

**IMPLEMENTASI *TEXT MINING* DALAM PERBAIKAN
KUALITAS APLIKASI GOOGLE CLASSROOM BERDASARKAN
REVIEW PENGGUNA**

TUGAS AKHIR

**Diajukan Sebagai Salah Satu Syarat untuk Memperoleh Gelar Sarjana
Strata-1 Pada Jurusan Teknik Industri Fakultas Teknologi Industri**



Disusun oleh:

Nama : Anggita Yekti Pawestri

NIM : 18522283

**PROGRAM STUDI TEKNIK INDUSTRI
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2022

PERNYATAAN KEASLIAN TUGAS AKHIR

PERNYATAAN KEASLIAN

Dengan surat pernyataan ini, saya yang bertanda tangan di bawah:

Nama : Anggita Yekti Pawestri

NIM : 18522283

Menyatakan bahwa penelitian tugas akhir yang dilakukan merupakan asli pekerjaan sendiri dengan beberapa kutipan dan sitasi yang sudah saya cantumkan satu persatu dari sumbernya. Apabila di kemudian hari ditemukan terdapat penyimpangan yang menunjukkan bahwa tulisan ini bukan asli milik saya, maka saya bersedia menerima konsekuensi yang diberikan oleh Universitas Islam Indonesia dengan penuh rasa tanggung jawab.

Yogyakarta, 6 Juni 2022



(Anggita Yekti Pawestri)

18522283

SURAT KETERANGAN PELAKSANAAN PENELITIAN TUGAS AKHIR DARI LABORATORIUM



FAKULTAS
TEKNIK INDUSTRI

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

SURAT KETERANGAN PENELITIAN

Nomor : 231/A/Ka.Lab DATMIN/FTI-UII/VI/2022

Assalamu'alaikum Warahmatullahi Wabarakatuh

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

Nama : Anggita Yekti Pawestri
No. Mhs : 18522283
Dosen Pembimbing : Annisa Uswatun Khasanah, ST., M.B.A., M.Sc.

Telah selesai melaksanakan penelitian yang berjudul "Implementasi Text Mining dalam Perbaikan Kualitas Aplikasi Google Classroom Berdasarkan Review Pengguna" di Laboratorium Data Mining, Program Studi Teknik Industri, Fakultas Teknologi Industri, Universitas Islam Indonesia tercatat mulai tanggal 01 Januari 2022 sampai dengan tanggal 15 Januari 2022.

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

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Yogyakarta, 08 Juni 2022

Kepala Laboratorium
Data Mining

Annisa Uswatun Khasanah, ST., M.B.A., M.Sc.



LEMBAR PENGESAHAN DOSEN PEMBIMBING

IMPLEMENTASI *TEXT MINING* DALAM PERBAIKAN KUALITAS
APLIKASI GOOGLE CLASSROOM BERDASARKAN *REVIEW* PENGGUNA

TUGAS AKHIR

Disusun oleh:

Nama : Anggita Yekti Pawestri

NIM : 18522283

Yogyakarta, 6 Juni 2022

Menyetujui,

Dosen Pembimbing



Annisa Uswatun K., ST., M.B.A., M.Sc.

LEMBAR PENGESAHAN DOSEN PENGUJI

**IMPLEMENTASI *TEXT MINING* DALAM PERBAIKAN KUALITAS
APLIKASI GOOGLE CLASSROOM BERDASARKAN *REVIEW* PENGGUNA**

TUGAS AKHIR

Disusun oleh:
 Nama : Anggita Yekti Pawestri
 NIM : 18522283

Telah dipertahankan di depan sidang penguji sebagai satu syarat untuk memperoleh gelar Sarjana-1 Teknik Industri

Yogyakarta, 1 Agustus 2022

Tim Penguji

Annisa Uswatun K., ST., M.B.A., M.Sc.

Ketua

Ira Promasanti Rachmadewi Ir. M.Eng.

Anggota I

Andrie Pasca Hendradewa, S.T., MT.

Anggota II

Mengetahui

Ketua Program Studi Teknik Industri



Imamq Immawan, S.T., M.M.

HALAMAN PERSEMBAHAN

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Tugas akhir yang saya tulis ini dipersembahkan kepada kedua orang tua dan kakak tersayang, serta teman-teman dekat yang senantiasa memberikan dorongan dan semangat selama berkuliah.

Tidak lupa juga tugas akhir ini dipersembahkan kepada seluruh dosen dan karyawan Universitas Islam Indonesia yang telah membantu saya dalam perkuliahan selama menjadi mahasiswa Universitas Islam Indonesia.

الجامعة الإسلامية
الاستدلاء بالاندية

HALAMAN MOTTO

وَلَا تَهِنُوا وَلَا تَحْزَنُوا وَأَنْتُمْ الْأَعْلَوْنَ إِنْ كُنْتُمْ مُؤْمِنِينَ

”Dan janganlah kamu (merasa) lemah, dan jangan (pula) bersedih hati, sebab kamu paling tinggi (derajatnya), jika kamu orang beriman.” (Q.S. Ali ‘Imran Ayat 139)

فَإِنَّ مَعَ الْعُسْرِ يُسْرًا , إِنَّ مَعَ الْعُسْرِ يُسْرًا

“Maka sesungguhnya beserta kesulitan ada kemudahan, sesungguhnya beserta kesulitan itu ada kemudahan.” (Q.S Al-Insyirah Ayat 5 – 6)

وَأَنْ لَيْسَ لِلْإِنْسَانِ إِلَّا مَا سَعَى ﴿٣٩﴾ وَأَنَّ سَعْيَهُ سَوْفَ يُرَى ﴿٤٠﴾ ثُمَّ يُجْزَاهُ الْجَزَاءَ
الْأَوْفَى ﴿٤١﴾ وَأَنَّ إِلَىٰ رَبِّكَ أَلْمُتَبَىٰ ﴿٤٢﴾

“Dan bahwa manusia hanya memperoleh apa yang telah diusahakannya, dan sesungguhnya usahanya itu kelak akan diperlihatkan (kepadanya), kemudian akan diberi balasan kepadanya dengan balasan yang paling sempurna, dan sesungguhnya kepada Tuhanmulah kesudahannya (segala sesuatu),” (Q.S. An-Najm Ayat 39 – 42)

الجمعة المباركة
الاستدلال بالاندية

KATA PENGANTAR

Assalamualaikum Warahmatullahi Wabarakatuh.

Alhamdulillahirabbil'alamin, puji dan syukur ke hadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul 'Implementasi *Text Mining* dalam Perbaikan Kualitas Aplikasi Google Classroom Berdasarkan *Review* Pengguna' dengan baik dan tepat pada waktunya. Tidak lupa, shalawat serta salam selalu tercurah kepada junjungan kita Nabi Muhammad SAW beserta keluarga dan sahabat yang telah menghantarkan umat manusia ke jalan yang diridhai Allah SWT.

Tugas akhir merupakan salah satu syarat yang harus dipenuhi untuk memperoleh gelar sarjana strata-1 pada Program Studi Teknik Industri Universitas Islam Indonesia. Dari tugas akhir yang dilaksanakan, penulis dapat mengetahui penerapan ilmu yang dipelajari di bangku kuliah dalam penelitian ilmiah.

Dalam penyusunan tugas akhir, penulis mendapatkan banyak bimbingan dan dukungan dari berbagai pihak, baik secara langsung maupun tidak langsung. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. Ir. Hari Purnomo M.T. selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
2. Bapak Dr. Taufiq Immawan S.T., M.M. selaku Ketua Program Studi Strata-1 Teknik Industri Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Ibu Annisa Uswatun Khasanah ST., M.B.A., M.Sc. selaku dosen pembimbing tugas akhir yang telah memberikan arahan serta bimbingannya selama penulis mengerjakan tugas akhir.
4. Ibu, bapak, dan kakak penulis yang selalu memberikan semangat, dukungan, serta doa selama penulis menyelesaikan penyusunan tugas akhir.
5. Elvina, Fara Anien, Hiz, dan Ismah selaku teman dekat penulis dari bangku SMA yang selalu memberikan dukungan mental bagi penulis untuk menyelesaikan tugas akhir.
6. Fildzah, Clarissa, dan Rani selaku teman dekat penulis selama di bangku kuliah semenjak semester awal yang senantiasa memberikan bantuan selama kuliah.

7. Taa selaku teman *online* penulis yang telah bersedia mendengarkan keluh kesah penulis selama proses penyusunan tugas akhir.
8. Teman-teman mahasiswa Teknik Industri Universitas Islam Indonesia Angkatan 2018.

Penulis menyadari bahwa tugas akhir ini masih jauh dari kata sempurna. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun sehingga laporan tugas akhir ini dapat menjadi lebih baik. Semoga laporan tugas akhir ini dapat memberikan manfaat bagi penulis dan pembaca.

Wassalamualaikum Warahmatullahi Wabarakatuh.

Yogyakarta, 6 Juni 2022

Anggita Yekti Pawestri

الجمهورية الإسلامية الإندونيسية

ABSTRAK

Internet sekarang ini menyerupai kebutuhan primer bagi seluruh lapisan masyarakat. Salah satu bidang yang terpengaruh dampak dari perkembangan internet adalah bidang pendidikan yang dikenal dengan istilah *e-learning*. Google Classroom merupakan salah satu *platform e-learning* yang banyak dipakai sejak adanya Covid-19. Google Classroom menduduki persentase sebesar 26,1% sebagai aplikasi yang banyak digunakan untuk pembelajaran *online*. Semakin meningkatnya pengguna dari suatu aplikasi diikuti dengan semakin banyaknya kritik dan saran yang diberikan oleh pengguna. Pada bulan Desember 2021 Google Classroom memperoleh *rating* rata-rata 3,3 dari 5 pada Google Play, sehingga perlu dilakukan penelitian untuk meningkatkan *rating* dari aplikasi Google Classroom berdasarkan *review* yang diberikan oleh pengguna dengan *text mining*. Data yang diambil untuk penelitian ini berjumlah 77.454 data *review* pengguna Google Classroom dari Google Play. Pengolahan data dengan analisis sentimen menggunakan metode *Textblob* dan *Naïve Bayes Classifier* untuk mengetahui sentiment dari *review* yang diberikan. Hasil dari analisis sentimen dengan ukuran latih 70 dan data uji 30 menghasilkan akurasi sebesar 92,39%. Data dengan sentiment negative kemudian diolah dengan *association rules* untuk menemukan kata yang digunakan sebagai kata kunci permasalahan. Dari *association rules* diperoleh kata kunci meliputi '*assignment*', '*upload*', '*submit*', '*file*', '*class*', '*notification*', dan '*dark*'. Kata tersebut kemudian dianalisis menggunakan *fishbone diagram* untuk mencari tahu akar masalahnya. Akar masalah yang ditemui meliputi kendala dalam meng-upload *file* yang terintegrasi dengan Google Drive, tidak adanya pengaturan untuk mengubah tampilan menjadi '*dark mode*', pengguna memiliki koneksi internet yang baik tetapi proses *upload file* lambat, dan lain-lain. Rekomendasi perbaikan yang diberikan adalah melakukan perbaikan integrasi *file* dengan Google Drive, pengaturan untuk mengubah tampilan ke mode gelap sesuai dengan preferensi pengguna, dan melakukan *update* pada *server* Google Classroom sehingga proses *upload file* dapat dilakukan lebih cepat.

Kata kunci: Analisis sentimen, *text mining*, *Textblob*, *Naïve Bayes Classifier*, *association rules*, *fishbone diagram*, Google Classroom

DAFTAR ISI

HALAMAN JUDUL	i
PERNYATAAN KEASLIAN TUGAS AKHIR	ii
SURAT KETERANGAN PELAKSANAAN PENELITIAN TUGAS AKHIR	iii
LEMBAR PENGESAHAN DOSEN PEMBIMBING	iv
LEMBAR PENGESAHAN DOSEN PENGUJI.....	v
HALAMAN PERSEMBAHAN	vi
HALAMAN MOTTO.....	vii
KATA PENGANTAR	viii
ABSTRAK.....	x
DAFTAR ISI.....	xi
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR.....	xiv
DAFTAR PERSAMAAN.....	xv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Permasalahan	4
1.3 Batasan Permasalahan	4
1.4 Tujuan Penelitian	5
1.5 Manfaat Penelitian	5
1.6 Sistematika Penulisan Tugas Akhir	5
BAB II KAJIAN LITERATUR.....	7
2.1 Kajian Induktif	7
2.2 Kajian Deduktif.....	14
2.2.1 <i>E-Learning</i>	14
2.2.2 <i>Learning Management System</i>	15
2.2.3 <i>Google Classroom</i>	15
2.2.4 <i>Text Data Mining</i>	16
2.2.5 <i>Analisis Sentimen</i>	18
2.2.6 <i>Machine Learning</i>	18
2.2.7 <i>Klasifikasi Sentimen</i>	19

2.2.8	<i>TextBlob</i>	20
2.2.9	<i>Naïve Bayes Classifier</i>	20
2.2.10	<i>Evaluasi Klasifikasi</i>	21
2.2.11	<i>Association Rules</i>	22
2.2.12	<i>Fishbone Diagram</i>	23
BAB III	METODE PENELITIAN	24
3.1	Jenis Data.....	24
3.2	Pengumpulan Data.....	24
3.3	Pengolahan Data	25
3.4	Alur Penelitian.....	25
BAB IV	PENGUMPULAN DAN PENGOLAHAN DATA.....	29
4.1	Pengumpulan Data	29
4.2	Pengolahan Data.....	30
4.2.1	<i>Preprocessing Data</i>	30
4.2.2	Pelabelan Data	35
4.2.3	Klasifikasi Menggunakan Algoritma <i>Naïve Bayes</i>	37
4.2.4	<i>Association Rules</i>	42
4.2.5	Analisis dengan <i>Fishbone Diagram</i>	53
BAB V	PEMBAHASAN.....	56
BAB VI	PENUTUP	62
	DAFTAR PUSTAKA	64
	LAMPIRAN.....	70

الجمعة، الامتداد الاندو

DAFTAR TABEL

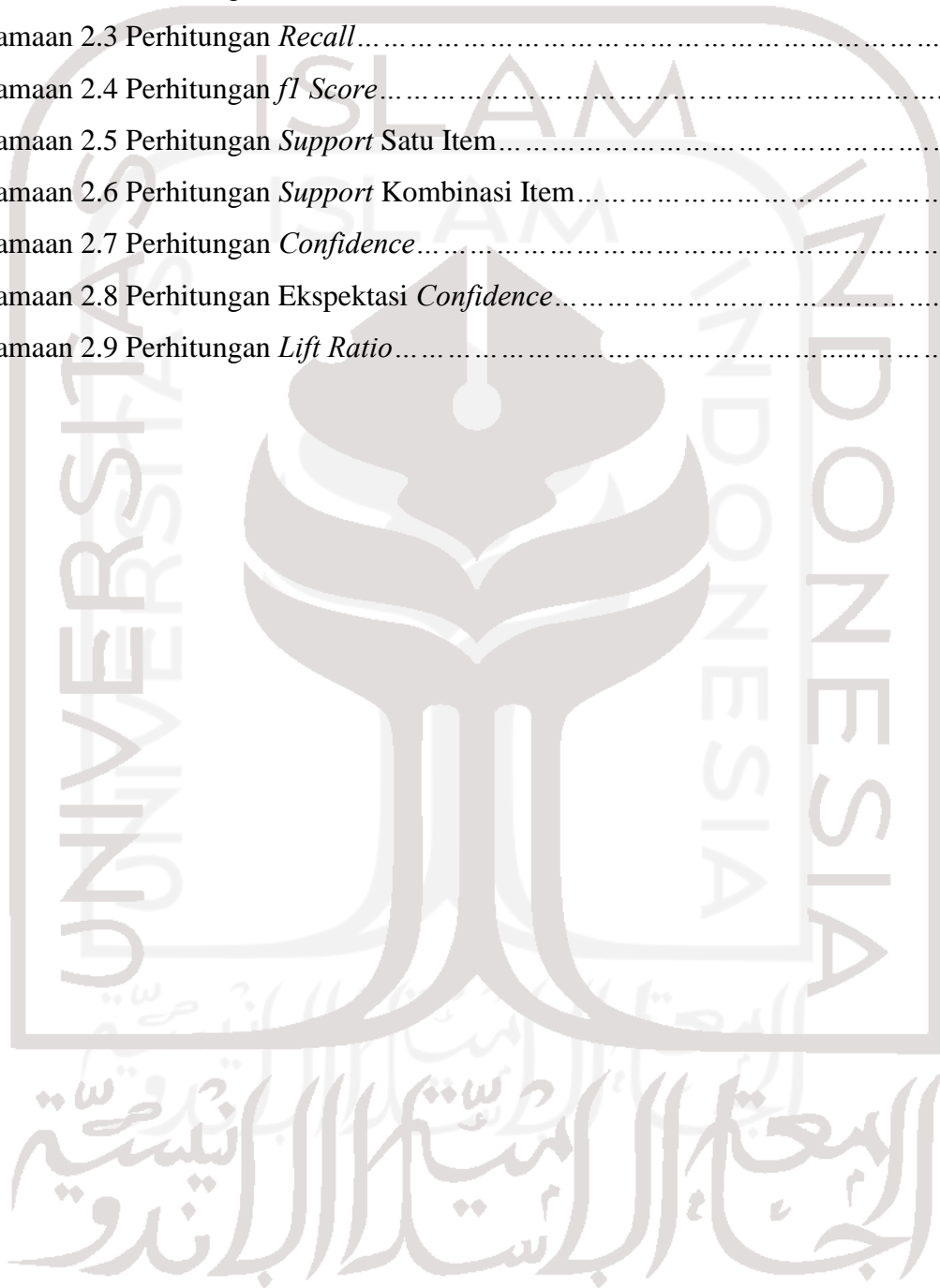
Tabel 2. 1 Kajian Induktif untuk Metode Analisis Sentimen	10
Tabel 2. 2 Kajian Induktif Metode <i>Association Rules</i>	13
Tabel 4. 1 Proses <i>Case Folding</i>	30
Tabel 4. 2 Proses <i>Remove Punctuation</i>	31
Tabel 4. 3 Proses <i>ASCII Character Removal</i>	32
Tabel 4. 4 Proses <i>Emoji Removal</i>	33
Tabel 4. 5 Proses <i>Tokenizing</i>	33
Tabel 4. 6 Proses <i>Stopwords Removal</i>	34
Tabel 4. 7 Proses <i>Lemmatization</i>	35
Tabel 4. 8 Hasil Pelabelan dengan <i>Textblob</i>	36
Tabel 4. 9 Hasil Metrics Klasifikasi Pelabelan <i>TextBlob</i> ukuran 50/50	38
Tabel 4. 10 Hasil Metrics Klasifikasi Pelabelan <i>TextBlob</i> ukuran 60/40	39
Tabel 4. 11 Hasil Metrics Klasifikasi Pelabelan <i>TextBlob</i> ukuran 70/30	39
Tabel 4. 12 Hasil Metrics Klasifikasi Pelabelan <i>TextBlob</i> ukuran 80/10	40
Tabel 4. 13 Perbandingan Ukuran <i>Split Data</i> , <i>Accuracy</i> , dan <i>F1 Score</i> untuk Klasifikasi Pelabelan dengan <i>Textblob</i>	41
Tabel 4. 14 <i>Frequent Words</i>	43
Tabel 4. 15 Tabel <i>Association Rules Conclusion Assignment</i>	44
Tabel 4. 16 Tabel <i>Association Rules Conclusion Assignment</i>	47
Tabel 4. 17 Tabel <i>Association Rules Conclusion Submit</i>	48
Tabel 4. 18 Tabel <i>Association Rules Conclusion File</i>	49
Tabel 4. 19 Tabel <i>Association Rules Conclusion File</i>	50
Tabel 4. 20 Tabel <i>Association Rules Conclusion Notification</i>	52
Tabel 4. 21 Tabel <i>Association Rules Conclusion Notification</i>	52
Tabel 5. 1 Rekomendasi Perbaikan Berdasarkan Aspek	60

DAFTAR GAMBAR

Gambar 1. 1 <i>Review</i> dan <i>Rating</i> yang diberikan pengguna Google Classroom	3
Gambar 2. 1 Google Classroom.....	16
Gambar 2. 2 <i>Fishbone diagram</i>	23
Gambar 3. 1 Diagram Alir Penelitian	26
Gambar 4. 1 Grafik Jumlah <i>Review</i> dan <i>Rating</i> Pengguna Google Classroom.....	29
Gambar 4. 2 Hasil Pengumpulan Data.....	30
Gambar 4. 3 Visualisasi Hasil Pelabelan Sentimen dengan <i>Textblob</i>	36
Gambar 4. 4 <i>Scatter Plot</i> Pelabelan <i>Textblob</i>	37
Gambar 4. 5 <i>Confusion Matrix</i> Klasifikasi Pelabelan <i>Textblob</i> ukuran 50/50	38
Gambar 4. 6 <i>Confusion Matrix</i> Klasifikasi Pelabelan <i>Textblob</i> ukuran 60/40	39
Gambar 4. 7 <i>Confusion Matrix</i> Klasifikasi Pelabelan <i>Textblob</i> ukuran 60/40	40
Gambar 4. 8 <i>Confusion Matrix</i> Klasifikasi Pelabelan <i>Textblob</i> ukuran 80/20	41
Gambar 4. 9 Desain Operasi <i>Association Rules</i> pada RapidMiner.....	43
Gambar 4. 10 <i>Fishbone diagram</i> Terkait Masalah <i>Assignment</i>	53
Gambar 4. 11 <i>Fishbone diagram</i> Terkait Masalah Proses <i>Upload</i> dan <i>Submit File</i>	54
Gambar 4. 12 <i>Fishbone diagram</i> Terkait Masalah Bergabung ke Dalam Kelas	54
Gambar 4. 13 <i>Fishbone diagram</i> Terkait Masalah Notifikasi	55
Gambar 4. 14 <i>Fishbone diagram</i> Terkait Masalah Masalah Tampilan Aplikasi.....	55

DAFTAR PERSAMAAN

Persamaan 2.1 Perhitungan <i>Accuracy</i>	21
Persamaan 2.2 Perhitungan <i>Precision</i>	22
Persamaan 2.3 Perhitungan <i>Recall</i>	22
Persamaan 2.4 Perhitungan <i>f1 Score</i>	22
Persamaan 2.5 Perhitungan <i>Support</i> Satu Item.....	22
Persamaan 2.6 Perhitungan <i>Support</i> Kombinasi Item.....	22
Persamaan 2.7 Perhitungan <i>Confidence</i>	22
Persamaan 2.8 Perhitungan Ekspektasi <i>Confidence</i>	23
Persamaan 2.9 Perhitungan <i>Lift Ratio</i>	23



BAB I

PENDAHULUAN

1.1 Latar Belakang

Internet sekarang ini menyerupai kebutuhan primer bagi seluruh lapisan masyarakat. Berdasarkan data hasil analisis *Hootsuite* dan *We are Social* melalui Drone Emprit menunjukkan bahwa pengguna internet secara global memiliki persentase sebesar 59,5% (Fahmi, 2021). Sementara itu, di Indonesia pengguna internet secara keseluruhan memiliki persentase 73,7%. Perkembangan yang signifikan dalam penggunaan internet tersebut berpengaruh terhadap aktivitas yang ada di bidang dunia nyata. Salah satu dari sekian bidang yang terpengaruh dampak dari perkembangan internet ini merupakan bidang pendidikan yang dikenal dengan istilah *e-learning*.

E-learning merupakan proses pembelajaran menggunakan teknologi informasi dan teknologi untuk membuat pelatihan, melakukan distribusi materi pembelajaran, dan berkomunikasi antara siswa dan pengajar untuk mengelola pembelajaran (Harandi, 2015). Pemanfaatan dari *e-learning* terbagi ke dalam beberapa jenis yang salah satunya adalah *Learning Management System (LMS)*. Menurut Cavus dan Alhih (2014) *learning management system (LMS)* atau dapat disebut dengan sistem manajemen pembelajaran didefinisikan sebagai sebuah teknologi atau sistem untuk mengelola dokumen pembelajaran atau training, mendistribusikan materi melalui internet, dan melakukan *monitoring* pada pembelajaran.

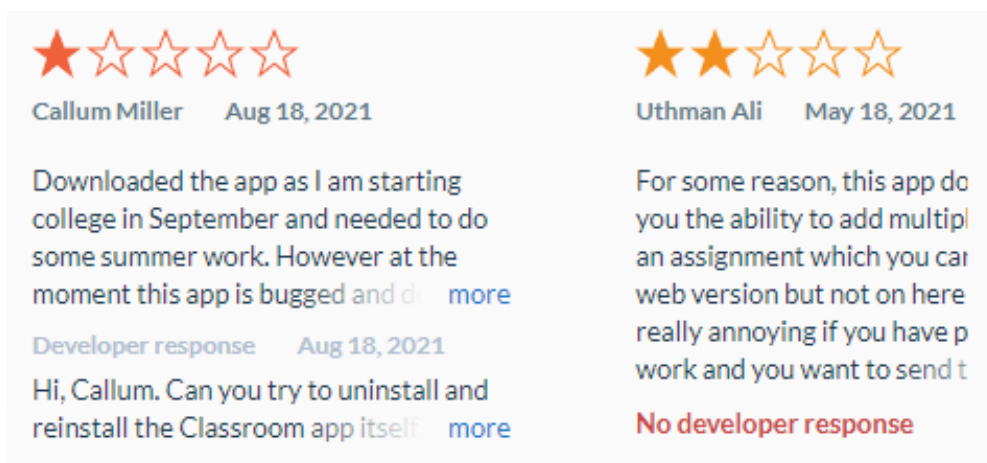
Penggunaan LMS dalam pembelajaran semakin banyak digunakan semenjak munculnya virus Covid-19. Maret 2020 menjadi awal mula masuknya virus Covid-19 ke negara Indonesia. Dengan masuknya virus ini, Indonesia memberlakukan Pembatasan Sosial Berskala Besar (PSBB) yang diberlakukan selama dua minggu, dimulai sejak 31 Maret 2020 dengan ditandatanganinya Peraturan Pemerintah Nomor 21 Tahun 2020. PSBB ini meliputi kebijakan libur sekolah, kegiatan keagamaan, tempat kerja, dan pembatasan penggunaan prasarana publik. Sektor pendidikan menjadi salah satu yang paling terdampak oleh wabah pandemi Covid-19 ini tidak dapat menunda kegiatan

belajar mengajar lebih lanjut. Melihat kondisi yang tidak kunjung membaik, Nadiem Makarim selaku Menteri Pendidikan dan Kebudayaan Indonesia menerbitkan surat edaran Nomor 4 Tahun 2020 menyampaikan bahwa kegiatan belajar mengajar akan dilakukan di rumah secara *online* atau daring. Seluruh jenjang pendidikan, dari SD, SMP, hingga SMA melakukan pembelajaran secara daring.

Selama satu tahun terakhir, kegiatan pembelajaran jarak jauh banyak melakukan kolaborasi dengan platform pembelajaran yang tersedia secara *online*. Google Classroom merupakan salah satu platform pembelajaran yang sekarang ini banyak digunakan oleh orang-orang. Berdasarkan survei yang dilakukan oleh Arus Survei Indonesia (ASI) menyebutkan bahwa sejak adanya pandemi Covid-19, sebesar 26,1 % orang menggunakan platform Google Classroom selama pembelajaran jarak jauh (Kamil, 2020). Google Classroom menduduki persentase tertinggi sebagai aplikasi yang digunakan untuk kegiatan pembelajaran jarak jauh, yang lalu diikuti dengan platform lain seperti Ruangguru (17,1 %) dan Rumah Belajar (15,2%).

Pada tahun 2014 Google mengembangkan platform Google Classroom sebagai alat bagi pendidik dan peserta didik untuk membuat kelas virtual secara gratis. Di dalam Google Classroom pendidik dapat berinteraksi bersama peserta didik secara *online*, mulai dari memberikan pengumuman dan informasi, menyebarkan materi, mengadakan kuis dan tugas, dan melakukan penilaian. Google Classroom dapat dijumpai dalam bentuk *website* maupun aplikasi pada telepon genggam. Bentuk aplikasi dari Google Classroom jauh lebih praktis karena pengguna dapat memperoleh informasi secara *real-time*.

Semakin banyaknya pengguna aplikasi dari Google Classroom, tentu menimbulkan banyak kritik dan saran dari pengguna. Berdasarkan tingkat *rating* yang diutarakan oleh pengguna di Google Play pada bulan Desember 2021, aplikasi Google Classroom memperoleh *rating* rata-rata 3,3 dari 5. Adapun gambaran dari *review* yang diberikan oleh pengguna aplikasi Google Classroom bersama dengan *rating* ditunjukkan pada gambar 1.1.



Gambar 1. 1 *Review* dan *Rating* yang diberikan pengguna Google Classroom

Sumber: AppFollow.io (2022)

Rating tersebut tergolong rendah apabila dibandingkan dengan aplikasi Google lainnya, sehingga perlu dilakukan perbaikan untuk meningkatkan *rating* dari aplikasi Google Classroom. Sebelum dilakukan perbaikan, perlu dicari akar masalah yang menyebabkan aplikasi Google Classroom memiliki *rating* yang rendah. *Review* dari pengguna aplikasi Google Classroom digunakan sebagai sumber data untuk digali untuk menemukan masalah dari aplikasi tersebut. *Text mining* didefinisikan sebagai salah satu teknik yang dapat diimplementasikan untuk mengklasifikasikan data dengan menemukan pola yang menarik dari kumpulan data teks yang besar (Pratama & Atmi, 2020). Menurut Antons *et al.* (2020), *text mining* pada dasarnya memerlukan pendekatan kuantitatif untuk menganalisis data teks tersebut. Implementasi *text mining* pertama kali dilakukan pada pertengahan tahun 1980 dan terus berkembang hingga sekarang merambah ke berbagai bidang, seperti bisnis, ekonomi, dan edukasi (Zanini & Dhawan, 2015)

Berdasarkan penjabaran dari latar belakang, maka perlu dilakukan penelitian dalam bentuk tugas akhir untuk menganalisis masalah dari *review* pengguna Google Classroom. Pada penelitian ini *text mining* digunakan untuk menggali data *review* pengguna aplikasi Google Classroom khususnya di platform android. *Review* yang menjadi kriteria data yang digunakan dalam penelitian ini adalah berupa kritik dan saran dengan bahasa Inggris. Bahasa tersebut dipilih karena merupakan mayoritas bahasa yang digunakan dalam pemberian *review*, sehingga dapat mewakili kritik dari pengguna. Pengumpulan data akan difokuskan dari platform Google Play Store dengan melakukan *scraping data* melalui *website* AppFollow. Dalam penelitian ini dilakukan analisis klasifikasi sentimen

untuk mengelompokkan *review* dengan sentimen positif dan negatif. Metode yang telah banyak diterapkan dalam riset terkait analisis sentimen meliputi *Support Vector Machine*, *Naïve Bayes Classifier*, *K-Nearest Neighbour*. Metode *Naïve Bayes Classifier* digunakan dalam penelitian ini karena dianggap sebagai salah satu metode analisis sentimen yang sederhana, mudah dimengerti, dan memiliki tingkat *accuracy* yang cukup tinggi (Suryani *et al.*, 2019). Selain metode *Naïve Bayes Classifier*, akan digunakan *library* pelabelan *lexicon-based* pada Python yakni *Textblob*. Klasifikasi analisis sentimen yang dilakukan bersifat *hybrid* menggunakan *machine learning* dan *lexicon-based sentiment analysis*. Menurut Santoso (2021) *association rule* digunakan untuk mencari aturan atau kombinasi item yang paling sering terjadi dalam suatu data. Dalam penelitian ini *association rule* digunakan untuk mencari kombinasi dari kata-kata tersebut. Setelah diklasifikasikan maka dapat dicari akar masalahnya dengan menggunakan *root cause analysis* berupa *fishbone diagram*, sehingga dapat dijadikan rujukan untuk menyusun rekomendasi perbaikan pada aplikasi Google Classroom.

1.2 Rumusan Permasalahan

Berdasarkan inti masalah yang telah dijabarkan di bagian latar belakang, maka dapat disusun rumusan masalah sebagai berikut:

1. Berdasarkan analisis sentimen dengan *Textblob* dan *Naïve Bayes Classifier* yang dilakukan, berapakah nilai akurasi yang diperoleh?
2. Berdasarkan *association rules*, kata apa saja yang paling sering muncul dalam data *review* pengguna Google Classroom?
3. Bagaimana hasil dari analisis akar masalah yang dapat dijadikan sebagai referensi perbaikan aplikasi Google Classroom?
4. Apa rekomendasi yang dapat diberikan sebagai bentuk perbaikan untuk aplikasi Google Classroom berdasarkan masalah yang telah ditemukan?

1.3 Batasan Permasalahan

Batasan masalah yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Data yang digunakan dalam penelitian ini adalah data *review* pengguna Google Classroom dari Google Play dengan kategori bahasa Inggris.

2. *Data Scraping* dilakukan dari *website* Appfollow dimulai dari tanggal Desember 2021 sampai dengan Juli 2021.
3. Untuk membantu pengolahan data, digunakan Microsoft Excel 365, Jupyter Notebook dengan bahasa pemrograman Python versi 3.9, dan RapidMiner Studio versi 9.10.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Untuk memperoleh nilai akurasi dari analisis sentimen dengan *Textblob* dan *Naïve Bayes Classifier*.
2. Untuk menemukan kata yang paling sering muncul dalam data *review* pengguna Google Classroom.
3. Untuk menemukan akar masalah yang dapat dijadikan referensi untuk perbaikan aplikasi Google Classroom.
4. Untuk membuat rekomendasi perbaikan terhadap aplikasi Google Classroom berdasarkan masalah yang telah ditemukan.

1.5 Manfaat Penelitian

Manfaat dengan adanya penelitian ini adalah untuk menerapkan *text mining* dalam proses perbaikan aplikasi Google Classroom berdasarkan kritik dan saran pengguna aplikasi. Pihak *developer* aplikasi Google Classroom dapat mengetahui aspek-aspek apa yang dikeluhkan secara *mayoritas*.

1.6 Sistematika Penulisan Tugas Akhir

Sistematika penulisan yang diterapkan dalam laporan tugas akhir ini terbagi ke dalam beberapa bab sebagai berikut:

BAB I Pendahuluan

Pada bagian ini memuat ringkasan dari laporan tugas akhir yang meliputi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan tugas akhir.

BAB II Kajian Literatur

Kajian literatur meliputi penelitian yang telah dilakukan dahulu yang menyerupai topik penelitian dari penelitian yang dilakukan. Pada bab ini memuat kajian literatur deduktif dan induktif untuk menunjukkan bahwa topik penelitian tugas akhir yang diambil memenuhi syarat dan kriteria.

BAB III Metode Penelitian

Bab ini meliputi struktur dari jalannya penelitian meliputi objek penelitian, metode pengumpulan data, metode pengolahan data, dan diagram dari proses penelitian.

BAB IV Pengumpulan dan Pengolahan Data

Membahas tahapan dari pengumpulan sampai dengan pengolahan data berdasarkan metode penelitian yang telah dijabarkan di bab sebelumnya. Hasil dari pengolahan data akan digunakan untuk tahapan berikutnya.

BAB V Pembahasan

Pada bab ini diuraikan hasil yang didapatkan dari bab sebelumnya yang lebih jelas dan detail. Dari bab ini akan diperoleh jawaban dari tujuan penelitian yang telah disebutkan.

BAB VI Penutup

Bab penutup berisi kesimpulan yang diperoleh dari penelitian yang dilakukan yang disampaikan secara singkat dan padat. Selain itu, terdapat saran yang diberikan untuk peneliti selanjutnya terkait topik riset yang serupa.

BAB II

KAJIAN LITERATUR

2.1 Kajian Induktif

Sari dan Wibowo (2019) sebelumnya melakukan penelitian yang bertujuan untuk mengetahui sentimen serta menambah fitur konversi emoji berdasarkan data dari aplikasi Twitter terkait JD.id. Dari pengambilan data, terkumpul sebanyak 900 *tweets* dengan pembagian masing-masing 300 *tweets* masuk ke dalam label sentimen positif, negatif, dan netral. Dari data yang terkumpul kemudian dilakukan *preprocessing data* untuk membersihkan data dan pembobotan menggunakan *tf-idf*. Data kemudian diklasifikasikan menggunakan algoritma *Naïve Bayes Classifier* dengan penambahan karakteristik berupa konversi emoji yang diperkirakan dapat menambah tingkat *accuracy*. Hasil menunjukkan bahwa analisis sentimen dengan metode *Naïve Bayes Classifier* tanpa adanya pembobotan *tf-idf* dan konversi emoji diperoleh nilai *accuracy* sebesar 96,44%. Sementara itu, *Naïve Bayes Classifier* yang ditambahkan pembobotan *tf-idf* dan konversi emoji mendapatkan nilai *accuracy* sebesar 98%.

Riset terdahulu yang dilakukan oleh Mubarok *et al.* (2017) terkait analisis sentimen terhadap *review* produk, menggunakan algoritma *Naïve Bayes*. Selain metode algoritma tersebut, juga digunakan *part-of-speech* (POS) dan *feature selection* menggunakan *Chi-Square* dalam *preprocessing data*. Proses *POS tagging* menggunakan *library* dari Stanford CoreNLP. Berdasarkan hasil klasifikasi dihasilkan *output f1 score* sebesar 78,12%.

Pada tahun 2019 Ali dan Gayar melakukan penelitian terkait analisis dengan tujuan untuk menemukan cara terbaik untuk melabeli sentimen data secara otomatis. Data yang digunakan dalam penelitian ini adalah data *e-mail* berjumlah 500.000 yang didapatkan dari perusahaan Enron. Data kemudian dilakukan pemrosesan untuk menghilangkan *header*, tanda tangan, kutipan, dan kode pemrograman. Setelah itu dilanjutkan *text preprocessing* pada isi *e-mail*. Selanjutnya dilakukan ekstraksi fitur yang dibagi ke dalam dua kelompok, fitur *unigram* dan grup *SentiWordNet Word*. TF-IDF juga digunakan

untuk menghitung bobot kata yang muncul di dalam dokumen. Proses pelabelan data menggunakan *K-means* yakni pendekatan pelabelan *unsupervised*. Selain itu digunakan pendekatan pelabelan *lexicon* dengan *VADER*. Setelah data dilabeli kemudian dimodelkan untuk menguji data dengan algoritma *Naïve Bayes* dan *SVM*. *Output* yang dihasilkan dengan algoritma *SVM* adalah *accuracy* sebesar 85,03%. Sementara, untuk algoritma *Naïve Bayes* diperoleh *accuracy* sebesar 73,13%.

Dey *et al.* (2016) dalam penelitiannya terkait analisis sentimen, melakukan perbandingan antara dua metode yakni *Naïve Bayes* dan *K-Nearest Neighbour* (K-NN). Hal yang dibandingkan dalam penelitian ini adalah *accuracy* secara keseluruhan, *precision*, dan *recall value*. Data yang digunakan merupakan data ulasan film dan hotel, yang berisi masing-masing 5.000 ulasan positif dan negatif. Berdasarkan hasil pengolahan data menggunakan dua algoritma, diketahui bahwa klasifikasi pada data ulasan film menggunakan *Naïve Bayes* menunjukkan hasil yang lebih baik dengan *accuracy* di atas 80% dibandingkan dengan k-NN. Sementara itu, klasifikasi pada ulasan hotel menunjukkan hasil *accuracy* yang lebih rendah untuk keduanya. Jadi, dapat disimpulkan bahwa *Naïve Bayes Classifier* berhasil dalam analisis ulasan film.

Najib *et al.* (2019) sebelumnya melakukan penelitian dengan tujuan untuk menganalisis sentimen yang berdasar pada ontologi. Metode yang digunakan sebagai perbandingan adalah *Lexicon-based* dan *Support Vector Machine* (SVM). Ontologi yang digunakan adalah ekonomi dengan atribut finansial, kesejahteraan, dan lapangan kerja. Pengumpulan data dilakukan dengan metode *crawling data* dari Twitter pada periode kampanye Presiden tahun 2019. Dari 700.000 data *tweet* yang terkumpul dengan kata kunci Jokowi dan Prabowo, kemudian dilakukan *labelling*, pembuatan ontologi, dan *preprocessing data*. Pembuatan ontologi baru dilakukan untuk validasi ontologi yang telah dibuat sebelumnya. Pengolahan data dilakukan dengan klasifikasi dan analisis sentimen untuk membandingkan kedua metode. Berdasarkan hasil pengolahan data menggunakan dua metode, ditemukan bahwa pada metode *Lexicon-based* menghasilkan *accuracy* sebesar 39% dengan sebagian sentimen masuk ke dalam kategori netral. Sedangkan, untuk metode *SVM* diperoleh *accuracy* sebesar 83% dengan sentimen positif sebagian besar ada di atribut finansial dan kesejahteraan.

Riset yang dilakukan (Perdana *et al.*, 2021) terkait analisis sentimen, menggunakan pendekatan *hybrid* dengan menggabungkan metode *Support Vector Machine* (SVM) dan

TextBlob. Data yang dijadikan objek penelitian adalah ulasan berupa opini pengguna Twitter terhadap layanan Telkomsel. Dari data yang terkumpul kemudian dianalisis menggunakan *Textblob* untuk memberikan label sentimen positif, negatif, atau netral. Dari analisis dengan *Textblob* saja diperoleh hasil *accuracy* sebesar 58,59%. Dengan dilakukannya penambahan metode klasifikasi menggunakan SVM diperoleh hasil *accuracy* sebesar 75%.

Pada penelitian terkait analisis sentimen yang dilakukan oleh Khan *et al.* pada tahun 2021, digunakan perpaduan dua metode yakni *Textblob* dengan beberapa algoritma *supervised machine learning*. Data yang digunakan dalam penelitian ini merupakan ulasan pengguna Twitter terkait Covid-19 dalam Bahasa Inggris yang dikumpulkan melalui *software* Rstudio. Data mentah kemudian dilakukan *preprocessing* supaya data menjadi bersih dan seragam. Selanjutnya dilakukan pelabelan data menggunakan *Textblob* untuk menentukan jenis sentimen dari *tweets*. Data kemudian diolah dengan *bag-of-words* (BoW) dan *term-frequency-inverse document frequency* (TF-IDF). Tahap terakhir adalah pemodelan dengan *supervised machine learning* meliputi *random forest*, *gradient boosting machine*, *extra tree classifier*, *logistic regression*, dan *Support Vector Machine* (SVM). Berdasarkan pemodelan yang dilakukan, ditemukan bahwa metode *gradient boosting machine* memiliki *output accuracy* tertinggi yakni 96%. Selain itu, diketahui bahwa hasil *accuracy* yang diperoleh lebih tinggi dengan menggunakan TF-IDF dibandingkan dengan BoW.

Laksono *et al.* (2019) melakukan riset terkait analisis sentimen terhadap ulasan pelanggan rumah makan. Objektif dari penelitian ini adalah untuk menganalisis sentimen ulasan secara online dari TripAdvisor terhadap sepuluh restoran terbaik di Kota Surabaya. Dalam riset ini juga dicari tahu perbandingan dua metode analisis sentimen yakni *Textblob* dan *Naïve Bayes* untuk memperoleh metode analisis terbaik. Berdasarkan 337 data yang terkumpul, sebanyak 269 data digunakan sebagai data latih dan sisanya digunakan sebagai data uji untuk klasifikasi *Naïve Bayes*. Berdasarkan pengolahan data menggunakan dua pendekatan diperoleh nilai *accuracy* untuk *Naïve Bayes* sebesar 72,06% dan *TextBlob* sebesar 69,12%.

Tabel 2. 1 Kajian Induktif untuk Metode Analisis Sentimen

Peneliti	Metode	Objek	Hasil Penelitian
(Sari & Wibowo, 2019)	<i>Naïve Bayes Classifier</i>	Ulasan pelanggan toko online JD.id	Dari analisis sentiment dengan <i>Naïves Bayes Classifier</i> diperoleh nilai <i>accuracy</i> sebesar 98%.
(Mubarok et al., 2017)	<i>Naïve Bayes Classifier</i>	Ulasan produk	Berdasarkan hasil klasifikasi dihasilkan <i>output f1 score</i> sebesar 78,12%.
(Ali & Gayar, 2019)	<i>K-Means, Support Vector Machine, dan Naïve Bayes</i>	Dataset <i>e-mail</i> perusahaan Enron	<i>Output</i> yang dihasilkan dengan algoritma SVM adalah <i>accuracy</i> sebesar 85,03%. Sementara, untuk algoritma <i>Naïve Bayes</i> diperoleh <i>accuracy</i> sebesar 73,13%.
(Dey et al., 2016)	<i>Naïve Bayes dan K-Nearest Neighbour (K-NN)</i>	Data ulasan film dan hotel	Berdasarkan hasil pengolahan data, diketahui bahwa hasil analisis dengan <i>Naïve Bayes</i> lebih unggul dengan <i>accuracy</i> di atas 80% dibandingkan dengan k-NN.
(Najib et al., 2019)	<i>Lexicon-based dan Support Vector Machine</i>	Ulasan pengguna Twitter pada periode kampanye Pilpres tahun 2019	Berdasarkan hasil pengolahan data, diketahui metode <i>Lexicon-based</i> memiliki <i>accuracy</i> 39%. Sedangkan, untuk metode SVM diperoleh <i>accuracy</i> sebesar 83%.
(Chaithra, 2019)	<i>VADER dan Naïve Bayes Classifier</i>	Komentar penonton YouTube pada video <i>unboxing</i>	Dari klasifikasi diperoleh <i>accuracy</i> sebesar 79,78% dengan persentase <i>f1 score</i> adalah 93,72%.

Peneliti	Metode	Objek	Hasil Penelitian
		telepon genggam	
(Perdana et al., 2021)	<i>TextBlob</i> dan <i>Support Vector Machine</i>	Ulasan pengguna Twitter terkait layanan Telkomsel	Dari analisis dengan <i>Textblob</i> saja diperoleh hasil <i>accuracy</i> sebesar 58,59%. Dengan dilakukannya optimasi dengan menggunakan SVM diperoleh hasil <i>accuracy</i> sebesar 75%.
(Khan et al., 2021)	<i>TextBlob</i> dan <i>Supervised Machine learnings</i>	Ulasan pengguna Twitter terkait Covid-19	Berdasarkan pemodelan yang dilakukan, ditemukan bahwa metode <i>gradient boosting machine</i> memiliki <i>output accuracy</i> tertinggi yakni 96%.
(Laksono et al., 2019)	<i>Textblob</i> dan <i>Naïve Bayes Classifier</i>	Ulasan pelanggan restoran Kota Surabaya	Berdasarkan pengolahan data menggunakan dua pendekatan di diperoleh nilai <i>accuracy</i> untuk <i>Naïve Bayes</i> sebesar 72,06% dan <i>TextBlob</i> sebesar 69,12%.

Sebelumnya Mishra dan Vishwakarma (2017) melakukan penelitian dengan menggabungkan *association rules* dan klasifikasi teks. Data yang menjadi objek penelitian merupakan data teks yang berhubungan dengan film, olahraga, dan teknologi. Data teks kemudian dilakukan *preprocessing* untuk selanjutnya diolah dengan *association rules*. Proses *association rules* menggunakan *software* RapidMiner dengan operator *FP-Growth*. Data kemudian dimodelkan dengan *Naïve Bayes Classifier* dan KNN. Berdasarkan hasil pengolahan data, *output* klasifikasi dengan persentase tertinggi diperoleh dari gabungan antara *FP-Growth* dan *Naïve Bayes* dengan *accuracy* 93,33%

Salloum et al. (2018) dalam penelitiannya mengimplementasikan teknik *text mining* untuk menganalisis artikel riset dengan topik *mobile learning*. Data artikel diambil dari situs yang menjadi *database* berbagai macam artikel ilmiah meliputi Springer, Wiley, Scencedirect, SAGE, IEEE, dan Cambridge. Untuk mengekstrak informasi dengan *text*

mining, terdapat beberapa teknik yang digunakan yakni *clustering*, *association rule*, *visualization*, dan *term frequency*. Berdasarkan pengolahan data yang dilakukan, diperoleh hasil kata yang paling sering digunakan dalam artikel ilmiah dengan topik penelitian *mobile learning* adalah “*learning*”, “*patients*”, “*students*”, “*education*”, “*care*”, “*mobile*”, “*university*”, “*study*”, “*clinical*”, dan “*medical*”.

Penelitian yang dilakukan oleh Afolabi *et al.* (2017) terhadap data teks yang diperoleh dari koran online terkait politik di Nigeria, menggunakan metode *association rules* Bersama dengan ontologi domain. Data diperoleh dari *supplier* koran di Nigeria yang dikenal dengan *Vanguard newspaper online*. Data yang terkumpul kemudian dilakukan *text preprocessing* meliputi penyaringan data, *stemming*, dan pembobotan dengan TF-IDF. Penyusunan ontologi kemudian dilakukan dengan memasukkan ontologi sejarah Nigeria. Hasil *association rules* menunjukkan perbandingan antara kata-kata yang diekstraksi dari ontologi dan yang tidak. Jumlah *rules* yang diperoleh dari asosiasi kata-kata yang diekstraksi dari ontologi lebih sedikit. Hal tersebut menunjukkan bahwa dengan pendekatan ontologi yang dilakukan menambah efisiensi.

Ruswati *et al.* (2018) sebelumnya melakukan riset analisis *data mining* menggunakan metode *association rules* terhadap pola kecelakaan lalu lintas. Data yang digunakan pada riset yang dilakukan bersumber dari data kecelakaan lalu lintas yang terdapat di Kota Tasikmalaya dan Ciamis. Pengolahan data menggunakan algoritma apriori dengan mencari pola dan kombinasi dari empat variabel yang ada dalam kecelakaan lalu lintas yang mendeskripsikan identitas dari pelaku meliputi jenis kelamin, usia, profesi, tingkat pendidikan, dan 22 atribut lainnya. Minimum limit, *support*, dan *confidence* yang digunakan adalah 15% dan 70%. Setelah melalui iterasi sebanyak tiga kali maka diperoleh pola asosiasi di Kota Tasikmalaya adalah antara profesi dan usia dari pelaku dengan atribut profesi siswa dan usia 16 sampai dengan 30 tahun. Sementara relasi yang ditemukan untuk Kota Ciamis adalah antara usia dan tingkat Pendidikan dengan atribut dalam dataset tingkat Pendidikan SMA dan usia 16 sampai dengan 30 tahun.

Tabel 2. 2 Kajian Induktif Metode *Association Rules*

Peneliti	Metode	Objek	Hasil Penelitian
(Mishra & Vishwakarma, 2017)	<i>Association Rules, Naïve Bayes Classifier, dan KNN</i>	Data teks berhubungan dengan film, olahraga, dan teknologi	Berdasarkan hasil pengolahan data, <i>output</i> klasifikasi dengan persentase tertinggi diperoleh dari gabungan antara <i>FP-Growth</i> dan <i>Naïve Bayes</i> dengan akurasi 93,33%
(Salloum <i>et al.</i> , 2018)	<i>Text Mining dan Association Rules</i>	Artikel penelitian	Dari pengolahan data, kata yang paling sering digunakan dalam artikel ilmiah dengan topik penelitian <i>mobile learning</i> adalah “ <i>learning</i> ” dan “ <i>patients</i> ”.
(Afolabi <i>et al.</i> , 2017)	<i>Association rules</i>	Data berita politik Nigeria	Jumlah <i>rules</i> yang diperoleh dari asosiasi kata-kata yang diekstraksi dari ontologi lebih sedikit. Hal tersebut menunjukkan bahwa dengan pendekatan ontologi yang dilakukan menambah efisiensi.
(Ruswati <i>et al.</i> , 2018)	<i>Association Rules</i> dan Algoritma Apriori	Pola dan kecelakaan lalu lintas	Hasil yang diperoleh adalah di Kota Tasikmalaya adalah antara profesi dan usia dari pelaku. Sementara relasi yang ditemukan untuk Kota Ciamis adalah antara usia dan tingkat pendidikan.

Dari penjabaran penelitian-penelitian dahulu yang memiliki topik serupa dengan penelitian dalam laporan ini, dapat diketahui bahwa teknik *text mining* banyak diterapkan dalam penelitian khususnya pada fokus penelitian klasifikasi untuk analisis sentimen. Metode yang secara mayoritas digunakan dalam klasifikasi meliputi *Naïve Bayes Classifier*, *Support Vector Machine* (SVM), *K-Nearest Neighbour* (K-NN) untuk jenis klasifikasi dengan *supervised machine learning*. Sementara metode analisis sentiment

yang banyak digunakan dengan pendekatan *lexicon-based* meliputi *Textblob* dan *VADER*. Tujuan akhir dari penelitian yang telah dikaji juga beragam mulai dari, perbandingan akurasi dan metode seperti penelitian milik Dey *et al.* (2016), Khan *et al.* (2021), dan Laksono *et al.* (2019). Selain itu terdapat beberapa penelitian dengan menggunakan metode *association rules* pada data teks meliputi penelitian milik Salloum *et al.* (2018). Sumber data yang digunakan dalam penelitian-penelitian terdahulu mayoritas mengambil sumber dari media sosial seperti Twitter, Facebook, dan Instagram. Sedangkan, data lain-lain bersumber dari *website*, data *e-mail*, dan koran online.

Pada penelitian yang dilakukan saat ini, *text mining* akan digunakan untuk menganalisis *review* dalam aplikasi. Hampir serupa dengan penelitian-penelitian sebelumnya, akan digunakan analisis sentimen untuk menentukan sentimen positif dan negatif. Hal yang membedakan penelitian ini dari penelitian terdahulu adalah pada hasil akhir yang diperoleh. Penelitian-penelitian sebelumnya hanya melakukan riset hingga tahapan analisis hasil. Seperti pada penelitian analisis sentimen analisis pelanggan toko online (Sari & Wibowo, 2019), analisis komentar Youtube (Chaithra, 2019), dan ulasan produk (Mubarak *et al.*, 2017). Sedangkan pada penelitian yang dilakukan saat ini, setelah informasi didapatkan dari analisis *text mining*, maka akan dilanjutkan dengan analisis akar masalah untuk mengetahui sumber masalah berdasarkan informasi yang didapatkan. Objek yang diteliti dalam riset ini yaitu *review* yang diberikan oleh pengguna pada *platform* pembelajaran Google Classroom, khususnya pada aplikasi ponsel. Berdasarkan analisis akar masalah yang dilakukan akan dijadikan referensi untuk perbaikan aplikasi.

2.2 Kajian Deduktif

2.2.1 E-Learning

Berdasarkan pernyataan dari Welsh *et al.* (2003), *e-learning* didefinisikan sebagai pemanfaatan jaringan internet pada komputer dengan tujuan untuk menyampaikan informasi dan instruksi kepada individu. Secara mayoritas *e-learning* yang dilakukan bersifat asinkron. Program pembelajaran secara online berfokus pada *student-centred learning* yang terbukti dapat membuat siswa pasif di kelas tatap muka, menjadi lebih partisipatif dalam pembelajaran online (Davies & Graff, 2005). Dalam *e-learning* peran guru atau pendidik sangat penting. Peran guru dapat mungkin berubah, yang awalnya

guru sebagai importir ilmu menjadi ekspeditor ilmu (Harandi, 2015). *E-learning* memiliki beberapa jenis alat yang berbeda menurut fungsinya, meliputi:

- a. *Technology Enhanced Learning*
- b. *Computer Based Instruction*
- c. *Computer Based Training*
- d. *Computer Assisted Instruction*
- e. *Internet Based Training*
- f. *Web Based Training*
- g. *Virtual Learning Environment*
- h. *M-Learning*
- i. Edukasi online
- j. Edukasi virtual

2.2.2 Learning Management System

Menurut Cavus dan Alhih (2014) *learning management system* (LMS) atau dapat disebut dengan sistem manajemen pembelajaran didefinisikan sebagai sebuah teknologi atau sistem untuk mengelola dokumen pembelajaran atau training, mendistribusikan materi melalui internet, dan melakukan *monitoring* pada pembelajaran. Dalam perkembangannya LMS memiliki beberapa istilah yang berbeda meliputi *Course Management System*, *Virtual Learning Environments*, *Collaborative Learning Environment*, dan lain lain. Banyak *platform* LMS sekarang ini telah menyediakan pengajar dan siswa yang dapat digunakan kapan saja dan dimana saja. LMS dapat membantu pengajar dalam memodifikasi materi *e-learning* dan siswa dapat memperoleh informasi yang dibutuhkan dari pengajar (Xin *et al.*, 2021).

2.2.3 Google Classroom

Google Classroom didefinisikan sebagai platform untuk pembelajaran gratis yang dikembangkan oleh Google. Sejak dirilis pada tahun 2014, Google Classroom diperkirakan memiliki pengguna yang terintegrasi dengan akun Google sejumlah 150 juta per tahun 2021. Tujuan utama dari platform ini adalah sebagai alat untuk pertukaran *file* dan informasi antara siswa dengan pendidik. Google Classroom disangka sebagai salah


satu platform yang dapat meningkatkan performa pembelajaran dari pendidik (Iftakhar, 2016).

Fitur yang dimiliki oleh Google Classroom cukup banyak. Platform ini terintegrasi dengan Google Applications for Education meliputi Google Drive, Google Sheets, Google Forms, Google Sites, dan Google Slides. Adapun fitur yang dapat membantu pendidik dan siswa dalam belajar meliputi pemberian tugas, penilaian, komunikasi antara pendidik dengan siswa, arsip kelas, aplikasi pada telepon genggam, dan laporan orisinalitas yang hanya tersedia sebagai fitur berbayar.


**Tempat kegiatan belajar mengajar
menjadi satu**

Google Classroom adalah tempat lengkap untuk kegiatan belajar mengajar Anda. Alat kami yang aman dan mudah digunakan membantu pengajar mengelola, mengukur, dan memperkaya pengalaman belajar.


[Mulai secara gratis](#)
[Buka Classroom](#)




Semuanya di satu tempat



Mudah digunakan



Cocok untuk berkolaborasi



Akses dari mana saja

Gambar 2. 1 Google Classroom

Sumber: edu.google.com (2022)

- a. Aplikasi penglihatan komputer. Terdiri dari pengenalan objek, identifikasi objek, pendeteksi wajah, *optical character recognition* (OCR), dan pengambilan data dalam bentuk gambar.
- b. Aplikasi komputasi biologi. Implementasinya dalam bentuk prediksi kegunaan protein, analisis protein, dan analisis jaringan protein.
- c. Permasalahan lain meliputi pendeteksi penipuan untuk kartu kredit, telepon, asuransi perusahaan, dan pembobolan jaringan.

2.2.4 *Text Data Mining*

Definisi dari *text data mining* menurut Rajman dan Besançon (1998) adalah sebuah teknik yang didedikasikan untuk mengekstrak data secara otomatis yang bersumber dari jenis

data tidak berstruktur. Aggarwal dan Zhai (2012) menyatakan bahwa kegunaan utama dari *text mining* adalah untuk menganalisis informasi yang hasil akhirnya akan ditemukan pola data. Implementasi dari *text mining* semakin bervariasi. Sejak awal populer digunakan pada pertengahan tahun 1980, kini *text mining* banyak diterapkan banyak bidang berbeda meliputi edukasi, sains, ekonomi, bisnis, dan media sosial (Zanini & Dhawan, 2015). Dalam bidang-bidang tersebut implementasi yang dilakukan dapat berupa *search engines*, pengelolaan hubungan dengan konsumen, penyaringan surel, analisis saran produk, mendeteksi penipuan, analisis sentimen, analisis tren, dan lain-lain (Talib *et al.*, 2016).

Pengolahan data *text mining* memiliki beberapa tahapan hingga memperoleh informasi data yang diinginkan. Berikut merupakan tahapan dari *text mining*.

a. *Data Collection*

Proses pertama dalam *text mining* adalah pengumpulan data yang dapat diperoleh dari berbagai sumber seperti *database*, arsip, atau *scraping data* yang berasal dari media sosial dan *website* (Antons *et al.*, 2020).

b. *Text Preprocessing*

Pada tahap *pre-processing*, data akan dibersihkan data dari anomali dan informasi yang tidak diperlukan (Talib *et al.*, 2016). Pembersihan data ini bertujuan agar memudahkan pengolahan data selanjutnya menggunakan algoritma. Secara umum menurut Herlinawati *et al.* (2020), *text preprocessing* terdiri dari beberapa tahapan, meliputi:

- *URL Removal* : Menghapus URL yang terdapat di dalam teks
- *Tokenization* : Pada proses *tokenizing* dilakukan pemecahan atau pemisahan pada suatu kalimat untuk menjadi kata-kata yang dibutuhkan.
- *Stemming* : *Stemming* merupakan proses penghapusan kata imbuhan depan dan belakang, menyisakan kata dasar saja. Hal ini dilakukan untuk mengurangi jumlah dari kata-kata unik.
- *Case folding* : *Case folding* adalah tahapan pada *preprocessing* teks dengan mengubah huruf yang awalnya kapital menjadi seragam huruf bukan kapital.

- *Stopwords Removal* : Menghapus kata-kata yang tidak memberikan makna dalam teks. Dalam bahasa Inggris contoh dari *stop words* meliputi *the, and, but,* dan *or*.
- *Term Frequency* : *Term frequency* adalah sebuah tahap pembobotan yang didasarkan pada jumlah kata yang memiliki frekuensi tinggi timbul di dalam data.

c. *Knowledge and Information Extraction*

Informasi yang diperoleh dari ekstraksi data teks tidak berstruktur, dalam sebagian besar kasus hanya informasi tertentu yang sebenarnya tidak dicari. Oleh karena itu, *text mining* perlu didampingi dengan teknik yang sesuai guna mengekstraksi data yang diinginkan oleh *data miner* (Salloum *et al.*, 2018). Teknik atau metode tambahan yang digunakan bergantung pada informasi, pengetahuan, atau pola data yang ingin dicari.

2.2.5 *Analisis Sentimen*

Feldman (2013) mendefinisikan analisis sentimen sebagai sebuah penugasan untuk menemukan opini dari penulis teks terkait entitas tertentu. Pembuatan keputusan oleh banyak orang, sebagian besar dipengaruhi oleh opini yang dibuat oleh orang terkemuka atau masyarakat. Sekarang ini, ketika konsumen ingin membeli suatu produk tidak lagi hanya menanyakan pendapat dari teman atau keluarga. Hal tersebut dikarenakan sekarang banyak ulasan dan diskusi terkait produk yang tersedia di internet (Zhang *et al.*, 2018). Aplikasi dari analisis sentimen sangat beragam mulai dari politik, pasar, bisnis, kesehatan, sosial, dan lain-lain.

2.2.6 *Machine Learning*

Menurut Mohri *et al.* (2018), *machine learning* didefinisikan sebagai metode komputasi yang memanfaatkan hal yang telah terjadi untuk meningkatkan performansi atau membuat prediksi yang akurat. hal yang sudah terjadi tersebut adalah informasi terdahulu dalam bentuk data elektronik yang kemudian dibuat agar dapat dianalisis. Berbagai algoritma untuk *machine learning* telah dikembangkan untuk mencakup banyak variasi data dan masalah untuk diterapkan dalam *machine learning* (Jordan & Mitchell, 2015).

Machine learning memiliki implementasi yang luas dalam praktik. Berikut merupakan implementasi dari *machine learning* menurut Mohri *et al.* (2018).

- a. Klasifikasi teks atau dokumen. Implementasi *machine learning* ini meliputi masalah dalam menugaskan suatu topik ke dalam teks atau dokumen, secara otomatis menentukan apakah konten di dalam suatu *website* tidak pantas atau terlalu eksplisit, dan mendeteksi spam.
- b. *Natural language processing* (NLP). Aplikasi dalam hal ini meliputi *speech tagging*, pengenalan nama entitas, *context-free passing*, dan lain-lain.
- c. Aplikasi pemrosesan pembicaraan. Implementasinya dalam hal pengenalan suara atau pembicaraan, sintesis pembicaraan, identifikasi pembicara, dan pemodelan bahasa.

2.2.7 *Klasifikasi Sentimen*

Klasifikasi sentimen merupakan komponen yang ada di dalam analisis sentimen yang digunakan untuk menganalisis polaritas pada suatu pernyataan ketika membaca data teks (Sunghetha & Sharma, 2020). Menurut Chaithra (2019), teknik dalam klasifikasi sentimen pada data teks terbagi ke dalam tiga kategori yakni pendekatan *machine learning*, pendekatan *lexicon based*, dan pendekatan *hybrid*. Adapun uraian dari setiap teknik adalah sebagai berikut.

- a. Pendekatan *machine learning*

Pada pendekatan klasifikasi sentimen dengan *machine learning* dibagi lagi ke dalam dua jenis yakni *unsupervised learning* dan *supervised learning*. Pada algoritma *unsupervised learning*, klasifikasi dapat dijalankan tanpa data latih dan tidak diperlukan proses pelabelan sentimen. Contoh dari algoritma klasifikasi untuk *unsupervised learning* adalah *K-means Clustering*. Sementara itu, *supervised learning* membutuhkan data latih yang telah diberi label sentimen. Beberapa metode klasifikasi yang tergolong ke dalam *supervised learning* meliputi *Naïve Bayes Classifier*, *Support Vector Machine*, *K-Nearest Neighbor Classification*, *Decision Tree*, dan *Random Forest Algorithm* (Nofriani, 2019).

- b. Pendekatan *lexicon-based*

Klasifikasi dengan pendekatan *lexicon-based* juga terbagi ke dalam dua jenis, yakni pendekatan *dictionary based* dan *corpus based*.

c. Pendekatan *hybrid*

Pendekatan *hybrid* merupakan teknik untuk melakukan klasifikasi sentimen dengan memadukan pendekatan *machine learning* dan *lexicon-based*.

2.2.8 *TextBlob*

Textblob merupakan salah satu *library* dalam *Python* untuk pemrosesan data bentuk teks dan menggunakan NLTK untuk *Natural Language Processing* (NLP) yang bersifat *unlabelled* (Kaur & Sharma, 2020). *Textblob* secara optimal digunakan untuk menganalisis sentimen pada data teks dengan bahasa formal. Dalam pengolahan analisis sentimen *Textblob* menggunakan dua nilai numerik untuk menentukan jenis sentimen, yakni *subjectivity* dan *polarity*. *Subjectivity* merupakan nilai dari opini bersifat subjektif yang ada di dalam sebuah data teks. Skor untuk *subjectivity* ada pada kisaran 0 sampai dengan 1. Skor 0 menunjukkan tingkat opini yang rendah di dalam suatu data teks dan sebaliknya. *Polarity* adalah variabel yang akan menentukan sentimen suatu data teks meliputi sentimen positif, negatif, dan netral. Skor untuk *polarity* berkisar antara -1 sampai dengan 1. Skor diidentifikasi sebagai data teks dengan sentimen negatif yang tinggi dan sebaliknya.

2.2.9 *Naïve Bayes Classifier*

Metode klasifikasi *Naïve Bayes* yakni teknik probabilitas dan statistika yang diusung oleh ilmuwan asal Inggris, Thomas Bayes (Lim *et al.*, 2016). Teknik ini dapat memproyeksikan kemungkinan yang dapat terjadi di masa depan berdasarkan pengalaman yang telah terjadi, yang sekarang dikenal dengan *Bayes' Theorem*. Menurut Wibawa *et al.* (2019), *Naïve Bayes Classifier* dikenal sebagai algoritma klasifikasi probabilitas yang sederhana. Meskipun dikenal dengan teknik sederhana, *Naïve Bayes Classifier* juga dapat bekerja dengan baik dalam aplikasi dunia nyata yang kompleks, seperti klasifikasi teks (Xu, 2018). Selain itu, *Naïve Bayes Classifier* hanya memerlukan jumlah data latih yang sedikit untuk proses klasifikasi. Bahkan, algoritma klasifikasi *Naïve Bayes* masuk ke dalam sepuluh besar algoritma dalam *data mining* (Wu *et al.*, 2008). *Naïve Bayes Classifier* terbagi ke dalam tiga tipe meliputi:

a. *Bernoulli Naïve Bayes*

Model *Bernoulli* menggunakan dokumen dengan vektor *binary* (McCallum & Nigam, 1998). Tipe *Bernoulli* menurut Singh *et al.* (2019) diterapkan pada konsep *binary* yang menunjukkan ada tidaknya suatu istilah atau kata di dalam dokumen. Pada model klasifikasi teks, terdapat dua *value* yakni 0 dan 1. *Value* 0 menunjukkan bahwa kata tersebut tidak ditemukan dalam dokumen, sementara *value* 1 mempresentasikan bahwa kata tersebut terdapat dalam dokumen.

b. *Multinomial Naïve Bayes*

Menurut McCallum dan Nigam (1998), *Naïve Bayes* dengan model *multinomial* merupakan metode *supervised learning* probabilitas. Metode ini mengikuti prinsip dari distribusi *multinomial* dalam probabilitas bersyarat untuk data dalam bentuk diskrit (Awwalu *et al.*, 2020). Pada kasus *text learning* model ini dapat menemukan informasi berupa banyaknya frekuensi dari sebuah data di dalam dokumen. Berbeda dengan model *Bernoulli* yang hanya memberikan informasi bahwa suatu kata terdapat dalam data tersebut.

c. *Gaussian Naïve Bayes*

Menurut Bustamante *et al.* (2006) *Naïve bayes classifier* dengan model *Gaussian* menggunakan data yang memiliki distribusi kontinu dan normal (*gaussian*).

2.2.10 Evaluasi Klasifikasi

Evaluasi dari klasifikasi dapat dilakukan dengan banyak cara. Jenis evaluasi yang paling banyak digunakan adalah dengan menghitung nilai *accuracy*, *precision*, *recall*, dan *f-measure* (Ratnawati, 2018). Perhitungan akurasi dapat dilakukan dengan *confusion matrix* yang merupakan teknik untuk mengukur kinerja suatu metode klasifikasi (Emrald & Lhaksmana, 2019). Nilai akurasi menunjukkan seberapa akurat sebuah model dalam melakukan prediksi klasifikasi positif dan negatif. Untuk melakukan uji *accuracy* dengan *confusion matrix* yang ditunjukkan pada persamaan (1).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (2.1)$$

Keterangan

TP (*True Positive*) : Jumlah data yang diprediksi positif

TN (*True Negative*) : Jumlah data yang diprediksi negatif

FP (*False Negative*) : Jumlah data yang negatif namun terprediksi positif

FN (*False Positive*) : Jumlah data yang positif namun terprediksi negatif

Precision adalah perbandingan jumlah data yang memiliki *value* positif dengan jumlah data salah yang bernilai positif.

$$Precision = \frac{TP}{TP+FP} \quad (2.2)$$

Recall adalah perbandingan jumlah data benar berlabel *value* positif dengan hasil penjumlahan dari data benar dengan label *value* positif dan data salah dengan *value* negatif.

$$Recall = \frac{TP}{TP+FN} \quad (2.3)$$

F1 score digunakan untuk mengukur kesuksesan *retrieval* yang menggabungkan *recall* dan *precision*. *F1 score* menunjukkan kemungkinan kesalahan prediksi pada suatu model. Nilai dari *f1 score* berkisar antara 1 sampai dengan 0, semakin tinggi nilai *f1 score* maka semakin baik.

$$f1\ score = 2 \times \frac{precision \times recall}{precision + recall} \quad (2.4)$$

2.2.11 Association Rules

Association rules mining menurut Kotsiantis dan Kanellopoulos (2006) bertujuan untuk mengekstraksi korelasi yang unik, pola yang sering terjadi, asosiasi, atau struktur sederhana dari data set dalam *database* atau *repository*. *Association rules* biasanya dibutuhkan untuk memuaskan ketentuan minimum *support* dan ketentuan minimum *confidence* dalam waktu yang bersamaan (I. A. Khan et al., 2015). Perhitungan dalam *association rules* terdiri dari tiga perhitungan dasar untuk menentukan nilai *support*, *confidence*, dan *lift ratio*. Adapun rumus untuk setiap perhitungannya adalah sebagai berikut.

1. Support

Nilai *support* dari sebuah item:

$$Support(A) = \frac{Jumlah\ transaksi\ yang\ mengandung\ A}{Total\ transaksi} \times 100 \quad (2.5)$$

Nilai *support* yang merupakan kombinasi dari dua item:

$$Support(A \cap B) = \frac{Jumlah\ transaksi\ yang\ mengandung\ A\ dan\ B}{Total\ transaksi} \times 100 \quad (2.6)$$

2. Confidence

Nilai *confidence* dari $A \rightarrow B$:

$$Confidence = \frac{Jumlah\ transaksi\ yang\ mengandung\ A\ dan\ B}{Jumlah\ transaksi\ yang\ mengandung\ A} \times 100 \quad (2.7)$$

3. Lift Ratio

Sebelum menghitung *lift ratio*, nilai ekspektasi *confidence* telah didapatkan dari formula berikut:

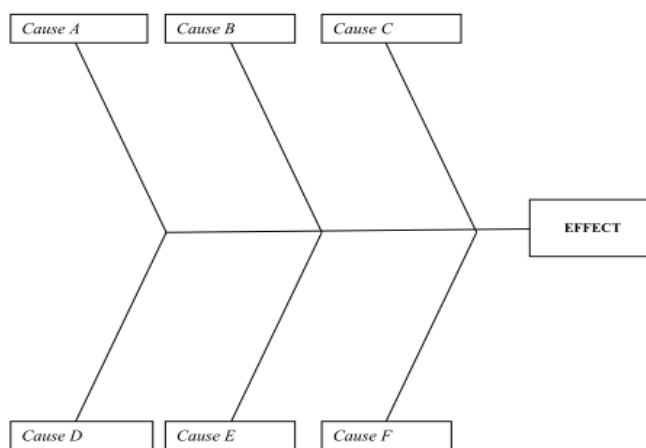
$$\text{Ekspektasi Confidence} = \frac{\text{Jumlah transaksi yang mengandung B}}{\text{Total kejadian}} \quad (2.8)$$

Hasil dari perhitungan dari nilai *ekspektasi confidence* akan digunakan sebagai perbandingan dengan nilai *confidence* yang telah dihitung.

$$\text{Lift ratio} = \frac{\text{Confidence}}{\text{Ekspektasi confidence}} \quad (2.9)$$

2.2.12 Fishbone Diagram

Fishbone diagram atau yang sering juga disebut dengan *ishikawa diagram* adalah sebuah alat yang berguna untuk mengidentifikasi akar penyebab dari masalah yang berhubungan dengan pengendalian kualitas (Ilie & Ciocoiu, 2010). Metode ini ditemukan oleh Kaoru Ishikawa pada tahun 1990, dengan tujuan untuk meningkatkan performa tim dalam menganalisis sebab dan akibat dari suatu masalah. Oleh karena itu, *fishbone diagram* juga dapat disebut dengan *cause and effect diagram* (Watson, 2004). Diagram ini memiliki bentuk seperti tulang ikan, dengan bagian kepala diinterpretasikan sebagai *effect* atau masalah. Sedangkan tulang ikan digambarkan sebagai kemungkinan penyebab dari masalah.



Gambar 2. 2 *Fishbone diagram*

Sumber: (Coccia, 2017)

BAB III

METODE PENELITIAN

3.1 Jenis Data

Data yang diolah dalam riset ini merupakan data jenis sekunder. Data sekunder merupakan data yang didapatkan bukan dari pengambilan data secara langsung, melainkan dari perantara. Dalam penelitian ini data sekunder yang digunakan sebagai pendukung bersumber dari literasi dan riset-riset yang sebelumnya telah dilakukan dengan mengangkat topik yang sama dengan penelitian yang dilaksanakan saat ini. Selain itu, digunakan data review pengguna aplikasi Google Classroom yang diambil dari Google Play Store.

3.2 Pengumpulan Data

Data yang diambil untuk selanjutnya diolah dalam penelitian ini bervariasi dari beberapa sumber yakni sebagai berikut:

1. *Web scraping*, merupakan teknik pengambilan data dalam jumlah besar yang bersumber dari situs web. Dalam penelitian ini *web scraping review* pengguna Google Classroom dilakukan melalui situs *website* Appfollow. Kriteria pengambilan data adalah *review* dari pengguna Google Classroom adalah pengguna *device* dan *review* bahasa Inggris.
2. Studi Literatur, adalah cara pengumpulan data yang dapat menunjang penelitian dengan menelusuri literatur-literatur yang berhubungan dengan penelitian yang sedang dilakukan. Sumber dari literatur yang digunakan dapat berasal dari buku, jurnal, internet, tesis, dan lain-lain. Studi pustaka yang digunakan dalam penelitian ini meliputi *text mining*, analisis sentimen, *association rules*, *fishbone diagram*, dan penelitian terkait.

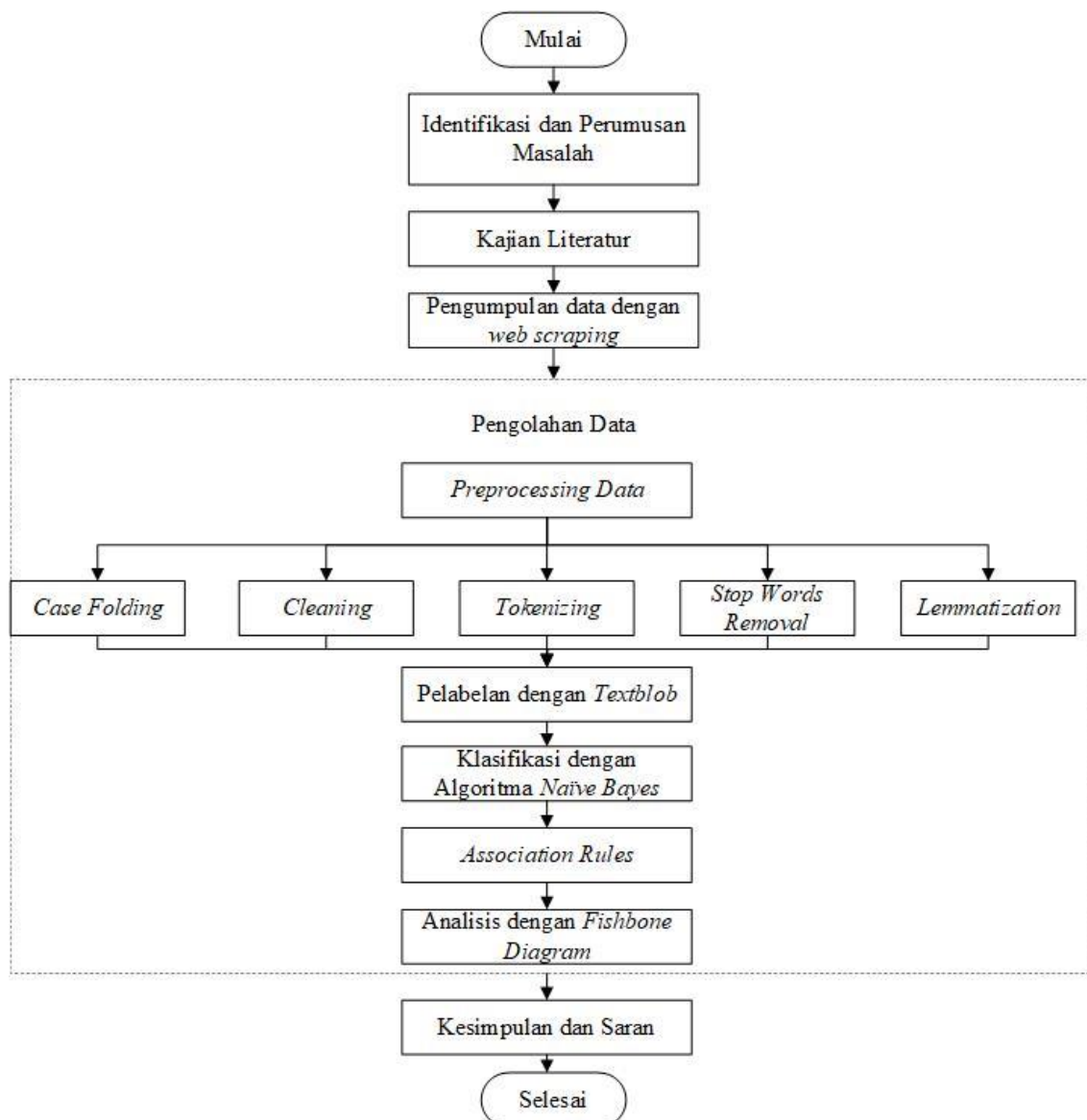
3.3 Pengolahan Data

Tahapan sistematis pengolahan data yang dilaksanakan pada penelitian ini adalah sebagai berikut:

1. *Labelling review* pengguna Google Classroom menjadi label sentimen positif dan negatif. Pelabelan dilakukan menggunakan *library Python Textblob*.
2. *Preprocessing data* dilakukan untuk guna membersihkan dan meminimalisir variasi item kata yang ada pada data, sehingga mesin dapat mempelajari data dengan mudah. Pada penelitian ini, *preprocessing data* yang dilakukan meliputi *case folding, cleaning data, tokenizing, stop words removal, dan lemmatization*.
3. Analisis sentimen dilakukan dengan mengaplikasikan algoritma *Naïve Bayes Classifier*. Untuk menemukan nilai *accuracy* dari analisis sentimen maka dilakukan evaluasi untuk mengetahui *accuracy, precision, dan recall*.
4. Pengolahan data dengan *association rules* digunakan untuk mencari hubungan antar kata berdasarkan *review* positif dan negatif.
5. Analisis dengan *fishbone* diagram dipakai untuk menemukan sumber dari inti permasalahan yang ditemukan dari *association rules*.

3.4 Alur Penelitian

Proses yang dilakukan dalam penelitian ini ditunjukkan alur penelitian pada gambar 3.1 sebagai berikut.



Gambar 3. 1 Diagram Alir Penelitian

Berikut merupakan uraian dari dari alur penelitian pada gambar 3.1:

1. Identifikasi dan Perumusan Masalah

Tahap pertama dalam penelitian ini adalah mengidentifikasi masalah yang ada pada aplikasi Google Classroom meliputi *rating* yang diberikan pada aplikasi oleh pengguna, gambaran umum *review* yang diberikan pengguna, dan kenaikan jumlah pengguna dari aplikasi. Berdasarkan identifikasi masalah yang dilakukan, maka dapat disusun rumusan masalah yang kemudian akan diteliti.

2. Kajian literatur

Setelah merumuskan masalah, penulis melakukan kajian literatur dari penelitian terdahulu yang memiliki korelasi dan dapat digunakan sebagai pendukung dengan penelitian yang dilakukan saat ini seperti topik *text mining*, analisis sentimen, *association rules*, dan *fishbone diagram*.

3. Pengumpulan data

Pengumpulan data sekunder berupa *review* pengguna aplikasi Google Classroom dilakukan dengan cara *web scraping* pada Google Play melalui situs AppFollow.

4. Labelling

Pelabelan *review* dilakukan untuk menentukan sentimen positif dan negatif dari suatu *review* pengguna. Dalam penelitian ini, pelabelan *review* dilakukan menggunakan *library Python TextBlob*,

5. Preprocessing Data

Tahap *preprocessing data* dilakukan untuk membersihkan dan menyiapkan data yang kemudian akan diolah, sehingga mesin akan lebih mudah untuk mengenali data. Adapun tahapan *preprocessing data* yang dilakukan pada penelitian ini adalah sebagai berikut:

a. Case Folding

Case folding merupakan tahapan untuk menyeragamkan seluruh huruf, sehingga huruf yang semula kapital menjadi huruf kecil.

b. Cleaning Data

Cleaning data yang dilakukan meliputi tahapan menghilangkan tanda baca, karakter non-ASCII, dan emoji di dalam data teks.

c. Tokenizing

Tokenizing adalah tahap *preprocessing* untuk memisahkan dalam suatu kalimat, sehingga hanya menyisakan kata-kata yang dibutuhkan saja.

d. Stop Words Removal

Proses *stopwords removal* yang digunakan untuk menghilangkan kata-kata yang tidak bermakna meliputi *the*, *and*, *but*, dan *or*, dan lain-lain.

e. Lemmatization

Merupakan proses untuk mengembalikan suatu kata menjadi bentuk dasarnya.

6. Klasifikasi dengan Algoritma *Naïve Bayes*

Analisis sentimen dalam penelitian ini dilakukan dengan algoritma *Naïve Bayes Classifier*. Klasifikasi ini dilakukan untuk mengetahui model analisis sentimen positif dan negatif berdasarkan *review* pengguna aplikasi Google Classroom beserta *accuracy* dari model. Klasifikasi dilakukan dengan bantuan Jupyter Notebook dengan bahasa pemrograman Python 3.9.

7. *Association Rules*

Association rules digunakan untuk menemukan aturan hubungan atau asosiasi antar satu kata dengan kata lainnya yang paling sering muncul di dalam data. Pengolahan data *association rules* menggunakan bantuan RapidMiner Studio 9.10.

8. Analisis dengan *Fishbone Diagram*

Berdasarkan hasil *association rules*, maka dilanjutkan dengan analisis menggunakan *fishbone diagram* untuk mencari akar dari permasalahan yang ditemukan. Dengan analisis akar masalah yang dilakukan dapat diperoleh referensi perbaikan untuk aplikasi Google Classroom.

9. Kesimpulan dan Saran

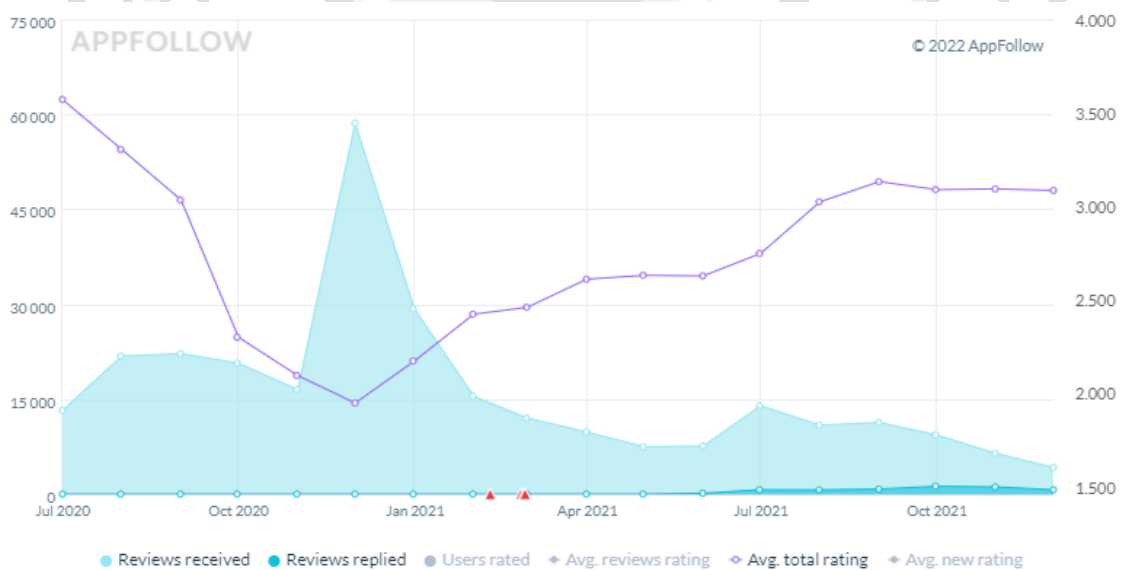
Berdasarkan hasil pengolahan data yang telah dilakukan, maka dapat diambil kesimpulan dan saran yang dapat membantu peneliti selanjutnya dengan topik yang serupa.

BAB IV

PENGUMPULAN DAN PENGOLAHAN DATA

4.1 Pengumpulan Data

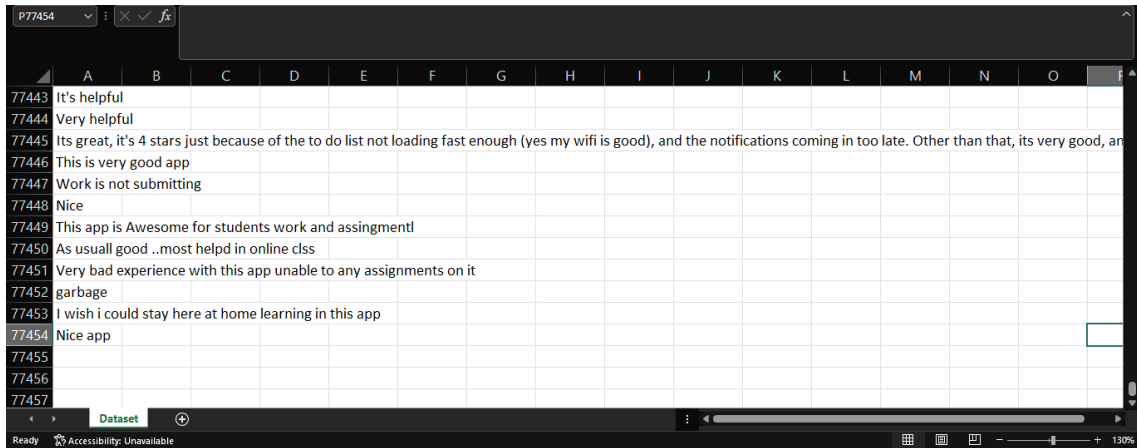
Data yang dijadikan sebagai bahan utama dalam penelitian ini merupakan data sekunder yang berasal dari *review* dan *rating* yang diberikan oleh pengguna aplikasi Google Classroom melalui platform Google Play. Proses pengambilan data yang dilakukan adalah dengan cara *web scraping* melalui *website* untuk AppFollow. Berdasarkan infografis dari *website* AppFollow, diperoleh grafik *review* dan *rating* yang diberikan pengguna ditunjukkan pada gambar 4.1.



Gambar 4. 1 Grafik Jumlah *Review* dan *Rating* Pengguna Google Classroom

Sumber: AppFollow.io (2022)

Berdasarkan grafik tersebut pengguna paling banyak memberikan *review* pada akhir tahun 2020 hingga awal tahun 2021 bersama dengan turunnya *rating* aplikasi Google Play secara signifikan. Oleh karena itu, pada penelitian ini digunakan sampel data dari tanggal 1 Desember 2020 sampai dengan tanggal 31 Juli 2021. Kriteria data yang akan diambil adalah pengguna yang menggunakan *device* dan memberikan *review* dalam bahasa Inggris. Dari pengumpulan data melalui AppFollow diperoleh *review* pengguna sebanyak 77.454 data.



ID	Review
77443	It's helpful
77444	Very helpful
77445	Its great, it's 4 stars just because of the to do list not loading fast enough (yes my wifi is good), and the notifications coming in too late. Other than that, its very good, an
77446	This is very good app
77447	Work is not submitting
77448	Nice
77449	This app is Awesome for students work and assingmentl
77450	As usuall good ..most helpd in online clss
77451	Very bad experience with this app unable to any assignments on it
77452	garbage
77453	I wish i could stay here at home learning in this app
77454	Nice app
77455	
77456	
77457	

Gambar 4. 2 Hasil Pengumpulan Data

4.2 Pengolahan Data

Data yang berhasil terkumpul kemudian diolah melalui beberapa tahapan untuk memperoleh hasil akhir dari penelitian ini. Adapun proses pengolahan data yang dilakukan adalah sebagai berikut.

4.2.1 *Preprocessing Data*

Tahap *preprocessing data* dilakukan untuk membersihkan data dari informasi dan anomali yang tidak dibutuhkan. Proses ini dilakukan dengan menggunakan bantuan Jupyter Notebook dan Python 3.9. Sebelum dilakukan *preprocessing data*, dilakukan impor modul dan pemanggilan data dengan format csv yang ditunjukkan pada gambar 4.6.

a. *Case Folding*

Tahap pertama dalam *preprocessing data* adalah *case folding*, yakni mengubah seluruh huruf kapital yang ada di dalam data menjadi *lowercase*. Modul yang digunakan adalah Python RegEx atau *Regular Expression*. Proses *case folding* ditunjukkan pada tabel 4.1.

Tabel 4. 1 Proses *Case Folding*

Sebelum <i>Case Folding</i>	Sesudah <i>Case Folding</i>
<i>This app is mostly like trash</i>	<i>this app is mostly like trash</i>
<i>ðŸŒˆ™€i, I've been using this app</i>	<i>ðŸŒˆ™€i, i've been using this app</i>
<i>for a while and i changed my acc</i>	<i>for a while and i changed my acc</i>
<i>password, and just like that, i can't log</i>	<i>password, and just like that, i can't log</i>

<p><i>out and whenever I use it, it will tell me something went wrong and wants me to try again...So I decided to reinstall this app, and now it's telling me I don't have to access the google classroom</i></p> <p><i>ðÿð â€â™€i, ðÿ~'</i> And also, it really takes my time when I try to upload photos or videos, please fix this asap 🙏🙏</p>	<p><i>out and whenever i use it, it will tell me something went wrong and wants me to try again...so i decided to reinstall this app, and now it's telling me i don't have to access the google classroom</i></p> <p><i>ðÿð â€â™€i, ðÿ~'</i> and also, it really takes my time when i try to upload photos or videos, please fix this asap 🙏🙏</p>
---	---

b. *Cleaning Data*

Sebelum dilakukan proses lebih lanjut, data perlu dibersihkan dari hal-hal yang tidak dibutuhkan, meliputi tanda baca (*punctuation*), karakter non-ASCII, dan emoji.

Tahap *remove punctuation* dilakukan untuk menghilangkan simbol atau tanda baca yang tidak diinginkan di dalam data teks. Adapun proses penghapusan tanda baca dan simbol ditunjukkan pada tabel 4.2.

Tabel 4. 2 Proses *Remove Punctuation*

Sebelum <i>Remove Punctuation</i>	Sesudah <i>Remove Punctuation</i>
<p><i>this app is mostly like trash</i></p> <p><i>ðÿð â€â™€i, i've been using this app for a while and i changed my acc password, and just like that, i can't log out and whenever i use it, it will tell me something went wrong and wants me to try again...so i decided to reinstall this app, and now it's telling me i don't have to access the google classroom</i></p> <p><i>ðÿð â€â™€i, ðÿ~'</i> and also, it really takes my time when i try to upload</p>	<p><i>this app is mostly like trash</i></p> <p><i>ðÿð â€â™€i, ive been using this app for a while and i changed my acc password and just like that i cant log out and whenever i use it it will tell me something went wrong and wants me to try again so i decided to reinstall this app and now its telling me i dont have to access the google classroom</i></p> <p><i>ðÿð â€â™€i, ðÿ~'</i> and also it really takes my time when i try to upload</p>

photos or videos, please fix this photos or videos please fix this
asap 🌸🌸 *asap 🌸🌸*

Proses selanjutnya adalah pembersihan data yang mengandung karakter non-ASCII. ASCII merupakan karakter standar *encoding* untuk komunikasi elektronik. Jadi, karakter non-ASCII merupakan karakter yang tidak dikodekan oleh ASCII meliputi *Unicode*, *ebscic*, dan lain-lain. Adapun perbedaan hasil antara sebelum dan sesudah pembersihan karakter non-ASCII ditunjukkan pada tabel 4.3.

Tabel 4. 3 Proses ASCII Character Removal

Sebelum Non-ASCII Character Removal	Sesudah Non-ASCII Character Removal
<p><i>this app is mostly like trash</i> <i>ðŸŒˆ™€i, ive been using this app</i> <i>for a while and i changed my acc</i> <i>password and just like that i cant log</i> <i>out and whenever i use it it will tell me</i> <i>something went wrong and wants me to</i> <i>try again so i decided to reinstall this</i> <i>app and now its telling me i dont have</i> <i>to access the google classroom</i> <i>ðŸŒˆ™€i, ðŸ˜ and also it really takes</i> <i>my time when i try to upload photos or</i> <i>videos please fix this asap 🌸🌸</i></p>	<p><i>this app is mostly like trash ive been</i> <i>using this app for a while and i</i> <i>changed my acc password and just</i> <i>like that i cant log out and whenever i</i> <i>use it it will tell me something went</i> <i>wrong and wants me to try again so i</i> <i>decided to reinstall this app and now</i> <i>its telling me i dont have to access the</i> <i>google classroom and also it really</i> <i>takes my time when i try to upload</i> <i>photos or videos please fix this</i> <i>asap 🌸🌸</i></p>

Tahap selanjutnya dalam pembersihan data adalah penghapusan emoji menggunakan modul emoji yang ada dalam Python. Proses penghapusan emoji ditunjukkan pada tabel 4.4.

Tabel 4. 4 Proses *Emoji Removal*

Sebelum <i>Emoji Removal</i>	Sesudah <i>Emoji Removal</i>
<i>this app is mostly like trash ive been using this app for a while and i changed my acc password and just like that i cant log out and whenever i use it it will tell me something went wrong and wants me to try again so i decided to reinstall this app and now its telling me i dont have to access the google classroom and also it really takes my time when i try to upload photos or videos please fix this asap 🙏🙏</i>	<i>this app is mostly like trash ive been using this app for a while and i changed my acc password and just like that i cant log out and whenever i use it it will tell me something went wrong and wants me to try again so i decided to reinstall this app and now its telling me i dont have to access the google classroom and also it really takes my time when i try to upload photos or videos please fix this asap</i>

c. **Tokenizing**

Tahap selanjutnya adalah *tokenizing* yakni membuat setiap kata yang ada di dalam kalimat menjadi sekumpulan kata tiap token. Hal ini dilakukan agar pada tahap penghapusan *stop words*, mesin dapat membaca tiap kata yang ada di dalam teks. Proses dan hasil *tokenizing* ditunjukkan pada tabel 4.5.

Tabel 4. 5 Proses *Tokenizing*

Sebelum <i>Tokenizing</i>	Sesudah <i>Tokenizing</i>
<i>this app is mostly like trash ive been using this app for a while and i changed my acc password and just like that i cant log out and whenever i use it it will tell me something went wrong and wants me to try again so i decided to reinstall this app and now its telling me i dont have to access the google classroom and also it really takes my</i>	<i>['this', 'app', 'is', 'mostly', 'like', 'trash', 'ive', 'been', 'using', 'this', 'app', 'for', 'a', 'while', 'and', 'i', 'changed', 'my', 'acc', 'password', 'and', 'just', 'like', 'that', 'i', 'cant', 'log', 'out', 'and', 'whenever', 'use', 'it', 'it', 'will', 'tell', 'me', 'something', 'went', 'wrong', 'and', 'wants', 'me', 'to', 'try', 'again', 'so', 'i', 'decided', 'to', 'reinstall', 'this',</i>

time when i try to upload photos or videos please fix this asap

['app', 'and', 'now', 'its', 'telling', 'me', 'i', 'dont', 'have', 'to', 'access', 'the', 'google', 'classroom', 'and', 'also', 'it', 'really', 'takes', 'my', 'time', 'when', 'i', 'try', 'to', 'upload', 'photos', 'or', 'videos', 'please', 'fix', 'this', 'asap']

d. **Stopwords Removal**

Tahap selanjutnya pada *preprocessing data* adalah untuk menghapus *stopwords* yang ada di dalam data teks. Dalam proses ini digunakan *stop words* yang ada di dalam *library Natural Language Toolkit (NLTK)* untuk bahasa Inggris. Adapun hasil dari penghapusan *stop words* ditunjukkan pada tabel 4.6.

Tabel 4. 6 Proses *Stopwords Removal*

Sebelum <i>Stopwords Removal</i>	Sesudah <i>Stopwords Removal</i>
['this', 'app', 'is', 'mostly', 'like', 'trash', 'ive', 'been', 'using', 'this', 'app', 'for', 'a', 'while', 'and', 'i', 'changed', 'my', 'acc', 'password', 'and', 'just', 'like', 'that', 'i', 'cant', 'log', 'out', 'and', 'whenever', 'use', 'it', 'it', 'will', 'tell', 'me', 'something', 'went', 'wrong', 'and', 'wants', 'me', 'to', 'try', 'again', 'so', 'i', 'decided', 'to', 'reinstall', 'this', 'app', 'and', 'now', 'its', 'telling', 'me', 'i', 'dont', 'have', 'to', 'access', 'the', 'google', 'classroom', 'and', 'also', 'it', 'really', 'takes', 'my', 'time', 'when', 'i', 'try', 'to', 'upload',	['app', 'mostly', 'like', 'trash', 'using', 'app', 'changed', 'acc', 'password', 'like', 'log', 'whenever', 'use', 'tell', 'something', 'went', 'wrong', 'wants', 'try', 'again', 'decided', 'reinstall', 'app', 'telling', 'access', 'google', 'classroom', 'also', 'really', 'takes', 'time', 'try', 'upload', 'photos', 'videos', 'please', 'fix', 'asap']

'photos', 'or', 'videos', 'please', 'fix',
'this', 'asap']

e. Lemmatization

Tahap *preprocessing* kemudian dilanjutkan dengan *lemmatization* yakni proses menemukan kata dasar dari suatu kata. Pada proses ini digunakan *library* Python untuk *Natural Language Processing* (NLP) yakni *spaCy*. Pada *library* ini juga dapat dilakukan pemrosesan mengubah *verb past tense* menjadi *present tense*. Adapun hasil dari proses *lemmatization* adalah sebagai berikut

Tabel 4. 7 Proses Lemmatization

Sebelum Lemmatization	Sesudah Lemmatization
['app', 'mostly', 'like', 'trash', 'using', 'app', 'changed', 'acc', 'password', 'like', 'log', 'whenever', 'use', 'tell', 'something', 'went', 'wrong', 'wants', 'try', 'again', 'decided', 'reinstall', 'app', 'telling', 'access', 'google', 'classroom', 'also', 'really', 'takes', 'time', 'try', 'upload', 'photos', 'videos', 'please', 'fix', 'asap']	['app', 'mostly', 'like', 'trash', 'use', 'app', 'change', 'acc', 'password', 'like', 'can', 'not', 'log', 'whenever', 'use', 'tell', 'something', 'go', 'wrong', 'want', 'try', 'againso', 'decide', 'reinstall', 'app', 'tell', 'not', 'access', 'google', 'classroom', 'also', 'really', 'take', 'time', 'try', 'upload', 'photo', 'video', 'please', 'fix', 'asap']

4.2.2 Pelabelan Data

Tahap pelabelan data diperlukan dalam proses analisis sentimen karena algoritma yang digunakan untuk pemodelan klasifikasi *naïve bayes* merupakan jenis *supervised learning*. Sehingga, data yang digunakan dalam *training data* perlu diberikan label terlebih dahulu. Pelabelan yang digunakan dalam penelitian ini adalah *library Textblob*.

Textblob adalah salah satu *library Python* untuk melakukan analisis sentimen dengan pendekatan *lexicon based*. Hasil dari analisis sentimen dengan *Textblob* akan memberikan *output* nilai *subjectivity* dan *polarity*. Nilai *polarity* dijadikan sebagai acuan untuk menentukan sentimen suatu data teks masuk ke dalam kategori negatif, netral, atau

positif. Kriteria pelabelan pada *Textblob* merujuk kepada skor *polarity* dengan ketentuan sebagai berikut (Shah, 2020):

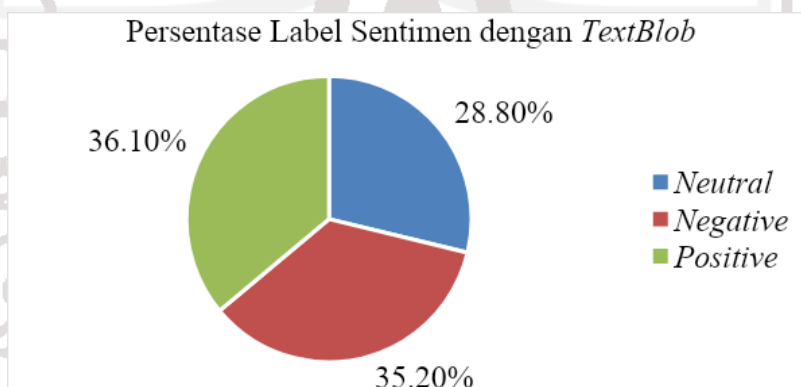
- Skor *polarity* < 0 , maka sentimen negatif
- Skor *polarity* $= 0$, maka sentimen netral
- Skor *polarity* > 0 , maka sentimen positif

Berikut merupakan contoh dari penerapan pelabelan sentimen menggunakan *Textblob* yang ditunjukkan pada tabel 4.7.

Tabel 4. 8 Hasil Pelabelan dengan Textblob

<i>Review</i>	<i>Subjectivity</i>	<i>Polarity</i>	<i>Sentiment</i>
<i>horrible experience can barely use this app without something messing up</i>	0.55	-0.475	<i>Negative</i>
<i>i still don't know what the function of this app is</i>	0	0	<i>Neutral</i>
<i>very useful indeed thank you google classroom</i>	0.475	0.275	<i>Positive</i>

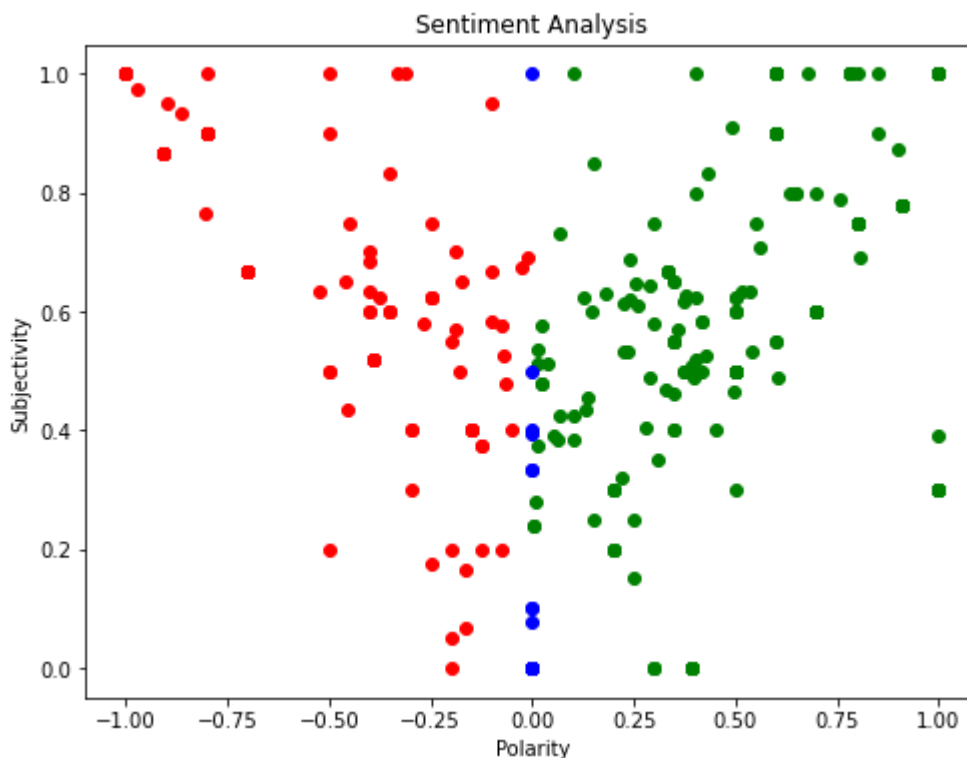
Berdasarkan hasil pelabelan sentimen dengan menggunakan *library Textblob*, maka diperoleh pembagian label sentimen negatif sebanyak 27.233 data, label sentimen positif sebanyak 27.931 data, dan label sentimen netral sebanyak 22.290 data. Adapun visualisasi persentase hasil pelabelan sentimen ditunjukkan pada grafik *pie chart* yang berada di gambar 4.3.



Gambar 4. 3 Visualisasi Hasil Pelabelan Sentimen dengan *Textblob*

Kemudian berdasarkan data hasil pelabelan menggunakan *Textblob*, dibuat *scatter plot* untuk mengetahui hubungan antara *subjectivity* dan *polarity* dalam menghitung label

sentimen. Pembuatan *scatter plot* menggunakan sampel sebanyak 500 data dengan *subjectivity* sebagai sumbu x dan *polarity* sebagai sumbu y. Adapun hasil dari *scatter plot* ditunjukkan pada gambar 4.4.



Gambar 4. 4 Scatter Plot Pelabelan Texblob

4.2.3 Klasifikasi Menggunakan Algoritma *Naïve Bayes*

Tahap berikutnya yaitu pemodelan dengan klasifikasi algoritma *Naïve Bayes*. Klasifikasi dengan algoritma *Naïve Bayes* dilakukan untuk mengetahui *accuracy* mesin dalam memprediksi analisis sentimen. Algoritma *Naïve Bayes Classifier* merupakan model *machine learning* yang bersifat *supervised* atau perlu dilatih. Oleh karena itu, untuk menjalankan model klasifikasi dibutuhkan data latih dan data uji. Pembagian data dilakukan dengan *splitting data* menggunakan library '*train_test_split*' dari *sklearn* yang ada di Python. Secara umum, pembagian data latih dan data uji meliputi 50/50, 60/40, 70/30 dan 80/20 (Rácz *et al.*, 2021). Klasifikasi sentimen dilakukan terhadap hasil pelabelan dengan library *Textblob* untuk dibandingkan hasilnya. Adapun pengolahan data klasifikasi sentimen adalah sebagai berikut.

a. *Splitting data ukuran 50/50*

Hasil klasifikasi sentimen yang diperoleh dari *splitting data* untuk ukuran data latih 50% dan data uji 50% berupa *Metrics* ditunjukkan pada tabel 4.9.

Tabel 4. 9 Hasil *Metrics* Klasifikasi Pelabelan TextBlob ukuran 50/50

<i>Accuracy</i>	92,19%
<i>Precision</i>	91,86%
<i>Recall</i>	92,27%
<i>F1 score</i>	92,06%

Dari pemodelan klasifikasi dengan *naïve bayes*, diperoleh nilai *accuracy* model dalam klasifikasi sebesar 92,19%, nilai *precision* 91,86%, nilai *recall* 92,27%, dan *f1 score* 92,06%. *Confusion matrix* yang didapatkan dari hasil klasifikasi diilustrasikan pada gambar 4.5.

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	True Positive 12.500	False Positive 1.107
	0 (Negative)	False Negative 1.047	True Negative 12.928

Gambar 4. 5 Confusion Matrix Klasifikasi Pelabelan Textblob ukuran 50/50

Berdasarkan hasil *confusion matrix* diketahui data yang bersentimen positif dan model memprediksi dengan benar ditunjukkan pada matriks *true positive* (TP) berjumlah 12.500. Sementara itu, untuk data bersentimen negatif dan model benar memprediksikan dengan benar ditunjukkan pada matriks *true negative* (TN) berjumlah 12.928. Pada matriks *false positive* (FP) dan *false negative* (FN) model mengalami *error* dalam memprediksi dengan masing-masing berjumlah 1.107 dan 1.047.

b. *Splitting data ukuran 60/40*

Hasil klasifikasi sentimen yang diperoleh dari *splitting data* untuk ukuran data latih 60% dan data uji 40% berupa *Metrics* ditunjukkan pada tabel 4.10.

Tabel 4. 10 Hasil *Metrics* Klasifikasi Pelabelan TextBlob ukuran 60/40

<i>Accuracy</i>	92,27%
<i>Precision</i>	91,88%
<i>Recall</i>	92,23%
<i>F1 score</i>	92,13%

Dari pemodelan klasifikasi dengan *naïve bayes*, diperoleh nilai *accuracy* model dalam klasifikasi sebesar 92,27%, nilai *precision* 91,88%, nilai *recall* 92,23%, dan *f1 score* 92,13%. *Confusion matrix* yang didapatkan dari hasil klasifikasi diilustrasikan pada gambar 4.6.

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	True Positive 9.991	False Positive 882
	0 (Negative)	False Negative 823	True Negative 10.370

Gambar 4. 6 *Confusion Matrix* Klasifikasi Pelabelan *Textblob* ukuran 60/40

Berdasarkan hasil *confusion matrix* diketahui data yang bersentimen positif dan model memprediksi dengan benar ditunjukkan pada matriks *true positive* (TP) berjumlah 9.991. Sementara itu, untuk data bersentimen negatif dan model benar memprediksikan dengan benar ditunjukkan pada matriks *true negative* (TN) berjumlah 10.370. Pada matriks *false positive* (FP) dan *false negative* (FN) model mengalami *error* dalam memprediksi dengan masing-masing berjumlah 882 dan 823.

c. ***Splitting data ukuran 70/30***

Hasil klasifikasi sentimen yang diperoleh dari *splitting data* untuk ukuran data latih 70% dan data uji 30% berupa *Metrics* ditunjukkan pada tabel 4.11.

Tabel 4. 11 Hasil *Metrics* Klasifikasi Pelabelan TextBlob ukuran 70/30

<i>Accuracy</i>	92,39%
<i>Precision</i>	91,88%

Recall	92,63%
F1 score	92,25%

Dari pemodelan klasifikasi dengan *naïve bayes*, diperoleh nilai *accuracy* model dalam klasifikasi sebesar 92,93%, nilai *precision* 92,65%, nilai *recall* 92,90%, dan *f1 score* 92,77%. *Confusion matrix* yang didapatkan dari hasil klasifikasi diilustrasikan pada gambar 4.7.

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	True Positive 7.494	False Positive 662
	0 (Negative)	False Negative 596	True Negative 7.798

Gambar 4. 7 *Confusion Matrix* Klasifikasi Pelabelan *Textblob* ukuran 60/40

Berdasarkan hasil *confusion matrix* diketahui data yang bersentimen positif dan model memprediksi dengan benar ditunjukkan pada matriks *true positive* (TP) berjumlah 7.494. Sementara itu, untuk data bersentimen negatif dan model benar memprediksikan dengan benar ditunjukkan pada matriks *true negative* (TN) berjumlah 7.798. Pada matriks *false positive* (FP) dan *false negative* (FN) model mengalami *error* dalam memprediksi dengan masing-masing berjumlah 662 dan 596.

d. **Splitting data ukuran 80/20**

Hasil klasifikasi sentimen yang diperoleh dari *splitting data* untuk ukuran data latih 80% dan data uji 20% berupa *Metrics* ditunjukkan pada tabel 4.12.

Tabel 4. 12 Hasil *Metrics* Klasifikasi Pelabelan *TextBlob* ukuran 80/10

Accuracy	92,33%
Precision	91,84%
Recall	92,49%
F1 score	92,17%

Dari pemodelan klasifikasi dengan *naïve bayes*, diperoleh nilai *accuracy* model dalam klasifikasi sebesar 92,33% nilai *precision* 91,84%, nilai *recall* 92,49%, dan *f1 score* 92,17%. *Confusion matrix* yang didapatkan dari hasil klasifikasi diilustrasikan pada gambar 4.8.

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	True Positive 4.980	False Positive 442
	0 (Negative)	False Negative 404	True Negative 5.207

Gambar 4. 8 Confusion Matrix Klasifikasi Pelabelan *Textblob* ukuran 80/20

Berdasarkan hasil *confusion matrix* diketahui data yang bersentimen positif dan model memprediksi dengan benar ditunjukkan pada matriks *true positive* (TP) berjumlah 4.980. Sementara itu, untuk data bersentimen negatif dan model benar memprediksikan dengan benar ditunjukkan pada matriks *true negative* (TN) berjumlah 5.207. Pada matriks *false positive* (FP) dan *false negative* (FN) model mengalami *error* dalam memprediksi dengan masing-masing berjumlah 442 dan 404.

Hasil yang diperoleh dari keempat jenis pembagian data kemudian dibandingkan untuk mengetahui performa mana yang memiliki *output* terbaik. Adapun perbandingannya ditunjukkan pada tabel 4.13.

Tabel 4. 13 Perbandingan Ukuran *Split Data*, *Accuracy*, dan *F1 Score* untuk Klasifikasi Pelabelan dengan *Textblob*

	<i>Split Data</i>			
	50/50	60/40	70/30	80/20
<i>Accuracy</i>	92,19%	92,27%	92,39%	92,33%
<i>F1 Score</i>	92,06%	92,13%	92,25%	92,17%

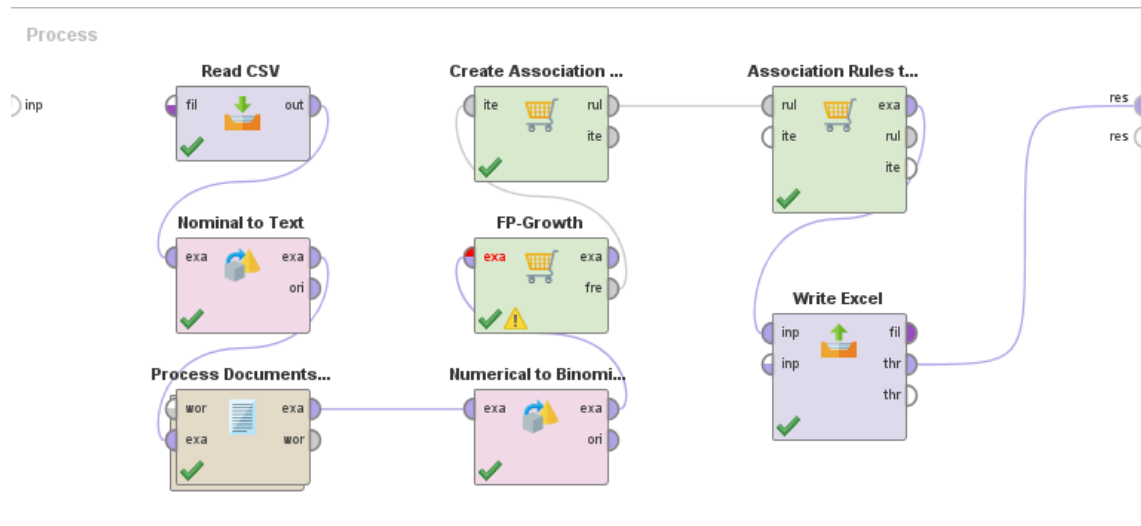
Dari perbandingan untuk tingkat *accuracy* dan nilai *F1 score* pada tabel 4.18 maka dapat diketahui bahwa *split data* dengan ukuran data latih 70% dan data uji 30% menghasilkan nilai tertinggi.

4.2.4 Association Rules

Pengolahan data kemudian dilanjutkan dengan *association rules* yang bertujuan untuk mencari pola atau asosiasi antara satu kata dengan kata lainnya. Pengolahan data ini juga menggunakan bantuan aplikasi RapidMiner Studio. Data yang diekstraksi adalah data *review* dengan sentimen negatif yang berisi kritik dan keluhan atas aplikasi Google Classroom sejumlah 27.233 data. Pada pengolahan data ini ditambahkan tahap *preprocessing data* berupa *stemming* menggunakan algoritma *Snowball Stemmer* yang bertujuan untuk meminimalisir keberagaman kata. Algoritma *Snowball Stemmer* merupakan pengembangan dari algoritma *Porter Stemmer* yang dinilai memiliki output terbaik dibandingkan algoritma *stemmer* lainnya (Anjali *et al.*, 2007).

Pengolahan *Association rules* menggunakan algoritma *FP-Growth*. Algoritma *FP-Growth* diketahui sebagai algoritma yang efisien untuk menemukan *frequent itemsets* yang dinilai memiliki performa lebih baik apabila dibandingkan dengan algoritma *Apriori* (Mythili & Shavanas, 2013). *Data mining* dengan tujuan mencari kumpulan yang paling banyak muncul di dalam data dengan *FP-Growth* dilakukan dengan membuat struktur *FP-Tree* (Widiastuti & Sofi, 2014).

Pengolahan data *association rules* dengan algoritma *FP-Growth* pada RapidMiner menggunakan *min support* sebesar 0,001 dan nilai *confidence* yang diterapkan adalah 0,1. Hal tersebut dikarenakan terlalu beragamnya jenis kata, sehingga dengan menurunkan nilai *min support* dan *confidence* dapat diperoleh hasil cakupan asosiasi yang luas. Nilai *support* menunjukkan persentase dari kata yang ada di dalam data, sedangkan nilai *confidence* menunjukkan kuatnya asosiasi antar kata yang terbentuk. Apabila nilai *lift ratio* yang dihasilkan lebih dari 1 maka dapat diketahui bahwa hubungan atau asosiasi antara *conclusion* dan *premises* lebih signifikan. Adapun desain proses untuk pengolahan data *association rules* menggunakan RapidMiner Studio ditunjukkan pada gambar 4.9.



Gambar 4. 9 Desain Operasi Association Rules pada RapidMiner

Pada penelitian ini hasil *association rules* yang digunakan dipilih berdasarkan kata yang berasosiasi dengan fitur dalam aplikasi Google Classroom. Dari *association rules* menghasilkan *conclusion* dan *premises* yang merupakan kesatuan *itemsets*. *Premises* merupakan elemen yang ditemukan dalam data, sedangkan *conclusion* ditemukan pada kombinasi yang terbentuk dengan *premises*. *Conclusion* disini akan dijadikan sebagai kata kunci permasalahan dari fitur Google Classroom dan *premises* menjadi asosiasi kata dari kata kunci permasalahan.

Dari pengolahan data *association rules* dengan *FP-Growth* diperoleh *frequent words* yang berhubungan dengan aplikasi Google Classroom ditunjukkan berdasarkan nilai *support*. Adapun kata-kata tersebut ditunjukkan pada table 4.14.

Tabel 4. 14 *Frequent Words*

Words	Support
<i>class</i>	0.042
<i>assignment</i>	0.036
<i>upload</i>	0.024
<i>submit</i>	0.019
<i>file</i>	0.018
<i>notification</i>	0.017
<i>dark</i>	0.009

Conclusion yang digunakan sebagai kata kunci dalam *association rules* meliputi 'assignment', 'upload', 'submit', 'file', 'class', 'notification', dan 'dark' yang ditemukan dari *frequent words*. Adapun hasil dari *association rules* berdasarkan kata kunci yang dipilih ditunjukkan pada tabel 4.15 sampai dengan table 4.21.

Tabel 4. 15 Tabel *Association Rules Conclusion Assignment*

Conclusion	Premises	Support	Confidence	Lift
<i>assignment</i>	<i>show, miss</i>	0.001102	0.625	17.40286
	<i>notification</i>	0.001102	0.508475	14.15826
	<i>submit, late</i>	0.001175	0.507937	14.14328
	<i>time, late</i>	0.001506	0.493976	13.75455
	<i>time, submit</i>	0.002497	0.450331	12.53928
	<i>teacher, late</i>	0.001028	0.444444	12.37537
	<i>google, submit</i>	0.001138	0.442857	12.33117
	<i>teacher, submit</i>	0.001579	0.434343	12.09411
	<i>turn</i>	0.003195	0.428571	11.93339
	<i>teacher, post</i>	0.001212	0.428571	11.93339
	<i>miss</i>	0.003929	0.424603	11.8229
	<i>date</i>	0.001138	0.413333	11.50909
	<i>worst, submit</i>	0.001359	0.406593	11.32142
	<i>take, submit</i>	0.001432	0.402062	11.19524
	<i>time, notification</i>	0.001285	0.380435	10.59305
	<i>time, send</i>	0.001028	0.373333	10.39531
	<i>submit</i>	0.006867	0.36811	10.24988
	<i>time, show</i>	0.001102	0.357143	9.944493
	<i>time, teacher</i>	0.001726	0.350746	9.766383
	<i>teacher, give</i>	0.002056	0.341463	9.507906
<i>late</i>	0.004076	0.330357	9.198656	
<i>post</i>	0.002313	0.324742	9.042312	
<i>grade</i>	0.001028	0.318182	8.859639	
<i>minute</i>	0.001359	0.310924	8.657559	

<i>Conclusion</i>	<i>Premises</i>	<i>Support</i>	<i>Confidence</i>	<i>Lift</i>
	<i>google, teacher</i>	0.001322	0.302521	8.423571
	<i>find</i>	0.001175	0.301887	8.405911
	<i>test</i>	0.001065	0.287129	7.994979
	<i>show</i>	0.004186	0.280788	7.818429
	<i>time, google</i>	0.001249	0.278689	7.759965
	<i>time, load</i>	0.001102	0.267857	7.45837
	<i>take, upload</i>	0.001946	0.259804	7.234131
	<i>student, teacher</i>	0.001469	0.25974	7.232359
	<i>time, take, upload</i>	0.001028	0.252252	7.023858
	<i>time, take</i>	0.002974	0.247706	6.897281
	<i>notification</i>	0.004186	0.245161	6.826413
	<i>worst, teacher</i>	0.001469	0.242424	6.750201
	<i>issue</i>	0.001065	0.233871	6.512039
	<i>teacher</i>	0.007822	0.218014	6.070518
	<i>time, upload</i>	0.001542	0.217617	6.059442
	<i>fail</i>	0.001102	0.215827	6.009622
	<i>know</i>	0.001726	0.211712	5.895024
	<i>send</i>	0.002203	0.20979	5.84152
	<i>say</i>	0.00213	0.208633	5.809301
	<i>take, load</i>	0.001212	0.207547	5.779064
	<i>time, work</i>	0.001469	0.205128	5.711709
	<i>time, slow</i>	0.001028	0.20438	5.690863
	<i>take, file</i>	0.001102	0.204082	5.682568
	<i>upload</i>	0.004811	0.203733	5.672846
	<i>load</i>	0.003121	0.201422	5.608506
	<i>attachment</i>	0.001873	0.200787	5.590841
	<i>work, teacher</i>	0.001432	0.19898	5.540503
	<i>worst, time</i>	0.001506	0.195238	5.436323
	<i>get</i>	0.001506	0.193396	5.385037
	<i>time, file</i>	0.001138	0.192547	5.361379

<i>Conclusion</i>	<i>Premises</i>	<i>Support</i>	<i>Confidence</i>	<i>Lift</i>
	<i>take</i>	0.005251	0.188158	5.239178
	<i>hate, teacher</i>	0.001249	0.184783	5.145194
	<i>problem</i>	0.002167	0.179331	4.993405
	<i>give</i>	0.00437	0.177612	4.94553
	<i>google, classroom</i>	0.00213	0.171598	4.778064
	<i>keep</i>	0.001285	0.168269	4.685386
	<i>file</i>	0.003011	0.163022	4.539276
	<i>annoying</i>	0.001542	0.160305	4.463635
	<i>upload, file</i>	0.001322	0.154506	4.302167
	<i>time</i>	0.008152	0.149596	4.165429
	<i>thing</i>	0.001469	0.145455	4.050121
	<i>classroom</i>	0.003048	0.145105	4.040385
	<i>able</i>	0.001028	0.136585	3.803162
	<i>make</i>	0.002791	0.131034	3.6486
	<i>wrong</i>	0.001249	0.130268	3.627263
	<i>update</i>	0.001579	0.126471	3.521521
	<i>please</i>	0.002864	0.111429	3.102682
	<i>work</i>	0.005141	0.109976	3.062248
	<i>google</i>	0.004296	0.109142	3.039007
	<i>want</i>	0.001102	0.106762	2.972731
	<i>homework</i>	0.001359	0.101093	2.814889
	<i>use</i>	0.001065	0.101045	2.813564

Dari tabel hasil *association rules* yang ditunjukkan pada tabel 4.15 untuk bagian *conclusion* kata '*assignment*' memiliki beberapa asosiasi yang ditunjukkan pada kolom *premises*. Masalah yang ditemui terkait kata kunci '*assignment*' di dalam aplikasi berkaitan dengan guru, keterlambatan, waktu pengumpulan, dan notifikasi.

Tabel 4. 16 Tabel Association Rules Conclusion Assignment

Conclusion	Premises	Support	Confidence	Lift
<i>upload</i>	<i>slow, speed</i>	0.001175	0.8	33.88118
	<i>speed</i>	0.001873	0.68	28.799
	<i>take, file</i>	0.003195	0.591837	25.06516
	<i>file, show</i>	0.001102	0.566038	23.97253
	<i>time, take, file</i>	0.001836	0.543478	23.01711
	<i>slow, file</i>	0.001763	0.527473	22.33924
	<i>time, file</i>	0.002938	0.496894	21.04421
	<i>worst, file</i>	0.001359	0.486842	20.61848
	<i>file</i>	0.008556	0.463221	19.61808
	<i>assignment, file</i>	0.001322	0.439024	18.59333
	<i>photo</i>	0.001212	0.4125	17.46998
	<i>document</i>	0.001285	0.37234	15.76917
	<i>assignment, take</i>	0.001946	0.370629	15.6967
	<i>time, assignment, take</i>	0.001028	0.345679	14.64002
	<i>time, take</i>	0.004076	0.33945	14.37619
	<i>work, take</i>	0.001285	0.294118	12.45632
	<i>worst, take</i>	0.001395	0.292308	12.37966
	<i>time, slow</i>	0.001432	0.284672	12.05626
	<i>hour</i>	0.001322	0.270677	11.46356
	<i>take</i>	0.007491	0.268421	11.36803
	<i>minute</i>	0.001065	0.243697	10.32095
	<i>video</i>	0.001689	0.240838	10.19983
	<i>time, assignment</i>	0.001542	0.189189	8.012442
	<i>time, work</i>	0.001138	0.158974	6.732799
	<i>slow</i>	0.005361	0.158009	6.6919
	<i>worst, time</i>	0.001212	0.157143	6.655232
	<i>attachment</i>	0.001322	0.141732	6.002572
	<i>show</i>	0.002093	0.140394	5.945897

<i>Conclusion</i>	<i>Premises</i>	<i>Support</i>	<i>Confidence</i>	<i>Lift</i>
	<i>late</i>	0.001726	0.139881	5.924165
	<i>assignment</i>	0.004811	0.133947	5.672846
	<i>time</i>	0.007087	0.130054	5.507975
	<i>problem</i>	0.001432	0.118541	5.020388
	<i>annoying</i>	0.001138	0.118321	5.011053

Pada tabel 4.16 dengan *conclusion*-nya merupakan kata 'upload', diperoleh asosiasi dengan kata-kata untuk masalah yang berhubungan dengan waktu mengunggah *file* atau dokumen. Selain itu terdapat kata-kata untuk jenis unggahan *file* dalam bentuk foto dan video.

Tabel 4. 17 Tabel Association Rules Conclusion Submit

<i>Conclusion</i>	<i>Premises</i>	<i>Support</i>	<i>Confidence</i>	<i>Lift</i>
<i>submit</i>	<i>time, assignment</i>	0.002497	0.306306	16.41995
	<i>assignment, late</i>	0.001175	0.288288	15.45407
	<i>assignment, take</i>	0.001432	0.272727	14.6199
	<i>google, assignment</i>	0.001138	0.264957	14.20338
	<i>worst, assignment</i>	0.001359	0.248322	13.31163
	<i>time, work</i>	0.001579	0.220513	11.82088
	<i>assignment, teacher</i>	0.001579	0.201878	10.82193
	<i>assignment</i>	0.006867	0.191207	10.24988
	<i>late</i>	0.002313	0.1875	10.05118
	<i>time, take</i>	0.002203	0.183486	9.836018
	<i>time, file</i>	0.001028	0.173913	9.322835
	<i>work, teacher</i>	0.001212	0.168367	9.02555
	<i>show</i>	0.00235	0.157635	8.450254
	<i>miss</i>	0.001322	0.142857	7.658043
	<i>worst, time</i>	0.001102	0.142857	7.658043
	<i>work</i>	0.006096	0.130401	6.990295
	<i>take</i>	0.003562	0.127632	6.841857

<i>Conclusion</i>	<i>Premises</i>	<i>Support</i>	<i>Confidence</i>	<i>Lift</i>
	<i>say</i>	0.001285	0.125899	6.748995
	<i>file</i>	0.002313	0.125249	6.714109
	<i>problem</i>	0.001432	0.118541	6.354546
	<i>attachment</i>	0.001065	0.114173	6.120404
	<i>time</i>	0.005545	0.101752	5.454549
	<i>teacher</i>	0.003635	0.101331	5.431959

Untuk kata kunci conclusion '*submit*' yang ditunjukkan pada tabel 4.17 diperoleh beberapa masalah dengan beberapa kata kunci. Kata-kata tersebut berkaitan dengan tugas atau pekerjaan yang dikerjakan oleh siswa mengalami kendala dalam proses pengumpulan.

Tabel 4. 18 Tabel *Association Rules Conclusion File*

<i>Conclusion</i>	<i>Premises</i>	<i>Support</i>	<i>Confidence</i>	<i>Lift</i>
<i>file</i>	<i>drive</i>	0.001395	0.622951	33.72604
	<i>please, attachment</i>	0.001028	0.54902	29.72346
	<i>upload, show</i>	0.001102	0.526316	28.4943
	<i>time, attachment</i>	0.001432	0.4875	26.39284
	<i>attachment</i>	0.00437	0.468504	25.36441
	<i>time, take, upload</i>	0.001836	0.45045	24.38701
	<i>take, upload</i>	0.003195	0.426471	23.08876
	<i>time, upload</i>	0.002938	0.414508	22.4411
	<i>take, attachment</i>	0.001065	0.408451	22.11318
	<i>upload</i>	0.008556	0.362364	19.61808
	<i>worst, upload</i>	0.001359	0.349057	18.89763
	<i>slow, upload</i>	0.001763	0.328767	17.79918
	<i>slow, take</i>	0.001028	0.307692	16.6582
	<i>time, take</i>	0.003378	0.281346	15.23181
	<i>assignment, upload</i>	0.001322	0.274809	14.87794
	<i>time, slow</i>	0.001359	0.270073	14.62153

<i>Conclusion</i>	<i>Premises</i>	<i>Support</i>	<i>Confidence</i>	<i>Lift</i>
	<i>assignment, take</i>	0.001102	0.20979	11.35787
	<i>take</i>	0.005398	0.193421	10.47165
	<i>time, submit</i>	0.001028	0.18543	10.03905
	<i>time, assignment</i>	0.001138	0.13964	7.559973
	<i>show</i>	0.001946	0.130542	7.067428
	<i>send</i>	0.001322	0.125874	6.81472
	<i>submit</i>	0.002313	0.124016	6.714109
	<i>load</i>	0.001689	0.109005	5.901426
	<i>time</i>	0.005912	0.108491	5.873589
	<i>say</i>	0.001102	0.107914	5.842356
	<i>late</i>	0.001322	0.107143	5.800625
	<i>annoying</i>	0.001028	0.10687	5.785865
	<i>problem</i>	0.001249	0.103343	5.594929

Tabel 4.18 menunjukkan *conclusion* untuk kata kunci 'file' yang memiliki asosiasi dengan beberapa kata untuk masalah pada penggunaan *drive* yang dihubungkan dengan aplikasi Google Classroom dan masalah dalam proses pengumpulan *file*.

Tabel 4. 19 Tabel Association Rules Conclusion File

<i>Conclusion</i>	<i>Premises</i>	<i>Support</i>	<i>Confidence</i>	<i>Lift</i>
<i>class</i>	<i>worst, online</i>	0.002828	0.810526	19.4298
	<i>attend</i>	0.00213	0.716049	17.16501
	<i>online</i>	0.014321	0.709091	16.99821
	<i>hate, online</i>	0.005618	0.701835	16.82427
	<i>google, online</i>	0.001359	0.698113	16.73505
	<i>student, online</i>	0.001212	0.6875	16.48063
	<i>teacher, online</i>	0.001028	0.666667	15.98122
	<i>join</i>	0.003635	0.622642	14.92586
	<i>room</i>	0.002167	0.578431	13.86606
	<i>fail</i>	0.001322	0.258993	6.208532

<i>Conclusion</i>	<i>Premises</i>	<i>Support</i>	<i>Confidence</i>	<i>Lift</i>
	<i>able</i>	0.001689	0.22439	5.379045
	<i>try</i>	0.001212	0.215686	5.170395
	<i>meet</i>	0.001285	0.203488	4.877989
	<i>update</i>	0.002387	0.191176	4.58285
	<i>worst, teacher</i>	0.001102	0.181818	4.358515
	<i>say</i>	0.001836	0.179856	4.31148
	<i>hate, teacher</i>	0.001175	0.173913	4.169014
	<i>went</i>	0.001065	0.168605	4.041762
	<i>boring</i>	0.001138	0.164894	3.952802
	<i>wrong</i>	0.001542	0.16092	3.857536
	<i>know</i>	0.001249	0.153153	3.671362
	<i>help</i>	0.001102	0.153061	3.669158
	<i>show</i>	0.00224	0.150246	3.601679
	<i>google, classroom</i>	0.001726	0.139053	3.333361
	<i>miss</i>	0.001285	0.138889	3.329421
	<i>classroom</i>	0.002901	0.138112	3.310795
	<i>get</i>	0.001028	0.132075	3.166091
	<i>teacher</i>	0.0047	0.131013	3.140629
	<i>problem</i>	0.001542	0.12766	3.060234
	<i>late</i>	0.001542	0.125	2.996479
	<i>want</i>	0.001249	0.120996	2.900506
	<i>notification</i>	0.002056	0.12043	2.88693
	<i>please</i>	0.003048	0.118571	2.842374
	<i>google</i>	0.00459	0.116604	2.795223
	<i>star</i>	0.001689	0.108235	2.594598

Hasil *conclusion* untuk kata kunci 'class' pada tabel 4.19 memiliki hubungan dengan kata-kata untuk masalah yang ditemui pengguna terkait fitur utama dari Google Classroom yakni kelas. Secara umum pengguna merupakan siswa yang menemukan kendala ketika mencoba masuk ke kelas yang dibuat oleh pengajar atau guru.

Tabel 4. 20 Tabel Association Rules Conclusion Notification

Conclusion	Premises	Support	Confidence	Lift
<i>notification</i>	<i>receive</i>	0.001285	0.673077	39.4177
	<i>come</i>	0.001469	0.305344	17.88197
	<i>late</i>	0.003599	0.291667	17.081
	<i>miss</i>	0.001689	0.18254	10.69015
	<i>post</i>	0.001285	0.180412	10.56557
	<i>time, assignment</i>	0.001285	0.157658	9.232975
	<i>annoying</i>	0.001432	0.148855	8.717459
	<i>turn</i>	0.001028	0.137931	8.077716
	<i>send</i>	0.001432	0.136364	7.985924
	<i>update</i>	0.001652	0.132353	7.751044
	<i>show</i>	0.001946	0.130542	7.644981
	<i>assignment</i>	0.004186	0.116564	6.826413

Asosiasi yang ditunjukkan pada tabel 4.20 berkaitan dengan masalah pada kata kunci '*notification*' yang memiliki sejumlah *premises* meliputi keterlambatan pengguna aplikasi Google Classroom dalam menerima notifikasinya di platform android. Pengguna aplikasi memberikan kritik untuk segera memberikan *update* dalam menangani *bug* tersebut.

Tabel 4. 21 Tabel Association Rules Conclusion Notification

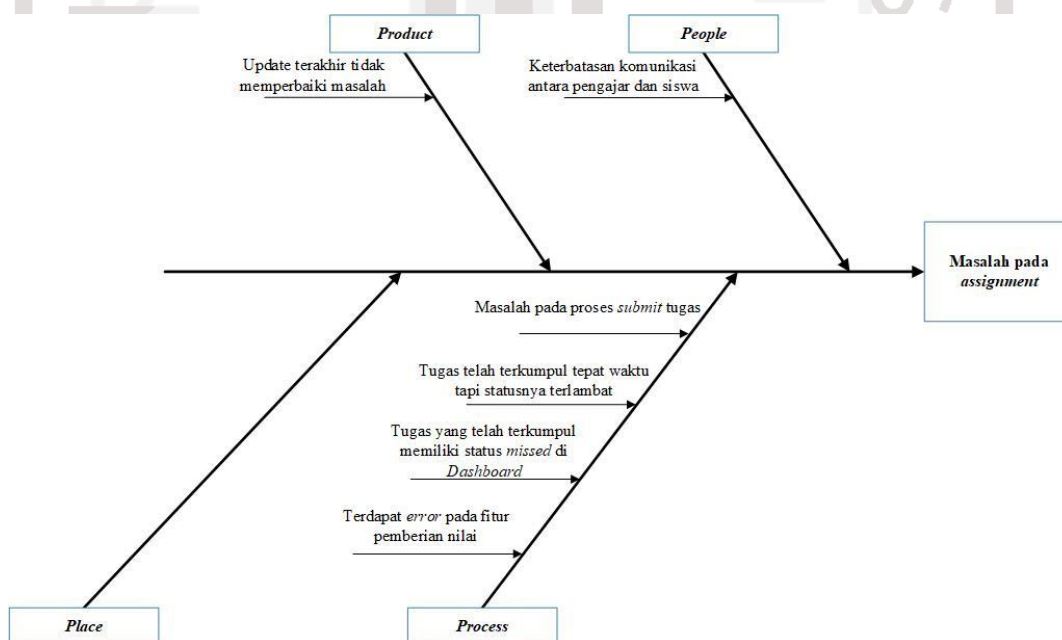
Conclusion	Premises	Support	Confidence	Lift
<i>dark</i>	<i>please, mode</i>	0.002534	0.971831	102.5771
	<i>mode</i>	0.007675	0.963134	101.6591
	<i>theme</i>	0.001579	0.934783	98.66667
	<i>please</i>	0.003158	0.122857	12.96762

Pada *conclusion* yang ditampilkan pada tabel 4.21 kata kunci '*dark*' memiliki asosiasi dengan kata '*please*', '*mode*', dan '*theme*'. Hal tersebut merujuk kepada tampilan dari aplikasi Google Classroom yang belum memiliki pengaturan untuk merubah mode tema.

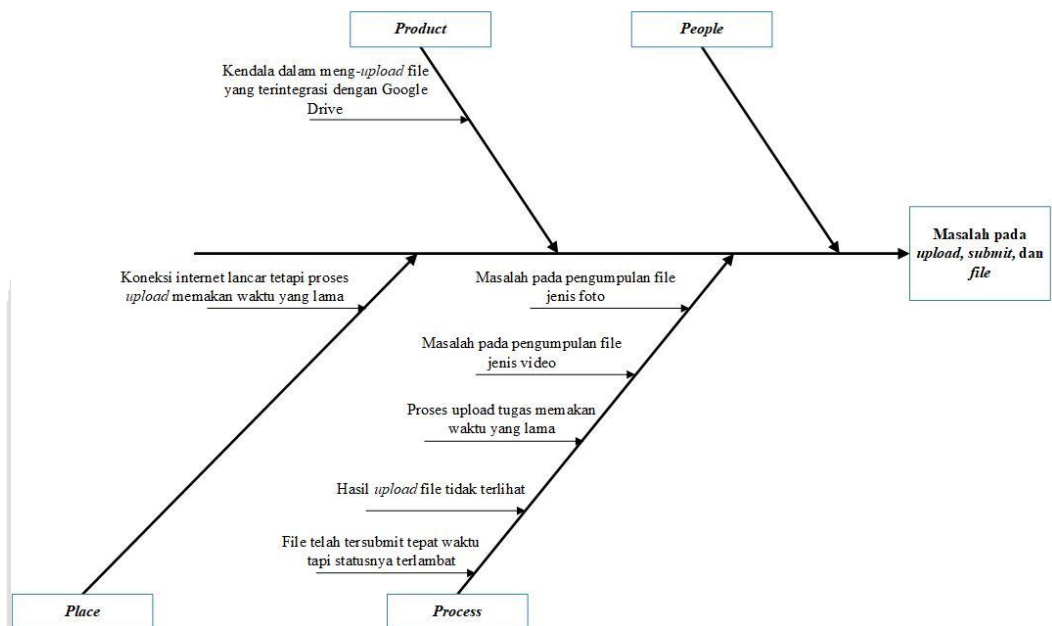
4.2.5 Analisis dengan *Fishbone Diagram*

Berdasarkan hasil pengolahan data pada *review* pengguna Google Classroom sentimen negatif dengan *association rules*, maka diperoleh kata kunci masalah dari aplikasi Google Classroom. Kata kunci dari masalah yang ditemukan meliputi *conclusion* yang ditemukan pada proses *association rules* yakni pada kata, 'assignment', 'upload', 'submit', 'file', 'class', 'notification', dan 'dark'. Dari permasalahan yang ditemukan kemudian dianalisis untuk mengetahui potensi dari akar masalah. Potensi akar masalah yang digunakan berdasarkan *premises* yang ada pada setiap *conclusion* dari hasil *association rules*. Kata 'assignment', 'class', dan 'notification' dijadikan sebagai inti permasalahan pada masing-masing fitur yang dibahas pada *fishbone diagram*. Kata 'upload', 'submit' dan 'file' dijadikan sebagai kesatuan dikarenakan kata tersebut memiliki *premises* yang serupa pada beberapa kata. Sementara itu, untuk kata 'dark' merujuk pada masalah pada tampilan dari aplikasi Google Classroom, sehingga inti permasalahan yang digunakan pada *fishbone diagram* adalah masalah pada tampilan aplikasi.

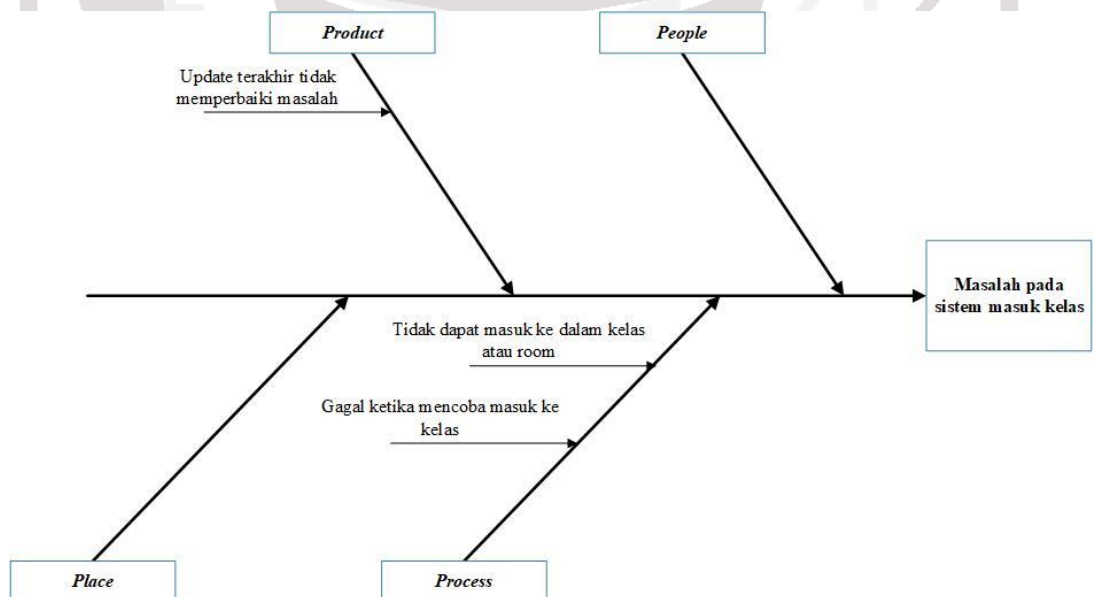
Analisis akar masalah menggunakan alat bantuan berupa *fishbone diagram* dengan menyertakan empat aspek pada *marketing mix* 4P yang berhubungan dengan permasalahan yang ditemukan pada aplikasi Google Classroom meliputi *product*, *people*, *place*, dan *process*. Adapun hasil dari pembuatan *fishbone diagram* untuk setiap masalah yang ditemukan ditunjukkan oleh gambar 4.10 sampai dengan 4.14.



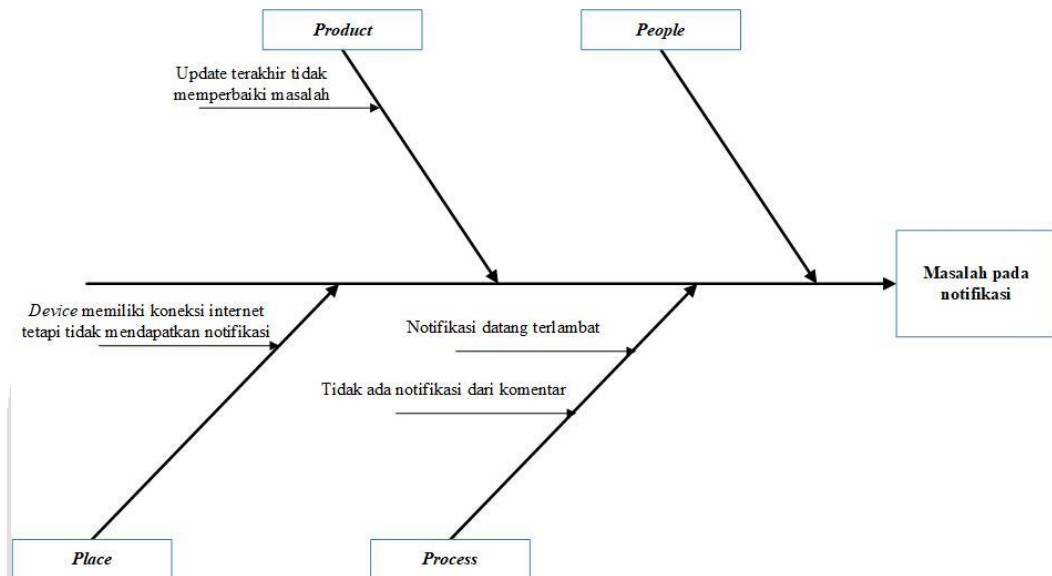
Gambar 4. 10 *Fishbone diagram* Terkait Masalah Assignment



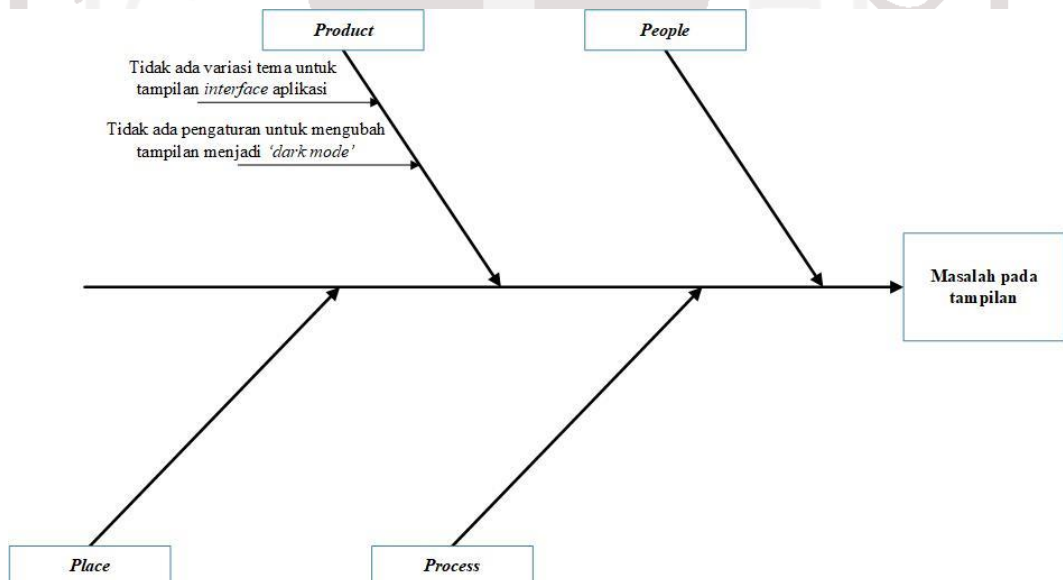
Gambar 4. 11 *Fishbone diagram* Terkait Masalah Proses *Upload* dan *Submit File*



Gambar 4. 12 *Fishbone diagram* Terkait Masalah Bergabung ke Dalam Kelas



Gambar 4. 13 *Fishbone diagram* Terkait Masalah Notifikasi



Gambar 4. 14 *Fishbone diagram* Terkait Masalah Masalah Tampilan Aplikasi

BAB V

PEMBAHASAN

5.1 Hasil dari Analisis Sentimen

Analisis sentiment dilakukan untuk mengidentifikasi sentimen dari setiap *review* pengguna Google Classroom. Klasifikasi sentimen yang digunakan dalam penelitian merupakan pendekatan *hybrid* dengan menggabungkan dua teknik analisis sentiment, yakni *lexicon-based* berupa *Textblob* dan *supervised machine learning* berupa *Naïve Bayes*. *Library Textblob* digunakan untuk menentukan label sentimen *review* pengguna. Hasil pelabelan sentimen kemudian digunakan sebagai data latih dalam proses klasifikasi menggunakan algoritma *Naïve Bayes*.

Berdasarkan hasil pelabelan dari data keseluruhan berjumlah 77.455 *review* pengguna, diperoleh pembagian *review* dengan sentimen negatif berjumlah 27.233 data, label sentimen positif sebanyak 27.931 data, dan label sentimen netral sebanyak 22.290 data. Selanjutnya dilakukan klasifikasi menggunakan algoritma *Naïve Bayes*. Algoritma ini memerlukan data latih dan data uji. Oleh karena itu, data yang telah dilabeli dengan *Textblob* digunakan sebagai data latih. Ukuran data latih dan data uji untuk klasifikasi dengan *Naïve Bayes* dibagi menjadi empat ukuran *splitting data*, yakni ukuran 50/50; 60/40; 70/30; dan 80/20.

Berdasarkan klasifikasi dengan menggunakan ukuran *splitting data* 50% data latih dan 50% data uji diperoleh hasil akurasi adalah 92,19% dengan nilai *f1 score* sebesar 92,06%. *Confusion matrix* menunjukkan bahwa model berhasil memprediksi dengan benar bahwa data *review* memiliki sentimen positif sebanyak 12.500 data. Sementara itu, model memprediksi *review* bersentimen positif yang salah sebanyak 1.107 data. Model berhasil memprediksi bahwa *review* benar memiliki sentimen negatif sebanyak 12.928 data dan hasil prediksi salah untuk sentimen negatif berjumlah 1.047 data.

Klasifikasi selanjutnya menggunakan ukuran *splitting data* 60% untuk data latih dan 40% untuk data uji. Akurasi yang diperoleh dari hasil klasifikasi adalah sebesar 92,27%

dengan nilai *f1 score* adalah 92,13%. Hasil dari *confusion matrix* menunjukkan bahwa jumlah data yang model berhasil prediksi bawah *review* memiliki sentimen positif sebanyak 9.991 dan hasil prediksi salah untuk data sentimen positif berjumlah 882 data. Untuk jumlah data yang berhasil diprediksi model memiliki sentimen negatif sebanyak 10.370 data. Model memprediksi data *review* negatif salah berjumlah 823 data.

Hasil dari klasifikasi menggunakan *splitting data* 70% untuk data latih dan 30% untuk data uji memperoleh akurasi sebesar 92,39% dengan nilai *f1 score* adalah 92,25%. *Confusion matrix* menunjukkan bahwa sebanyak 7.494 data benar diprediksi oleh model memiliki sentimen positif. Sedangkan, sebanyak 662 data merupakan *false positive*. Model berhasil memprediksi data dengan sentimen negatif dengan benar sebanyak 7.798 data. Sedangkan, model salah memprediksi data sentimen negatif sebanyak 596 data.

Dari klasifikasi dengan menggunakan *splitting data* ukuran 80% data latih dan 20% data uji diperoleh hasil akurasi 92,33% dengan nilai *f1 score* sebesar 92,17%. Hasil dari *confusion matrix* menunjukkan bahwa model berhasil memprediksi *review* dengan sentimen positif sebanyak 4.980 data dan salah prediksi sejumlah 442 data. Sementara itu, model berhasil memprediksi *review* dengan sentimen negatif dengan benar sebanyak 5.207 data dan prediksi salah sejumlah 404 data.

Berdasarkan hasil akurasi dan nilai *f1 score* yang diperoleh, diketahui bahwa klasifikasi dengan ukuran *splitting data* dengan data latih 70% dan data uji 30% menghasilkan *output* terbaik dengan nilai akurasi 92,39%. *Output* terbaik yang kedua diperoleh dari hasil *splitting data* dengan ukuran 80/20 dengan nilai akurasi 92,33%. Hasil tersebut secara berturut-turut diikuti oleh *splitting data* dengan ukuran 60/40 dan 50/50 dengan masing-masing akurasi 92,27% dan 92,19%.

5.2 Hasil dari Association Rules

Association rules menggunakan data *review* bersentimen negatif hasil dari pengolahan analisis sentiment berjumlah 27.233 data. *Output* dari proses *association rules* menggunakan kata kunci dari fitur yang bermasalah di aplikasi Google Classroom. Kata yang dipilih sebagai masalah yang akan dibahas dalam aplikasi Google Classroom berdasarkan *association rules* menggunakan nilai *min support* 0,001 dan *confidence* 0,1 meliputi kata 'assignment', 'upload', 'submit', 'file', 'notification', dan 'dark'.

Kata '*assignment*' sebagai *conclusion* berasosiasi dengan kata *premises* meliputi '*show*', '*miss*', '*notification*', '*submit*', '*late*', '*teacher*' dan seterusnya. Berdasarkan kata-kata tersebut dapat diketahui secara garis besar masalah yang ditemukan berkaitan dengan fitur *assignment* dalam aplikasi Google Classroom berkaitan dengan keterlambatan dalam pengumpulan tugas, tidak munculnya notifikasi pemberitahuan tugas, dan masalah terkait dengan guru. Ketika pengguna melakukan *submit* tugas, tugas yang telah dikumpulkan tepat waktu statusnya terlambat, tugas yang telah terkumpul memiliki status *missed* di bagian *dashboard assignment* aplikasi Google Classroom, dan masalah terkait pemberian nilai atau *grading*. Asosiasi kata '*teacher*' menunjukkan bahwa masalah dari sisi *human error* pengguna yakni antara pengajar dan siswa yang kurang komunikasi terkait tugas yang dilampirkan pada aplikasi Google Classroom.

Kata kunci selanjutnya adalah pada masalah '*upload*' memiliki *premises* kata meliputi '*slow*', '*speed*', '*take*', '*file*', '*show*', '*file*', '*photo*', '*video*' dan seterusnya. Dari kata-kata tersebut diketahui secara garis besar masalah yang ditemui oleh pengguna aplikasi Google Classroom terkait kata kunci *upload* adalah pada kecepatan *upload* baik dalam bentuk file, foto, atau video.

Pada kata kunci '*submit*' terdapat asosiasi dengan *premises* meliputi kata '*time*', '*assignment*', '*late*', '*take*', '*work*', '*miss*' dan lain-lain. Berdasarkan asosiasi kata yang dihasilkan dapat diketahui secara garis besar masalah yang ditemui serupa dengan kata kunci '*upload*', yakni masih berkaitan dengan lambat proses *submit* dan *upload*.

Masalah yang ditemui berdasarkan asosiasi kata '*file*' dengan beberapa kata meliputi '*drive*', '*attachment*', '*upload*', '*show*', '*time*', '*upload*' dan seterusnya masih berkaitan dengan kata kunci '*upload*' dan '*submit*'. Secara garis besar dapat diketahui masalah yang ditemui pengguna adalah mengacu pada kendala *upload* file yang bersumber dari *drive* dan kecepatan *upload file* yang lambat.

Kata '*class*' dipilih menjadi salah satu permasalahan yang dibahas dengan *premises* kata meliputi '*worst*', '*online*', '*attend*', '*join*', '*fail*', '*able*', dan seterusnya. Berdasarkan dengan asosiasi kata yang dihasilkan maka dapat diketahui bahwa pengguna menemui masalah pada fitur kelas dalam aplikasi Google Classroom terkait proses bergabung ke dalam kelas virtual.

Pada kata kunci '*notification*' berasosiasi dengan *premises* meliputi kata '*receive*', '*come*', '*late*', '*miss*', '*post*', '*time*', '*bug*', dan seterusnya. Asosiasi kata tersebut

menunjukkan bahwa pengguna mengeluhkan masalah terkait notifikasi pada aplikasi Google Classroom yang sering datang terlambat, terutama pada *post* bentuk *assignment*. Selain itu, pengguna juga meminta *update* untuk mengatasi *bug* terkait dengan notifikasi tersebut. Hal tersebut dapat disebabkan dari sistem notifikasi pada aplikasi yang masih belum optimal, sehingga banyak pengguna yang mengeluhkan tidak menerima notifikasi atau mendapatkan notifikasi tapi terlambat. Padahal *device* yang digunakan oleh pengguna dapat digunakan untuk mengakses internet.

Kata kunci yang terakhir adalah '*dark*' yang merujuk kepada tampilan aplikasi Google Classroom. Kata tersebut berasosiasi dengan *premises* meliputi '*please*', '*mode*', dan '*theme*'. Berdasarkan kata dalam *premises* tersebut dapat diketahui bahwa pengguna mengeluhkan bahwa tidak adanya pengaturan bagi pengguna untuk melakukan kustomisasi tampilan aplikasi seperti mengganti tema atau mengubah tampilan menjadi mode malam.

Berdasarkan hasil dari pengolahan *association rules* dapat diketahui bahwa sebagian besar *review* diberikan oleh siswa dibandingkan dengan pengajar. Hal tersebut dikarenakan masalah atau kendala yang ditemukan sebagian besar berorientasi pada proses pengumpulan tugas.

5.3 Hasil dari Analisis *Fishbone Diagram*

Pembuatan *fishbone diagram* dilakukan berdasarkan asosiasi kata yang ditemukan dari tahap *association rules*. Inti permasalahan yang dibahas pada analisis menggunakan *fishbone diagram* meliputi masalah pada *assignment*; masalah pada proses *upload*, *submit*, dan *file*; masalah pada notifikasi; masalah pada system masuk kelas; dan masalah pada tampilan aplikasi. Berdasarkan inti permasalahan yang diperoleh maka dicari akar masalah dari *premises* yang diperoleh dari hasil *association rules*. Analisis *fishbone diagram* dalam penelitian ini menggunakan empat aspek untuk pemecahan akar masalah yakni *product*, *people*, *place*, dan *process*. Aspek *product* terkait dengan masalah berkaitan dengan fitur produk yang merupakan aplikasi Google Classroom. Aspek *people* berkaitan dengan masalah *human error* yang ditemui sehingga terjadi kendala dalam penggunaan aplikasi. Aspek *place* berhubungan dengan masalah pada akses aplikasi Google Classroom. Aspek terakhir adalah *process* yakni berkaitan dengan masalah yang ditemui pengguna ketika menjalankan aplikasi. Berdasarkan analisis akar masalah yang

telah dilakukan untuk setiap kata kunci masalah pada *fishbone diagram*, maka dapat disusun rekomendasi perbaikan menurut setiap aspek yang ada.

Tabel 5. 1 Rekomendasi Perbaikan Berdasarkan Aspek

Aspek	Permasalahan	Rekomendasi Perbaikan
<i>Product</i>	<ol style="list-style-type: none"> 1. Kendala dalam meng-<i>upload file</i> yang terintegrasi dengan Google Drive. 2. Tidak adanya variasi tema untuk tampilan <i>interface</i> aplikasi. 3. Tidak adanya pengaturan untuk mengubah tampilan menjadi '<i>dark mode</i>'. 	<ol style="list-style-type: none"> 1. Melakukan perbaikan integrasi <i>file</i> dengan Google Drive. 2. Penambahan fitur untuk mengubah tema <i>interface</i> dari aplikasi Google Classroom agar lebih menarik bagi pengguna. 3. Pengaturan untuk mengubah tampilan ke mode gelap sesuai dengan preferensi pengguna.
<i>People</i>	<ol style="list-style-type: none"> 1. Kurang optimalnya komunikasi antara pengajar dan siswa sehingga dapat menimbulkan miskomunikasi, 	<ol style="list-style-type: none"> 1. Memperbaiki dan meningkatkan forum komunikasi tugas antara siswa dan pengajar.
<i>Place</i>	<ol style="list-style-type: none"> 1. Tidak adanya notifikasi aplikasi padahal pengguna memiliki koneksi internet. 2. Pengguna memiliki koneksi internet yang baik tetapi proses <i>upload file</i> lambat. 	<ol style="list-style-type: none"> 1. Optimisasi notifikasi agar dapat tetap diterima oleh pengguna dan tidak mengalami keterlambatan. 2. Melakukan <i>update</i> pada <i>server</i> Google Classroom sehingga proses <i>upload file</i> dapat dilakukan lebih cepat.
<i>Process</i>	<ol style="list-style-type: none"> 1. Tugas terkumpul tepat waktu tetapi statusnya terlambat. 	<ol style="list-style-type: none"> 1. Melakukan perbaikan pada fitur penilaian tugas. 2. Perlu dilakukan perbaikan pada proses <i>upload</i> untuk berbagai jenis <i>file</i>.

-
2. Tugas yang telah terkumpul memiliki status *missed* di *Dashboard*.
 3. Terdapat *error* pada fitur pemberian nilai.
 4. Masalah pada pengumpulan jenis file foto dan video.
 5. Hasil *upload file* tidak terlihat.
 6. Gagal ketika masuk ke *room* kelas.
 7. Notifikasi datang terlambat.
 8. Tidak ada notifikasi dari komentar.
3. Perlu dilakukan perbaikan pada *server*.
 4. Perlu dilakukannya *update* untuk mengatasi *bug*.
-

الجامعة الإسلامية
الاستدراكية

BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan tujuan penelitian yang telah dijabarkan pada Bab I, maka dapat dirumuskan kesimpulan sebagai berikut.

1. Dari hasil analisis sentiment menggunakan *Textblob* dan *Naïve Bayes Classifier* diperoleh nilai akurasi dan *f1 score* sebesar 92,39% dan 92,25% dengan menggunakan *splitting* data ukuran 70% data latih dan 30% data uji.
2. Berdasarkan pengolahan *association rules* dengan metode *FP-Growth* maka dapat ditemukan kata yang paling sering muncul dengan melihat nilai *support* dari kata tersebut. Adapun kata yang paling sering muncul dalam *review* pengguna aplikasi Google Classroom adalah '*class*' dengan nilai *support* 0,042, '*assignment*' dengan nilai *support* 0,036, '*upload*' dengan nilai *support* 0,024, '*submit*' dengan nilai *support* 0,019, '*file*' dengan nilai *support* 0,018, '*notification*' dengan nilai *support* 0,017, dan '*dark*' dengan nilai *support* 0,009
3. Akar masalah yang ditemukan dari hasil *association rules* berkaitan dengan kendala dalam meng-*upload file* yang terintegrasi dengan Google Drive, tidak adanya variasi tema untuk tampilan *interface* aplikasi, tidak adanya pengaturan untuk mengubah tampilan menjadi '*dark mode*', kurang optimalnya komunikasi antara pengajar dan siswa sehingga dapat menimbulkan miskomunikasi, Tidak adanya notifikasi aplikasi padahal pengguna memiliki koneksi internet, tidak adanya notifikasi aplikasi padahal pengguna memiliki koneksi internet, pengguna memiliki koneksi internet yang baik tetapi proses *upload file* lambat, tugas terkumpul tepat waktu tetapi statusnya terlambat, tugas yang telah terkumpul memiliki status *missed* di *Dashboard*, terdapat *error* pada fitur pemberian nilai, masalah pada pengumpulan jenis file foto dan video, hasil *upload file* tidak terlihat, gagal ketika

masuk ke *room* kelas, notifikasi datang terlambat, tidak ada notifikasi dari komentar.

4. Berdasarkan masalah yang telah ditemukan dari analisis menggunakan *fishbone diagram* maka dapat disusun rekomendasi perbaikan meliputi melakukan perbaikan integrasi *file* dengan Google Drive, penambahan fitur untuk mengubah tema *interface* dari aplikasi Google Classroom agar lebih menarik bagi pengguna, pengaturan untuk mengubah tampilan ke mode gelap sesuai dengan preferensi pengguna, memperbaiki dan meningkatkan forum komunikasi tugas antara siswa dan pengajar, optimisasi notifikasi agar dapat tetap diterima oleh pengguna dan tidak mengalami keterlambatan, melakukan *update* pada *server* Google Classroom sehingga proses *upload file* dapat dilakukan lebih cepat, melakukan perbaikan pada fitur penilaian tugas, perlu dilakukan perbaikan pada proses *upload* untuk berbagai jenis *file*, perlu dilakukan perbaikan pada *server*, dan perlu dilakukannya *update* untuk mengatasi *bug*

6.2 Saran

Adapun saran yang dapat diberikan berdasarkan hasil yang diperoleh dari penelitian ini adalah sebagai berikut:

1. Bagi aplikasi Google Classroom, saran yang diberikan adalah untuk menambahkan fitur dan perbaikan pada aplikasi berdasarkan kritik dan saran yang diberikan oleh pengguna aplikasi Google Classroom. Dengan perbaikan yang didasarkan atas permintaan pengguna, diharapkan dapat meningkatkan *rating* dari aplikasi Google Classroom.
2. Bagi peneliti selanjutnya, saran yang dapat diberikan berupa menggunakan data dalam bahasa lain untuk melihat keberagaman variabel masalah berdasarkan bahasa dari negara lain. Selain itu, untuk penelitian selanjutnya dapat digunakan metode yang berbeda untuk memperoleh hasil yang lebih baik.

DAFTAR PUSTAKA

- Aggarwal, C. C., & Zhai, C. (Eds.). (2012). Mining text data. Springer Science & Business Media.
- Afolabi, I., Sowunmi, O., & Daramola, O. (2017). Semantic association rule mining in text using domain ontology. *International Journal of Metadata, Semantics and Ontologies*, 12(1), 28–34. <https://doi.org/10.1504/IJMSO.2017.087646>
- Ali, R. S. H., & Gayar, N. El. (2019). Sentiment Analysis using Unlabeled Email data. *Proceedings of 2019 International Conference on Computational Intelligence and Knowledge Economy, ICCIKE 2019*, 328–333. <https://doi.org/10.1109/ICCIKE47802.2019.9004372>
- Anjali, Jivani, G., & Anjali, M. (2007). *A Comparative Study of Stemming Algorithms*. October, 2(2004), 1930–1938.
- Antons, D., Grünwald, E., Cichy, P., & Salge, T. O. (2020a). *The application of text mining methods in innovation research: current state, evolution patterns, and development priorities*. 50(3), 329–351. <https://doi.org/10.1111/radm.12408>
- Awwalu, J., Umar, N. A., Ibrahim, M. S., & Nonyelum, O. F. (2020). a Multinomial Naïve Bayes Decision Support System for Covid-19 Detection. *Fudma Journal of Sciences*, 4(2), 704–711. <https://doi.org/10.33003/fjs-2020-0402-331>
- Bustamante, C., Garrido, L., & Soto, R. (2006). Comparing fuzzy Naive Bayes and Gaussian Naive Bayes for decision making in RoboCup 3D. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4293 LNAI, 237–247. https://doi.org/10.1007/11925231_23
- Cavus, N., & Alhih, M. S. (2014). Learning Management Systems Use in Science Education. *Procedia - Social and Behavioral Sciences*, 143, 517–520. <https://doi.org/10.1016/j.sbspro.2014.07.429>
- Chaithra. (2019). Hybrid approach: naive bayes and sentiment VADER for analyzing sentiment of mobile unboxing video comments. *International Journal of Electrical and Computer Engineering (IJECE)*, 9(5), 4452. <https://doi.org/10.11591/ijece.v9i5.pp4452-4459>
- Davies, J., & Graff, M. (2005). Performance in e-learning: online participation and

- student grades. *British Journal of Educational Technology*, 36(4), 657–663.
- Dey, L., Chakraborty, S., Biswas, A., Bose, B., & Tiwari, S. (2016). Sentiment Analysis of Review Datasets Using Naïve Bayes' and K-NN Classifier. *International Journal of Information Engineering and Electronic Business*, 8(4), 54–62. <https://doi.org/10.5815/ijieeb.2016.04.07>
- Emrald, & Lhaksmana, K. M. (2019). Klasifikasi Kategori Hadits Menggunakan Naive Bayes Classifier. *E-Proceeding of Engineering*, 6(2), 9848–9858.
- Fahmi, I. (2021). Drone Emprit: Software for media monitoring and analytics. Available at <http://pers.droneemprit.id>.
- Feldman, R. (2013). Techniques and applications for sentiment analysis. *Communications of the ACM*, 56(4), 82–89. <https://doi.org/10.1145/2436256.2436274>
- Harandi, S. R. (2015). Effects of e-learning on students' motivation. *Procedia - Social and Behavioral Sciences*, 181, 423–430. <https://doi.org/10.1016/j.sbspro.2015.04.905>
- Herlinawati, N., Yuliani, Y., Faizah, S., Gata, W., & Samudi, S. (2020). Analisis Sentimen Zoom Cloud Meetings di Play Store Menggunakan Naïve Bayes dan Support Vector Machine. *CESS (Journal of Computer Engineering, System and Science)*, 5(2), 293. <https://doi.org/10.24114/cess.v5i2.18186>
- Iftakhar, S. (2016). Google classroom: what works and how? *Journal of Education and Social Sciences*, 3, 12–18.
- Ilie, G., & Ciocoiu, C. N. (2010). Ilie G. and. Ciocoiu C.N. APPLICATION OF FISHBONE DIAGRAM TO DETERMINE THE RISK OF AN EVENT WITH MULTIPLE CAUSES MANAGEMENT RESEARCH APPLICATION OF FISHBONE DIAGRAM TO DETERMINE THE RISK OF AN EVENT WITH MULTIPLE CAUSES. *Management Research and Practice*, 2(1), 1–20. <http://mrp.ase.ro/no21/f1.pdf>
- Jordan, M. I., & Mitchell, T. M. (2015). *Machine learning: Trends, perspectives, and prospects*. 349(6245).
- Kamil, I. (2020, Oktober 16). Survei: Google Classroom Jadi Platform Belajar Paling Sering Digunakan Saat PJJ. Diambil kembali dari Kompas.com:

- <https://nasional.kompas.com/read/2020/10/16/18264341/survei-google-classroom-jadi-platform-belajar-paling-sering-digunakan-saat?page=all#page2>
- Kaur, C., & Sharma, A. (2020). *Twitter Sentiment Analysis on Coronavirus using Textblob EasyChair Preprint Twitter Sentiment Analysis on Coronavirus using Textblob Chhinder Kaur and Anand Sharma. March.*
- Khan, I. A., Woo, J., Seo, J. H., & Choi, J. T. (2015). Text mining: Extraction of interesting association rule with frequent itemsets mining for Korean language from unstructured data. *International Journal of Multimedia and Ubiquitous Engineering*, 10(11), 11–20. <https://doi.org/10.14257/ijmue.2015.10.11.02>
- Khan, R., Rustam, F., Kanwal, K., Mehmood, A., & Choi, G. S. (2021). US Based COVID-19 Tweets Sentiment Analysis Using TextBlob and Supervised Machine Learning Algorithms. *2021 International Conference on Artificial Intelligence, ICAI 2021*, 1–8. <https://doi.org/10.1109/ICAI52203.2021.9445207>
- Kotsiantis, S., & Kanellopoulos, D. (2006). Association Rules Mining: A Recent Overview. *Science*, 32(1), 71–82.
- Laksono, R. A., Sungkono, K. R., Sarno, R., & Wahyuni, C. S. (2019). Sentiment analysis of restaurant customer reviews on tripadvisor using naïve bayes. *Proceedings of 2019 International Conference on Information and Communication Technology and Systems, ICTS 2019*, 49–54. <https://doi.org/10.1109/ICTS.2019.8850982>
- Lim, M., Khudhair, R., & Ahmed, A. (2016). Data Mining Classification Comparison (Naïve Bayes and C4.5 Algorithms) Cite this paper Related papers Relevant Feature Selection Model Using Data Mining for Intrusion Detection System Ayman I Madbouly A Comprehensive Review of Current and Future. *International Journal of Engineering Trends and Technology (IJETT)*, 38. <http://www.ijettjournal.org>
- McCallum, A., & Nigam, K. (1998). A Comparison of Event Models for Naive Bayes Text Classification. *Learning for Text Categorization: Papers from the 1998 AAAI Workshop*, 41–48. <https://doi.org/10.1002/em.22125>
- Mishra, M., & Vishwakarma, S. K. (2017). Text Classification based on Association Rule Mining Technique. *International Journal of Computer Applications*, 169(10), 46–50. <https://doi.org/10.5120/ijca2017914905>
- Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). *Foundations of machine learning.* MIT press.

- Mubarok, M. S., Adiwijaya, A., & Aldhi, M. D. (2017). Aspect-based sentiment analysis to review products using Naïve Bayes. *AIP Conference Proceedings*, 1867(August). <https://doi.org/10.1063/1.4994463>
- Mythili, M. S., & Shavanas, A. M. (2013). Performance Evaluation of Apriori and FP-Growth Algorithms. *International Journal of Computer Applications*, 79(10), 34–37. <https://doi.org/10.5120/13779-1650>
- Najib, A. C., Irsyad, A., Qandi, G. A., & Rakhmawati, N. A. (2019). Perbandingan Metode Lexicon-based dan SVM untuk Analisis Sentimen Berbasis Ontologi pada Kampanye Pilpres Indonesia Tahun 2019 di Twitter. *Fountain of Informatics Journal*, 4(2), 41. <https://doi.org/10.21111/fij.v4i2.3573>
- Nofriani, N. (2019). Comparisons of Supervised Machine Learning Techniques in Predicting the Classification of the Household's Welfare Status. *Journal Pekommas*, 4(1), 43. <https://doi.org/10.30818/jpkm.2019.2040105>
- Peraturan Pemerintah No. 21 Tahun 2020* tentang Pembatasan Sosial Berskala Besar dalam Rangka Percepatan Penanganan *Corona Virus Disease* 2019 (COVID-19). Diakses tanggal 17 November 2021 dari https://jdih.setkab.go.id/PUUdoc/176085/PP_Nomor_21_Tahun_2020.pdf
- Perdana, K., Pricillia, T., & Zulfachmi. (2021). Optimasi Textblob Menggunakan Support Vector Machine Untuk Analisis Sentimen (Studi Kasus Layanan Telkomsel). *Jurnal Bangkit Indonesia*, 10(1), 13–15. <https://doi.org/10.52771/bangkitindonesia.v10i1.120>
- Pratama, E. E., & Atmi, R. L. (2020). A Text Mining Implementation Based on Twitter Data to. *Journal of Computers for Society*, 1(1), 91–100.
- Rácz, A., Bajusz, D., & Héberger, K. (2021). Effect of dataset size and train/test split ratios in qsar/qspr multiclass classification. *Molecules*, 26(4), 1–16. <https://doi.org/10.3390/molecules26041111>
- Rajman, M., & Besançon, R. (1998). Text Mining: Natural Language techniques and Text Mining applications. *Data Mining and Reverse Engineering*, 1998, 50–64. https://doi.org/10.1007/978-0-387-35300-5_3
- Ratnawati, F. (2018). Implementasi Algoritma Naive Bayes Terhadap Analisis Sentimen Opini Film Pada Twitter. *INOVTEK Polbeng - Seri Informatika*, 3(1), 50. <https://doi.org/10.35314/isi.v3i1.335>

- Ruswati, R., Gufroni, A. I., & Rianto, R. (2018). Associative Analysis Data Mining Pattern Against Traffic Accidents Using Apriori Algorithm. *Scientific Journal of Informatics*, 5(2), 91–104. <https://doi.org/10.15294/sji.v5i2.16199>
- Salloum, S. A., Al-Emran, M., Monem, A. A., & Shaalan, K. (2018). Using text mining techniques for extracting information from research articles. *Studies in Computational Intelligence*, 740, 373–397. https://doi.org/10.1007/978-3-319-67056-0_18
- Santoso, M. H. (2021). *Application of Association Rule Method Using Apriori Algorithm to Find Sales Patterns Case Study of Indomaret Tanjung Anom*. 1(2), 54–66.
- Sari, F. V., & Wibowo, A. (2019). Analisis Sentimen Pelanggan Toko Online Jd. Id Menggunakan Metode Naïve Bayes Classifier Berbasis Konversi Ikon Emosi. *Simetris: Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer*, 2(2), 681–686.
- Shah, P. (2020, Juni 28). Sentiment Analysis using TextBlob. Diambil kembali dari Medium: <https://towardsdatascience.com/my-absolute-go-to-for-sentiment-analysis-textblob-3ac3a11d524>
- Singh, G., Kumar, B., Gaur, L., & Tyagi, A. (2019). Comparison between Multinomial and Bernoulli Naïve Bayes for Text Classification. *2019 International Conference on Automation, Computational and Technology Management, ICACTM 2019, May 2020*, 593–596. <https://doi.org/10.1109/ICACTM.2019.8776800>
- Sungheetha, D. A., & Sharma, D. R. (2020). TransCapsule Model for Sentiment Classification. *Journal of Artificial Intelligence and Capsule Networks*, 2(3), 163–169. <https://doi.org/10.36548/jaicn.2020.3.003>
- Suryani, P. S. M., Linawati, L., & Saputra, K. O. (2019). Penggunaan Metode Naïve Bayes Classifier pada Analisis Sentimen Facebook Berbahasa Indonesia. *Majalah Ilmiah Teknologi Elektro*, 18(1), 145. <https://doi.org/10.24843/mite.2019.v18i01.p22>
- Talib, R., Hanif, M. K., Ayesha, S., & Fatima, F. (2016). *Text Mining: Techniques, Applications, and Issues*. 7(11), 414–418.
- Watson, G. (2004). The Legacy of Ishikawa. *Quality Progress*, 37(4), 54–57.
- Welsh, E. T., Wanberg, C. R., Brown, K. G., & Simmering, M. J. (2003). *E-learning: emerging uses, empirical results, and future directions*. 245–258.
- Wibawa, A. P., Kurniawan, A. C., Murti, D. M. P., Adiperkasa, R. P., Putra, S. M.,

- Kurniawan, S. A., & Nugraha, Y. R. (2019). Naïve Bayes Classifier for Journal Quartile Classification. *International Journal of Recent Contributions from Engineering, Science & IT (IJES)*, 7(2), 91. <https://doi.org/10.3991/ijes.v7i2.10659>
- Widiastuti, D., & Sofi, N. (2014). Analisis Perbandingan Algoritma Apriori Dan Fp-Growth Pada Transaksi Koperasi. *UG Jurnal Vol.*, 8(01), 21–24.
- Wu, X., Kumar, V., Ross, Q. J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Yu, P. S., Zhou, Z. H., Steinbach, M., Hand, D. J., & Steinberg, D. (2008). Top 10 algorithms in data mining. In *Knowledge and Information Systems* (Vol. 14, Issue 1). <https://doi.org/10.1007/s10115-007-0114-2>
- Xin, N. S., Shibghatullah, A. S., Subaramaniam, K. A., & Wahab, M. H. A. (2021). A Systematic Review for Online Learning Management System. *Journal of Physics: Conference Series*, 1874(1). <https://doi.org/10.1088/1742-6596/1874/1/012030>
- Xu, S. (2018). Bayesian Naïve Bayes classifiers to text classification. *Journal of Information Science*, 44(1), 48–59. <https://doi.org/10.1177/0165551516677946>
- Zanini, N., & Dhawan, V. (2015). Text Mining: An introduction to theory and some applications. *Research Matters*, 19(June), 38–44. https://www.researchgate.net/profile/Nadir_Zanini/publication/304140500_Text_Mining_An_introduction_to_theory_and_some_applications/links/5767cfeb08ae1658e2f89662.pdf
- Zhang, L., Wang, S., & Liu, B. (2018). Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4), 1–25. <https://doi.org/10.1002/widm.1253>

LAMPIRAN

Import Modules

```
import string
import pandas as pd
import numpy as np
import xgboost as xgb
```

Read Data

```
data = pd.read_csv('Dataset_new.csv', sep=',', encoding='utf-8')
data.head()
```

Case Folding

```
import re
def casefolding(Review):
    Review = Review.lower()
    return Review
data['Review'] = data['Review'].apply(casefolding)
data.head()
```

Remove Punctuation

```
import re
def remove_punctuation(Review):
    Review = Review.strip(" ")
    Review = Review.replace("''", "")
    Review = re.sub(r'[?.$|.!^_::"](-+,]', "", Review)
    return Review
data['Review'] = data['Review'].apply(remove_punctuation)
data.head()
```

Non-ASCII Character Removal

```
def strip_non_ascii(string):
```



```

    """ Returns the string without non ASCII characters"""
    stripped = (c for c in string if 0 < ord(c) < 127)
    return "".join(stripped)
data['Review'] = data['Review'].apply(strip_non_ascii)
data.head()

```

Emoji Removal

```

import emoji

def remove_emoji(string):
    emoji_pattern = re.compile("[
        u"\U0001F600-\U0001F64F"
        u"\U0001F300-\U0001F5FF"
        u"\U0001F680-\U0001F6FF"
        u"\U0001F1E0-\U0001F1FF"
        u"\U00002702-\U000027B0"
        u"\U000024C2-\U0001F251"
    ]+", flags=re.UNICODE)
    return emoji_pattern.sub(r'', string)
    return data
data['Review'] = data['Review'].apply(remove_emoji)
data.head()

```

Tokenizing

```

def token(Review):
    nstr = Review.split(' ')
    dat= []
    a = -1
    for hu in nstr:
        a = a + 1
    if hu == "":
        dat.append(a)

```

```

p = 0
b = 0
for q in dat:
    b = q - p
    del nstr[b]
    p = p + 1
return nstr
data['Review'] = data['Review'].apply(token)
data.head()

```

Stop Word Removal

```

import nltk
stopwords = nltk.corpus.stopwords.words('english')
stopwords[0:10]
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're"]
def remove_stopwords(Review):
    output= [i for i in Review if i not in stopwords]
    return output
data['Review']= data['Review'].apply(lambda x:remove_stopwords(x))
data.head()

```

Lemmatization

```

import spacy
nlp = spacy.load('en_core_web_sm')

def lemmatize(text):
    """Perform lemmatization and stopword removal in the clean text
    Returns a list of lemmas
    """
    doc = nlp(text)
    lemma_list = [str(tok.lemma_).lower() for tok in doc
                   if tok.is_alpha and tok.text.lower() not in stopwords]

```

```

return lemma_list
data['Review'] = data[Review].apply(lemmatize)
data.head()

```

Data Labelling

```

from textblob import TextBlob

def getSubjectivity(review):
    return TextBlob(review).sentiment.subjectivity

def getPolarity(review):
    return TextBlob(review).sentiment.polarity

def analyze(score):
    if score < 0:
        return 'Negative'
    elif score == 0:
        return 'Neutral'
    else:
        return 'Positive'

data['Subjectivity'] = data['Review'].apply(getSubjectivity)
data['Polarity'] = data['Review'].apply(getPolarity)
data['TextBlob'] = data['Polarity'].apply(analyze)
data.head()

tb_counts = data.TextBlob.value_counts()
tb_counts

plt.figure(figsize=(10, 7))
plt.pie(tb_counts.values, labels=tb_counts.index, explode=(0,0,0.25),
autopct='% 1.1f%%', shadow=False)

```

```
plt.figure(figsize=(8,6))
for i in range(0, data.shape[0]):
    plt.scatter(data['Polarity'][i],data['Subjectivity']
               [i],color='Blue')
plt.title('Sentiment Analysis')
plt.xlabel('Polarity')
plt.ylabel('Subjectivity')
plt.show()
```

Define Data Type

```
data_clean = data_clean.astype({'Value' : 'category'})
data_clean = data_clean.astype({'Review' : 'string'})
data_clean.dtypes
```

TF-IDF

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
```

```
tf = TfidfVectorizer()
text_tf = tf.fit_transform(data_clean['Review'].astype('U'))
text_tf
```

Splitting Data

```
from sklearn.model_selection import train_test_split
```

```
# ukuran 50/50
```

```
X_train, X_test, y_train, y_test = train_test_split(text_tf, data_clean['Value'],
test_size=0.5, random_state = 42)
```

```
# ukuran 60/40
```

```
X_train, X_test, y_train, y_test = train_test_split(text_tf, data_clean['Value'],
test_size=0.4, random_state = 42)
```

```
# ukuran 70/30
```

```
X_train, X_test, y_train, y_test = train_test_split(text_tf, data_clean['Value'],
test_size=0.3, random_state = 42)
# ukuran 80/20
X_train, X_test, y_train, y_test = train_test_split(text_tf, data_clean['Value'],
test_size=0.2, random_state = 42)
```

Algoritma Naïves Bayes

```
from sklearn.naive_bayes import MultinomialNB
from sklearn.Metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.Metrics import classification_report
from sklearn.Metrics import confusion_matrix

clf = MultinomialNB().fit(X_train, y_train)
predicted = clf.predict(X_test)
print("MultinomialNB accuracy:" , accuracy_score(y_test,predicted))
print("MultinomialNB Precision:" , precision_score(y_test,predicted, average="binary",
pos_label="NEGATIF"))
print("MultinomialNB Recall:" , recall_score(y_test,predicted, average="binary",
pos_label="NEGATIF"))
print("MultinomialNB f1_score:" , f1_score(y_test,predicted, average="binary",
pos_label="NEGATIF"))

print(f'confusion matrix:\n{confusion_matrix(y_test, predicted)}')
print('=====\n')
print(classification_report(y_test, predicted, zero_division=0))
```