

**Analisis Sentimen Data Ulasan Menggunakan Algoritma *Support Vector Machine*
(Studi Kasus: Aplikasi Iflix)**

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Sastra-1
Pada Program Studi Teknik Industri Fakultas Teknologi Industri



Disusun oleh:

Salwa Kamilia Hasna (17522136)

**JURUSAN TEKNIK INDUSTRI
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2021

SURAT KETERANGAN PENELITIAN



FAKULTAS
TEKNIK INDUSTRI

Gedung KH. Mas Mansour
Jl. Kaliurang Km 14,5 Yogyakarta
Telp. (0274) 895287, 898444 ext 2511;
Fax. (0274) 895007

SURAT KETERANGAN PENELITIAN

Nomor : 221/A/Ka.Lab DATMIN/FTI-UIII/VI/2021

Assalamu'alaikum Warahmatullahi Wabarakatuh

Kami yang bertanda tangan dibawah ini, menerangkan bahwa mahasiswa dengan keterangan sebagai berikut :

Nama : Salwa Kamilia Hasna
No. Mhs : 17522136
Dosen Pembimbing : Andrie Pasca Hendradewa, S.T., M.T.

Telah selesai melaksanakan penelitian yang berjudul "Analisis Sentimen Data Ulasan Menggunakan Algoritma Support Vector Machine (Studi Kasus : Aplikasi Iflix)" di Laboratorium Data Mining, Program Studi Teknik Industri, Fakultas Teknologi Industri, Universitas Islam Indonesia tercatat mulai tanggal 1 Januari sampai dengan tanggal 17 Maret 2021.

Demikian surat keterangan kami keluarkan, agar dapat dipergunakan sebagaimana mestinya.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Dikeluarkan : di Yogyakarta
Tanggal : 09 Juni 2021

Mengetahui,
Kepala Lab. Data Mining

Annisa Uswatun Khasanah, S.T., M.Sc



PERNYATAAN KEASLIAN

Demi Allah, saya akui karya ini merupakan karya saya sendiri kecuali kutipan dan ringkasan yang setiap salah satunya telah dicantumkan sumbernya. Jika ditemukan di kemudian hari ternyata terbukti pengakuan saya ini tidak benar dan melanggar peraturan yang sah dalam karya tulis dan hak kekayaan intelektual maka saya bersedia ijazah yang saya terima untuk ditarik oleh Universitas Islam Indonesia.

Yogyakarta, 7 Juni 2021



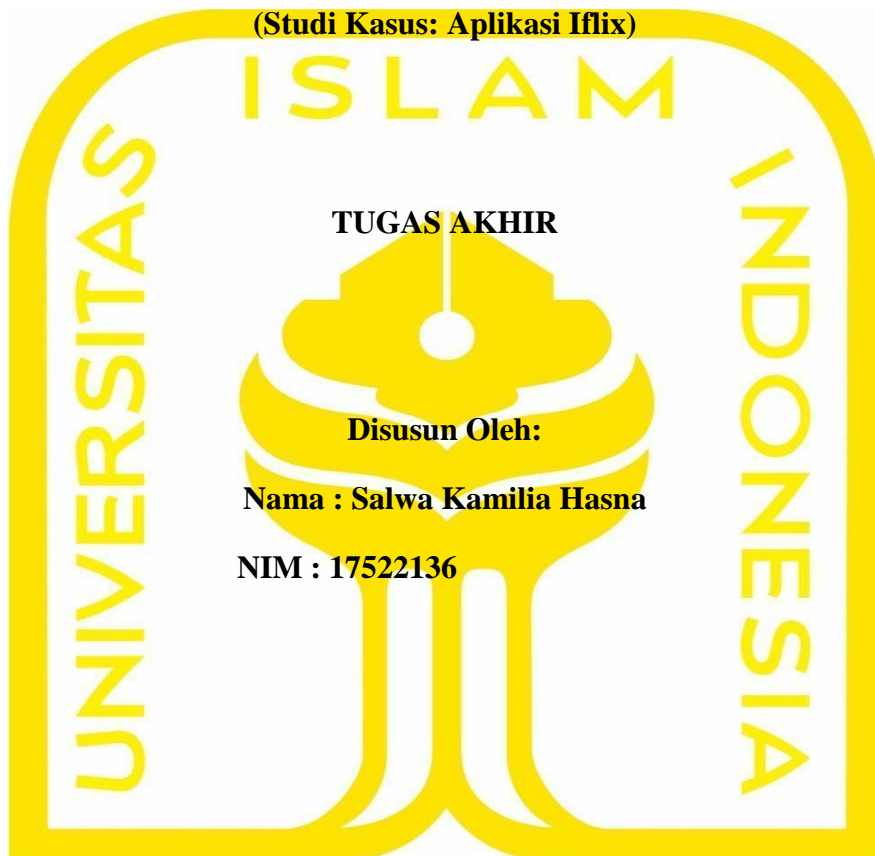
Salwa Kamilia Hasna

الجامعة الإسلامية
الاستدراكية

LEMBAR PENGESAHAN PEMBIMBING

Analisis Sentimen Data Ulasan Menggunakan Algoritma *Support Vector Machine*

(Studi Kasus: Aplikasi Iflix)



TUGAS AKHIR

Disusun Oleh:

Nama : Salwa Kamilia Hasna

NIM : 17522136



Yogyakarta, 7 Juni 2021

Dosen Pembimbing

A handwritten signature in black ink, which appears to read 'Andrie Pasca Hendradewa', is written over the text 'Dosen Pembimbing'.

Andrie Pasca Hendradewa, S.T., M.T

LEMBAR PENGESAHAN PENGUJI

ANALISIS SENTIMEN DATA ULASAN MENGGUNAKAN ALGORITMA SUPPORT VECTOR MACHINE (STUDI KASUS: APLIKASI IFLIX)

TUGAS AKHIR

Oleh :

Nama : Salwa Kamilia Hasna
NIM : 17522136

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk
memperoleh gelar Sarjana Strata-1 Teknik Industri
Yogyakarta, Agustus 2021

Tim Penguji

Andrie Pasca Hendradewa, S.T., M.T.

Ketua

Winda Nur Cahyo, S.T., M.T., Ph.D

Anggota I

Abdullah 'Azzam, S.T., M.T.

Anggota II

Mengetahui,

Ketua Program Studi Teknik Industri

Fakultas Teknologi Industri

Universitas Islam Indonesia



Dr. Taufiq Humawan, S.T., M.M.

HALAMAN PERSEMBAHAN

Alhamdulillahirabbil'alamiin

Dengan Nama Allah yang Maha Pengasih dan Maha Penyayang. Segala puji dan syukur tiada henti atas kehadiran Allah SWT atas nikmat yang telah diberikan. Skripsi ini saya persembahkan kepada kedua orang tua yang tercinta yang selalu memberikan doa, dukungan dan pengorbanan yang tiada henti diberikan kepada penulis.



HALAMAN MOTTO

“Maka sesungguhnya sesudah kesulitan itu ada kemudahan. Sesungguhnya sesudah kesulitan itu ada kemudahan” (QS.Al-Insyirah:5-6)



KATA PENGANTAR



Assalamu 'alaikum Warahmatullahi Wabarakatuh

Alhamulillahirobbil'alamin, segala puji dan syukur bagi Allah SWT atas berkah, rahmat, dan nikmat yang diberikan kepada hamba-Nya. Tidak lupa shalawat dan salam senantiasa tercurah kepada Baginda Nabi Muhammad SAW atas petunjuk untuk selalu berada di jalan-Nya sehingga penulis dapat menyelesaikan laporan yang berjudul “Analisis Analisis Sentimen Data Ulasan Menggunakan Algoritma *Support Vector Machine* (Studi Kasus: Aplikasi Iflix)” dengan baik serta dapat memenuhi salah satu persyaratan dalam menempuh gelar Sarjana Teknik (S.T) di Fakultas Teknologi Industri Universitas Islam Indonesia.

Dalam penulisan laporan tugas akhir ini penulis menyadari kiranya tidak akan selesai tanpa bantuan dari beberapa pihak yang terus membantu dan mendorong penulis untuk menyelesaikannya. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. Ir. Hari Purnomo, M.T selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
2. Bapak Muhammad Ridwan Andi Purnomo, S.T.,M.Sc., Ph.D. selaku Ketua Jurusan Teknik Industri Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Bapak Taufiq Immawan,S.T.,M.M. selaku Ketua Program Studi Teknik Industri Universitas Islam Indonesia.
4. Bapak Andrie Pasca Hendradewa,S.T.,M.T., selaku Dosen Pembimbing yang telah memberikan ilmu, bimbingan, saran serta waktunya dalam penyusunan Tugas Akhir.
5. Kedua orang tua, saudara dan keluarga yang selalu memberikan semangat, perhatian serta doa kepada penulis.
6. Serta seluruh pihak yang tidak dapat penulis sebutkan satu persatu yang telah membantu proses pelaksanaan penelitian dan penulisan laporan tugas akhir.

Semoga kebaikan serta dukungan dari semua pihak yang telah diberikan kepada penulis senantiasa menjadi amal ibadah yang dibalas berlipat ganda oleh Allah SWT.

Demikian laporan Tugas Akhir ini, penulis menyadari bahwa masih banyak kekurangan. Oleh karena itu penulis mengharapkan saran dan kritik yang bersifat membangun untuk Tugas Akhir ini. Semoga Tugas Akhir ini dapat bermanfaat bagi semua pihak.

Wassalamu 'alaikum Warohmatullahi Wabarakatuh



Yogyakarta, 7 Juni 2021

A handwritten signature in black ink, appearing to be 'Skus' or similar, written in a cursive style.

Salwa Kamilia Hasna

ABSTRAK

Pandemi *Coronavirus Disease 2019* (Covid-19) dinyatakan telah menyebar ke seluruh penjuru dunia pada akhir tahun 2019 lalu. Hal tersebut memberikan dampak yang cukup signifikan kepada seluruh negara di dunia tidak terkecuali Indonesia. Pandemi Covid-19 menyebabkan kemunduran pada industri perfilman Indonesia. Pandemi ini mengakibatkan berbagai proses *recording* film yang harus ditunda, phk dari puluhan ribu pekerja industri film hingga penutupan bioskop – bioskop di Indonesia. Para pelaku industri film berusaha melakukan adaptasi dengan memanfaatkan perkembangan kemajuan teknologi digital dalam memproduksi dan mendistribusikan film mereka melalui media streaming online. Iflix merupakan suatu teknologi streaming online yang dapat dimanfaatkan oleh para pelaku industri perfilman. Aplikasi Iflix memiliki rating sebesar 4.1 dari 611.711 ulasan. Pada penelitian ini data ulasan yang digunakan hanya data ulasan sebesar 4.501 ulasan yang telah ter update pada rentang waktu dari 1 Januari 2021 hingga 17 Maret 2021. Ulasan dari pengguna aplikasi dapat bersifat positif dan negatif data akan melalui proses klasifikasi dengan menggunakan metode *support vector machine*. Hasil proses klasifikasi dengan menggunakan metode SVM dengan akurasi tertinggi adalah skenario yaitu data latih 70 % dan data uji 30 % sebesar 93,45 %. Selanjutnya pada proses visualisasi memiliki hasil pada sentimen positif yaitu film, bagus, aplikasi, gambar, menghibur dan nonton. Sedangkan visualisasi pada kelas sentimen negatif yaitu film, vip, aplikasi, iflix, jelek, lama, nonton, dan pulsa. Hasil ulasan negatif di analisis dengan menggunakan diagram *fishbone* yang terbagi menjadi beberapa faktor yaitu *price, product, people, process* dan *promotion*.

Kata Kunci: analisis sentimen, Iflix, *Support Vector Machine* (SVM), *fishbone diagram*

DAFTAR ISI

SURAT KETERANGAN PENELITIAN	ii
PERNYATAAN KEASLIAN	iii
LEMBAR PENGESAHAN PEMBIMBING	iv
LEMBAR PENGESAHAN PENGUJI	v
HALAMAN PERSEMBAHAN	vi
HALAMAN MOTTO	vii
KATA PENGANTAR	viii
ABSTRAK	x
DAFTAR TABEL	xiv
DAFTAR GAMBAR	xv
BAB 1	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	6
1.3 Batasan Masalah	7
1.4 Tujuan Penelitian	7
1.5 Manfaat Penelitian	8
1.6 Sistematika Penulisan	8
BAB II	10
2.1 Kajian Induktif	10
2.2 Kajian Deduktif	15
2.2.1 Iflix	15
2.2.2 <i>Google Play</i>	16
2.2.3 <i>Online Review</i>	16
2.2.4 <i>Web Scraping</i>	17
2.2.5 <i>Machine Learning</i>	18
2.2.6 Analisis Sentimen	19
2.2.7 <i>Text Mining</i>	19
2.2.8 <i>Pre Processing</i>	20
2.2.9 Klasifikasi	21
2.2.10 Fitur dan Pembobotan	22
2.2.11 <i>Support Vector Machine (SVM)</i>	22
2.2.12 Asosiasi Kata	27

2.2.13	Diagram Fishbone	27
2.2.14	Python	27
BAB III		29
3.1	Objek dan Subjek Penelitian	29
3.2	Populasi dan Sampel Penelitian	29
3.3	Pengumpulan Data	29
3.3.1	Metode Pengumpulan Data	29
3.3.2	Jenis dan Sumber Data	30
3.4	Metode Analisis Data	30
3.5	Variabel Penelitian	31
3.6	Alur Penelitian	32
Berikut merupakan alur dari penelitian ini:		32
BAB IV		36
4.1	Pengumpulan Data	36
4.2	Pengolahan Data	39
4.2.1	Analisis Deskriptif	39
4.2.2	<i>Pre Processing Data</i>	42
Pada Tabel 4.9 merupakan <i>pseudo code</i> pada tahap <i>Stopword Removal</i> .		45
4.2.3	Pelabelan	46
4.2.4	Pembagian Data Latih dan Data Uji	50
4.2.5	Klasifikasi dengan Metode <i>Support Vector Machine</i>	52
4.2.6	Visualisasi dan Asosiasi	57
4.2.7	Interpretasi Hasil	67
BAB V		68
5.1	Analisis Deskriptif Data Awal	68
5.2	Hasil Klasifikasi SVM	69
5.3	Visualisasi	70
5.3.1	Sentimen Positif	71
5.3.2	Sentimen Negatif	72
5.4	Fishbone Diagram	73
BAB VI		75
6.1	Kesimpulan	75
6.2	Saran	76

DAFTAR PUSTAKA
LAMPIRAN

78
84



DAFTAR TABEL

Tabel 2 1 Kajian Induktif.....	13
Tabel 3 1 Variabel Penelitian.....	31
Tabel 4 1 Hasil <i>Scraping</i> Data.....	38
Tabel 4 2 Proses <i>Spelling Normalization</i>	42
Tabel 4 3 <i>Pseudo Code Spelling Normalization</i>	42
Tabel 4 4 Proses Case Folding.....	43
Tabel 4 5 <i>Pseudo Code Case Folding</i>	43
Tabel 4 6 Proses Tokenizing.....	44
Tabel 4 7 <i>Pseudo Code Tokenizing</i>	44
Tabel 4 8 <i>Stopword Removal</i>	45
Tabel 4 9 Pseudo Code Stopword Removal.	45
Tabel 4 10 Proses Pelabelan dengan Lexicon.....	46
Tabel 4 11 Hasil Pelabelan	47
Tabel 4 12 Hasil Akhir Pelabelan	47
Tabel 4 13 <i>Pseudo Code</i> Proses Pelabelan	47
Tabel 4 14 Pembagian Data Uji dan Data Latih	51
Tabel 4 15 <i>Pseudo Code</i> Pembagian Data <i>Training</i> dan <i>Testing</i>	51
Tabel 4 16 Hasil Klasifikasi 70:30	54
Tabel 4 17 <i>Pseudo Code</i> Klasifikasi SVM	54
Tabel 4 18 <i>Pseude Code</i> Visualisasi Positif dan Negatif.....	58
Tabel 4 19 <i>Association Rule</i> Sentimen Positif.....	63
Tabel 4 20 <i>Association Rule</i> Sentimen Negatif	65
Tabel 5 1 Perbandingan Performasi Klasifikasi	69

DAFTAR GAMBAR

Gambar 1. 1 Data Pengguna Smartphone Dunia	4
Gambar 2 1 Hyperplane Terbaik	25
Gambar 2 2 Transformasi Data.....	25
Gambar 2 3 SVM Non Linier Kernel Map	26
Gambar 3 1 Alur Penelitian	32
Gambar 4 1 Ekstensi Data Miner.....	36
Gambar 4 2 Website Aplikasi Iflix	37
Gambar 4 3 Proses Row.....	37
Gambar 4 4 Proses Cols.....	38
Gambar 4 5 Membuat Recipe	38
Gambar 4 6 Grafik Jumlah Komentar.....	40
Gambar 4 7 Jumlah Rating	40
Gambar 4 8 Perbandingan Rating Ulasan.....	41
Gambar 4 9 Hasil Klasifikasi.....	53
Gambar 4 10 Persebaran Kata Sentimen Positif.....	62
Gambar 4 11 <i>Wordcloud</i> Visualisasi Sentimen Positif.....	62
Gambar 4 12 Persebaran Kata Sentimen Negatif	64
Gambar 4 13 <i>Wordcloud</i> Visualisasi Sentimen Negatif	65
Gambar 4 14 Diagram Fishbone Ulasan Negatif.....	67

BAB 1

PENDAHULUAN

1. 1 Latar Belakang

Pandemi *Coronavirus Disease 2019* (Covid-19) dinyatakan telah menyebar ke seluruh penjuru dunia pada akhir tahun 2019 lalu. Hal tersebut memberikan dampak yang cukup signifikan kepada seluruh negara di dunia tidak terkecuali Indonesia. Pada bulan Maret 2020, Indonesia pertama kali menyatakan bahwa wabah Covid-19 telah menyebar ke Indonesia. WHO atau *World Health Organisation* menyatakan wabah ini merupakan sebuah pandemi global yang dapat menular secara cepat. Hal tersebut membuat berbagai negara melakukan pencegahan dengan penetapan berbagai peraturan seperti melakukan *social distancing*, menetapkan *lockdown* dan berbagai peraturan lainnya. Pandemi ini menimbulkan efek yang cukup berpengaruh dari sisi kesehatan, masalah sosial, ekonomi sampai kepada pelaku usaha. Menurut Badan Pusat Statistika menyatakan laju pertumbuhan ekonomi pada Kuartal I yaitu Januari hingga Maret 2020 hanya bertumbuh sebesar 2,97%. Angka tersebut merambat dari nilai yang didapatkan pada Kuartal VI 2019 yaitu 4,97%. Sedangkan pada Kuartal II Tahun 2020 laju pertumbuhan ekonomi Indonesia minus 5,32%. Pertumbuhan ekonomi di Indonesia dapat dinyatakan mengalami pertumbuhan negatif (Wuryandani, 2020). Hal tersebut memberikan pengaruh yang cukup besar terhadap perekonomian Indonesia di berbagai sektor seperti pariwisata, restoran, bioskop, mall dan ritel, otomotif dan MICE (finance.detik.com, 2021)

Pada tahun 2018 hingga 2019 lalu, industri perfilman Indonesia mengalami peningkatan yang cukup tinggi. Hal tersebut ditunjukkan dengan jumlah penonton yang mulanya hanya sebesar 10 juta penonton di seluruh film Indonesia meningkatkan hingga 52 juta penonton. Peningkatan industri film ini tentu saja memberikan pengaruh positif bagi perekonomian Indonesia seperti memberikan

lowongan pekerjaan, investasi dari luar untuk film Indonesia hingga pengaruh positif lainnya (republika.co.id, 2021). Namun setelah pandemi ini terjadi pada akhir tahun kemarin, Industri film kini mengalami pukulan keras. Perfilman Indonesia yang sebelum pandemi sedang menuju puncak kejayaannya, hal tersebut terjadi dikarenakan kerja keras dari puluhan ribu pekerja industri film. Film Indonesia membawa wajah Indonesia kepenjuru dunia (kerincitime.co.id, 2021). Namun terjadinya pandemi Covid-19 menyebabkan kemunduran pada industri perfilman Indonesia. Pandemi ini mengakibatkan berbagai proses *recording* film yang harus ditunda, phk dari puluhan ribu pekerja industri film hingga penutupan bioskop – bioskop di Indonesia. Penutupan gedung bioskop ini dilakukan untuk mencegah penyebaran Covid-19 pada ruang tertutup. Bioskop merupakan suatu tempat dimana berbagai hasil dari industri film dapat ditunjukkan kepada para penonton.

Pada Oktober 2020 lalu terdapat beberapa bioskop yang mulai dibuka kembali, namun pembukaan kembali bioskop tidak dapat memberikan pendapatan seperti sebelumnya dikarenakan sepi penonton dan peraturan *social distancing* yang tetap harus dilakukan (finance.detik.com, 2021). Hal tersebut yang mengharuskan para tokoh industri film untuk memikirkan strategi terbaik untuk bertahan di kondisi saat ini. Industri film Indonesia melakukan usaha adaptasi di tengah masa pandemi yang sedang terjadi. Para pelaku industri film memanfaatkan perkembangan kemajuan teknologi digital dalam memproduksi dan mendistribusikan film mereka. Dalam upaya melakukan usaha untuk memperbaiki industri film di masa saat ini, pelaku industri memanfaatkan perkembangan teknologi. Salah satu teknologi yang dimanfaatkan adalah streaming online.

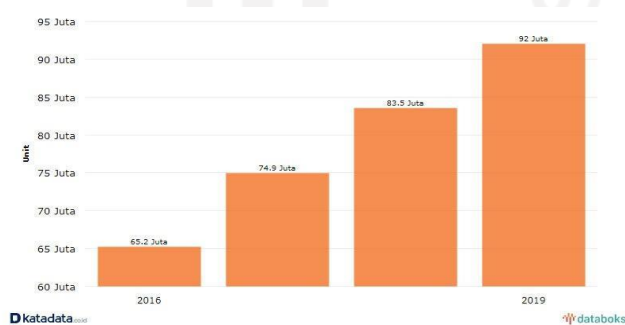
Streaming Online ini dilakukan untuk mempermudah para penonton untuk dapat terhibur dan tetap mematuhi peraturan *social distancing* dengan menikmati film yang tersedia dari rumah. Streaming online ini dilengkapi dengan fitur berlangganan untuk film – film tertentu sehingga hal tersebut membantu para tokoh industri film untuk menuangkan karya – karyanya serta mendapatkan keuntungan. Perkembangan streaming online video di Indonesia semakin meningkat. Hal tersebut didorong oleh jumlah peminat yang semakin meningkat. Pada saat pandemi saat ini jumlah pengguna platform streaming online di tanah air meningkat hingga lebih dari 50 % dan pada awal bulan Maret 2020 mengalami peningkatan hingga 20 %. Oleh karena hal tersebut diyakini kedepannya pengguna platform

streaming online akan terus meningkat didukung dengan jumlah pengguna internet di Indonesia sudah mencapai 173 juta jiwa atau setengah dari populasi yang ada (teknologi.bisnis.com, 2020). Iflix merupakan suatu teknologi streaming online yang dapat dimanfaatkan oleh para pelaku industri perfilman. Pemanfaat platform streaming online sebagai sumber hiburan memberikan berbagai keuntungan seperti mengurangi biaya konsumtif hiburan yang awalnya diperlukan biaya transportasi, makan dan tiket nonton ketika bioskop namun sekarang hanya diperlukan biaya internet dan berlangganan platform streaming online. Platform streaming online memberikan kemudahan layanan, dan kepraktisan. Penggunaan platform streaming online memberikan peningkatan pendapatan pada tahun 2020 yang diperkirakan naik 28,9% menjadi 140 juta dolar AS (Suryanto, 2020). Di Indonesia terdapat beberapa platform streaming online legal di Indonesia, antara lain Ifix, Netflix, dan platform lainnya. Pada tahun 2019 tercatat jumlah pengguna dari Iflix adalah 17 juta pelanggan dan 14,5 juta pelanggan pengguna Netflix. Namun jumlah pengguna Netflix meningkat pada dua tahun terakhir ini (Barker, 2020). Netflix memiliki jumlah pengguna yang lebih banyak di dibandingkan dengan Iflix. Namun apabila dilihat dari segi biaya berlangganan, Netflix memiliki biaya berlangganan yang berkali – kali lipat lebih mahal di dibandingkan Iflix. Dimana jumlah biaya berlangganan Netflix sebesar Rp. 120.000/bulan sedangkan biaya berlangganan Iflix sebesar Rp. 39.000/bulan. Selain itu Iflix memberikan penawaran gratis berlangganan 1 bulan di awal penggunaan. Biaya berlangganan Iflix yang lebih mudah dijangkau memudahkan Iflix dapat digunakan untuk berbagai kalangan. Berdasarkan pernyataan yang dilakukan oleh pihak MD Picture, Iflix telah melakukan kolaborasi atau kerjasama dengan MD Picture untuk menayangkan konten – konten lokal unggulan. Kerjasama ini dilakukan dalam rangka mendukung industri perfilman lokal Indonesia, sehingga mendapat tempat di masyarakat global, terutama penonton Asia (mdpictures.com, 2020).

Iflix merupakan salah satu platform berbasis aplikasi dan website yang menyediakan layanan *video on demand* (VOD) gratis dan berlangganan yang berfokus pada pemasaran. Iflix saat ini tersedia untuk berbagai negara di Asia yaitu Malaysia, Indonesia, Filipina, Thailand, Sri Lanka, Brunei, Pakistan, Myanmar, Vietnam, Kamboja, Nepal, Bangladesh dan Maladewa. Pada bulan Maret 2019,

pengguna Iflix memiliki lebih dari 25 juta pelanggan di negara cakupannya. Iflix berdiri sejak tahun 2014 lalu dan masuk ke Indonesia pada akhir tahun 2015. Jenis konten yang tersedia pada aplikasi ini adalah Serial TV, Film Box Office hingga acara khusus anak – anak. Aplikasi Iflix dapat dijalankan pada *smartphone* maupun *smart tv*. Aplikasi Iflix dapat dijalankan pada *smartphone* maupun *smart tv*. Ifix diketahui merupakan salah satu aplikasi seluler hiburan yang berada di peringkat keempat pada perusahaan populer layanan *video on demand* dengan persentase sebesar 24,35% di Indonesia.

Perkembangan teknologi informasi yang terus berkembang pesat dari masa ke masa memberikan pengaruh penting dalam kehidupan bermasyarakat. Teknologi informasi yang berkembang pesat selaras dengan berkembangnya teknologi era digital di dunia. Kemajuan dibidang teknologi mendorong manusia untuk mengembang teknologi – teknologi dengan tujuan untuk mempermudah kehidupan manusia. Kemajuan tersebut menjadikan masyarakat tergantung akan teknologi khususnya *smartphone*. *Smartphone* adalah ponsel yang memiliki kemampuan tingkat tinggi, dan umumnya memiliki fungsi menyerupai komputer (Mujahid, 2018). Pada gambar 1.1, menjelaskan bahwa terdapat peningkatan pengguna *smartphone* di dunia setiap tahunnya, peningkatan yang terjadi antara tahun 2018 dan tahun 2019 sekitar 20 juta pengguna (Katadata, 2019).



Gambar 1. 1 Data Pengguna Smartphone Dunia

Sumber: (APJI,2020)

Selain *smartphone* terdapat teknologi lain yang dapat dimanfaatkan yaitu *Smart TV*. *Smart TV* merupakan TV yang mempunyai prosesor atau otak tambahan sehingga dapat dioperasikan seperti mengoperasikan komputer. *Smart TV* merupakan TV yang mampu berinteraksi dengan menggunakan internet yang memungkinkan untuk mengakses layanan online seperti video streaming, pesan

instan dan social media. Pengguna smart TV di Indonesia saat ini mencapai 2 % dari populasi di Indonesia. Kedua teknologi tersebut dapat digunakan untuk mengakses berbagai platform streaming online yang ada, seperti Iflix, Netflix dan platform *streaming* online lainnya.

Berdasarkan permasalahan yang telah diuraikan diatas, dimana imbas dari pandemic Covid-19 yang menyerang Indonesia pada awal tahun 2020 lalu, mengakibatkan industri perfilman Indonesia yang terpuak mundur dan mengharuskan para tokoh industri maupun yang berkaitan didalamnya untuk dapat beradaptasi dan bertahan. Pertahanan dan adaptasi yang dilakukan oleh para tokoh industri adalah dengan memanfaatkan kemajuan teknologi digital yaitu streaming online yang terus meningkat meski di tengah pandemi. Iflix merupakan suatu aplikasi yang menjadi suatu jawaban sebagai media para tokoh industri film untuk menuangkan bakat dan karya – karya serta menjadi media penghibur para penonton film Indonesia pada masa pandemi saat ini. Untuk mempertahankan rasa nyaman dari aplikasi ini maka aplikasi perlu dirancang sebagai suatu aplikasi yang *user friendly* serta memberikan pelayanan sebaik mungkin. Berdasarkan yang tercatat pada *website Google Play*, produk Aplikasi Iflix memiliki rating sebesar 4.1 dari 611.711 ulasan. Hal tersebut menunjukkan bahwa aplikasi tersebut masih dapat melakukan peningkatan kualitas serta inovasi untuk mencapai rating yang lebih tinggi. *Google Play* memiliki fitur yang berisi ulasan atau komentar dari pengguna aplikasi yang ada di dalam *Google Play* seperti Iflix. Pada penelitian ini data ulasan yang digunakan hanya data ulasan yang telah ter update pada rentang waktu dari 1 Januari 2021 hingga 17 Maret 2021. Rentang waktu itu dipilih, dikarenakan pada awal bulan januari dilakukan pembaharuan aplikasi dan pembaharuan kedua dilakukan pada tanggal 15 Maret 2021. Ulasan dari pengguna aplikasi dapat bersifat positif berupa saran maupun negatif yang berupa keluhan terhadap aplikasi.

Ulasan – ulasan tersebut dapat memberikan pengaruh bagi aplikasi Iflix dalam melakukan perbaikan terhadap aplikasi. Namun ulasan – ulasan tersebut perlu dilakukan analisis untuk mengetahui pola dari tanggapan yang diberikan oleh pengguna. Metode yang dapat digunakan dalam penelitian ini adalah Text Mining, Dimana *Text Mining* sendiri merupakan suatu analisis teks dengan tujuan menggali informasi yang berkualitas dari suatu rangkaian teks (Kamber, 2006). Analisis sentimen bertujuan untuk menganalisis pendapat, sentiment, evaluasi, sikap,

penilaian, dan emosi seseorang berkenaan dengan suatu topik, produk, layanan, organisasi, individu, ataupun kegiatan tertentu lainnya (Liu, 2010). Analisis sentimen akan menghasilkan suatu kumpulan informasi berdasarkan ulasan – ulasan yang telah diberikan oleh pengguna aplikasi Iflix. Dimana informasi tersebut akan digunakan sebagai sumber atau acuan dalam melakukan perbaikan terhadap aplikasi tersebut. Selanjutnya persepsi pengguna tersebut perlu dilakukan klasifikasi ulasan negatif maupun ulasan positif. Dimana proses klasifikasi akan dilakukan dengan menggunakan metode *Support Vector Machine (SVM)*. *Support Vector Machine (SVM)* memiliki suatu konsep sentral dalam mengklasifikasikan data, yaitu menentukan *hyperplane* terbaik untuk memberi jarak atau pemisah antara dua kelas yang telah ditentukan (Kowalczyk, 2017). Pada penelitian yang dilakukan sebelumnya diketahui bahwa *Support Vector Machine* memiliki nilai akurasi yang lebih unggul 6,85 % dibandingkan metode *Naïve Bayes* (Herlinawati, 2020). Selain itu pada penelitian lainnya diketahui bahwa metode *Support Vector Machine* memiliki nilai akurasi yang lebih tinggi sebesar 80,54% dibandingkan dengan metode *Naïve Bayes* sebesar 73,30% (Badr,E.M., et al,2019).

Setelah dilakukan proses klasifikasi, selanjutnya dilakukan proses visualiasi dengan menggunakan *wordcloud* dan asosiasi untuk mengetahui jumlah kata yang sering muncul pada keseluruhan ulasan yang diberikan pengguna aplikasi Iflix. Pada sentimen ulasan negatif yang dihasilkan akan dilakukan analisis menggunakan diagram *fishbone* untuk mengetahui faktor – faktor penyebabnya. Hasil dari *fishbone* diagram tersebut diharapkan dapat digunakan oleh Pihak manajemen Iflix dalam melakukan perbaikan serta mengetahui faktor – faktor apa saja yang menjadi keluhan atau pandangan dari pengguna aplikasi Iflix.

Penelitian ini dilakukan dengan tujuan untuk memenuhi tanggapan atau pendapat dari pengguna aplikasi Iflix terhadap pelayanan yang diberikan. Hasil penelitian ini diharapkan akan berfungsi sebagai sumber evaluasi serta pengembangan untuk meningkatkan sistem pelayanan dari Aplikasi Iflix.

1. 2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka rumusan masalah dari penelitian ini adalah sebagai berikut ini:

1. Bagaimana gambaran umum terhadap tanggapan dari pengguna aplikasi Iflix?
2. Bagaimana hasil dari implementasi metode *Support Vector Machine* (SVM) dalam melakukan klasifikasi sentimen data ulasan pengguna Aplikasi Iflix?
3. Bagaimana informasi yang dihasilkan dalam setiap klasifikasi dan asosiasi teks yang telah dilakukan?

1.3 Batasan Masalah

Batasan ruang lingkup penelitian yang dilakukan dalam penelitian ini adalah sebagai berikut:

1. Pada penelitian ini, data yang digunakan merupakan data ulasan mengenai Aplikasi Iflix pada *website Google Play*.
2. Data ulasan yang digunakan dalam penelitian yaitu data ulasan mulai 1 Januari 2021 – 17 Maret 2021.
3. Ulasan yang digunakan adalah ulasan yang menggunakan bahasa Indonesia.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut ini:

1. Mengetahui gambaran umum mengenai tanggapan dari pengguna Aplikasi Iflix.
2. Mengetahui hasil dari penerapan atau implementasi metode *Support Vector Machine* (SVM) dalam mengklasifikasikan sentimen data ulasan dari pengguna Aplikasi Iflix.
3. Mengetahui dan memperoleh informasi penting dari hasil klasifikasi dan asosiasi teks yang dilakukan.

1. 5 Manfaat Penelitian

Manfaat yang didapatkan dari penelitian ini adalah sebagai berikut ini:

1. Mengetahui gambaran umum dari data Aplikasi Iflix berdasarkan data ulasan di *website Google Play*.
2. Hasil dari penelitian ini dapat digunakan sebagai gambaran mengenai persepsi yang dirasakan oleh pengguna dari Aplikasi Iflix sehingga dapat digunakan sebagai acuan bagi pihak – pihak terkait dalam melakukan perbaikan dan peningkatan kualitas pelayanan dalam memenuhi kepuasan dari pengguna.

1. 6 Sistematika Penulisan

Penulisan tugas akhir ini ditulis secara sistematis berdasarkan akar permasalahan hingga solusi penyelesaian permasalahan yang dibagi ke dalam beberapa bab yang dijelaskan sebagai berikut ini:

BAB I PENDAHULUAN

Pada bab pendahuluan ini meliputi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

BAB II KAJIAN LITERATUR

Bab kajian literatur ini terdapat 2 sub bab utama yang meliputi kajian deduktif dan kajian induktif. Pada bab ini menjelaskan mengenai landasan teori yang digunakan sebagai bahan pendukung dalam memperkuat penelitian yang dilakukan. Kajian deduktif berisi mengenai landasan penelitian dalam bentuk fakta yang dilakukan oleh peneliti terdahulu, sedangkan kajian induktif berisi suatu bentuk dasar teori umum yang dijelaskan oleh peneliti terdahulu.

BAB III METODOLOGI PENELITIAN

Bab metodologi penelitian ini meliputi alur penelitian, objek penelitian dan metode yang digunakan dalam proses pengumpulan dan pengolahan data. Pada bab ini menjelaskan mengenai bagaimana proses penelitian yang dilakukan secara keseluruhan serta bagaimana data didapatkan dan diproses menggunakan metode yang telah ditetapkan sebelumnya.

BAB IV PENGUMPULAN DAN PENGOLAHAN DATA

Bab pengumpulan dan pengolahan data meliputi beberapa sub bab yaitu pengumpulan data dan pengolahan data. Pada bab ini, pengumpulan data dilakukan dengan tujuan memperoleh data yang akan digunakan dalam penelitian dengan menggunakan metode yang telah ditetapkan, sedangkan pengolahan data merupakan suatu proses dimana data mentah didapatkan, diolah hingga proses visualisasi baik berupa teks, tabel maupun grafik dari hasil yang didapatkan.

BAB V ANALISIS DAN PEMBAHASAN

Pada bab analisis dan pembahasan menjelaskan mengenai hasil dari pengolahan serta penelitian yang dilakukan.

BAB VI KESIMPULAN DAN SARAN

Pada bab ini, menjelaskan mengenai kesimpulan yang diambil dari penelitian yang dilakukan. Kesimpulan dalam penelitian ini menjawab rumusan masalah yang telah ditetapkan pada bab sebelumnya serta berisi mengenai rekomendasi atas hasil yang didapatkan untuk memperbaiki permasalahan yang terjadi.

BAB II

KAJIAN LITERATUR

2.1 Kajian Induktif

Pada penelitian ini dilakukan kajian literatur induktif yang menjelaskan berbagai informasi terkait penelitian terdahulu yang berkaitan dengan penelitian yang dilakukan. Berikut adalah kajian literature dari penelitian terdahulu:

Penelitian yang dilakukan Utama, et al. (2019) yang berjudul “Analisis Sentimen Sistem Ganjil Genap di Tol Bekasi Menggunakan Algoritma Support Vector Machine”, bertujuan untuk mengetahui tingkat *accuracy*, *recall* dan *precision* dari *opinion mining* yang dihasilkan dengan menggunakan algoritma SVM serta memberikan informasi sentiment masyarakat terhadap efektivitas sistem ganjil genap tol bekasi di medial sosial. Metode yang digunakan dalam penelitian ini adalah melakukan *text mining* pada komentar – komentar terkait mengenai efektivitas ganjil genap di tol bekasi pada *Twitter*, *Instagram* dan *Facebook*. Berdasarkan hasil penelitian, dengan menggunakan model SVM adalah didapatkan hasil *confusion matrix*, yaitu *accuracy* sebesar 78.18%, *precision* sebesar 74.03%, dan *sensitivity* atau *recall* sebesar 86,82%. Dengan melakukan penelitian ini dapat menyimpulkan bahwa dengan menggunakan Algoritma Support Vector Machine dapat menganalisis sentimen ganjil genap di tol bekasi.

Pada penelitian yang dilakukan Herlinawati, et al (2020) yang berjudul “Analisis Sentimen *Zoom Cloud Meeting* di Playstore Menggunakan naïve Bayes dan Support Vector Machine”, tujuan dari penelitian ini adalah untuk memprediksi label sentiment pada ulasan para pengguna aplikasi *zoom cloud meetings* di *Google Play Store* menggunakan metode *Naïve Bayes* dan *Support Vector Machine* sebagai model klasifikasi ulasan dan ulasan dibagi menjadi kategori positif dan negatif. Metode yang digunakan dalam penelitian ini adalah *Naïve Bayes* dan

Support Vector Machine. Hasil dari penelitian ini adalah evaluasi model dengan menggunakan 10 *fold cross validation* diperoleh nilai akurasi dan nilai AUC dari masing – masing algoritma yaitu NB nilai akurasi sebesar 74,37% dan nilai AUC adalah 0,659. Sedangkan untuk algoritma SVM nilai akurasi sebesar 81,22% dan nilai AUC sebesar 0,886. Dalam penelitian ini dapat diketahui nilai akurasi yang didapatkan algoritma *Support Vector Machine* (SVM) lebih unggul 6,85% dibandingkan algoritma *Naïve Bayes*.

Penelitian berjudul “Analisis Sentimen Aplikasi Go-Jek Menggunakan Metode SVM dan NBC (Studi Kasus: Komentar Pada *Play Store*)” yang ditulis oleh Anjasmoros, et al pada tahun 2020. Dimana penelitian ini bertujuan membangun suatu sistem yang dapat menghasilkan suatu informasi tentang perbandingan dari metode *Support Vector Machine* (SVM) dan *Naïve Bayes* berupa *score accuracy*, *recall* dan *precision*. Metode yang digunakan dalam penelitian ini adalah SVM dan *Naïve Bayes*. Hasil dari penelitian ini adalah *score accuracy*, *recall* dan *precision* yang paling tinggi adalah 0.987 dengan data testing sebanyak 50% dan 50 % data *training* dari 2.000 komentar menggunakan metode SVM kernel linear dan *score accuracy*, *recall* dan *precision* dengan menggunakan metode SVM kernel RBF, polynomial dan sigmoid adalah 0.486 dengan data testing sebanyak 40 % dan data *training* sebesar 60% dengan menggunakan metode SVM kernel RBF, polynomial dan sigmoid.

Pada penelitian berjudul “Analisis Sentimen Aplikasi Pembelajaran Online di *Play Store* pada Masa Pandemi Menggunakan Algoritma *Support Vector Machine*” yang ditulis oleh Erfina, et al pada tahun 2020. Tujuan dari penelitian ini adalah menganalisa komentar ulasan pengguna aplikasi E-learning sebagai variabel tambahan untuk menentukan aplikasi *E-learning* mana yang paling terbaik di *Play Store*. Metode yang digunakan dalam penelitian ini adalah *Support Vector Machine*, dimana algoritma SVM memiliki tingkat keakuratan yang dinilai cukup tinggi untuk menentukan aplikasi *E-learning* mana yang terbaik di *Play Store*. Data yang digunakan dalam penelitian ini adalah data ulasan sejumlah 1.500 ulasan dengan masing – masing aplikasi sebanyak 300 ulasan. Hasil dari penelitian didapatkan bahwa tingkat akurasi untuk aplikasi terbaik pertama adalah Ruang Guru sebesar 83,33%, kedua Zenius sebesar 82,67, ketiga Kelas Pintar sebesar 82 %, keempat Edmodo sebesar 74,67 % dan kelima Google Classroom sebesar 64,33%.

Pada penelitian yang dilakukan oleh Tineges, et al pada tahun 2020 yang berjudul “Analisis Sentimen Terhadap Layanan Indihome Berdasarkan Twitter dengan Metode Klasifikasi Support Vector Machine”. Tujuan dari penelitian ini adalah untuk dapat mengklasifikasi sentimen dengan *Support Vector Machine*, untuk mengetahui seberapa besar akurasi yang dihasilkan oleh metode SVM serta untuk mengetahui seberapa puas pengguna layanan indihome berdasarkan Twitter. Metode yang digunakan dalam penelitian ini adalah *Support Vector Machine*. Setelah dilakukan pengujian dengan metode SVM maka dihasilkan nilai *accuracy* adalah 87% , *precision* adalah 86%, *recall* adalah 95%, *error rate* adalah 13% dan *f1-score* adalah 90%.

Penelitian yang dilakukan oleh Romadoni (2020), dengan judul “Text Mining Untuk Analisis Sentimen Pelanggan Terhadap Layanan Uang Elektronik Menggunakan Algoritma Support Vector Machine”, dengan tujuan penelitian untuk mengklasifikasikan pendapat para pelanggan terhadap layanan uang elektronik OVO ke dalam kelas positif dan negatif. Penelitian ini menggunakan algoritma *Support Vector Machine* dengan *dataset* berjumlah 3.852 yang diambil melalui *Twitter*. Proses klasifikasi dilakukan empat skenario *splitting data* yaitu rasio 60:40, 70:30, 80:20 dan 90:10. Hasil akhir dari penelitian ini adalah nilai akurasi terbesar didapatkan oleh kernel linier dengan rasio data 90:10 yang mendapatkan nilai akurasi sebesar 98,7%.

Pada penelitian yang ditulis oleh Chory (2018) dengan judul “Sentiment Analysis on User Satisfaction Level of Mobile Data Services Using Support Vector Machine (SVM) Algorithm”, tujuan dari penelitian ini adalah untuk mengklasifikasi dokumen data yang diperoleh dari twitter dan Instagram kedalam tiga kelas yaitu positif, netral dan negatif serta mengetahui nilai akurasinya. Metode yang digunakan dalam penelitian ini adalah *Support Vector Machine* dengan TF-IDF pada pembobotan. Hasil dari penelitian ini adalah nilai presisi rata – rata data sebesar 95,43%, *recall* sebesar 92,45 , *f1-score* sebesar 93,90 % dan nilai akurasi sebesar 99,01%.

Pada penelitian yang berjudul “Social Media Sentiment Analysis Using Machine Learning and Optimization Techniques” yang ditulis oleh Badr, et al pada tahun 2019. Tujuan dari penelitian mengklasifikasikan pembicaraan atau sikap penulis terhadap berbagai peristiwa atau topik dan menyusun data kedalam kategori positif,

negatif atau netral. Metode yang digunakan dalam penelitian ini adalah *support vector machine* dan *naïve bayes*. Hasil dari penelitian ini menunjukkan nilai akurasi SVM PSO lebih baik dibandingkan dengan Naïve bayes, dimana nilai akurasi SVM sebesar 80,54 % dan nilai akurasi Naïve bayes adalah 73,30 %.

Berdasarkan pada penelitian – penelitian sebelumnya, maka dapat diambil kesimpulan terkait algoritma *support vector machine* sebagai berikut ini:

Tabel 2 1 Kajian Induktif

Penulis dan tahun	Judul	Metode	Hasil Penelitian
(Utama,et al.,2019)	Analisis Sentiment Sistem Ganjil Genap di Tol Bekasi Menggunakan Algoritma Support Vector Machine	Support Vector Machine	Berdasarkan hasil penelitian ini dapat menyimpulkan bahwa dengan menggunakan Algoritma <i>Support Vector Machine</i> dapat menganalisis sentiment ganjil genap di tol bekasi
(Herlinawati, 2020)	Analisis Sentimen <i>Zoom Cloud Meeting</i> di PlayStore Menggunakan naïve Bayes dan Support Vector Machine	Naïve Bayes dan Support Vector Machine	Hasil penelitian ini dapat diketahui nilai akurasi yang didapatkan algoritma <i>Support Vector Machine</i> (SVM) lebih unggul 6,85% dibandingkan algoritma <i>Naïve Bayes</i> .
(Anjasmos, 2020)	Analisis Sentimen Aplikasi Go- Jek Menggunakan Metode SVM dan NBC (Studi Kasus: Komentar Pada Play Store)	Naïve Bayes dan Support Vector Machine	Hasil penelitian ini <i>score accuracy</i> tertinggi sebesar 0,987 dengan menggunakan data <i>testing</i> 50 % dan 50 % data <i>training</i> menggunakan metode SVM kernel linear serta <i>score accuracy</i> terendah sebesar 0,486 dengan data <i>testing</i>

				sebesar 40 % dan data <i>training</i> 60 % dengan menggunakan metode SVM kernel RBF, polynominal dan sigmoid.
(Erfina, et al.,2020)	Analisis Sentimen Aplikasi Pembelajaran Online di <i>Play Store</i> pada Masa Pandemi	Support Vector Machine	SVM	maka dihasilkan nilai <i>accuracy</i> adalah 87% , <i>precision</i> adalah 86%, <i>recall</i> adalah 95%, <i>error rate</i> adalah 13% dan <i>f1-score</i> adalah 90%.
	Menggunakan Algoritma <i>Support Vector Machine</i>			
(Tineges,et al.,2020)	Analisis Sentimen Terhadap Layanan Indihome Berdasarkan Twitter dengan Metode Klasifikasi Support Vector Machine	Support Vector Machine	SVM	Pada penelitian ini dihasilkan bahwa nilai akurasi dengan menggunakan metode SVM sebesar 87%, <i>precision</i> adalah 86%, <i>recall</i> adalah 95%, <i>error rate</i> adalah 13% dan <i>f1-score</i> adalah 90%.
(Romadoni, 2020)	Text Mining Untuk Analisis Sentimen Pelanggan Terhadap Layanan Uang Elektronik Menggunakan Algoritma Support Vector Machine	Support Vector Machine	SVM	Hasil dari penelitian ini dinyatakan bahwa nilai akurasi terbesar didapatkan oleh kernel linier dengan rasio data 90:10 yang mendapatkan nilai akurasi sebesar 98,7%.
(Chory, 2018)	Sentiment Analysis on User Satisfaction Level of Mobile Data Services Using Support Vector Machine (SVM)	Support Vector Machine dengan TF-IDF	SVM	Hasil penelitian ini adalah nilai presisi rata – rata data sebesar 95,43%, <i>recall</i> sebesar 92,45%, <i>f1-score</i> sebesar 93,90 % dan nilai akurasi sebesar 99,01%.

	Algorithm			
(Badr,E.M., et al,2019)	Social Sentiment using Learning Optimization Technique.	Media Analysis Machine and Bayes.	Support Vector Machine dan Naïve Bayes.	Penelitian ini menunjukkan bahwa nilai akurasi SVM PSO memiliki nilai lebih akurat dibandingkan dengan metode Naïve Bayes. Dimana nilai akurasi SVM sebesar 80,54% dan Naïve Bayes sebesar 73,30 %.

Berdasarkan tinjauan pustaka diatas, mengenai beberapa penelitian yang telah dilakukan sebelumnya yang memiliki kemiripan teknik dan tujuan dengan penelitian yang dilakukan. Hasil penelitian terdahulu menunjukkan bahwa klasifikasi menggunakan metode *Support Vector Machine* memiliki nilai akurasi yang cukup memiliki akurasi yang baik. Pada penelitian ini dilakukan implementasi metode *Support Vector Machine* dalam analisis sentimen di *platform streaming online*. Sehingga dengan dilakukannya penelitian ini diharapkan dapat menggambarkan serta memberikan evaluasi terhadap pelayanan dan kinerja dari Aplikasi Iflix.

2.2 Kajian Deduktif

2.2.1 Iflix

Iflix merupakan salah satu platform berbasis aplikasi dan website yang menyediakan layanan *video on demand* (VOD) gratis dan berlangganan yang berfokus pada pemasaran. Iflix saat ini tersedia untuk berbagai negara di Asia yaitu Malaysia, Indonesia, Filipina, Thailand, Sri Lanka, Brunei, Pakistan, Myanmar, Vietnam, Kamboja, Nepal, Bangladesh dan Maladewa. Pada bulan Maret 2019, pengguna Iflix memiliki lebih dari 25 juta pelanggan di negara cakupannya. Iflix berdiri sejak tahun 2014 lalu dan masuk ke Indonesia pada akhir tahun 2015. Jenis konten yang tersedia pada aplikasi ini adalah Serial TV, Film Box Office hingga acara khusus anak – anak. Iflix merupakan perusahaan layanan jasa berbasis teknologi sehingga

diperlukan kualitas pelayanan yang baik karena hal tersebut menjadi tolak ukur suatu kepuasan dari pelanggan atau pengguna (Linggi, 2020).

2.2.2 Google Play

Google Play merupakan toko aplikasi resmi milik *Google* untuk perangkat yang menggunakan sistem operasi Android. *Google Play* pertama kali dirilis pada tanggal 22 Oktober 2008, dengan nama terdahulu Android Market sebelum diganti menjadi *Google Play Store* pada bulan Maret 2012. Layanan dari *Google Play* sendiri dapat diakses melalui website, aplikasi android dan Google TV.

2.2.3 Online Review

Online review adalah salah satu bentuk dari *electronic word of mouth* (eWOM). *Online review* dapat dipahami sebagai salah satu media untuk konsumen melihat *review* dari konsumen lain terhadap suatu produk, layanan perusahaan dan tentang bagaimana sebuah perusahaan produsen. Menurut (Abdulrahman, 2013), *Online Review* adalah sebuah sarana untuk konsumen mencari dan mendapatkan informasi yang nantinya akan mempengaruhi keputusan pembelian atau penggunaan sesuatu produk. *Online review* juga berfungsi sebagai alat bentuk pengambilan keputusan untuk *feedback* yang diberikan konsumen dan sistem rekomendasi pada *platform* online. Menurut (Atik, 2016), *Online review* adalah salah satu bentuk *electronic word of mouth* dan dapat dikatakan sebagai komunikasi pemasaran yang baru dan dapat mempengaruhi serta memegang peran dalam proses keputusan pembelian atau pandangan terhadap produk. Informasi yang didapatkan digunakan untuk mengidentifikasi dan mengevaluasi produk yang konsumen butuhkan.

2.2.4 Web Scraping

Menurut (Utomo M. S., 2012), Salah satu cara untuk memisahkan konten utama halaman situs dengan bagian – bagian yang tidak berhubungan dengan isi adalah dengan menggunakan teknik scraping. Web Scraping adalah sebuah perangkat lunak komputer teknik penggalian informasi dari situs web. Biasanya, program perangkat lunak tersebut mensimulasikan eksplorasi manusia dari web dengan menerapkan *Hypertext Transfer Protocol* (HTTP) atau *embedding web browser* seperti Internet Explorer (IE) dan Mozilla Web Browser (Utomo M. S., 2013).

Web Scraping adalah proses pengambilan sebuah dokumen semi- terstruktur dari internet, umumnya berupa halaman – halaman *web* dalam bahasa *markub* seperti HTML atau XHTML, dan menganalisis dokumen tersebut untuk digunakan bagi kepentingan lain (Turland, 2010). Manfaat dari *web scraping* yaitu agar informasi yang diambil lebih terfokus sehingga memudahkan dalam melakukan pencarian sesuatu. *Web Scraping* sering dikenal sebagai *screen scraping*. *Web scraping* memiliki beberapa langkah, sebagai berikut (Josi & Abdillah, 2014):

1. *Create Scraping Template*

Pembuat program mempelajari dokumen HTML dari *website* yang akan diambil informasinya untuk tag HTML yang mengapit informasi yang akan diambil.

2. *Explore Site Navigation*

Pembuat program mempelajari teknik navigasi pada *website* yang akan diambil informasinya untuk ditirukan pada aplikasi *web scraper* yang akan dibuat.

3. *Automate Navigation and Extraction*

Berdasarkan informasi yang didapatkan dari langkah pertama dan kedua diatas, aplikasi *web scraper* dibuat untuk mengotomatisasi pengambilan informasi dari *website* yang ditentukan.

4. *Extracted Data and Package History*

Informasi yang didapatkan dari langkah sebelumnya disimpan dalam tabel atau tabel – tabel *database*.

2.2.5 *Machine Learning*

Machine learning adalah aplikasi atau bagian dari kecerdasan buatan yang membuat sistem memiliki kemampuan belajar secara otomatis dan meningkatkan kemampuannya berdasarkan pengalaman tanpa diprogram secara eksplisit. *Machine learning* memiliki fokus pada pengembangan sistem yang mampu belajar sendiri untuk memutuskan sesuatu tanpa harus berulang kali diprogram oleh manusia. *Machine learning* bekerja apabila tersedia data sebagai input untuk dilakukan analisis terhadap kumpulan data besar sehingga menemukan pola tertentu. Data merupakan bahan input yang akan digunakan dalam melakukan pembelajaran atau *training* sehingga mesin dapat menghasilkan analisis yang benar. Pada *machine learning* terdapat dua data yaitu data *testing* dan data *training*, dimana data *training* berfungsi untuk melatih algoritma dalam *machine learning* dan data *testing* berfungsi untuk mengetahui performa dari algoritma dalam *machine learning* yang telah dilatih yaitu ketika menemukan data baru yang belum pernah diberikan dalam data *training* (Fikriya, Soetrisno, & Irawan, 2017).

Machine learning merupakan bagian dari kecerdasan buatan dimana *machine learning* bertujuan untuk memahami atau mengenali struktur suatu data dan mengonversi data tersebut kedalam suatu model. Hal ini berbeda dengan pemrograman tradisional, di dalam *machine learning code* program ditulis secara eksplisit. Kemampuan belajar tersebut diperoleh dengan mengakses data dan informasi secara terus-menerus. Implementasi *machine learning* dalam menyelesaikan suatu permasalahan membutuhkan beberapa persyaratan dimana jika persyaratan tersebut tidak terpenuhi maka akan menurunkan akurasi hasil atau solusi. Pertama, proses *machine learning* membutuhkan ruang penyimpanan atau memori yang besar mengingat data yang diolah juga pada umumnya besar. Kedua, dibutuhkan proses pelabelan, khususnya untuk *supervised learning*. Proses pelabelan seringkali membutuhkan waktu yang lama. ketiga, tidak jarang hasil atau solusi yang muncul tidak akurat karena datanya bias (Kusuma, 2020)

Machine learning menggunakan teknik untuk menangani data besar (*big data*) dengan cara yang cerdas untuk memberikan hasil yang tepat. Berdasarkan teknik pembelajarannya, tipe - tipe *machine learning* dapat dibedakan menjadi *supervised learning*, *unsupervised learning*, semi *supervised learning* dan *reinforcement*

learning. *Supervised learning* merupakan salah satu teknik teknik *machine learning* yang menggunakan *dataset* (*data training*) yang sudah berlabel (*labeled*) untuk melakukan pembelajaran pada mesin, sehingga mesin mampu mengidentifikasi label input dengan menggunakan fitur yang dimiliki untuk selanjutnya melakukan prediksi maupun klasifikasi, sedangkan *unsupervised learning* adalah teknik dengan menarik kesimpulan berdasarkan *dataset* yang merupakan input dari data *labeled response* (Iswara, 2019). Algoritma yang termasuk kedalam teknik *supervised learning* diantaranya *Decision Tree*, *K-Nearest Neighbor* (KNN), *Naïve Bayes*, *Regresi*, dan *Super Vector Machine*.

2.2.6 Analisis Sentimen

Analisis sentimen biasanya dilakukan dalam penelitian berbentuk analisis, salah satunya mengenai opini dan emosi banyak orang terhadap suatu entitas sebagai contoh permasalahan, topik atau layanan. Analisis sentimen merupakan proses ekstraksi emosi atau opini dari sebuah teks atau bacaan (Mangat, 2017). *Sentiment analysis* atau *opinion mining* mengacu pada bidang yang luas dari pengolahan bahasa alami, komputasi linguistic dan *text mining* yang bertujuan menganalisis pendapat, sentiment, evaluasi, sikap, penilaian, dan emosi seseorang berkenaan dengan suatu topik, produk, layanan, organisasi, individu, ataupun kegiatan tertentu lainnya (Liu, 2010). Tugas dasar dari analisis sentimen adalah mengelompokkan teks yang ada dalam sebuah kalimat atau dokumen kemudian menentukan pendapat yang dikemukakan dalam kalimat atau dokumen tersebut apakah termasuk dalam kelas positif atau negatif.

2.2.7 Text Mining

Text mining adalah suatu langkah dari analisis teks yang secara otomatis dilakukan oleh komputer dengan tujuan menggali informasi yang berkualitas dari suatu rangkaian teks yang terkandung dalam sebuah dokumen (Kamber, 2006). *Text Mining* merupakan salah satu proses yang bidang dipakai untuk mengerjakan

klasifikasi dokumen yang berusaha untuk menemukan bentuk yang menarik dari sekumpulan data yang terkumpul. *Text mining* merupakan bagian dari data mining dimana proses yang dilakukan utamanya adalah mengerjakan ekstraksi pengetahuan dan informasi dari pola – pola yang terdapat dalam sekumpulan dokumen teks (Samuel & Antonius, 2015). *Text mining* dalam melakukan analisis, data yang digunakan adalah data yang diperoleh dari dokumen yang bertujuan untuk mencari kata yang dapat menggantikan dari dokumen kata – kata yang lain. *Text mining* memiliki define sebagai sebuah proses penemuan informasi, relasi dan fakta yang tersembunyi di dalam teks ketika dilakukan pemrosesan dan analisis data dalam jumlah besar, struktur teks yang kompleks dan tidak lengkap, dimensi tinggi, serta data yang *noise*. *Text mining* memiliki berbagai kegunaan yang dibutuhkan dalam kehidupan sehari-hari (Adiwijaya, 2006).

Text mining merupakan penggalian data untuk menyelesaikan masalah kebutuhan informasi dengan menerapkan teknik *data mining, machine learning, natural language processing*, pencarian informasi dan manajemen pengetahuan. *Text mining* melibatkan pra proses atau *preprocess* dokumen yang terdiri dari kategorisasi teks, ekstraksi informasi, dan ekstraksi kata. Metode ini digunakan dengan tujuan dapat mengekstraksi informasi dari sumber data melalui identifikasi dan eksplorasi pola yang menarik (Feldman R. &, 2007).

2.2.8 Pre Processing

Pre Processing merupakan tahapan awal dalam melakukan pengolahan teks yang digunakan untuk mengubah bentuk dokumen menjadi data terstruktur sesuai kebutuhannya agar dapat diolah lebih lanjut dalam proses *text mining*. Tahapan *pre process* dalam klasifikasi bertujuan untuk meningkatkan akurasi klasifikasi data. Tahapan dalam *pre process* adalah sebagai berikut ini:

1. *Case Folding*, merupakan proses untuk mengubah semua karakter teks menjadi huruf kecil serta menghilangkan tanda baca dan angka. Cara kerja *case folding* adalah memproses huruf alphabet dari “a” hingga “z” saja sehingga karakter selain huruf tersebut dihapus (Weiss, 2010)

2. *Tokenizing*, merupakan proses memecah yang semula kalimat menjadi kata – kata atau memutus urutan string menjadi potongan – potongan, seperti kata – kata berdasarkan tiap kata yang menyusunnya.
3. *Stopwords*, merupakan kosa kata yang bukan termasuk kata unik atau ciri pada suatu dokumen atau tidak menyampaikan pesan apapun secara signifikan pada teks atau kalimat (Dragut, 2009). Kosakata yang dimaksud yaitu seperti kata penghubung dan kata keterangan yang bukan merupakan kata unik, missal “dari”, “akan”, “seorang”.

2.2.9 Klasifikasi

Menurut (Zhu, 2015), Klasifikasi merupakan model yang memprediksi suatu nilai yang tidak diketahui. Model klasifikasi terbagi menjadi dua yaitu, melatih data kemudian membedakan kedalam *class* dan melakukan evaluasi terhadap objek tidak dikenal (Fahrizal, 2020). Klasifikasi memerlukan persiapan data yang baik dan pemilihan bahasa pemrograman yang mampu mendukung *machine learning*. Salah satu tugas yang dapat dilakukan dengan *data mining* adalah pengklasifikasian. Dalam klasifikasi terdapat target variabel kategori. Metode – metode yang telah dikembangkan oleh periset untuk menyelesaikan kasus klasifikasi antara lain sebagai berikut ini (Mardi, 2017):

1. Pohon Keputusan
2. *Naïve Bayes*
3. Jaringan Saraf Tiruan
4. Analisis Statistik
5. Algoritma Genetik
6. *Rough Sets*
7. Pengklasifikasi *K-nearest Neighbour*
8. Metode Berbasis Aturan
9. *Memory Based Reasoning*
10. *Support Vector Machine*

2.2.10 Fitur dan Pembobotan

Pembobotan merupakan metode untuk mengubah input data menjadi suatu fitur vector. Setiap dokumen mengandung berbagai macam kata yang berbeda. Hal yang perlu dipahami dalam melakukan pencarian informasi dari dokumen yang heterogen adalah pembobotan kata, karena setiap kata memiliki tingkat kepentingan yang berbeda dalam dokumen. Oleh karena itu dilakukan bobot *term* (Manning, 2009). beberapa metode pembobotan yang biasa digunakan adalah sebagai berikut ini:

1. *Term Frequency* (TF)

Metode ini menghitung beberapa kali jumlah kemunculan fitur di dalam dokumen, misal sebuah kata muncul dua kali maka bobotnya sebesar 2.

2. *Feature Term Presence* (TP)

Metode ini menunjukkan ada atau tidaknya fitur di dalam dokumen. Jika ada maka nilainya 1 dan jika tidak maka nilainya 0. Tidak diperhatikan berapa kali kemunculannya.

3. *Term Frequency-Inverse Document Frequency* (TF-IDF)

Algoritma TF-IDF adalah algoritma yang berdasarkan nilai statistik kemunculan kata – kata dalam dokumen. TF (*Term Frequency*) menyatakan banyaknya suatu kata muncul dalam dokumen maupun kalimat. DF (*Document Frequency*) adalah banyaknya dokumen yang mengandung satu kata dalam segmen publikasi (Feldman R. &, 2007).

2.2.11 *Support Vector Machine* (SVM)

Support Vector Machine merupakan salah satu metode klasifikasi yang dapat digunakan dalam data mining. Metode *Support Vector Machine* (SVM) memungkinkan perhitungan untuk masalah linier dengan menerapkan transformasi matematis untuk ruang belajar menggunakan fungsi kernel (Alamarani, 2018). Metode *Support Vector Machine* (SVM) memiliki konsep sentral dalam mengklasifikasikan data, yaitu menentukan *hyperplane* terbaik untuk memberi jarak atau pemisah antara dua kelas yang telah ditentukan (Kowalczyk, 2017). Cara kerja *Support Vector Machine* (SVM) yaitu dengan cara menemukan *hyperplane* optimal

yang memberi jarak atau pemisah antara dua kelas. *Hyperplane* memiliki margin maksimum. Jarak antara titik data terdekat dengan *hyperplane* diklaim sebagai margin (d). Tahapan dalam metode *Support Vector Machine* adalah sebagai berikut ini (Tineges, 2020):

1. Menentukan kata yang sering muncul dari tiap dokumen yang digunakan
2. Menentukan inisialisasi awal untuk nilai $\alpha=0.5, C=1, \lambda=0.5, \text{gamma}=0.5$ dan $\text{epsilon}=0.001$
3. Menghitung matrik dengan rumus seperti dibawah ini:

$$D_{ij} = y_i y_j ((x_i^{\rightarrow} \cdot x_j^{\rightarrow}) + \lambda^2) \quad (2.1)$$

Keterangan:

D_{ij} = elemen matriks data ke $-ij$

y_i = kelas atau label data ke $-i$

y_j = kelas atau label data ke $-j$

λ = turunan batas teoritis

$K(x_i^{\rightarrow} \cdot x_j^{\rightarrow})$ = fungsi kernel

4. Untuk daya ke $n= 1,2,3,\dots n$ gunakan persamaan seperti dibawah ini :

$$E_i = \sum_{j=1}^n \alpha_j D_{ij} \quad (2.2)$$

$$\delta \alpha_i = \min\{\max[\gamma(1 - E_i), -\alpha_i], C - \alpha_i\} \quad (2.3)$$

$$\alpha_i = \alpha_i + \delta \alpha_i \quad (2.4)$$

Keterangan:

E_i = nilai error data ke $-i$

γ = tingkat pembelajaran

$\text{Max}_{(i)} D_{ij}$ = nilai maksimum diagonal matriks persamaan

5. Mencari nilai bias (b) dengan menggunakan persamaan seperti dibawah ini:

$$b = - \frac{1}{2} [w \cdot x_+ + w \cdot x_-] \quad (2.5)$$

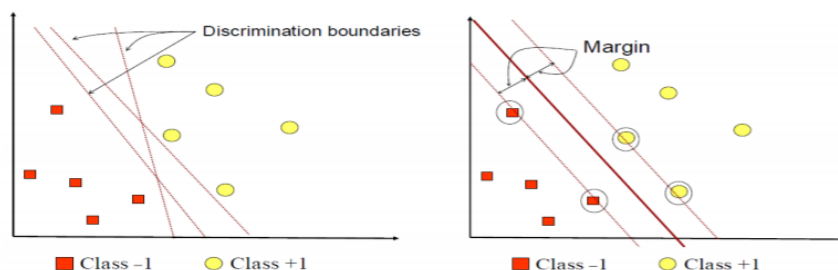
6. Pengujian pada dokumen yang diuji
7. Perhitungan keputusan

Jika perhitungan keputusan lebih dari sama dengan 0 maka nilai sign $h(x)$ adalah +1, maka termasuk kelas positif dan jika hasil perhitungan keputusan kurang dari 0 maka nilai sign $h(x)$ nilai -1, maka termasuk kelas negatif. Perhitungan keputusan dengan menggunakan persamaan seperti dibawah ini:

$$h(x) = w \cdot x + b \quad (7) \text{ atau } h(x) = \sum_{i=1}^m \alpha_i y_i K(x, x_i) + b \quad (2.6)$$

Support Vector Machine termasuk dalam kelas *supervised learning*. *Support Vector Machine* merupakan suatu metode yang dapat melakukan prediksi baik pada klasifikasi ataupun regresi. Pada teknik ini, dilakukan untuk menemukan fungsi pemisah yang optimal yang dapat memisahkan dua kelas yang berbeda. SVM dapat digunakan pada jenis set data yang berbeda yaitu set data dengan pemisahan secara linier dan non linier. SVM linier dapat dilakukan apabila data yang akan digunakan dapat dipisahkan secara linier. Namun apabila tidak dapat dilakukan secara linier maka perlu dilakukan pemisahan secara non linier (Sembiring, 2007). *Hyperplane* terbaik adalah *hyperplane* yang terletak ditengah – tengah antara dua set objek dari dua kelas. Pada Gambar 2.1 menunjukkan bagaimana SVM mencari *Hyperplane* terbaik ekuivalen dengan memaksimalkan margin atau jarak dua set dari kelas yang berbeda (Pratiwi, 2020). Memaksimalkan margin artinya suatu *hyperplane* yang memiliki jarak terjauh terhadap data pada suatu kelas. Hal tersebut membuat adanya jarak yang besar antara *hyperplane* dengan data. *Hyperplane* yang memiliki margin besar dapat melakukan klasifikasi dengan baik pada data yang belum dilihat, dengan memberikan margin yang besar maka suatu data baru dapat masuk ke kelas masing – masing (Putra, 2020). Dimana margin sendiri dapat digambarkan seperti pada Gambar 2.1 dibawah ini. Dengan dilakukan memaksimumkan margin, maka akan dihasilkan suatu *Hyperplane* tertentu. Margin maksimum adalah pilihan yang akan aman karena jika terjadi sedikit kesalahan pada data maka akan memberikan kemungkinan terkecil terjadi kesalahan klasifikasi. Selain itu maksimum margin akan memberikan kapabilitas generalisasi terbaik. Untuk mendapatkan *hyperplane* terbaik adalah dengan mencari *hyperplane* yang terletak

di tengah – tengah antara dua bidang pembatas kelas margin atau jarak antara dua set objek dari kelas yang berbeda (Santosa, 2007).



Gambar 2 1 Hyperplane Terbaik

Sumber: (Drajana, 2017)

SVM merupakan suatu algoritma yang bekerja menggunakan pemetaan nonlinear untuk mengubah data pelatihan asli ke dimensi yang lebih tinggi. Dimensi baru akan mencari *hyperplane* untuk memisahkan secara linier dengan pemetaan nonlinear yang tepat ke dimensi yang lebih tinggi, data dari dua kelas selalu dapat dipisahkan dengan *hyperplane* tersebut (Widodo, 2013). Dimensi yang lebih tinggi pada data mining merupakan suatu data yang memiliki puluhan sampai ribuan atribut (Margiastuti, 2010). *Feature space* atau ruang fitur mengacu pada n-dimensi tempat variabel anda berada. Ruang fitur mengacu pada kumpulan fitur yang digunakan untuk mengkarakteristik data dalam penelitian. Fitur yang dimaksud merupakan semua variabel yang ada. Dalam SVM, dilakukan pertimbangan mengenai karakteristik yang berbeda untuk menggambarkan data (Cam.Davidson, 2012).

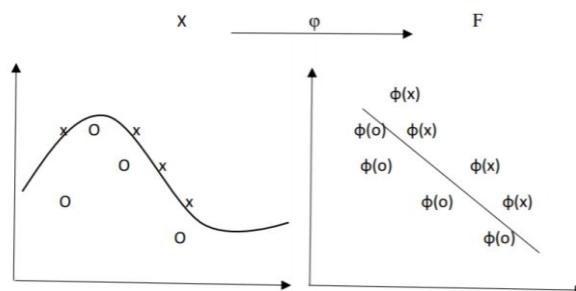
Dalam mengatasi *Support Vector Machine* pada kasus non linier yang terjadi maka digunakan metode kernel untuk mengatasinya. Suatu fungsi dari kernel adalah melakukan transformasi suatu data ke suatu ruang atau dimensi lainnya yang dapat diilustrasikan pada Gambar 2.2 dibawah ini (Putra, 2020).



Gambar 2 2 Transformasi Data

Sumber: (Putra, 2020)

Dengan menggunakan metode kernel suatu data x di input ke *feature space* F dengan dimensi yang lebih tinggi. *feature mapping* dari ruang dua dimensi ke *feature space* dua dimensi (Ritonga, 2018).



Gambar 2 3 SVM Non Linier Kernel Map
Sumber: (Ritonga,2018)

Terdapat beberapa fungsi kernel yang biasanya digunakan dalam SVM sebagai berikut ini:

1. Linier

Kernel linier biasa digunakan untuk melakukan perkalian titik normal pada dua pengamatan yang diberikan. Hasil dari produk di antara kedua vektor itu adalah penjumlahan dari perkalian setiap pasang nilai masukan.

$$K(x, x_i) = \sum(x * x_i) \quad (2.7)$$

2. Polynomial

Kernel polynomial adalah bentuk kernel linier yang lebih umum. Kernel polynomial dapat membedakan ruang masukan melengkung atau nonlinear.

$$K(x, x_i) = 1 + \sum(x * x_i)^d \quad (2.8)$$

3. Radial Basis Function (RBF)

Kernel RBF merupakan fungsi kernel yang populer yang biasa digunakan dalam *support vector machine*. RBF dapat memetakan ruang masukan dalam ruang dimensi tak terhingga (Nablani, 2019).

$$K(x, x_i) = \exp(-\gamma * \sum((x - x_i)^2)) \quad (2.9)$$

4. Sigmoid

Kernel Sigmoid cukup populer untuk *support vector machine*. Klasifikasi dengan sigmoid telah diusulkan secara luas dalam praktiknya karena berasal dari jaringan neural (Sitiorus, 2020).

$$K(x,x_j) = \tanh(\alpha x_i \cdot x_j + \beta) \quad (2.10)$$

2.2.12 Asosiasi Kata

Asosiasi kata merupakan suatu keeratan hubungan antara dua atau lebih variabel yang bersifat kualitatif (Ulwan, 2016). Pendekatan asosiasi kata akan digunakan untuk mencari hubungan antara teks *review* pada aplikasi Iflix sehingga akan menghasilkan suatu informasi yang bermanfaat.

2.2.13 Diagram Fishbone

Menurut (Besterfield, 2009), Diagram sebab akibat adalah suatu diagram yang menggambarkan garis dan simbol – simbol yang menunjukkan hubungan antara penyebab dan akibat suatu masalah serta selanjutnya diambil tindakan perbaikan atas masalah tersebut. Diagram sebab akibat dikenal sebagai *fishbone* diagram.

Fishbone diagram memiliki bentuk menyerupai tulang ikan. Dimana setiap tulang mewakili kemungkinan kesalahan yang terjadi. Diagram ini berguna untuk memperlihatkan faktor – faktor utama yang berpengaruh pada kualitas dan mempunyai pada masalah yang kita pelajari (Haizer, 2014). *Fishbone* Diagram dapat diartikan sebagai diagram yang mampu untuk mengidentifikasi berbagai kemungkinan penyebab dari suatu masalah. *Fishbone* diagram digunakan untuk menggambarkan identifikasi akar masalah dari suatu *outcome*.

2.2.14 Python

Python dikenal sebagai bahasa pemrograman interpreter, karena *Python* dieksekusi dengan sebuah interpreter (Kurniawan, 2011). *Python* saat ini telah menduduki posisi

4 atau 5 bahasa pemrograman yang paling sering digunakan di seluruh dunia. Bahasa pemrograman *Python* memiliki beberapa fitur yang dapat digunakan oleh pengembang perangkat lunak. Berikut adalah beberapa fitur yang ada pada bahasa pemrograman *Python* (Lutz, 2010):

1. *Multi Paradigm Design*
2. *Open Source*
3. *Simplicity*
4. *Library Support*
5. *Portability*
6. *Extendable*
7. *Scalability*

Bahasa pemrograman *Python* memiliki beberapa keunggulan seperti *readability*, efisien, multifungsi, interoperabilitas dan memiliki dukungan komunitas yang memadai (Wahyono, 2018). *Python* dikatakan *reliability* karena *Piton* memiliki *source code* yang sederhana sehingga mudah ditulis, mudah diingat dan mudah digunakan secara berulang. Hal tersebut memudahkan pengembang aplikasi dalam melakukan coding, *testing*, perbaikan apabila terjadi kesalahan, *bug* atau *error* lainnya. *Python* dikatakan efisien, dikarenakan *Python* memiliki *library* yang lengkap dan *Python code* akan lebih sederhana apabila dibandingkan kode yang ditulis pada pemrograman lainnya. *Python* memiliki banyak modul yang siap digunakan untuk pengembangan aplikasi sesuai dengan kebutuhan. *Python* memiliki interoperabilitas yang tinggi karena mampu berinteraksi dengan bahasa – bahasa pemrograman lain. Pemrograman yang dikembangkan oleh *Python* dapat di operasionalisasi pada hampir semua sistem operasi baik *Windows*, *Linux*, *Mac Os*, *Unix* dan juga sistem operasi pada perangkat lunak berbasis *mobile* seperti *Android* atau *IOS*. *Python* memiliki dukungan komunitas yang cukup kuat, dikarenakan *Python* bersifat *open source*. Dengan komunitas yang baik, mempermudah pengguna untuk saling berbagi dan mengembangkan bahasa pemrograman *Python* menjadi bahasa yang handal (Pramudita, 2020).

BAB III

METODE PENELITIAN

3.1 Objek dan Subjek Penelitian

Objek pada penelitian ini adalah suatu *platform video online streaming* yaitu Aplikasi Iflix. Subjek pada penelitian ini adalah persepsi atau pandangan dari pengguna aplikasi Aplikasi Iflix terhadap pelayanan yang telah diberikan melalui ulasan di *website Google Play*.

3.2 Populasi dan Sampel Penelitian

Populasi yang ada dalam penelitian ini merupakan semua data ulasan dari pengguna Aplikasi Iflix yang berasal dari *website Google Play* dengan total komentar sebanyak 611.711 ulasan. Sedangkan sampel data yang digunakan dalam penelitian ini adalah data ulasan mulai dari tanggal 1 Januari 2021 hingga 17 Maret 2021. Dimana jumlah sampel yang digunakan sebanyak 4.501 ulasan.

3.3 Pengumpulan Data

3.3.1 Metode Pengumpulan Data

Metode yang digunakan dalam penelitian adalah sebagai berikut ini:

1. *Web Scraping*

Pengambilan data dalam penelitian ini dilakukan secara online dengan menggunakan teknik *scraping data* menggunakan *Data Scraper* yang ada di *Google Chrome*. Teknik *scraper* ini dapat menghasilkan suatu informasi dari website yang telah ditetapkan secara otomatis.

2. Studi Pustaka

Pada penelitian ini digunakan beberapa kajian literatur, jurnal dan buku yang berkaitan dengan metode penelitian, tujuan serta rumusan masalah yang berkaitan dengan penelitian yang dilakukan.

3.3.2 Jenis dan Sumber Data

Jenis data yang digunakan dalam penelitian ini adalah data sekunder. Data sekunder digunakan sebagai data yang akan diolah dalam proses pengolahan data untuk mengetahui tanggapan atau persepsi pengguna terhadap pelayanan yang diberikan oleh Aplikasi Iflix yang dituangkan dalam *website Google Play*. Sumber data dalam *Google Play* terhadap pelayanan yang diberikan oleh Aplikasi Iflix. Pada penelitian ini dalam melakukan pengambilan data digunakan teknik *scraping data* yang bernama *Data Scraper* dari halaman situs *Google Chrome*. Data yang diperoleh berupa ulasan dari pengguna Aplikasi Iflix yang diambil sejak tanggal 1 Januari 2021 hingga 17 Maret 2021. Dimana jumlah sampel yang digunakan sebanyak 4.501 ulasan.

3.4 Metode Analisis Data

Pada penelitian ini terdapat beberapa metode analisis data yang digunakan. Berikut merupakan metode analisis data yang digunakan:

1. Metode analisis deskriptif

Metode ini digunakan dengan tujuan untuk memberikan gambaran secara umum mengenai komentar yang diberikan oleh pengguna Aplikasi Iflix pada aplikasi website *Google Play*.

2. Metode *Machine Learning* yaitu *Support Vector Machine*

Metode *Support Vector Machine* digunakan sebagai suatu metode yang menunjang ketika melakukan proses klasifikasi komentar, baik komentar negatif maupun positif.

3. *Wordcloud*

Metode ini digunakan untuk melakukan proses visualisasi dari kata – kata yang sering muncul pada komentar yang diberikan pengguna Aplikasi Iflix.

4. *Association*

Metode ini digunakan untuk melakukan identifikasi dan membentuk pola kata yang berasosiasi dengan kata lainnya untuk mendapatkan informasi yang dianggap penting.

5. Diagram *Fishbone*

Diagram *fishbone* digunakan dalam melakukan identifikasi ulasan berdasarkan kelas negatif.

3.5 Variabel Penelitian

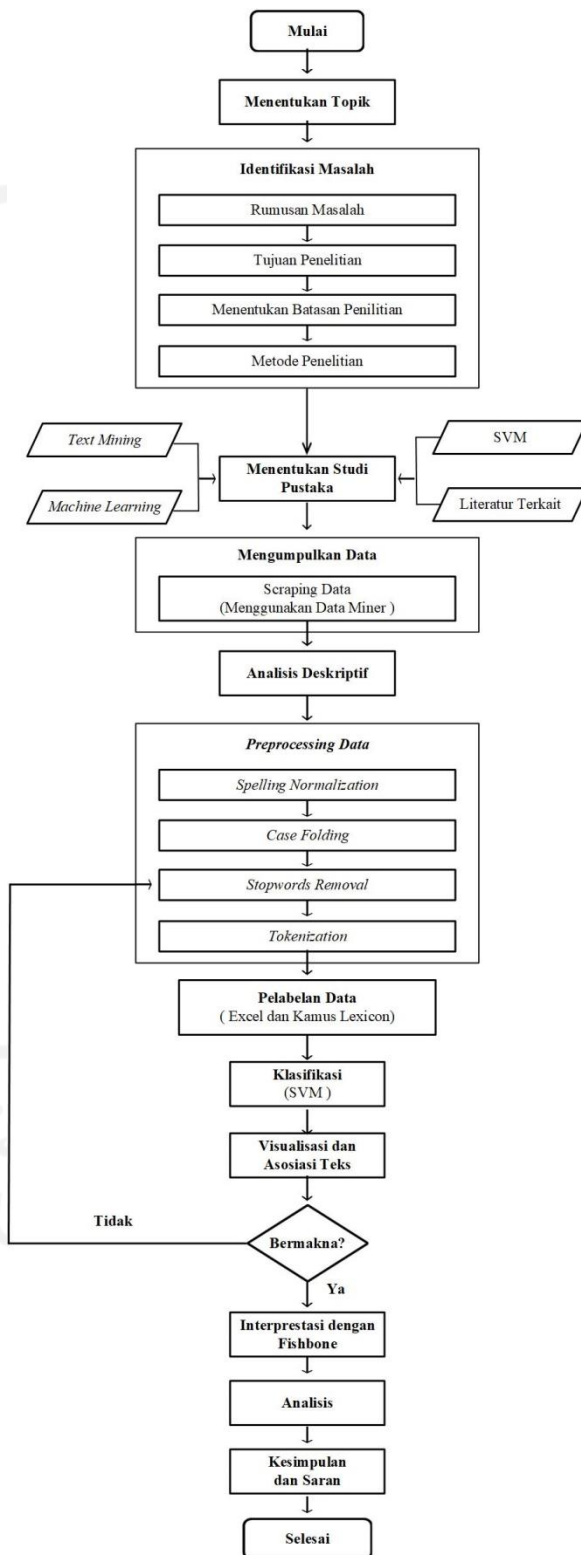
Variabel penelitian adalah atribut – atribut yang digunakan dalam menentukan karakteristik atau batasan dalam mendapatkan informasi yang berkaitan dengan penelitian yang dilakukan. Beberapa variabel yang digunakan dalam penelitian ini adalah sebagai berikut:

Tabel 3 1 Variabel Penelitian

Variabel	Definisi
Komentar	Berisi komentar yang diberikan oleh pengguna terkait pelayanan yang diberikan oleh Aplikasi Iflix di <i>website Google Play</i> .
Waktu	Waktu menunjukkan tanggal, bulan dan tahun dibuatnya suatu komentar oleh pengguna pada <i>website Google Play</i> .
Rating	Rating atau penilaian yang diberikan oleh pengguna aplikasi Iflix.

3.6 Alur Penelitian

Berikut merupakan alur dari penelitian ini:



Gambar 3 1 Alur Penelitian

Berdasarkan alur penelitian diatas menunjukkan bagaimana proses penelitian yang terjadi. Berikut merupakan penjelasan mendetail mengenai langkah – langkah pada penelitian:

1. Mulai

2. Menentukan Topik

Pada penelitian ini, langkah pertama yang dilakukan adalah menentukan topik yang akan dilakukan. Topik dari penelitian ini adalah analisis sentimen pada Aplikasi Iflix.

3. Identifikasi Masalah

Pada tahap ini dilakukan identifikasi masalah secara *online* melalui *website Google Play*. Aplikasi Iflix merupakan salah satu *platform video online streaming* yang sering digunakan khususnya masyarakat Indonesia, sehingga perlu dilakukan analisis sentimen untuk mengetahui bagaimana persepsi pengguna terhadap Aplikasi Iflix.

4. Rumusan Masalah dan Tujuan Penelitian

Tahap ini dilakukan perumusan masalah terkait bagaimana gambaran umum terhadap tanggapan dari pengguna Aplikasi Iflix, bagaimana hasil dari implementasi metode *Support Vector Machine* dalam melakukan klasifikasi data ulasan pengguna Aplikasi Iflix menjadi kelas positif dan negatif, bagaimana informasi yang dihasilkan dalam setiap klasifikasi dan asosiasi teks yang telah dilakukan, bagaimana rencana pemecahan terhadap Aplikasi Iflix. Pada tahap ini dilakukan penentuan tujuan penelitian, dimana ditentukan sesuai dengan apa yang ingin dicapai berdasarkan rumusan masalah yang telah ditentukan sebelumnya. Tujuan dari penelitian ini adalah mengetahui gambaran umum mengenai tanggapan dari pengguna Aplikasi Iflix, menentukan hasil dari penerapan pada metode *Support Vector Machine* dalam mengklasifikasikan data ulasan dari pengguna Aplikasi Iflix menjadi kelas positif dan negatif, mengetahui dan memperoleh informasi penting dari hasil klasifikasi dan asosiasi teks yang dilakukan, dan mengetahui informasi faktor – faktor yang berkaitan dalam melakukan perbaikan dari hasil ulasan negatif yang didapatkan.

5. Menentukan Batasan dan Metode Penelitian

Pada tahap ini ditentukan beberapa batasan dalam penelitian seperti data yang digunakan berupa data ulasan sejak 1 Januari 2021 sampai dengan 17 Maret 2021 yang diambil pada *website Google Play* serta data ulasan menggunakan bahasa Indonesia. Metode penelitian yang digunakan dalam penelitian ini adalah *Support Vector Machine*.

6. Menentukan Studi Pustaka

Untuk menunjang penelitian yang dilakukan, dilakukan studi pustaka yang berkaitan dengan topik yang telah ditentukan. Berikut beberapa studi pustaka yang digunakan sebagai referensi dalam penelitian ini:

- a. *Machine Learning*
- b. *Text Mining*
- c. *Support Vector Machine*
- d. Literatur Terkait mengenai pengolahan data.

7. Mengumpulkan Data

Pada tahap ini dilakukan pengumpulan data dengan menggunakan teknik *scraping data* menggunakan *software* yang dimiliki oleh *Google Chrome*. Pada tahap ini dilakukan *scraping data* pada data ulasan pengguna Aplikasi Iflix di *website Google Play*.

8. Analisis Deskriptif

Pada tahap ini dilakukan analisis deskriptif. Tujuan dari tahap ini adalah untuk menggambarkan persepsi pengguna Aplikasi Iflix.

9. *Preprocessing Data*

Pada tahap ini dilakukan proses *pre processing data*, dimana pada proses ini digunakan *software Python*. Berikut merupakan beberapa proses yang terjadi:

- a. *Spelling Normalization*
- b. *Case Folding*
- c. *Stopward*
- d. *Tokenizing*

10. Pelabelan Data

Pada tahap ini dilakukan pelabelan dan pembobotan. Dimana proses pembobotan dilakukan dengan menggunakan Kamus Lexicon pada Python. Sedangkan untuk tahap pelabelan menggunakan Excel untuk mengetahui pembagian kelas positif dan negatif sesuai dengan pembobotan yang telah dilakukan sebelumnya.

11. Klasifikasi

Pada tahap ini dilakukan proses klasifikasi dengan menggunakan metode *Support Vector Machine*.

12. Visualisasi dan Asosiasi Teks

Pada tahap ini, dimana dilakukan proses visualisasi dan asosiasi teks pada data sentimen positif dan negatif.

13. Interpretasi Hasil dengan *Fishbone*

Pada tahap ini dilakukan proses pembuatan *fishbone* atau diagram sebab akibat berdasarkan hasil visualisasi sentimen negatif dengan tujuan untuk menganalisa faktor – faktor penyebab terjadinya masalah berdasarkan persepsi pengguna. Dimana hasil tersebut dapat berfungsi sebagai referensi dilakukan evaluasi atau perbaikan pelayanan dari Aplikasi Iflix.

14. Kesimpulan dan Saran

Pada tahap ini dilakukan pengambilan kesimpulan berdasarkan pengolahan data serta analisis yang telah dilakukan serta dapat memberikan saran berdasarkan penelitian yang telah dilakukan.

15. Selesai

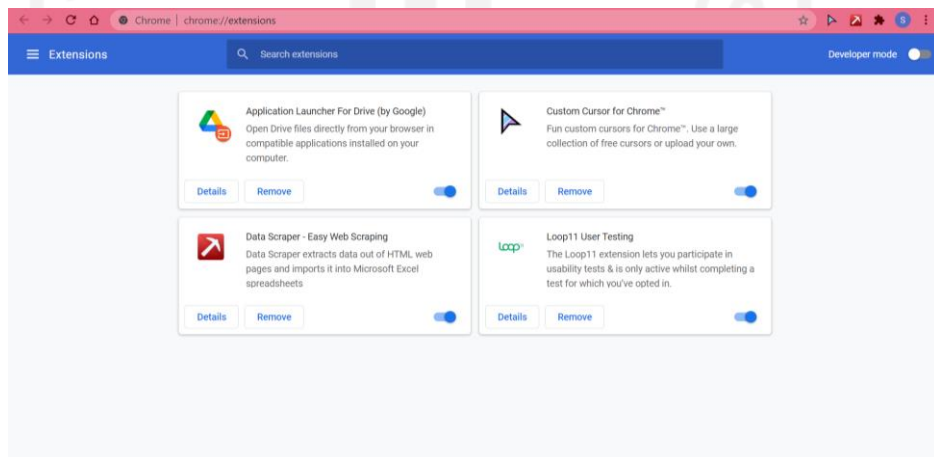


BAB IV

PENGUMPULAN DAN PENGOLAHAN DATA

4.1 Pengumpulan Data

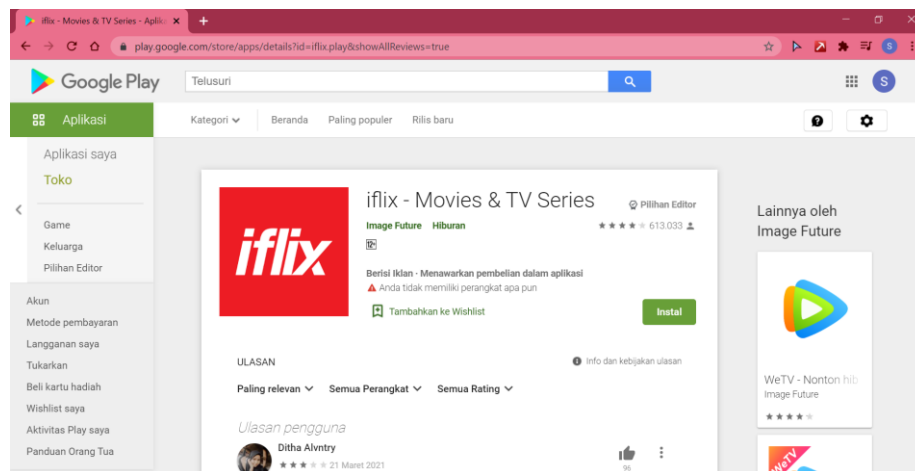
Pada penelitian ini dilakukan proses pengumpulan data, dimana data yang diambil merupakan data ulasan dari pengguna aplikasi Iflix pada *website Google Play*. Proses pengambilan data dilakukan dengan menggunakan teknik *scraping* melalui *Data Scraper* versi 5.2.7.4 yang dimiliki oleh *Google Chrome*. Terdapat beberapa langkah pengerjaan dalam melakukan *scraping data*. Langkah pertama yang dilakukan adalah menginstal aplikasi dan mengaktifkan ekstensi *Data Scraper* pada *Google Chrome* seperti yang ditunjukkan pada Gambar 4.1 dibawah ini.



Gambar 4 1 Ekstensi Data Miner

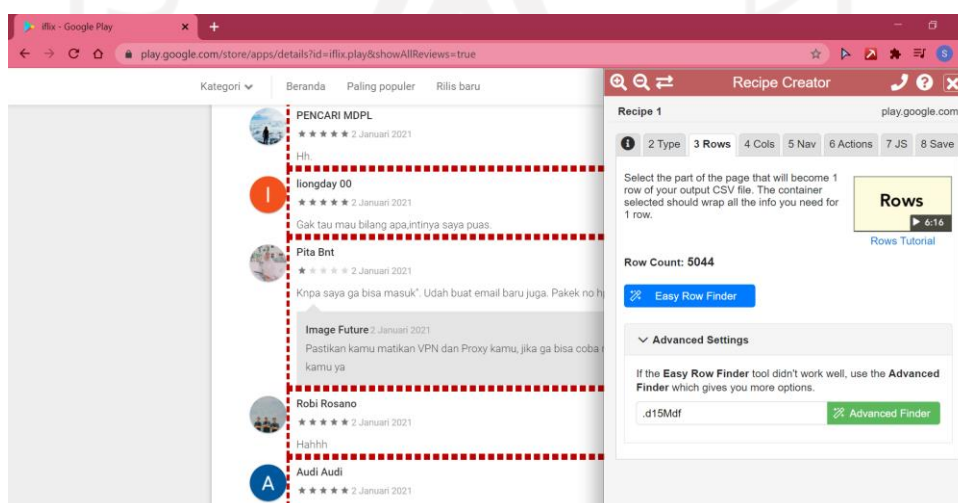
Pada penelitian ini terdapat sumber data yang digunakan berupa ulasan yang diberikan oleh penggunaan aplikasi Iflix, dimana alamat domain dari website ini <https://play.google.com/store/apps/details?id=iflix.play&showAllReviews=true>.

Pada langkah selanjutnya yang dilakukan adalah membuka alamat website diatas yang ditunjukkan seperti pada Gambar 4.2.

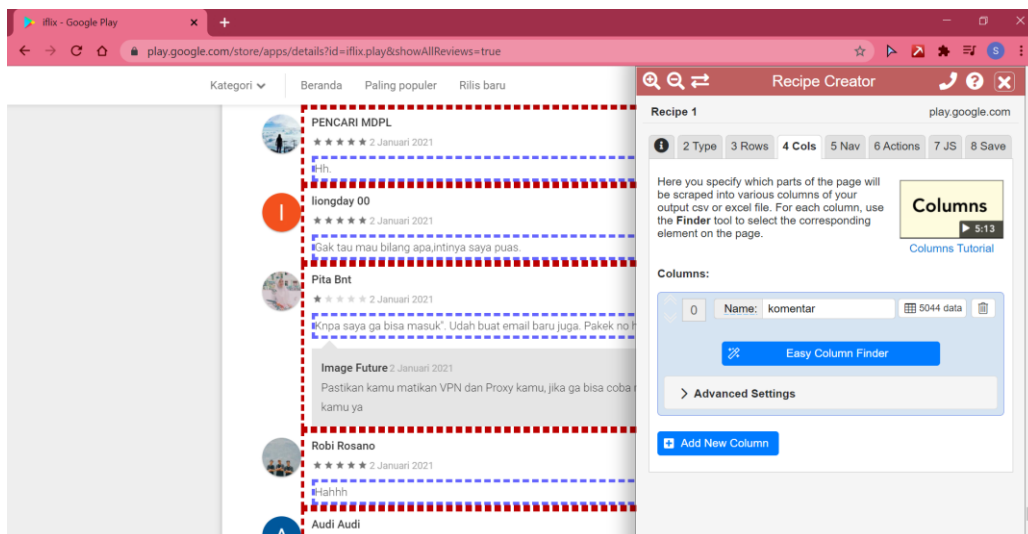


Gambar 4 2 Website Aplikasi Iflix

Pada tahap ini dilakukan pengambilan data sesuai dengan data yang dibutuhkan. Pertama dilakukan pemilihan *new recipe*. Selanjutnya klik *Type* untuk memiliki jenis tipe halaman yang diinginkan yaitu *list page*. Lalu, klik bagian *Rows* dan pilih *row* yang akan digunakan dalam penelitian seperti yang dilakukan pada Gambar 4.3 dibawah. Selanjutnya klik *Cols* dan pilih sesuai data yang digunakan dalam penelitian. Pada penelitian ini data yang diambil adalah tanggal dan komentar. Proses pada bagian *cols* dapat dilihat pada Gambar 4.4 dibawah ini.

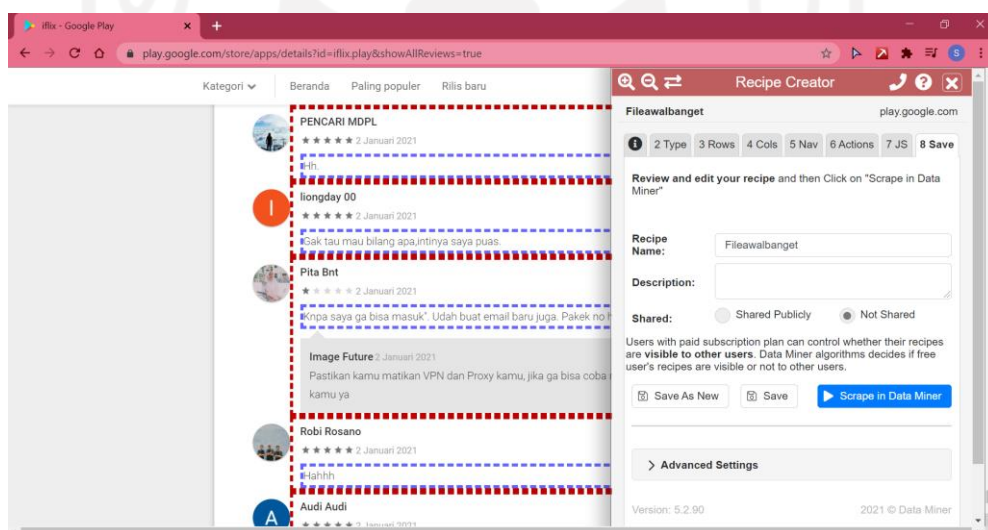


Gambar 4 3 Proses Row



Gambar 4 4 Proses Cols

Selanjutnya dilakukan *scraping data* dengan mengklik *scraping in Data Miner* dan mengisi *recipe name* seperti Gambar 4.5.



Gambar 4 5 Membuat Recipe

Hasil dari *scraping data* yang telah dilakukan dapat disimpan dengan beberapa format yaitu *CSV File* dan *Excel File*. Hasil dari *scraping data* dapat dilihat pada Tabel 4.1 dibawah ini.

Tabel 4 1 Hasil Scraping Data

No	Tanggal	Komentar	Rating
1	17 Maret 2021	Cukup menghibur dan bagus gambarnya...jernih	5
2	17 Maret 2021	Good	4

No	Tanggal	Komentar	Rating
3	17 Maret 2021	cuma pengen ngasih bintang 5 aja si,,kan lumayan dapet review bagus ydh salam ya buat yg buat ni aplikasi	5
4	17 Maret 2021	sangat bagus telah filem2 drama ??	5
5	17 Maret 2021	Sukaa aja sama aplksi ini	5
...
4501	1 Januari 2021	Terlalu banyak banget iklan klo bisa iklannya di kurangin lagi	4

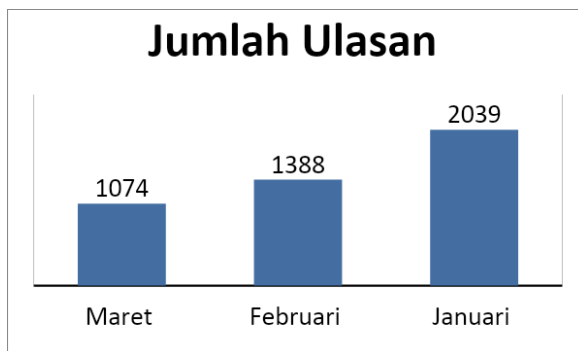
Hasil dari proses *scraping* yang telah dilakukan menghasilkan jumlah ulasan atau komentar sebesar 4.501 ulasan dengan 3 kolom yang terdiri atas kolom tanggal yang berisi tanggal ulasan itu dibuat, kolom komentar yang berisi ulasan serta kolom rating penilaian aplikasi Iflix yang diberikan oleh pengguna aplikasi.

4.2 Pengolahan Data

Berdasarkan data yang telah didapatkan pada proses sebelumnya. Selanjutnya dilakukan proses pengolahan data dengan menggunakan data ulasan pengguna Aplikasi Iflix sebanyak 4.501 ulasan. Pada penelitian ini dilakukan beberapa proses pengolahan data yaitu analisis deskriptif dan *pre processing data*.

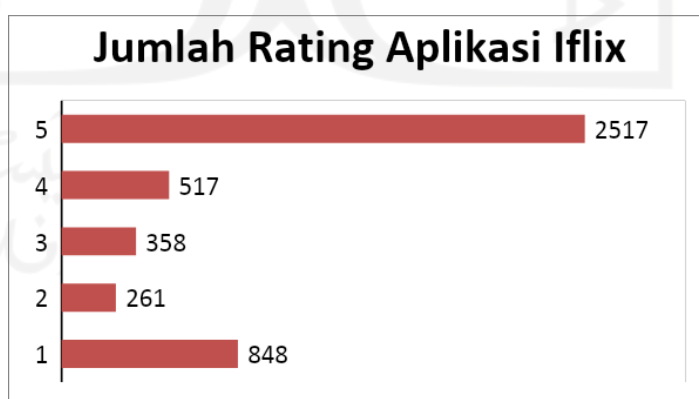
4.2.1 Analisis Deskriptif

Pada penelitian ini dilakukan analisis deskriptif, dimana analisis ini dilakukan dengan menggunakan hasil *scraping data* yang dilakukan pada tahap sebelumnya untuk mengetahui mengenai gambaran dari persepsi pengguna terkait pelayanan yang diberikan oleh Iflix.



Gambar 4 6 Grafik Jumlah Komentar

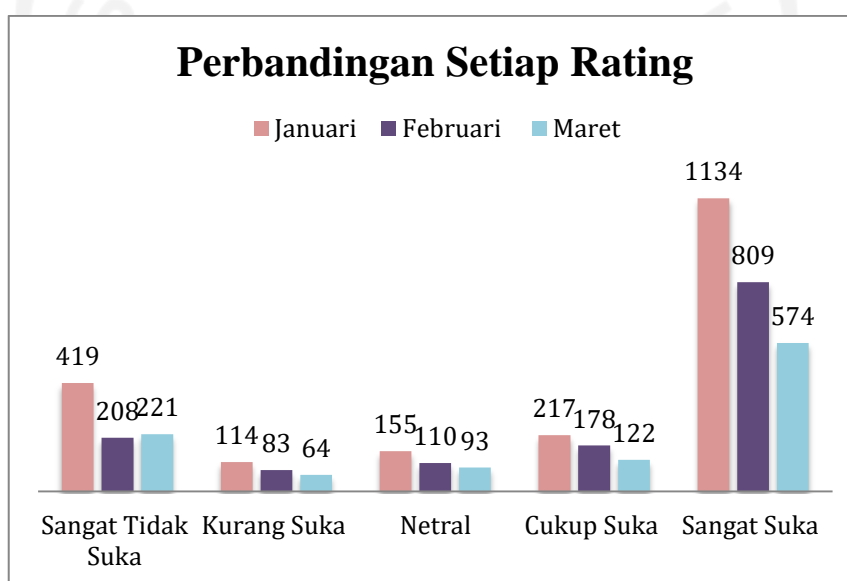
Pada Gambar 4.6 diatas, dimana rentang pengambilan data dari 1 januari 2021 sampai dengan 17 Maret 2021. Berdasarkan grafik diketahui bahwa hasil grafik sedikit menurun. Dimana terdapat selisih ulasan pada bulan Januari dan Februari sebesar 651 ulasan. Diketahui pada bulan Januari 2021 terdapat jumlah ulasan sebesar 2039 ulasan, sedangkan pada bulan Februari 2021 menurun menjadi 1388 ulasan. Selanjutnya pada bulan Maret 2021 sebesar 1074, meski data ulasan yang dimiliki pada bulan Maret hanya sampai tanggal 17 Maret 2021 namun sudah memiliki jumlah ulasan yang cukup banyak. Hal ini dapat disebabkan karena pada tanggal 8 Maret 2021 aplikasi ini baru saja diperbaharui sehingga para pengguna memberikan ulasan mengenai hasil perbaharuan dari aplikasi Iflix serta presepsi yang dirasakan.



Gambar 4 7 Jumlah Rating

Berdasarkan Gambar 4.7 diketahui jenis rating terbagi menjadi beberapa bagian yaitu rating 1 termasuk dalam kategori sangat tidak suka, rating 2 termasuk dalam kategori kurang suka, rating 3 termasuk dalam kategori netral, rating 4 masuk

dalam kategori cukup suka dan rating 5 termasuk dalam kategori sangat bagus. Pada grafik diatas, diketahui nilai terbanyak pertama dengan jumlah pengguna memberikan rating 5 atau sangat suka adalah 2.517 pengguna. Selanjutnya untuk jumlah rating terbanyak kedua dengan 848 pengguna memberikan rating 1 atau sangat tidak suka. Lalu untuk posisi ketiga dimiliki oleh rating 4 atau cukup suka dengan 517 pengguna. Kemudian diikuti oleh rating 3 atau netral dengan 358 pengguna dan rating 2 atau kurang suka dengan 261 pengguna.



Gambar 4 8 Perbandingan Rating Ulasan

Pada grafik diatas, dapat diketahui jumlah rating pada setiap kategori pada bulan 1 Januari 2021 hingga 17 Maret 2021. Berdasarkan Gambar 4.8 diketahui bahwa bulan Januari memiliki jumlah rating terbanyak untuk setiap kategorinya. Dimana pada rating kategori sangat suka dipilih oleh pengguna Iflix sebanyak 1.134 kali. Sedangkan pada kategori cukup suka sebanyak 217 kali. Lalu pada kategori netral, kurang suka dan sangat tidak suka masing – masing sebanyak 155 kali, 114 kali dan 419 kali. Sedangkan pada bulan selanjutnya yaitu bulan Februari mengalami sedikit penurunan. Dimana pada rating kategori sangat suka sebanyak 809 kali. Sedangkan pada kategori cukup suka sebanyak 178 kali. Lalu pada kategori netral, kurang suka dan sangat tidak suka masing – masing sebanyak 110 kali, 83 kali dan 208 kali. Pada bulan Maret awal, didapatkan jumlah rating sebanyak untuk setiap kategori

yaitu kategori sangat suka dipilih oleh pengguna Iflix sebanyak 574 kali. Sedangkan pada kategori cukup suka sebanyak 122 kali. Lalu pada kategori netral, kurang suka dan sangat tidak suka masing – masing sebanyak 93 kali, 64 kali dan 221 kali.

4.2.2 Pre Processing Data

Data yang didapatkan pada proses *scraping* pada tahap sebelumnya belum dapat digunakan karena masih memiliki bentuk kalimat yang masih belum terstruktur sehingga perlu dilakukan proses *preprocessing data*. Tahap *preprocessing* ini dilakukan dengan menggunakan *tools* berupa Python. Adapun tahapan *preprocessing* yang akan dilakukan adalah sebagai berikut ini.

4.2.2.1 Spelling Normalization

Pada bagian *spelling normalization* ini dilakukan dengan tujuan untuk melakukan perbaikan terkait kata – kata singkatan atau *slang* yang ada pada data ulasan yang digunakan dalam penelitian, sebagai contoh pada kata “lg” diubah menjadi “lagi” dan contoh lainnya terdapat kesalahan ejaan seperti “tdk” yang diperbaiki menjadi “tidak”. Adapun contoh pada kalimat ulasan sebagai berikut ini:

Tabel 4 2 Proses *Spelling Normalization*

Sebelum	Sesudah
Mantap Tambah lg film barunya TDK ada	Mantap Tambah lagi film barunya tidak ada

Adapun *pseudo code* pada proses *Spelling Normalization* seperti dibawah ini.

Tabel 4 3 *Pseudo Code Spelling Normalization*

<i>Pseudo Code</i>	Fungsi
<code>dictawal = ['yg', 'udh', 'gtu', 'ancur']</code>	Mengubah kata – kata

<i>Pseudo Code</i>	Fungsi
<pre>dictakhir = ['yang', 'udah', 'gitu', 'hancur'] data['komentar'] = data['komentar'].replace(dictawal, dictakhir, regex=True)</pre>	singkatan seperti “yg”, “udh”, “gtu”, “ancur” menjadi “yang”, “sudah”, “gitu”, “hancur”.

4.2.2.2 Case Folding

Pada tahap *preprocessing* terdapat bagian *case folding* dengan tujuan dari bagian ini adalah untuk menyeragamkan bentuk huruf atau kata kedalam bentuk huruf kecil serta dapat menghilangkan tanda baca yang ada dalam kalimat ulasan. Hal tersebut dilakukan agar kata yang memiliki huruf kecil dan besar tidak terdeteksi atau diartikan memiliki perbedaan arti. Tidak terdapat aturan pasti mengenai tahapan pada *text preprocessing*. Namun memastikan hasil yang lebih baik dan konsisten apabila semua tahapan dilakukan. Semakin banyaknya tahapan yang dilakukan, maka semakin banyak pula lapisan yang dikupas dalam data tersebut. Namun salah satu tahapan yang perlu dilakukan adalah *Case folding*. Dimana tahapan ini merupakan tahapan *preprocessing* yang paling sederhana (Nugroho, 2019). Pemrosesan dalam *Case Folding* seperti konversi huruf kecil dapat meningkatkan keberhasilan klasifikasi dalam hal akurasi dan membantu mengelompokkan istilah yang mengandung informasi yang sama (Uysal, 2014). Adapun contoh pada kalimat ulasan sebagai berikut ini:

Tabel 4 4 Proses Case Folding

Sebelum	Sesudah
BAGUS banget aplikasi NYA	bagus banget aplikasinya
WOW BAGUS banget dah itu aja	wow bagus banget dah itu aja
TOLONG PERBAIKI LAGI aplikasi iflix NYA!	tolong perbaiki lagi aplikasi iflix nya

Berikut merupakan *psudo code* yang digunakan dalam proses ini.

Tabel 4 5 *Pseudo Code Case Folding*

<i>Pseudo Code</i>	Fungsi
--------------------	---------------

<i>Pseudo Code</i>	Fungsi
<pre>def cleansing (data): data = data.lower() remove = string.punctuation translator = str.maketrans(remove, '*1 en(remove)) data = data.translate (translator) angber = ''.join([x for x in data if not (x.isdigit()or x[0] == '-' ' and x[1:].isdigit())]) return angber</pre>	<p>Mengubah huruf besar pada kalimat menjadi huruf kecil.</p> <p>Menghilangkan tanda baca pada kalimat yang ada pada dokumen.</p>

4.2.2.3 Tokenizing

Pada tahap tokenizing merupakan suatu proses pemisahan teks kedalam dokumen menjadi potongan – potongan kata yang saling berpengaruh. Kata-kata tersebut menggunakan tanda koma (,) sebagai pemisahannya. Berikut merupakan contoh penerapan *tokenizing* pada kalimat ulasan yang dapat dilihat pada tabel dibawah ini:

Tabel 4 6 Proses Tokenizing

Sebelum	Sesudah
Iklannya lama harus pakai video apa itu minta minta pulsa tolong iklannya kurangi terlalu lama	'iklannya', 'lama', 'harus', 'pakai', 'video', 'apa', 'itu', 'minta', 'minta', 'pulsa', 'tolong', 'iklannya', 'kurangi', 'terlalu', 'lama'
bagus banget	'bagus', 'banget'
sangat mudah untuk ditonton	'sangat', 'mudah', 'untuk', 'ditonton'

Pada tabel 4.7 dibawah ini memperlihatkan *psecode* pada tahap *Tokenizing*.

Tabel 4 7 Pseudo Code Tokenizing

<i>Pseudo Code</i>	Fungsi
<pre>def tokenize (txt): word_token = word_tokenize (txt) return word_token</pre>	<p>Mengubah kalimat menjadi potongan kata yang dipisahkan dengan koma.</p>

4.2.2.4 Stopword Removal

Pada tahap ini dilakukan proses penghilangan kata yang ada pada kalimat atau dokumen dengan menggunakan *stopword removal*. *Stopword Removal* digunakan dengan tujuan menghilangkan kata yang tidak memiliki pengaruh pada kalimat dan tidak mengurangi informasi atau makna dari kalimat itu sendiri. Tahap *stopword removal* akan banyak menghemat waktu proses, ruang memori serta tidak merusak efektivitas dari informasi yang diambil (Jha, 2016). Dilakukannya proses *stopword removal* maka informasi yang dihasilkan lebih terfokus pada informasi yang penting dan penghapusan dilakukan kepada *text* yang memiliki informasi tingkat rendah. Proses *Stopword removal* akan mengurangi ukuran dataset dengan demikian akan mengurangi waktu proses pelatihan dikarenakan jumlah data yang digunakan akan berkurang. Apabila tahap *stopword removal* dihilangkan maka akan mempengaruhi waktu pengolahan data serta jumlah data set juga lebih banyak dibandingkan ketika menggunakan *stopword removal* (Khanna, 2021). Adapun proses *Stopword Removal* dapat dilakukan sebagai berikut ini:

Tabel 4 8 *Stopword Removal*

Sebelum	Sesudah
<i>saya</i> sangat menyukai aplikasi <i>ini</i> sangat mudah <i>untuk</i> mencari film kadang <i>di</i> youtube <i>tidak ada dan</i> mudah nyari <i>di</i> sini	'sangat', 'menyukai', 'aplikasi', 'sangat', 'mudah', 'mencari', 'film', 'kadang', 'youtube', 'mudah', 'nyari', 'sini'
<i>saya</i> sangat puas <i>dengan</i> aplikasi <i>ini</i>	'sangat', 'puas', 'aplikasi'
maaf <i>ya</i> cuma satu <i>karena saya</i> mau nonton <i>tidak ada</i> jadi mohon diperbaiki dulu	'maaf', 'cuma', 'satu', 'mau', 'nonton', 'jadi', 'mohon', 'diperbaiki', 'dulu']

Pada Tabel 4.9 merupakan *pseudo code* pada tahap *Stopword Removal*.

Tabel 4 9 Pseudo Code Stopword Removal.

<i>Pseudo Code</i>	Fungsi
def tokenize (txt):	Mengubah kalimat menjadi potongan
word_token = word_tokenize (txt)	kata yang dipisahkan dengan koma.
return word_token	

4.2.3 Pelabelan

Setelah dilakukan tahap *preprocessing*, maka tahap selanjutnya adalah tahap pelabelan. Dalam melakukan proses pelabelan dapat dilakukan dengan beberapa cara yaitu secara manual dan menggunakan *Lexicon-based*. Proses pelabelan secara manual dilakukan berdasarkan pengetahuan domain individu dan pemahaman bahasa. Namun proses ini akan memakan waktu yang cukup banyak (Verma, 2018). Metode lainnya yang sering digunakan adalah *Lexicon-based*. Metode *Lexicon-based* dilakukan dengan cara membuat suatu kamus *Lexicon*. Dimana kamus tersebut akan digunakan untuk mengidentifikasi apakah kata tersebut mengandung opini atau tidak. *Lexicon* adalah kumpulan kata sentimen yang telah diketahui dan terhimpun (Desai, 2016). Proses pelabelan dibutuhkan kamus atau *lexicon* yang berisi kata – kata yang mengandung sentimen yang disebut dengan *sentient dictionaries* (Buntoro, 2014). Sentimen yang digunakan berupa sentimen negatif dan positif. Dalam kamus *lexicon* setiap kata sudah dilengkapi dengan bobot kata. Rumus perhitungan skor sentimen yang digunakan pada tahap ini adalah sebagai berikut ini:

$$\text{Skor Sentimen} = \sum \text{Bobot Kata Positif} + \sum \text{Bobot Kata Negatif} \quad (4.1)$$

Dimana apabila di akumulatiskan kata – kata tersebut dapat digolongkan menjadi kategori sentimen positif dan sentimen negatif. Dinyatakan sebagai kategori sentimen negatif apabila nilai skor < 0 atau bernilai negatif dan dinyatakan termasuk kedalam kategori sentimen positif apabila nilai skor > 0 atau bernilai positif. Proses tersebut dapat dilihat pada gambar dibawah ini:

Tabel 4 10 Proses Pelabelan dengan Lexicon

Ulasan	Kata Positif	Kata Negatif	Skor
Filmnya lengkap dikit banget	Lengkap(2) banget(1)	dikit (-4)	-1
Seru banget ingilangin jenuh	seru (5) banget (1)	-	6

Hasil dari proses pelabelan dengan menggunakan kamus *lexicon* sangat bergantung dengan bobot kata yang telah tersedia di dalam kamus *lexicon*. Hal

tersebut mengakibatkan terdapat beberapa kata yang tidak dapat diidentifikasi dengan tepat. Adapun hasil dari proses pelabelan dengan menggunakan kamus *lexicon* dibawah ini.

Tabel 4 11 Hasil Pelabelan

Kelas Sentimen	Kamus <i>Lexicon</i>
Positif	3255
Negatif	661
Netral	445
Total	4361

Berdasarkan pada tabel diatas, dapat dijelaskan bahwa kelas sentimen berdasarkan kamus *lexicon* yaitu kelas sentimen positif berjumlah 3.255 ulasan, kelas sentimen negatif sebesar 661 ulasan dan kelas sentimen netral sebesar 455 ulasan. Dimana kelas sentimen negatif merupakan pernyataan ketidakpuasan pengguna terhadap aplikasi dan kelas sentimen positif menandakan kepuasan pengguna terhadap aplikasi.

Selanjutnya dilakukan penghilangan sentimen netral pada hasil pelabelan, hal itu dikarenakan pada penelitian ini hanya berfokus pada sentimen negatif dan positif saja sehingga sentimen netral perlu dihilangkan. Berikut merupakan hasil pelabelan kelas sentimen pada ulasan aplikasi Iflix:

Tabel 4 12 Hasil Akhir Pelabelan

Kelas Sentimen	Kamus <i>Lexicon</i>
Positif	3.255
Negatif	661
Total	3.916

Maka hasil pelabelan yang didapatkan adalah hasil kelas sentimen positif sebesar 3.255 ulasan dan kelas sentimen negatif sebesar 661 ulasan. Adapun *pseudo code* yang digunakan pada proses pelabelan adalah sebagai berikut ini.

Tabel 4 13 *Pseudo Code* Proses Pelabelan

<i>Pseudo Code</i>	Fungsi
<code>from google.colab import files</code>	Memasukkan file
<code>uploaded = files.upload()</code>	KamusLexiconFull

<i>Pseudo Code</i>	Fungsi
<pre> import pandas as pd lexicon = pd.read_csv('KamusLexicon Full.csv', encoding= 'unicode_escape') print(lexicon) from google.colab import files uploaded = files.upload() data = pd.read_csv('Preproces.csv').reset_index(drop=True) import nltk nltk.download('punkt') from nltk.tokenize import word_tokenize import numpy as np !pip install Sastrawi from Sastrawi.Stemmer.StemmerFactory import StemmerFactory sencol = [] senrow = np.array([]) nsen = 0 factory = StemmerFactory() stemmer = factory.create_stemmer() sentiment_list = [] def found_word(ind, words, word, sen, sencol, sentiment, add): if word in sencol: sen[sencol.index(word)] += 1 else: sencol.append(word) sen.append(1) add += 1 if (words[ind-1] in negasi): sentiment += - lexicon['weight'][lexicon._word.index(word)] else: sentiment += lexicon['weight'][lexicon._word.index(word)] return sen, sencol, sentiment, add for i in range(len(data)): </pre>	<p>Menjalankan <i>library</i> pandas</p> <p>Membaca file KamusLexiconFull</p> <p>Memasukkan file Preprocess yang akan digunakan dalam proses pembobotan dengan kamus lexicon.</p> <p>Membaca file Preproces</p> <p>Menjalankan <i>library</i> nltk</p> <p>Menjalankan <i>library</i> numpy</p> <p>Menginstall <i>library</i> Sastrawi</p> <p>Melakukan proses pelabelan :</p> <ol style="list-style-type: none"> 1. Melakukan pelabelan sesuai dengan kamus lexicon yang ada 2. Menulis sentimen kata jika di temukan. 3. Apabila sudah menemukan masuk kedalam bagan matrik maka tambahkan nilai.

<i>Pseudo Code</i>	Fungsi
<pre> nsen = senrow.shape[0] words = word_tokenize(data['data_ke mbali'][i]) sentiment = 0 add = 0 prev = [0 for ii in range(len(words))] n_words = len(words) if len(sencol)>0: sen =[0 for j in range(len(sencol))] else: sen =[] for word in words: ind = words.index(word) if word in lexicon_word : sen,sencol,sentiment,add= found_wor d(ind,words,word,sen,sencol,sentimen t,add) else: kata_dasar = stemmer.stem(word) if kata_dasar in lexicon_word: sen,sencol,sentiment,add= found_wor d(ind,words,kata_dasar,sen,sencol,sen timent,add) elif(n_words>1): if ind-1>-1: back_1 = words[ind-1]+' '+word if (back_1 in lexicon_word): sen,sencol,sentiment,add= found_wor d(ind,words,back_1,sen,sencol,sentim ent,add) elif(ind-2>-1): back_2 = words[ind-2]+' '+back_1 if back_2 in lexicon_word: sen,sencol,sentiment,add= found_wor d(ind,words,back_2,sen,sencol,sentim ent,add) if add>0: if i>0: if (nsen==0): senrow = np.zeros([i,add],dtype=int) elif(i!=nsen): padding_h = np.zeros([nsen,add],dtyp e=int) </pre>	<p>4. Memeriksa setiap kata</p> <p>5. jika kata – kata tersebut ada di dalam kamus lexicon, lalu menghitung sentimen jika muncul</p> <p>6. Periksa apakah kata ada di dalam lexicon, jika tidak ada maka periksa kata dasar.</p> <p>7. Jika nilai kata negatif maka periksa dengan kata terdekatnya.</p>

<i>Pseudo Code</i>	Fungsi
<pre> senrow = np.hstack((senrow,padding_ h)) padding_v = np.zeros([(i- nsen),senrow.shape[1]],dtype=int) senrow = np.vstack((senrow,padding_ v)) else: padding =np.zeros([nsen,add],dtype=i nt) senrow = np.hstack((senrow,padding)) senrow = np.vstack((senrow,sen)) if i==0: senrow = np.array(sen).reshape(1,len(sen)) elif(nsen>0): senrow = np.vstack((senrow,sen)) sentiment_list.append(sentiment) sencol.append('sentiment') sentiment_array = np.array(sentiment_ list).reshape(senrow.shape[0],1) sentiment_data = np.hstack((senrow,s entiment_array)) df_sen = pd.DataFrame(sentiment_dat a,columns = sencol) df_sen </pre>	<p>8. Jika ada kata baru yang di temukan maka perluas matriks.</p> <p>9. Jika tidak ada maka perbaharui matriks lama.</p> <p>Mendefinisikan hasil proses sebelumnya menjadi sebuah tabel.</p> <p>Mendefinisikan hasil proses sebelumnya yang diubah bentuknya menjadi ‘df_sen’</p> <p>Memanggil fungsi hasil pada proses pelabelan.</p>

4.2.4 Pembagian Data Latih dan Data Uji

Setelah dilakukan proses pelabelan, maka selanjutnya dilakukan tahap pembagian data, dimana data dibagi menjadi beberapa bagian yaitu data latih (*training*) dan data latih (*testing*). Pembagian data latih dan data uji dilakukan berdasarkan beberapa penelitian yang telah dilakukan. Dimana penelitian pertama dilakukan oleh Romadoni (Romadoni, 2020). Lalu penelitian lainnya sebagai acuan pembagian data latih dan data uji adalah penelitian yang dilakukan oleh Erni Harahap (Harahap, 2018). Dimana pembagian data yang digunakan dari beberapa penelitian itu terbagi

menjadi beberapa skenario yaitu 60:40,70:30,80:20 dan 90:10. Berikut merupakan jumlah data latih dan data uji yang digunakan dalam penelitian ini:

Tabel 4 14 Pembagian Data Uji dan Data Latih

Skenario	Data Latih	Data Uji	Total
Skenario 1 (60:40)	2.349	1.567	3.916
Skenario 2 (70:30)	2.741	1.175	3.916
Skenario 3 (80:20)	3.132	784	3.916
Skenario 4 (90:10)	3.254	392	3.916

Pada pembagian data latih dan data uji di ketahui pada skenario 1 dengan 60 % data latih atau 2.349 ulasan dan 40% data uji atau 1.567 ulasan. Skenario 2 dengan 70 % data latih atau 2.741 ulasan dan 30% data uji atau 1.175 ulasan. Skenario 3 dengan 80 % data latih atau 3.132 ulasan dan 20% data uji atau 784 ulasan. Lalu skenario 4 dengan 90 % data latih atau 3.254 ulasan dan 10% data uji atau 392 ulasan.

Dalam melakukan pembagian data *training* dan data *testing* terdapat *pseudo code*. Berikut merupakan contoh pembagian data *training* dan *testing* pada salah satu skenario yang digunakan.

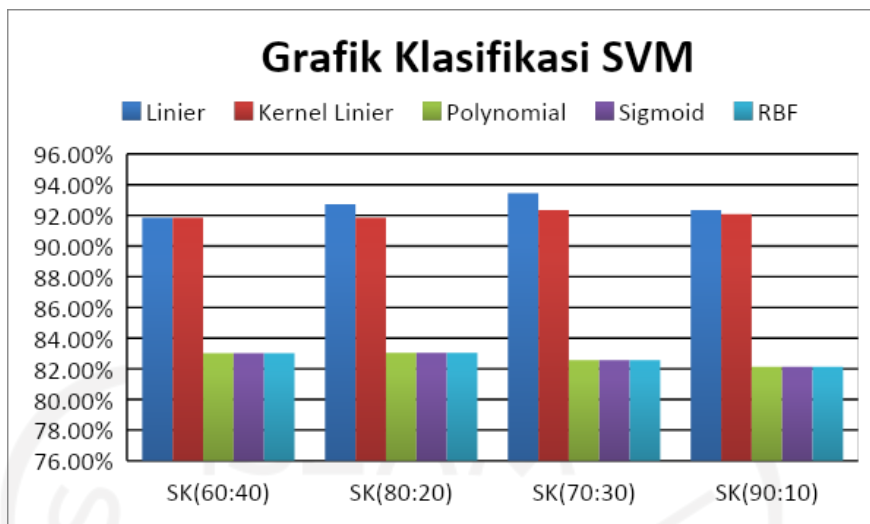
Tabel 4 15 *Pseudo Code* Pembagian Data *Training* dan *Testing*

<i>Pseudo Code</i>	Fungsi
<code>from google.colab import files</code> <code>uploaded = files.upload()</code>	Memasukkan file data yang digunakan pada tahap ini
<code>Data = pd.read_csv('hasil_label_sentimen.csv')</code> <code>print(Data)</code>	Membaca isi dari file yang dimasukkan.
<code>import numpy as np</code> <code>import pandas as pd</code> <code>import nltk</code> <code>from nltk.corpus import stopwords</code> <code>import string</code> <code>import pickle</code>	Memasukkan <i>library</i> numpy, <i>library</i> pandas, <i>library</i> nltk, <i>library</i> string, <i>library</i> pickle, <i>library</i> sklearn.
<code>from sklearn import svm</code> <code>x, y = Data.komentar, Data.kelas</code>	Mendefinisikan nama row data pada

<i>Pseudo Code</i>	Fungsi
<pre> from sklearn.model_selection import train_test_split X_train, X_test, y_train, y_test = train_test_split(x,y, test_size = 0.20, random_state = 0) </pre>	<p>tabel di file "hasil_label_sentimen.csv"</p> <p>Melakukan split data pada data <i>training</i> 80 dan data <i>testing</i> 20.</p>
<pre> from sklearn.feature_extraction.text import TfidfVectorizer vectorizer=TfidfVectorizer(input = 'content',lowercase=True,) features_train_transformed = vectorizer.fit_transform(X_train) features_test_tranformer = vectorizer.transform(X_test) print (features_train_transformed.shape) print (features_test_tranformer.shape) </pre>	<p>Mengekstrak data berdasarkan data yang telah dimasukkan.</p> <p>Melihat jumlah pembagian data <i>training</i> dan data <i>testing</i> pada skenario 80:20.</p>

4.2.5 Klasifikasi dengan Metode *Support Vector Machine*

Setelah dilakukan pembagian data uji dan data testing sesuai dengan skenario yang dilakukan, maka tahap selanjutnya adalah tahap klasifikasi dengan menggunakan metode *Support Vector Machine*. Pada penelitian ini dilakukan klasifikasi untuk dua jenis set yang berbeda yaitu linier dan nonlinier. Pada klasifikasi non linier menggunakan fungsi kernel yang terdiri atas *kernel linier*, *kernel polynomial*, *sigmoid* dan *radial basis function* (RBF). Kernel adalah fungsi yang digunakan dalam SVM untuk membantu memecahkan masalah. Kernel merupakan suatu fungsi yang memetakan data sehingga mudah dipisahkan (Souza, 2010). Selanjutnya dilakukan tahap pemilihan parameter optimum dengan tujuan mendapatkan hasil yang maksimum maka perlu dilakukan uji dengan menggunakan fungsi klasifikasi SVM diatas. Setelah dilakukan perbandingan dengan menggunakan beberapa skenario, maka didapatkan hasil seperti grafik dibawah ini:



Gambar 4 9 Hasil Klasifikasi

Berdasarkan hasil yang ditunjukkan pada Gambar 4.9 diatas, diketahui nilai akurasi untuk setiap algoritma SVM baik linier maupun non linier. Pada algoritma SVM linier menggunakan 4 skenario dengan nilai masing – masing akurasi untuk setiap skenario yaitu skenario 1 dengan data latih 60% dan data uji 40% sebesar 92,98%, skenario 2 dengan data latih 70 % dan data uji 30% sebesar 93,45% , skenario 3 dengan data latih 80% dan data uji 20% sebesar 92,73%, skenario 4 dengan data latih 90% dan data uji 10% sebesar 92,35%.

Pada algoritma SVM kernel linear (non linear) akurasi yang dihasilkan yaitu skenario 1 dengan data latih 60% dan data uji 40% sebesar 91,83%, skenario 2 dengan data latih 70 % dan data uji 30% sebesar 92,34% , skenario 3 dengan data latih 80% dan data uji 20% sebesar 91,83%, skenario 4 dengan data latih 90% dan data uji 10% sebesar 92,09%. Pada algoritma SVM non linear polynomial, sigmoid dan RBF memiliki nilai akurasi yang sama dengan nilai akurasi yaitu skenario 1 dengan data latih 60% dan data uji 40% sebesar 83,02%, skenario 2 dengan data latih 70 % dan data uji 30% sebesar 82,55% , skenario 3 dengan data latih 80% dan data uji 20% sebesar 83,04%, skenario 4 dengan data latih 90% dan data uji 10% sebesar 82,14%.

Pada Gambar 4.9 diatas, ditunjukkan bahwa skenario yang memiliki nilai akurasi tertinggi adalah scenario data latih 70 persen dan data uji 30 persen dengan nilai akurasi sebesar 93,45% dengan fungsi klasifikasi linier.

Tabel 4 16 Hasil Klasifikasi 70:30

	Aktual		
	Linier	Positif	Negatif
Prediksi	Positif	145	60
	Negatif	17	653
Akurasi		93.45%	

Pada tabel 4.16 memperlihatkan hasil dari *confusion matrix*, dimana diketahui dengan pada kasus ini jumlah prediksi positif atau (*true positive*) sebesar 145 dan jumlah prediksi negatif yang sesuai (*true negative*) sebesar 653.

Proses yang dilakukan diatas, dilakukan dengan menjalannya *pseude code*. Berikut *pseude code* pada salah satu skenario yang dijalankan.

Tabel 4 17 *Pseudo Code* Klasifikasi SVM

<i>Pseudo Code</i>	Fungsi
<pre>SVM_Clasifier = svm.Linear SVC () SVM_Clasifier.fit(features _train_transformed,y_train) SVM_Clasifier_kernel_linie r = svm.SVC(C=1.0, kernel= 'linear', degree=8, gamma= 'auto')</pre>	<p>Mendefinisikan SVM <i>Classifier</i> dengan parameter yang digunakan.</p>
<pre>SVM_Clasifier_kernel_linie r.fit(features_train_trans formed,y_train) SVM_Clasifier_kernel_poly = svm.SVC(C=1.0, kernel='p oly', degree=8,gamma='auto ')</pre>	<p>Mencocokkan data pada pembagian data <i>training</i> dan data <i>testing</i> sesuai fungsi SVM yang digunakan.</p>
<pre>SVM_Clasifier_kernel_poly. fit(features_train_transfo rmed,y_train) SVM_Clasifier_kernel_sigmo id = svm.SVC(C=1.0, kernel ='sigmoid', degree=8, gamm a='auto')</pre>	
<pre>SVM_Clasifier_kernel_sigmo id.fit(features_train_tran sformed,y_train) SVM_Clasifier_rbf = svm.SV</pre>	

<i>Pseudo Code</i>	Fungsi
<pre> C(C=1.0, kernel='rbf', deg ree=8, gamma='auto') SVM_Clasifier_rbf.fit(featu res_train_transformed,y_t rain) from sklearn.metrics impor t classification_report,co nfusion_matrix,accuracy_sc ore predictions_SVM = SVM_Clas ifier.predict(features_tes t_tranformer) print('linier') print(classification_repor t(y_test ,predictions_SVM)) print('Confusion Matrix: \ n',confusion_matrix(y_test ,predictions_SVM)) print() print('Accuracy: ', accura cy_score(y_test,predictio s_SVM)) from sklearn.metrics impor t classification_report,co nfusion_matrix,accuracy_sc ore predictions_SVM = SVM_Clas ifier_kernel_linier.pedic t(features_test_tranformer) print('kernel linier') print(classification_repor t(y_test ,predictions_SVM)) print('Confusion Matrix: \ n',confusion_matrix(y_test ,predictions_SVM)) print() print('Accuracy: ', accura cy_score(y_test,predictio s_SVM)) from sklearn.metrics impor </pre>	<p>Melakukan prediksi pada SVM Linier</p> <p>Memanggil hasil dari proses klasifikasi yang berisi atas laporan klasifikasi, <i>confusion matrix</i> dan akurasi.</p>
<pre> from sklearn.metrics impor t classification_report,co nfusion_matrix,accuracy_sc ore predictions_SVM = SVM_Clas ifier_kernel_linier.pedic t(features_test_tranformer) print('kernel linier') print(classification_repor t(y_test ,predictions_SVM)) print('Confusion Matrix: \ n',confusion_matrix(y_test ,predictions_SVM)) print() print('Accuracy: ', accura cy_score(y_test,predictio s_SVM)) from sklearn.metrics impor </pre>	<p>Melakukan prediksi pada SVM Kernel Linier</p> <p>Memanggil hasil dari proses klasifikasi yang berisi atas laporan klasifikasi, <i>confusion matrix</i> dan akurasi.</p>
<pre> from sklearn.metrics impor </pre>	<p>Melakukan prediksi pada SVM Kernel</p>

<i>Pseudo Code</i>	Fungsi
<pre>t classification_report,co nfusion_matrix,accuracy_sc ore predictions_SVM = SVM_Clas ifier_kernel_poly.predict(features_test_tranformer) print('polynomial') print(classification_repor t(y_test ,predictions_SVM)) print('Confusion Matrix: \ n',confusion_matrix(y_test ,predictions_SVM)) print() print('Accuracy: ', accura cy_score(y_test,predictio s_SVM)) from sklearn.metrics impor t classification_report,co nfusion_matrix,accuracy_sc ore predictions_SVM = SVM_Clas ifier_kernel_sigmoid.predi ct(features_test_tranforme r) print('sigmoid') print(classification_repor t(y_test ,predictions_SVM)) print('Confusion Matrix: \ n',confusion_matrix(y_test ,predictions_SVM)) print() print('Accuracy: ', accura cy_score(y_test,predictio s_SVM)) from sklearn.metrics impor t classification_report,co nfusion_matrix,accuracy_sc ore predictions_SVM = SVM_Clas ifier_rbf.predict(features _test_tranformer) print('RBF')</pre>	<p>Polynomial.</p> <p>Memanggil hasil dari proses klasifikasi yang berisi atas laporan klasifikasi, <i>confusion matrix</i> dan akurasi.</p> <p>Melakukan prediksi pada SVM Kernel Sigmoid</p> <p>Memanggil hasil dari proses klasifikasi yang berisi atas laporan klasifikasi, <i>confusion matrix</i> dan akurasi.</p> <p>Melakukan prediksi pada SVM Kernel RBF.</p> <p>Memanggil hasil dari proses klasifikasi yang berisi atas laporan klasifikasi, <i>confusion matrix</i> dan akurasi..</p>

<i>Pseudo Code</i>	Fungsi
<pre> print(classification_report(y_test ,predictions_SVM)) print('Confusion Matrix: \n',confusion_matrix(y_test ,predictions_SVM)) print() print('Accuracy: ', accuracy_score(y_test,predictions_SVM)) input_transformed = vectorizer.transform(Data) SVM_linier_prediction = SVM_Clasifier.predict(input_transformed) SVM_kernel_linier_prediction = SVM_Clasifier_kernel_linier.predict(input_transformed) SVM_poly_prediction = SVM_Clasifier_kernel_poly.predict(input_transformed) SVM_sigmoid_prediction = SVM_Clasifier_kernel_sigmoid.predict(input_transformed) SVM_rbf_prediction = SVM_Clasifier_rbf.predict(input_transformed) result = pd.DataFrame({"text awal" : Data,"label" : SVM_linier_prediction}) </pre>	<p>Melakukan klasifikasi kedua kali namun tidak dilakukan pada keseluruhan data tidak dilakukan pembagian.</p> <p>Memanggil hasil dari klasifikasi sesuai dengan model yang memiliki akurasi tertinggi.</p>

4.2.6 Visualisasi dan Asosiasi

Pada tahap ini dilakukan proses visualisasi dan asosiasi dilakukan pada sentimen positif maupun sentimen negatif. Tahap ini dilakukan dengan tujuan untuk mendapatkan informasi mengenai apa yang sering muncul terkait dengan persepsi dari pengguna aplikasi Iflix. Proses visualisasi dilakukan dengan menggunakan diagram batang dan *wordcloud* untuk memberikan kemudahan dalam melakukan

analisis. Sedangkan pada proses asosiasi dilakukan dengan tujuan dapat mengetahui kata – kata yang sering keluar secara bersamaan pada sentimen positif dan sentimen negatif. Proses visualisasi, dilakukan dengan menjalankan *pseude code* dibawah ini.

Tabel 4 18 *Pseude Code* Visualisasi Positif dan Negatif

<i>Pseude Code</i>	Fungsi
<pre>import nltk nltk.download('punkt') from nltk.tokenize import word_tokenize from nltk.probability import FreqDist visual = pd.DataFrame([]) visual['teks'] = df['label'].values.copy() visual</pre>	<p>Memanggil <i>library</i> nltk.</p> <p>Mendefinisikan visual .</p>
<pre>import matplotlib.pyplot as plt</pre>	<p>Memanggil <i>library</i> matplotlib untuk barplot.</p>
<pre>df = pd.read_csv('HasilSVM.csv') neg = df.loc[df['label'] == 'Negatif'] pos = df.loc[df['label'] == 'Positif'] pos_dict = {} for i in range(0, len(positif['text awal'])): sentence = positif['text awal'][i] token = word_tokenize(str(sentence)) for b in token: if b not in pos_dict: pos_dict[b] = 1 else: pos_dict[b] += 1 positif_top = pd.DataFrame((pos_dict.items())) positif_top = positif_top.rename(columns={0:'word', 1:'counts'}) positif_top = positif_top.sort_values(by='counts', ascending=False) positif_top.head(20).sort_values(by='counts',ascending=True).plot.barh(x='word', y='counts', rot=0,figsize=(20,10),fontsize=30) plt.savefig('grafik positif .png',bbo</pre>	<p>Mendefinisikan ‘HasilSVM’ menjadi ‘neg’ dan ‘positif’</p> <p>Menghitung berapa jumlah kata yang muncul pada keseluruhan ulasan positif.</p> <p>Menampilkan Bar plot pada visualisasi positif.</p>

<i>Pseudo Code</i>	Fungsi
<pre>x_inches='tight') neg_dict = {} for i in range(0, len(negatif['text awal'))): sentence = negatif['text awal'][i] tokenisasi = word_tokenize(sentence) for c in tokenisasi: if c not in neg_dict: neg_dict[c] = 1 else: neg_dict[c] += 1 negatif_top = pd.DataFrame((neg_dict.items())) negatif_top = negatif_top.rename(columns={0:'word', 1:'counts'}) negatif_top = negatif_top.sort_values(by='counts', ascending=False) negatif_top negatif_top.head(20).sort_values(by='counts',ascending=True).plot.barh(x='word', y='counts', rot=0,figsize=(20,10),fontsize=30) plt.savefig('grafik negatif .png',bbox_inches='tight')</pre>	<p>Menghitung berapa jumlah kata yang muncul pada keseluruhan ulasan negatif</p> <p>Menampilkan Bar plot pada visualisasi negatif.</p>
WordCloud	
<i>Pseude Code</i>	Fungsi
<pre>from wordcloud import WordCloud, STOPWORDS import matplotlib.pyplot as plt import pandas as pd import os from PIL import Image import numpy as np cloud = os.path.dirname('awan.png') mask = np.array(Image.open(os.path.join(cloud,'awan.png'))) pos_word = str(pos['text awal']) neg_word = str(neg['text awal'])</pre>	<p>Meanggil <i>library</i> wordcloud, <i>library</i> matplotlib, <i>library</i> pandas, <i>library</i> image, <i>library</i> numpy.</p> <p>Memasukkan <i>image</i> awan untuk wordcloud</p> <p>Memanggil pos_word dan neg_word</p>

<i>Pseudo Code</i>	Fungsi
<pre>wordcloud_pos = WordCloud(background_color='white', mask=mask).generate(pos_word) plt.imshow(wordcloud_pos, interpolation='bilinear') plt.axis("off") plt.show() wordcloud_pos.to_file("SVM_pos.png")</pre>	Menampilkan wordcloud sentimen positif.
<pre>wordcloud_neg = WordCloud(background_color='white', mask=mask).generate(neg_word) plt.imshow(wordcloud_neg, interpolation='bilinear') plt.axis("off") plt.show() wordcloud_neg.to_file("SVM_negatif.png")</pre>	Menampilkan wordcloud negatif.

Asosiasi

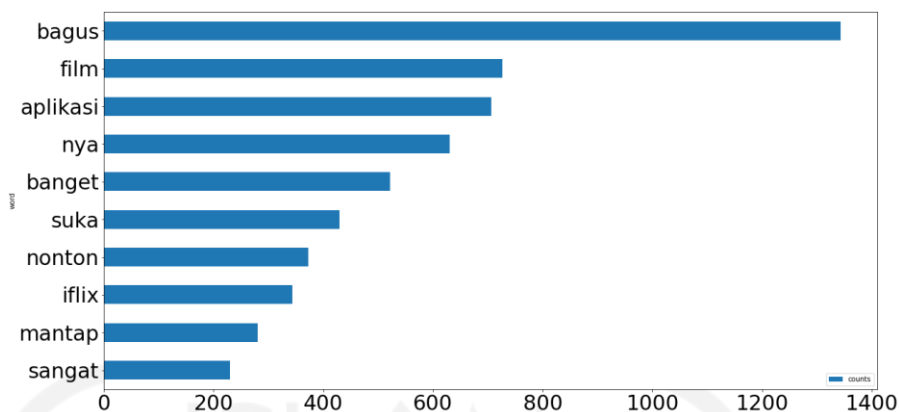
<i>Pseude Code</i>	Fungsi
<pre>import numpy as np import pandas as pd from mlxtend.frequent_patterns import apriori from mlxtend.frequent_patterns import association_rules warnings.filterwarnings('ignore') data = pd.read_csv('komentar_neg_lat.csv', encoding='unicode_escape') print(data) data = data.join(data['komentar'].str.split(' ', expand=True).add_prefix('')) items=(data['0'].unique()) items itemset = set(items) encoded_vals = [] for index, row in data.iterrows(): rowset = set(row) labels = {}</pre>	<p>Memanggil <i>library</i> numpy, <i>library</i> pandas, <i>library</i> apriori, <i>library</i> association_rules.</p> <p>Memanggil file komentar negatif.</p> <p>Melakukan <i>split</i> atau pemisahan kalimat menjadi kata – kata yang terpisah</p> <p>Membaca item atau kata yang ada didalam file.</p> <p>Mengkonversi data set.</p>

<i>Pseudo Code</i>	Fungsi
<pre> uncommons = list(items et - rowset) commons = list(itemset .intersection(rowset)) for uc in uncommons: labels[uc] = 0 for com in commons: labels[com] = 1 encoded_vals.append(la bels) encoded_vals[0] ohe_df = pd.DataFrame(enco ded_vals) freq_items = apriori(ohe_d f, min_support=0.01, use_c olnames=True) freq_items rules = association_rules(freq_items, metric="confid ence", min_threshold=0.25) rules </pre>	<p>Menghitung nilai <i>support</i> pada setiap kata.</p> <p>Melakukan proses asosiasi pada ulasan negatif yang menghasilkan nilai seperti <i>support, lift, confidence</i></p>

Berikut merupakan visualisasi dan asosiasi kata dari setiap klasifikasi kelas sentimen.

4.2.6.1 Sentimen Positif

Pada tahap visualisasi dan asosiasi ulasan positif dilakukan data berdasarkan hasil dari proses pelabelan dengan kamus *lexicon* maupun manual yang telah dilakukan pada tahap sebelumnya. Berdasarkan data tersebut didapatkan persebaran kata pada sentimen positif yang ditunjukkan pada Gambar 4.10 dibawah ini.



Gambar 4 10 Persebaran Kata Sentimen Positif

Grafik pada Gambar 4.10 menunjukkan hasil persebaran kata pada kelas sentimen positif. Dimana kata yang paling sering muncul adalah kata bagus yang muncul sebagai 1.343 kali. Visualisasi selanjutnya dilakukan dengan menggunakan *wordcloud*. Berikut merupakan visualisasi dengan menggunakan *wordcloud* yang telah dilakukan:



Gambar 4 11 *Wordcloud* Visualisasi Sentimen Positif

Berdasarkan Gambar 4.11 menunjukkan hasil dari *wordcloud* pada sentimen positif. Visualisasi dengan menggunakan *wordcloud* dilakukan dengan tujuan dapat memberikan gambaran lebih jelas mengenai kata – kata yang sering muncul pada sentimen positif. Apabila kata yang terlihat semakin besar maka menandakan kata tersebut semakin sering muncul atau semakin tinggi frekuensi

kata tersebut muncul pada sentimen positif. Pada Gambar 4.11 menunjukkan kata – kata yang memiliki frekuensi paling tinggi adalah bagus, film, menghibur, aplikasi, gambar dan lainnya. Selanjutnya dilakukan proses asosiasi kata. Dimana proses ini dilakukan untuk mengetahui kata – kata yang muncul secara bersamaan dengan kata – kata yang memiliki frekuensi muncul tertinggi.

Tabel 4.19 *Association Rule* Sentimen Positif

Antecedents	Consequents	Support	Confidence	Lift
(bagus, banyak, anime)	(film)	0.00122	1.000	5.908
(kualitas)	(bagus)	0.00244	0.571	1.474
(kualitas)	(gambar)	0.00153	0.357	55.663
(terjemahan, film)	(bagus)	0.00122	0.235	77.012
(menghibur)	(aplikasi)	0.00122	0.235	1.252
(terjemahan, film)	(episode, bagus)	0.00122	0.235	85.569
(menghibur)	(terimakasih)	0.00122	0.235	12.625
(jelas, bagus)	(suara)	0.00122	0.190	3.801
(bagus)	(gambar)	0.00244	0.006	0.983

Berdasarkan Tabel 4.19 diketahui *association rule* diatas berdasarkan kata – kata yang sering muncul pada sentimen positif. Asosiasi yang dihasilkan seperti dilihat pada kata - kata “bagus” berasosiasi dengan kata “gambar” dan suara”.

Selanjutnya terdapat kata “film” berasosiasi dengan “bagus”, “anime”, “banyak” serta “film”. Selain itu kata “film” juga berasosiasi dengan “terjemahan” “bagus” dan “episode”.

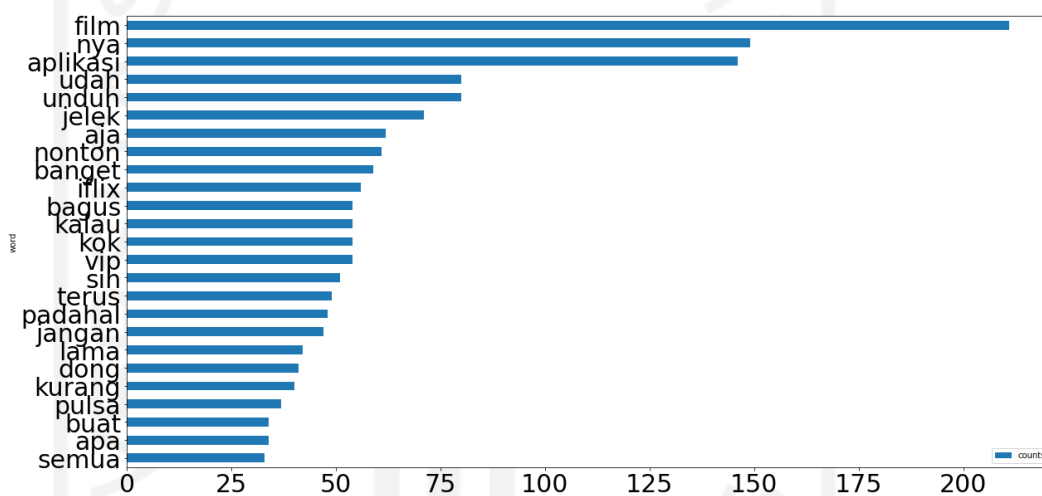
Pada kata “aplikasi” terdapat beberapa kata yang berasosiasi dengan kata itu seperti “menghibur” dan “terimakasih”. Pada kata “gambar” berasosiasi dengan “bagus” dan “kualitas”. Sedangkan kata “bagus” berasosiasi dengan “suara” dan “gambar”.

Pada Tabel 4.19 diketahui bahwa masing – masing kata yang berasosiasi memiliki nilai *support*, *confidence* serta *lift*. Dimana nilai *support* menggambarkan nilai besarnya kombinasi item yang ada di dalam data serta besarnya kemungkinan kata tersebut muncul di keseluruhan kata. Sedangkan untuk nilai *confidence* menggambarkan kuatnya hubungan antara item dalam

aturan asosiasi. Untuk nilai *lift* sendiri apabila nilainya >1 maka data tersebut bersifat kuat atau valid sebagai acuan dalam prediksi.

4.2.6.2 Sentimen Negatif

Proses visualisasi dan asosiasi tidak dilakukan pada sentimen positif saja namun dilakukan pada sentimen negatif juga. Proses visualisasi pada sentimen negatif dilakukan dengan menggunakan data dari proses sebelumnya. Berdasarkan data didapatkan persebaran kata pada sentimen negatif seperti dibawah ini:



Gambar 4 12 Persebaran Kata Sentimen Negatif

Berdasarkan grafik yang ditunjukkan pada Gambar 4.12 menunjukkan kata yang paling sering muncul pada sentimen negatif adalah kata film dengan jumlah kemunculan sebanyak 211 kali dan kata aplikasi dengan frekuensi kemunculan tertinggi kedua sebesar 146. Untuk memberikan gambaran yang lebih jelas maka dilakukan visualisasi dengan menggunakan *wordcloud* sebagai berikut ini:

Antecedents	Consequents	Support	Confidence	Lift
(muter)	(aplikasi)	0.0062	0.364	1.93
(bayar', 'iklan)	(vip', 'banyak)	0.0047	0.273	14.61
(lemot)	(nonton)	0.0047	0.250	2.87
(iklan)	(lama)	0.0078	0.217	3.68
(kurang)	(lengkap', 'film)	0.0124	0.216	9.27
(lama)	(unduh)	0.0109	0.184	1.74
(iklan)	(banyak)	0.0062	0.174	3.49
(pulsa)	(kesedot)	0.0062	0.148	2.03
(film)	(nya', 'kurang)	0.0171	0.069	3.18
(perbaharui)	(kurang)	0.0062	0.364	6.32
(susah)	(login)	0.0047	0.214	9.19

Berdasarkan data yang ditunjukkan pada Tabel 4.20 diatas, menunjukkan hasil asosiasi antara kata pada ulasan negatif yang sering muncul. Berikut merupakan asosiasi dari kata – kata yang sering muncul pada ulasan negatif.

Pada kata “aplikasi” berasosiasi dengan beberapa kata seperti “sangat” “jelek” dan “nyesel” “unduh” “muter”. Pada kata “film” berasosiasi dengan beberapa kata yaitu “susah” “cari”, “perbaharui” dan “kurang” “lengkap”.

Selanjutnya kata “iklan” berasosiasi dengan kata kata “vip” “lama”. Selain itu kata “iklan” juga berasosiasi dengan kata “banyak” “bayar”.

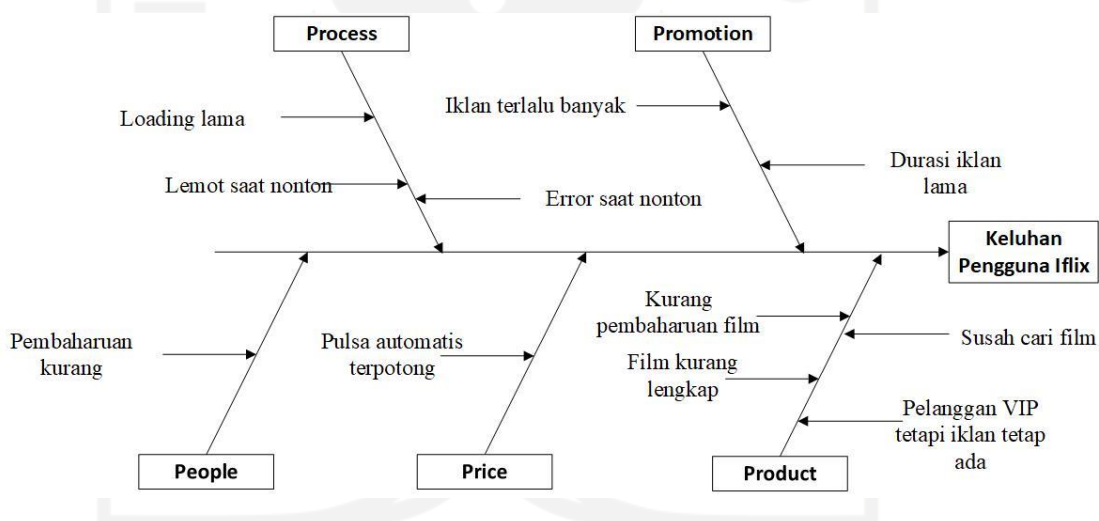
Lalu pada kata “nonton” berasosiasi dengan beberapa kata seperti “eror” dan “lemot”. Selanjutnya pada kata “pulsa” yang berasosiasi dengan “otomatis” dan “tersedot”.

Pada proses *association rule* menghasilkan beberapa *output* yaitu *support*, *confidence* dan *lift*. *Support* merupakan persentase kombinasi dari item yang ada dalam data yang digunakan. Lalu *Confidence* atau nilai kepastian adalah kuatnya hubungan antara item dalam aturan asosiasi. Sedangkan *lift ratio* adalah suatu ukuran untuk mengetahui kekuatan aturan asosiasi (*association rule*) yang telah terbentuk. Nilai *lift ratio* biasanya digunakan sebagai penentu apakah aturan asosiasi valid atau tidak valid. Apabila nilai *lift ratio* >1, maka

menunjukkan bahwa *rules* bersifat kuat atau valid untuk digunakan sebagai acuan dalam memprediksi.

4.2.7 Interpretasi Hasil

Pada tahap ini dilakukan proses analisis dengan menggunakan diagram fishbone. Berdasarkan data dari hasil ulasan negatif yang diperoleh suatu informasi terkait permasalahan atau keluhan terkait aplikasi Iflix yang diberikan oleh pengguna aplikasi iflix. Pada penelitian ini diagram fishbone akan digunakan dalam menunjang proses analisis. Diagram fishbone tersebut sebagai berikut:



Gambar 4 14 Diagram Fishbone Ulasan Negatif

Pada Gambar 4.14 menunjukkan hasil dari *diagram fishbone* berdasarkan hasil visualisasi ulasan negatif yang diberikan pengguna untuk aplikasi Iflix. Berdasarkan diagram diatas, terlihat beberapa faktor utama yaitu *process*, *promotion*, *people*, *price* dan *product*.

BAB V

ANALISIS DAN PEMBAHASAN

5.1 Analisis Deskriptif Data Awal

Analisis deskriptif adalah cara mendeskripsikan atau menggambarkan data yang telah dikumpulkan sebagaimana adanya tanpa bermaksud membuat kesimpulan berlaku untuk umum atau generalisasi (Purwoto, 2007). Analisis deskriptif dilakukan untuk mengetahui gambaran umum mengenai data ulasan yang diberikan oleh pengguna Aplikasi Iflix.

Berdasarkan data yang telah diambil pada tahap *scraping* data untuk pada 1 Januari 2021 sampai 17 Maret 2021, dimana ulasan yang didapatkan sebanyak 4.501 ulasan. Jumlah ulasan pada awal bulan Januari yang tinggi dapat diakibatkan peningkatan jumlah ulasan pada bulan Januari dapat ditunjang dengan pengeluaran beberapa karya populer garapan tokoh industri film Indonesia seperti Imperfect Series dan My Lecture My Husband. Dimana kedua series tersebut merupakan karya tokoh - tokoh industri yang sangat ditunggu oleh para pecinta perfilman Indonesia. Sebelumnya Iflix telah melakukan kerjasama dengan pihak MD Pictures untuk menampilkan karya – karya atau konten unggulan milik Indonesia.

Pada Gambar 4.7 menampilkan jumlah umpan balik dari dari pengguna kepada aplikasi serta pada Gambar 4.8 memperlihatkan perbandingan jumlah rating yang diuraikan setiap bulannya. Pada bulan Januari memiliki tingkat jumlah umpan balik terbanyak dibandingkan bulan lainnya hal ini berbanding lurus dengan jumlah ulasan seperti yang ditunjukkan pada Gambar 4.6 diatas. Gambar 4.8 menunjukkan bahwa umpan balik kategori sangat suka memiliki nilai tertinggi pada bulan Januari. Hal tersebut bisa terjadi dikarenakan pada

awal bulan tersebut terdapat beberapa film populer yang dibuat menjadi series dan menarik minat para pecinta perfilman Indonesia. Sedangkan untuk bulan selanjutnya terjadi penurunan umpan balik dalam kategori sangat suka. Hal ini terjadi di karena pada bulan tersebut tidak banyak terdapat konten atau film lokal maupun luar ditampilkan di Aplikasi Iflix sehingga pengguna memiliki persepsi bahwa Aplikasi Iflix tidak memiliki koleksi film maupun series yang lengkap. Pada umpan balik kategori sangat tidak suka posisi tertinggi terjadi pada bulan Januari dan posisi kedua terjadi pada bulan Maret. Hal tersebut dapat terjadi dikarenakan adanya pembaharuan aplikasi Iflix. Umpan balik kategori tidak suka tersebut dapat menggambarkan bahwa pengguna tidak cukup puas dengan perbaikan yang telah dilakukan.

Pada rentang waktu 1 Januari 2021 hingga 17 Maret 2021 terjadi dua kali pembaharuan aplikasi yaitu tanggal 1 Januari dan 14 Maret 2021. Dimana rentang data yang didapatkan pada pembaharuan kedua pada tanggal 14 Maret 2021 tidak cukup banyak, sehingga diharapkan pada penelitian selanjutnya dapat menggunakan data dengan rentang yang lebih banyak sehingga menambah keakuratan dari persepsi pengguna dan dapat membandingkan persepsi pengguna terhadap pembaharuan pertama dan kedua yang dilakukan oleh manajemen Iflix.

5.2 Hasil Klasifikasi SVM

Proses klasifikasi pada penelitian ini menggunakan klasifikasi *Support Vector Machine*. Proses ini diawali dengan menentukan skenario serta melakukan pembagian data uji dan data latih seperti yang ditunjukkan diatas. Berikut merupakan hasil perbandingan performasi klasifikasi yang telah dilakukan.

Tabel 5 1 Perbandingan Performasi Klasifikasi

	Linier	Kernel Linier	Polynomial	Sigmoid	RBF
SK(60:40)	91.83%	91.83%	83.02%	83.02%	83.02%
SK(80:20)	92.73%	91.84%	83.04%	83.04%	83.04%
SK(70:30)	93.45%	92.34%	82.55%	82.55%	82.55%
SK(90:10)	92.35%	92.09%	82.14%	82.14%	82.14%

Berdasarkan hasil klasifikasi dengan metode SVM didapatkan hasil nilai akurasi tertinggi pada skenario data latih 70% dan data uji 30% sebesar 93,45% dengan fungsi klasifikasi linier. Ketepatan suatu klasifikasi dapat dilihat melalui nilai akurasi klasifikasi. Dimana nilai akurasi menunjukkan performansi model klasifikasi, dimana semakin tinggi nilai akurasi maka semakin baik performansi dari model klasifikasi (Primanda, 2021). Hal tersebut menunjukkan bahwa model dan fungsi klasifikasi tersebut sesuai dengan klasifikasi data ulasan yang digunakan dalam penelitian ini.

Pada tahap sebelumnya telah diketahui bagaimana hasil klasifikasi dari setiap algoritma SVM. Setiap algoritma klasifikasi terdapat kesalahan dalam melakukan klasifikasi maupun prediksi. Kesalahan prediksi tersebut dapat dilihat melalui hasil tabel *confusion matrix*. Dimana pada Tabel 4.17 menunjukkan hasil dari *confusion matrix*. *Confusion matrix* berfungsi untuk melakukan evaluasi performansi model yang dibentuk oleh algoritma klasifikasi yaitu SVM. Hasil yang ditunjukkan oleh Tabel 4.17 untuk algoritma klasifikasi SVM linier skenario data latih 70% dan data uji 30%.

Dimana kemampuan prediksi kelas positif yang tepat benar adalah 145 ulasan dan kemampuan prediksi kelas negatif yang tepat benar adalah 653 ulasan. Sedangkan tingkat kesalahan yang dapat terjadi dalam melakukan prediksi kelas positif (*false negative*) sebesar 60 ulasan dan tingkat kesalahan yang dapat terjadi dalam melakukan prediksi kelas negatif (*false positive*) sebesar 17 ulasan. *True Positive* dapat dijelaskan bahwa prediksi positif pada model klasifikasi yang dilakukan pada penelitian ini dan memiliki aktual positif atau memang benar positif. Sedangkan *True Negative* dapat dijelaskan bahwa prediksi negatif pada model klasifikasi yang dilakukan pada penelitian ini dan memiliki aktual negatif atau memang benar negatif.

5.3 Visualisasi

Proses visualisasi dilakukan dengan menggunakan diagram batang dan *wordcloud* untuk memberikan kemudahan dalam melakukan analisis. Proses

visualisasi terbagi menjadi visualisasi pada sentimen positif dan visualisasi pada sentimen negatif.

5.3.1 Sentimen Positif

Berdasarkan Gambar 4.10 menunjukkan visualisasi dari hasil persebaran pada sentimen positif yang paling sering muncul adalah kata “bagus”, “film” dan “aplikasi”. Frekuensi kemunculan kata – kata diatas, terjadi dikarenakan pengguna sering menggunakan kata – kata tersebut dalam menggambarkan persepsi atau pandangan terhadap aplikasi Iflix. Kata

Selanjutnya pada Gambar 4.11 menunjukkan hasil visualisasi dengan menggunakan *wordcloud* dimana apabila kata yang muncul semakin besar maka semakin tinggi frekuensi kata tersebut digunakan. Terdapat kata – kata yang sering muncul berdasarkan hasil *wordcloud* yaitu “film”, “bagus”, “aplikasi”, “gambar”, “menghibur” dan “nonton”.

Pada Tabel 4.19 menunjukkan hasil *association rule* pada sentimen positif yang berkaitan dengan beberapa kata yang memiliki frekuensi kemunculan yang tinggi seperti kata “bagus” berasosiasi dengan kata “gambar” dan “suara”. Hal tersebut menggambarkan bahwa pengguna aplikasi ini memiliki gambar dan suara yang bagus. Hal tersebut dapat menjadi salah satu alasan mengapa pengguna menyukai aplikasi ini.

Lalu “menghibur” berasosiasi dengan “aplikasi” dan “terimakasih”. Hal tersebut memberikan gambaran bahwa aplikasi ini cukup memberikan hiburan kepada pengguna.

Pada kata “film” dengan kata “bagus”, “banyak”, “anime” serta kata “episode” dengan “terjemahan”, “bagus”. Hal tersebut dapat di gambarkan bahwa film yang tersedia episode atau pilihan genre anime yang cukup banyak dan aplikasi ini di lengkapi terjemahan yang bagus pada film yang ditampilkan oleh aplikasi ini. Hal tersebut dapat menambah point positif dari pengguna terhadap aplikasi Iflix.

Selanjutnya kata “gambar” berasosiasi dengan “kualitas” dan “bagus”. Pada kata gambar yang berasosiasi ini dapat memberikan gambaran bahwa aplikasi ini

menampilkan film – film dengan gambar yang bagus dan berkualitas. Hal tersebut dapat mempermudah pengguna dalam menikmati film yang di sediakan dan menambahkan rasa suka pengguna terhadap aplikasi.

Berdasarkan hasil asosiasi serta kata – kata yang muncul pada wordcloud dapat menggambarkan persepsi pengguna terhadap aplikasi Iflix yaitu suatu aplikasi yang menghibur dengan kualitas gambar yang bagus dan suara yang bagus selain itu aplikasi ini dilengkapi dengan terjemahan yang bagus dan memiliki variasi anime yang cukup bagus.

5.3.2 Sentimen Negatif

Pada bagian sentimen negatif juga dilakukan proses visualisasi dengan menggunakan diagram batang dan *wordcloud*. Pada Gambar 4.12 menunjukkan hasil persebaran kata yang terjadi pada kelas sentimen negatif.

Pada persebaran diatas terdapat beberapa kata yang muncul seperti pada persebaran sentimen negatif dengan frekuensi kemunculan teratas yaitu “film”, “aplikasi”, “jelek” dan “nonton”.

Pada proses visualisasi terdapat beberapa kata yang muncul dimana tingkat frekuensi kemunculan dapat ditunjukkan dengan semakin besar ukuran tulisan kata yang muncul. Berdasarkan gambar yang ditunjukkan pada Gambar 4.13 diketahui beberapa kata yang memiliki tingkat frekuensi tinggi yaitu “film”, “vip”, “aplikasi”, “iflix”, “jelek”, “lama”, “nonton”, dan “pulsa”.

Berdasarkan Tabel 4.20 diatas menunjukkan hasil dari proses *association rule*. Dimana dapat diketahui kata – kata apa saja yang memiliki kemungkinan untuk muncul secara bersamaan. Pada kata – kata yang berasosiasi dengan kata “aplikasi” yaitu kata “sangat” “jelek” dan “nyesel” “unduh” “muter”. Hal tersebut menggambarkan bahwa pengguna merasa menyesal setelah mengunduh aplikasi ini dan merasa aplikasi ini jelek serta sering terjadi *loading* yang lama ketika menggunakan aplikasi ini serta pengguna merasa kurang dilakukan pembaharuan aplikasi.

Pada kata yang berasosiasi dengan “film” yaitu “susah” “cari”, “perbaharui” dan “kurang” “lengkap” menggambarkan bahwa aplikasi ini

memiliki film yang variasinya kurang lengkap, pembaharuan yang kurang dan pengguna mengalami kesusahan ketika mencari film yang diinginkan.

Selanjutnya kata- kata yang berasosiasi dengan kata “iklan” yaitu “vip” “lama”. Selain itu kata “iklan” juga berasosiasi dengan kata “banyak” “bayar” menggambarkan bahwa aplikasi ini meski telah membayar untuk berlangganan vip namun tetap muncul iklan yang berdurasi lama dan mengganggu pengguna ketika menggunakan aplikasi ini.

Lalu pada kata - kata yang berasosiasi dengan kata “nonton” yaitu “eror” dan “lemot” menggambarkan bahwa ketika pengguna menggunakan aplikasi sering terjadi masalah seperti lemot dan *error*. Selain itu sering terjadi kesalahan sistem dimana pulsa yang sering tersedot secara otomatis dan hal tersebut merugikan pengguna yang ditunjukkan pada kata “pulsa” yang berasosiasi dengan “otomatis” dan “tersedot”.

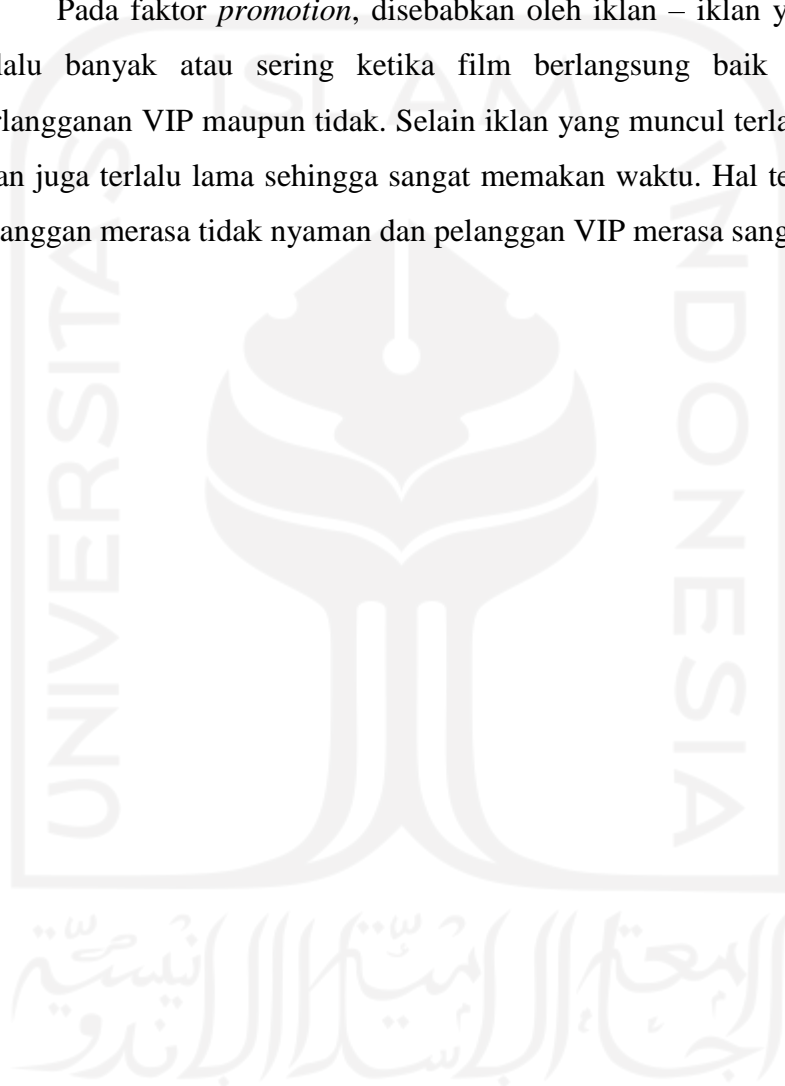
5.4 Fishbone Diagram

Berdasarkan diagram fishbone yang ditunjukkan pada Gambar 4.14 dapat dilakukan analisis serta informasi penyebab pengguna memberikan ulasan negatif oleh pengguna terkait aplikasi Iflix. Beberapa faktor utama yang menyebabkan ulasan negatif terhadap aplikasi Iflix muncul adalah *people*, *price*, *product promotion*, dan *process*. Pada faktor *process*, disebabkan oleh beberapa hal seperti loading yang terlalu lama ketika mengaplikasikan Iflix. Selain itu penyebab lainnya adalah sering terjadi eror dan proses yang lambat ketika pengguna menggunakan aplikasi ini untuk menonton. Hal tersebut mengganggu pengguna aplikasi. Pada faktor *people* terlihat bahwa pengguna merasa bahwa pembaharuan yang dilakukan oleh aplikasi terasa kurang. Dimana aplikasi tentunya diperbaharui oleh tim pengembang dari manajemen aplikasi Iflix.

Pada faktor *product* terdapat beberapa penyebab yaitu pilihan film yang kurang lengkap. Penyebab lainnya adalah kurang pembaharuan film sehingga film tidak *terupdate* dengan baik. Selain itu pengguna sering merasa kesusahan ketika mencari film yang diinginkan. Serta pelanggan sering terganggu dengan iklan yang sering muncul meski telah berlangganan VIP. Hal tersebut

mengakibatkan pengguna tidak dapat menikmati atau menonton dengan baik. Pada faktor *price*, disebabkan oleh proses pembayaran yang tidak berjalan semestinya dimana sering terjadi kesalahan berupa sistem pemotongan pulsa otomatis yang dilakukan oleh pihak Iflix yang merugikan pengguna. Dimana belum ada persetujuan pengguna ingin lanjut berlangganan atau tidak, namun sudah langsung terpotong.

Pada faktor *promotion*, disebabkan oleh iklan – iklan yang ditampilkan terlalu banyak atau sering ketika film berlangsung baik pada pengguna berlangganan VIP maupun tidak. Selain iklan yang muncul terlalu sering, durasi iklan juga terlalu lama sehingga sangat memakan waktu. Hal tersebut membuat pelanggan merasa tidak nyaman dan pelanggan VIP merasa sangat dirugikan.



BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan hasil pengolahan data dan hasil analisis, maka dapat ditarik beberapa kesimpulan yakni sebagai berikut ini:

1. Berdasarkan hasil pengambilan data pada 1 Januari 2021 hingga 17 Maret 2021 diketahui jumlah umpan balik pada kategori sangat suka di bulan Januari memiliki nilai terbanyak yaitu sebesar 1.134 ulasan. Hal tersebut terjadi dikarenakan proses pembaharuan serta terdapat beberapa konten lokal populer yang baru saja keluar pada bulan tersebut. Tingkat umpan balik yang tinggi menandakan kepuasan pengguna terhadap aplikasi. Sedangkan bulan selanjutnya memiliki penurunan umpan balik dalam kategori sangat tidak suka, terjadi dikarenakan tidak banyak terdapat konten atau film lokal maupun luar ditampilkan di Aplikasi Iflix sehingga pengguna memiliki persepsi bahwa Aplikasi Iflix tidak memiliki koleksi film maupun series yang lengkap. Pada umpan balik dalam kategori tidak suka posisi tertinggi terjadi pada bulan Januari dan posisi kedua pada bulan Maret. Tingkat umpan balik yang tinggi pada kedua bulan tersebut dapat menggambarkan bahwa pengguna tidak cukup puas dengan perbaikan yang telah dilakukan.
2. Berdasarkan proses klasifikasi dengan menggunakan metode SVM yang dilakukan menunjukkan nilai akurasi tertinggi dimiliki oleh skenario data latih 70 % dan data uji 30 % sebesar 93,45 %. Sedangkan hasil *confusion matrix* adalah kemampuan prediksi kelas positif yang tepat benar adalah 145 ulasan dan kemampuan prediksi kelas negatif yang tepat benar adalah 653 ulasan. Sedangkan tingkat kesalahan yang dapat terjadi dalam melakukan prediksi kelas positif (*false negative*) sebesar 60 ulasan dan tingkat kesalahan

yang dapat terjadi dalam melakukan prediksi kelas negatif (*false positive*) sebesar 17 ulasan.

3. Informasi yang didapatkan dari hasil proses klasifikasi pada kelas sentimen positif yaitu terdapat beberapa kata yang sering muncul seperti “film”, “bagus”, “aplikasi”, “gambar”, “menghibur” dan “nonton”. Hal tersebut menggambarkan iflix merupakan suatu aplikasi yang menghibur dengan kualitas gambar yang bagus dan suara yang bagus selain itu aplikasi ini dilengkapi dengan terjemahan yang bagus dan memiliki variasi anime yang cukup bagus. Pada kelas sentimen negatif terdapat beberapa kata yang sering muncul seperti “film”, “vip”, “aplikasi”, “iflix”, “jelek”, “lama”, “nonton”, dan “pulsa”. Berdasarkan hasil asosiasi kata dengan kata yang sering keluar menggambarkan bahwa pengguna merasa kurang puas dengan aplikasi dikarenakan film yang kurang lengkap, kurangnya pembaharuan pada film, pengguna mengalami kesulitan dalam mencari film serta sering terjadi gangguan ketika menonton dan proses *loading* yang lambat. Selain itu sering terjadi kesalahan dalam pemotongan pulsa berlangganan yang secara otomatis dan iklan yang berdurasi terlalu lama dan banyak.

6.2 Saran

Berdasarkan penelitian yang telah dilakukan. Berikut merupakan saran yang dapat diberikan pada peneliti selanjutnya:

1. Pada penelitian selanjutnya diharapkan dapat melakukan pengambilan data dengan rentang yang lebih lama dengan tujuan menunjang keakuratan dari penelitian itu sendiri.
2. Pada proses pengolahan data pada bagian pelabelan, pada penelitian ini dilakukan dengan menggunakan kamus *lexicon*. Sedangkan pada kamus tersebut tidak semua memiliki kata beserta bobot yang sesuai dengan data yang dimiliki sehingga perlu dilakukan perbaikan dengan menggunakan tambahan *tools* lainnya.

3. Pada penelitian selanjutnya, diharapkan melakukan proses klasifikasi dengan menambahkan berbagai macam skenario dengan tujuan dapat menguji model klasifikasi yang lebih bervariasi.



DAFTAR PUSTAKA

- Abdulrahman, M. A. (2013). The Impact of Electronic Word of Mouth on Consumers Purchasing Decisions. *International Journal of Computer Applications. Volume 82*.
- Adiwijaya, I. (2006). Text mining dan knowledge discovery. *Komunitas Data mining Indonesia & Soft-omputing Indonesia*.
- Aktual, K. K. (n.d.). Retrieved from <https://kerincitime.co.id/efek-pandemi-covid-19-terhadap-perfilman-indonesia.html>
- Alamarani, Y. E. (2018). Sentiment Analysis Using Hybrid Method Of Support Vector Machine And Decision Tree. *Journal of Theoretical & Applied Information Technology*, 1886–1895.
- Anjasmoros, M. T. (2020). ANALISIS SENTIMEN APLIKASI GO-JEK MENGGUNAKAN METODE SVM DAN NBC (STUDI KASUS: KOMENTAR PADA PLAY STORE). *Conference on Innovation and Application of Science and Technology (CIASTECH)* .
- Asosiasi Penyelenggara Jasa Internet Indonesia. (2020, November).
- Atik, M. A. (2016). The Effect of Online Consumer Review toward Purchase Intention: A Study in Premium Cosmetic in Indonesia. *Journal International Conference on Ethics of Business, Economics, and Social Science*.
- B, L. (2010.). *Handbook of natural language processing*. Vol. 2. CRC Press.
- Badr, E. M. (2019). Social Media Sentiment Analysis using Machine Learning and Optimization Techniques. *International Journal of Computer Applications*.
- Barker, T. (2020). *Peta Kompetisi Penyedia Layanan Streaming di Asia Tenggara*. Retrieved from thecoversation.com.
- Besterfield, D. (2009). *Quality Control 8th Edition*. New Jersey: Pearson Prentice.
- Bruce R.Maxim, R. (2014). *Software Engineering: A Practitioner's Approach (8th edition)*.
- Buntoro, G. A. (2014). Sentiment Analysis Twitter dengan Kombinasi Lexicon Based dan Double Propagation. *CITEE 2014*.
- Buyens, J. (2001). *Web Database Development*. . Jakarta: Elex Media Komputindo.
- Cam.Davidson. (2012). *stats.stackexchange*.

- Chory, R. N. (2018). Sentiment analysis on user satisfaction level of mobile data services using Support Vector Machine (SVM) algorithm. *IEEE International Conference on Internet of Things and Intelligence*.
- Desai, M. &. (2016). Techniques for Sentiment Analysis of Twitter Data: A Comprehensive Survey. *Internasional Conference on Computing, Communication and Automation*
- Dragut, E. F. (2009). *Stop Word And Related Problem in Web Interface Integration*. VLDB Endowment.
- Drajana, I. C. (2017). Metode support vector machine dan forward selection prediksi pembayaran pembelian bahan baku kopra. *ILKOM Jurnal Ilmiah*, 116-123.
- Fahrizal, F. I. (2020). Implementasi Machine Learning Pada Sistem Pets Identification Menggunakan Python Berbasis Ubuntu. *Journal of Information System, Informatics and Computing 4.1*, 86-91.
- Feldman, R. &. (2007). *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. New York: Cambridge University Press.
- Fikriya, Z. A., Soetrisno, & Irawan, M. I. (2017). Implementasi Extreme Learning Machine untuk Pengenalan Objek Citra Digital. *Jurnal Sains dan Seni ITS*, vol. 6, pp. A18–A23.
- finance.detik.com. (2021). Retrieved from <https://finance.detik.com/berita-ekonomi-bisnis/d-5477536/sederet-sektor-terdampak-paling-parah-selama-setahun-corona/2>
- finance.detik.com. (2021, Maret). *Sederet Sektor Terdampak Paling Parah Selama Setahun Corona*. Retrieved from <https://finance.detik.com/berita-ekonomi-bisnis/d-5477536/sederet-sektor-terdampak-paling-parah-selama-setahun-corona/2>
- Haizer, J. &. (2014). *Manajemen Operasi*. Jakarta: Salemba.
- Harahap, E. H. (2018). Implementasi Algoritma Support Vector Machine (SVM) Untuk Penentuan Seleksi Atlet Pencak Silat. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 3843-3848.
- Herlinawati, N. Y. (2020). Analisis Sentimen Zoom Cloud Meetings di Play Store Menggunakan Naïve Bayes dan Support Vector Machine. *Journal of Computer Engineering, System and Science*, 293-298.
- Iswara, B. N. (2019). Nurhayati, Busman, and Rayi Pradono Iswara. "Pengembangan Algoritma Unsupervised Learning Technique Pada Big Data Analysis di Media

- Sosial sebagai media promosi Online Bagi Masyarakat. *Jurnal Teknik Inform 12.1*, 79-96.
- Jha, V. M. (2016). HSRA: Hindi stopword removal algorithm. *International Conference on Microelectronics, Computing and Communications (MicroCom)*.
- Josi, A., & Abdillah, S. &. (2014). Penerapan teknik web scraping pada mesin pencari artikel ilmiah. *arXiv preprint arXiv*, 1410.5777.
- Kamber, J. H. (2006). *Data Mining : Concepts and Techniques Second Edition*. USA: Elsevier.
- Katadata. (2019). Retrieved from <https://databoks.katadata.co.id/datapublish/2020/01/20/berapa-jumlah-pengguna-smartphone-dunia>
- kerincitime.co.id. (2021, Maret). *efek-pandemi-covid-19-terhadap-perfilman-indonesia*.
- Khanna, C. (2021). Text pre-processing: Stop words removal using different libraries.
- Kowalczyk, A. (2017). *Support vector machines succinctly*. Syncfusion Inc.
- Kurniawan, H. I. (2011). Aplikasi Penjawab Pesan Singkat Automatis dengan Bahasa Python. *Jurusan Teknik Elektro Fakultas Teknik Undip*.
- Kusuma, P. D. (2020). *Machine Learning Teori, Program dan Studi Kasus* . Deepublish.
- Li, N., & Wu, D. D. (2010). Using text mining and sentiment analysis for online forums hotspot detection and forecast. 354–368.
- Linggi, M. P. (2020). *Analisis Kepuasan Pengunjung Website Iflix Dengan Metode Customer Satisfaction Index Dan Importance Performance Analysis*. eProceedings of Management.
- Liu, B. (2010). *Handbook of Natural Lenguage Processing 2nd*. Boca Raton: CRC Press.
- Lutz, M. (2010). *Programming Python: powerful object-oriented programming*. O'Reilly Media, Inc.
- Mangat, H. K. (2017). A survey of sentiment analysis techniques. *International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)* (pp. (pp. 921-925)). IEEE.
- Manning, C. D. (2009). *An Introduction to Information Retrieval*. England: Cambridge University Press.

- Mardi, Y. (2017). Data Mining: Klasifikasi Menggunakan Algoritma C4. 5. *Edik Informatika* , 213-219.
- Margiastuti, N. &. (2010). KLASTERISASI DATA BERDIMENSI TINGGI MENGGUNAKAN KOMBINASI PRINCIPAL COMPONENT ANALYSIS (PCA) DAN SELF ORGANIZING MAP (SOM).
- mdpictures.com. (2020).
- Mujahid, A. J. (2018). Pengaruh Penggunaan Smartphone terhadap Perkembangan Personal Sosial Anak Usia Pra-Sekolah. *Jurnal Keperawatan Silampari 2.1*, 330-342.
- Nablani, A. (2019, Desember 28). *datacamp.com*.
- Nugroho, K. S. (2019). Dasar Text Preprocessing dengan Python.
- plus.kapanlagi.com. (2020, Agustus). Retrieved from <https://plus.kapanlagi.com/lakukan-kerja-sama-wetv-dan-iflix-siap-tayangkan-120-lebih-film-terbaik-indonesia-dari-md-pictures-809d90.html>
- Pramudita, E. R. (2020). Mengenal Machine Learning Dengan Teknik Supervised Dan Unsupervised Learning Menggunakan Python . *Bina Insani Ict Journal* , 156-165.
- Pratiwi, K. S. (2020). *Medium.com* .
- Primanda, D. L. (2021, July). *Classification of Iris with Support Vector Machine*. Retrieved from *Medium.com* .
- Putra, J. W. (2020). *Pengenalan Konsep Pembelajaran Mesin dan Deep Learning*. Tokyo: Wiragotama.
- Putra, J. W. (2020). Support Vector Machine. In *Pengenalan Konsep Pembelajaran Mesin dan Deep Learning*.
- Purwoto, A. (2007). Panduan Laboratorium Statistik Inferensial. Jakarta: Gramedia.
- republika.co.id. (2021, Maret). *Indutri Indonesia mulai beradaptasi ditengah pandemi*.
- Ritonga, A. S. (2018). Penerapan Metode Support Vector Machine (SVM) Dalam Klasifikasi Kualitas Pengelasan Smaw (Shield Metal Arc Welding) . *Eduitic-Scientific Journal of Informatics Education*, 5(1).
- Romadoni, F. U. (2020). Text Mining Untuk Analisis Sentimen Pelanggan Terhadap Layanan Uang Elektronik Menggunakan Algoritma Support Vector Machine. *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, 247-253.

- Samuel, Y., & Antonius, D. R. (2015). Implementasi metode K-Nearest Neighbor dengan decision rule untuk klasifikasi subtopik berita. *Jurnal Informatika*.
- Sembiring, K. (2007). *Penerapan Teknik Support Vector Machine untuk Pendeteksi Instruksi pada Jaringan*. Skripsi: Program Studi Teknik Informatika Institut Teknologi Bandung.
- Santosa, B. (2007). *Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis*. Yogyakarta: Graha Ilmu.
- Sitiorus, I. (2020, Agustus 27). *Support Vector Machine (SVM) and Kernels Trick*. teknologi.bisnis.com. (2020, Oktober). *Bisnis.com*.
- Souza, C. (2010). Kernel function for machine learning application.
- Suryanto. (2020). *Layanan Streaming Tantangan dan Peluang Perfilman Indonesia*. Retrieved from Antaranews.
- Tineges, R. A. (2020). Analisis Sentimen Terhadap Layanan Indihome Berdasarkan Twitter Dengan Metode Klasifikasi Support Vector Machine (SVM). *Jurnal Media Informatika Budidarma* 4.3, 650-658.
- Turland, M. (2010). php| architect's Guide to Web Scraping with PHP. *Introduction-Web Scraping*, str, 2.
- Ulwan, M. N. (2016). *Pattern Recognition pada Unstructured Data Teks Menggunakan Support Vector Machine dan Association*. Skripsi: Program Studi Statistika Universitas Islam Indonesia.
- Uysal, A. K. (2014). The impact of preprocessing on text classification. *Information Processing & Management*. *Information Processing & Management*, 50(1), 104–112.
- Utama, H. S. (2019). Analisis Sentimen Sistem Ganjil Genap di Tol Bekasi Menggunakan Algoritma Support Vector Machine. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 243-250.
- Utomo, M. S. (2012). Implementasi PHP Sebagai Penghasil Konten Otomatis Pada Halaman Situs. *Jurnal Teknologi Informasi Dinamik Vol 17 No 2*, 147-153.
- Utomo, M. S. (2013). Web Scraping pada Situs Wikipedia menggunakan Metode Ekspresi Regular. *Dinamik* 18.2.
- Verma, B. &. (2018). Sentiment analysis using lexicon and machine learning-based approaches: A survey. *In Proceedings of international conference on recent advancement on computer and communication*, (pp. (pp. 441-447)). Springer, Singapore

- Wahyono, T. (2018). *Fundamental of Python for Machine Learning: Dasar-Dasar Pemrograman Python untuk Machine Learning dan Kecerdasan Buatan*. Yogyakarta: Gava Media.
- Weiss, S. M. (2010). *Text Mining: Predictive Methods for Analyzing Unstructural Information*. New York: Springer.
- Widodo, P. P. (2013). *Penerapan Data Mining Dengan Matlab*. Bandung: Rekayasa Sains.
- Wuryandani, D. (2020, Agustus). Badan Ekonomi dan Kebijakan Publik Info Singkat.
- Zhu, M. I. (2015). SUPERVISED MACHINE LEARNING APPROACHES: A SURVEY. *ICTACT Journal on Soft Computing* 5.3.



LAMPIRAN

Source Code Python

Pre Processing

#Slangword

```
dictawal = ['Good','good','bs','yg','udh','gtu','ja','syuka','tp','bgt','bnget','tdk','jd','sy','km','kl
u','flm','lg','pdhl','bnyk','km','Abis','ad','adlh','ak','alow','ambilin','ancur']
dictakhir = ['bagus','bagus','bisa','yang','udah','gitu','aja','suka','tapi','banget','banget','tidak
','jadi','saya','karena','kalau','film','lagi','padahal','banyak','kamu','Habis','ada','adalah','aku'
,'halo','ambilkan','hancur,']
data['komentar'] = data['komentar'].replace(dictawal,dictakhir, regex=True)
```

#Cleaning

```
import string, re
def cleansing (data):
    data = data.lower()
    remove = string.punctuation
    translator = str.maketrans(remove, '*len(remove))
    data = data.translate (translator)
    angber = ".join([x for x in data if not (x.isdigit()or x[0] == '-' and x[1:].isdigit())])
    return angber
```

#tokenize

```
import nltk
nltk.download('punkt')
from nltk.tokenize import word_tokenize
def tokenize (txt):
    word_token = word_tokenize (txt)
    return word_token
```

#StopWord

```
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemove
rFactory
factory = StopWordRemoverFactory()
stopwords = factory.get_stop_words()
def stopw (txt_tokenized):
    stopw = [word for word in txt_tokenized if word not in stopwords]
    return stopw
```

#data_kembali

```
def kembali(txt):
```

```
data_kembali = ' '.join([c for c in txt if c not in string.punctuation])
return data_kembali
```

```
pd.options.display.max_colwidth = None
data['data_lower'] = data['komentar'].apply(lambda x: cleansing(x))
data['data_token'] = data['data_lower'].apply(lambda x: tokenize(x))
data['data_stopword'] = data['data_token'].apply(lambda x: stopw(x))
data['data_kembali'] = data['data_stopword'].apply(lambda x: kembali(x))
data
```

Pelabelan

```
negasi = ['bukan','tidak','ga','gk']
lexicon_word = lexicon['word'].to_list()
lexicon_number_word = lexicon['number_of_word']
len(lexicon_word)
import nltk
nltk.download('punkt')
from nltk.tokenize import word_tokenize
sencol = []
senrow = np.array([])
nsen = 0
factory = StemmerFactory()
stemmer = factory.create_stemmer()
sentiment_list = []

# function to write the word's sentiment if it is founded
def found_word(ind,words,word,sen,sencol,sentiment,add):
    # if it is already included in the bag of words matrix, then just increase the value
    if word in sencol:
        sen[sencol.index(word)] += 1
    else:
        #if not, than add new word
        sencol.append(word)
        sen.append(1)
        add += 1
    #if there is a negation word before it, the sentiment would be the negation of it's senti
ment
    if (words[ind-1] in negasi):
        sentiment += -lexicon['weight'][lexicon_word.index(word)]
    else:
        sentiment += lexicon['weight'][lexicon_word.index(word)]

return sen,sencol,sentiment,add
```

```

# checking every words, if they are appear in the lexicon, and then calculate their sentiment if they do
for i in range(len(data)):
    nsen = senrow.shape[0]
    words = word_tokenize(data['data_kembali'][i])
    sentiment = 0
    add = 0
    prev = [0 for ii in range(len(words))]
    n_words = len(words)
    if len(sencol)>0:
        sen =[0 for j in range(len(sencol))]
    else:
        sen =[]

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

```

```

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

    sentiment_list.append(sentiment)
    sencol.append('sentiment')
    sentiment_array = np.array(sentiment_list).reshape(senrow.shape[0],1)
    sentiment_data = np.hstack((senrow,sentiment_array))
    df_sen = pd.DataFrame(sentiment_data,columns = sencol)

```

Klasifikasi

```

pd.set_option('max_columns', 8)
Data
x, y = Data.komentar, Data.kelas
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x,y, test_size = 0.20, random_state = 0
)
X_train, X_test, y_train, y_test
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer=TfidfVectorizer(input = 'content',lowercase=True,)
features_train_transformed = vectorizer.fit_transform(X_train)
features_test_tranformer = vectorizer.transform(X_test)

#SVM linier
SVM_Clasifier = svm.LinearSVC()
SVM_Clasifier.fit(features_train_transformed,y_train)
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
predictions_SVM = SVM_Clasifier.predict(features_test_tranformer)
print('linier')
print(classification_report(y_test ,predictions_SVM ))
print('Confusion Matrix: \n',confusion_matrix(y_test,predictions_SVM))

```

```

print()
print('Accuracy: ', accuracy_score(y_test,predictions_SVM))

#SVM non linier
#kernel linier
SVM_Classifier_kernel_linier = svm.SVC(C=1.0, kernel='linear', degree=8, gamma='auto')
SVM_Classifier_kernel_linier.fit(features_train_transformed,y_train)
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
predictions_SVM = SVM_Classifier_kernel_linier.predict(features_test_tranformer)
print('kernel linier')
print(classification_report(y_test ,predictions_SVM ))
print('Confusion Matrix: \n',confusion_matrix(y_test,predictions_SVM))
print()
print('Accuracy: ', accuracy_score(y_test,predictions_SVM))

#kernel poly
SVM_Classifier_kernel_poly = svm.SVC(C=1.0, kernel='poly', degree=8,gamma='auto')
SVM_Classifier_kernel_poly.fit(features_train_transformed,y_train)
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
predictions_SVM = SVM_Classifier_kernel_poly.predict(features_test_tranformer)
print('polynomial')
print(classification_report(y_test ,predictions_SVM ))
print('Confusion Matrix: \n',confusion_matrix(y_test,predictions_SVM))
print()
print('Accuracy: ', accuracy_score(y_test,predictions_SVM))

#kernel sigmoid
SVM_Classifier_kernel_sigmoid = svm.SVC(C=1.0, kernel='sigmoid', degree=8, gamma='auto')
SVM_Classifier_kernel_sigmoid.fit(features_train_transformed,y_train)
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
predictions_SVM = SVM_Classifier_kernel_sigmoid.predict(features_test_tranformer)
print('sigmoid')
print(classification_report(y_test ,predictions_SVM ))
print('Confusion Matrix: \n',confusion_matrix(y_test,predictions_SVM))
print()
print('Accuracy: ', accuracy_score(y_test,predictions_SVM))

#kernel rbf
SVM_Classifier_rbf = svm.SVC(C=1.0, kernel='rbf', degree=8, gamma='auto')

```

```

SVM_Classifier_rbf.fit(features_train_transformed,y_train)
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
predictions_SVM = SVM_Classifier_rbf.predict(features_test_tranformer)
print('RBF')
print(classification_report(y_test ,predictions_SVM ))
print('Confusion Matrix: \n',confusion_matrix(y_test,predictions_SVM))
print()
print('Accuracy: ', accuracy_score(y_test,predictions_SVM))

#bagian klasifikasi untuk melabelkan.

input_transformed = vectorizer.transform(Data)
SVM_linier_prediction = SVM_Classifier.predict(input_transformed)
SVM_kernel_linier_prediction = SVM_Classifier_kernel_linier.predict(input_transforme
d)
SVM_poly_prediction = SVM_Classifier_kernel_poly.predict(input_transformed)
SVM_sigmoid_prediction = SVM_Classifier_kernel_sigmoid.predict(input_transformed)
SVM_rbf_prediction = SVM_Classifier_rbf.predict(input_transformed)
SVM_linier_prediction
#jika keluar banyak row bener, harusnya keluaranya positif negatif.
result = pd.DataFrame({"text awal" : Data,"label" : SVM_linier_prediction})
result.to_csv("HasilSVM.csv", index = False)

```

Visualisasi

#Diagram Batang

```

visual = pd.DataFrame([])
visual['teks'] = df['label'].values.copy()
visual

```

#Keseluruhan

```

word_dict = {}
for i in range(0,len(visual['teks'])):
    sentence = visual['teks'][i]
    word_token = word_tokenize(str(sentence))
    for j in word_token:
        if j not in word_dict:
            word_dict[j] = 1
        else:
            word_dict[j] += 1
df_top = pd.DataFrame((word_dict.items()))
df_top = df_top.rename(columns={0:'word', 1:'counts'})
df_top = df_top.sort_values(by='counts', ascending=False)
df_top.head(10)

```



```

import matplotlib.pyplot as plt
df_top.head(10).sort_values(by='counts',ascending=True).plot.barh(x='word', y='counts'
, rot=0,figsize=(20,10),fontsize=30)
plt.savefig('grafik .png',bbox_inches='tight')
df = pd.read_csv('HasilSVM.csv')
neg = df.loc[df['label'] == 'Negatif']
pos = df.loc[df['label'] == 'Positif']
neg.to_csv('ulasan negatif.csv')
pos.to_csv('ulasan positif.csv')
Positif=pos.to_csv('ulasan positif.csv')
Negatif = pd.read_csv('ulasan negatif.csv')
positif = pos.reset_index(drop=True)
negatif = neg.reset_index(drop=True)

```

#Visualisasi Positif

```

pos_dict = { }
for i in range(0, len(positif['text awal'])):
    sentence = positif['text awal'][i]
    token = word_tokenize(str(sentence))
    for b in token:
        if b not in pos_dict:
            pos_dict[b] = 1
        else:
            pos_dict[b] += 1
positif_top = pd.DataFrame((pos_dict.items()))
positif_top = positif_top.rename(columns={0:'word', 1:'counts'})
positif_top = positif_top.sort_values(by='counts', ascending=False)
positif_top.head(10).sort_values(by='counts',ascending=True).plot.barh(x='word', y='co
unts', rot=0,figsize=(20,10),fontsize=30)
plt.savefig('grafik positif .png',bbox_inches='tight')

```

#Visualisasi Negatif

```

neg_dict = { }
for i in range(0, len(negatif['text awal'])):
    sentence = negatif['text awal'][i]
    tokenisasi = word_tokenize(sentence)
    for c in tokenisasi:
        if c not in neg_dict:
            neg_dict[c] = 1
        else:
            neg_dict[c] += 1

```

```

negatif_top = pd.DataFrame((neg_dict.items()))
negatif_top = negatif_top.rename(columns={0:'word', 1:'counts'})
negatif_top = negatif_top.sort_values(by='counts', ascending=False)
negatif_top
negatif_top.head(10).sort_values(by='counts',ascending=True).plot.barh(x='word', y='counts', rot=0,figsize=(20,10),fontsize=30)
plt.savefig('grafik negatif .png',bbox_inches='tight')

```

#Wordcloud

```

cloud = os.path.dirname('awan_png.png')
mask = np.array(Image.open(os.path.join(cloud,'awan_png.png')))
pos_word = str(pos['text awal'])
neg_word = str(neg['text awal'])
wordcloud_pos = WordCloud(background_color='white',mask=mask).generate(pos_word)
plt.imshow(wordcloud_pos, interpolation='bilinear')
plt.axis("off")
plt.show()
wordcloud_pos.to_file("SVM_pos.png")
wordcloud_neg = WordCloud(background_color='white', mask=mask).generate(neg_word)
plt.imshow(wordcloud_neg, interpolation='bilinear')
plt.axis("off")
plt.show()
wordcloud_neg.to_file("SVM_negatif.png")

```

Asosiasi

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
warnings.filterwarnings('ignore')

```

#Split Data

<https://stackoverflow.com/questions/48207115/splitting-a-column-into-multiple-columns-with-specific-name-in-pandas-dataframe>

#positif

```

data = data.join(data['komentar'].str.split(' ', expand=True).add_prefix(""))
print(data[['2']])

```

```

del data['komentar']
items=(data['0'].unique())
items

itemset = set(items)
encoded_vals = []
for index, row in data.iterrows():
    rowset = set(row)
    labels = {}
    uncommons = list(itemset - rowset)
    commons = list(itemset.intersection(rowset))
    for uc in uncommons:
        labels[uc] = 0
    for com in commons:
        labels[com] = 1
    encoded_vals.append(labels)
encoded_vals[0]
ohe_df = pd.DataFrame(encoded_vals)
freq_items = apriori(ohe_df, min_support=0.01, use_colnames=True)
freq_items
rules = association_rules(freq_items, metric="confidence", min_threshold=0.25)
rules

#negatif data = data.join(data['komentar'].str.split(' ', expand=True).add_prefix(""))
print(data[['2']])
del data['komentar']
items=(data['0'].unique())
items
itemset = set(items)
encoded_vals = []
for index, row in data.iterrows():
    rowset = set(row)
    labels = {}
    uncommons = list(itemset - rowset)
    commons = list(itemset.intersection(rowset))
    for uc in uncommons:
        labels[uc] = 0
    for com in commons:
        labels[com] = 1
    encoded_vals.append(labels)
encoded_vals[0]
ohe_df = pd.DataFrame(encoded_vals)
freq_items = apriori(ohe_df, min_support=0.01, use_colnames=True)

```

```
freq_items
```

```
rules = association_rules(freq_items, metric="confidence", min_threshold=0.25)  
rules
```

