

**SENTIMEN ANALISIS ULASAN PENGGUNA AGODA DAN
TRAVELOKA MENGGUNAKAN SUPPORT VECTOR MACHINE
DENGAN DAN TANPA QUERY EXPANSION RANKING**

SKRIPSI

Diajukan untuk memenuhi salah satu syarat
memperoleh gelar Sarjana Komputer
Program Studi Informatika



Disusun oleh:
Guido Mario Enno Epriliano
195314006

**PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS SANATA DHARMA
YOGYAKARTA
2023**

**SENTIMENT ANALYSIS OF AGODA AND TRAVELOKA USER
REVIEWS USING SUPPORT VECTOR MACHINE WITH AND WITHOUT
QUERY EXPANSION RANKING**

THESIS

Present as Partial Fulfillment of the Requirement
To Obtain the *Sarjana Komputer* Degree
In Informatics Study Program



By:
Guido Mario Enno Epriliano
195314006

**INFORMATICS STUDY PROGRAM
FACULTY OF SCIENCE AND TECHNOLOGY
SANATA DHARMA UNIVERSITY
YOGYAKARTA
2023**

HALAMAN PERSETUJUAN

SKRIPSI

**SENTIMEN ANALISIS ULASAN PENGGUNA AGODA DAN
TRAVELOKA MENGGUNAKAN SUPPORT VECTOR MACHINE
DENGAN DAN TANPA QUERY EXPANSION RANKING**

Disusun oleh :

Guido Mario Enno Epriliano

NIM: 195314006

Telah disetujui oleh :

Dosen Pembimbing,

Capti
Dr. Ir. Ridowati Gunawan, S.Kom., M.T.

Tanggal 19 Juli 2023

HALAMAN PENGESAHAN

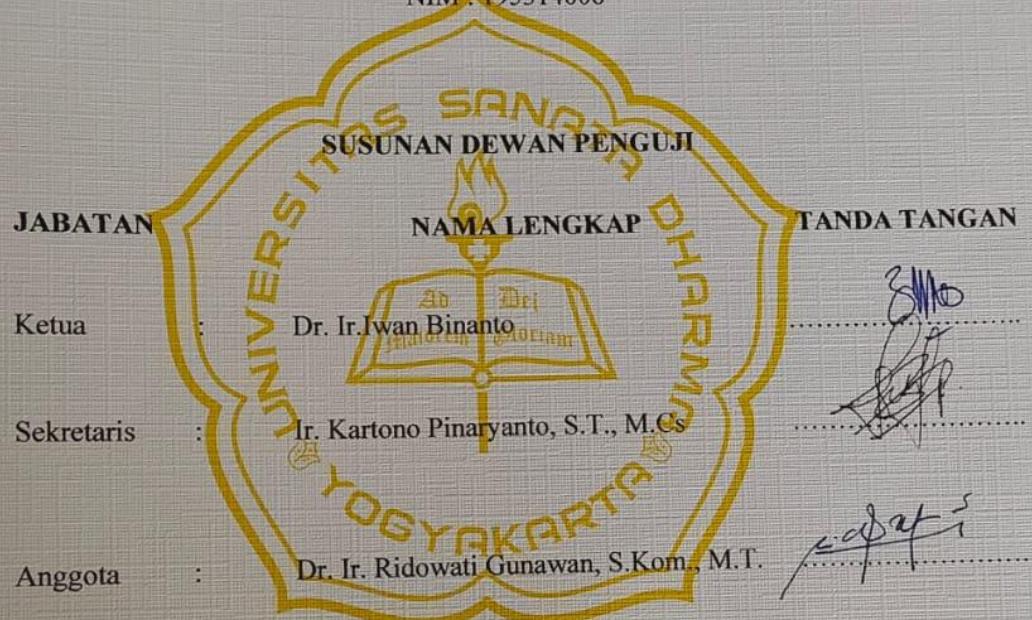
SKRIPSI

SENTIMEN ANALISIS ULASAN PENGGUNA AGODA DAN TRAVELOKA MENGGUNAKAN *SUPPORT VECTOR MACHINE* DENGAN DAN TANPA *QUERY EXPANSION RANKING*

Dipersiapkan dan ditulis oleh :

Guido Mario Enno Epriliano

NIM. 195314006



Yogyakarta, 19 Juli 2023

Fakultas Sains dan Teknologi
Universitas Sanata Dharma

Dekan,



MOTTO DAN PERSEMBAHAN

Karya ini saya persembahkan kepada:

Tuhan Yang Maha Esa

Keluarga

Teman-teman dan kerabat

Dan secara khusus

Universitas Sanata Dharma

PERNYATAAN KEASLIAN KARYA

Saya menyatakan dengan sesungguhnya bahwa skripsi yang saya tulis ini tidak memuat karya atau bagian karya orang lain, kecuali yang telah disebutkan dalam kutipan dan daftar pustaka dengan mengikuti ketentuan sebagaimana layaknya karya ilmiah.

Apabila di kemudian hari ditemukan indikasi plagiarisme dalam naskah ini, saya bersedia menanggung segala sanksi sesuai peraturan perundang-undangan yang berlaku.

Yogyakarta, 10 Juli 2023

Penulis,



Guido Mario Enno Epriliano

LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPERLUAN AKADEMIS

Yang bertanda tangan di bawah ini, saya mahasiswa Universitas Sanata Dharma:

Nama : Guido Mario Enno Epriliano

NIM : 195314006

Demi perkembangan ilmu pengetahuan, saya memberikan kepada Perpustakaan Universitas Sanata Dharma karya ilmiah saya yang berjudul:

“ANALISIS SENTIMENT ULASAN PENGGUNA AGODA DAN TRAVELOKA MENGGUNAKAN SUPPORT VECTOR MACHINE DENGAN DAN TANPA QUERY EXPANSION RANKING”

Dengan demikian saya memberikan hak kepada Perpustakaan Universitas Sanata Dharma baik untuk menyimpan, mengalihkan dalam bentuk media lain, mengolah dalam bentuk pangkalan data, mendistribusikan secara terbatas, dan mempublikasikannya di internet atau media lain untuk kepentingan akademis tanpa perlu meminta izin dari saya atau memberikan royalti kepada saya selama tetap mencantumkan nama saya sebagai penulis.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di Yogyakarta

Pada tanggal: 10 Juli 2023

Yang menyatakan,



Guido Mario Enno Epriliano

KATA PENGANTAR

Puji syukur dan terima kasih kepada Tuhan yang Maha Esa atas Rahmat dan kasih karunia-Nya penulis dapat menyelesaikan skripsi yang berjudul “Sentimen Analisis Ulasan Pengguna Agoda Dan Traveloka Menggunakan *Support vector machine* Dengan Dan Tanpa *Query expansion ranking*”. Skripsi ini ditulis sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer Program Studi Informatika, Fakultas Sains dan Teknologi, Universitas Sanata Dharma Yogyakarta. Penulis menyadari bahwa skripsi ini dapat selesai dengan baik berkat doa, dukungan, bimbingan, dan bantuan dari berbagai pihak. Untuk itu penulis ingin mengucapkan terima kasih kepada:

1. Tuhan Yang Maha Esa yang selalu memberikan berkat dan rahmat-Nya yang berlimpah kepada penulis sehingga dapat menyelesaikan skripsi ini.
2. Bapak Robertus Adi Nugroho, S.T., M.Eng. selaku Ketua Program Studi Informatika Fakultas Sains dan Teknologi Universitas Sanata Dharma Yogyakarta.
3. Bapak Ir. Drs. Haris Sriwindono, M.Kom., Ph.D. selaku Dekan Fakultas Sains dan Teknologi, Universitas Sanata Dharma Yogyakarta.
4. Ibu Dr. Ir. Ridowati Gunawan, S.Kom., M.T. selaku dosen pembimbing yang telah mengarahkan dan membimbing penulis dengan sepenuh hati dan kesabaran sehingga penulis dapat menyelesaikan skripsi dengan baik.
5. Seluruh dosen dan karyawan sekretariat Program Studi Informatika, Fakultas Sains dan Teknologi, Universitas Sanata Dharma yang telah membantu dan mendukung peneliti dalam menyelesaikan skripsi ini.
6. Ayah Yohanes Margono yang selalu memberikan kasih sayang, doa, dan dukungan kepada penulis selama proses pembuatan skripsi. Serta Mami Bernadeta Endrawati yang dari awal kuliah sampai sekarang sudah ada di surga yang selalu mendoakan penulis tanpa henti.
7. Kakak saya, Lucia Mauren Puspita Septiani yang membantu dan memberikan dukungan kepada penulis supaya segera menyelesaikan studi.

8. Keluarga besar Aloysius Sugiyo dan Petrus Ngatiman yang mendoakan dan mendukung penulis untuk menyelesaikan skripsi ini dengan tepat waktu.
9. Agatha Sekar Ayu Wiwied Widyasari yang telah membantu, menemani, mendengarkan keluh kesah, memberikan doa dan dukungan sejak awal penyusunan skripsi.
10. Teman- teman yang selalu menghibur, membantu, dan mendukung penulis selama mengerjakan skripsi yaitu Rian, Refi, Iwan, Andre, Cindy, dan Arum.
11. Sahabat- sahabat penulis, Visna, Bagas, Antok, Bagus, Vito, dan Acil yang selalu meluangkan waktu untuk menemani, mendukung, menghibur, dan mendengarkan keluh kesah penulis.
12. Semua pihak yang memberikan doa dan dukungan kepada penulis sehingga penulis dapat menyelesaikan skripsi dengan baik.

Penulis menyadari masih banyak kekurangan dalam penulisan skripsi ini. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun dari para pembaca guna menyempurnakan skripsi ini. Semoga skripsi ini bermanfaat bagi semua pihak yang membacanya.

Yogyakarta, 10 Juli 2023

Penulis



Guido Mario Enno Epriliano

ABSTRAK

Melalui banyaknya fitur yang ditawarkan agen perjalanan *online*, membuat masyarakat semakin dimudahkan dalam melakukan transaksi seperti memesan tiket akomodasi sampai memesan kamar hotel. Beberapa contoh agen perjalanan *online* yang sudah cukup eksis di Indonesia yaitu Agoda dan Traveloka. Dalam situs *PlayStore*, agoda sendiri sudah memiliki lebih dari 10 juta pengguna, sedangkan traveloka jauh lebih banyak yaitu 50 juta pengguna. Karena banyaknya pengguna yang sudah mengunduh kedua aplikasi tersebut, tentu banyak juga ulasan yang telah diberikan baik itu ulasan positif maupun ulasan negatif. Ulasan yang sudah ada sebelumnya sangat berpengaruh kepada pengguna baru yang akan menggunakan aplikasi agoda dan traveloka. Oleh karena itu, penelitian ini bertujuan untuk mengetahui sentimen terhadap ulasan pengguna agoda dan traveloka dalam situs *PlayStore*. Data ulasan yang didapatkan dari aplikasi agoda sebanyak 26.031 ulasan, sedangkan dari aplikasi traveloka sebanyak 130.332 ulasan. Data tersebut akan melewati beberapa proses diantaranya *preprocessing*, pelabelan data, pembagian data *training* dan *testing*, pembobotan kata menggunakan TF-IDF, seleksi fitur menggunakan *Query expansion ranking*, dan klasifikasi menggunakan metode *Support Vector Machine*. Kernel yang digunakan adalah kernel *linear*. Tujuan dari penelitian ini adalah membandingkan akurasi yang didapatkan dengan menggunakan seleksi fitur dan tanpa menggunakan seleksi fitur serta pengaruh penggunaan seleksi fitur *Query expansion ranking*. Jumlah fitur teratas yang digunakan adalah 1000, 1300, dan 1500. Hasil akurasi paling tinggi jika menggunakan seleksi fitur terdapat pada jumlah fitur 1000, 1300, dan 1500 pada aplikasi agoda sebesar 96,20% dan pada aplikasi traveloka sebesar 97,09%. Sedangkan akurasi tanpa menggunakan seleksi fitur pada aplikasi agoda sebesar 96,20% dan pada aplikasi traveloka sebesar 97,09%. Seleksi fitur *Query expansion ranking* tidak berpengaruh dalam peningkatan akurasi analisis sentimen pada ulasan aplikasi agoda dan traveloka.

Kata kunci : Agoda, Traveloka, *PlayStore*, *Query expansion ranking*, *Support Vector Machine*

ABSTRACT

Through the many features offered by online travel agents, it makes it easier for people to make transactions such as ordering accommodation tickets to booking hotel rooms. Some examples of online travel agents that already exist in Indonesia are Agoda and Traveloka. On the PlayStore website, Agoda itself already has more than 10 million users, while Traveloka has far more, namely 50 million users. Due to the large number of users who have downloaded the two applications, of course there are also many reviews that have been given, both positive and negative reviews. Previous reviews are very influential for new users who will use the Agoda and Traveloka applications. Therefore, this study aims to determine sentiment towards the reviews of Agoda and Traveloka users on the PlayStore website. The review data obtained from the Agoda application were 26,031 reviews, while from the Traveloka application there were 130,332 reviews. The data will go through several processes including preprocessing, data labeling, distribution of training and testing data, weighting of words using TF-IDF, feature selection using *Query Expansion Ranking*, and classification using the *Support Vector Machine* method. The kernel used is a linear kernel. The purpose of this study is to compare the accuracy obtained by using feature selection and without using feature selection and the effect of using *Query Expansion Ranking* feature selection. The highest number of features used are 1000, 1300 and 1500. The highest accuracy results when using feature selection are found in the number of features 1000, 1300 and 1500 on the Agoda application at 96.20% and on the Traveloka application at 97.09%. While accuracy without using feature selection on the Agoda application is 96.20% and on the Traveloka application is 97.09%. The selection of the *Query Expansion Ranking* feature has no effect on increasing the accuracy of sentiment analysis on Agoda and Traveloka application reviews.

Keywords: Agoda, Traveloka, *PlayStore*, *Query Expansion Ranking*, *Support Vector Machine*

DAFTAR ISI

SENTIMEN ANALISIS ULASAN PENGGUNA AGODA DAN TRAVELOKA MENGGUNAKAN <i>SUPPORT VECTOR MACHINE</i> DENGAN DAN TANPA <i>QUERY EXPANSION RANKING</i>	i
SENTIMENT ANALYSIS OF AGODA AND TRAVELOKA USER REVIEWS USING <i>SUPPORT VECTOR MACHINE</i> WITH AND WITHOUT <i>QUERY EXPANSION RANKING</i>	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN.....	Error! Bookmark not defined.
MOTTO DAN PERSEMBAHAN.....	v
PERNYATAAN KEASLIAN KARYA.....	vi
LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPERLUAN AKADEMIS.....	vii
KATA PENGANTAR	viii
ABSTRAK.....	x
ABSTRACT	xi
DAFTAR ISI	xii
DAFTAR GAMBAR	xiv
DAFTAR TABEL	xvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	4
1.3 Tujuan Penelitian.....	4
1.4 Manfaat Penelitian	5
1.5 Batasan Masalah	5
BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI.....	7
2.1 Tinjauan Pustaka.....	7
2.2 Landasan Teori	10
2.2.1 Analisis Sentimen.....	10
2.2.2 <i>Preprocessing</i>	11
2.2.4 Pelabelan Data	15
2.2.5 Pembagian Data Menggunakan K-Fold Cross Validation	16
2.2.6 TF-IDF	17
2.2.7 <i>Query expansion ranking</i>	18
2.2.8 <i>Support Vector Machine</i>	19

2.2.9	Confusion Matrix.....	22
BAB III METODOLOGI PENELITIAN.....		25
3.1	Sumber Data.....	25
3.2	Gambaran Umum Penelitian	27
3.2.1	Pengambilan Data	27
3.2.2	<i>Preprocessing</i>	29
3.2.4	Pelabelan Data.....	39
3.2.5	TF-IDF	40
3.2.6	Pembagian Data Training dan Data Testing Menggunakan <i>K-Fold Cross Validation</i>	52
3.2.7	<i>Query expansion ranking</i>	53
3.2.8	<i>Support Vector Machine</i>	59
BAB IV IMPLEMENTASI SISTEM DAN ANALISIS HASIL		70
4.1	Pengambilan Data.....	70
4.2	<i>Preprocessing</i>	72
4.2.1	<i>Case folding</i>	72
4.2.2	<i>Cleaning</i>	73
4.2.3	<i>Stopword Removal</i>	74
4.2.4	<i>Tokenization</i>	76
4.2.5	<i>Normalizations</i>	77
4.2.6	<i>Stemming</i>	79
4.4	Pelabelan Data.....	83
4.5	Pembagian Data <i>Testing</i> dan Data <i>Training</i>	86
4.6	Pembobotan TF-IDF	86
4.7	Impementasi Seleksi Fitur <i>Query expansion ranking</i>	88
4.8	Implementasi Analisis Sentimen <i>Support Vector Machine</i>	89
4.9	Implementasi Pengukuran Performa Menggunakan <i>Confusion Matrix</i> .	90
4.10	Analisis Hasil	92
BAB V KESIMPULAN DAN SARAN		xviii
5.1	Kesimpulan.....	xviii
5.2	Saran	xix
DAFTAR PUSTAKA		xx

DAFTAR GAMBAR

Gambar 2.1 SVM Berusaha Menemukan Hyperplane	20
Gambar 3.1 Gambaran Umum Penelitian.....	27
Gambar 3.2 Proses Pengambilan Data	28
Gambar 3.3 Alur <i>Preprocessing Data</i>	29
Gambar 3.4 Alur Proses Perhitungan TF-IDF	41
Gambar 3.5 Alur Proses Perhitungan <i>Query expansion ranking</i>	53
Gambar 3.6 Alur Proses <i>Support Vector Machine</i>	59
Gambar 4.1 Gambar <i>Source Code</i> Pengambilan Data Dengan google_play_scrapper	71
Gambar 4.2 Data Mentah Hasil <i>Scraping</i> Aplikasi Agoda.....	71
Gambar 4.3 Data Mentah Hasil <i>Scraping</i> Aplikasi Traveloka	71
Gambar 4.4 Source Code Tahap <i>Case Folding</i>	72
Gambar 4.5 Hasil <i>Case Folding</i> Ulasan Agoda.....	72
Gambar 4.6 Hasil <i>Case Folding</i> Ulasan Traveloka.....	73
Gambar 4.7 <i>Source Code</i> Tahap <i>Cleaning Data</i>	73
Gambar 4.8 <i>Source Code</i> Menghapus Index Kosong	73
Gambar 4.9 Hasil <i>Cleaning Data</i> Aplikasi Agoda.....	74
Gambar 4.10 Hasil <i>Cleaning Data</i> Aplikasi Traveloka	74
Gambar 4.11 <i>Source Code Stopword Removal</i>	75
Gambar 4.12 Hasil <i>Stopword</i> Aplikasi Agoda.....	75
Gambar 4.13 Hasil <i>Stopword</i> Aplikasi Traveloka	75
Gambar 4.14 <i>Source Code Tokenization</i>	76
Gambar 4.15 Hasil <i>Tokenization</i> Agoda.....	76
Gambar 4.16 Hasil <i>Tokenization</i> Traveloka	77
Gambar 4.17 <i>Source Code Normalizations</i>	78
Gambar 4.18 Isi kamus_alay.csv	78
Gambar 4.19 Hasil <i>Normalization</i> Pada Agoda.....	79
Gambar 4.20 Hasil <i>Normalization</i> Pada Traveloka	79
Gambar 4.21 Paket Python <i>Swifter</i> dan <i>Pysastrawi</i>	80
Gambar 4.22 Import Stemmerfactory dan Swifter.....	80

Gambar 4.23 <i>Source Sode stemming</i>	81
Gambar 4.24 Hasil <i>Stemming</i> Aplikasi Agoda	82
Gambar 4.25 Hasil <i>Stemming</i> Aplikasi Agoda.....	82
Gambar 4.26 <i>Source Code Translate Data</i>	83
Gambar 4.27 Library Vadersentimen	84
Gambar 4.28 <i>Polarity Score</i>	84
Gambar 4.29 Pemberian Label.....	84
Gambar 4.30 Menghapus Data Kosong.....	85
Gambar 4.31 Grafik Jumlah Label.....	85
Gambar 4.32 Reset Urutan Tabel.....	85
Gambar 4.33 Hasil Pelabelan.....	86
Gambar 4.34 <i>K-Fold Cross Validation</i>	86
Gambar 4.35 TF-IDF.....	87
Gambar 4.36 Kode Seleksi Fitur QER	89
Gambar 4.37 Kode SVM.....	90
Gambar 4.38 Kode <i>Confusion Matrix</i>	91
Gambar 4.39 Hasil <i>Confusion Matrix</i>	92
Gambar 4.40 Grafik Perbandingan Akurasi Pada Traveloka	99
Gambar 4.41 <i>Confusion Matrix</i> Akurasi Tertinggi Traveloka.....	101

DAFTAR TABEL

Tabel 2.1 Hasil Perbandingan Penelitian.....	10
Tabel 2.2 Contoh <i>Case folding</i>	12
Tabel 2.3 Contoh <i>Cleaning</i>	12
Tabel 2.4 Contoh <i>Stopword Removal</i>	13
Tabel 2.5 Contoh <i>Tokenizing</i>	13
Tabel 2.6 Contoh <i>Normalization</i>	14
Tabel 2.7 Contoh <i>Stemming</i>	14
Tabel 2.8 Pengujian Menggunakan <i>K-Fold Cross Validation</i>	17
Tabel 2.9 Rumus Kernel.....	21
Tabel 2.10 <i>Confusion Matrix</i>	22
Tabel 3.1 Contoh Ulasan Aplikasi Agoda.....	26
Tabel 3.2 Contoh Ulasan Aplikasi Traveloka.....	26
Tabel 3.3 Data Mentah Ulasan Agoda	28
Tabel 3.4 Data Mentah Ulasan Traveloka	29
Tabel 3.5 Proses <i>Case folding</i>	30
Tabel 3.6 Proses <i>Cleaning Data</i>	31
Tabel 3.7 Proses Hasil <i>Stopword Removal</i>	33
Tabel 3.8 Proses Hasil <i>Tokenizing</i>	34
Tabel 3.9 Contoh Kamus Normalisasi	35
Tabel 3.10 Proses Hasil <i>Normalization</i>	36
Tabel 3.11 Proses Hasil <i>Stemming</i>	37
Tabel 3.12 Proses <i>Translate Data</i>	38
Tabel 3.13 Proses Hasil Pelabelan Data.....	40
Tabel 3.14 Perhitungan TF Aplikasi Agoda.....	41
Tabel 3.15 Perhitungan TF Aplikasi Traveloka.....	43
Tabel 3.16 Perhitungan DF Aplikasi Agoda.....	44
Tabel 3.17 Perhitungan DF Aplikasi Traveloka	45
Tabel 3.18 Perhitungan IDF Aplikasi Agoda	47
Tabel 3.19 Perhitungan IDF Aplikasi Traveloka	48
Tabel 3.20 Perhitungan TF-IDF Aplikasi Agoda	49

Tabel 3.21 Perhitungan TF-IDF Aplikasi Traveloka	50
Tabel 3.22 Skenario Pengujian <i>K-Fold</i>	52
Tabel 3.23 Frekuensi Kemunculan Kata	54
Tabel 3.24 Hasil Perhitungan <i>scoref</i> Aplikasi Agoda.....	56
Tabel 3.25 Hasil Perhitungan <i>scoref</i> Aplikasi Traveloka	57
Tabel 3.26 Hasil Perhitungan <i>Kernel Linear</i>	61
Tabel 3.27 Hasil Perhitungan <i>Matrix Hessian</i>	61
Tabel 3.28 Perhitungan <i>Error Rate</i>	63
Tabel 3.29 Hasil Perhitungan <i>Delta Alpha</i>	63
Tabel 3.30 Perhitungan Nilai αi	64
Tabel 3.31 Perhitungan Iterasi Maksimum.....	64
Tabel 3.32 Nilai x^- dan x^+	65
Tabel 3.33 Perhitungan $K(X_i \cdot x^-)$	65
Tabel 3.34 Perhitungan $K(X_i \cdot x^+)$	66
Tabel 3.35 Hasil Perhitungan Nilai Bias	67
Tabel 3.36 Hasil Perhitungan $h(x)$	68
Tabel 3.37 Hasil Perhitungan $Sign(h(x))$	68
Tabel 3.38 Hasil Perhitungan <i>Confusion Matrix</i>	69
Tabel 4.1 Perbandingan Akurasi Tiap <i>K-Fold</i> Aplikasi Agoda.....	94
Tabel 4.2 Perbandingan Akurasi Tiap <i>K-Fold</i> Aplikasi Traveloka	95
Tabel 4.3 Perbandingan Akurasi Tiap K Aplikasi Agoda	97
Tabel 4.4 Perbandingan Akurasi Tiap K Aplikasi Traveloka.....	98

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di era yang serba canggih dan digital ini, teknologi yang berkembang dapat mengubah pola pikir dan perilaku manusia. Teknologi adalah keseluruhan dari metode yang secara rasional mengarah dan memiliki ciri – ciri efisiensi dalam setiap bidang kegiatan manusia (Elul dalam Miarso, 2007). Beberapa contoh perilaku manusia yang dipengaruhi akibat perkembangan teknologi adalah meringankan pekerjaan manusia dan kemudahan dalam melakukan transaksi serta pertukaran data. Salah satu contoh kemudahannya adalah pada zaman dahulu, masyarakat mungkin masih membeli tiket kereta, pesawat, maupun hotel dengan mendatangi loket atau hotelnya secara langsung. Namun dengan kemajuan teknologi yang sudah semakin canggih dan masyarakat dipermudah dengan adanya agen penyedia jasa perjalanan atau *Online Travel Agent (OTA)* sehingga tidak perlu datang secara langsung untuk membeli tiket atau memesan hotel. *Online travel agent* atau agen perjalanan *online* adalah bisnis jasa pemesanan tiket pesawat, pemesanan tiket kereta api, pemesanan akomodasi, dan sebagainya yang terkait dengan kebutuhan konsumen dalam melakukan perjalanan wisata. Pemesanan dan pembayarannya pun dilakukan secara *online*, tanpa tatap muka (OnlinePajak, 2020). Sudah banyak hotel, maskapai pesawat, dan kereta api yang terdaftar pada *online travel agent*. Beberapa contoh *online travel agent* adalah Agoda dan

Traveloka. Kedua agen tersebut banyak digunakan karena memiliki banyak fitur perjalanan dan mudah untuk digunakan.

Pada tahun 2019 hingga 2021, penjualan tiket secara *online* sangat menurun akibat pandemi covid-19, tetapi pada tahun 2022 ini karena pelonggaran PPKM yang terjadi di Indonesia membuat penjualan tiket transportasi naik menjadi 56 persen terutama di aplikasi Traveloka. Sepanjang April hingga pertengahan Mei 2022 Traveloka mencatatkan kenaikan trafik dan juga pemesanan transportasi selama libur lebaran. Seperti pemesanan tiket kereta api yang melonjak 56 persen, bus dan shuttle naik 51 persen dan pesawat 39 persen dibandingkan dengan periode yang sama tahun lalu (Tirto.id, 2022). Sedangkan untuk data penelusuran Agoda dari Januari 2022 hingga Mei 2022 menunjukkan pertumbuhan sebesar 102 persen dalam penelusuran masuk dan pertumbuhan sebesar 90 persen dalam pencarian keluar (BWHotelier, 2022). Menurut catatan di *Google PlayStore* hingga tulisan ini dibuat, aplikasi Agoda sendiri sudah diunduh sebanyak lebih dari 10 juta pengguna, sedangkan untuk Traveloka jauh lebih banyak yaitu lebih dari 50 juta pengguna (Google Play, 2022). Karena banyaknya pengguna yang telah mengunduh kedua aplikasi tersebut, tentu banyak juga yang telah memberikan ulasan, baik itu ulasan positif maupun ulasan negatif di *PlayStore*.

Komentar atau ulasan yang diberikan oleh pengguna dapat digunakan untuk mengukur analisis sentimen masyarakat terhadap aplikasi Agoda dan Traveloka. Analisis sentimen yang diambil dari aplikasi Agoda dan Traveloka ini juga dapat dimanfaatkan oleh pengguna lain yang akan

mengunduh dan menggunakan kedua aplikasi ini sebagai pedoman apakah aplikasi ini bagus dan cocok digunakan untuk memesan tiket atau hotel dengan baik atau tidak.

Komentar pada *PlayStore* untuk aplikasi Agoda dan Traveloka jumlahnya telalu banyak sehingga tidak mungkin melakukan proses klasifikasi secara manual. Oleh karena itu, penelitian ini akan membantu dalam proses klasifikasi menggunakan metode *Support vector machine* (SVM). *Support vector machine* (SVM) dikembangkan oleh Boser, Guyon, Vapnik, dan pertama kali dipresentasikan pada tahun 1992 di *Annual Workshop on Computational Learning Theory*. Konsep dasar SVM sebenarnya merupakan kombinasi harmonis dari teori-teori komputasi yang telah ada puluhan tahun sebelumnya, seperti *margin hyperplane* (Duda & Hart tahun 1973, Cover tahun 1965, Vapnik 1964, dsb.), kernel diperkenalkan oleh Aronszajn tahun 1950, dan demikian juga dengan konsep-konsep pendukung yang lain. Berbeda dengan strategi *neural network* yang berusaha mencari *hyperplane* pemisah antar kelas, SVM berusaha menemukan *hyperplane* yang terbaik pada *input space*. Prinsip dasar SVM adalah *linear classifier*, dan selanjutnya dikembangkan agar dapat bekerja pada problem non-*linear* (Nugroho et al., 2003). Prinsip dasar itulah yang membuat penelitian ini menggunakan metode SVM.

Proses klasifikasi pada penelitian ini juga menggunakan seleksi fitur yaitu *Query expansion ranking* (*QER*). *Query expansion ranking* (*QER*) adalah sebuah metode seleksi fitur yang terinspirasi dari metode *Query Expansion* yang berguna untuk meningkatkan kualitas *query* yang

dimasukkan oleh pengguna kemudian digabung dengan cara *probabilistic weighting* model untuk memberi skor pada setiap fitur (Parlar & Ozel, 2016). Teknik seleksi fitur sendiri digunakan untuk mengurangi jumlah fitur dalam proses klasifikasi dalam menentukan kelas target.

Penelitian ini dilakukan untuk mengetahui opini pengguna tentang layanan dan melihat respon pengguna di dalam kedua *online travel agent* tersebut. Untuk memudahkan pengelolaan dan analisis data ulasan, perlu dilakukan analisis sentimen.

1.2 Rumusan Masalah

Rumusan masalah yang diperoleh dari latar belakang adalah sebagai berikut:

- (1) Bagaimana perbandingan akurasi tingkat kepuasan pengguna berdasarkan ulasan pada *PlayStore* untuk aplikasi Agoda dan Traveloka?
- (2) Seberapa besar pengaruh penggunaan *Query expansion ranking* (QER) pada metode *Support Vector Machine* (SVM) terhadap ulasan Agoda dan Traveloka?

1.3 Tujuan Penelitian

Tujuan yang ingin dicapai dari penelitian ini sebagai berikut:

- (1) Untuk mengetahui perbandingan akurasi tingkat kepuasan pengguna melalui ulasan pada *PlayStore* untuk kedua aplikasi yaitu Agoda dan Traveloka.

- (2) Untuk mengetahui pengaruh penggunaan *Query expansion ranking* (QER) pada metode *Support Vector Machine* (SVM) terhadap ulasan di aplikasi Agoda dan Traveloka.

1.4 Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut:

- (1) Mengetahui performa metode *Support Vector Machine* menggunakan seleksi fitur *Query expansion ranking* dalam melakukan analisis sentimen pada aplikasi Agoda dan Traveloka berdasarkan ulasan pada *PlayStore*.
- (2) Hasil dari penelitian sentimen analisis ini diharapkan dapat membantu pengguna baru dalam memilih aplikasi yang cocok dan sesuai untuk melakukan pemesanan tiket atau hotel secara *online* berdasarkan perbandingan tingkat performa ulasan di *PlayStore* pengguna lain pada aplikasi Agoda dan Traveloka.
- (3) Hasil dari penelitian sentimen analisis ini diharapkan dapat menghasilkan informasi yang bisa digunakan oleh kedua *online travel agent* tersebut agar dapat meningkatkan performa layanan dan tingkat kepuasan pelanggan di masa mendatang.

1.5 Batasan Masalah

Adapun batasan masalah dalam penelitian ini sebagai berikut:

- (1) Data penelitian yang digunakan bersumber dari ulasan terhadap layanan *online travel agent* yaitu Agoda dan Traveloka
- (2) Data yang digunakan adalah ulasan yang berbahasa Indonesia.

- (3) Data penelitian yang digunakan merupakan ulasan pengguna yang diambil dari situs *Google PlayStore*.
- (4) Ulasan diklasifikasikan ke dalam dua kelas yaitu positif dan negatif.
- (5) Menggunakan metode *Vader* dalam pelabelan data ulasan.

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

2.1.1 Analisis Sentimen Calon Gubernur DKI Jakarta 2017 Di Twitter (Buntoro, 2017).

Penelitian ini dilakukan untuk melihat opini masyarakat yang mengandung sentimen positif, netral dan negatif tentang pemilihan calon gubernur DKI Jakarta tahun 2017. Data opini masyarakat diambil dari twitter sebanyak 300 tweet dalam bahasa Indonesia dengan kata kunci AHY, Ahok, dan Anies. Metode klasifikasi yang digunakan adalah Naïve Bayes *Classifier* (NBC) dan *Support vector machine* (SVM), lalu untuk menentukan *class* sentimen menggunakan Lexicon Based. Hasil dari penelitian ini didapatkan bahwa metode klasifikasi Naïve Bayes *Classifier* (NBC) mendapatkan akurasi tertinggi dengan nilai rata-rata 95% dibandingkan dengan metode *Support vector machine* (SVM) yang hanya mendapatkan akurasi dengan nilai rata-rata 90% untuk klasifikasi data AHY karena metode klasifikasi Naïve Bayes *Classifier* (NBC) cenderung lebih stabil karena berbasis probabilitas kemunculan kata dalam sebuah kalimat. Meskipun hasil akurasi yang didapatkan dari metode Naïve Bayes *Classifier* (NBC) cukup baik, tapi model yang dibangun masih melakukan sedikit kesalahan pada saat proses klasifikasi data yang pembagian sentimennya tidak

seimbang. Karena dengan menggunakan data yang tidak seimbang akan menyebabkan data *minority class* yang salah diklasifikasi sebagai data *majority class* (Kohavi, 1998). Pada akhirnya menjadikan selisih nilai menjadi besar.

2.1.2 Analisis Sentimen Tentang Opini Maskapai Penerbangan pada Dokumen Twitter Menggunakan Algoritme Support vector machine (SVM) (Pravina et al., 2019)

Penelitian ini dilakukan untuk melihat opini masyarakat dalam menyampaikan keluh kesah maupun apresiasinya tentang maskapai-maskapai penerbangan. Opini tersebut dibagi dalam dua kelas yaitu kelas positif dan kelas negatif. Algoritma yang digunakan adalah *Support vector machine* (SVM) dengan fitur Lexicon Based Features yang dapat menerima opini berbahasa lain selain Bahasa Indonesia (dalam penelitian ini digunakan Bahasa Inggris). Pengujian yang dilakukan adalah pengujian terhadap parameter SVM yaitu nilai learning rute (gamma), nilai C, iterasi maksimum, dan pengaruh implementasi Lexicon Based Features. Data yang digunakan diambil dari Twitter dengan data latih sebanyak 200 data dan data uji sebanyak 50 data. Hasil pengujian menggunakan nilai learning rate (gamma) sebesar 0,03 karena merupakan nilai yang paling optimal dan stabil dibanding nilai lainnya. Kemudian nilai C yang digunakan bernilai 10 karena terjadi peningkatan drastis terutama pada nilai f-measure. Lalu iterasi yang digunakan sebanyak 50 kali iterasi karena optimal dan

stabil. Tingkat akurasi yang didapatkan sebesar 40%, *precision* sebesar 40%, *recall* sebesar 100%, dan f-measure sebesar 57,14%. Nilai tersebut didapatkan dengan mengimplementasikan fitur Lexicon Based.

2.1.3 Analisis Sentimen Pariwisata di Kota Malang Menggunakan Metode Naïve Bayes dan Seleksi Fitur *Query expansion ranking*

(Fauzi & Adinugroho, 2018)

Penelitian ini dilakukan untuk menganalisis ulasan dari masyarakat tentang pariwisata Kota Malang sebagai destinasi. Ulasan tersebut diambil dari website TripAdvisor dan akan diklasifikasikan menjadi dua kelas yaitu positif dan negatif. Metode yang digunakan adalah Naïve Bayes dengan seleksi fitur *Query expansion ranking* untuk mengurangi jumlah fitur pada proses klasifikasi. Pengujian dilakukan dengan mengganti rasio seleksi fitur antara 0-100 menggunakan data latih sebanyak 200 data (100 data positif dan 100 data negatif) dan data uji sebanyak 30 dokumen. Seleksi fitur *Query expansion ranking* bekerja dengan baik pada metode Naïve Bayes dengan akurasi tertinggi sebesar 86,6% pada seleksi fitur 75%. Sedangkan untuk seleksi fitur 25%, 50%, dan 100% memiliki tingkat akurasi sebesar 83,3%.

Tabel 2.1 Hasil Perbandingan Penelitian

No	Peneliti	Algoritma	Topik	Keterangan
1	Buntoro G, 2017	Naïve Bayes Classifier (NBC) , Support vector machine (SVM)	Analisis Sentimen Calon Gubernur DKI Jakarta 2017 Di Twitter	Melihat opini masyarakat yang mengandung sentimen positif, netral, dan negatif pada twitter. Menggunakan metode NBC dan SVM. Akurasi yang didapatkan NBC lebih tinggi yaitu 95% dibandingkan SVM yang hanya 90% untuk klasifikasi data AHY.
2	Pravina A, Cholissodin I, Adikara P, 2019	Support vector machine (SVM), Lexicon Based Features	Analisis Sentimen Tentang Opini Maskapai Penerbangan pada Dokumen Twitter Menggunakan Algoritme Support vector machine (SVM)	Penelitian ini melihat opini masyarakat yang terbagi dalam dua kelas yaitu positif dan negatif tentang maskapai penerbangan. Menggunakan metode SVM dengan fitur Lexicon Based Features yang dapat menerima opini bahasa lain selain Bahasa Indonesia. Data diambil dari twitter dengan data latih 200 data dan data uji 50 data. Tingkat akurasi yang didapatkan sebesar 40%, precision sebesar 40%, recall sebesar 100%, dan f-measure sebesar 57,14%. Nilai tersebut didapatkan dengan mengimplementasikan fitur Lexicon Based.
3	Fauzi M, Adinugroho S, 2018	Naïve Bayes, Query expansion ranking	Analisis Sentimen Pariwisata di Kota Malang Menggunakan Metode Naive Bayes dan Seleksi Fitur Query expansion ranking	Penelitian ini melihat opini masyarakat tentang Kota Malang sebagai destinesi. Opini terbaagi menjadi dua kelas yaitu positif dan negatif dan data yang diambil dari website TripAdvisor. Menggunakan metode SVM dengan seleksi fitur QER. Menggunakan data latif sebanyak 200 data dan data uji 30 dokumen. akurasi yang didapatkan sebesar 86,6% pada seleksi fitur 75%

2.2 Landasan Teori

2.2.1 Analisis Sentimen

Analisis sentimen, juga disebut penambangan opini, 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 (Hirst et al., n.d.). Pada analisis sentimen dan

penambangan opini, biasanya terfokus pada opini yang mengekspresikan sentimen positif atau negatif.

Opini atau ulasan pengguna biasanya digunakan oleh pelaku bisnis yang dapat diambil dari website atau media sosial. Salah satu contoh adalah untuk melakukan market research atau riset pasar. Riset pasar digunakan untuk melihat opini pengguna apakah mereka menyukai produk yang ditawarkan atau tidak sehingga pelaku bisnis dapat mengambil keputusan yang tepat.

2.2.2 *Preprocessing*

Preprocessing adalah suatu proses pengubahan bentuk data yang belum terstruktur menjadi data yang terstruktur sesuai dengan kebutuhan untuk proses mining yang lebih lanjut atau singkatnya adalah merubah teks menjadi term index. *Preprocessing* diperlukan untuk memilih kata yang akan digunakan sebagai indeks. Indeks merupakan kata-kata yang mewakili dokumen yang nantinya digunakan untuk membuat pemodelan untuk aplikasi *text mining*. Tujuan dari *preprocessing* adalah menghasilkan sebuah *set term index* yang bisa mewakili dokumen (Fauzi, n.d.). Pada penelitian ini terdapat beberapa tahap *preprocessing* yaitu *Case folding*, *Cleaning*, *Stopword removal*, *Tokenization*, *Normalization*, *Stemming* dan *Translate Data*.

a) *Case folding*

Tahap ini dilakukan untuk mengubah seluruh dokumen teks yang menggunakan huruf kapital (*uppercase*) menjadi dokumen teks berbentuk standar yang biasanya menggunakan huruf kecil (*lowercase*).

Tabel 2.2 Contoh *Case folding*

Sebelum Case folding	“terdapat 2 aplikasi <i>Online Travel Agent</i> yang banyak digunakan, yaitu Agoda dan Traveloka”
Setelah Case folding	“terdapat 2 aplikasi <i>online travel agent</i> yang banyak digunakan yaitu agoda dan traveloka”

b) *Cleaning*

Tahap selanjutnya adalah *cleaning*. *Cleaning* data diperlukan untuk menghilangkan karakter-karakter yang tidak diperlukan untuk mengurangi *noise* seperti tanda baca, angka, URL, *emoticon*, simbol, dan lain sebagainya.

Tabel 2.3 Contoh *Cleaning*

Sebelum Cleaning	“terdapat 2 aplikasi <i>online travel agent</i> yang banyak digunakan yaitu agoda dan traveloka”
Setelah Cleaning	“terdapat aplikasi <i>online travel agent</i> yang banyak digunakan yaitu agoda dan traveloka”

c) *Stopword Removal*

Stopword adalah kata-kata yang mengandung sedikit informasi yang biasanya tidak diperlukan. Tujuan utama *Stopword removal* adalah untuk meningkatkan kecepatan eksekusi dan akurasi. *Stopword removal* tidak hanya

mengurangi ruang vektor tetapi juga membantu meningkatkan kinerja secara keseluruhan serta membantu untuk mengurangi ukuran teks (Buttar et al., 2018).

Tabel 2.4 Contoh *Stopword Removal*

Sebelum Stopword removal	“terdapat aplikasi <i>online travel agent</i> yang banyak digunakan yaitu agoda dan traveloka”
Setelah Stopword removal	“aplikasi <i>online travel agent</i> agoda traveloka”

d) *Tokenization*

Tahap *Tokenizing* adalah tahap pemotongan string input berdasarkan tiap kata yang menyusunnya. Karakter selain huruf akan dianggap delimiter dan akan dihilangkan atau dihapus untuk proses mendapat kata-kata penyusun teks (Riyani et al., 2019). Contoh *tokenizing* dapat dilihat pada tabel 2.5.

Tabel 2.5 Contoh *Tokenizing*

Sebelum Tokenizing	“aplikasi <i>online travel agent</i> agoda traveloka”
Setelah Tokenizing	“aplikasi”, “ <i>online</i> ”, ‘travel’, “agent”, “agoda”, “traveloka”

e) *Normalization*

Normalisasi dilakukan untuk menghilangkan perbedaan-perbedaan dangkal, karena ketika beberapa teks sangat mirip, analisis sentimen menganggap sebagai

pendapat yang berbeda. Operasi utama yang dilakukan pada fase normalisasi ini adalah mengkonversi huruf besar ke huruf kecil, menghapus aksen, menghapus spasi kosong, menghapus tanda hubung, menghapus tanda baca, dan menghilangkan *watermarks* (Sohrabi & Hemmatian, 2019).

Contoh *normalization* dapat dilihat pada tabel 2.6.

Tabel 2.6 Contoh *Normalization*

Sebelum Normalization	“aplikasi”, “online”, ‘travel’, “agent”, “agoda”, “traveloka”
Setelah Normalization	“aplikasi”, “online”, ‘travel’, “agent”, “agoda”, “traveloka”

f) *Stemming*

Stemming adalah proses yang memberikan pemetaan varian morfologis kata yang berbeda ke dalam kata dasar / umumnya (stem) (Tala, n.d.). Dengan kata lain *stemming* digunakan untuk mengubah kata imbuhan menjadi kata dasar. Contoh *stemming* dapat dilihat pada tabel 2.7.

Tabel 2.7 Contoh *Stemming*

Sebelum Stemming	“aplikasi”, “online”, ‘travel’, “agent”, “agoda”, “traveloka”
Setelah Stemming	“aplikasi”, “online”, ‘travel’, “agent”, “agoda”, “traveloka”

g) *Translate Data*

Translate Data atau menerjemahkan data ke dalam bahasa Inggris diperlukan agar data dapat masuk ke dalam proses selanjutnya yaitu pelabelan data. Proses *translating data* menggunakan library milik python yaitu *googletrans* dengan mengimport modul *Translator*.

2.2.4 Pelabelan Data

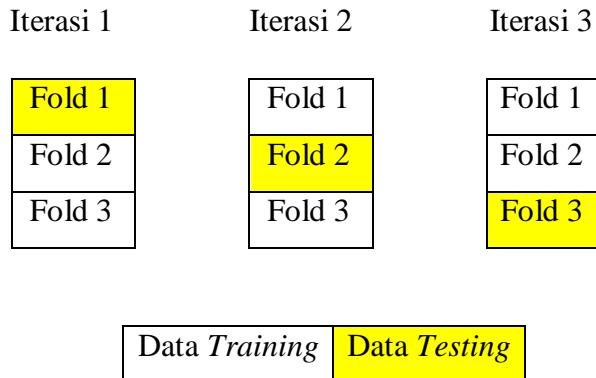
Data yang sudah diterjemahkan ke dalam bahasa Inggris kemudian akan diberi label per kalimat setiap barisnya. Dalam penelitian ini, label yang dipakai adalah label positif dan label negatif. Metode yang digunakan untuk memberi label adalah *Vader*. Proses pelabelan data menggunakan library milik python yaitu *Vadersentiment* dengan mengimport modul *SentimentIntensityAnalyzer*. Hasil yang diperoleh dari proses pelabelan berupa *polarity score*. Karena hasilnya masih berupa numerik atau skor, maka hasil tersebut harus diubah menjadi kata negatif atau positif. Untuk hasil atau skor yang nilainya diatas 0 maka akan diberi label positif, sedangkan untuk hasil yang dibawah 0 akan diberi label negatif. Jika terdapat skor yang bernilai 0 atau netral maka akan dihapus karena dalam penelitian ini, nilai netral tidak digunakan.

2.2.5 Pembagian Data Menggunakan *K-Fold Cross Validation*

Cross validation adalah teknik evaluasi model yang digunakan dalam *machine learning* untuk mengukur kinerja model pada data yang tidak terlihat. Pada *cross validation*, dataset dibagi menjadi dua bagian, yaitu data *training* dan data *testing*. Model dilatih pada data *training* dan kemudian diuji pada data *testing*. Proses ini diulang beberapa kali dengan membagi dataset ke dalam bagian yang berbeda-beda untuk memastikan bahwa model tidak hanya mengingat data *training* tetapi juga dapat digeneralisasi pada data yang tidak terlihat. *Cross validation* membantu menghindari *overfitting* dan memastikan bahwa model dapat digunakan pada data yang belum pernah dilihat sebelumnya.

K-fold cross validation adalah salah satu jenis *cross validation* yang umum digunakan. Pada *k-fold*, dataset dibagi menjadi *k-fold* atau bagian yang sama besar. Dalam setiap iterasi yang terjadi, satu lipatan (fold) digunakan sebagai data *testing*, sedangkan $k-1$ fold yang tersisa menjadi data *training*. Proses tersebut diulang sampai semua lipatan (fold) dievaluasi (Nti et al., 2021).

Sebagai contoh tentang *K-fold cross validation* dapat dilihat pada tabel 2.8. Contoh dari tabel 2.8 menggunakan $K = 3$ sehingga melakukan iterasi sebanyak 3 kali.

Tabel 2.8 Pengujian Menggunakan *K-Fold Cross Validation*

2.2.6 TF-IDF

Data yang sudah diberi label selanjutnya akan masuk ke dalam proses pembobotan kata menggunakan metode TF-IDF. *Term Frequency - Inverse Document Frequency* merupakan suatu algoritma yang mengalikan antara *Term Frequency* dengan *Inverse Document Frequency*. *Term frequency* yaitu jumlah kemunculan sebuah term pada sebuah dokumen. *Inverse Document Frequency* yaitu pengurangan dominasi *term* yang sering muncul diberbagai dokumen, dengan memperhitungkan kebalikan frekuensi dokumen yang mengandung suatu kata. Rumus untuk TF-IDF: (Nurjannah et al., 2013).

$$tf = 0,5 + 0,5 \times \frac{tf}{\max(tf)} \quad (2.1)$$

$$idf_t = \log\left(\frac{D}{df_t}\right) \quad (2.2)$$

$$W_{d,t} = tf_{d,t} \times IDF_{d,t} \quad (2.3)$$

Keterangan:

tf	= banyaknya kata yang dicari pada sebuah dokumen
$\max(tf)$	= jumlah kemunculan terbanyak term pada dokumen yang sama.
D	= total dokumen
df_t	= jumlah dokumen yang mengandung term t.
IDF	= Inversed Document Frequency ($\log_2(D/df)$)
d	= dokumen ke-d
t	= kata ke-t dari kata kunci
W	= bobot dokumen ke-d terhadap kata ke-t

2.2.7 *Query expansion ranking*

Query expansion ranking (QER) secara khusus dikembangkan untuk mengurangi dimensi ruang fitur dari masalah analisis sentimen. QER didasarkan pada metode *query expansion term weighting* yang digunakan untuk meningkatkan kinerja pencarian pada sistem pencarian informasi. Dalam metode pemilihan fitur QER, skor fitur dihitung sebelum fitur yang memiliki skor terendah dipilih dan digunakan dalam proses klasifikasi (Parlar et al., 2018). Persamaan pada QER sebagai berikut:

$$Score_f = \frac{p_f + q_f}{p_f - q_f} \quad (2.4)$$

$Score_f$ merupakan nilai QER dimana p_f adalah rasio dokumen positif yang mengandung fitur f dan q_f adalah rasio dari dokumen negatif yang mengandung fitur f .

Persamaan (2.4) dihitung berdasarkan persamaan (2.5) dan (2.6) berikut:

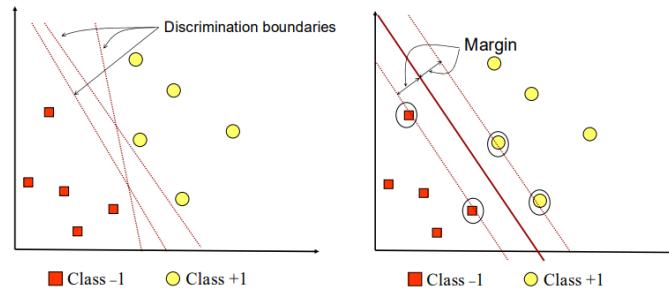
$$p_f = \frac{df_+^f + 0.5}{n^+ + 1.0} \quad (2.5)$$

$$q_f = \frac{df_-^f + 0.5}{n^- + 0.5} \quad (2.6)$$

Dimana df_+^f adalah jumlah dokumen positif yang mengandung fitur f dan df_-^f adalah jumlah dokumen negatif yang mengandung fitur f . n^+ dan n^- adalah jumlah dokumen dalam kelas positif dan negatif.

2.2.8 Support Vector Machine

Support Vector Machine diperkenalkan pertama kali oleh Vapnik pada tahun 1992 sebagai rangkaian harmonis konsep konsep unggulan dalam bidang *pattern recognition*. Secara sederhana, SVM adalah usaha mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah *class* pada input space. *Hyperplane* pemisah terbaik antara kedua *class* dapat ditemukan dengan mengukur margin *hyperplane* tersebut dan mencari titik maksimalnya. Margin adalah jarak antara *hyperplane* dengan data terdekat dari masing-masing *class* (Faihah, n.d.).



Gambar 2.1 SVM Berusaha Menemukan *Hyperplane*

Terbaik Yang Memisahkan Kedua Class -1
Dan +1. (Sumber : Nugroho et al., 2003)

Persamaan (2.7) digunakan untuk mencari margin *hyperplane*.

$$W \cdot X_i + b = 0 \quad (2.7)$$

Keterangan:

W = Bobot vector

X_i = Nilai atribut

b = Bias

Pada gambar 2.1 dapat dilihat bahwa terdapat lingkaran kuning dan kotak merah yang menandakan bahwa terdapat dua kelas yaitu -1 dan +1. +1 ditandai dengan lingkaran berwarna kuning, sedangkan untuk -1 ditandai dengan kotak berwarna merah. Warna kuning melambangkan kelas positif sedangkan merah melambangkan kelas negatif, sehingga diperoleh persamaan (2.8) dan (2.9).

$$W.X_i + b \geq +1 , jika y_i = +1 \quad (2.8)$$

$$W.X_i + b \leq -1 , jika y_i = -1 \quad (2.9)$$

Margin diminimalkan menggunakan rumus pada persamaan diatas dengan syarat menggunakan persamaan (2.10) dan (2.11).

$$y_i(w.x_i + b) \geq 1, i = 1, 2, \dots, N \quad (2.10)$$

$$y_i(w_1+x_i+w_2+x_i+b) \geq 1 \quad (2.11)$$

Berdasarkan dengan karakteristiknya metode SVM dibagi menjadi dua yaitu *linear* dan *nonlinear*, SVM *linear* merupakan data yang dipisahkan secara *linear* yaitu memisahkan dua kelas pada *hyperplane* dengan soft margin. Sedangkan *nonlinear* yaitu merupakan fungsi dari *kernel trick* terhadap ruang yang berdimensi tinggi. Permasalahan *nonlinear* dapat diatasi dengan memodifikasi *trick kernel* ke dalam SVM yang akan menjadi pemisah kelas atau *hyperplane* menjadi dua kelas didalam ruang vector dalam penelitian ini kernel yang akan digunakan adalah kernel *linear*. Seperti yang dapat dilihat persamaannya pada Tabel 2.9 di bawah ini (Rahman Isnain et al., 2021)

Tabel 2.9 Rumus Kernel

Jenis Kernel	Model
<i>Linear</i>	$K(x.x') = x.x'$
<i>Polynomial</i>	$K(x.x') = (x.x' + c)'$
<i>RBF Gaussian</i>	$K(x.x') = \exp(-\gamma x - x' ^2)$
<i>Sigmoid</i>	$K(x.x') = \tanh(\alpha x.x' + \beta)$

2.2.9 Confusion Matrix

Confusion matrix adalah sebuah tabel yang digunakan untuk mengevaluasi performa dari sebuah model klasifikasi dengan membandingkan prediksi model dengan nilai sebenarnya dari data yang diuji. *Confusion matrix* terdiri dari empat nilai yaitu *True positive* (TP) yaitu jika data bernilai positif dan diprediksi positif, *False positive* (FP) yaitu jika data bernilai negatif dan diprediksi positif, *True negative* (TN) yaitu jika data bernilai negatif dan diprediksi negatif, dan *False negative* (FN) yaitu jika data bernilai positif dan diprediksi negatif (Fawcett, 2006).

Tabel 2.10 Confusion Matrix

		Aktual	
Prediksi	Class	Positive	Negatif
	Positive	TP	FP
	Negatif	FN	TN

Dari nilai-nilai tersebut, dapat dihitung berbagai metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *F1-score*. *Confusion matrix* sangat penting dalam evaluasi performa model klasifikasi karena dapat memberikan informasi yang lebih detail tentang kekuatan dan kelemahan model tersebut.

2.2.9.1 Accuracy

Accuracy adalah ukuran seberapa akurat suatu sistem atau

model dalam memprediksi atau mengklasifikasikan data dengan membandingkan nilai label yang terdapat pada dataset dengan nilai label yang diprediksi (Duei Putri et al., 2022). Perhitungan *accuracy* dapat dilihat pada persamaan (2.12).

$$\text{Accuracy} = \frac{TP+FN}{TP+FP+TN+FN} \quad (2.12)$$

2.2.9.2 Precision

Precision adalah evaluasi kinerja pengambilan (klasifikasi) yang mengukur proporsi dokumen relevan yang diambil dari total dokumen yang diambil (Fawcett, 2006). Perhitungan *precision* dapat dilihat pada persamaan (2.13).

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2.13)$$

2.2.9.3 Recall

Recall adalah mengukur proporsi dari nilai positif yang sebenarnya yang benar diidentifikasi sebagai nilai positif (Naser & Alavi, n.d.). Perhitungan *recall* dapat dilihat pada persamaan (2.14).

$$\text{Recall} = \frac{TP}{TP+FN} \quad (2.14)$$

2.2.9.4 *F1-score*

F1-score menggabungkan *precision* dan *recall* dengan menghitung nilai rata-rata harmonik dari kedua metrik tersebut. Hal ini memberikan bobot yang seimbang antara presisi dan *recall*, dan *F1-score* akan mencapai nilai maksimum ketika *precision* dan *recall* memiliki nilai yang sama (Naser & Alavi, n.d.). Perhitungan *F1-score* dapat dilihat pada persamaan (2.15).

$$\textbf{F1-score} = \frac{2 * (\textit{Precision} * \textit{Recall})}{\textit{Precision} + \textit{Recall}} \quad (2.15)$$

BAB III

METODOLOGI PENELITIAN

Pada bab metodologi penelitian ini menjelaskan secara keseluruhan tentang tahapan-tahapan serta teknik yang digunakan. Tahap pertama pada penelitian ini yaitu pengambilan data. Data yang diambil merupakan data ulasan dari *Google PlayStore* untuk aplikasi Agoda dan Traveloka. Setelah data diambil selanjutnya masuk pada tahapan *preprocessing*. Data yang sudah melewati tahapan *preprocessing* data akan diberi label positif dan negatif menggunakan kamus *Vader Sentiment*. Setelah dilakukan pelabelan, masuk ke tahap pembagian data *training* dan data *testing*. Data *training* dan data *testing* dibagi menggunakan *K-fold cross validation*. Selanjutnya data tersebut akan dilakukan pembobotan kata menggunakan *Term Frequency - Inverse Document Frequency* (TF-IDF). Setelah dilakukan pembobotan TF-IDF, data akan diproses menjadi dua bagian yaitu menggunakan seleksi fitur dan tanpa menggunakan seleksi fitur. Seleksi fitur yang dipakai pada penelitian ini yaitu *Query expansion ranking*. Tahapan selanjutnya yaitu pemodelan menggunakan *Support vector machine (SVM)*. Hasil dari pemodelan tersebut akan di evaluasi berdasarkan akurasi tertinggi. Tahapan yang terakhir yaitu menganalisis kesimpulan dari metode penelitian yang digunakan.

3.1 Sumber Data

Data yang digunakan dalam penelitian ini adalah kumpulan data yang diambil dari ulasan *Google PlayStore* pada aplikasi Agoda dan Traveloka. Data diperoleh dengan cara *crawling* menggunakan bahasa pemrograman python. *Crawling* data pada python dilakukan dengan

menginstall *Google-Play-Scrapper*. *Google-Play-Scrapper* sendiri menyediakan API untuk membantu proses *crawling* pada *Google PlayStore*. Fungsi yang digunakan dalam melakukan *crawling* adalah *reviews*. Pada fungsi *reviews*, terdapat beberapa properti seperti id dari aplikasi Agoda maupun Traveloka, bahasa, negara, menyortir, jumlah ulasan yang akan diambil, dan rating dari ulasan yang diambil. Penelitian ini mengambil semua ulasan dari rating terendah sampai rating tertinggi. Ulasan yang digunakan hanya ulasan yang berbentuk teks dan bahasa yang digunakan berbahasa Indonesia. Contoh dari ulasan *Google PlayStore* dari aplikasi Agoda dan Traveloka dapat dilihat pada Tabel 3.1 dan Tabel 3.2.

Tabel 3.1 Contoh Ulasan Aplikasi Agoda

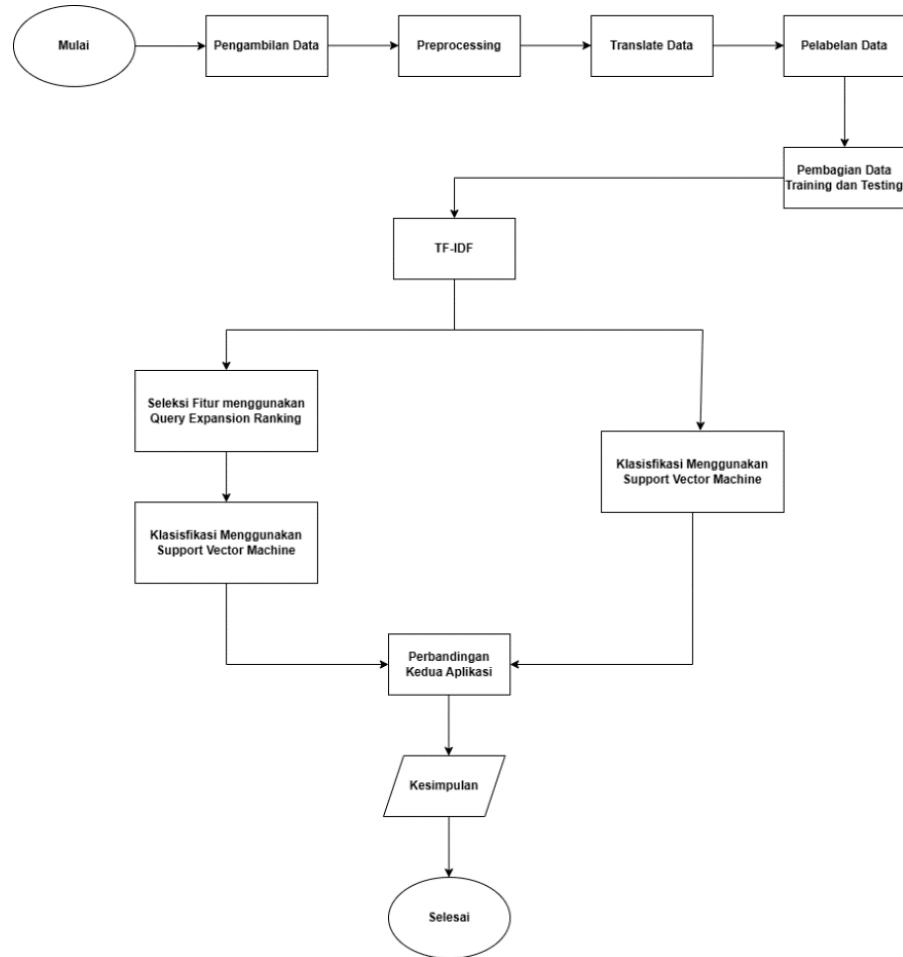
Ulasan
Yg terbaru payah.... Pembayaran cuman dan hanya kartu kredit... gak bisa pembayaran yg lain... Hubungi CS juga gak bisa... 41
Aplikasi Zonk.... Banyak masalah, buang waktu, buang pulsa, ternyata jd gak terima di hotelnya jadi bayar 2x... 106
Kenapa setelah di update.. tdk bs lagi di buka aplikasi nya.. Cek signal provider tdk ad masalah.. Please..help me 50
mantap,mudah dan banyak pilihan hotel dengan harga terbaik 273

Tabel 3.2 Contoh Ulasan Aplikasi Traveloka

Ulasan
Update terbaru banyak iklan pop up menutup layar bertubi menutup layar, apa2an ini.. kita hanya mau pesan hotel dgn cepat, itu sj... 25
Ini aplikasi se enak nya aj!kita beli tiket pesawat dengan satu kali perjalanan/tujuan, setelah kita bayar kita di nermalkan di kota transit dengan biaya sendiri!!!! 77
Sangat membanru dlm perjalanan saya yg sering traveling,, cepat tepat dan akurat. Semoga ttp jadi yg terbaik. 104
Apk tiba-tiba ga bisa dibuka, ngelag terus, sampai pakai WiFi pun tetap ga bisa, tolong di perbaiki lagi 117

3.2 Gambaran Umum Penelitian

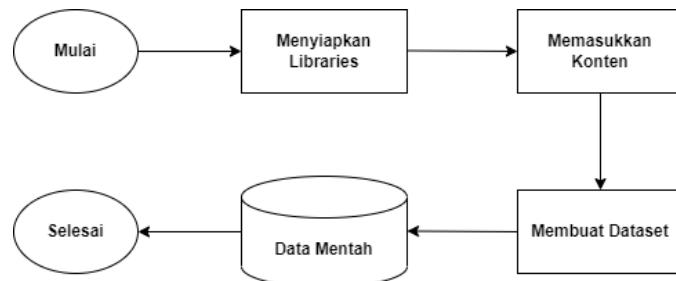
Gambaran umum penelitian menampilkan alur atau tahapan penelitian yang akan dikerjakan, mulai dari awal hingga akhir penelitian ini selesai. Gambar 3.1 menampilkan desain gambaran umum penelitian.



Gambar 3.1 Gambaran Umum Penelitian

3.2.1 Pengambilan Data

Pada tahap ini, pengambilan data ulasan pengguna dari aplikasi Agoda dan Traveloka yang terdapat pada *PlayStore*. Gambar 3.2 menggambarkan ilustrasi pengambilan data.



Gambar 3.2 Proses Pengambilan Data

Pengambilan data dilakukan dengan teknik *web crawling* menggunakan python. Data yang diambil merupakan data ulasan pengguna pada aplikasi Agoda dan Traveloka pada situs *Google PlayStore*. Data yang didapatkan masih berupa data mentah yang harus diolah lagi pada proses *Preprocessing*. Data mentah yang didapatkan dapat dilihat pada tabel 3.3 untuk aplikasi Agoda dan tabel 3.4 untuk aplikasi Traveloka.

Tabel 3.3 Data Mentah Ulasan Agoda

	,userName,content,score,at
0,budi ardiansah,"Sudah booking, dibayar lunas eh pas mau check in kata pegawai hotel nya gak bisa check in yg dari aplikasi ini, gimana cerita nya penginapan masih ada di aplikasi tapi koq malah gak bisa chek in pas udah waktu nya, malah disuruh nambah biaya sama pegawai nya dengan dalih harga aplikasi gak sesuai dengan harga booking langsung. Makin kesana makin kesini nih kayak nya, kalo memang sudah gak kerjasama atau gimana ya silahkan tidak dimasukan lagi untuk properti penginapan nya, bikin susah orang aja.",1,2022-10-23 07:57:29	
1,Ashfiya R,"Sebenarnya oke. Tp tampilan aplikasinya kenapa beda dengan tampilan web. Di aplikasi justru tidak ada info profil atau akun. Jadi susah kalo mau memperbaiki dan melihat nama akun, email atau nomer yg mungkin lupa kita gunakan . Semoga lebih diperbaiki lagi aplikasi nyaa",4,2022-10-27 06:50:31	
2,Sayogi Yogi,"Pembatalan oleh Agoda, REFUND sangat lama sekali prosesnya, sejak 3 okt 22 s.d hari ini TIDAK JELAS. Alasan gangguan proses pembayaran, namun dana DITERIMA AGODA. Refund belum dikembalikan dan hanya janji yg diterima tanpa melampirkan data. Sangat tidak memuaskan sekali, yg membatalkan AGODA yg repot pelanggannya.",1,2022-10-12 08:49:05	

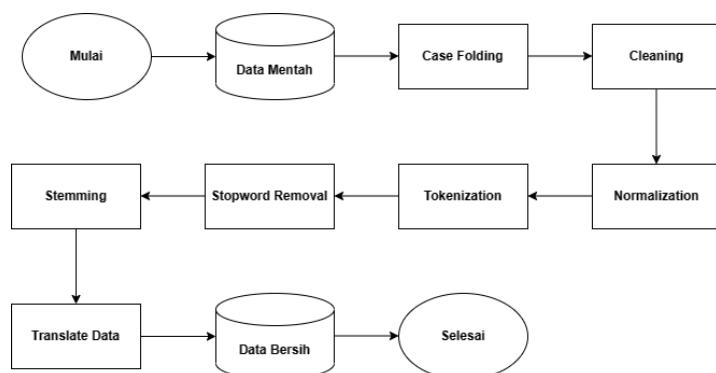
Tabel 3.4 Data Mentah Ulasan Traveloka

,content
0,"Kecewa bgt sih, parah, sama traveloka. Yg tadinya limit paylater 2,9juta di potong jadi 1,3 juta. Dengan alasan naik level trs emailnya masuk katanya pembayaran saya sering terlambat. Sy jatuh tempo tiap tgl 2. Tiap tgl 25 bulan sebelumnya itu sy byr dluan, jd ga prnh telat slama pake paylater. Tohh bukti2 pembayaran ada semua dr awal bayar ga prnh telat samsek. Tp alasan di potong krn itu. Yg masuk akal kek. Misal dlm kurun bbrp bln sy tdk melakukan perjalanan yah bisa2 saja. Dah kapok bgtðŸ“a"
1,"Tolong ya traveloka kalau mau nawarin paylater isi data yang sudah benar tapi kok masih di tidak valid kan, kalau emang gak ada fitur paylater nya mending di hapus fiturnya, karena sangat mengecewakan. Percuma banyak fitur tapi tidak bisa digunakan. Terima kasih."
2,"Kembalikan fitur aplikasi seperti dulu, ini ko eror terus klo pake wifi, buka web nya juga ga bisa, harus pake data, udh pake data malah ga ada koneksi, fitur pencarian juga ga ada"

Tidak semua atribut diatas digunakan dalam proses klasifikasi dalam penelitian ini, hanya atribut *content* yang digunakan.

3.2.2 *Preprocessing*

Pada proses *preprocessing* ini, data mentah yang sudah didapatkan saat proses *crawling* kemudian diolah melalui beberapa tahapan. Tahapan pada proses *preprocessing* adalah *Case folding*, *Cleaning*, *Stopword removal*, *Tokenization*, *Normalization*, dan *Stemming*.

**Gambar 3.3** Alur *Preprocessing* Data

Berikut merupakan penjelasan dan proses dalam melakukan tahap *preprocessing*.

1. Case folding

Tahap ini dilakukan untuk mengubah seluruh dokumen teks yang menggunakan huruf kapital (*uppercase*) menjadi dokumen teks berbentuk standar yang biasanya menggunakan huruf kecil (*lowercase*). Proses *Case folding* dapat dilihat pada tabel 3.5.

Tabel 3.5 Proses *Case Folding*

Aplikasi	Content	Case folding
Agoda	Yg terbaru payah.... Pembayaran cuman dan hanya kartu kredit... gak bisa pembayaran yg lain... Hubungi CS juga gak bisa...	yg terbaru payah.... pembayaran cuman dan hanya kartu kredit... gak bisa pembayaran yg lain... hubungi cs juga gak bisa...
	Aplikasi Zonk.... Banyak masalah, buang waktu, buang pulsa, ternyata jd gak terima di hotelnya jadi bayar 2x...	Aplikasi zonk.... banyak masalah, buang waktu, buang pulsa, ternyata jd gak terima di hotelnya jadi bayar 2x...
	Kenapa setelah di update.. tdk bs lagi di buka aplikasi nya.. Cek signal provider tdk ad masalah.. Please..help me	kenapa setelah di update.. tdk bs lagi di buka aplikasi nya.. cek signal provider tdk ad masalah.. please..help me
	mantap,mudah dan banyak pilihan hotel dengan harga terbaik	mantap,mudah dan banyak pilihan hotel dengan harga terbaik
	Agoda oke lah untuk jaringan hotelnya	agoda oke lah untuk jaringan hotelnya
Traveloka	Update terbaru banyak iklan pop up menutup layar bertubi menutup layar, apa2an ini.. kita hanya mau pesan hotel dgn cepat, itu sj...	update terbaru banyak iklan pop up menutup layar bertubi menutup layar, apa2an ini.. kita hanya mau pesan hotel dgn cepat, itu sj...
	Ini aplikasi se enak nya aj!kita beli tiket pesawat dengan satu	ini aplikasi se enak nya aj!kita beli tiket pesawat dengan satu kali

Tabel 3.5 Proses *Case Folding* (lanjutan)

Aplikasi	Content	Case folding
Traveloka	kali perjalanan/tujuan, setelah kita bayar kita di nermalkan di kota transit dengan biaya sendiri!!!!	perjalanan/tujuan, setelah kita bayar kita di nermalkan di kota transit dengan biaya sendiri!!!!
	Sangat membanru dlm perjalanan saya yg sering traveling,, cepat tepat dan akurat. Semoga ttp jadi yg terbaik.	sangat membanru dlm perjalanan saya yg sering traveling,, cepat tepat dan akurat. semoga ttp jadi yg terbaik.
	Apk tiba-tiba ga bisa dibuka, ngelag terus, sampai pakai WiFi pun tetap ga bisa, tolong di perbaiki lagi	apk tiba-tiba ga bisa dibuka, ngelag terus, sampai pakai WiFi pun tetap ga bisa, tolong di perbaiki lagi
	app yg mudah, banyak promo, nyaris ga ada bug, pertahankan dan tingkatkan trs..	app yg mudah, banyak promo, nyaris ga ada bug, pertahankan dan tingkatkan trs..

2. *Cleaning*

Tahap selanjutnya adalah *cleaning*. *Cleaning* data diperlukan untuk menghilangkan karakter-karakter yang tidak diperlukan untuk mengurangi *noise* seperti tanda baca, *emoticon*, *hashtag*, angka, simbol, dan lain sebagainya. Proses *cleaning* dapat dilihat pada tabel 3.6.

Tabel 3.6 Proses *Cleaning Data*.

Aplikasi	Case folding	Cleaning
Agoda	yg terbaru payah.... pembayaran cuman dan hanya kartu kredit... gak bisa pembayaran yg lain... hubungi cs juga gak bisa...	yg terbaru payah pembayaran cuman dan hanya kartu kredit gak bisa pembayaran yg lain hubungi cs juga gak bisa
	Aplikasi zonk.... banyak masalah, buang waktu, buang pulsa, ternyata jd gak terima di hotelnya jadi bayar 2x...	aplikasi zonk banyak masalah buang waktu buang pulsa ternyata jd gak terima di hotelnya jadi bayar
	kenapa setelah di update.. tdk bs lagi di buka aplikasi nya..	kenapa setelah di update tdk bs lagi di buka aplikasi nya cek signal

Tabel 3.6 Proses *Cleaning* Data (lanjutan).

Aplikasi	Case folding	Cleaning
Agoda	cek signal provider tdk ad masalah.. please..help me	provider tdk ad masalah pleasehelp me
	mantap,mudah dan banyak pilihan hotel dengan harga terbaik	mantapmudah dan banyak pilihan hotel dengan harga terbaik
	agoda oke lah untuk jaringan hotelnya	agoda oke lah untuk jaringan hotelnya
Traveloka	update terbaru banyak iklan pop up menutup layar bertubi menutup layar, apa2an ini.. kita hanya mau pesan hotel dgn cepat, itu sj...	update terbaru banyak iklan pop up menutup layar bertubi menutup layar apaan ini kita hanya mau pesan hotel dgn cepat itu sj
	ini aplikasi se enak nya aj!kita beli tiket pesawat dengan satu kali perjalanan/tujuan, setelah kita bayar kita di nermalkan di kota transit dengan biaya sendiri!!!!	ini aplikasi se enak nya ajkita beli tiket pesawat dengan satu kali perjalanan/ujuansetelah kita bayar kita di nermalkan di kota transit dengan biaya sendiri
	sangat membanru dlm perjalanan saya yg sering traveling,, cepat tepat dan akurat. semoga ttp jadi yg terbaik.	sangat membanru dlm perjalanan saya yg sering traveling cepat tepat dan akurat semoga ttp jadi yg terbaik
	apk tiba-tiba ga bisa dibuka, ngelag terus, sampai pakai wifi pun tetap ga bisa, tolong di perbaiki lagi	apk tibatiba ga bisa dibuka ngelag terus sampai pakai wifi pun tetap ga bisa tolong di perbaiki lagi
	app yg mudah, banyak promo, nyaris ga ada bug, pertahankan dan tingkatkan trs..	app yg mudah banyak promo nyaris ga ada bug pertahankan dan tingkatkan trs

3. *Stopword Removal*

Proses selanjutnya adalah *stopword removal*. *Stopword removal* sendiri digunakan untuk menghilangkan

kata-kata yang mengandung sedikit informasi yang biasanya tidak diperlukan. Tujuan dari proses ini adalah untuk meningkatkan kecepatan eksekusi dan akurasi. Pada penelitian ini, *Stopwords* yang digunakan adalah *stopwords* bahasa Indonesia. Beberapa kata ditambahkan dalam kamus *stopwords* yaitu ['eh', 'ia', 'nya', 'yg', 'yang', 'jnj', 'kl', 'nge', 'juga', 'jg', 'krn', 'karena', 'y', 'kan', 'atw', 'tks', 'agak', 'sih', 'kali', 'x', 'k'. Proses hasil *stopword removal* dapat dilihat pada tabel 3.7.

Tabel 3.7 Proses Hasil *Stopword Removal*.

Aplikasi	Cleaning	Stopword removal
Agoda	yg terbaru payah pembayaran cuman dan hanya kartu kredit gak bisa pembayaran yg lain hubungi cs juga gak bisa	terbaru payah pembayaran cuman kartu kredit gak pembayaran hubungi cs gak
	aplikasi zonk banyak masalah buang waktu buang pulsa ternyata juga gak terima di hotelnya jadi bayar	aplikasi zonk buang buang pulsa juga gak terima hotelnya bayar
	kenapa setelah di update tdk bs lagi dibuka aplikasi nya cek signal provider tdk ad masalah pleasehelp me	setelah update tdk bisa buka aplikasi cek signal provider tdk ada pleasehelp me
	mantapmudah dan banyak pilihan hotel dengan harga terbaik	mantapmudah pilihan hotel harga terbaik
	agoda oke lah untuk jaringan hotelnya	agoda oke jaringan hotelnya
Traveloka	update terbaru banyak iklan pop up menutup layar bertubi menutup layar apaan ini kita hanya mau pesan hotel dgn cepat itu sj	update terbaru iklan pop up menutup layar bertubi menutup layar pesan hotel dgn cepat sj
	ini aplikasi se enak nya ajkita beli tiket pesawat dengan satu kali perjalanan tujuan setelah kita bayar kita di nermalkan di kota	aplikasi enak ajkita beli tiket pesawat perjalanan tujuan setelah bayar nermalkan kota transit biaya

Tabel 3.7 Proses Hasil *Stopword Removal* (lanjutan).

Aplikasi	Cleaning	Stopword removal
Traveloka	transit dengan biaya sendiri	
	sangat membanru dlm perjalanan saya yg sering traveling cepat tepat dan akurat semoga ttp jadi yg terbaik	membanru dlm perjalanan traveling cepat akurat semoga ttp terbaik
	apk tibatiba ga bisa dibuka ngelag terus sampai pakai wifi pun tetap ga bisa tolong di perbaiki lagi	apk tibatiba ga dibuka ngelag pakai wifi ga tolong perbaiki
	app yg mudah banyak promo nyaris ga ada bug pertahankan dan tingkatkan trs	app mudah promo ga bug pertahankan tingkatkan trs

4. *Tokenizing*

Setelah dilakukan tahap *Stopword removal*, selanjutnya masuk ke tahap *tokenizing*. Proses *tokenizing* adalah proses pemotongan kalimat menjadi potongan kata-kata penyusun kalimat tersebut yang disebut *token*. Proses *tokenizing* terdapat pada tabel 3.8.

Tabel 3.8 Proses Hasil *Tokenizing*.

Aplikasi	Stopword removal	Tokenizing
Agoda	terbaru payah pembayaran cuman kartu kredit gak pembayaran hubungi cs gak	['terbaru', 'payah', 'pembayaran', 'cuman', 'kartu', 'kredit', 'gak', 'pembayaran', 'hubungi', 'cs', 'gak']
	aplikasi zonk buang buang pulsa jd gak terima hotelnya bayar	['aplikasi', 'zonk', 'buang', 'buang', 'pulsa', 'jd', 'gak', 'terima', 'hotelnya', 'bayar']
	setelah update tdk bs buka aplikasi cek signal provider tdk ad pleasehelp me	['setelah', 'update', 'tdk', 'bs', 'buka', 'aplikasi', 'cek', 'signal', 'provider', 'tdk', 'ad', 'pleasehelp', 'me']
	mantapmudah pilihan hotel harga terbaik	['mantapmudah', 'pilihan', 'hotel', 'harga', 'terbaik']

Tabel 3.8 Proses Hasil *Tokenizing* (lanjutan).

Aplikasi	Stopword removal	Tokenizing
Agoda	agoda oke jaringan hotelnya	['agoda', 'oke', 'jaringan', 'hotelnya']
Traveloka	update terbaru iklan pop up menutup layar bertubi menutup layar pesan hotel dgn cepat sj	['update', 'terbaru', 'iklan', 'pop', 'up', 'menutup', 'layar', 'bertubi', 'menutup', 'layar', 'pesan', 'hotel', 'dgn', 'cepat', 'sj']
	aplikasi enak ajkita beli tiket pesawat perjalanan tujuan setelah bayar nermalkan kota transit biaya	['aplikasi', 'enak', 'ajkita', 'beli', 'tiket', 'pesawat', 'perjalanan', 'tujuan', 'setelah', 'bayar', 'nermalkan', 'kota', 'transit', 'biaya']
	membanru dlm perjalanan traveling cepat akurat semoga ttp terbaik	['membanru', 'dlm', 'perjalanan', 'traveling', 'cepat', 'akurat', 'semoga', 'ttp', 'terbaik']
	apk tibatiba ga dibuka ngelag pakai wifi ga tolong perbaiki	['apk', 'tibatiba', 'ga', 'dibuka', 'ngelag', 'pakai', 'wifi', 'ga', 'tolong', 'perbaiki']
	app mudah promo ga bug pertahankan tingkatkan trs	['app', 'mudah', 'promo', 'ga', 'bug', 'pertahankan', 'tingkatkan', 'trs']

5. *Normalization*

Setelah melewati tahap *tokenization*, selanjutnya adalah tahap *normalization*. *Normalization* ini digunakan untuk memperbaiki ejaan kata agar sesuai dengan Kamus Besar Bahasa Indonesia (KBBI). Dalam penelitian ini, menggunakan file kamus_alay.csv sebagai kamus normalisasi. Tabel 3.9 merupakan contoh kamus normalisasi. Proses hasil *normalization* dapat dilihat pada tabel 3.10.

Tabel 3.9 Contoh Kamus *Normalisasi*

Kata	Normalization
Udh	Sudah
Gajelas	Tidak Jelas
Ja	Saja
Lg	Lagi
Gak	Tidak
Ga	Tidak

Tabel 3.9 Contoh Kamus Normalisasi (lanjutan)

Kata	Normalization
Sdh	Sudah
App	Aplikasi
Yg	Yang
Aja	Saja
Rb	Ribu
Dr	Dari

Tabel 3.10 Proses Hasil Normalization

Aplikasi	Tokenizing	Normalization
Agoda	['terbaru', 'payah', 'pembayaran', 'cuman', 'kartu', 'kredit', 'gak', 'pembayaran', 'hubungi', 'cs', 'gak']	['terbaru', 'payah', 'pembayaran', 'cuman', 'kartu', 'kredit', 'enggak', 'pembayaran', 'hubungi', 'cs', 'enggak']
	['aplikasi', 'zonk', 'buang', 'buang', 'pulsa', 'jg', 'gak', 'terima', 'hotelnya', 'bayar']	['aplikasi', 'zonk', 'buang', 'buang', 'pulsa', 'juga', 'enggak', 'terima', 'hotelnya', 'bayar']
	['seteleh', 'update', 'tdk', 'bs', 'buka', 'aplikasi', 'cek', 'signal', 'provider', 'tdk', 'ad', 'pleasehelp', 'me']	['seteleh', 'update', 'tidak', 'bisa', 'buka', 'aplikasi', 'cek', 'signal', 'provider', 'tidak', 'ada', 'pleasehelp', 'me']
	['mantapmudah', 'pilihan', 'hotel', 'harga', 'terbaik']	['mantapmudah', 'pilihan', 'hotel', 'harga', 'terbaik']
	['agoda', 'oke', 'jaringan', 'hotelnya']	['agoda', 'oke', 'jaringan', 'hotelnya']
Traveloka	['update', 'terbaru', 'iklan', 'pop', 'up', 'menutup', 'layar', 'bertubi', 'menutup', 'layar', 'pesan', 'hotel', 'dgn', 'cepat', 'sj']	['update', 'terbaru', 'iklan', 'pop', 'up', 'menutup', 'layar', 'bertubi', 'menutup', 'layar', 'pesan', 'hotel', 'dengan', 'cepat', 'saja']
	['aplikasi', 'enak', 'ajkita', 'beli', 'tiket', 'pesawat', 'perjalananujuansetelah', 'bayar', 'nermalkan', 'kota', 'transit', 'biaya']	['aplikasi', 'enak', 'ajkita', 'beli', 'tiket', 'pesawat', 'perjalananujuansetelah', 'bayar', 'nermalkan', 'kota', 'transit', 'biaya']
	['membanru', 'dlm', 'perjalanan', 'traveling', 'cepat', 'akurat', 'semoga', 'ttp', 'terbaik']	['membanru', 'dalam', 'perjalanan', 'traveling', 'cepat', 'akurat', 'semoga', 'tetap', 'terbaik']
	['apk', 'tibatiba', 'ga', 'dibuka', 'ngelag', 'pakai', 'wifi', 'ga', 'tolong', 'perbaiki']	['apk', 'tibatiba', 'enggak', 'dibuka', 'ngelag', 'pakai', 'wifi', 'enggak', 'tolong', 'perbaiki']

Tabel 3.10 Proses Hasil *Normalization* (lanjutan)

Aplikasi	Tokenizing	Normalization
Traveloka	['app', 'mudah', 'promo', 'ga', 'bug', 'pertahankan', 'tingkatkan', 'trs']	['app', 'mudah', 'promo', 'enggak', 'bug', 'pertahankan', 'tingkatkan', 'terus']

6. *Stemming*

Tahap terakhir dalam proses *preprocessing* yaitu tahap *stemming*. *Stemming* adalah proses mengubah kata imbuhan menjadi kata dasar atau baku, proses ini membutuhkan bantuan Kamus Besar Bahasa Indonesia (KBBI) yang dapat diakses di internet. Dalam penelitian ini menggunakan library Sastrawi untuk mengubah kata menjadi kata baku. Proses hasil *stemming* dapat dilihat pada tabel 3.11.

Tabel 3.11 Proses Hasil *Stemming*

Aplikasi	Normalization	Stemming
Agoda	['terbaru', 'payah', 'pembayaran', 'cuman', 'kartu', 'kredit', 'enggak', 'pembayaran', 'hubungi', 'cs', 'enggak']	['baru', 'payah', 'bayar', 'cuman', 'kartu', 'kredit', 'enggak', 'bayar', 'hubung', 'cs', 'enggak']
	['aplikasi', 'zonk', 'buang', 'buang', 'pulsa', 'juga', 'enggak', 'terima', 'hotelnya', 'bayar']	['aplikasi', 'zonk', 'buang', 'buang', 'pulsa', 'enggak', 'terima', 'hotel', 'bayar']
	['setelah', 'update', 'tidak', 'bisa', 'buka', 'aplikasi', 'cek', 'signal', 'provider', 'tidak', 'ada', 'pleasehelp', 'me']	['teh', 'update', 'tidak', 'bisa', 'buka', 'aplikasi', 'cek', 'signal', 'provider', 'tidak', 'ada', 'pleasehelp', 'me']
	['mantapmudah', 'pilihan', 'hotel', 'harga', 'terbaik']	['mantapmudah', 'pilih', 'hotel', 'harga', 'baik']
	['agoda', 'oke', 'jaringan', 'hotelnya']	['agoda', 'oke', 'jaring', 'hotel']
Traveloka	['update', 'terbaru', 'iklan', 'pop', 'up', 'menutup', 'layar', 'bertubi', 'menutup', 'layar', 'pesan', 'hotel', 'dengan', 'cepat', 'saja']	['update', 'baru', 'iklan', 'pop', 'up', 'tutup', 'layar', 'tubi', 'tutup', 'layar', 'pesan', 'hotel', 'dengan', 'cepat', 'saja']

Tabel 3.11 Proses Hasil *Stemming* (lanjutan)

Aplikasi	Normalization	Stemming
	['aplikasi', 'enak', 'ajkita', 'beli', 'tiket', 'pesawat', 'perjalananantujuansetelah', 'bayar', 'nermalkan', 'kota', 'transit', 'biaya']	['aplikasi', 'enak', 'ajkita', 'beli', 'tiket', 'pesawat', 'perjalananantujuansetelah', 'bayar', 'nermalkan', 'kota', 'transit', 'biaya']
	['membanru', 'dalam', 'perjalanan', 'traveling', 'cepat', 'akurat', 'semoga', 'tetap', 'terbaik']	['membanru', 'dalam', 'jalan', 'traveling', 'cepat', 'akurat', 'moga', 'tetap', 'baik']
	['apk', 'tibatiba', 'enggak', 'dibuka', 'ngelag', 'pakai', 'wifi', 'enggak', 'tolong', 'perbaiki']	['apk', 'tibatiba', 'enggak', 'buka', 'ngelag', 'pakai', 'wifi', 'enggak', 'tolong', 'baik']
	['app', 'mudah', 'promo', 'enggak', 'bug', 'pertahankan', 'tingkatkan', 'terus']	['app', 'mudah', 'promo', 'enggak', 'bug', 'tahan', 'tingkat', 'terus']

7. *Translate Data*

Setelah melewati tahap *stemming*, data kemudian diterjemahkan ke dalam bahasa Inggris supaya dapat diberi label secara otomatis menggunakan *Vader Sentiment*. Untuk melakukan *translate data*, membutuhkan library milik python yaitu *googletrans*. Proses *translate data* dapat dilihat pada tabel 3.12.

Tabel 3.12 Proses *Translate Data*

Aplikasi	Stemming	Translate
Agoda	['baru', 'payah', 'bayar', 'cuman', 'kartu', 'kredit', 'enggak', 'bayar', 'hubung', 'cs', 'enggak']	['new', 'bad', 'pay', 'only', 'card', 'credit', 'no', 'pay', 'connect', 'cs', 'nope']
	['aplikasi', 'zonk', 'buang', 'buang', 'pulsa', 'enggak', 'terima', 'hotel', 'bayar']	['app', 'zonk', 'throw away', 'throw away', 'pulse', 'no', 'accept', 'hotel', 'pay']

Tabel 3.12 Proses *Translate Data* (lanjutan)

Aplikasi	Stemming	Translate
	['teh', 'update', 'tidak', 'bisa', 'buka', 'aplikasi', 'cek', 'signal', 'provider', 'tidak', 'ada', 'pleasehelp', 'me']	['the', 'update', 'no', 'can', 'open', 'application', 'check', 'signal', 'provider', 'no', 'there', 'please help', 'me']
	['mantapmudah', 'pilih', 'hotel', 'harga', 'baik']	['stayeasy', 'select', 'hotel', 'price', 'good']
	['agoda', 'oke', 'jaring', 'hotel']	['agoda', 'okay', 'net', 'hotel']
Traveloka	['update', 'baru', 'iklan', 'pop', 'up', 'tutup', 'layar', 'tubi', 'tutup', 'layar', 'pesan', 'hotel', 'dengan', 'cepat', 'saja']	['update', 'new', 'advertisement', 'pop', 'up', 'close', 'screen', 'tubi', 'close', 'screen', 'message', 'hotel', 'with', 'fast', 'only']
	['aplikasi', 'enak', 'ajkita', 'beli', 'tiket', 'pesawat', 'perjalananujuansetelah', 'bayar', 'nermalkan', 'kota', 'transit', 'biaya']	['app', 'delicious', 'us', 'buy', 'ticket', 'plane', 'travel destination after', 'pay', 'normalize', 'city', 'transit', 'cost']
	['membanru', 'dalam', 'jalan', 'traveling', 'cepat', 'akurat', 'moga', 'tetap', 'baik']	['help', 'in', 'road', 'traveling', 'fast', 'accurate', 'hopefully', 'stay', 'good']
	['apk', 'tibatiba', 'enggak', 'buka', 'ngelag', 'pakai', 'wifi', 'enggak', 'tolong', 'baik']	['apk', 'suddenly', 'no', 'open', 'lags', 'use', 'wifi', 'no', 'help', 'ok']
	['app', 'mudah', 'promo', 'enggak', 'bug', 'tahan', 'tingkat', 'terus']	['app', 'easy', 'promo', 'nope', 'bug', 'hold', 'rate', 'continue']

3.2.4 Pelabelan Data

Setelah melewati tahap *preprocessing*, data ulasan tersebut belum memiliki label. Label diberikan supaya mengetahui apakah ulasan tersebut masuk ke dalam sentimen positif atau sentimen negatif. Proses pemberian label dilakukan menggunakan library *vadersentiment*. Hasil pelabelan berupa *score* dimana jika *score* dibawah nol maka sentimen tersebut bernilai negatif, sebaliknya

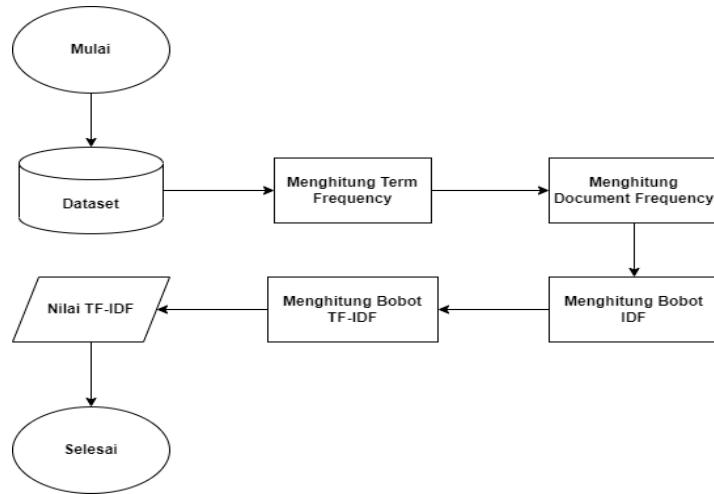
jika *score* bernilai diatas nol maka sentimen tersebut bernilai positif. Hasil pelabelan dapat dilihat pada tabel 3.13.

Tabel 3.13 Proses Hasil Pelabelan Data.

Aplikasi	Translate	Label
Agoda	['new', 'bad', 'pay', 'only', 'card', 'credit', 'no', 'pay', 'connect', 'cs', 'nope']	NEGATIF
	['app', 'zonk', 'throw away', 'throw away', 'pulse', 'no', 'accept', 'hotel', 'pay']	NEGATIF
	['the', 'update', 'no', 'can', 'open', 'application', 'check', 'signal', 'provider', 'no', 'there', 'please help', 'me']	NEGATIF
	['stayedeasy', 'select', 'hotel', 'price', 'good']	POSITIF
	['agoda', 'okay', 'net', 'hotel']	POSITIF
Traveloka	['update', 'new', 'advertisement', 'pop', 'up', 'close', 'screen', 'tubi', 'close', 'screen', 'message', 'hotel', 'with', 'fast', 'only']	NEGATIF
	['app', 'delicious', 'us', 'buy', 'ticket', 'plane', 'travel destination after', 'pay', 'normalize', 'city', 'transit', 'cost']	NEGATIF
	['help', 'in', 'road', 'traveling', 'fast', 'accurate', 'hopefully', 'stay', 'good']	POSITIF
	['apk', 'suddenly', 'no', 'open', 'lags', 'use', 'wifi', 'no', 'help', 'ok']	NEGATIF
	['app', 'easy', 'promo', 'nope', 'bug', 'hold', 'rate', 'continue']	POSITIF

3.2.5 TF-IDF

Setelah melewati proses *preprocessing*, dan pelabelan data, selanjutnya adalah proses pembobotan kata menggunakan metode TF-IDF. Gambar 3.4 merupakan alur proses perhitungan TF-IDF.



Gambar 3.4 Alur Proses Perhitungan TF-IDF

Dalam proses *Term Frequency-Inverse Document Frequency* (TF-IDF), *Term Frequency* berarti berapa kali suatu istilah diulang dalam sebuah teks, dan *Inverse Document Frequency* adalah singkatan dari IDF, sebuah algoritma yang digunakan untuk menghitung probabilitas terbalik untuk menemukan kata dalam sebuah teks. Persamaan 3.1 untuk menghitung nilai TF-IDF. (Dadgar et al., 2016)

$$W_{ij} = tf_{ij} * \log \frac{N}{df_i} \quad (3.1)$$

Dalam persamaan 3.1, W_{ij} adalah bobot dari kata i dalam dalam dokumen j , N adalah jumlah dokumen dalam himpunan total dokumen, tf_{ij} adalah frekuensi kata i dalam dokumen j , dan df_i adalah jumlah dokumen yang mengandung kata i . Perhitungan nilai tf untuk aplikasi Agoda dapat dilihat pada tabel 3.14, sedangkan untuk aplikasi Traveloka dapat dilihat pada tabel 3.15.

Tabel 3.14 Perhitungan TF Aplikasi Agoda

Kata	TF				
	D1	D2	D3	D4	D5
new	1	0	0	0	0
bad	1	0	0	0	0
pay	2	1	0	0	0
only	1	0	0	0	0

Tabel 3.14 Perhitungan TF Aplikasi Agoda (lanjutan)

Kata	TF				
	D1	D2	D3	D4	D5
card	1	0	0	0	0
credit	1	0	0	0	0
no	1	1	2	0	0
connect	1	0	0	0	0
cs	1	0	0	0	0
nope	1	0	0	0	0
app	0	1	0	0	0
zonk	0	1	0	0	0
throw	0	2	0	0	0
away	0	2	0	0	0
pulse	0	1	0	0	0
accept	0	1	0	0	0
hotel	0	1	0	1	1
the	0	0	1	0	0
update	0	0	1	0	0
can	0	0	1	0	0
open	0	0	1	0	0
application	0	0	1	0	0
check	0	0	1	0	0
signal	0	0	1	0	0
provider	0	0	1	0	0
there	0	0	1	0	0
pleasehelp	0	0	1	0	0
me	0	0	1	0	0
stayeasy	0	0	0	1	0
select	0	0	0	1	0
price	0	0	0	1	0
good	0	0	0	1	0
agoda	0	0	0	0	1
okay	0	0	0	0	1
net	0	0	0	0	1

Tabel 3.15 Perhitungan TF Aplikasi Traveloka

Kata	TF				
	D1	D2	D3	D4	D5
update	1	0	0	0	0
new	1	0	0	0	0
advertisement	1	0	0	0	0
pop	1	0	0	0	0
up	1	0	0	0	0
close	2	0	0	0	0
screen	2	0	0	0	0
tubi	1	0	0	0	0
message	1	0	0	0	0
hotel	1	0	0	0	0
with	1	0	0	0	0
fast	1	0	1	0	0
only	1	0	0	0	0
app	0	1	0	0	1
delicious	0	1	0	0	0
us	0	1	0	0	0
buy	0	1	0	0	0
ticket	0	1	0	0	0
plane	0	1	0	0	0
traveldestination after	0	1	0	0	0
pay	0	1	0	0	0
normalize	0	1	0	0	0
city	0	1	0	0	0
transit	0	1	0	0	0
cost	0	1	0	0	0
help	0	0	1	1	0
in	0	0	1	0	0
road	0	0	1	0	0
traveling	0	0	1	0	0
accurate	0	0	1	0	0
hopefully	0	0	1	0	0
stay	0	0	1	0	0
good	0	0	1	0	0
apk	0	0	0	1	0
suddenly	0	0	0	1	0
no	0	0	0	2	0

Tabel 3.15 Perhitungan TF Aplikasi Traveloka (lanjutan)

Kata	TF				
	D1	D2	D3	D4	D5
open	0	0	0	1	0
lags	0	0	0	1	0
use	0	0	0	1	0
wifi	0	0	0	1	0
ok	0	0	0	1	0
easy	0	0	0	0	1
promo	0	0	0	0	1
nope	0	0	0	0	1
bug	0	0	0	0	1
hold	0	0	0	0	1
rate	0	0	0	0	1
continue	0	0	0	0	1

Selanjutnya menghitung DF yaitu jumlah dokumen yang mengandung Term, minimal 1 dokumen. Perhitungan DF untuk aplikasi Agoda dapat dilihat pada tabel 3.16 dan untuk aplikasi Traveloka dapat dilihat pada tabel 3.17.

Tabel 3.16 Perhitungan DF Aplikasi Agoda

Kata	TF					DF
	D1	D2	D3	D4	D5	
new	1	0	0	0	0	1
bad	1	0	0	0	0	1
pay	2	1	0	0	0	2
only	1	0	0	0	0	1
card	1	0	0	0	0	1
credit	1	0	0	0	0	1
no	1	1	2	0	0	3
connect	1	0	0	0	0	1
cs	1	0	0	0	0	1
nope	1	0	0	0	0	1
app	0	1	0	0	0	1
zonk	0	1	0	0	0	1
throw	0	2	0	0	0	1
away	0	2	0	0	0	1

Tabel 3.16 Perhitungan DF Aplikasi Agoda (lanjutan)

Kata	TF					DF
	D1	D2	D3	D4	D5	
pulse	0	1	0	0	0	1
accept	0	1	0	0	0	1
hotel	0	1	0	1	1	3
the	0	0	1	0	0	1
update	0	0	1	0	0	1
can	0	0	1	0	0	1
open	0	0	1	0	0	1
application	0	0	1	0	0	1
check	0	0	1	0	0	1
signal	0	0	1	0	0	1
provider	0	0	1	0	0	1
there	0	0	1	0	0	1
pleasehelp	0	0	1	0	0	1
me	0	0	1	0	0	1
stayeasy	0	0	0	1	0	1
select	0	0	0	1	0	1
price	0	0	0	1	0	1
good	0	0	0	1	0	1
agoda	0	0	0	0	1	1
okay	0	0	0	0	1	1
net	0	0	0	0	1	1

Tabel 3.17 Perhitungan DF Aplikasi Traveloka

Kata	TF					DF
	D1	D2	D3	D4	D5	
update	1	0	0	0	0	1
new	1	0	0	0	0	1
advertisement	1	0	0	0	0	1
pop	1	0	0	0	0	1
up	1	0	0	0	0	1
close	2	0	0	0	0	1
screen	2	0	0	0	0	1
tubi	1	0	0	0	0	1
message	1	0	0	0	0	1
hotel	1	0	0	0	0	1

Tabel 3.17 Perhitungan DF Aplikasi Traveloka (lanjutan)

Kata	TF					DF
	D1	D2	D3	D4	D5	
with	1	0	0	0	0	1
fast	1	0	1	0	0	2
only	1	0	0	0	0	1
app	0	1	0	0	1	2
delicious	0	1	0	0	0	1
us	0	1	0	0	0	1
buy	0	1	0	0	0	1
ticket	0	1	0	0	0	1
plane	0	1	0	0	0	1
Traveldestination after	0	1	0	0	0	1
pay	0	1	0	0	0	1
normalize	0	1	0	0	0	1
city	0	1	0	0	0	1
transit	0	1	0	0	0	1
cost	0	1	0	0	0	1
help	0	0	1	1	0	2
in	0	0	1	0	0	1
road	0	0	1	0	0	1
traveling	0	0	1	0	0	1
accurate	0	0	1	0	0	1
hopefully	0	0	1	0	0	1
stay	0	0	1	0	0	1
good	0	0	1	0	0	1
apk	0	0	0	1	0	1
suddenly	0	0	0	1	0	1
no	0	0	0	2	0	1
open	0	0	0	1	0	1
lags	0	0	0	1	0	1
use	0	0	0	1	0	1
wifi	0	0	0	1	0	1
ok	0	0	0	1	0	1
easy	0	0	0	0	1	1
promo	0	0	0	0	1	1
nope	0	0	0	0	1	1
bug	0	0	0	0	1	1
hold	0	0	0	0	1	1
rate	0	0	0	0	1	1
continue	0	0	0	0	1	1

Dari perhitungan DF pada tabel 3.16 dan tabel 3.17, selanjutnya adalah menghitung nilai IDF menggunakan persamaan $\log \frac{N}{DF}$. Berikut ini perhitungan nilai IDF untuk tabel 3.18 pada aplikasi Agoda dan tabel 3.19 pada aplikasi Traveloka.

Tabel 3.18 Perhitungan IDF Aplikasi Agoda

Kata	TF					DF	N/DF	IDF
	D1	D2	D3	D4	D5			
new	1	0	0	0	0	1	5	0,699
bad	1	0	0	0	0	1	5	0,699
pay	2	1	0	0	0	2	2,5	0,398
only	1	0	0	0	0	1	5	0,699
card	1	0	0	0	0	1	5	0,699
credit	1	0	0	0	0	1	5	0,699
no	1	1	2	0	0	3	1,7	0,222
connect	1	0	0	0	0	1	5	0,699
cs	1	0	0	0	0	1	5	0,699
nope	1	0	0	0	0	1	5	0,699
app	0	1	0	0	0	1	5	0,699
zonk	0	1	0	0	0	1	5	0,699
throw	0	2	0	0	0	1	5	0,699
away	0	2	0	0	0	1	5	0,699
pulse	0	1	0	0	0	1	5	0,699
accept	0	1	0	0	0	1	5	0,699
hotel	0	1	0	1	1	3	1,7	0,222
the	0	0	1	0	0	1	5	0,699
update	0	0	1	0	0	1	5	0,699
can	0	0	1	0	0	1	5	0,699
open	0	0	1	0	0	1	5	0,699
application	0	0	1	0	0	1	5	0,699
check	0	0	1	0	0	1	5	0,699
signal	0	0	1	0	0	1	5	0,699
provider	0	0	1	0	0	1	5	0,699
there	0	0	1	0	0	1	5	0,699
pleasehelp	0	0	1	0	0	1	5	0,699
me	0	0	1	0	0	1	5	0,699
stayeasy	0	0	0	1	0	1	5	0,699
select	0	0	0	1	0	1	5	0,699
price	0	0	0	1	0	1	5	0,699

Tabel 3.18 Perhitungan IDF Aplikasi Agoda (lanjutan)

Kata	TF					DF	N/DF	IDF
	D1	D2	D3	D4	D5			
good	0	0	0	1	0	1	5	0,699
agoda	0	0	0	0	1	1	5	0,699
okay	0	0	0	0	1	1	5	0,699
net	0	0	0	0	1	1	5	0,699

Tabel 3.19 Perhitungan IDF Aplikasi Traveloka

Kata	TF					DF	N/DF	IDF
	D1	D2	D3	D4	D5			
update	1	0	0	0	0	1	5	0,699
new	1	0	0	0	0	1	5	0,699
advertisement	1	0	0	0	0	1	5	0,699
pop	1	0	0	0	0	1	5	0,699
up	1	0	0	0	0	1	5	0,699
close	2	0	0	0	0	1	5	0,699
screen	2	0	0	0	0	1	5	0,699
tubi	1	0	0	0	0	1	5	0,699
message	1	0	0	0	0	1	5	0,699
hotel	1	0	0	0	0	1	5	0,699
with	1	0	0	0	0	1	5	0,699
fast	1	0	1	0	0	2	2,5	0,398
only	1	0	0	0	0	1	5	0,699
app	0	1	0	0	1	2	2,5	0,398
delicious	0	1	0	0	0	1	5	0,699
us	0	1	0	0	0	1	5	0,699
buy	0	1	0	0	0	1	5	0,699
ticket	0	1	0	0	0	1	5	0,699
plane	0	1	0	0	0	1	5	0,699
traveldestination	0	1	0	0	0	1	5	0,699
after								
pay	0	1	0	0	0	1	5	0,699
normalize	0	1	0	0	0	1	5	0,699
city	0	1	0	0	0	1	5	0,699
transit	0	1	0	0	0	1	5	0,699
cost	0	1	0	0	0	1	5	0,699
help	0	0	1	1	0	2	2,5	0,398
in	0	0	1	0	0	1	5	0,699
road	0	0	1	0	0	1	5	0,699
traveling	0	0	1	0	0	1	5	0,699
accurate	0	0	1	0	0	1	5	0,699

Tabel 3.19 Perhitungan IDF Aplikasi Traveloka (lanjutan)

Kata	TF					DF	N/DF	IDF
	D1	D2	D3	D4	D5			
hopefully	0	0	1	0	0	1	5	0,699
stay	0	0	1	0	0	1	5	0,699
good	0	0	1	0	0	1	5	0,699
apk	0	0	0	1	0	1	5	0,699
suddenly	0	0	0	1	0	1	5	0,699
no	0	0	0	2	0	1	5	0,699
open	0	0	0	1	0	1	5	0,699
lags	0	0	0	1	0	1	5	0,699
use	0	0	0	1	0	1	5	0,699
wifi	0	0	0	1	0	1	5	0,699
ok	0	0	0	1	0	1	5	0,699
easy	0	0	0	0	1	1	5	0,699
promo	0	0	0	0	1	1	5	0,699
nope	0	0	0	0	1	1	5	0,699
bug	0	0	0	0	1	1	5	0,699
hold	0	0	0	0	1	1	5	0,699
rate	0	0	0	0	1	1	5	0,699
continue	0	0	0	0	1	1	5	0,699

Proses selanjutnya adalah menghitung TF-IDF. Proses menghitung TF-IDF hanya mengalikan nilai TF pada masing-masing dokumen dengan nilai IDF pada masing-masing kata. Perhitungan TF-IDF pada aplikasi Agoda dapat dilihat pada tabel 3.20 dan untuk aplikasi Traveloka dapat dilihat pada tabel 3.21.

Tabel 3.20 Perhitungan TF-IDF Aplikasi Agoda

Kata	TF-IDF				
	D1	D2	D3	D4	D5
new	0,699	0	0	0	0
bad	0,699	0	0	0	0
pay	0,796	0,398	0	0	0
only	0,699	0	0	0	0
card	0,699	0	0	0	0
credit	0,699	0	0	0	0
no	0,222	0,222	0,444	0	0
connect	0,699	0	0	0	0
cs	0,699	0	0	0	0

Tabel 3.20 Perhitungan TF-IDF Aplikasi Agoda (lanjutan)

Kata	TF-IDF				
	D1	D2	D3	D4	D5
nope	0,699	0	0	0	0
app	0	0,699	0	0	0
zonk	0	0,699	0	0	0
throw	0	1,398	0	0	0
away	0	1,398	0	0	0
pulse	0	0,699	0	0	0
accept	0	0,699	0	0	0
hotel	0	0,222	0	0,222	0,222
the	0	0	0,699	0	0
update	0	0	0,699	0	0
can	0	0	0,699	0	0
open	0	0	0,699	0	0
application	0	0	0,699	0	0
check	0	0	0,699	0	0
signal	0	0	0,699	0	0
provider	0	0	0,699	0	0
there	0	0	0,699	0	0
pleasehelp	0	0	0,699	0	0
me	0	0	0,699	0	0
stayeasy	0	0	0	0,699	0
select	0	0	0	0,699	0
price	0	0	0	0,699	0
good	0	0	0	0,699	0
agoda	0	0	0	0	0,699
okay	0	0	0	0	0,699
net	0	0	0	0	0,699

Tabel 3.21 Perhitungan TF-IDF Aplikasi Traveloka

Kata	TF-IDF				
	D1	D2	D3	D4	D5
update	0,699	0	0	0	0
new	0,699	0	0	0	0
advertisement	0,699	0	0	0	0
pop	0,699	0	0	0	0
up	0,699	0	0	0	0
close	1,398	0	0	0	0
screen	1,398	0	0	0	0
tubi	0,699	0	0	0	0

Tabel 3.21 Perhitungan TF-IDF Aplikasi Traveloka (lanjutan)

Kata	TF-IDF				
	D1	D2	D3	D4	D5
message	0,699	0	0	0	0
hotel	0,699	0	0	0	0
with	0,699	0	0	0	0
fast	0,398	0	0,398	0	0
only	0,699	0	0	0	0
app	0	0,398	0	0	0,398
delicious	0	0,699	0	0	0
us	0	0,699	0	0	0
buy	0	0,699	0	0	0
ticket	0	0,699	0	0	0
plane	0	0,699	0	0	0
Traveldestination after	0	0,699	0	0	0
pay	0	0,699	0	0	0
normalize	0	0,699	0	0	0
city	0	0,699	0	0	0
transit	0	0,699	0	0	0
cost	0	0,699	0	0	0
help	0	0	0,398	0,398	0
in	0	0	0,699	0	0
road	0	0	0,699	0	0
traveling	0	0	0,699	0	0
accurate	0	0	0,699	0	0
hopefully	0	0	0,699	0	0
stay	0	0	0,699	0	0
good	0	0	0,699	0	0
apk	0	0	0	0,699	0
suddenly	0	0	0	0,699	0
no	0	0	0	1,398	0
open	0	0	0	0,699	0
lags	0	0	0	0,699	0
use	0	0	0	0,699	0
wifi	0	0	0	0,699	0
ok	0	0	0	0,699	0
easy	0	0	0	0	0,699
promo	0	0	0	0	0,699
nope	0	0	0	0	0,699
bug	0	0	0	0	0,699
hold	0	0	0	0	0,699
rate	0	0	0	0	0,699
continue	0	0	0	0	0,699

3.2.6 Pembagian *Data Training* dan *Data Testing* Menggunakan *K-Fold Cross Validation*

Setelah melewati proses pembobotan kata, selanjutnya dilakukan pembagian data uji dan data latih menggunakan *K-fold*. Pada data sampel menggunakan nilai K=5. Tabel 3.22 merupakan skenario pengujian metode *K-fold*.

Tabel 3.22 Skenario Pengujian *K-Fold*

Fold 1	Doc 1	Doc 2	Doc 3	Doc 4	Doc 5
Fold 2	Doc 1	Doc 2	Doc 3	Doc 4	Doc 5
Fold 3	Doc 1	Doc 2	Doc 3	Doc 4	Doc 5
Fold 4	Doc 1	Doc 2	Doc 3	Doc 4	Doc 5
Fold 5	Doc 1	Doc 2	Doc 3	Doc 4	Doc 5

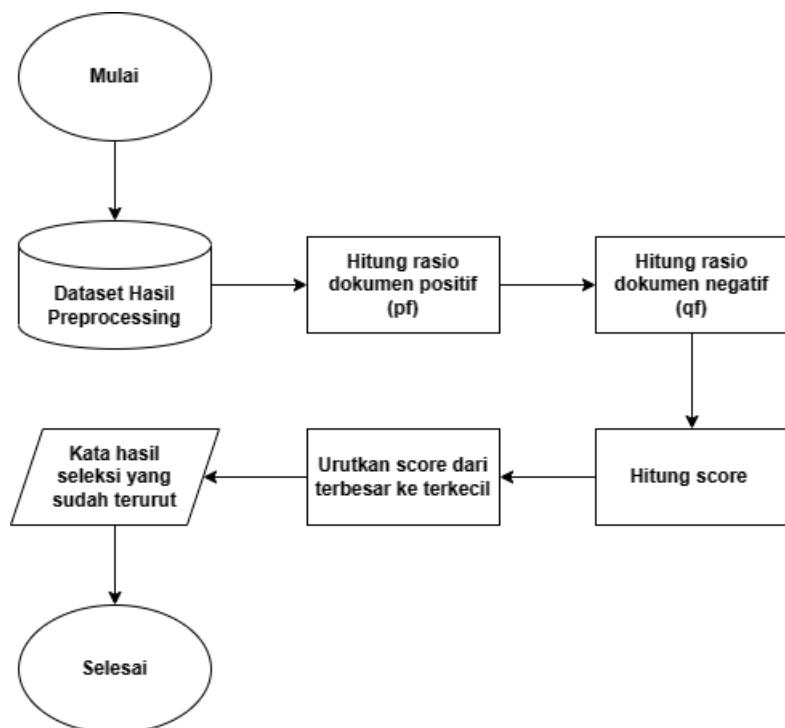
Berikut merupakan langkah-langkah yang yang dilakukan untuk pembagian data menggunakan metode *K-fold cross validation*:

1. Menentukan nilai k. Nilai k menentukan jumlah fold atau lipatan yang akan membagi dataset. Dalam contoh pada tabel 3.22, nilai k adalah 5.
2. Fold 1 pada Doc 1 menjadi *data testing*, sisanya menjadi *data training*.
3. Fold 2 pada Doc 2 menjadi *data testing*, sisanya menjadi *data training*.
4. Fold 3 pada Doc 3 menjadi *data testing*, sisanya menjadi *data training*.
5. Fold 4 pada Doc 4 menjadi *data testing*, sisanya menjadi *data training*.
6. Fold 5 pada Doc 5 menjadi *data testing*, sisanya menjadi *data training*.

7. Menghitung akurasi dari setiap *fold*. Kemudian semua akurasi dari tiap *fold* dijumlahkan dan dihitung nilai rata-ratanya.

3.2.7 *Query Expansion Ranking*

Seleksi fitur ini secara khusus dikembangkan untuk mengurangi dimensi ruang fitur dari masalah analisis sentimen. QER didasarkan pada metode *query expansion term weighting* yang digunakan untuk meningkatkan kinerja pencarian pada sistem pencarian informasi. Dalam metode seleksi fitur QER, skor fitur dihitung sebelum fitur yang memiliki skor terendah dipilih dan digunakan dalam proses klasifikasi (Parlar et al., 2018). Kata yang dipilih adalah kata dengan tertinggi, sedangkan untuk kata yang memiliki *score* rendah maka akan dihilangkan. Gambar 3.5 merupakan alur proses perhitungan *query expansion ranking*.



Gambar 3.5 Alur Proses Perhitungan *Query expansion ranking*

Persamaan pada QER sebagai berikut:

$$Score_f = \frac{p_f + q_f}{p_f - q_f} \quad (3.2)$$

$Score_f$ merupakan nilai QER dimana p_f adalah rasio dokumen positif yang mengandung fitur f dan q_f adalah rasio dari dokumen negatif yang mengandung fitur f .

Persamaan (3.2) dihitung berdasarkan persamaan (3.3) dan (3.4) berikut:

$$p_f = \frac{df_+^f + 0.5}{n^+ + 1.0} \quad (3.3)$$

$$q_f = \frac{df_-^f + 0.5}{n^- + 0.5} \quad (3.4)$$

dimana df_+^f adalah jumlah dokumen positif yang mengandung fitur f dan df_-^f adalah jumlah dokumen negatif yang mengandung fitur f . n^+ dan n^- adalah jumlah dokumen dalam kelas positif dan negatif. Sedangkan nilai 0,5 dan 1,0 digunakan untuk menghindari hasil bernilai 0. Sebagai contoh perhitungan untuk mencari nilai p_f dan q_f , menggunakan tabel 3.23.

Tabel 3.23 Frekuensi Kemunculan Kata

Kata	Dokumen	Kelas	Frekuensi
hotel	D1	NEGATIF	0
	D2	NEGATIF	1
	D3	NEGATIF	0
	D4	POSITIF	1
	D5	POSITIF	1

Dari tabel 3.23, dapat dihitung nilai pf dan df menggunakan persamaan (3.3) dan persamaan (3.4) sebagai berikut:

$$p_f = \frac{df_+^f + 0.5}{n^+ + 1.0}$$

$$p_f = \frac{2+0.5}{2+1.0}$$

$$p_f = \frac{2+0.5}{2+1.0} = 0.83$$

df_+^f merupakan banyaknya kata “hotel” pada dokumen positif, maka df_+^f bernilai 2. Sedangkan n^+ merupakan banyaknya dokumen pada kelas positif, maka n^+ bernilai 2.

Selanjutnya mencari nilai qf . qf merupakan probabilitas fitur f ditempatkan pada dokumen kelas negatif.

$$q_f = \frac{df_-^f + 0.5}{n^- + 0.5}$$

$$q_f = \frac{1+0.5}{3+0.5}$$

$$q_f = \frac{1+0.5}{3+0.5} = 0.43$$

df_-^f merupakan banyaknya kata “hotel” pada dokumen negatif, maka df_-^f bernilai 1. Sedangkan n^- merupakan banyaknya dokumen pada kelas positif, maka n^- bernilai 3.

Langkah selanjutnya adalah mencari persamaan nilai score atau $score_f$ menggunakan persamaan (3.2). Perhitungan mencari nilai score menggunakan perhitungan pf dan df diatas.

$$Score_f = \frac{p_f + q_f}{p_f - q_f}$$

$$Score_f = \frac{0.83 + 0.43}{0.83 - 0.43}$$

$$Score_f = \frac{0,83 + 0,43}{0,83 - 0,43} = \frac{1,26}{0,4} = 3,15$$

Langkah selanjutnya adalah merangkai semua perhitungan tiap kata dan dokumen menggunakan rumus diatas, didapatkan hasil pada tabel 3.24 untuk aplikasi Agoda dan tabel 3.25 untuk aplikasi Traveloka.

Tabel 3.24 Hasil Perhitungan $score_f$ Aplikasi Agoda

Kata	pf	qf	score	ranking
hotel	0,83	0,43	3,117647	1
new	0,17	0,43	2,272727	2
bad	0,17	0,43	2,272727	3
only	0,17	0,43	2,272727	4
card	0,17	0,43	2,272727	5
credit	0,17	0,43	2,272727	6
connect	0,17	0,43	2,272727	7
cs	0,17	0,43	2,272727	8
nope	0,17	0,43	2,272727	9
app	0,17	0,43	2,272727	10
zonk	0,17	0,43	2,272727	11
pulse	0,17	0,43	2,272727	12
accept	0,17	0,43	2,272727	13
the	0,17	0,43	2,272727	14
update	0,17	0,43	2,272727	15
can	0,17	0,43	2,272727	16
open	0,17	0,43	2,272727	17
application	0,17	0,43	2,272727	18
check	0,17	0,43	2,272727	19
signal	0,17	0,43	2,272727	20
provider	0,17	0,43	2,272727	21
there	0,17	0,43	2,272727	22
pleasehelp	0,17	0,43	2,272727	23
me	0,17	0,43	2,272727	24
stayedeasy	0,50	0,14	1,8	25
select	0,50	0,14	1,8	26

Tabel 3.24 Hasil Perhitungan *scoref* Aplikasi Agoda (lanjutan)

Kata	pf	qf	score	ranking
price	0,50	0,14	1,8	27
good	0,50	0,14	1,8	28
agoda	0,50	0,14	1,8	29
okay	0,50	0,14	1,8	30
net	0,50	0,14	1,8	31
pay	0,17	1,00	1,4	32
throw	0,17	0,71	1,608696	32
away	0,17	0,71	1,608696	33
no	0,17	1,29	1,297872	35

Tabel 3.25 Hasil Perhitungan *scoref* Aplikasi Traveloka

Kata	pf	qf	score	ranking
fast	0,50	0,43	13	1
app	0,50	0,43	13	2
help	0,50	0,43	13	3
update	0,17	0,43	2,272727	4
new	0,17	0,43	2,272727	5
advertisement	0,17	0,43	2,272727	6
pop	0,17	0,43	2,272727	7
up	0,17	0,43	2,272727	8
tubi	0,17	0,43	2,272727	9
message	0,17	0,43	2,272727	10
hotel	0,17	0,43	2,272727	11
with	0,17	0,43	2,272727	12
only	0,17	0,43	2,272727	13
delicious	0,17	0,43	2,272727	14
us	0,17	0,43	2,272727	15
buy	0,17	0,43	2,272727	16
ticket	0,17	0,43	2,272727	17
plane	0,17	0,43	2,272727	18
Traveldestination after	0,17	0,43	2,272727	19
pay	0,17	0,43	2,272727	20
normalize	0,17	0,43	2,272727	21
city	0,17	0,43	2,272727	22
transit	0,17	0,43	2,272727	23

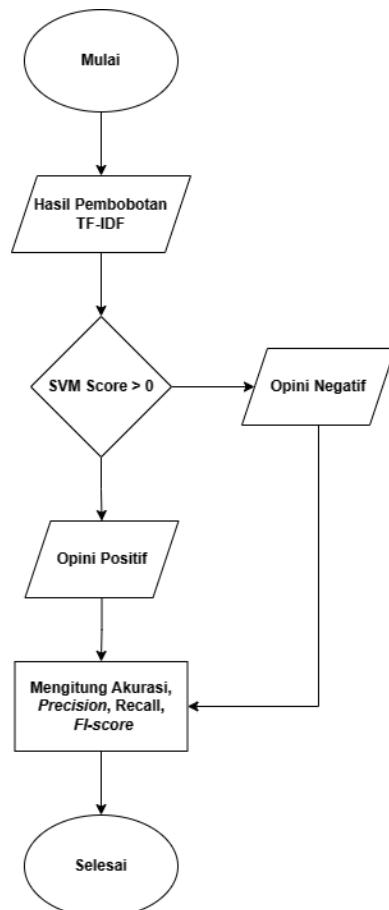
Tabel 3.25 Hasil Perhitungan *scoref* Aplikasi Traveloka (lanjutan)

Kata	pf	qf	score	ranking
cost	0,17	0,43	2,272727	24
apk	0,17	0,43	2,272727	25
suddenly	0,17	0,43	2,272727	26
open	0,17	0,43	2,272727	27
lags	0,17	0,43	2,272727	28
use	0,17	0,43	2,272727	29
wifi	0,17	0,43	2,272727	30
ok	0,17	0,43	2,272727	31
open	0,17	0,43	2,272727	32
lags	0,17	0,43	2,272727	33
use	0,17	0,43	2,272727	34
wifi	0,17	0,43	2,272727	35
ok	0,17	0,43	2,272727	36
in	0,50	0,14	1,8	37
road	0,50	0,14	1,8	38
traveling	0,50	0,14	1,8	39
accurate	0,50	0,14	1,8	40
hopefully	0,50	0,14	1,8	41
stay	0,50	0,14	1,8	42
good	0,50	0,14	1,8	43
easy	0,50	0,14	1,8	44
promo	0,50	0,14	1,8	45
nope	0,50	0,14	1,8	46
bug	0,50	0,14	1,8	47
hold	0,50	0,14	1,8	48
rate	0,50	0,14	1,8	49
continue	0,50	0,14	1,8	50
easy	0,50	0,14	1,8	51
promo	0,50	0,14	1,8	52
nope	0,50	0,14	1,8	53
bug	0,50	0,14	1,8	54
hold	0,50	0,14	1,8	55
rate	0,50	0,14	1,8	56
continue	0,50	0,14	1,8	57
close	0,17	0,71	1,608696	58
screen	0,17	0,71	1,608696	59
no	0,17	0,71	1,608696	60

Dari hasil pengurutan berdasarkan *ranking* pada Tabel 3.24 dan Tabel 3.25, jumlah fitur yang digunakan untuk proses selanjutnya dipilih berdasarkan rasio seleksi fitur yang ditetapkan. Rasio yang dipakai dalam penelitian ini adalah 10, 100, dan 1000.

3.2.8 *Support Vector Machine*

Pada penelitian ini, metode yang digunakan untuk melakukan analisis sentimen adalah *support vector machine*. Setelah melewati beberapa tahap yaitu *preprocessing*, pelabelan, pembagian data menjadi data *training* dan data *testing*, TF-IDF, dan *query expansion ranking*, selanjutnya masuk ke dalam tahap klasifikasi. Klasifikasi pada penelitian ini dibagi menjadi dua kelas yaitu kelas positif dan kelas negatif. Gambar 3.6 merupakan alur proses dari metode *support vector machine*.



Gambar 3.6 Alur Proses *Support Vector Machine*

Berikut ini merupakan contoh perhitungan *Support vector machine*.

1. Menentukan nilai variabel

Dalam melakukan perhitungan menggunakan *Support vector machine*, diperlukan nilai awal dari setiap variabel. Variabel yang diperlukan yaitu α_i , λ , γ , C , ε , dan i_{max} . Nilai awal dari setiap variabel yaitu:

$$\alpha_i = 0$$

$$\lambda = 0.5$$

$$\gamma = 0.0001$$

$$C = 1$$

$$\varepsilon = 0.0001$$

$$i_{max} = 3$$

2. Menghitung kernel *linear*

Dalam menghitung kernel *linear*, rumus yang dipakai terdapat pada persamaan (3.5). berikut contoh perhitungan menggunakan persamaan *linear* menggunakan tabel milik agoda.

$$K(x_i \cdot x_j) = x_i^T x_j \quad (3.5)$$

$$K(x_i x_j) = ((0,699 * 0,699) + \dots + (0 * 0))$$

$$K(x_i x_j) = 0,4761$$

Tabel 3.26 Hasil Perhitungan *Kernel Linear*

	D1	D2	D3
D1	0,4761	0	0
D2	0,4761	0	0
D3	0,6241	0,1521	0,1936
Sentimen	-1	-1	-1

3. Mengitung *matrix hessian*

Proses selanjutnya adalah menghitung *matrix hessian* dari data latih. Persamaan dari *matrix hessian*:

$$D_{ij} = Y_i Y_j (K(X_i \cdot X_i) + \lambda)^2 \quad (3.6)$$

Keterangan:

$Y_i Y_j$ = baris x kolom

$K(X_i \cdot X_i)$ = kernel

λ = 0,5

Berikut contoh perhitungannya:

$$D_{ij} = Y_i Y_j (K(X_i \cdot X_i) + \lambda)^2$$

$$D_{11} = 1 \cdot (-1)(0,4761 + 0,5)^2$$

$$D_{11} = -(0,7264)^2$$

$$D_{11} = -0,522771$$

Hasil perhitungan *matrix hessian* dapat dilihat pada tabel 3.27.

Tabel 3.27 Hasil Perhitungan *Matrix Hessian*

	D1	D2	D3
D1	0,952771	0	0
D2	0,952771	0	0
D3	1,263601	0,425234	0,48108096
Sentimen	-1	-1	-1

Perhitungan selanjutnya adalah mencari nilai Ei , αi , dan $\delta \alpha i$ menggunakan rumus pada persamaan berikut:

$$\text{a) } Ei = \sum \alpha_i N_i D_{ij} \quad (3.7)$$

$$\text{b) } \delta \alpha i = \min (\max [\gamma (1 - Ei) - \alpha i], C - \alpha i) \quad (3.8)$$

$$\text{c) } \alpha i = \alpha i + \delta \alpha \quad (3.9)$$

Keterangan:

Ei = error rate

D = matrix hessian

C = variabel slack

αi = variabel alpha

y = learning rate

$\delta \alpha$ = variabel tunggal

Mencari error rate menggunakan persamaan 3.7

Langkah-langkahnya sebagai berikut:

$$Ei = \sum \alpha_i N_i D_{ij}$$

$$E1 = (0,952771.0) + \dots + (1,460568 .0)$$

$$E1=0$$

Hasil perhitungan error rate pada data latih dapat dilihat pada tabel 3.28.

Tabel 3.28 Perhitungan *Error Rate*

	D1	D2	D3
E1	0	0	0
E2	1,905542	0	0
E3	3,811084	0	0
Sentimen	-1	-1	-1

Berikutnya adalah menghitung nilai *delta alpha* yang dilakukan pada iterasi maksimum pada data latih sebanyak 3. Perhitungan menggunakan persamaan 3.8 Berikut contoh perhitungan *delta alpha*.

$$\delta\alpha = \min(\max[\gamma(1 - E_i) - \alpha_i], C - \alpha_i)$$

$$\delta\alpha_i = \min(\max[\gamma(1 - E_i)\alpha_i], C - \alpha_i)$$

$$\delta\alpha_i = \min(\max[0.0001(1 - 1,905542)0], 1 - 0)$$

$$\delta\alpha_i = \min(\max[0.0001, 0], 1)$$

$$\delta\alpha_i = \min(0.0001, 1)$$

$$\delta\alpha_i = 0,0001$$

Tabel 3.29 Hasil Perhitungan *Delta Alpha*

$\delta\alpha_i$	D1	D2	D3
$\delta\alpha_1$	0.0001	0.0001	0.0001
$\delta\alpha_2$	-9,055	0.0001	0.0001
$\delta\alpha_3$	-0,0003	0.0001	0.0001
Sentimen	-1	-1	-1

Langkah selanjutnya adalah mencari nilai α_i menggunakan persamaan 3.9 Proses perhitungan dihentikan sampai iterasi mencapai nilai maks ($|\delta\alpha| < \epsilon$). Nilai dari $\epsilon = 0,0001$. Berikut adalah perhitungan nilai α_i .

$$\alpha_i = \delta\alpha_i + \alpha_i$$

$$\alpha_i = 0 + 0,0001$$

$$\alpha_i = 0,0001$$

Tabel 3.30 Perhitungan Nilai α_i

α_i	D1	D2	D3
α_0	0	0	0
α_1	0,0001	0,0001	0,0001
α_2	-9,0549	0,0002	0,0002
α_3	-9,0552	0,0003	0,0003
Sentimen	-1	-1	-1

Setelah menghitung *error rate*, delta aplha dan α_i , selanjutnya mencari nilai iterasi maksimum pada setiap hitungannya. Tabel 3.31 hasil iterasi maksimum.

Tabel 3.31 Perhitungan Iterasi Maksimum

$\delta\alpha_i$	D1	D2	D3
E3	0,0001	0,0001	0,0001
$\delta\alpha_3$	-0,0003	0,0001	0,0001
α_3	-9,0552	0,0003	0,0003
Sentimen	-1	-1	-1

Max(P)	0,0003
Max(N)	-9,0552

4. Mencari nilai bias

Nilai bias digunakan untuk melakukan klasifikasi. Nilai akhir α_i merupakan nilai *Support Vector*. Berikut persamaan untuk mencari nilai bias.

$$b = -\frac{1}{2} (\sum_{i=0}^n \alpha_i y_i K(X_i \cdot x) + (\sum_{i=0}^n \alpha_i y_i K(X_i \cdot x^+)) \quad (3.10)$$

Keterangan:

$$b = bias$$

$$K = kernel$$

$\alpha = \text{Lagrange Multiplier}$

$y = \text{kelas data}$

$Xi = \text{data latih}$

$x^- = \text{data kelas negatif dengan } \alpha \text{ negatif terbesar}$

$x^+ = \text{data kelas positif dengan } \alpha \text{ positif terbesar}$

Sebelum nencari nilai bias, terlebih dahulu mencari nilai x^- dan x^+ yang diperoleh dari nilai maksimum $\alpha i(x^-)$ pada kelas negatif dan nilai maksimum $\alpha i(x^+)$ pada kelas positif.

Tabel 3.32 Nilai x^- dan x^+

Max (P)	0,0003
Max (N)	-9,0552

Untuk mencari nilai $K(Xi .x^-)$ dan $K(Xi .x^+)$, dilakukan perhitungan normalisasi x^- dan x^+ . Pada tabel 3.33 berikut adalah hasil perhitungannya.

Tabel 3.33 Perhitungan $K(Xi .x^-)$

$K(Xi .x^-)$		
D1. x^-	D2. x^-	D3. x^-
-6,32958	0	0
-6,32958	0	0
-7,20794	-3,60397	0
-6,32958	0	0
-6,32958	0	0
-6,32958	0	0
-2,01025	-2,01025	-4,02051
-6,32958	0	0
-6,32958	0	0
-6,32958	0	0
0	-6,32958	0
0	-6,32958	0
0	-12,6592	0
0	-12,6592	0

Tabel 3.33 Perhitungan $K(X_i \cdot x^-)$ (lanjutan)

$K(X_i \cdot x^-)$		
D1. x^-	D2. x^-	D3. x^-
0	-6,32958	0
0	-6,32958	0
0	-2,01025	0
0	0	-6,32958
0	0	-6,32958
0	0	-6,32958
0	0	-6,32958
0	0	-6,32958
0	0	-6,32958
0	0	-6,32958
0	0	-6,32958
0	0	-6,32958
0	0	-6,32958
0	0	-6,32958
0	0	-6,32958
0	0	-6,32958
0	0	-6,32958
0	0	-6,32958
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
Jumlah	-59,8549	-58,2612
		-73,6459

Tabel 3.34 Perhitungan $K(X_i \cdot x^+)$

$K(X_i \cdot x^+)$		
D1. x^+	D1. x^+	D1. x^+
0,0002097	0	0
0,0002097	0	0
0,0002388	0,0001194	0
0,0002097	0	0
0,0002097	0	0
0,0002097	0	0
0,0000666	0,0000666	0,000133
0,0002097	0	0
0,0002097	0	0
0,0002097	0	0
0	0,0002097	0
0	0,0002097	0
0	0,0004194	0

Tabel 3.34 Perhitungan $K(X_i .x^+)$ (lanjutan)

$K(X_i .x^+)$		
D1. x^+	D1. x^+	D1. x^+
0	0,0004194	0
0	0,0002097	0
0	0,0002097	0
0	0,0000666	0
0	0	0,00021
0	0	0,00021
0	0	0,00021
0	0	0,00021
0	0	0,00021
0	0	0,00021
0	0	0,00021
0	0	0,00021
0	0	0,00021
0	0	0,00021
0	0	0,00021
0	0	0,00021
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
Jumlah	0,001983	0,0019302
		0,00244

Selanjutnya adalah mencari nilai bias menggunakan $(\Sigma_i^n =_0 \alpha_i \gamma_i K(X_i .x^-))$ dan $(\Sigma_i^n =_0 \alpha_i \gamma_i K(X_i .x^+))$ yang didapatkan dari nilai α_i dari kelas positif (x^+) dan negatif (x^-).

Berikut hasil perhitungan nilai bias pada tabel 3.35

Tabel 3.35 Hasil Perhitungan Nilai Bias

-541,998	0,0179565
0,017478	0,0000006
0,022094	0,0000007
Jumlah	-541,958
Nilai Bias	-541,9404702

5. Perhitungan hasil klasifikasi

Langkah selanjutnya adalah proses klasifikasi menggunakan data uji. Persamaan yang digunakan sebagai berikut:

$$h(x) = \sum_i^n =_0 \alpha_i \gamma_i K(X_i \cdot x) + b \quad (3.11)$$

Berikut merupakan hasil perhitungan menggunakan persamaan (3.11) dan hasilnya terdapat pada tabel 3.36.

Tabel 3.36 Hasil Perhitungan $h(x)$

	D1	D2	D3	H(x)
D4	0	0	0	-541,9404702
D5	0	0	0	-541,9404702

Terakhir untuk langkah klasifikasi nilai positif dan negatif menggunakan perhitungan $\text{Sign}(h(x))$. Jika nilai yang dihasilkan bernilai +1 maka data yang diujikan bernilai kelas positif, namun jika bernilai -1, maka data bernilai kelas negatif.

Tabel 3.37 Hasil Perhitungan $\text{Sign}(h(x))$

	D1	D2	D3	H(x)	Sign(h(x))	Label Kelas
D4	0	0	0	-541,9404702	-1	Negatif
D5	0	0	0	-541,9404702	-1	Negatif

6. Mengukur performa menggunakan *confusion matrix*

Setelah mendapatkan hasil pemodelan, langkah selanjutnya adalah melakukan pengukuran performa dari model yang terbentuk. Pengukuran performa dilakukan dengan melakukan perbandingan antara nilai aktual dan nilai prediksi. *Confusion matrix* digunakan untuk mengukur kinerja dengan membuat tabel berisi hasil pengukuran kinerja yang meliputi *Accuracy*,

Recall, *F1 score*, dan *Precision*. berikut hasil perhitungan *confusion matrix* pada tabel 3.38.

Tabel 3.38 Hasil Perhitungan *Confusion Matrix*

<i>Confusion matrix</i>		Hasil Prediksi	
		Prediksi Positif	Prediksi Negatif
Hasil Aktual	Aktual Positif	0	1
	Aktual Negatif	0	1
<i>Accuracy</i>		50,0%	
<i>Recall</i>		0%	
<i>Precision</i>		0%	
<i>F1 score</i>		0%	

BAB IV

IMPLEMENTASI SISTEM DAN ANALISIS HASIL

Pada bab ini akan membahas tentang proses yang dilakukan untuk melakukan analisis sentimen pada aplikasi Agoda dan Traveloka melalui aplikasi *PlayStore*. Hasil pengujian menggunakan metode *support vector machine* dengan seleksi fitur *query expansion ranking*. Kernel yang digunakan pada penelitian ini adalah kernel *linear* yang merupakan variasi kernel yang dimiliki oleh metode *support vector machine*.

4.1 Pengambilan Data

Pada penelitian ini, data yang digunakan adalah ulasan pengguna aplikasi Agoda dan Traveloka yang didapatkan dari situs *PlayStore*. Ulasan yang didapatkan dari aplikasi Agoda sebanyak 26.031 dan ulasan yang didapatkan dari aplikasi Traveloka sebanyak 130.332. Data tersebut diambil pada tanggal 5 Oktober 2022 dan hanya data yang berbahasa Indonesia yang diambil. Pengambilan data tersebut menggunakan teknik scrapping dengan menggunakan modul app dari library `google_play_scrapper` berdasarkan ID aplikasi Agoda dan Traveloka dan diurutkan berdasarkan relevansi. Berikut *source code* yang digunakan untuk melakukan scrapping dari situs *PlayStore* beserta hasilnya.

```

1 pip install google-play-scraper

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting google-play-scraper
  Downloading google_play_scraper-1.2.3-py3-none-any.whl (28 kB)
Installing collected packages: google-play-scraper
Successfully installed google-play-scraper-1.2.3

1 from google_play_scraper import app
2 import pandas as pd
3 import numpy as np

4 from google_play_scraper import Sort, reviews_all
5
6 us_reviews= reviews_all(
7     'com.traveloka.android', #ID aplikasi
8     lang='id', # defaults to 'en'
9     country='id', # defaults to 'us'
10    sort=Sort.MOST_RELEVANT, # defaults to Sort.MOST_RELEVANT
11
12
13 df_data = pd.DataFrame(np.array(us_reviews),columns=['review'])
14 df_data = df_data.join(pd.DataFrame(df_data.pop('review').tolist()))
15 df_data_spesific = pd.DataFrame(df_data, columns = ['content'])
16 df_data_spesific.to_csv('traveloka.csv')
17 files.download('traveloka.csv')

```

Gambar 4.1 Gambar Source Code Pengambilan Data Dengan google_play_scrapper

UserNaMe_content@coro-0	0.budi erdiansyah,"Situs booking dibayar lunas eh pas mau check in yg cari pagawai hotel nya gak bisa check in pas udah waktu nya, malah disuruh nambahin tgl 1,Astrijaya R,"Sebenarnya oke. To tampilan aplikasinya kerap beda dengan tampilan web. Di aplikasi justru tidak ada info profil atau akun. Jadi susah kalo memperbaiki dan melihat nama akun, email atau nomer yg mungkin lupa kita gunakan. Semoga le 2,Sayoga Yogi,"Pembatalan oleh Agoda. REURN sangat cepat dan selalu prosesnya, sejak 3-4 hari ini TIDAK JELAS. Alasan gangguan proses pembayaran, namun dana DITERIMA AGODA. Refund belum dikembalikan dan hanya janji yg diterima tanpa melar 3,Wina Setianingsih,Menggecewakan!!!! Agoda tunai tidak bisa di gunakan untuk memesan hotel dan tidak dapat di cairkan!!!! Aplikasi pulang bali. Dah paling bener pesen traveloka dñx Gak ada no tlf CS. Info pusat bantuan yg gak ada manfaatnya sa 4,Riki Pratama,"Kacau banget ini aplikasinya. Tampilannya sih oke, tapi harga yang ditampilkan sangat mengejutkan, saat pembayaran keluar harga aslinya. Akun juga gaga tbah 5,yudi ana,"Utuk masyarakat Indonesia yg tergiur harga murah diagoda, itu penipuan. Udah pesen tiba2 tercancel otomatis,pembayaran sudah berhasil,dan complain berkilau lagi ngga ada tindakan dan itikat buat ngembalikan dana yg sudah tertransfer. Cs tid 6,Fidi Bidan, sudah check out pesanan,lalu keluar ke kota karena pembatalan, saat pembayaran keluar akta alfaramt,saat ternyata voucher hokding tidak kosong di property nya, dan cara cek di aplikasi pesanan yg sudah bayar itu masuk me 7,Alfiti Wijaya,"Development sama pengelola nya kacau nh.. Gausah bikin ag/AGODA/webstite. Karna hanya ingin curang... Saya pesen hotel levat Agoda. Ternyata malah dr cancel dengan secara laenggan dari pihak Agoda. Kalau dr cancel sama saya sendiri 8,edward talutis,"Aplikasi yg buruk, pernah pesan kamari ku karena, tetapi tetap juga saldo kepotongan ampe 3x, dan saldo ku gk balikin lagi di pihak aplikasi, padahal tolk pernah cek di ke hotel yg bersangkutan. Mohon perbaiki sistemnya, minimal jgn mier 9,anto prabowo,Aplikasinya buruk dan menggecewakan. Sy pernah pesan tiket pesawat tapi setelah hari pembangkangan tiba2 batal dan minnt refund di bulan Maret sampe bulan September ini refund sy belum masuk katanya paling lama 3 bulan. Ini suda 10,Ntan 24,"Sangat membantu namun saya tertipi disaat pemesanan hotel di area bandung, hotel tersebut ternyata kosong sudah lama, tidak di gunakan namun akun nya masih aktif di agoda. Banyak orang2 yg tertipi dengan akun tersebut bukan har 11,Rudi Susanto,"Jangan coba2 pesen hotel atau tiket melalui aplikasi ini. Kalau tidak ingin ang lenyap. Pengalamank pesan hotel sudah transfer sesuai intruksi, begitu sudah transfer status di batalakan oleh aplikasi, dan sampai di hotel di tolak oleh pihak hote 12,Nisa Fathun,"Harga hotel tidak jessis dengan harga awal pemilihan kamarnya dan biaya detail tidak diiringin, yg dari tiket kreta kenapa yg keluar tiket bus dan travel . 2,2022-10-16 17:04:40 13,Andra Yanto,"AGODA SANGAT BURUK! Saya booking hotel untuk 2 malam (21-23 Oktober 2022) lama terbarter via transfer Bank Mandiri, di Aplikasi Agoda booking hotel sudah terkonfirmasi dan sudah diberikan 10 persenan, namun saat chek in tangg 14,Retro Kusumastuti,"Uang deposit yg dibayar tidak dikembalikan lagi yg diberikan oleh agoda. Namun yg benar dimana. Hotel juga tidak jelas. Minnt refund pun semua tidak bisa di proses. Sumpah KESEL bangegett, 15,Chandra Ramadhan,"Tgl 8 sy di prangk, agoda dan hotel yg pesan sampai hotel, di blng pentul, agoda janji relokasi dengan fasilitas yg sama dan harga yg sesuai, setelah 3 jam tidak ada hasil dan malah di relokasi dengan fasilitas home stay, yg fasilitasny 16,Elyna Rizka,"Saya pd saat ini menginap dihlm 1 hotel selama 2 hari,pas check inq receptionnya blg klg dpt break fastnya,aya seblm bayar perhatin dpt apa saja termasuk breakfast, kenyataannya agoda penipu besar dibleg gk ada!!!!!!!!!!!!!!hati2 deh bar 17,Nurfitri Afidiani,"Agoda ga jelas, pesen hotel include sarapan tapi pas cekin malah tanpa sarapan. Kata responisinya sama yg salah agodanya. Ruggiiiii, ga lagiiii, ga pakai agoda. Kalo adi bintang setengah tak kasih bintang setengah",1,2022-10-23 16:32:32 18,Pawon Pily,"Sangat sangat kecewa dengan Aplikasi ini, saya pesan 3x berturut-turut. Sy cek di aplikasi ini menginap di salah satu hotel di tlkt selama 2 malam, sy pesan dengan sarapan setelah konfirmasi pd pihak hotel ternyata zonk, krn ternyata hotel tsb tdk menyediakan 19,Gus Panji,"Tgl 15 booking hotel pembayaran melalui Alfamart, dibatalkan saatnya pihak Agoda. Jlhariyg sama pihak agoda yg close.,tau2nya tiba2 dapat notif pembayaran melalui CC Pad, bahanik yang terpesan 2 kamar padahal keterangannya hanya tersedia 1 k 20,Meg Shylla,"Tgl 15 booking hotel pembayaran melalui Indomaret, pihak agoda yg close.,sekarang udh tg 23 Oktober uang sy blm balik2 ..ngrusin refaudnya riet bg 21,Prasitria Dila,Pajak nya terlalu besar. Untungnya dpt diskon yg besar. Contoh : Dari harga permalam 145.000 menjadi 98.000. Itu padahal udh ongkos2 dengan bonusan lain yg tertera distu,,2,2022-10-24 08:35:12 22,Fadila Dwi Widayasti,"Uang sayang belon di prosesN juga, necccc di mbaning masih milih... ,tapi uda dapat emas klo pihak hotel maupun paket di apikasi ini... nyessel banget transaksi di apikasi ini... Dan go worth it bangetttt",1,2022-10-14 08:26:27 23,Juli okie,"Aplikasi penipuan ini, berkedok kebijakan kebijakan. Tidak ada pengembangan dan tanggung jawab baik dari pihak agoda nya. Niat ingin pesan biar cepat dan ga ribet malah zonk jadiinya. Sudah di bayar lunas pada saat per 24,zoldyck deraka,"Klau ada tanpa bintang dikasih tampa bintang.. Pikir 1000x klau pesan di aplikasi ini.. Terlalu berbelit2.. Jelas2 transaksi penambahan hari sudah kedebit dibilang belum ada transaksi.. Pelayannaa sangat buruk.. Email pertama menyebutkan 25,Rani Anjenia,"dear agoda kenapa aplikasi sayak tidak bisa di buka sekaliunya bisa ke buka gak lama kemudian langsung keluar gitu aja, udh di uninstal masih sama tolong perbaiki kendalanya",2,2022-10-08 17:46:56
-------------------------	--

Gambar 4.2 Data Mentah Hasil Scrapping Aplikasi Agoda

content	0,"Kecewa lgst slh, parah, sama traveloka. Yg tadinya limit paylater 2,9juta di potong ladi 1,3 juta. Dengan alasan naik level tsr emailnya masuk katanya pembayaran saya sering terlambat. Sy jatus tempo tiap tgl 2. Tiap tgl 25 bulan sebelumnya itu sy by duar 1,"Tolong ya traveloka kalau mau ntarin paylater isi data yang sudah benar tapi kok masih di tild valid kan, kalau emang gak ada fitur paylater nya mending di hapus fiturnya, karena sangat menggecewakan. Percuma banyak fitur tapi tidak bisa digunakan. 2,"Kembalikan fitur aplikasi seperti dulu, klo eror terus klo wifi, buka web nya juga gisa, harus pale data, udh pale data malah ga ada koneksi, fitur pencarian ga goa" 3,"Untuk pengalaman yang menggecewakan, untuk pengajuan refund dana atas pembatalan pembayaran. Tidak berkenan casti secara utuh... Seharusnya bentuk pengembalannya harus sesuai dengan dana yang di ajukan... Bukan berbentuk credit, voucher, 4,"Baca ni, serius, penting, pengalaman pribadi, kalo mesen butl perjalanan, mending pakai official nya aja, jangan pakai perantara klo traveloka gini, serius, kalo ada kendala yg tidak di inginkan saat perjalanan ntah pesawat dll. Lu mua klima ke traveloka mal 5,"Tolong ya traveloka ini selalu bdti tulisan bisa reschedule tpi ternyata ps ditanya ke agennyab gabisa, sama sajoh bohong. Udah gitu nomor agennya salin juga, ini gimanis sih? Kalau gini jadi gabisa reschedule. Ditanya ke cs juga gaterlak 6,"Waktu kita setting aplikasi download bahasa Indonesia tapi selalu muncul gagal karena internet terputus padahal internet blg baik baik. Apkili wifii maupun paket data juga sama" 7,"Ini kenapa ya klok gk bisa login ke akunku, malah masuknya jadi login baru, pihal emailnya sama. Udah logrout trus log in tetep sama. Masuknya ke akun yg baru, trus akun lama yg kemanalanin coba" 8,"Kecewa bgt ajukan refund tetapi terus aja suruh nunggu, buruknya lagip soart yang suh tidak oprasional tidak di blok aplikasi sehingga banyak korban yg tertipi dan saat mengajukan refund slow res dari pihak traveloka nya. Bahkan sampe dana tidak 9,"Akun travoloka susah untuk diakses, mande di logo burung. Harus di unistal dulu, kemuadian instal banlgr, beberapakali saya menglamaini. Traveloka jadi tidak nyaman untuk diakses kalau turus menerus seperti itu. Kalau hanya sanya (karena perangkat 10) Hallo mohon bantuanya. Sy tidak bisa login saat instal banlgr aja karena sy lgain untuk memindahkan akun traveloka. Saya stack di pilhan negara dan bahasa. Tidak bisa lanjut di hape ximoco pmo pro 11,"Sangat menggecewakan. Saya sudah pesan kamari di salah 1 wisma di daerah Palopo, sulsel selesaikan pembayaran juga. Tapi ps ke tempat nya, katanya sudah tidak berjekerjasama, dan pesanan saya tidak masuk di pihak mereka. Jadi sy tidak bisa 12,"sudah sangat lama menggunakna jasa traveloka untuk semua keperluan travelling stay & kelurga, tapi sangat2 menggecewakan karena traveloka baru membertahukan pembatalan hotel yang sudah pesan jauh hari sebelumnya hanya sebelur 13,makasih traveloka ..sempat was2 gak dpt tiket murah buat perjalanan luar kota...tetapi memberikan yg terbaik buat pelanggan yg lain ya...sukes selalu 14,Sebelumnya check in di line mudah skrg kenapa tambah susah lewat aplikasi. Kalai di upgrade tu semakin mempermudah bukan semakin mempersulit. Kami menggunakan aplikasi dan kru mudah penggunaannya tp kalau sp1 mungkin akan banyak yg 15,"Pengalamank saya pernah beberapa kali refund.. semu refund yg sy ajukan di kembalikan unganya ke rekening. Hanya waktu yg dtb bisa dipotek sebagian berasa lama akan kembali. Ada yg cepat ad yg lg lama. Bersabar aja." 16,Paymentnya sering trouble. Tolong lah benerin. Utdn 2x trouble dr bank berbeda. BRI kena BCA kena padahal udah kedebit. Gimana sih? Males skrg pakai traveloka klo bermasalah terus. Blkin panik. Mending beralfit ke aplikasi sebelah. 17,"Tentang refund, traveloka jangan terlalu lama dgn kaul dana nya, masa harus sampai 90 hari,. Sayar bayar transfer aja cuma diikis waktu beberapa jam harus transfer... ,sayar bayar cash bukan kredit sampai 90 hari, lagan yg membatalkan penerbangan 18,"Bingung jadwalnya buat booking tiket pesawat PP plus hotel,klo ribet banget. Rencana staycation,sekeleluar untuk dhari tpi klok ribet urusan tnggal chekin nya. Apa gunanya ada fitur penerbangan+hotel kalau gitu" 19,pelanggan yg ada setelahnya yg se schedule mudah. Proses refund juga mudah dan fair selalu refund ke 2 pembayaran agak lama, tapi dibayarannya punten karenne schedule dibatalkan oleh maskapai. Harapannya sayar pelayanan bagus ini dipertahankan 20,Tentang PayLater, diambilnya dana yg di kirimkan juga yg di ambilnya. Ehm.. di lunasakn di karenan kda yg nyata yg dikenakan yg lama banget keluar janji 30 hari keluar. Sistem yang ribet. 21,"Setelah update malah ga bisa buka kunci sama sekali, tulisananya terdapat malah kunci kunci" 22,"Refund tidak bisa 100%, padahal pembatalan di maskapai. Harusnya di ketentuan awal ditulis tahu lalu refund dan dikemasan biaya admin sebesar 50 ribu rupiah. Bukan hanya pasang stiker refundable di produknya" 23,"Update terbaru banyak iklan pop up menutup layar tertutup memutus layar, apa2an ini.. kita hanya pesan hotel dgpn cepat, itu sj.." 24,"Saya sudah mengajukan proses refund dana dari tiket yg sudah sah booking tapi sampai sekarang belum ada proses atau konfirmasi lanjut dari pihak terkait. Saya sudah lama refund sampai sekarang belum ada, gimana ini car kerjanya traveloka uan 25,Kali ini klok bell tiket dan asuransi semua total sesuai. Pas mau Pembayar selisih harga hamipr 200 ribu. Cek liheng normal tlk ada selisi. Pas mau bayar ada selisi lgj dan hnpri 200 ribu. Padahal langganan di traveloka. Kayak gini jd mls
---------	---

Gambar 4.3 Data Mentah Hasil Scrapping Aplikasi Traveloka

4.2 Preprocessing

Setelah melakukan pengambilan data menggunakan teknik scrapping, data mentah yang didapatkan dari kedua aplikasi tersebut akan melewati tahap *preprocessing*. Tujuan dari *preprocessing* adalah untuk menghasilkan data yang lebih terstruktur dan siap untuk digunakan pada proses selanjutnya. Tahap *preprocessing* yang dilakukan pada penelitian ini adalah *Case folding*, *Cleaning*, *Stopword removal*, *Tokenization*, dan *Normalization*.

4.2.1 Case Folding

Proses pertama pada tahap *preprocessing* adalah tahap *case folding*. *Case folding* adalah mengubah seluruh dokumen teks yang menggunakan huruf kapital (*uppercase*) menjadi dokumen teks berbentuk standar yang biasanya menggunakan huruf kecil (*lowercase*). Berikut *source code* dan hasil dari *case folding*.

```
1 # CaseFolding
2 data['Case Folding'] = data['content'].str.lower()
3 data
```

Gambar 4.4 Source Code Tahap Case Folding

	content	Case Folding
0	Sudah booking, dibayar lunas eh pas mau check ...	sudah booking, dibayar lunas eh pas mau check ...
1	Sebenarnya oke. Tp tampilan aplikasinya kenapa...	sebenarnya oke. tp tampilan aplikasinya kenapa...
2	Pembatalan oleh Agoda, REFUND sangat lama sek...	pembatalan oleh agoda, refund sangat lama sek...
3	Mengecewakan!!! Agoda tunai tidak bisa di gun...	mengecewakan!!! agoda tunai tidak bisa di gun...
4	Kacau banget ini aplikasinya. Tampilannya sih ...	kacau banget ini aplikasinya. tampilannya sih ...
...
26027	Ok	ok
26028	Coba	coba
26029	Bagus	bagus
26030	Good	good
26031	⭐⭐⭐⭐	⭐⭐⭐⭐
26032 rows × 2 columns		

Gambar 4.5 Hasil Case Folding Ulasan Agoda

	content	Case Folding
0	Kecewa bgt sih, parah, sama traveloka. Yg ladi...	kecewa bgt sih, parah, sama traveloka. yg tadi...
1	Tolong ya traveloka kalau mau nawarin paylater...	tolong ya traveloka kalau mau nawarin paylater...
2	Kembalikan fitur aplikasi seperti dulu, ini ko...	kembalikan fitur aplikasi seperti dulu, ini ko...
3	Untuk pengalaman yang mengecewakan, untuk peng...	untuk pengalaman yang mengecewakan, untuk peng...
4	Baca ni, serius, penting, pengalaman pribadi, ...	baca ni, serius, penting, pengalaman pribadi, ...
...
130328	NaN	NaN
130329	NaN	NaN
130330	NaN	NaN
130331	NaN	NaN
130332	NaN	NaN

130333 rows × 2 columns

Gambar 4.6 Hasil Case Folding Ulasan Traveloka

4.2.2 Cleaning

Tahap kedua adalah *cleaning*. *Cleaning* data diperlukan untuk menghilangkan karakter-karakter yang tidak diperlukan untuk mengurangi *noise* seperti tanda baca, angka, URL, *emoticon*, simbol, spasi yang tidak diinginkan di awal atau akhir kalimat, karakter *whitespace* ganda, *single character*, dan memisahkan dan menggabungkan kata. Berikut merupakan *source code* dan hasil dari *cleaning* data.

```

1 #Cleaning
2
3 #menghapus tabel kosong
4 data.dropna(inplace=True)
5
6 # menghapus karakter-karakter yang tidak diinginkan, seperti tanda baca dan angka
7 data['Cleaning'] = data['Case Folding'].apply(lambda x: re.sub('[\u00a1-\u00d7\s]', '', x))
8 data['Cleaning'] = data['Cleaning'].apply(lambda x: re.sub(r'\w\s', '', x))
9
10 #menghapus URL
11 data['Cleaning'] = data['Cleaning'].apply(lambda x: re.sub(r'http\S+', '', x))
12
13 # menghilangkan spasi yang tidak diinginkan di awal atau akhir kalimat
14 data['Cleaning'] = data['Cleaning'].apply(lambda x: x.strip())
15 data['Cleaning'] = data['Cleaning'].apply(lambda x: re.sub("\n", " ",x))
16
17 # Menghapus karakter whitespace ganda
18 data['Cleaning'] = data['Cleaning'].apply(lambda x: re.sub(r'\s+', ' ', x))
19
20 #menghilangkan single char
21 data['Cleaning'] = data['Cleaning'].apply(lambda x: re.sub(r"\b[a-zA-Z]\b", " ",x))
22
23 # memisahkan dan gabung kata
24 data['Cleaning'] = data['Cleaning'].apply(lambda x: ' '.join(x.split()))

```

Gambar 4.7 Source Code Tahap Cleaning Data

```

1 data=data[data['Cleaning']!= '']
2 data=data.reset_index(drop=True)
3 data

```

Gambar 4.8 Source Code Menghapus Index Kosong

Jika terdapat ulasan yang hanya berupa emoticon atau angka, maka ulasan tersebut menjadi ulasan kosong. Maka dari itu, gambar 4.8 merupakan *code* yang digunakan untuk menghilangkan ulasan kosong tersebut, lalu ulasan tersebut dilakukan *reset_index* untuk mengurutkan kembali ulasan atau mengisi urutan kosong yang terhapus.

	content	Case Folding	Cleaning
0	Sudah booking, dibayar lunas eh pas mau check ...	sudah booking, dibayar lunas eh pas mau check ...	sudah booking dibayar lunas eh pas mau check i...
1	Sebenarnya oke. Tp tampilan aplikasinya kenapa...	sebenarnya oke. tp tampilan aplikasinya kenapa...	sebenarnya oke tp tampilan aplikasinya kenapa ...
2	Pembatalan oleh Agoda, REFUND sangat lama seka...	pembatalan oleh agoda, refund sangat lama seka...	pembatalan oleh agoda refund sangat lama sekal...
3	Mengecewakan!!!! Agoda tunai tidak bisa di gun...	mengecewakan!!!! agoda tunai tidak bisa di gun...	mengecewakan agoda tunai tidak bisa di gunakan...
4	Kacau banget ini aplikasinya. Tampilannya sih ...	kacau banget ini aplikasinya. tampilannya sih ...	kacau banget ini aplikasinya tampilannya sih o...
...
25771	Top	top	top
25772	Ok	ok	ok
25773	Coba	coba	coba
25774	Bagus	bagus	bagus
25775	Good	good	good

25776 rows × 3 columns

Gambar 4.9 Hasil *Cleaning Data* Aplikasi Agoda

	content	Case Folding	Cleaning
0	Kecewa bgt sih, parah, sama traveloka. Yg tadi...	kecewa bgt sih, parah, sama traveloka. yg tadi...	kecewa bgt sih parah sama traveloka yg tadinya...
1	Tolong ya traveloka kalau mau nawarin paylater...	tolong ya traveloka kalau mau nawarin paylater...	tolong ya traveloka kalau mau nawarin paylater...
2	Kembalikan fitur aplikasi seperti dulu, ini ko...	kembalikan fitur aplikasi seperti dulu, ini ko...	kembalikan fitur aplikasi seperti dulu ini ko ...
3	Untuk pengalaman yang mengecewakan, untuk peng...	untuk pengalaman yang mengecewakan, untuk peng...	untuk pengalaman yang mengecewakan untuk peng...
4	Baca ni, serius, penting, pengalaman pribadi, ...	baca ni, serius, penting, pengalaman pribadi, ...	baca ni serius penting pengalaman pribadi kalo...
...
127729	Ok	ok	ok
127730	coba coba	coba coba	coba coba
127731	Ok	ok	ok
127732	Ok	ok	ok
127733	Nice	nice	nice

Gambar 4.10 Hasil *Cleaning Data* Aplikasi Traveloka

4.2.3 Stopword Removal

Stopword adalah kata-kata yang mengandung sedikit informasi yang biasanya tidak diperlukan dari suatu teks berbahasa Indonesia. Pada gambar 4.11, pertama dilakukan adalah mengimport library nltk dan mengunduh kamus *Stopwords*. Dari library

`nltk.corpus`, selanjutnya adalah mengimport *Stopwords* yang telah diunduh sebelumnya dan mengatur teks menjadi berbahasa Indonesia. Pada code tersebut, ditambahkan beberapa kata yang sering muncul dan tidak diperlukan yang dimasukkan ke dalam variabel `new_Stopwords` dalam bentuk list. Lalu variabel `stop_words` di perbarui agar kata-kata yang dimasukkan ke dalam variabel `new_Stopwords` masuk ke dalam variabel `stop_words`. Hasil dari *Stopword* dapat dilihat pada gambar 4.12 dan gambar 4.13.

```

1 # menghapus kata-kata yang tidak relevan atau stop words
2 import nltk
3 nltk.download('stopwords')
4 from nltk.corpus import stopwords
5 stop_words = set(stopwords.words('indonesian'))
6 new_stopwords = ['eh','ta','nya','yang','jinj','ki','nge','juga','jg','krn','karena','y','kan','atw','tks','agak','sih','kali','x','k',
7 'tae','tae','tai','tek'] # daftar kata baru yang ingin ditambahkan
8 stop_words.update(new_stopwords)
9 print(len(stop_words))
10 data['stopwords'] = data['Cleaning'].apply(lambda x: ' '.join([word for word in x.split() if word not in stop_words]))
11 data

```

Gambar 4.11 Source Code Stopword Removal

	content	Case Folding	Cleaning	Stopwords
0	Sudah booking, dibayar lunas eh pas mau check ...	sudah booking, dibayar lunas eh pas mau check ...	sudah booking dibayar lunas eh pas mau check ...	booking dibayar lunas pas check in pegawai hot... ...
1	Sebenarnya oke. Tp tampilan aplikasinya kenapa...	sebenarnya oke. tp tampilan aplikasinya kenapa...	sebenarnya oke tp tampilan aplikasinya kenapa ...	sebenarnya oke tp tampilan aplikasinya beda ta...
2	Pembatalan oleh Agoda, REFUND sangat lama sek...	pembatalan oleh agoda, refund sangat lama sek...	pembatalan oleh agoda refund sangat lama sekal...	pembatalan agoda refund prosesnya okt sd alasa...
3	Mengecewakan!!! Agoda tunai tidak bisa di gun...	mengecewakan!!! agoda tunai tidak bisa di gun...	mengecewakan agoda tunai tidak bisa di gunakan...	mengecewakan agoda tunai memesan hotel cairkan...
4	Kacau banget ini aplikasinya. Tampilannya sih ...	kacau banget ini aplikasinya. tampilannya sih ...	kacau banget ini aplikasinya tampilannya sih o...	kacau banget aplikasinya tampilannya oke harga...
...
25771	Top	top	top	top
25772	Ok	ok	ok	ok
25773	Coba	coba	coba	coba
25774	Bagus	bagus	bagus	bagus
25775	Good	good	good	good

Gambar 4.12 Hasil Stopword Aplikasi Agoda

0	Kecewa bgt sih, parah, sama traveloka. Yg tad...	kecwea bgt sih, parah, sama traveloka. yg tad...	kecwea bgt sih parah sama traveloka yg tadinya...	kecwea bgt parah traveloka limit paylater juta...
1	Tolong ya traveloka kalau mau nawarin paylater...	tolong ya traveloka kalau mau nawarin paylater...	tolong ya traveloka kalau mau nawarin paylater...	tolong ya traveloka nawarin paylater isi data ...
2	Kembalikan fitur aplikasi seperti dulu, ini ko...	kembalikan fitur aplikasi seperti dulu, ini ko...	kembalikan fitur aplikasi seperti dulu ini ko ...	kembalikan fitur aplikasi ko eror klo pakai wifi...
3	Untuk pengalaman yang mengecewakan, untuk peng...	untuk pengalaman yang mengecewakan, untuk peng...	untuk pengalaman yang mengecewakan untuk penga...	pengalaman mengecewakan pengajuan refund dana ...
4	Baca ni, serius, penting, pengalaman pribadi, ...	baca ni, serius, penting, pengalaman pribadi, ...	baca ni serius penting pengalaman pribadi kalio...	baca ni serius pengalaman pribadi kalio mesen p...
...
127729	Ok	ok	ok	ok
127730	coba coba	coba coba	coba coba	coba coba
127731	Ok	ok	ok	ok
127732	Ok	ok	ok	ok
127733	Nice	nice	nice	nice

Gambar 4.13 Hasil Stopword Aplikasi Traveloka

4.2.4 Tokenization

Setelah tahap *Stopword removal*, selanjutnya adalah *tokenization*. *Tokenization* berfungsi untuk memisahkan kalimat menjadi potongan-potongan kata atau token. Proses pertama untuk melakukan *tokenization* adalah mengimport kelas RegexpTokenizer dari library nltk.tokenize. Kelas tersebut dihunakan untuk melakukan tokenisasi berdasarkan pola yang didefinisikan. Kemudian, membuat objek tokenizer baru dengan menggunakan kelas RegexpTokenizer dan menyimpannya dalam variabel regexp. Pola yang digunakan dalam tokenizer ini adalah `r"\w+|[0-9]+|\S+"`. Pola ini akan mencocokkan dengan kata-kata (\w+), angka ([0-9]+), atau karakter selain spasi (\S+). Berikut *source code* dan hasil dari *tokenization*.

```

1 #tokenize
2 from nltk.tokenize import RegexpTokenizer
3 regexp = RegexpTokenizer(r'\w+|[0-9]+|\S+')
4 data['Token'] = data['stopwords'].apply(regexp.tokenize)
5 data

```

Gambar 4.14 Source Code Tokenization

	content	Case Folding	Cleaning	Stopwords	Token
0	Sudah booking, dibayar lunas eh pas mau check ...	sudah booking, dibayar lunas eh pas mau check ...	sudah booking dibayar lunas eh pas mau check ...	booking dibayar lunas pas check in pegawai hot...	[booking, dibayar, lunas, pas, check, in, pega...
1	Sebenarnya oke. Tp tampilan aplikasinya kenapa...	sebenarnya oke. tp tampilan aplikasinya kenapa...	sebenarnya oke tp tampilan aplikasinya kenapa ...	sebenarnya oke tp tampilan aplikasinya beda ta...	[sebenarnya, oke, tp, tampilan, aplikasinya, b...
2	Pembatalan oleh Agoda, REFUND sangat lama sek...a...	pembatalan oleh agoda, refund sangat lama sek...	pembatalan oleh agoda refund sangat lama sekal...	pembatalan agoda refund prosesnya okt sd alasa...	[pembatalan, agoda, refund, prosesnya, okt, sd...
3	Mengecewakan!!! Agoda tunai tidak bisa di gun...	mengecewakan!!! agoda tunai tidak bisa di gun...	mengecewakan agoda tunai tidak bisa di gunakan...	mengecewakan agoda tunai memesan hotel cairkan...	[mengecewakan, agoda, tunai, memesan, hotel, c...
4	Kacau banget ini aplikasinya. Tampilannya sih ...	kacau banget ini aplikasinya. tampilannya sih ...	kacau banget ini aplikasinya tampilannya sih o...	kacau banget aplikasinya tampilannya oke harga...	[kacau, banget, aplikasinya, tampilannya, oke, ...
...
25771	Top	top	top	top	[top]
25772	Ok	ok	ok	ok	[ok]
25773	Coba	coba	coba	coba	[coba]
25774	Bagus	bagus	bagus	bagus	[bagus]
25775	Good	good	good	good	[good]

Gambar 4.15 Hasil Tokenization Agoda

	content	Case Folding	Cleaning	Stopwords	Token
0	Kecewa bgt sih, parah, sama traveloka. Yg tadi...	kecewa bgt sih, parah, sama traveloka. yg tadi...	kecewa bgt sih parah sama traveloka yg tadinya...	kecewa bgt parah traveloka limit paylater juta...	[kecewa, bgt, parah, traveloka, limit, paylate...
1	Tolong ya traveloka kala mau nawarin paylater...	tolong ya traveloka kala mau nawarin paylater...	tolong ya traveloka kala mau nawarin paylater...	tolong ya traveloka nawarin paylater isi data ...	[tolong, ya, traveloka, nawarin, paylater, isi...
2	Kembalikan fitur aplikasi seperti dulu, ini ko...	kembalikan fitur aplikasi seperti dulu, ini ko...	kembalikan fitur aplikasi seperti dulu ini ko ...	kembalikan fitur aplikasi ko eror klo pake wif...	[kembalikan, fitur, aplikasi, ko, eror, klo, p...
3	Untuk pengalaman yang mengecewakan, untuk peng...	untuk pengalaman yang mengecewakan, untuk peng...	untuk pengalaman yang mengecewakan untuk penga...	pengalaman mengecewakan pengajuan refund dana ...	[pengalaman, mengecewakan, pengajuan, refund, ...
4	Baca ni, serius, penting, pengalaman pribadi, ...	baca ni, serius, penting, pengalaman pribadi, ...	baca ni serius penting pengalaman pribadi kalo...	baca ni serius pengalaman pribadi kalo mesen p...	[baca, ni, serius, pengalaman, pribadi, kalo, ...
...
127729	Ok	ok	ok	ok	[ok]
127730	coba coba	coba coba	coba coba	coba coba	[coba, coba]
127731	Ok	ok	ok	ok	[ok]
127732	Ok	ok	ok	ok	[ok]
127733	Nice	nice	nice	nice	[nice]

Gambar 4.16 Hasil Tokenization Traveloka

4.2.5 Normalizations

Normalizations atau normalisasi adalah proses mengubah kata yang tidak standar atau tidak konvensional menjadi kata yang lebih umum atau kata baku. Tujuan dari normalisasi ini adalah untuk mencapai konsistensi, mempermudah pemrosesan dan analisis teks. Pada gambar 4.17, yang pertama dilakukan adalah membaca file kamus_alay.csv yang disimpan dalam variabel normalized_word. Isi dari kamus_alay.csv dapat dilihat pada gambar 4.18. File kamus_alay.csv tersebut didapatkan dari internet dan beberapa kata juga ditambahkan secara manual sesuai dengan teks yang akan dinormalisasi. Langkah selanjutnya adalah membuat kamus untuk memetakan kata-kata alay atau tidak baku ke dalam bentuk yang normal atau baku. Setiap baris dalam kamus_alay.csv dibaca menggunakan perulangan for. Kata alay atau tidak baku pada kolom pertama dijadikan kata kunci dalam kamus normalized_word_dict, untuk kata normal atau baku diletakkan pada kolom kedua dan dijadikan nilai atau value dalam kamus. Kemudian masuk ke dalam

fungsi normalized_term(document). Fungsi ini menerima argumen document, yang merupakan teks yang akan dinormalisasi. Fungsi ini menggunakan list comprehension untuk menghasilkan list baru, di mana setiap elemen dalam daftar tersebut adalah kata normal atau baku jika kata tersebut terdapat dalam kamus normalized_word_dict, dan kata tersebut tetap atau tidak diubah menjadi kata normal atau baku jika tidak terdapat dalam kamus.

Hasil dari normalisasi dapat dilihat pada gambar 4.19 dan 4.20.

```

1 #normalisasi
2 normalized_word = pd.read_csv("kamus_alay.csv")
3
4 normalized_word_dict={}
5 for index, row in normalized_word.iterrows():
6     if row[0] not in normalized_word_dict:
7         normalized_word_dict[row[0]] = row[1]
8
9 def normalized_term(document):
10    return [normalized_word_dict[term] if term in normalized_word_dict else term for term in document]
11
12 data['Normalisasi'] = data['Token'].apply(normalized_term)
13 data

```

Gambar 4.17 Source Code Normalizations

A	B	C
1	slang,formal	
2	gw,aku	
3	gua,aku	
4	lo,kamu	
5	www,wow	
6	ad,ada	
7	amin,amin	
8	met,selamat	
9	netaas,menetas	
10	keberpa,keberapa	
11	eeeehhhh,eh	
12	kata2nyaaa,kata-katanya	
13	hallo,halo	
14	kaka,kakak	
15	ka,kak	
16	chek,check	
17	check,cek	
18	ku,aku	
19	laaah,lah	
20	mhn,mohon	
21	nggak,tidak	
22	an, atas nama	
23	kacauu,kacau	
24	banggettt,bangget	
25	rugiyyy,rugi	
26	jnj,ini	
27	daah,dah	

Gambar 4.18 Isi kamus_alay.csv

	content	Case Folding	Cleaning	Stopwords	Token	Normalisasi
0	Sudah booking, dibayar lunas eh pas mau check ...	sudah booking, dibayar lunas eh pas mau check ...	sudah booking dibayar lunas eh pas mau check ...	booking dibayar lunas pas check in pegawai hot...	[booking, dibayar, lunas, pas, check, in, pega...]	[booking, dibayar, lunas, pas, cek, in, pegaw...
1	Sebenarnya oke. Tp tampilan aplikasinya kenapa...	sebenarnya oke. tp tampilan aplikasinya kenapa...	sebenarnya oke tp tampilan aplikasinya kenapa ...	sebenarnya oke tp tampilan aplikasinya beda ta...	[sebenarnya, oke, tp, tampilan, aplikasinya, b...]	[sebenarnya, oke, tapi, tampilan, aplikasinya...
2	Pembatalan oleh Agoda, REFUND sangat lama sek...	pembatalan oleh agoda, refund sangat lama sek...	pembatalan oleh agoda refund sangat lama sek...	pembatalan agoda refund prosesnya okt sd alasa...	[pembatalan, agoda, refund, prosesnya, okt, sd...]	[pembatalan, agoda, refund, prosesnya, okt, sd...
3	Mengecewakan!!! Agoda tunai tidak bisa di gun...	mengecewakan!!! agoda tunai tidak bisa di gun...	mengecewakan agoda tunai tidak bisa di gunakan...	mengecewakan agoda tunai memesan hotel cairkan...	[mengecewakan, agoda, tunai, memesan, hotel, c...	[mengecewakan, agoda, tunai, memesan, hotel, c...
4	Kacau banget ini aplikasinya. Tampilannya sih ...	kacau banget ini aplikasinya. tampilannya sih ...	kacau banget ini aplikasinya tampilannya sih o...	kacau banget aplikasinya tampilannya oke harga...	[kacau, banget, aplikasinya, tampilannya, oke,...]	[kacau, banget, aplikasinya, tampilannya, oke,...]
...
25771	Top	top	top	top	[top]	[top]
25772	Ok	ok	ok	ok	[ok]	[ok]
25773	Coba	coba	coba	coba	[coba]	[coba]
25774	Bagus	bagus	bagus	bagus	[bagus]	[bagus]
25775	Good	good	good	good	[good]	[good]

Gambar 4.19 Hasil Normalization Pada Agoda

	content	Case Folding	Cleaning	Stopwords	Token	Normalisasi
0	Kecewa bgt sih, parah, sama traveloka. Yg tad...	kecewa bgt sih, parah, sama traveloka. yg tad...	kecewa bgt sih parah sama traveloka yg tadinya...	kecewa bgt parah traveloka limit paylater juta...	[kecewa, bgt, parah, traveloka, limit, paylate...]	[kecewa, banget, parah, traveloka, limit, payl...
1	Tolong ya traveloka kalaun mau nawarin paylater...	tolong ya traveloka kalaun mau nawarin paylater...	tolong ya traveloka kalaun mau nawarin paylater...	tolong ya traveloka nawarin paylater isi data ...	[tolong, ya, traveloka, nawarin, paylater, isi...]	[tolong, ya, traveloka, menawari, paylater, isi...
2	Kembalikan fitur aplikasi seperti dulu, ini ko...	kembalikan fitur aplikasi seperti dulu, ini ko...	kembalikan fitur aplikasi seperti dulu ini ko ...	kembalikan fitur aplikasi ko eror klo pake wif...	[kembalikan, fitur, aplikasi, ko, eror, klo, p...]	[kembalikan, fitur, aplikasi, kok, eror, kalo,...]
3	Untuk pengalaman yang mengecewakan, untuk peng...	untuk pengalaman yang mengecewakan, untuk peng...	untuk pengalaman yang mengecewakan untuk penga...	pengalaman mengecewakan untuk penga... pengajuan refund dana ...	[pengalaman, mengecewakan, pengajuan, refund, ...]	[pengalaman, mengecewakan, pengajuan, refund, ...]
4	Baca ni, serius, penting, pengalaman pribadi, ...	baca ni, serius, penting, pengalaman pribadi, ...	baca ni serius penting pengalaman pribadi kalo...	baca ni serius pengalaman pribadi kalo mesen p...	[baca, ni, serius, pengalaman, pribadi, kalo, ...]	[baca, ini, serius, pengalaman, pribadi, kalo, ...]
...
127729	Ok	ok	ok	ok	[ok]	[ok]
127730	coba coba	coba coba	coba coba	coba coba	[coba, coba]	[coba, coba]
127731	Ok	ok	ok	ok	[ok]	[ok]
127732	Ok	ok	ok	ok	[ok]	[ok]
127733	Nice	nice	nice	nice	[nice]	[nice]

Gambar 4.20 Hasil Normalization Pada Traveloka

4.2.6 Stemming

Stemming adalah proses untuk mengubah kata menjadi kata

dasar. Proses pertama sebelum memulai proses *stemming* adalah menginstall paket yang dimiliki oleh python yaitu *swifter* dan *Pysastrawi*. *Swifter* adalah sebuah pustaka *Python* yang dirancang untuk meningkatkan kecepatan pemrosesan data dengan menggunakan operasi vektorisasi pada *Pandas DataFrames*. Dengan menginstal *swifter*, dapat mempercepat proses pemrosesan data yang melibatkan operasi pada setiap baris dalam *DataFrame*.

Pysastrawi adalah pustaka *Python* untuk memanipulasi teks berbahasa Indonesia. Pustaka ini menyediakan berbagai fungsi dan algoritma untuk memproses dan menganalisis teks dalam bahasa Indonesia, termasuk *stemming*.

```
1 get_ipython().system('pip3 install swifter')
2 get_ipython().system(['pip3 install PySastrawi'])
```

Gambar 4.21 Paket Python *Swifter* dan *Pysastrawi*

Selanjutnya adalah mengimport kelas *Stemmerfactory* dari modul Stemmer dalam library Sastrawi. *Stemmerfactory* untuk membuat objek stemmer dan mengimport *swifter*. Lalu membuat objek bernama factory dari kelas *Stemmerfactory* yang digunakan untuk membuat objek stemmer.

```
1 from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
2 import swifter
3
4 factory = StemmerFactory()
5 stemmer = factory.create_stemmer()
```

Gambar 4.22 Import *Stemmerfactory* dan *Swifter*

Kode selanjutnya adalah mendefinisikan fungsi *stemmed_wrapper* yang mengambil argumen *term*. Fungsi ini mengembalikan kata yang telah di stem menggunakan objek stemmer. Lalu membuat variabel baru bernama *term_dict* yang berfungsi untuk menyimpan kata-kata unik yang ditemukan dalam dokumen. Selanjutnya membuat looping atau perulangan melalui setiap dokumen pada data kolom Normalisasi. Lalu melakukan loop untuk setiap kata (*term*) pada *document*. Jika kata (*term*) belum ada dalam *term_dict*, maka tambahkan *term* ke dalam *term_dict* dengan

nilai awal berupa string kosong lalu mencetak jumlah kata unik dalam term_dict pada setiap iterasi document. Lalu membuat perulangan lagi untuk setiap kata (term) dalam term_dict. Menggunkana fungsi stemmed_wrapper untuk mendapatkan kata dasar dari setiap kata. Menyimpan kata dasar yang diperoleh dalam term_dict sebagai nilai untuk kata kunci dalam term lalu mencetak kata dan kata dasar yang sesuai. Kemudian mendeklarasikan fungsi get_stemmed_term yang mengambil argumen document. Fungsi tersebut akan mengembalikan daftar kata dasar dari dokumen yang diberikan. Lalu mengembalikan daftar kata dasar untuk setiap kata dalam document.

```

1 def stemmed_wrapper(term) :
2     return stemmer.stem(term)
3
4 term_dict = {}
5
6 for document in data['Normalisasi']:
7     for term in document:
8         if term not in term_dict:
9             term_dict[term] = ' '
10
11 print(len(term_dict))
12 print("-----")
13
14 for term in term_dict:
15     term_dict[term] = stemmed_wrapper(term)
16     print(term,":",term_dict[term])
17
18 print(term_dict)
19 print("-----")
20
21 def get_stemmed_term(document):
22     return [term_dict[term] for term in document]
23
24 data['stemming'] = data['Normalisasi'].swifter.apply(get_stemmed_term)
25 print(data['Stemming'])

```

Gambar 4.23 Source Sode Stemming

Stemming		
0	[booking', 'bayar', 'lunas', 'pas', 'cek', 'i...	
1	['benar', 'oke', 'tapi', 'tampil', 'aplikasi',...	
2	['batal', 'agoda', 'refund', 'proses', 'okt', ...	
3	['kecewa', 'agoda', 'tunai', 'mes', 'hotel', '...	
4	['kacau', 'banget', 'aplikasi', 'tampil', 'oke...	
...
25771		['top']
25772		['ok']
25773		['coba']
25774		['bagus']
25775		['good']

Gambar 4.24 Hasil *Stemming* Aplikasi Agoda

Stemming		
	[kecewa, banget, parah, traveloka, limit, payl...	
	[tolong, ya, traveloka, tawar, paylater, isi, ...	
	[kembali, fitur, aplikasi, kok, eror, kalo, pa...	
	[alam, kecewa, aju, refund, dana, batal, terba...	
	[baca, ini, serius, alam, pribadi, kalo, mesen...	
...
	[ok]	
	[coba, coba]	
	[ok]	
	[ok]	
	[nice]	

Gambar 4.25 Hasil *Stemming* Aplikasi Agoda

4.2.7 Translate Data

Proses selanjutnya adalah melakukan *translate data* menggunakan library googletrans. Dari library googletrans, kemudian mengimport Translator untuk menerjemahkan teks. Membuat kamus kosong bernama translations yang akan digunakan untuk menyimpan terjemahan dari elemen-elemen unik dalam DataFrame. Selanjutnya membuat perulangan untuk melakukan iterasi pada setiap kolom pada DataFrame. Dalam perulangan, kemudian mengambil elemen-elemen unik pada kolom DataFrame saat ini. Metode unique() digunakan untuk mengambil nilai unik dari

suatu kolom. Di dalam perulangan berikutnya, kode melakukan iterasi pada setiap elemen unik dalam kolom menggunakan for element in unique_elements. Kemudian, dengan menggunakan objek translator, kode menerjemahkan elemen tersebut menjadi teks terjemahan menggunakan translator.translate(element).text. Teks terjemahan kemudian ditambahkan ke dalam kamus translations dengan menggunakan elemen asli sebagai kunci dan teks terjemahan sebagai nilai. Setelah semua elemen unik dalam kolom-kolom telah diterjemahkan, kamus translations berisi pasangan kunci-nilai, di mana kunci adalah elemen asli dan nilai adalah teks terjemahan dari elemen tersebut.

```

1 translator = Translator()
2 translations = {}
3 for column in df.columns:
4     #unique elements of the column
5     unique_elements = df[column].unique()
6     for element in unique_elements:
7         #Adding all the translations to a dictionary (translations)
8         translations[element] = translator.translate(element).text
9 translations
10

```

Gambar 4.26 Source Code Translate Data

4.4 Pelabelan Data

Setelah melakukan *preprocessing*, kemudian melakukan pelabelan data sentimen menggunakan metode *vadersentiment*. Metode ini menganalisis sentimen pada sebuah data yang berisi kata-kata dengan skor sentimen yang telah ditentukan sebelumnya. Data yang didapatkan mencakup kata-kata positif, kata-kata negatif, dan kata-kata netral. Pertama adalah menginstall *vadersentiment* lalu mengimport kelas SentimentIntensityAnalyzer.

```
1 from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
2 analyser = SentimentIntensityAnalyzer()
```

Gambar 4.27 Library Vadersentimen

Pada gambar 4.28 menggunakan metode polarity_scores dari objek analyser untuk menghitung skor sentimen dari setiap teks pada kolom *Stemming* dari dataframe dataAgoda. Metode ini digunakan untuk mengembalikan sebuah kamus yang berisi empat komponen skor sentimen yaitu positif, negatif, netral, dan compound. Hasil skor sentimen dari setiap teks pada kolom *Stemming* disimpan dalam variabel scores sebagai sebuah list lalu menyimpan skor compound pada setiap teks pada kolom *Stemming* pada kolom baru bernama Compound-Score pada dataframe dataAgoda.

```
1 scores = [analyser.polarity_scores(x) for x in dataAgoda['Stemming']]
2 print(scores)
3 dataAgoda['Compound-Score']=[x['compound'] for x in scores]
```

Gambar 4.28 Polarity Score

Kode pada gambar 4.29 digunakan untuk memberi label pada setiap teks pada kolom *Stemming* berdasarkan skor pada kolom Compound-Score. Jika skor pada kolom Compound-Score kurang dari nol, maka akan diberikan label Negatif, dan jika skor pada kolom Compound-Score lebih dari nol, maka akan diberikan label Positif. Karena yang diberikan label hanya skor yang diatas dan dibawah nol, maka nilai yang bernilai nol atau netral akan dihapus dan kodennya dapat dilihat pada gambar 4.30

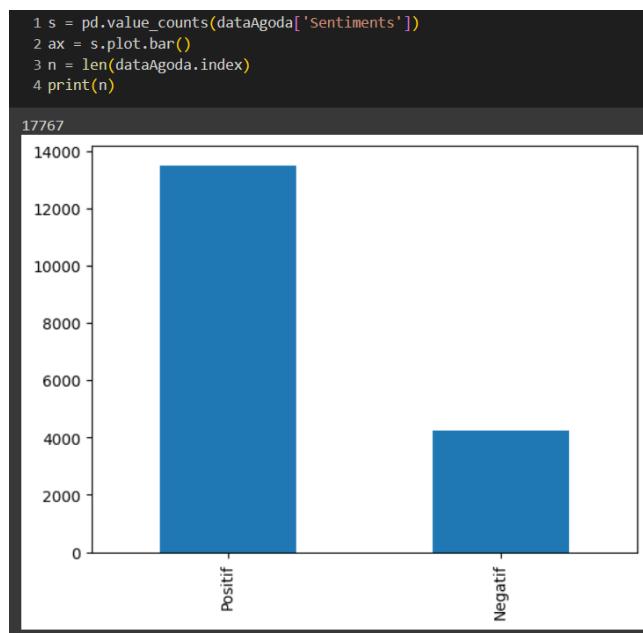
```
1 #compound score lexicon based
2 dataAgoda.loc[dataAgoda['Compound-Score'] < 0, 'Sentiments'] = 'Negatif'
3
4 dataAgoda.loc[dataAgoda['Compound-Score'] > 0, 'Sentiments'] = 'Positif'
5
6 dataAgoda
```

Gambar 4.29 Pemberian Label

```
1 dataAgoda = dataAgoda.dropna()
2 len(dataAgoda)
```

Gambar 4.30 Menghapus Data Kosong

Gambar 4.31 memperlihatkan jumlah sentimen negatif dan sentimen positif yang sudah diberi label sebelumnya dalam bentuk grafik batang dan juga terlihat bahwa jumlah sentimen negatif dan sentimen positif jika digabungkan adalah 17.767 sentimen.



Gambar 4.31 Grafik Jumlah Label

Karena pada gambar 4.32 sentimen yang bernilai NaN dihapus karena yang diambil hanya sentimen yang bernilai positif dan negatif, maka terdapat juga baris yang hilang. Oleh karena itu, diperlukan reset_index(drop=True) agar indeks baris dataframe diatur ulang mulai dari nol secara berurutan.

```
1 dataAgoda=dataAgoda.reset_index(drop=True)
```

Gambar 4.32 Reset Urutan Tabel

		Stemming	Compound-Score	Sentiments
0	['booking', 'pay', 'paid', 'pas', 'check', 'th...	0.7561	Positif	
1	['true', 'okay', 'but', 'appears', 'app', 'dif...	0.7876	Positif	
2	['cancel', 'agoda', 'refund', 'process', 'oct'...	-0.2023	Negatif	
3	['disappointed', 'agoda', 'cash', 'mes', 'hote...	0.1280	Positif	
4	['mess', 'really', 'app', 'appears', 'okay', '...	-0.5325	Negatif	
...	
17762	['good']	0.4404	Positif	
17763	['Good']	0.4404	Positif	
17764	['top']	0.2023	Positif	
17765	['Good']	0.4404	Positif	
17766	['good']	0.4404	Positif	

17767 rows × 3 columns

Gambar 4.33 Hasil Pelabelan

4.5 Pembagian Data *Testing* dan Data *Training*

Pembagian data *training* dan data *testing* menggunakan fungsi KFold milik library sklearn.model_selection. Dalam penelitian ini, nilai *k-folds* yang digunakan adalah 10. Nilai *k-folds* digunakan untuk membagi data menjadi 10 kali lipatan, yang berarti membagi data sebanyak 10 bagian yang sama ukurannya. Lalu data diacak sebelum pembagian fold untuk menghindari bias yang mungkin terjadi jika data memiliki urutan tertentu.

Gambar 4.34 merupakan kode *K-fold cross validation*.

```
# Inisialisasi KFold
kfold = KFold(n_splits=10, shuffle=True, random_state=42)
```

Gambar 4.34 *K-Fold Cross Validation*

4.6 Pembobotan TF-IDF

Pembobotan kata menggunakan kelas TfidfVectorizer dari library sklearn. Fitur-fitur ini mempresentasikan kata-kata atau istilah dalam teks yang digunakan untuk membangun model klasifikasi. Pada baris pertama, membuat variabel bernama vectorizer yang digunakan untuk

menginisialisasi objek `TfidfVectorizer()`. Objek tersebut digunakan untuk mengubah teks menjadi representasi vektor fitur TF-IDF. Kemudian membuat variabel `X` yang digunakan untuk menyimpan semua data pada kolom *Stemming*. Selanjutnya, menggunakan metode `'fit_transform'` pada objek `'vectorizer'` untuk mengubah teks dalam `'X_train'` menjadi matriks fitur TF-IDF yang akan digunakan sebagai data latih. Dalam proses ini, objek `'vectorizer'` mempelajari kosakata dari data latih dan menghitung skor TF-IDF untuk setiap kata dalam teks. Selanjutnya, menggunakan metode `'transform'` pada objek `'vectorizer'`, teks dalam `'X_test'` diubah menjadi matriks fitur TF-IDF yang akan digunakan sebagai data uji. Proses transformasi ini hanya menerapkan transformasi TF-IDF pada data uji, dengan mempertahankan kosakata yang telah dipelajari dari data latih. Dengan menggunakan `'fit_transform'` pada data latih dan `'transform'` pada data uji, memastikan bahwa representasi TF-IDF yang digunakan untuk pelatihan dan pengujian konsisten. Hal ini penting agar model dapat memahami dan mengklasifikasikan teks yang belum pernah dilihat sebelumnya dengan benar, berdasarkan kosakata yang telah dipelajari dari data latih. Gambar 4.35 merupakan kode untuk melakukan pembobotan kata menggunakan TF-IDF.

```
# Inisialisasi TfidfVectorizer
vectorizer = TfidfVectorizer()

# Ubah teks menjadi vektor fitur TF-IDF
X = data['Stemming'] # Ambil kolom 'Stemming' sebagai input teks

# Feature extraction using TF-IDF vectorizer
X_train_vectorized = vectorizer.fit_transform(X_train)
X_test_vectorized = vectorizer.transform(X_test)
```

Gambar 4.35 TF-IDF

4.7 Impementasi Seleksi Fitur *Query expansion ranking*

Pertama yang dilakukan untuk mengimplementasikan seleksi fitur *query expansion ranking* adalah menghitung jumlah kemunculan fitur pada dalam kelas positif dan negatif yang diinisialisai pada variabel data1_positif dan data1_negatif. Kemudian terdapat tiga fungsi yaitu hitung_pf, hitung_qf, dan hitung_score. Hitung_pf berfungsi untuk menghitung jumlah kemunculan fitur yang terjadi dalam sampel pada sentimen yang berlabel positif. Kemudian fungsi hitung_qf sama seperti fungsi hitung_pf hanya saja yang dihitung adalah kemunculan fitur yang berlabel negatif. Lalu hitung_score digunakan untuk menghitung skor untuk setiap fitur berdasarkan frekuensi kemunculan sampai dalam kelas positif dan negatif. Dalam fungsi hitung_score terdapat rumus skor yaitu $(pf-qf)/(pf+qf)$. Skor hanya dihitung untuk fitur-fitur dimana kemunculan frekuensi dalam dua kelas tidak nol. Lalu ada fungsi utama yaitu qer, fungsi ini mengimplementasikan algoritma QER. Variabel scores sebagai array kosong yang digunakan untuk menyimpan skor untuk setiap iterasi QER. Kemudian fungsi perulangan sebanyak num_features kali untuk menghitung skor dan memperbarui frekuensi kemunculan fitur. Variabel k digunakan untuk menentukan jumlah fitur teratas yang dipilih setelah peringkat fitur selesai. Fitur teratas k diambil dari nilai scores menggunakan slicing. Hasil rata-rata fitur tersebut kemudian diurutkan menggunakan argsort() dan disimpan pada variabel top_k_features.

```

# Query expansion ranking
num_features = X_train_vectorized.shape[1]
data1_positif = np.asarray(X_train_vectorized[y_train == 'Positif'].sum(axis=0)).flatten()
data1_negatif = np.asarray(X_train_vectorized[y_train == 'Negatif'].sum(axis=0)).flatten()
n_positif = sum(y_train == 'Positif')
n_negatif = sum(y_train == 'Negatif')

def hitung_pf(X_train_vectorized, y_train):
    # Hitung frekuensi kemunculan fitur dalam kelas positif
    pf = np.zeros(num_features)
    for i in range(len(X_train_vectorized)):
        if y_train.iloc[i] == 'Positif':
            for j in np.nonzero(X_train_vectorized[i])[0]:
                pf[j] += 1
    return pf

def hitung_qf(X_train_vectorized, y_train):
    # Hitung frekuensi kemunculan fitur dalam kelas negatif
    qf = np.zeros(num_features)
    for i in range(len(X_train_vectorized)):
        if y_train.iloc[i] == 'Negatif':
            for j in np.nonzero(X_train_vectorized[i])[0]:
                qf[j] += 1
    return qf

def hitung_score(pf, qf):
    mask = np.logical_or(pf == 0, qf == 0)
    score = np.zeros_like(pf)
    score[~mask] = np.abs(pf[~mask] - qf[~mask]) / np.abs(pf[~mask] + qf[~mask])
    return score

def qer(num_features, X_train_vectorized, y_train):
    scores = []
    pf = hitung_pf(X_train_vectorized, y_train)
    qf = hitung_qf(X_train_vectorized, y_train)
    for _ in range(num_features):
        scores.append(hitung_score(pf, qf))
        selected = np.argsort(scores[-1])[::-1][0]
        pf[selected] += 1
        qf[selected] += 1
    scores = np.asarray(scores)
    scores.sort(axis=0)
    scores = scores[::-1]
    scores = scores[::1]
    return scores

k = 1500
scores = qer(num_features, X_train_vectorized, y_train)
top_k_features = scores[:, :].mean(axis=0).argsort()

# Filter train and test sets with top k features
X_train_filtered = X_train_vectorized[:, top_k_features]
X_test_filtered = X_test_vectorized[:, top_k_features]

```

Gambar 4.36 Kode Seleksi Fitur QER

4.8 Implementasi Analisis Sentimen *Support Vector Machine*

Proses analisis sentimen dilakukan dengan model *Support vector machine* (SVM). Data yang digunakan merupakan data yang sudah melewati proses *preprocessing* dan *labeling*. Kemudian data dibagi menjadi data *training* dan data *testing* menggunakan *k-fold cross validation*. Nilai *split* yang digunakan untuk membagi data *training* dan *testing* adalah 3, 5, 7, dan 9.

Tahapan selanjutnya adalah pembobotan kata menggunakan metode TF-IDF dilanjutkan dengan seleksi fitur dengan metode *Query expansion ranking*. Seleksi fitur menggunakan *Query expansion ranking* digunakan untuk meningkatkan kualitas pencarian dengan memperluas atau menambahkan kata-kata yang relevan. Lalu masuk ke tahap pemodelan menggunakan *Support vector machine* menggunakan bantuan modul svc yang terdapat pada library sklearn.svm. Parameter yang digunakan dalam pemodelan SVM ini adalah *kernel*, *C(Cost)*, dan *gamma*. Kernel *linear* dipilih karena sesuai jika sentimen positif dan negatif dapat dipisahkan dengan baik oleh sebuah garis lurus. Sentimen positif dan negatif dapat memiliki perbedaan yang jelas dalam teks, sehingga dapat terjadi pemisahan linier antara keduanya. Selain itu, dalam analisis sentimen yang melibatkan volume data yang besar, menggunakan kernel *linear* dapat mengurangi waktu pelatihan.

```

13 # Train SVM classifier
14 svm = SVC(kernel = 'linear', C = 10, gamma = 0.001)
15 svm.fit(X_train_vectorized, y_train)
16
17 # Make predictions
18 y_pred = svm.predict(X_test_vectorized)
19

```

Gambar 4.37 Kode SVM

4.9 Implementasi Pengukuran Performa Menggunakan *Confusion Matrix*

Proses selanjutnya setelah pembentukan model adalah mengukur performa hasil analisis sentimen menggunakan metode *Confusion matrix*. Pengukuran performa dilakukan dengan cara membandingkan hasil analisis sentimen dengan label yang telah diberikan sebelumnya menggunakan metode *Vader*. Hasil yang diperoleh berupa nilai akurasi hasil prediksi

keseluruhan dari model tersebut. *Confusion matrix* dibentuk menggunakan library `sklearn.metrics` dengan fungsi `confusion_matrix`. Pertama dilakukan adalah mencetak semua hasil akurasi pada tiap fold. *K-fold* yang digunakan dalam *k-fold* adalah 3, 5, 7, dan 9 maka hasil akurasi akan muncul pada tiap fold mulai dari fold pertama hingga fold yang ditentukan. Kemudian fungsi `confusion_matrix(y_test, y_pred)` digunakan untuk menghasilkan *confusion matrix* berdasarkan label aktual dari variabel `y_test` dan label yang diprediksi dari variabel `y_pred`. Untuk *classification report* dibentuk menggunakan library yang sama dengan *confusion matrix* namun menggunakan fungsi `classification_report`. Fungsi `classification_report(y_test, y_pred)` digunakan untuk menghasilkan *classification report* berdasarkan label aktual dari variabel `y_test` dan label yang diprediksi dari variabel `y_pred`. Lalu membuat variabel untuk menghitung nilai rata-rata dari akurasi pada tiap fold. Menghitung rata-rata akurasi dengan cara menjumlahkan semua akurasi tiap fold kemudian dibagi dengan jumlah *k-fold* yang digunakan. Hasil rata-rata akurasi dikalikan dengan 100 agar mendapatkan hasil dalam bentuk persentase.

```
# Cetak hasil akurasi setiap fold
print(f"Fold-{len(accuracy_scores)} accuracy: {accuracy * 100}")

# Cetak confusion matrix
print("Confusion Matrix")
print(confusion_matrix(y_test, y_pred))
print()
print("Classification report:")
print(classification_report(y_test, y_pred))
print("-----")

# Evaluate the model
average_accuracy = sum(accuracy_scores) / len(accuracy_scores)
print("Average Accuracy:", average_accuracy * 100)
```

Gambar 4.38 Kode *Confusion Matrix*

Hasil pengujian *confusion matrix* menggunakan kernel *linear*, $c = 100$, $\gamma = 0,1$, dan nilai $k = 1300$ dapat dilihat pada gambar 4.39. Hasil tersebut menggunakan nilai $k\text{-fold}$ $k\text{-fold} = 9$ sehingga hasil akurasi yang muncul akan sebanyak 9 kali.

```

Fold-1 accuracy: 9529.113924050633
Fold-2 accuracy: 9544.072948328267
Fold-3 accuracy: 9488.348530901721
Fold-4 accuracy: 9468.08510638298
Fold-5 accuracy: 9589.66565349544
Fold-6 accuracy: 9564.33637284701
Fold-7 accuracy: 9508.611955420465
Fold-8 accuracy: 9513.677811550151
Fold-9 accuracy: 9620.060790273556

-----
Accuracy: 96.20060790273556
Confusion Matrix
[[ 430   36]
 [  39 1469]]

Classification report:
              precision    recall    f1-score   support
Negatif        0.92      0.92      0.92       466
Positif        0.98      0.97      0.98      1508

accuracy          0.96      0.96      0.96      1974
macro avg       0.95      0.95      0.95      1974
weighted avg    0.96      0.96      0.96      1974

-----
Average Accuracy: 95.36219232583358

```

Gambar 4.39 Hasil *Confusion Matrix*

4.10 Analisis Hasil

4.10.1 Pengaruh Nilai *K-Fold Cross Validation*, *C*, dan *Gamma*

Setelah melakukan pengujian menggunakan metode *support vector machine*, dapat diperoleh bahwa nilai $k\text{-fold}$ yang dimiliki oleh $k\text{-fold}$, nilai C (*Penalty*), dan Γ berpengaruh dalam hasil akurasi dalam sentimen analisis. Dalam penelitian ini, nilai $k\text{-fold}$ yang digunakan adalah 3, 5, 7, dan 9. Nilai 3, 5, 7, dan 9 dipilih karena pada nilai $k\text{-fold}$ yang kecil, proses $k\text{-fold}$ akan lebih cepat karena dataset dibagi menjadi sedikit bagian ($k\text{-fold} = 3$), kemudian untuk $k\text{-fold} = 9$ dipilih karena estiasi performa model menjadi lebih

stabil karena setiap iterasi menggunakan bagian data uji yang lebih besar dan berbeda, tetapi proses *k-fold* menjadi lebih lambar karena dataset dibagi menjadi banyak bagian. Kemudian $k-fold = 5$ dan 7 dipilih karena dapat membantu memastikan bahwa setiap lipatan memiliki ukuran yang hampir sama, sehingga tidak ada subset yang sangat kecil atau besar secara drastis.

Nilai C yang digunakan adalah 1 , 10 , dan 100 karena nilai $C = 1$ dapat mencegah *overfitting*, selain itu nilai C yang kecil dapat menghasilkan *margin* yang lebih lebar dan lebih dapat memahami *outlier*. Nilai $C = 10$, dan 100 dipilih karena nilai C yang tinggi dapat memberikan hukuman yang sangat besar pada kesalahan klasifikasi pada data pelatihan. Model C yang tinggi cenderung mencari *margin* yang sangat ketat dan lebih fokus pada klasifikasi dengan benar sebanyak mungkin pada data pelatihan.

Nilai *gamma* yang digunakan adalah $0,1$; $0,01$; dan $0,001$ tetapi pada kernel *linear*, nilai *gamma* tidak berpengaruh pada performa model karena garis batas yang dihasilkan pada kernel *linear* selalu berbentuk garis lurus atau *hyperplane* di ruang fitur asli. Kernel *linear* hanya menghitung dot product antara vektor fitur tanpa melakukan transformasi *non-linear*. Parameter *gamma* tidak digunakan dalam perhitungan fungsi kernel *linear* dan hanya diperlukan pada kernel-kernel yang memperkenalkan dimensi baru atau transformasi *non-linear* pada data. Kernel yang digunakan adalah kernel *linear*. Kernel ini dipilih karena sesuai jika sentimen

positif dan negatif dapat dipisahkan dengan baik oleh sebuah garis lurus. Sentimen positif dan negatif dapat memiliki perbedaan yang jelas dalam teks, sehingga dapat terjadi pemisahan linier antara keduanya. Selain itu, dalam analisis sentimen yang melibatkan volume data yang besar, menggunakan kernel *linear* dapat mengurangi waktu pelatihan. Tabel 4.1 merupakan perbandingan dari akurasi pada tiap *k-fold*, C, dan Gamma pada aplikasi agoda dan tabel 4.2 merupakan milik traveloka. Dilihat dari tabel 4.1 dan 4.2 pada bagian akurasi, warna hijau tua merupakan akurasi tertinggi, putih merupakan akurasi sedang, dan merah merupakan akurasi rendah dari tiap *k-fold*.

Tabel 4.1 Perbandingan Akurasi Tiap *K-Fold* Aplikasi Agoda

K-Fold	C	Gamma	Akurasi	K-Fold	C	Gamma	Akurasi
3	1	0,1	95,25%	5	1	0,1	95,92%
		0,01	95,25%			0,01	95,92%
		0,001	95,25%			0,001	95,92%
	10	0,1	94,56%		10	0,1	94,85%
		0,01	94,56%			0,01	94,85%
		0,001	94,56%			0,001	94,85%
	100	0,1	92,69%		100	0,1	93,47%
		0,01	92,69%			0,01	93,47%
		0,001	92,69%			0,001	93,47%
7	1	0,1	95,82%	9	1	0,1	96,20%
		0,01	95,82%			0,01	96,20%
		0,001	95,82%			0,001	96,20%
	10	0,1	95,43%		10	0,1	95,80%
		0,01	95,43%			0,01	95,80%
		0,001	95,43%			0,001	95,80%
	100	0,1	93,66%		100	0,1	94,68%
		0,01	93,66%			0,01	94,68%
		0,001	93,66%			0,001	94,68%

Nilai *k-fold* = 3 lebih rendah akurasinya daripada nilai *k-fold* = 5 pada semua nilai C dan Gamma pada aplikasi agoda. Kemudian

pada $k\text{-fold} = 7$ kembali terjadi penurunan akurasi pada nilai C 100. Pada $k\text{-fold} = 9$, pada nilai C = 1 di semua nilai Gamma, tingkat akurasi yang didapatkan merupakan akurasi tertinggi. Nilai C = 100 di semua Gamma juga merupakan akurasi terendah dari nilai K = 3, 5, 7, dan 9. Tabel 4.1 dan 4.2 membuktikan bahwa nilai dari $k\text{-fold}$ dan C berpengaruh dalam pengukuran tingkat akurasi menggunakan metode *support vector machine*. Untuk nilai gamma, tidak ada perubahan peningkatan atau penurunan akurasi pada tiap $k\text{-fold}$. Nilai gamma tidak berpengaruh terhadap tingkat akurasi kecuali mengikuti nilai C nya.

Tabel 4.2 Perbandingan Akurasi Tiap $K\text{-Fold}$ Aplikasi Traveloka

K-Fold	C	Gamma	Akurasi	K-Fold	C	Gamma	Akurasi
3	1	0,1	95,28%	5	1	0,1	95,39%
		0,01	95,28%			0,01	95,39%
		0,001	95,28%			0,001	95,39%
	10	0,1	95,87%		10	0,1	95,71%
		0,01	95,87%			0,01	95,71%
		0,001	95,87%			0,001	95,71%
	100	0,1	93,88%		100	0,1	94,74%
		0,01	93,88%			0,01	94,74%
		0,001	93,88%			0,001	94,74%
K-Fold	C	Gamma	Akurasi	K-Fold	C	Gamma	Akurasi
7	1	0,1	96,18%	9	1	0,1	96,26%
		0,01	96,18%			0,01	96,26%
		0,001	96,18%			0,001	96,26%
	10	0,1	97,09%		10	0,1	96,61%
		0,01	97,09%			0,01	96,61%
		0,001	97,09%			0,001	96,61%
	100	0,1	95,54%		100	0,1	95,79%
		0,01	95,54%			0,01	95,79%
		0,001	95,54%			0,001	95,79%

Untuk hasil akurasi pada aplikasi traveloka lebih bervariasi.

Akurasi tertinggi terdapat pada nilai k=7 dengan C=10 yaitu 97,09%.

Sama seperti pada aplikasi agoda, nilai gamma tidak berpengaruh terhadap tingkat akuras kecuali mengikuti nilai C nya.

4.10.2 Pengaruh Nilai K Dalam *Query expansion ranking*

Nilai k dalam *query expansion ranking* merupakan jumlah fitur teratas yang akan digunakan dalam proses *query expansion ranking* dan dalam set pelatihan dan pengujian. Pada penelitian ini, k yang digunakan yaitu 1000, 1300, dan 1500. Hasil akurasi menggunakan *query expansion ranking* dapat dilihat pada tabel 4.3 untuk aplikasi agoda dan tabel 4.4 untuk aplikasi traveloka. Pada aplikasi agoda, akurasi tertinggi terdapat pada $k\text{-}fold} = 9$. Nilai selected features atau nilai k tidak memberikan perubahan yang signifikan dalam peningkatan atau penurunan akurasi. Hasil akurasi milik aplikasi agoda, perubahan yang disebabkan oleh jumlah seleksi fitur hanya terdapat pada nilai $k\text{-}fold} = 3$. Perubahan terjadi di nilai k = 1300 dan 1500 pada C = 100. Selebihnya tidak ada lagi perubahan akurasi yang disebabkan oleh seleksi fitur yang dipilih. Untuk akurasi tertinggi terdapat pada $k\text{-}fold} = 9$ untuk setiap seleksi fitur pada nilai C = 1.

Tabel 4.3 Perbandingan Akurasi Tiap K Aplikasi Agoda

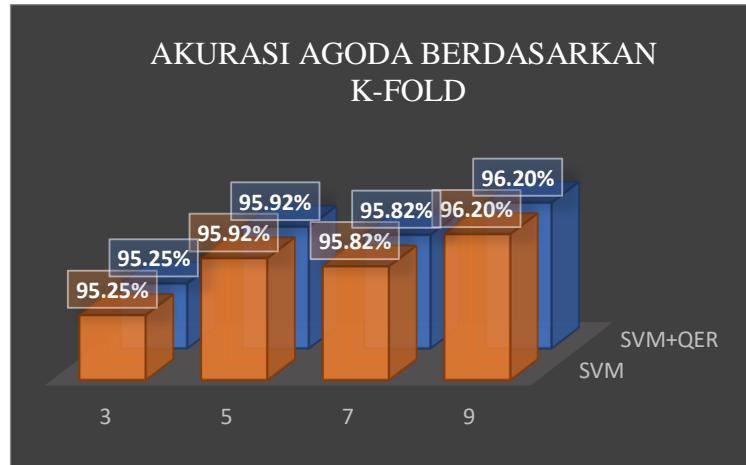
K-Fold	K	C	Gamma	Akurasi	K-Fold	ected Feat	C	Gamma	Akurasi
3	1300	1000	1	0.1 95,25% 0.01 95,25% 0.001 95,25%	5	1300	1	0.1 95,92% 0.01 95,92% 0.001 95,92%	
			10	0.1 94,56% 0.01 94,56% 0.001 94,56%			10	0.1 94,85% 0.01 94,85% 0.001 94,85%	
			100	0.1 92,71% 0.01 92,71% 0.001 92,71%			100	0.1 93,47% 0.01 93,47% 0.001 93,47%	
		1300	1	0.1 95,25% 0.01 95,25% 0.001 95,25%			1	0.1 95,92% 0.01 95,92% 0.001 95,92%	
			10	0.1 94,56% 0.01 94,56% 0.001 94,56%			10	0.1 94,85% 0.01 94,85% 0.001 94,85%	
			100	0.1 92,69% 0.01 92,69% 0.001 92,69%			100	0.1 93,47% 0.01 93,47% 0.001 93,47%	
		1500	1	0.1 95,25% 0.01 95,25% 0.001 95,25%			1	0.1 95,92% 0.01 95,92% 0.001 95,92%	
			10	0.1 94,56% 0.01 94,56% 0.001 94,56%			10	0.1 94,85% 0.01 94,85% 0.001 94,85%	
			100	0.1 92,69% 0.01 92,69% 0.001 92,69%			100	0.1 93,47% 0.01 93,47% 0.001 93,47%	
7	1300	1000	1	0.1 95,82% 0.01 95,82% 0.001 95,82%	9	1300	1	0.1 96,20% 0.01 96,20% 0.001 96,20%	
			10	0.1 95,43% 0.01 95,43% 0.001 95,43%			10	0.1 95,80% 0.01 95,80% 0.001 95,80%	
			100	0.1 93,66% 0.01 93,66% 0.001 93,66%			100	0.1 94,68% 0.01 94,68% 0.001 94,68%	
		1300	1	0.1 95,82% 0.01 95,82% 0.001 95,82%			1	0.1 96,20% 0.01 96,20% 0.001 96,20%	
			10	0.1 95,43% 0.01 95,43% 0.001 95,43%			10	0.1 95,80% 0.01 95,80% 0.001 95,80%	
			100	0.1 93,66% 0.01 93,66% 0.001 93,66%			100	0.1 94,68% 0.01 94,68% 0.001 94,68%	
		1500	1	0.1 95,82% 0.01 95,82% 0.001 95,82%			1	0.1 96,20% 0.01 96,20% 0.001 96,20%	
			10	0.1 95,43% 0.01 95,43% 0.001 95,43%			10	0.1 95,80% 0.01 95,80% 0.001 95,80%	
			100	0.1 93,66% 0.01 93,66% 0.001 93,66%			100	0.1 94,68% 0.01 94,68% 0.001 94,68%	

Pada aplikasi traveloka, sama seperti aplikasi yang sudah dijelaskan sebelumnya. Perubahan yang terjadi akibat pemilihan seleksi fitur tidak terlalu signifikan. Nilai QER tidak terlalu mempengaruhi dalam peningkatan akurasi. Perubahan akibat seleksi fitur QER hanya terjadi pada nilai $k\text{-fold} = 3$ dengan nilai C = 100. Untuk hasil akurasi tertinggi bernilai 97,09% pada nilai $k\text{-fold} = 7$ dan nilai C = 10.

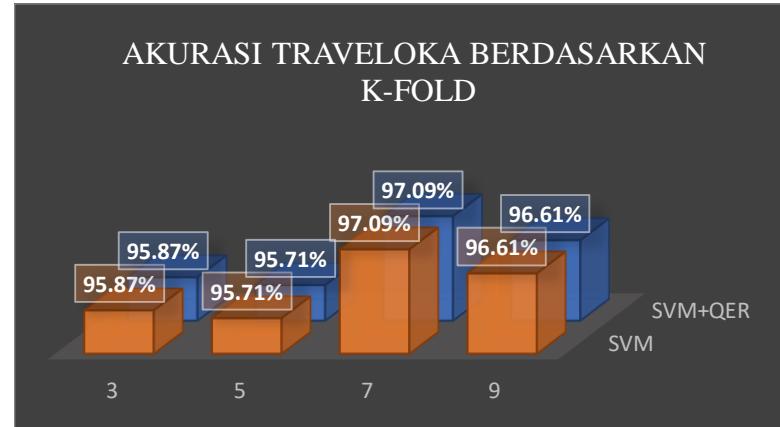
Tabel 4.4 Perbandingan Akurasi Tiap K Aplikasi Traveloka

K-Fold	K	C	Gamma	Akurasi	K-Fold	K	C	Gamma	Akurasi
3	1000	1	0,1	95,28%	5	1300	1	0,1	95,39%
			0,01	95,28%				0,01	95,39%
			0,001	95,28%				0,001	95,39%
		10	0,1	95,87%				0,1	95,71%
			0,01	95,87%				0,01	95,71%
			0,001	95,87%				0,001	95,71%
		100	0,1	93,84%				0,1	94,74%
			0,01	93,84%				0,01	94,74%
			0,001	93,84%				0,001	94,74%
		1300	0,1	95,28%				0,1	95,39%
			0,01	95,28%				0,01	95,39%
			0,001	95,28%				0,001	95,39%
		1000	0,1	95,87%				0,1	95,71%
			0,01	95,87%				0,01	95,71%
			0,001	95,87%				0,001	95,71%
		1500	0,1	93,84%				0,1	94,74%
			0,01	93,84%				0,01	94,74%
			0,001	93,84%				0,001	94,74%
		1	0,1	95,28%				0,1	95,39%
			0,01	95,28%				0,01	95,39%
			0,001	95,28%				0,001	95,39%
		10	0,1	95,87%				0,1	95,71%
			0,01	95,87%				0,01	95,71%
			0,001	95,87%				0,001	95,71%
		100	0,1	93,88%				0,1	94,74%
			0,01	93,88%				0,01	94,74%
			0,001	93,88%				0,001	94,74%

K-Fold	K	C	Gamma	Akurasi	K-Fold	K	C	Gamma	Akurasi
7	1000	1	0,1	96,18%	9	1300	1	0,1	96,26%
			0,01	96,18%				0,01	96,26%
			0,001	96,18%				0,001	96,26%
		10	0,1	97,09%				0,1	96,61%
			0,01	97,09%				0,01	96,61%
			0,001	97,09%				0,001	96,61%
		100	0,1	95,54%				0,1	95,79%
			0,01	95,54%				0,01	95,79%
			0,001	95,54%				0,001	95,79%
		1300	0,1	96,18%				0,1	96,26%
			0,01	96,18%				0,01	96,26%
			0,001	96,18%				0,001	96,26%
		10	0,1	97,09%				0,1	96,61%
			0,01	97,09%				0,01	96,61%
			0,001	97,09%				0,001	96,61%
		100	0,1	95,54%				0,1	95,79%
			0,01	95,54%				0,01	95,79%
			0,001	95,54%				0,001	95,79%
		1500	0,1	96,18%				0,1	96,26%
			0,01	96,18%				0,01	96,26%
			0,001	96,18%				0,001	96,26%
		1	0,1	97,09%				0,1	96,61%
			0,01	97,09%				0,01	96,61%
			0,001	97,09%				0,001	96,61%
		10	0,1	95,54%				0,1	95,79%
			0,01	95,54%				0,01	95,79%
			0,001	95,54%				0,001	95,79%
		100	0,1	95,54%				0,1	95,79%
			0,01	95,54%				0,01	95,79%
			0,001	95,54%				0,001	95,79%



Gambar 4. 40 Grafik Perbandingan Akurasi Pada Agoda



Gambar 4.40 Grafik Perbandingan Akurasi Pada Traveloka

4.10.3 Hasil *Precision* , *Recall*, dan *F1-Score*

Precision merupakan evaluasi kinerja pengambilan (klasifikasi) yang mengukur proporsi dokumen relevan yang diambil dari total dokumen yang diambil. *Recall* merupakan proporsi dari nilai positif yang sebenarnya yang benar diidentifikasi sebagai nilai positif. Sedangkan *F1-score* merupakan gabungan antara *precision* dan *recall*. Hasil dari *precision*, *recall*, dan *F1-score* yang didapatkan berfungsi sebagai evaluasi kinerja dari sebuah model.

Gambar 4.42 merupakan hasil *confusion matrix* akurasi tertinggi dari aplikasi Agoda.

Confusion Matrix				
		precision	recall	f1-score
Negatif	0.92	0.92	0.92	466
	0.98	0.97	0.98	1508

Gambar 4.42 *Confusion Matrix* Akurasi Tertinggi Agoda

Dalam tabel *confusion matrix* gambar 4.42, terdapat *true negative* sebanyak 430 kasus, *false positive* sebanyak 36 kasus, *false negative* sebanyak 39 kasus, dan *true positive* sebanyak 1469 kasus.

Dalam classification report terdapat ringkasan statistik evaluasi model berdasarkan *precision*, *recall*, dan *f1-score* untuk masing-masing kelas positif dan negatif. *Precision* untuk kelas negatif adalah 0,92 yang berarti dari semua kasus yang diprediksi negatif oleh model, 92% diantaranya adalah benar-benar negatif. Untuk kelas positif, nilai *precision* adalah 0,98 yang artinya dari semua kasus yang diprediksi positif, 98% diantaranya adalah benar-benar positif. Kemudian nilai *recall* untuk kelas negatif adalah 0,92 yang berarti model dapat mengidentifikasi 92% dari semua kasus negatif yang sebenarnya. Untuk kelas positif, nilai *recall* adalah 0,97 yang berarti model dapat mengidentifikasi 97% dari semua kasus positif yang sebenarnya. *F1-score* untuk kelas negatif adalah 0,92 sedangkan untuk kelas positif adalah 0,98. Gambar 4.41 merupakan hasil *confusion matrix* akurasi tertinggi dari aplikasi Traveloka.

Confusion Matrix				
[[113 19]				
[13 954]]				
Classification report:				
	precision	recall	f1-score	support
	Negatif	0.90	0.86	0.88
	Positif	0.98	0.99	0.98
				132
				967

Gambar 4.41 *Confusion Matrix* Akurasi Tertinggi Traveloka

Dalam tabel *confusion matrix* gambar 4.41, terdapat *true negative* sebanyak 113 kasus, *false positive* sebanyak 19 kasus, *false negative* sebanyak 13 kasus, dan *true positive* sebanyak 954 kasus.

Dalam classification report terdapat ringkasan statistik evaluasi model berdasarkan *precision*, *recall*, dan *f1-score* untuk masing-masing kelas positif dan negatif. *Precision* untuk kelas negatif adalah 0,90 yang berarti dari semua kasus yang diprediksi negatif oleh model, 90% diantaranya adalah benar-benar negatif. Untuk kelas positif, nilai *precision* adalah 0,98 yang artinya dari semua kasus yang diprediksi positif, 98% diantaranya adalah benar-benar positif. Kemudian nilai *recall* untuk kelas negatif adalah 0,86 yang berarti model dapat mengidentifikasi 86% dari semua kasus negatif yang sebenarnya. Untuk kelas positif, nilai *recall* adalah 0,99 yang berarti model dapat mengidentifikasi 99% dari semua kasus positif yang sebenarnya. *F1-score* untuk kelas negatif adalah 0,88 sedangkan untuk kelas positif adalah 0,98.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan pada perbandingan tingkat akurasi pengguna agoda dan traveloka menggunakan seleksi fitur *Query expansion ranking* dengan metode *support vector machine* yang menggunakan variasi nilai $k\text{-}fold = 3, 5, 7$, dan 9 , nilai $C = 1, 10$, dan 100 , nilai $\gamma = 0,1; 0,01; 0,001$ dan jumlah seleksi fitur untuk menentukan jumlah fitur teratas adalah $1000, 1300$, dan 1500 diperoleh kesimpulan dari pengujian dengan menggunakan kernel *linear* sebagai berikut:

1. Pada aplikasi agoda tanpa menggunakan seleksi fitur *query expansion ranking*, diperoleh akurasi tertinggi sebesar $96,20\%$ pada nilai $k\text{-}fold = 9$, $C = 1$, dan $\gamma = 0,1; 0,01; 0,001$.
2. Pada aplikasi traveloka tanpa menggunakan seleksi fitur *query expansion ranking*, diperoleh akurasi tertinggi sebesar $97,09\%$ pada nilai $k\text{-}fold = 7$, $C = 10$, dan $\gamma = 0,1; 0,01; 0,001$.
3. Kemudian pada aplikasi agoda menggunakan seleksi fitur *query expansion ranking*, diperoleh akurasi tertinggi sebesar $96,20\%$ pada nilai $k\text{-}fold = 9$, seleksi fitur = $1000, 1300, 1500$, $C = 1$, dan $\gamma = 0,1; 0,01; 0,001$.
4. Selanjutnya pada aplikasi traveloka menggunakan seleksi fitur *query expansion ranking*, diperoleh akurasi tertinggi sebesar $97,09\%$ pada

nilai k -fold = 7, seleksi fitur = 1000, 1300, 1500, C = 10, dan gamma = 0,1; 0,01; 0,001.

5. Akurasi tertinggi yang didapatkan untuk aplikasi agoda menggunakan seleksi fitur dengan rasio 1000, 1300, 1500 dan tanpa seleksi fitur adalah sama yaitu 96,20% pada nilai k -fold = 9, C = 1, dan gamma = 0,1; 0,01; 0,001.
6. Akurasi tertinggi yang didapatkan untuk aplikasi traveloka menggunakan seleksi fitur dengan rasio 1000, 1300, 1500 dan tanpa seleksi fitur adalah sama yaitu 97,09% pada nilai k -fold = 7, C = 10, dan gamma = 0,1; 0,01; 0,001.
7. Seleksi fitur *query expansion ranking* tidak berpengaruh dalam peningkatan akurasi menggunakan metode *support vector machine* sehingga tidak perlu menggunakan seleksi fitur *query expansion ranking* untuk melakukan analisis sentimen pada ulasan aplikasi agoda dan traveloka.

5.2 Saran

Saran yang diberikan untuk penelitian ini, yaitu:

1. Menambahkan kata pada kamus normalisasi dan *Stopword* agar data yang dihasilkan pada *preprocessing* semakin optimal.
2. Menggunakan variasi kernel yaitu RBF, Sigmoid, dan Polynomial agar dapat menjadi perbandingan dengan kernel lainnya.
3. Mengoptimalkan penggunaan seleksi fitur agar akurasi yang didapatkan meningkat.

DAFTAR PUSTAKA

- Buntoro, G. A. (2017). Analisis Sentimen Calon Gubernur DKI Jakarta 2017 Di Twitter. In *Integer Journal* (Vol. 2, Issue 1). <https://t.co/jrvMsgBdH>
- Buttar, P., Kaur, J., & Kaur Buttar, P. (2018). *A Systematic Review on Stopword removal Algorithms*. <http://www.ijfrcsce.org>
- Dadgar, S., Araghi, M., & Farahani, M. (2016). *A novel text mining approach based on TF-IDF and Support vector machine for news classification*. <https://doi.org/10.1109/ICETECH.2016.7569223>
- Duei Putri, D., Nama, G. F., & Sulistiono, W. E. (2022). Analisis Sentimen Kinerja Dewan Perwakilan Rakyat (DPR) Pada Twitter Menggunakan Metode Naive Bayes Classifier. *Jurnal Informatika Dan Teknik Elektro Terapan*, 10(1). <https://doi.org/10.23960/jitet.v10i1.2262>
- Faihah, R. T. (n.d.). *MAKALAH DATA MINING SUPPORT VECTOR MACHINE (SVM)*.
- Fauzi, M. A. (n.d.). *Text Pre-Processing*.
- Fauzi, M. A., & Adinugroho, S. (2018). *Analisis Sentimen Pariwisata di Kota Malang Menggunakan Metode Naive Bayes dan Seleksi Fitur Query expansion ranking Automatic Essay Scoring View project Food Image Classification, Retrieval, and Analysis View project*. <https://www.researchgate.net/publication/322959527>
- Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8), 861–874. <https://doi.org/10.1016/j.patrec.2005.10.010>
- Hirst, G., Seri, E., Liu, B., Opini, P., & Sentimen, A. (n.d.). *Analisis Sentimen dan KULIAH SINTESIS PADA TEKNOLOGI BAHASA MANUSIA KULIAH SINTESIS PADA TEKNOLOGI BAHASA MANUSIA TEKNOLOGI BAHASA MANUSIA KULIAH SINTESIS PADA*.
- Naser, M. Z., & Alavi, A. H. (n.d.). *Insights into Performance Fitness and Error Metrics for Machine Learning*. www.mznaser.com
- Nti, I. K., Nyarko-Boateng, O., & Aning, J. (2021). Performance of Machine Learning Algorithms with Different K Values in K-fold CrossValidation. *International Journal of Information Technology and Computer Science*, 13(6), 61–71. <https://doi.org/10.5815/ijitcs.2021.06.05>
- Nugroho, A. S., Witarto, A. B., & Handoko, D. (2003). *Support vector machine-Teori dan Aplikasinya dalam Bioinformatika 1*. <http://asnugroho.net>
- Nurjannah, M., Fitri Astuti, I., & Program Studi, D. (2013). PENERAPAN ALGORITMA TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF) UNTUK TEXT MINING Mahasiswa S1 Program Studi Ilmu Komputer FMIPA Universitas Mulawarman 2,3). In *Jurnal Informatika Mulawarman* (Vol. 8, Issue 3).

- Parlar, T., & Ozel, S. A. (2016, September 19). A new feature selection method for sentiment analysis of Turkish reviews. *Proceedings of the 2016 International Symposium on INnovations in Intelligent SysTems and Applications, INISTA 2016*. <https://doi.org/10.1109/INISTA.2016.7571833>
- Parlar, T., Öznel, S. A., & Song, F. (2018). QER: a new feature selection method for sentiment analysis. *Human-Centric Computing and Information Sciences*, 8(1). <https://doi.org/10.1186/s13673-018-0135-8>
- Pravina, A. M., Cholissodin, I., & Adikara, P. P. (2019). *Analisis Sentimen Tentang Opini Maskapai Penerbangan pada Dokumen Twitter Menggunakan Algoritme Support vector machine (SVM)* (Vol. 3, Issue 3). <http://j-ptiik.ub.ac.id>
- Rahman Isnain, A., Indra Sakti, A., Alita, D., & Satya Marga, N. (2021). SENTIMEN ANALISIS PUBLIK TERHADAP KEBIJAKAN LOCKDOWN PEMERINTAH JAKARTA MENGGUNAKAN ALGORITMA SVM. *JDMSI*, 2(1), 31–37. <https://t.co/NfhnfMjtXw>
- Riyani, A., Zidny Naf'an #2, M., & Burhanuddin, A. (2019). Penerapan Cosine Similarity dan Pembobotan TF-IDF untuk Mendekripsi Kemiripan Dokumen. In *JLK* (Vol. 2, Issue 1).
- Sohrabi, M. K., & Hemmatian, F. (2019). An efficient *preprocessing* method for supervised sentiment analysis by converting sentences to numerical vectors: a twitter case study. *Multimedia Tools and Applications*. <https://doi.org/10.1007/s11042-019-7586-4>
- Tala, F. Z. (n.d.). *A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia*.
- Perlakuan Pajak Atas Bisnis Online Travel Agent (online-pajak.com) (Diakses tanggal 14 Oktober 2022)
- <https://tirto.id/traveloka-pelonggaran-ppkm-penjualan-tiket-transportasi-naik-56-gslR> (Diakses tanggal 14 Oktober 2022)
- <https://bwhotelier.businessworld.in/article/Inbound-travel-searches-grew-102-between-January-and-May-this-year/05-07-2022-435581/> (Diakses tanggal 14 Oktober 2022)
- Agoda: Book Hotels and Flights - Apps on Google Play (Diakses tanggal 22 Oktober 2022)
- Traveloka: Book Flight & Hotel - Apps on Google Play (Diakses tanggal 22 Oktober 2022)