_curve, auc
```

```
from sklearn.decomposition import NMF
from sklearn import cluster
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import GridSearchCV, train_test_split
from imblearn.under_sampling import RandomUnderSampler
from imblearn.over_sampling import RandomOverSampler
from imblearn.combine import SMOTETomek
from sklearn import neighbors
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import KFold #for K-fold cross
validation
from sklearn.model_selection import cross_val_score #score
evaluation
from sklearn.model_selection import cross_val_predict #prediction
from sklearn import metrics
from sklearn.feature_extraction.text import
TfidfVectorizer,CountVectorizer
from sklearn.model_selection import train_test_split
from imblearn.datasets import make_imbalance
from imblearn.under_sampling import NearMiss # underSampling
from imblearn.over_sampling import SMOTE # OverSampling
from imblearn.combine import SMOTEENN # Combination of the 2
from imblearn.pipeline import make_pipeline
from imblearn.metrics import classification_report_imbalanced
from sklearn.metrics import accuracy_score,confusion_matrix,
classification_report
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import cross_val_score
from sklearn import svm, tree, neighbors
from sklearn.naive_bayes import GaussianNB, MultinomialNB,
BernoulliNB
from sklearn.svm import SVC
from sklearn.neural_network import MLPClassifier
from sklearn.decomposition import LatentDirichletAllocation as LDA
from tqdm import tqdm_notebook as tqdm


def LoadStopWords(lang):
    L = lang.lower().strip()
    if L == 'en' or L == 'english' or L == 'inggris':
        lemmatizer = WordNetLemmatizer()
        stops =  set([t.strip() for t in LoadDocuments(file =
'C:/WinPython_64bit/notebooks/Google-Play-Store-Review-Extractor-
master/stopwords_eng.txt')[0]])
     elif L == 'id' or L == 'indonesia' or L=='indonesian':
        lemmatizer = Indonesian()
        stops = set([t.strip() for t in LoadDocuments(file =
'C:/WinPython_64bit/notebooks/Google-Play-Store-Review-Extractor-
master/stopwords_id.txt')[0]])
    else:
        print('Warning, language not recognized. Empty StopWords
Given')
         stops = set(); lemmatizer = None
     return stops, lemmatizer
```

```
def fixTags(T):
    getHashtags = re.compile(r"#(\w+)")
    pisahtags = re.compile(r'[A-Z][^A-Z]*')
    t = T
    tagS = re.findall(getHashtags, T)
    for tag in tagS:
        proper_words = ' '.join(re.findall(pisahtags, tag))
        t = t.replace('#'+tag,proper_words)
    return t

def readBz2(file):
    with bz2(file, "r") as bzData:
        txt = []
        for line in bzData:
            try:
                txt.append(line.strip().decode('utf-8','replace'))
            except:
                pass
    return ' '.join(txt)

def LoadDocuments(dPath=None,types=None, file = None): # types =
['pdf','doc','docx','txt','bz2']
    Files, Docs = [], []
    if types:
        for tipe in types:
            Files += crawlFiles(dPath,tipe)
    if file:
        Files = [file]
    if not types and not file: # get all files regardless of their
extensions
        Files += crawlFiles(dPath)
    for f in Files:
        if f[-3:].lower()=='pdf':
            try:
                Docs.append(PDF(f).string)
            except:
                print('error reading{0}'.format(f))
        elif f[-3:].lower()=='txt' or f[-3:].lower()=='dic':
            try:
                df=open(f,"r",encoding="utf-8", errors='replace')
                Docs.append(df.readlines());df.close()
            except:
                   print('error reading{0}'.format(f))
        elif f[-3:].lower()=='bz2':
            try:
                Docs.append(readBz2(f))
            except:
                print('error reading{0}'.format(f))
        elif f[-4:].lower()=='docx':
            try:
                Docs.append(docx2txt.process(f))
            except:
                print('error reading{0}'.format(f))
        elif f[-3:].lower()=='csv':
            Docs.append(pd.read_csv(f))
```

```
        else:
            print('Unsupported format {0}'.format(f))
    if file:
        Docs = Docs[0]
    return Docs, Files

def DelPic(text): #untuk menghilangkan informasi gambar
    D = text.split()
    D = [d for d in D if 'pic.twitter.com' not in d]
    return ' ' .join(D)

def LoadSlang(DirSlang):
    Slangs =LoadDocuments(file = DirSlang)
    SlangDict={}
    for slang in Slangs[0]:
        try:
            key, value = slang.split(':')
            SlangDict[key.strip()] = value.strip()
        except:
            pass
    return SlangDict

#POS Tagging
from nltk.tag import CRFTagger
def postag(text):
    #Tokenisasi Data
    tokenized_sents = word_tokenize(text)
    #pemberian Tag tiap token
    ct = CRFTagger()
    ct.set_model_file('C:/WinPython_64bit/notebooks/Google-Play-
Store-Review-Extractor-master/CRFTagger-
1.0/CRFTagger/model/model.txt')
    #directorynya disesuaikan meletakan file crfnya, harus
download dlu file crfnya
    pt = ct.tag(tokenized_sents)
    ptN = []
    noun = set(['NN','NNP', 'NNS','NNPS'])
    tmp = []
    for w in pt:
        if w[1] in noun:
            tmp.append(w[0])
    if len(tmp)>0:
        ptN.append(' '.join(tmp))
    return ' '.join(ptN)

def cleanText(T, fix={}, lang = 'id', lemma=None, stops = set(),
symbols_remove = False, min_charLen = 0):
    # lang & stopS only 2 options : 'en' atau 'id'
    # symbols ASCII atau alnum
    pattern = re.compile(r'http[s]?://(?:[a-zA-Z]|[0-9]|[$-
_@.&+]|[!*\(\),]|(?:%[0-9a-fA-F][0-9a-fA-F]))+')
    t = re.sub(pattern,' ',T) #remove urls if any
    t = DelPic(t)
    t = unescape(t) # html entities fix
    t = fixTags(t) # fix abcDef
```

```
    t = t.lower().strip() # lowercase
    t = unidecode(t)
    t = ''.join(''.join(s)[:2] for _, s in itertools.groupby(t)) #
remove repetition
    t = sent_tokenize(t) # sentence segmentation. String to list
    for i, K in enumerate(t):
        if symbols_remove:
            K = re.sub(r'[^.,a-zA-Z0-9 \n\.]',' ',K)

        cleanList = []
        if lang =='en':
            listKata = word_tokenize(K) # word tokenize
            for token in listKata:
                if token in fix.keys():
                    token = fix[token]
                if lemma:
                    token = lemma.lemmatize(token)
                if stops:
                    if len(token)>=min_charLen and token not in
stops:
                        cleanList.append(token)
                else:
                    if len(token)>=min_charLen:
                        cleanList.append(token)
            t[i] = ' '.join(cleanList)
        else:
            if lemma:
                K = lemma(K)
                listKata = [token.text for token in K]
            else:
                listKata = TextBlob(K).words

            for token in listKata:
                if token in fix.keys():
                    token = fix[token]

                if lemma:
                    token = lemma(token)[0].lemma_
                if stops:
                    if len(token)>=min_charLen and token not in
stops:
                        cleanList.append(token)
                else:
                    if len(token)>=min_charLen:
                        cleanList.append(token)
            t[i] = ' '.join(cleanList)
    return ' '.join(t)

stops, lemmatizer = LoadStopWords(lang='en')
Slangs=LoadSlang( 'C:/WinPython_64bit/notebooks/Google-Play-Store-
Review-Extractor-master/slang.txt')
```

C. Preprocessing

```
# VSM dari csv hasil hapus review yang aneh kalimatnya
```

```
import pandas as pd
from sklearn.feature_extraction.text import
TfidfVectorizer,CountVectorizer

data = pd.read_csv('tokped_bersih3 fix - Copy.csv')
cleanreview = data['Cleaned_review']

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)

#print kata unik tfidf
tfidf_vectorizer=TfidfVectorizer(use_idf=True)

# just send in all your docs here
tfidf_vectorizer_vectors=tfidf_vectorizer.fit_transform(listdf)

first_vector_tfidfvectorizer=tfidf_vectorizer_vectors[1]

# place tf-idf values in a pandas data frame
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 == 0]

print('banyaknya data positif :',len(pos))
print('banyaknya data negatif :',len(neg))

#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)
```

D. Split Kalimat

```
nama_kolom = list(data.columns)
dates =[]
rating=[]
sentimen=[]
aspek=[]
review=[]
clean_rv=[]
for idx,dt in enumerate(data['Review']):
    l_dt = dt.split('.')
    for dts in l_dt:
        if dts!='':
            try:
                dates.append(data[nama_kolom[0]][idx])
                rating.append(data[nama_kolom[1]][idx])
                sentimen.append(data[nama_kolom[2]][idx])
                aspek.append(data[nama_kolom[3]][idx])
                review.append(dts)
                clean_rv.append(data[nama_kolom[5]][idx])
            except Exception as err:
                print(err)

dicti =
{'Date':dates,'Rating':rating,'Sentimen':sentimen,'aspek':aspek,'R
eview':review,'clean_review':clean_rv}
new_data = pd.DataFrame(dicti)
new_data.to_excel('split_kalimat.xlsx')
```

E. Clustering Topik

```
lda=[]
for n_topics in tqdm(range(2,9)):
    model=LDA(n_components=n_topics, learning_method='batch',
random_state=0).fit(tfidf)
    lda.append(model)

vsm_topics=[]
for i in tqdm(lda):
    vsm_topics.append(i.transform(tfidf))

doc_topic=[]
for i in tqdm(range(len(vsm_topics))):
    doc_topic.append([a.argmax()+1 for a in vsm_topics[i]])

for i in doc_topic:
    sns.countplot(i)
    plt.show()

n_topics = 4
lda = LDA(n_components=n_topics, learning_method='batch',
random_state=0).fit(tfidf)

doc_topic =  [a.argmax()+1 for a in tqdm(vsm_topics)] # topic of
docs
doc_topic[:10]
sns.countplot(doc_topic)
```

```
Top_Words=20
print('Printing top {0} Topics, with top {1}
Words:'.format(n_topics, Top_Words))
ittc.print_Topics(lda, tfidf_term, n_topics, Top_Words)
```

F. Klasifikasi Sentimen

```
# bandingkan dengan hasil original dengan undersampling dan
oversampling
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('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('Akurasi oversampling test = ', accuracy_score(y_test,
y_bnb))

#both
smt = SMOTEENN(ratio='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('Akurasi combination = ', accuracy_score(y_test, y_bnb))

#probs sebelum resampling
bern=BernoulliNB()
bern.fit(X_train.toarray(), y_train)
probs_bnb = bern.predict_proba(X_test)
probs_bnb = probs_bnb[:, 1]
fpr, tpr, thresholds = roc_curve(y_test, probs_bnb)
```

```
roc_auc = auc(fpr, tpr)

#probs dari undersampling
bnb=BernoulliNB()
bnb.fit(X_rm, y_rm)
probs_rm_bnb = bnb.predict_proba(X_test)
probs_rm_bnb = probs_rm_bnb[:, 1]
fpr, tpr, thresholds = roc_curve(y_test, probs_rm_bnb)
roc_auc = auc(fpr, tpr)

#probs oversampling
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)

#probs both sampling
bnb.fit(X_smt, y_smt)
probs_smt_bnb = bnb.predict_proba(X_test)
probs_smt_bnb = probs_smt_bnb[:, 1]
fpr, tpr, thresholds = roc_curve(y_test, probs_smt_bnb)
roc_auc = auc(fpr, tpr)

probsNB=[probs_bnb, probs_rm_bnb, probs_ros_bnb, probs_smt_bnb]
models=["BNB sebelum sampling", "BNB undersampling", "BNB
oversampling", "BNB SMOTEENN"]

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

#Save Model
Pkl_Filename = 'bnb_sentimen.pkl'
with open(Pkl_Filename, 'wb') as file:
    pickle.dump(bnb, file)
print(bnb)
```

```
with open(Pkl_Filename, 'rb') as file:
    svm = pickle.load(file)
print(svm)
```