

**ANALISIS PERMASALAHAN PADA APLIKASI  
SMARTPHONE KAI ACCESS BERDASARKAN *END-USER*  
*REVIEW* MENGGUNAKAN METODE *TEXT-MINING*  
DAN *FISHBONE DIAGRAM***

**TUGAS AKHIR**

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



**Disusun oleh:**

Nama : Alfian Willy Saputra

NIM : 16522062

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

**2020**

## PERNYATAAN KEASLIAN TUGAS AKHIR

### PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Alfian Willy Saputra

NIM : 16522062

Dengan tugas akhir berjudul:

### **ANALISIS PERMASALAHAN PADA APLIKASI SMARTPHONE KAI ACCESS BERDASARKAN *END-USER* *REVIEW* MENGGUNAKAN METODE *TEXT-MINING* DAN *FISHBONE DIAGRAM***

Dengan ini, saya sebagai penulis menyatakan bahwa tugas akhir ini merupakan hasil penelitian saya sendiri dengan beberapa kutipan yang telah saya cantumkan sumbernya. Bila di kemudian hari ditemukan kesamaan ataupun penyimpangan yang merugikan orang lain, saya bersedia menerima konsekuensi yang diberikan oleh Universitas Islam Indonesia dengan penuh rasa tanggung jawab.

Yogyakarta, 7 Desember 2020



(Alfian Willy Saputra)

16522062

## LEMBAR PENGESAHAN PEMBIMBING

### LEMBAR PENGESAHAN PEMBIMBING

ANALISIS PERMASALAHAN PADA APLIKASI *SMARTPHONE* KAI ACCESS  
BERDASARKAN *END-USER REVIEW* MENGGUNAKAN METODE *TEXT-MINING*  
DAN *FISHBONE DIAGRAM*

ISLAM  
TUGAS AKHIR

Disusun oleh:

Nama : Alfian Willy Saputra  
NIM : 16 522 062

Yogyakarta, 7 Desember 2020

Dosen Pembimbing

UNIVERSITAS INDONESIA  
الجامعة الإسلامية  
Indonesian Islamic University

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

NIP. 155220110

## LEMBAR PENGESAHAN PENGUJI TUGAS AKHIR

### LEMBAR PENGESAHAN PENGUJI TUGAS AKHIR

**ANALISIS PERMASALAHAN PADA APLIKASI SMARTPHONE KAI ACCESS  
BERDASARKAN END-USER REVIEW MENGGUNAKAN METODE TEXT-MINING**

**DAN FISHBONE DIAGRAM**

Disusun oleh:

Nama : Alfian Willy Saputra

NIM 16522062

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

Yogyakarta, 28 Desember 2020

#### Tim Penguji

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

Ketua

Ir. Erlangga Fausa, MCIS

Anggota I

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

Anggota II

Mengetahui

Ketua Program Studi Teknik Industri

Universitas Islam Indonesia



Dr. Taufiq Anmawan, S.T., M.M. —

## HALAMAN PERSEMBAHAN

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Tugas akhir ini saya persembahkan kepada Allah *Subhanahu wata'ala*, kedua orang tua beserta adik tercinta, dan kerabat dekat yang telah memberikan motivasi yang begitu berarti kepada saya.

Tak lupa saya persembahkan pula kepada seluruh dosen dan karyawan Universitas Islam Indonesia yang telah mencurahkan ilmu dan bantuan selama saya mengenakan alma mater Universitas Islam Indonesia.

الجامعة الإسلامية  
الاستاذة الأستاذة

## MOTTO

فَإِنَّ مَعَ الْعُسْرِ يُسْرًا ۖ إِنَّ مَعَ الْعُسْرِ يُسْرًا ۖ

“Maka sesungguhnya beserta kesulitan ada kemudahan, sesungguhnya beserta kesulitan ada kemudahan”

(Q.S. Al-Insyirah ayat 5 – 6)

لَا يُكَلِّفُ اللَّهُ نَفْسًا إِلَّا وُسْعَهَا

“Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya”

(Q.S. Al-Baqarah ayat 286)

لَقَدْ خَلَقْنَا الْإِنْسَانَ فِي أَحْسَنِ تَقْوِيمٍ

“Sungguh, Kami telah menciptakan manusia dalam bentuk yang sebaik-baiknya”

(Q.S. At-Tin ayat 4)

*“Protect what you hold dear with your own hands”*

## KATA PENGANTAR



*Assalamu 'alaikum warahmatullahi wabarakatuh*

Alhamdulillah rabbil'alamin, puji dan syukur penulis panjatkan kehadirat Allah *Subhanahu wata'ala* karena berkat rahmat, hidayah dan karunia-Nya penulis dapat menyusun laporan Tugas Akhir dengan judul **“Analisis Permasalahan pada Aplikasi Smartphone KAI Access Berdasarkan End-User Review Menggunakan Metode Text Mining dan Fishbone Diagram”** ini dengan lancar. Shalawat serta salam tak lupa penulis sanjungkan kepada Nabi Muhammad *Shallallahu 'alaihi wasallam* beserta keluarga, sahabat dan pengikut-pengikutnya yang telah memberikan syafaat-Nya kepada umat manusia.

Adapun Tugas Akhir ini tidak akan terselesaikan dengan baik tanpa adanya masukan, bantuan dan dukungan dari berbagai pihak. Oleh karena itu, dalam kesempatan ini penulis dengan rendah hati menyampaikan ucapan dan rasa terima kasih kepada yang terhormat:

1. Bapak Prof. Dr. Ir. Hari Purnomo, M.T. selaku Dekan Fakultas Teknologi Industri, Universitas Islam Indonesia.
2. Bapak Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D. selaku Ketua Jurusan Teknik Industri Fakultas Teknologi Industri, Universitas Islam Indonesia.
3. Bapak Dr. Taufiq Immawan, S.T., M.M. selaku Ketua Program Studi Teknik Industri Fakultas Teknologi Industri, Universitas Islam Indonesia.

4. Bapak Andrie Pasca Hendradewa, S.T., M.T. selaku dosen pembimbing Tugas Akhir yang telah memberikan bimbingan dan motivasi kepada penulis.
5. Ibu, Bapak dan Adik penulis yang tiada hentinya memberikan do'a, kasih sayang dan dukungan yang tak ternilai sehingga penulis mampu menyelesaikan Tugas Akhir dengan baik dan lancar.
6. Huzein Mulyawan, Muhammad Agung Pratama, dan Munawar Hakki selaku teman dekat seperjuangan sehobi sejak semester tiga.
7. Teman-teman mahasiswa Teknik Industri Universitas Islam Indonesia, khususnya angkatan 2016.

Penulis berharap semoga seluruh pihak yang terlibat tetap diberikan nikmat jasmani dan rohani, serta dilancarkan segala urusannya oleh Allah SWT.

Dalam penulisan Tugas Akhir ini, penulis menyadari masih banyak kekurangan yang perlu diberikan kritik dan saran yang membangun. Oleh karena itu, penulis memohon maaf kepada pihak yang terlibat serta pembaca Tugas Akhir ini. Namun terlepas dari itu semua, penulis berharap Tugas Akhir ini dapat memberikan manfaat bagi penulis dan para pembaca.

*Wassalamu'alaikum warahmatullahi wabarakatuh*

Yogyakarta, 7 Desember 2020

Alfian Willy Saputra



## ABSTRAK

Perkembangan teknologi yang begitu cepat mendorong manusia untuk ikut beradaptasi dengan segala macam perubahan. Hal ini mendorong para perusahaan di berbagai sektor untuk menyesuaikan proses bisnis mereka terhadap *trend* perkembangan teknologi. Salah satunya yakni PT. Kereta Api Indonesia (Persero) yang merilis aplikasi *smartphone* KAI Access agar mempermudah pengguna jasa kereta api. Namun ternyata banyak pengguna KAI Access menyuarakan keluhannya di Google Play Store. Penelitian ini bertujuan untuk mengetahui permasalahan KAI Access berdasarkan ulasan pengguna, meneliti akar permasalahan menggunakan *fishbone diagram* serta memberikan saran perbaikan. Dari hasil analisis ulasan, diperoleh delapan kata kunci permasalahan atau keluhan utama pengguna aplikasi KAI Access, yakni permasalahan “bad”, “email”, “error”, “fix”, “login”, “ticket”, “time”, dan “update”. Kemudian berdasarkan analisis *association rules*, terdapat 45 *rules* yang berhubungan dengan kata kunci permasalahan. Beberapa penyebab permasalahan di antaranya terdapat masalah yang berkaitan dengan pencarian, pemesanan hingga pembayaran tiket, terdapat keluhan terkait *server*, serta masalah terkait akun pengguna. Solusi penyelesaian masalah yang dapat dilakukan yakni melakukan diskusi atau komunikasi antara pihak KAI Access dan pengguna aplikasi, sehingga kedua belah pihak dapat menyelaraskan informasi serta mengetahui letak kesalahan dari permasalahan tersebut. Selanjutnya, pengecekan hingga perbaikan *server* ataupun sistem informasi internal aplikasi dapat dilakukan sebagai tindak lanjut atas diskusi yang telah dilakukan.

Kata kunci: *Fishbone diagram*, KAI Access, *text mining*, *user review*, *association rules*

## DAFTAR ISI

PERNYATAAN KEASLIAN TUGAS AKHIR.....	ii
LEMBAR PENGESAHAN PEMBIMBING.....	iii
LEMBAR PENGESAHAN PENGUJI TUGAS AKHIR .....	iv
HALAMAN PERSEMBAHAN .....	v
MOTTO .....	vi
KATA PENGANTAR .....	vii
ABSTRAK.....	ix
DAFTAR ISI.....	x
DAFTAR TABEL.....	xii
DAFTAR GAMBAR .....	xiii
<b>BAB I PENDAHULUAN.....</b>	<b>15</b>
1.1 Latar Belakang.....	15
1.2 Rumusan Masalah.....	19
1.3 Tujuan Penelitian .....	19
1.4 Batasan Masalah .....	20
1.5 Manfaat Penelitian .....	20
1.6 Sistematika Penulisan .....	20
<b>BAB II KAJIAN PUSTAKA.....</b>	<b>22</b>
2.1 Kajian Deduktif .....	22
2.2 Kajian Induktif.....	37
<b>BAB III METODE PENELITIAN .....</b>	<b>44</b>
3.1 Jenis Data.....	44
3.2 Prosedur Penelitian .....	44
3.3 Diagram Alir.....	49
<b>BAB IV PENGUMPULAN DAN PENGOLAHAN DATA.....</b>	<b>51</b>
4.1 Pengumpulan Data.....	51
4.2 Pengolahan Data .....	52
<b>BAB V PEMBAHASAN.....</b>	<b>77</b>
5.1 <i>Word Cloud</i> dan <i>Association Rules</i> .....	77

5.2	<i>Fishbone Diagram</i> .....	77
BAB VI PENUTUP .....		82
6.1	Kesimpulan .....	82
6.2	Saran .....	83
DAFTAR PUSTAKA .....		84
LAMPIRAN.....		88



## DAFTAR TABEL

Tabel 2.1 Perbandingan fokus penelitian .....	41
Tabel 4.1 <i>Association rules</i> terkait kata kunci permasalahan .....	70
Tabel 4.2 <i>Association rules</i> pendukung cabang penyebab lanjutan.....	76



## DAFTAR GAMBAR

Gambar 1.1 Grafik Penumpang Kereta Api Periode 2012 – 2019.....	17
Gambar 1.2 <i>Rating</i> KAI Access per 10 April 2020 .....	18
Gambar 2.1 Tampilan beranda KAI Access .....	24
Gambar 2.2 Proses <i>Knowledge Discovery in Database</i> .....	25
Gambar 2.3 Perbandingan parameter <i>supervised</i> dan <i>unsupervised learning</i> .....	29
Gambar 2.4 <i>Fishbone Diagram</i> .....	31
Gambar 2.5 <i>Check Sheet</i> .....	32
Gambar 2.6 <i>Diagram Pareto</i> .....	33
Gambar 2.7 <i>Histogram</i> .....	34
Gambar 2.8 <i>Control Chart</i> .....	34
Gambar 2.9 <i>Scatter Diagram</i> .....	35
Gambar 2.10 Pengelompokan Data ( <i>Stratification</i> ) pada <i>Scatter Diagram</i> .....	36
Gambar 3.1 Diagram alir penelitian.....	50
Gambar 4.1 Hasil <i>review scraping</i> dalam format csv (bagian 1).....	51
Gambar 4.2 Hasil <i>review scraping</i> dalam format csv (bagian 2).....	52
Gambar 4.3 <i>Data inputting</i> menggunakan modul <i>pandas</i> .....	52
Gambar 4.4 <i>Tokenization</i> dan <i>stop word removal</i> .....	53
Gambar 4.5 Proses eliminasi emoji.....	54
Gambar 4.6 Perbandingan ulasan bersih dengan ulasan awal ( <i>raw</i> ).....	54
Gambar 4.7 Proses pelabelan ulasan berdasarkan <i>rating</i> .....	55
Gambar 4.8 Proses pemilahan ulasan berdasarkan label .....	56
Gambar 4.9 Proses prediksi dan pelabelan ulasan berdasarkan bahasa .....	57
Gambar 4.10 Proses pengecekan frekuensi penggunaan bahasa .....	57
Gambar 4.11 Pemilahan ulasan berlabel EN.....	58
Gambar 4.12 Pemilahan ulasan berlabel ID – MS – JV .....	58
Gambar 4.13 <i>Word clouding</i> untuk ulasan berlabel EN .....	59
Gambar 4.14 <i>Word cloud</i> dari ulasan berlabel EN .....	59
Gambar 4.15 <i>Word clouding</i> untuk ulasan berlabel ID – MS – JV .....	60
Gambar 4.16 <i>Word cloud</i> dari ulasan berlabel ID – MS – JV .....	60
Gambar 4.17 Perhitungan frekuensi kata berlabel EN.....	61
Gambar 4.18 Perhitungan frekuensi kata berlabel ID – MS – JV.....	61
Gambar 4.19 Tahap penentuan <i>unique words</i> (langkah pertama).....	62
Gambar 4.20 Tahap penentuan <i>unique words</i> (langkah kedua).....	63
Gambar 4.21 Penerapan <i>one hot encoding</i> .....	63
Gambar 4.22 Hasil proses <i>one hot encoding</i> dari ulasan berlabel EN.....	64
Gambar 4.23 Proses analisis <i>association rules</i> .....	64
Gambar 4.24 <i>Association rules</i> pada ulasan berlabel EN (bagian 1).....	65
Gambar 4.25 <i>Association rules</i> pada ulasan berlabel EN (bagian 2).....	66
Gambar 4.26 <i>Association rules</i> pada ulasan berlabel EN (bagian 3).....	66

Gambar 4.27 <i>Association rules</i> pada ulasan berlabel ID – MS – JV (bagian 1)...	67
Gambar 4.28 <i>Association rules</i> pada ulasan berlabel ID – MS – JV (bagian 2)...	68
Gambar 4.29 <i>Association rules</i> pada ulasan berlabel ID – MS – JV (bagian 3)...	69
Gambar 4.30 <i>Association rules</i> pada ulasan berlabel ID – MS – JV (bagian 4)...	70
Gambar 4.31 <i>Fishbone diagram</i> untuk kata kunci “bad” .....	72
Gambar 4.32 <i>Fishbone diagram</i> untuk kata kunci “email” .....	73
Gambar 4.33 <i>Fishbone diagram</i> untuk kata kunci “error” .....	73
Gambar 4.34 <i>Fishbone diagram</i> untuk kata kunci “fix” .....	74
Gambar 4.35 <i>Fishbone diagram</i> untuk kata kunci “login” .....	74
Gambar 4.36 <i>Fishbone diagram</i> untuk kata kunci “ticket” .....	75
Gambar 4.37 <i>Fishbone diagram</i> untuk kata kunci “update” .....	75
Gambar 4.38 <i>Fishbone diagram</i> untuk kata kunci “pembayaran” .....	76
Gambar A.1 <i>Interface</i> aplikasi AppFollow .....	88
Gambar A.2 <i>Interface</i> Jupyter Notebook .....	88
Gambar A.3 <i>Home interface</i> pada AppFollow .....	89
Gambar A.4 Langkah pencarian aplikasi KAI Access .....	89
Gambar A.5 Pilihan "Manage Reviews" pada <i>pop-up</i> .....	90
Gambar A.6 Opsi "Filter by Reviews" .....	90
Gambar A.7 <i>Download file as csv</i> pada AppFollow .....	91
Gambar A.8 Membuka lembar kerja baru pada Jupyter .....	91
Gambar A.9 Contoh <i>data inputting</i> menggunakan <i>pandas</i> .....	92
Gambar A.10 Proses <i>tokenization</i> dan <i>case folding</i> .....	92
Gambar A.11 Penerapan <i>stop word removal</i> .....	92
Gambar A.12 Tahap eliminasi emoji menggunakan modul “unicodedata” .....	92
Gambar A.13 Penerapan <i>word cloud</i> dan <i>output framing</i> .....	93
Gambar A.14 Contoh hasil <i>word clouding</i> .....	93
Gambar A.15 <i>Association rules</i> pada Python .....	93

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan teknologi yang begitu cepat mendorong manusia untuk ikut beradaptasi dengan segala macam perubahan. Sebagai contoh yaitu berbagai mesin manufaktur yang mampu mengurangi biaya dan waktu produksi, kemudian teknologi informasi seperti *internet* yang mampu “menjangkau dunia”. Hal ini mendorong para perusahaan di berbagai sektor untuk menyesuaikan proses bisnis mereka terhadap *trend* perkembangan teknologi agar tidak tergerus oleh zaman. Pada era revolusi industri ketiga atau yang dikenal dengan istilah *Industry 3.0*, penerapan mesin hingga robot memberikan dampak yang signifikan bagi aspek efektivitas dan efisiensi suatu industri. Setelah *internet* menjadi bagian dari tatanan kehidupan, pada tahun 2015 muncul gagasan dari ilmuwan dan pemerintah Jerman untuk menggabungkan teknologi *automation* dengan *cyber* (Baenanda, 2019). Implementasinya tidak hanya di bidang produksi/*manufacturing* saja, namun merambah hingga ke bidang *logistics*, *marketing*, dan *human resource management*. Segala perubahan yang signifikan ini memunculkan era baru dalam perindustrian, yang dikenal dengan istilah *Industry 4.0*. Menurut Menteri Perindustrian, Airlangga Hartanto, revolusi industri terbaru ini mengintegrasikan manusia serta *automation* ke dalam *internet of things* (IoT) sehingga meningkatkan efisiensi yang umumnya terimplementasi dari pertukaran data dan informasi yang semakin banyak dan cepat (Debora, 2019).

Salah satu pemanfaatan teknologi internet di dunia usaha adalah *e-commerce*, suatu model bisnis yang mempertemukan penjual dan pembeli di dalam dunia maya, serta melakukan transaksi jual beli melalui media aplikasi yang terhubung oleh *internet*. Dengan akses *internet* yang semakin mudah diperoleh serta media *cyber* yang dapat dimiliki dengan harga terjangkau, maka tidak heran apabila semakin banyak pelaku industri hingga wirausaha yang beralih menerapkan *e-commerce* dalam memasarkan produk maupun jasanya (Bloomenthal, 2019). Dalam survey yang dilakukan oleh Global Web Index, dikutip dari Kemp dan Moey (2019), tingkat aktivitas masyarakat Indonesia dalam bertransaksi berbasis *e-commerce* pada bulan September 2019 mencapai level tertinggi dibandingkan negara lainnya. Sekitar 90% responden yang merupakan warga Indonesia berusia 16 – 64 tahun sudah pernah membeli produk ataupun jasa secara *online*. Hal ini menegaskan peranan *e-commerce* yang mampu mengubah pandangan masyarakat terhadap cara bertransaksi, dengan berbagai kemudahan hingga keamanannya. Berdasarkan pihak yang terlibat, *e-commerce* dapat dibedakan menjadi enam macam, dengan tiga di antaranya yang paling dikenal masyarakat yakni *Business-to-Consumers (B2C) E-commerce*, *Consumer-to-Consumer (C2C) E-commerce*, dan *Online-to-Offline (O2O) E-commerce*. Beberapa contoh yang terkenal dari B2C *e-commerce* di antaranya Amazon, Lazada, dan Traveloka, sedangkan contoh dari C2C *e-commerce* yakni Tokopedia, Shopee, Bukalapak hingga OLX. Kemudian contoh dari O2O *e-commerce* yakni Gojek, Grab, Airbnb dan RedDoorz (Desra, 2019).

Dalam rangka mengikuti *trend* bisnis tersebut, PT. Kereta Api Indonesia (Persero), atau disingkat PT. KAI, sebagai perusahaan milik negara yang bergerak di bidang jasa perkeretaapian Indonesia dengan basis pengguna kereta api yang meningkat dalam tujuh tahun terakhir (ditunjukkan pada Gambar 1.1), juga turut melakukan pengembangan layanan berbasis aplikasi *online* demi memudahkan para pengguna jasanya, dengan wujud berupa peluncuran aplikasi *smartphone* KAI Access. Aplikasi KAI Access memiliki fitur pemesanan tiket, *ticket rescheduling*, pembatalan tiket hingga *e-boarding pass*. Terdapat tiga keunggulan KAI Access bila dibandingkan dengan pesaingnya, yakni penerapan *e-boarding* dimana



pengguna hanya perlu menunjukkan *barcode* (yang diperoleh setelah pembelian tiket) ke alat pemindai sehingga tidak perlu lagi mencetak tiket. Selanjutnya yakni pemesanan kereta api lokal secara *online* hanya dapat dilakukan di KAI Access. Kemudian dari segi finansial, pembelian tiket tidak dikenakan biaya administrasi sebesar Rp 7.500 dimana biaya tersebut diterapkan untuk penyedia layanan lain seperti Tokopedia dan Alfamart. KAI Access dapat diunduh melalui layanan Google Play Store bagi pengguna perangkat Android. Dengan fitur-fitur yang ditawarkan, maka menurut Satriya Andoyo selaku Vice President E-Commerce PT. KAI, pengguna KAI Access bertambah sekitar 6.600 orang per hari (Martaon, 2019). Di lain kesempatan, Riko Oktora selaku E-Commerce Business Development PT. KAI menuturkan bahwa pengguna aktif KAI Access per tanggal 6 Desember 2019 mencapai sekitar dua juta akun (Budianto, 2019).

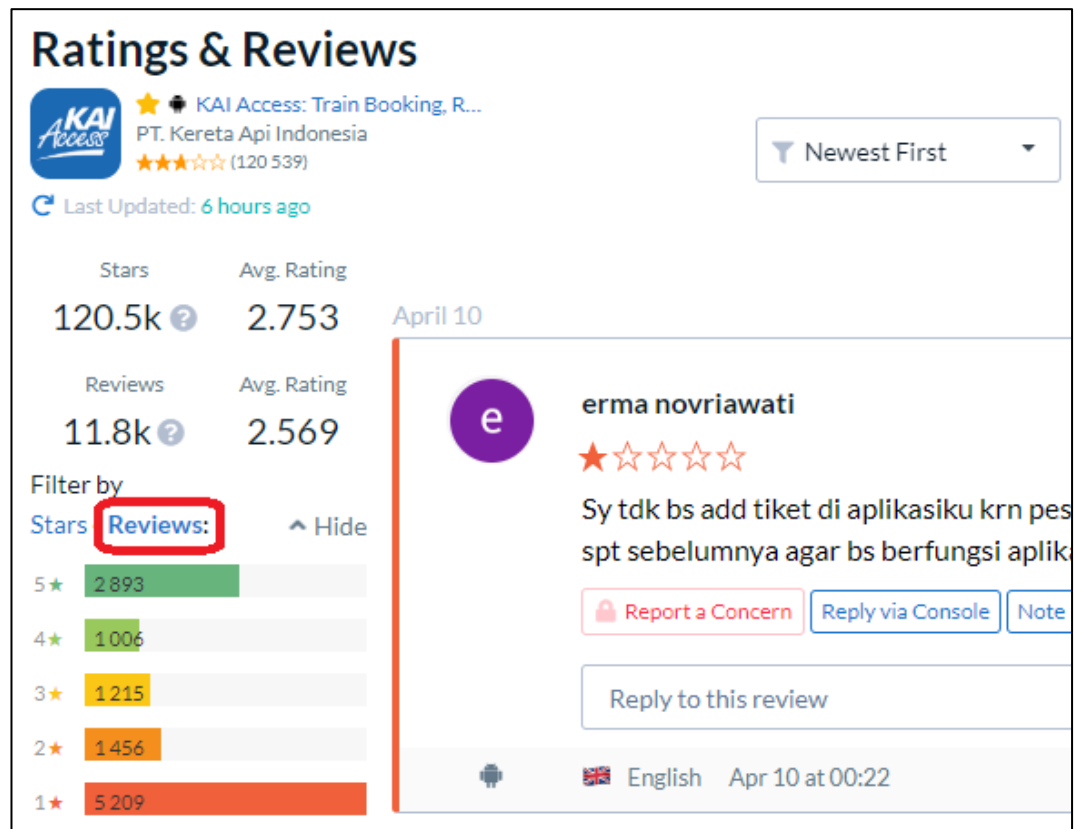


Gambar 1.1 Grafik Penumpang Kereta Api Periode 2012 – 2019

Sumber: Badan Pusat Statistik (2020)

Namun hingga awal tahun 2020, banyak pengguna aplikasi KAI Access yang mengeluhkan sulitnya menggunakan beberapa fitur yang disediakan. Tercatat hingga 9 April 2020, aplikasi KAI Access hanya memperoleh *rating* 2,7 dari 5 pada Google Play Store, yang ditampilkan pada Gambar 1.1. *Rating* tersebut lebih rendah bila dibandingkan dengan aplikasi lain dengan tawaran layanan yang relatif sama,

seperti Tokopedia dan tiket.com, dimana kedua aplikasi tersebut memperoleh *rating* di kisaran 4 (empat) pada periode yang sama. Sebagian besar pengguna memberikan *rating* 1 (satu) dikarenakan berbagai alasan, seperti kesulitan dalam *login* ke dalam aplikasi, sedikitnya pilihan pembayaran, *user interface* yang dianggap sulit dipahami hingga notifikasi *error* yang tidak jelas.



Gambar 1.2 Rating KAI Access per 10 April 2020

Untuk mengetahui permasalahan yang sering dikeluhkan pengguna KAI Access, salah satu caranya yakni dengan menggali informasi pada kumpulan ulasan pengguna yang dapat diperoleh dari Google Play Store, dengan metode yang digunakan adalah *text mining*. Termasuk dalam ranah *data mining*, *text mining* merupakan kegiatan memperoleh suatu informasi baru yang bersumber dari kumpulan kalimat atau teks yang tidak terstruktur, seperti percakapan dan opini masyarakat. Dikatakan tidak terstruktur karena program komputer yang ada belum mampu memahami *natural language* (bahasa manusia) secara utuh. Umumnya tahapan *text mining* dimulai dengan proses restrukturisasi data seperti penguraian

kalimat (*parsing or tokenizing*), dan eliminasi kata yang tidak mengandung informasi (*removing unnecessary words*) seperti kata sambung dan imbuhan. Data yang telah ditransformasi dapat menjadi *input* bagi sistem *machine learning* untuk dianalisis sehingga mampu memperoleh suatu informasi yang relevan dengan kebutuhan (Nisbet, 2009).

Dari uraian di atas, perlu dilakukan penelitian lebih lanjut terkait “Analisis Permasalahan pada Aplikasi *Smartphone* KAI Access Berdasarkan *End-User Review* Menggunakan Metode *Text Mining* dan *Fishbone Diagram*” dalam bentuk skripsi. Penelitian ini bertujuan untuk mengetahui permasalahan apa saja yang dikeluhkan oleh pengguna aplikasi KAI Access, serta meneliti akar permasalahan (*root cause*) dari keluhan pengguna tersebut berdasarkan hasil analisis *fishbone diagram* sehingga dapat menjadi referensi perbaikan dan pengembangan pada suatu sistem informasi ataupun fitur pada aplikasi. Dengan demikian, aplikasi KAI Access diharapkan tetap mampu menyesuaikan pelayanannya terhadap perilaku konsumen dalam menggunakan *e-commerce* pada umumnya sehingga dapat meningkatkan kepuasan konsumen.

## **1.2 Rumusan Masalah**

Setelah menjabarkan permasalahan yang terjadi dan fokus utama yang akan diteliti, maka rumusan permasalahan yang akan dihadapi dalam penelitian ini adalah:

1. Apa saja permasalahan yang sering dikeluhkan pengguna aplikasi *smartphone* KAI Access?
2. Bagaimana analisis akar permasalahan pada tiap keluhan pengguna aplikasi?

## **1.3 Tujuan Penelitian**

Berdasarkan rumusan masalah yang telah disusun, diperoleh tujuan dari penelitian ini yaitu:

1. Mengetahui permasalahan yang sering dikeluhkan pengguna aplikasi, sehingga dapat dilakukan analisis akar permasalahannya.

2. Mengetahui apa saja akar masalah pada tiap keluhan pengguna aplikasi.

#### **1.4 Batasan Masalah**

Adapun batasan masalah yang terdapat dalam penelitian ini adalah sebagai berikut:

1. Analisis yang digunakan menggunakan ulasan yang bersumber dari Google Play Store.
2. Ulasan (*review*) yang digunakan dalam penelitian ini mencakup rentang waktu dari 16 Juli 2014 hingga 10 April 2020.

#### **1.5 Manfaat Penelitian**

Manfaat dari penelitian ini yaitu dapat memberikan saran terkait perubahan maupun penambahan fitur dari aplikasi *smartphone* KAI Access agar mampu memberikan kemudahan bagi para pengguna tetap maupun pengguna baru, serta menyesuaikan fitur aplikasi dengan *trend* masyarakat saat ini.

#### **1.6 Sistematika Penulisan**

Sistematika penulisan dijadikan acuan dalam rangka melakukan penulisan penelitian yang terstruktur, dengan urutan dalam penelitian sebagai berikut.

#### **BAB I PENDAHULUAN**

Memuat penulisan latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

#### **BAB II KAJIAN PUSTAKA**

Memuat kajian literatur deduktif dan induktif yang menguatkan topik penelitian sehingga memenuhi syarat dan kriteria.

#### **BAB III METODE PENELITIAN**

Memuat tahapan penelitian yang dilakukan dalam penelitian secara terstruktur dari obyek penelitian, metode pengumpulan data, pengolahan data yang digunakan, dan disertai *flowchart* penelitian.

#### BAB IV PENGUMPULAN DAN PENGOLAHAN DATA

Membahas tentang proses pengumpulan data hingga pengolahan data dengan menggunakan metode yang digunakan, serta menjadi tolak ukur untuk melanjutkan pembahasan hasil penelitian.

#### BAB V PEMBAHASAN

Memuat analisis terkait pengolahan data yang telah dilakukan pada bab sebelumnya dengan menyesuaikan pada tujuan, teori, dan alur penelitian sehingga dapat diperoleh kesimpulan dan saran yang tepat.

#### BAB VI PENUTUP

Memuat kesimpulan dan saran dari hasil penelitian yang telah dilakukan untuk direkomendasikan pada objek yang diteliti, serta untuk penelitian selanjutnya.



## **BAB II**

### **KAJIAN PUSTAKA**

#### **2.1 Kajian Deduktif**

##### **2.1.1 *E-commerce***

*E-commerce* merupakan suatu model bisnis yang mempertemukan penjual dan pembeli di dalam dunia maya, serta melakukan transaksi jual beli melalui media *internet*. Dengan akses *internet* yang semakin mudah diperoleh serta media *cyber* yang dapat dimiliki dengan harga terjangkau, maka tidak heran jika semakin banyak pelaku industri hingga wirausahawan yang beralih menerapkan *e-commerce* dalam memasarkan produk maupun jasanya (Bloomenthal, 2019). Berdasarkan penelitian yang dilakukan oleh Kemp dan Moey (2019), tingkat aktivitas masyarakat Indonesia dalam bertransaksi berbasis *e-commerce* pada bulan September 2019 mencapai level tertinggi dibandingkan negara lainnya. Sekitar 90% masyarakat Indonesia yang berusia 16 – 64 tahun sudah pernah membeli produk ataupun jasa secara *online*. Hal ini menegaskan peranan *e-commerce* yang mampu mengubah pandangan masyarakat terhadap cara bertransaksi, dengan berbagai kemudahan hingga keamanannya.

##### **2.1.2 *Customer Satisfaction***

Kepuasan pelanggan menurut Hill et al. (2007) merupakan penilaian konsumen terhadap suatu produk secara keseluruhan, yang berkorelasi positif dengan loyalitas konsumen. Hal ini selaras dengan pernyataan Oliver (1981) dimana kepuasan pelanggan merupakan respon emosional yang kompleks terhadap penggunaan

produk atau jasa. Sedangkan menurut Kotler (2013), kepuasan pelanggan ditentukan oleh seberapa baik ekspektasi pelanggan terhadap produk atau jasa terpenuhi, yang dapat diekspresikan dengan respon positif ataupun negatif. Demi menjaga relasi dengan konsumen, salah satu cara yang terbilang efektif adalah dengan meningkatkan kualitas produk dan/atau pelayanan. Pengukuran kepuasan pelanggan dapat dilakukan dengan membandingkan ulasan pelanggan terhadap produk terdahulu dengan produk yang telah ditingkatkan kualitasnya, serta membandingkan produk sendiri dengan produk pesaing sehingga dapat dilihat perbedaan kualitasnya.

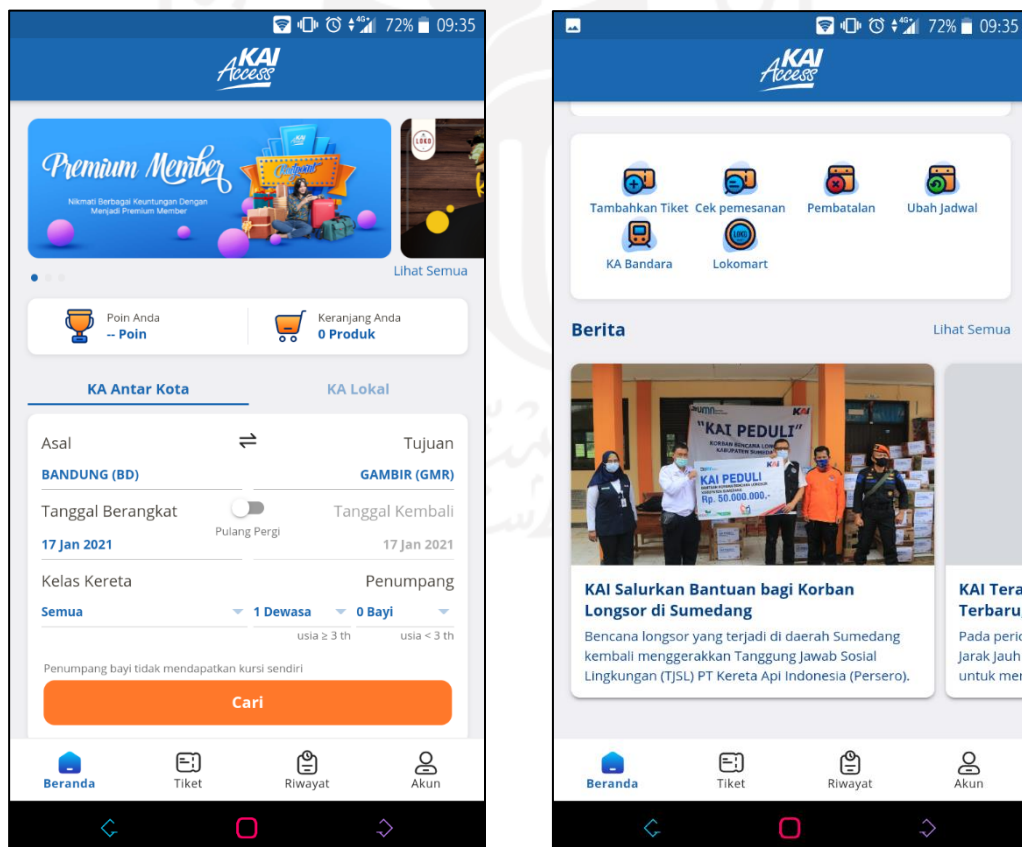
### 2.1.3 KAI Access

PT. Kereta Api Indonesia (Persero) atau umumnya disingkat PT. KAI merupakan perusahaan milik negara yang bergerak di bidang jasa transportasi kereta api penumpang dan barang. Diresmikan pada tahun 1998 yang mengacu pada Peraturan Pemerintah (PP) Republik Indonesia No. 19 Tahun 1998, PT. KAI menggalang visi “Menjadi solusi ekosistem transportasi terbaik untuk Indonesia” dengan tiga misi utama sebagai perwujudan visi perusahaan (PT. Kereta Api Indonesia (Persero), 2020), yakni:

1. Menyediakan sistem transportasi yang aman, efisien, berbasis digital, dan berkembang pesat untuk memenuhi kebutuhan pelanggan.
2. Mengembangkan solusi transportasi massal yang terintegrasi melalui investasi dalam sumber daya manusia, infrastruktur, dan teknologi.
3. Untuk memajukan pembangunan nasional melalui kemitraan dengan para pemangku kepentingan, termasuk memprakarsai dan melaksanakan pengembangan infrastruktur-infrastruktur penting terkait transportasi.

Dalam rangka meningkatkan pelayanan serta menawarkan kemudahan bagi para pengguna kereta api seperti yang tertuang dalam misi perusahaan, maka pada tahun 2014 PT. KAI meluncurkan aplikasi bernama KAI Access yang dapat diunduh oleh pengguna *smartphone* Android (melalui Google Play Store) maupun iOS (melalui Apple Store). KAI Access menawarkan beragam fitur utama selain pembelian tiket, yakni:

1. Pembatalan tiket; pengguna kereta api tidak perlu mendatangi hingga mengantri di loket stasiun.
2. Perubahan jadwal; selama kursi kereta masih tersedia, pengguna KAI Access cukup mengakses menu “Ubah Jadwal”.
3. Pembelian tiket kereta api lokal; pengguna dapat memesan tiket KA lokal 3 (tiga) jam sebelum keberangkatan kereta.
4. Reservasi tiket 1 (satu) jam sebelum keberangkatan; fitur ini bersifat eksklusif untuk pengguna KAI Access, yang mampu melayani pengguna dengan kebutuhan mendesak atau mendadak.
5. *E-boarding pass*; fitur ini bersifat eksklusif untuk pengguna KAI Access, yang menawarkan kemudahan saat proses *check-in*. Pengguna tidak perlu lagi mencetak tiket di stasiun, cukup menunjukkan *e-boarding pass* yang akan diperoleh 2 (dua) jam sebelum keberangkatan.

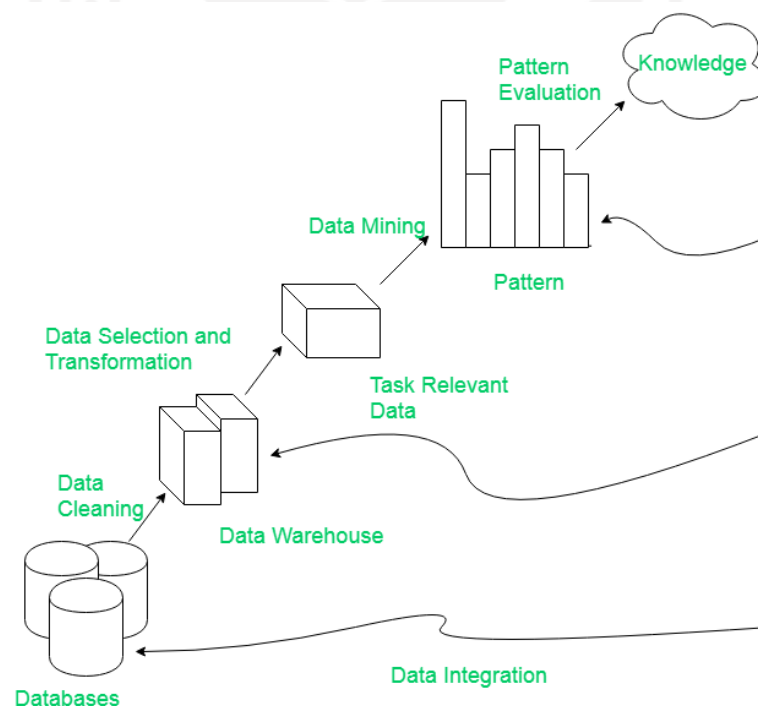


Gambar 2.1 Tampilan beranda KAI Access



### 2.1.4 Knowledge Discovery in Database (KDD)

Dengan semakin banyaknya aliran data setiap hari, maka akan semakin banyak informasi, baik bersifat tersirat maupun tersurat, yang dapat diperoleh. Adapun istilah untuk kegiatan pemerolehan suatu informasi dari kumpulan data adalah *knowledge discovery in database* (KDD) atau juga dikenal dengan *data mining*, yang mana kegiatan ini umumnya berperan penting dalam pengambilan suatu keputusan ataupun kebijakan. KDD memerlukan suatu sistem/algorithm yang mampu menggali informasi secara cepat, serta mampu menghasilkan *output* yang dapat dipahami oleh suatu organisasi. Saat ini, KDD dapat dilakukan dengan bantuan *machine learning*, *pattern recognition*, ilmu statistik hingga *artificial intelligence* (AI), dengan informasi yang diperoleh dari kegiatan KDD bersifat baru (*novel*) dan esensial sehingga sangat baik dijadikan acuan dalam *decision making* (Fayyad et al., 1996). Menurut Han et al. (2011), garis besar kegiatan KDD adalah sebagai berikut:



Gambar 2.2 Proses *Knowledge Discovery in Database*

Sumber: Han et al. (2011)

1. *Data Integration*; pengumpulan data dari berbagai sumber ke dalam satu *database*.
2. *Data Selection*; pemilihan atau pengambilan data yang relevan dengan analisis penelitian. Beberapa metode dapat digunakan dalam tahapan ini, di antaranya *decision trees*, *naïve bayes*, dan *clustering*.
3. *Data Cleaning*; pembersihan data yang dianggap *noisy* dan tidak relevan. Contoh data yang perlu dibersihkan adalah data duplikat, *missing values*, dan *stop words*. Selain itu, perbaikan kesalahan data juga merupakan salah satu kegiatan *data cleaning*.
4. *Data Transformation*; penyusunan data ke dalam bentuk yang sesuai dengan persyaratan *input* bagi *machine learning*.
5. *Data Mining*; analisis data untuk memperoleh informasi yang diharapkan. Metode *data mining* bergantung pada tujuan serta proses KDD.
6. *Pattern Evaluation*; identifikasi pola yang terbentuk sebagai representasi dari informasi.
7. *Knowledge Representation*; visualisasi hasil dari *data mining*. Dapat berupa tabel, *report*, atau *rules* (seperti *classification rules*, *association rules*).

### 2.1.5 Text Data Mining

*Data mining* merupakan kegiatan memperoleh informasi esensial yang bersumber dari kumpulan data, dengan melihat pola dan *trend* hasil analisis dari algoritma *machine learning*. Seiring berjalannya waktu, akan terdapat banyak data dari berbagai elemen yang disimpan maupun terpublikasikan melalui *platform* berbasis *internet*, maka akan terdapat banyak informasi yang terkandung di dalamnya. Namun tidak semua informasi tersebut bermanfaat atau dibutuhkan oleh setiap individu maupun perusahaan. Oleh karena itu, dengan tiga disiplin utama dalam *data mining* (statistika, *artificial intelligence*, dan *machine learning*), kumpulan data yang tersebar di seluruh dunia dapat dikerucutkan dan disaring sesuai dengan kebutuhan, serta ditata dan dianalisis demi memperoleh informasi spesifik dalam kurun waktu yang relatif cepat (SAS, 2020).

Salah satu kegiatan dari ruang lingkup *data mining* ialah *text mining*, yakni kegiatan memperoleh suatu informasi yang bersumber dari kumpulan kalimat atau teks yang tidak terstruktur, seperti konten di *website*, percakapan dan opini masyarakat. Dikatakan tidak terstruktur karena program komputer yang ada belum mampu memahami *natural language* (bahasa manusia) secara utuh (Nisbet, 2009). Menurut Kumar dan Karthika (2014), secara umum tahap-tahap dalam *text mining* terdiri dari:

1. *Text Preprocessing*, yakni proses restrukturisasi data menjadi lebih bersih dan terstruktur. Tahap ini penting dilakukan karena data mentah yang diperoleh dari *crawling* ataupun *scraping* masih mengandung *noise* seperti *HTML link*, *code script*, hingga kumpulan kata dan karakter yang tidak bermakna. Dalam *text preprocessing*, umumnya dilakukan beberapa sub-proses seperti:
  - a. *Tokenization*; penguraian kalimat menjadi kumpulan kata, serta eliminasi spasi, tanda baca dan karakter spesial lainnya. Contohnya yakni kalimat “Para pemburu mengumpulkan kayu dan daging” diurai menjadi “Para”, “pemburu”, “mengumpulkan”, “kayu”, “dan”, “daging”.
  - b. *Case folding*; perubahan karakter besar (*uppercase*) menjadi karakter kecil (*lowercase*). Karena perbedaan ukuran karakter dalam bahasa pemrograman memiliki makna yang berbeda, maka *case folding* diperlukan agar seluruh karakter menjadi ‘homogen’.
  - c. *Stop word removal*; eliminasi kata yang tidak bermakna atau tidak mengandung informasi penting, seperti kata “saya”, “ke”, “atau”. Tahap ini juga berlaku untuk bahasa lainnya.
  - d. *Stemming*; perubahan kata ke dalam bentuk dasarnya. Contohnya yaitu “perubahan” menjadi “ubah”, serta “menyulitkan” menjadi “sulit”.
2. *Text Transformation*, yakni konversi teks dalam dokumen menjadi *bag of words* yang bertujuan untuk meningkatkan efektivitas dalam analisis pemerolehan informasi.
3. *Feature Selection*, merupakan pemilihan beberapa variabel yang relevan sehingga dapat mengurangi *dataset* yang berimbas pada efisiensi *data mining*.

4. *Text Mining*, yakni kegiatan pencarian informasi dari kumpulan kata. Terdapat beberapa metode *mining* seperti *clustering*, *classification*, *information retrieval*, dan penelusuran topik.

Selain tahapan di atas, untuk data berupa opini atau ulasan masyarakat dapat dilakukan *sentiment analysis* yakni proses identifikasi karakteristik penilaian suatu opini atau ulasan masyarakat (positif, netral, atau negatif) terhadap penggunaan produk atau jasa, dengan bantuan *natural language processing* (NLP). Dengan demikian, informasi yang diperoleh dapat dikelompokkan ataupun disaring berdasarkan kecenderungan nilai tersebut (Gupta, 2018).

Dalam penelitian ini, data-data tidak terstruktur berupa *review* dari pengguna aplikasi diproses menjadi informasi yang diperlukan dalam perbaikan dan pengembangan sistem informasi maupun fitur terbaru. Beberapa tahapan *text mining* terdiri dari:

1. *Web scraping*; merupakan proses ekstraksi data dari *website*. Tahap ini bertujuan untuk meningkatkan efisiensi pengambilan data, yang dapat dilakukan menggunakan Python atau aplikasi *web scraping* yang tersedia, seperti AppFollow, WebHarvy atau Web Scraper (Nayoan, 2020).
2. *Text preprocessing*; adalah proses transformasi data yang tidak terstruktur menjadi data terstruktur, yang kemudian dapat dijadikan *input* bagi sistem *machine learning*. Beberapa tahapan dalam *text preprocessing* yakni *case folding* (mengubah huruf besar menjadi huruf kecil), *sentence tokenizing* (mengurai kalimat menjadi kata-kata), *stemming* (mengubah kata ke bentuk dasarnya), dan *stop word removal* (eliminasi kata yang dianggap tidak penting) (Nugroho, 2019).
3. *Tag clouding*; merupakan proses representasi beberapa kata yang sering muncul dalam kumpulan kalimat. Umumnya kata-kata tersebut akan ditampilkan dengan *font* yang lebih besar dibandingkan dengan kata lainnya. Dengan demikian, diperoleh suatu pola yang dapat menjadi informasi penting bagi suatu penelitian (Halvey, 2007).

### 2.1.6 Machine Learning

Dalam penerapan *machine learning*, perlu adanya *machine training* yang bertujuan untuk memberikan pemahaman kepada *machine* terhadap data yang akan diolah. Menurut Sethi (2020), terdapat dua jenis *training* yang umum digunakan oleh *data scientist*, dengan kriteria penggunaan yang berbeda.

1. *Supervised learning* merupakan sistem pembelajaran bagi *machine* dengan cara memberikan data yang telah diberi label. Data tersebut dikategorikan sebagai *train data*. *Machine* akan belajar dari data berlabel tersebut dan kemudian *machine* dapat memprediksi dataset baru berdasarkan pemahaman dari *train data*. Sistem ini biasanya digunakan untuk mengkategorikan teks, meramal cuaca hingga *spam detection*.
2. *Unsupervised learning* merupakan sistem pembelajaran dengan menggunakan data yang belum atau tidak dilabeli, sehingga *machine* dirancang untuk mempelajari sendiri data tersebut dan kemudian mengklasifikasi/melabelinya. Sistem ini digunakan untuk kegiatan *fraud detection*, *malware detection*, *language detection* dan *market basket analysis* (MBA).

Adapun perbandingan parameter untuk kedua jenis *training* ditampilkan pada Gambar 2.2 berikut ini.

Parameters	Supervised machine learning	Unsupervised machine learning
Input Data	Algorithms are trained using labeled data.	Algorithms are used against data which is not labelled
Computational Complexity	Simpler method	computationally complex
Accuracy	Highly accurate	Less accurate

Gambar 2.3 Perbandingan parameter *supervised* dan *unsupervised learning*

Sumber: Bansal (2020)

Selain kedua jenis *training* tersebut, menurut Hajaj (2020) terdapat dua teknik *training* lainnya, yakni:

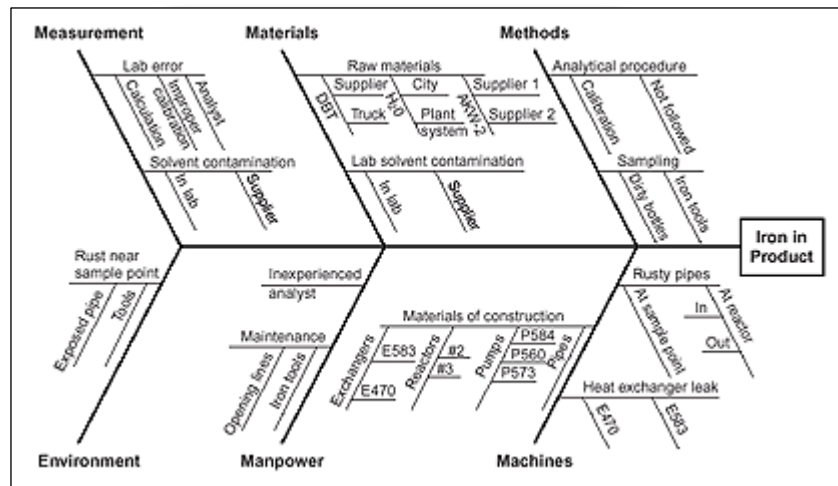
1. *Semi-supervised learning*, merupakan sistem pembelajaran bagi *machine* dengan cara memberikan label pada sebagian data saja, sedangkan sebagian lainnya dibiarkan tanpa label. *Machine* akan belajar dari data berlabel tersebut dan kemudian *machine* memprediksi label untuk data tanpa label tersebut. Teknik ini sering digunakan bila *dataset* yang dimiliki tergolong besar sehingga diperlukan waktu dan biaya untuk melabelinya. Salah satu penerapan *semi-supervised learning* adalah pengklasifikasian dokumen teks.
2. *Reinforced learning*, merupakan sistem pembelajaran bagi *machine* yang berkaitan dengan Tindakan atau keputusan yang harus diambil saat dihadapkan dengan “lingkungan” tertentu. Dalam *reinforced learning*, dikenal dua istilah pembelajaran yakni “*positive experience*” yang akan didapat oleh *machine* setelah memperoleh “hadiah” atas keputusannya, serta “*negative experience*” yang akan didapat oleh *machine* setelah memperoleh “hukuman” atas keputusannya. Metode pembelajaran ini membantu mempelajari cara mencapai tujuan yang kompleks atau memaksimalkan dimensi tertentu melalui *trial-and-error*. Beberapa penerapan *reinforced learning* yakni teknik sampling Thompson, *Q-Learning*, dan *Monte Carlo tree search*.

### 2.1.7 Quality Control (QC)

Dalam manajemen kualitas terdapat dua aspek yang saling berkaitan namun memiliki perbedaan, yakni *quality assurance* (QA) dan *quality control* (QC). Adapun QA merupakan kegiatan yang bertujuan untuk menjamin kualitas *output* yang ditargetkan perusahaan hingga sampai ke tangan konsumen, sedangkan QC adalah kegiatan realisasi target kualitas suatu produk demi memenuhi kebutuhan konsumen. Secara teknis, kegiatan QA bersifat preventif sedangkan QC lebih bersifat korektif (Russell, 2012). Dalam aktivitas QC, *tools* yang umumnya diterapkan untuk pengendalian kualitas terdiri dari tujuh macam yang lebih dikenal dengan istilah *seven basic quality tools* atau *seven tools*. Dikutip dari Tague (2005), penjelasan dari ketujuh *tools* tersebut adalah sebagai berikut.

### 1. Cause-and-effect Diagram

Dicetuskan oleh Dr. Kaoru Ishikawa, *tools* ini lebih dikenal dengan sebutan *fishbone diagram* yang berfungsi untuk mengidentifikasi inti dari suatu permasalahan. Kegiatan identifikasi ini biasanya melibatkan orang-orang yang ahli atau mengerti tentang masalah yang terjadi dengan sudut pandang tersendiri. Contoh dari *fishbone diagram* ditunjukkan pada Gambar 2.1 berikut ini.



Gambar 2.4 Fishbone Diagram

Sumber: Tague (2005)

Analisis akar permasalahan ditinjau dari aspek tertentu, yang tentunya berbeda untuk tiap jenis industri. Dikutip dari Kiran (2017), berikut merupakan aspek-aspeknya.

- a. 6Ms (untuk industri manufaktur), terdiri dari:
  - *Machine* (mesin dan teknologi)
  - *Method* (proses)
  - *Material*, termasuk bahan mentah, *consumables*, dan informasi
  - *Manpower* ataupun *mindpower*
  - *Measurement* (pengukuran/inspeksi)
  - *Millew/Mother nature* (lingkungan)
- b. 7Ps (untuk industri pemasaran), meliputi:
  - *Product/service*

- *Price*
- *Place*
- *Promotion*
- *People*
- *Positioning*
- *Packaging*

c. 4Ps (untuk industri jasa), yakni:

- *Policies*
- *Procedures*
- *People*
- *Process technology*

## 2. *Check Sheet*

*Tools* ini umumnya digunakan untuk mengumpulkan data berupa frekuensi kejadian, banyaknya cacat (*defect*) pada produk, dan sebagainya. Data pada *check sheet* dapat divisualisasikan menggunakan histogram dan diagram pareto.

Telephone Interruptions						
Reason	Day					Total
	Mon	Tues	Wed	Thurs	Fri	
Wrong number	+++			+++	+++	20
Info request						10
Boss	+++		+++			19
Total	12	6	10	8	13	49

Gambar 2.5 *Check Sheet*

Sumber: Tague (2005)



### 3. *Pareto Diagram*

*Tools* ini berupa grafik batang yang menunjukkan urutan permasalahan suatu proses, dari yang paling sering terjadi hingga yang sangat jarang terjadi, dengan urutan dari sebelah kiri hingga paling kanan. Adapun persentase kejadiannya diakumulasi serta divisualisasikan dalam bentuk grafik garis hingga mencapai 100 persen. Dalam diagram pareto terdapat prinsip *80/20 rule* yang menyatakan bahwa 80% efek dari suatu permasalahan disebabkan oleh 20% penyebabnya. Dengan demikian, *tools* ini berguna untuk menentukan fokus perbaikan sehingga meningkatkan efisiensi waktu dan biaya.

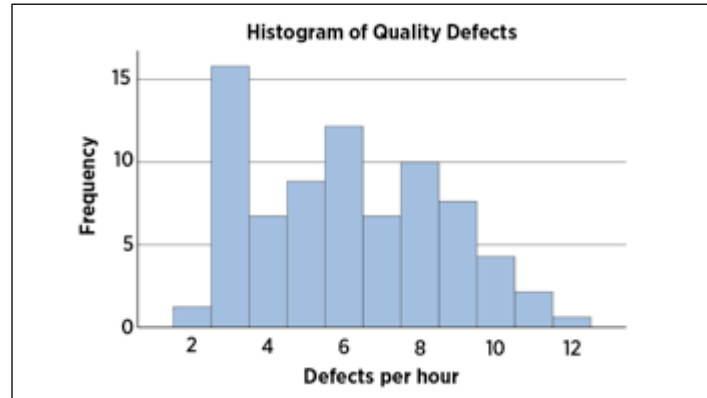


Gambar 2.6 Diagram Pareto

Sumber: Tague (2005)

### 4. *Histogram*

*Tools* ini menampilkan distribusi data dalam bentuk grafik, yang berguna untuk memberikan informasi tentang variasi data sehingga dapat menjadi bahan pengambilan keputusan.

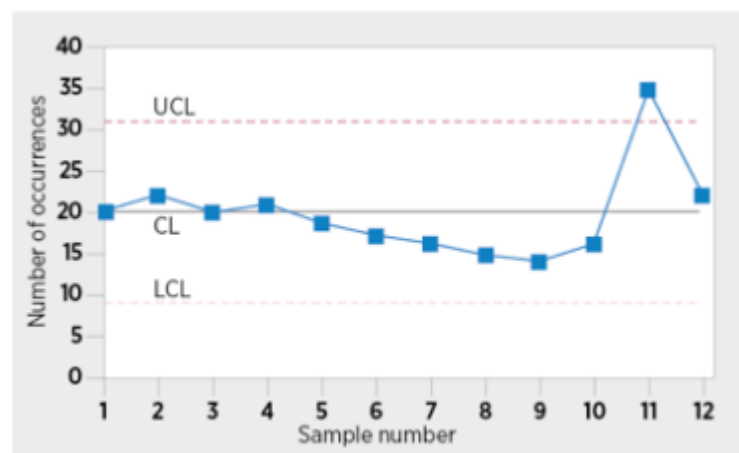


Gambar 2.7 Histogram

Sumber: Tague (2005)

### 5. Control Chart

*Tools* ini berbentuk grafik garis, yang digunakan untuk memantau stabilitas suatu proses serta mempelajari perubahan proses yang mungkin terjadi selama kurun waktu tertentu. *Control chart* terdiri dari tiga garis kontrol yang diperoleh dari *data record* terdahulu, yakni *upper control limit* (UCL), *lower control limit* (LCL), dan *center line* (CL) yang bermakna rata-rata. Dengan memasukkan data terkini serta membandingkannya terhadap ketiga garis kontrol tersebut, maka diperoleh informasi tentang konsistensi data sehingga dapat menjadi pertimbangan dalam melakukan penyesuaian suatu proses.

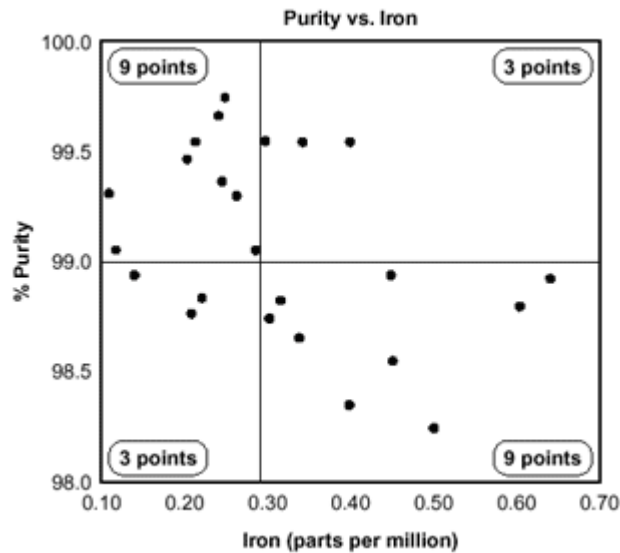


Gambar 2.8 Control Chart

Sumber: Tague (2005)

## 6. Scatter Diagram

*Tools* ini digunakan dalam pengujian tingkat serta jenis korelasi antara dua variabel. Bila kedua variabel berkorelasi dengan kuat, maka plot-plot data akan berdekatan hingga menyerupai atau membentuk garis lurus bila korelasinya sangat kuat.

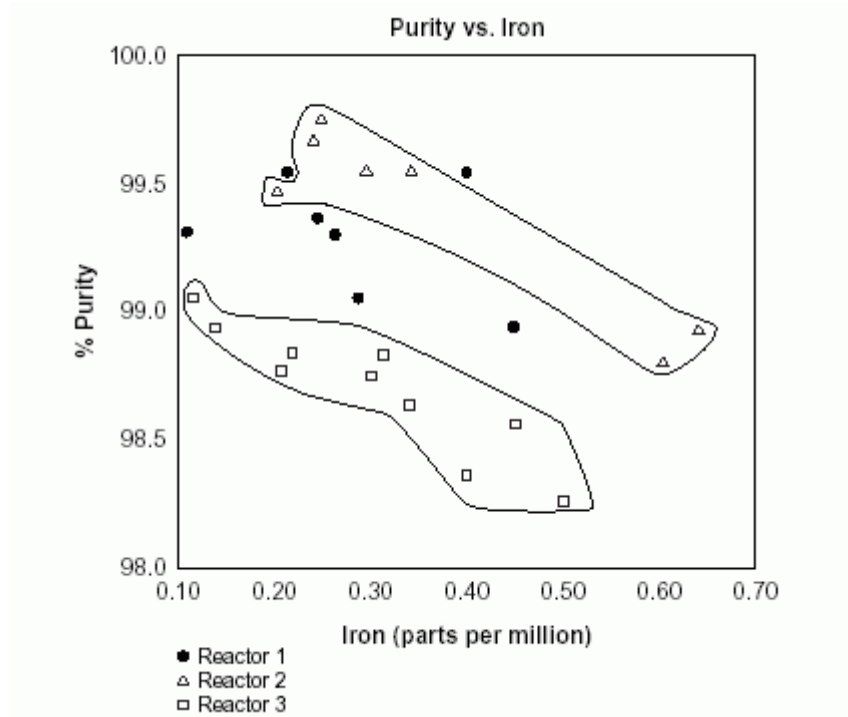


Gambar 2.9 Scatter Diagram

Sumber: Tague (2005)

## 7. Stratification

*Stratification* merupakan teknik pengelompokan data ke dalam beberapa kategori yang memiliki ciri-ciri tersendiri. Teknik ini berguna untuk mengidentifikasi apa saja penyebab suatu permasalahan. *Stratification* biasanya dikombinasikan dengan *tools* lainnya yang telah disebutkan sebelumnya.



Gambar 2.10 Pengelompokan Data (*Stratification*) pada *Scatter Diagram*

Sumber: Tague (2005)

### 2.1.7 Association Rules

*Association rules* dalam *data mining* merupakan kegiatan pencarian hubungan antar *data item* dalam kumpulan *dataset (database)*, atau dengan kata lain yakni pencarian atribut yang akan diperoleh secara bersamaan. Diperkenalkan oleh Rakesh Agrawal pada tahun 1994, *association rules* memiliki bentuk pernyataan dasar yakni “Jika muncul *item X*, maka muncul juga *item Y* sebagai konsekuensinya ( $X \rightarrow Y$ )”. Perhitungan yang digunakan dalam penentuan *association rules* menurut Zhao (2013) adalah:

1. *Support*; merupakan peluang munculnya *item X* dan *Y* dalam *database*. Perhitungan matematisnya ditunjukkan pada Persamaan 2.1 berikut ini.

$$\text{Support}(X \rightarrow Y) = P(X \cup Y) \quad \dots\dots (2.1)$$

2. *Confidence*; merupakan peluang munculnya *item Y* bila dalam *dataset* terdapat *item X*. Perhitungan matematisnya ditunjukkan pada Persamaan 2.2 berikut ini.

$$\text{Confidence}(X \rightarrow Y) = P(Y|X) = \frac{P(X \cup Y)}{P(X)} \quad \dots\dots (2.2)$$

3. *Lift ratio*; merupakan rasio kepercayaan terhadap persentase kasus yang mengandung *item* Y. Perhitungan matematisnya ditunjukkan pada Persamaan 2.3 berikut ini.

$$\text{Lift ratio}(X \rightarrow Y) = \frac{\text{Confidence}(X \rightarrow Y)}{P(Y)} = \frac{P(X \cup Y)}{P(X)P(Y)} \quad \dots (2.3)$$

Nilai *lift* menentukan seberapa kuat hubungan antar item, dalam arti bahwa item Y akan sering muncul bila diiringi dengan item X. Apabila nilai *lift* lebih besar dari 1 (satu) maka antar item memiliki hubungan yang kuat. Semakin besar nilai *lift* maka semakin kuat hubungannya (Hahsler, 2005).

## 2.2 Kajian Induktif

Kajian induktif dilakukan untuk mencari penelitian terdahulu yang memiliki ruang lingkup *data mining*. Kegiatan penelitian terkait *data mining* telah dilakukan oleh beberapa peneliti.

Li et al. (2020) melakukan analisis ulasan (*review*) pengguna suplemen *beta-hydroxybutyrate* (BHB) dari berbagai merk yang diperoleh dari situs Amazon.com. Penelitian ini dilakukan untuk membantu mengembangkan produk kesehatan yang sesuai dengan kebutuhan pelanggan. Metode yang digunakan dalam penelitian ini yaitu *text mining* menggunakan Web Scraper Chrome Extension untuk pengambilan data ulasan, *sentiment analysis* untuk mengetahui sifat penilaian pengguna dari ulasan yang diberikan, serta *complexity analysis* untuk mengetahui informasi statistik terkait jumlah karakter, kata, dan kalimat yang mengandung sentimen pada *review* yang diteliti. Hasil dari analisis *sentiment analysis* terdiri dari dua jenis, yakni berdasarkan *word-level* dan *sentence-level*. Untuk *word-level analysis* direpresentasikan menggunakan *word cloud* sehingga kata-kata bermakna

“menilai” dapat teridentifikasi dengan mudah. Sedangkan untuk *sentence-level analysis* menunjukkan bahwa ulasan yang mengandung penilaian positif, negatif dan netral untuk produk 1A (notabene menjadi salah satu fokus peneliti) berturut-turut sebesar 17,76%; 7,31%; dan 74,84%.

Abu-Shanab dan Harb (2019) melakukan penelitian terhadap perkembangan *e-government* dalam menjangkau masyarakat. Penelitian ini bertujuan untuk memahami pergerakan *e-government* dengan cara meneliti artikel serta publikasi yang berkaitan dengan topik- topik relevan di *e-government*. Metode *text mining* digunakan untuk memperoleh data, dengan pertimbangan kemampuan identifikasi kata kunci yang lebih banyak sehingga peneliti dapat memahami *trend* dalam *e-government*. *Text mining* dilakukan dengan algoritma *K-Means clustering* menggunakan *software* Python. Dari 2018 artikel yang diteliti, diperoleh data distribusi publikasi jurnal yang terbagi atas tiga interval waktu, yakni publikasi tahun 2002 – 2007, 2008 – 2012, dan 2013 – 2016. Kemudian topik yang sering dibahas selama ketiga rentang waktu tersebut di antaranya mengenai demokrasi, *open government*, penerapan teknologi, masalah pelayanan dan masalah pemerintahan.

Dewi et al. (2019) melakukan penelitian tentang *web scraping* media sosial untuk mengetahui kapasitas *application program interface* (API) yang disediakan oleh Facebook dan Twitter. *Regular expression* (Regex) digunakan dalam proses pencocokan teks dengan pola tertentu, serta algoritma yang diterapkan terdiri dari *web scraping* dan *information trend*. Hasil dari penelitian menunjukkan bahwa penerapan Facebook API dan Twitter API bagi keperluan *web scraping* dapat mempermudah penyimpanan data, mengurangi *redundancy information*, serta mempermudah pencarian dan penyimpanan informasi yang relevan.

Drury dan Roche (2019) melakukan survey terkait penerapan *text mining* untuk memperoleh informasi di bidang agrikultur yang bersumber dari hasil penelitian sebelumnya seperti *paper* dan artikel. *Paper* yang diteliti diperoleh melalui Google ScholarDBLP dan The ACL Anthology. Teknik dalam *text mining* yang digunakan

mencakup *information retrieval*, *information extraction* dan *sentiment analysis*. Hasil penelitian menunjukkan bahwa teknik pengumpulan informasi yang sering digunakan peneliti sebelumnya di antaranya yakni pengelompokan informasi (*information classification*), diskusi, *knowledge extraction*, serta *information monitoring*.

Gokhberg et al. (2019) melakukan penelitian terhadap perkembangan teknologi di sektor industri ekstraktif, seperti *oil and gas* dan pertambangan. Informasi yang akan digali bersumber dari artikel ilmiah, berita global, *proceeding*, hingga *website* menggunakan metode *text mining*. Tujuan dari penelitian ini yakni untuk memperkuat persaingan di dunia manufaktur peralatan industri ekstraktif, dengan cara “mendengar” perbincangan para peneliti dan pengusaha di bidang pertambangan dan perminyakan. *Text mining* yang dilakukan mengacu pada *N-gram*, *stem analysis*, *information retrieval*, dan pengelompokan data (*clustering*). Hasil penelitian menunjukkan bahwa topik yang sering dibicarakan di dunia industri ekstraktif saat ini di antaranya yaitu sistem untuk *deep-level mining*, *bioleaching*, *smart mines*, teknik untuk meningkatkan *oil recovery*, dan teknologi dengan energi terbarukan.

Moro et al. (2020) melakukan perbandingan informasi terkait teknologi baru di sektor energi terbarukan, antara sumber dari pendapat ahli (*expert review*) dengan hasil analisis kuantitatif menggunakan *text mining*. *Tools for Innovation Monitoring (TIM) software* digunakan untuk mencari dan mengumpulkan dokumen yang memiliki kata kunci “*solar cell*”, “*hydropower*”, dan “*wind power*”. Terdapat dua algoritma yang diterapkan dalam TIM serta dibandingkan satu sama lain, yakni perhitungan frekuensi kata kunci tiap dokumen/publikasi dan pencarian kata kunci yang relevan (*term frequency-inverse document frequency*; TF-IDF). Hasil dari penelitian menunjukkan bahwa informasi berdasarkan *output* TIM (dengan algoritma perhitungan frekuensi) yang berhasil diidentifikasi oleh para ahli sebesar 65% untuk topik *photovoltaics*, 56% untuk *tidal energy*, 31% untuk energi angin, dan 38% untuk *hydropower*. Sedangkan untuk hasil algoritma TF-IDF, kecocokan informasi dengan topik tersebut berturut-turut sebesar 59%, 67%, 38%, dan 25%.

Hasil tersebut diperoleh dengan menggunakan 300 kata kunci yang dianggap optimal.

Pejic-Bach et al. (2019) melakukan pencarian informasi tentang persyaratan pencari kerja dalam *job advertisement* yang tersebar di *website*. Penelitian ini bertujuan untuk melihat *trend* terkait posisi yang dicari serta kompetensi yang diperlukan oleh pelaku industri 4.0. Metode *text mining* melalui *software* WordStat Provalis dilakukan untuk menggali informasi *job advertisement* yang terdapat di *website* LinkedIn, dengan pencarian menggunakan kata kunci “*Industry 4.0*” dan “*Smart Factory*”. Dari 1460 *job advertisements* berbahasa Inggris yang diteliti, 48% di antaranya berasal dari Jerman dan Amerika Serikat. Lowongan kerja dengan kontrak *full-time* mendominasi di *website* LinkedIn dengan proporsi lebih dari 90%, serta lowongan kerja bagi pemula atau *fresh graduate* sebesar 32% dari seluruh lowongan kerja. Adapun kompetensi atau pemahaman yang banyak dicari di antaranya *supply chain*, manajemen proyek, *machine-learning*, *big data*, *computer science*, dan *software development*.

Villeneuve dan O’Brien (2019) melakukan penelitian terhadap ulasan (*review*) pengguna layanan *Airbnb* untuk mengetahui bagaimana kualitas lingkungan *indoor* (IEQ) yang sesuai dengan kepuasan pengguna. Teknik *text mining* dilakukan untuk memperoleh informasi tersebut, yang diterapkan pada 1,35 juta ulasan *Airbnb* di Kanada. *Software* yang digunakan untuk menerapkan *text mining* yakni MATLAB 2019 *Text Analysis Toolbox*, dengan tahapan *text mining* seperti *data preprocessing*, *trends identifying* dan *sentiment analysis*. Jenis komplain dikategorikan menjadi empat macam, yakni masalah kebisingan, *indoor air quality* (IAQ), suhu ruangan dan pandangan visual. Hasil penelitian menunjukkan bahwa terdapat komplain kebisingan sebanyak 35.026, complain IAQ sebanyak 5.739, komplain suhu ruangan sebanyak 14.883 dan komplain visual sebanyak 2.655. Selain itu, terdapat pula pelanggan *Airbnb* yang mengeluhkan lebih dari satu macam aspek.



Wu et al. (2019) melakukan evaluasi terhadap keakuratan *diagnostic code* untuk penyakit-penyakit psikis yang terdapat dalam *Taiwan's National Health Insurance Research Database* (NHIRD). Peneliti menggunakan metode *text mining* untuk mengambil profil gejala-gejala serta kerusakan fungsional yang diakibatkan oleh penyakit psikis, yang kemudian dijadikan bahan validasi *diagnostic code* tersebut. Model *text mining* yang digunakan yakni *dictionary-based mining* dan *linear-chain conditional random filed* (CRF). Hasil penelitian menunjukkan bahwa keakuratan dari *diagnostic code* untuk penyakit psikis berdasarkan data profil yang diteliti dapat diterima.

Yildiz (2019) melakukan penelitian terkait perkembangan studi pada industri 4.0 dari tahun 2012 hingga 2018. Peneliti mencari artikel dan jurnal yang tersedia di SCOPUS *database* dengan kata kunci "*Industry 4.0*". Dari 4.029 jurnal dan artikel yang berkaitan dengan industri 4.0, sebanyak dua publikasi dibuat pada tahun 2012, kemudian jumlah publikasi meningkat hingga 2.071 pada tahun 2018. Adapun negara dengan publikasi industri 4.0 terbanyak dipegang oleh Jerman, yakni sekitar 1100 publikasi. Kemudian diikuti oleh Italia, China, dan Amerika Serikat dengan jumlah publikasi hingga 800 dokumen.

Tabel 2.1 menunjukkan fokus penelitian terdahulu, yang sebagian besar menggunakan metode *text mining*.

Tabel 2.1 Perbandingan fokus penelitian

No.	Penulis (tahun)	Judul	Metode	Obyek Penelitian
1	Li et al. (2020)	<i>Text mining datasets of <math>\beta</math>-hydroxybutyrate (BHB) supplement products' consumer online reviews</i>	<i>Text mining, Complex analysis</i>	<i>Review pengguna produk</i>
2	Abu-Shanab & Harb (2019)	<i>E-Government Research Insights: Text Mining Analysis</i>	<i>Text mining, Keyword analysis</i>	Artikel dengan topik <i>e-government</i>

No.	Penulis (tahun)	Judul	Metode	Obyek Penelitian
3	Lusiana Citra Dewi et al. (2019)	<i>Social Media Web Scraping using Social Media Developers API and Regex</i>	<i>Web scraping</i>	Konten media sosial
4	Drury & Roche (2019)	<i>A survey of the applications of text mining for agriculture</i>	<i>Text mining</i>	Publikasi dengan topik agrikultur
5	Gokhberg et al. (2019)	<i>Advanced text-mining for trend analysis of Russia's Extractive Industry</i>	<i>Text mining</i>	Artikel, <i>newsfeed</i> , dan <i>proceeding</i>
6	Moro et al. (2020)	<i>Emerging technologies in the renewable energy sector: A comparison of expert review with a text mining software</i>	<i>Text mining</i>	Publikasi dengan topik energi terbarukan
7	Pejic-Bach et al. (2019)	<i>Text mining of industry 4.0 job advertisement</i>	<i>Text mining</i>	<i>Job advertisement</i> pada situs LinkedIn
8	Villeneuve & O'Brien (2019)	<i>Listen to the guests: Text-mining Airbnb reviews to explore indoor environmental quality</i>	<i>Text mining</i>	<i>Review</i> pelanggan <i>Airbnb</i> di Kanada
9	Wu et al. (2019)	<i>Using text mining to extract depressive symptoms and to validate the diagnosis of major depressive disorder from electronic health records</i>	<i>Text mining</i>	Dokumen medis terkait penyakit psikis
10	Yildiz (2019)	<i>Examining The Concept of Industry 4.0 Studies Using Text Mining and Scientific Mapping Method</i>	<i>Text mining, Scientific Mapping</i>	Artikel pada SCOPUS <i>database</i>

Beberapa penelitian sudah mulai mengembangkan penelitian yang menggunakan metode *text mining* dengan obyek penelitian seperti *job advertisement*, dokumen medis hingga ulasan pengguna produk. Sedangkan peneliti yang meneliti ulasan pengguna suatu aplikasi *mobile* masih terbilang sedikit, sehingga penulis berkesempatan untuk memulai penelitian terkait topik tersebut.



## BAB III

### METODE PENELITIAN

#### 3.1 Jenis Data

Dalam penelitian ini, data yang digunakan adalah data sekunder, yaitu data yang tidak diperoleh dari pengukuran langsung, melainkan diperoleh dari penelitian terdahulu sebagai landasan, kajian pustaka sebagai pendukung kredibilitas penelitian, serta dokumentasi ulasan yang bersumber dari Google Play Store sebagai sumber data.

#### 3.2 Prosedur Penelitian

##### 3.2.1 Tahap Persiapan

Pada tahap ini, dilakukan persiapan sebelum proses penelitian berlangsung, yakni menyiapkan alat dan bahan pengambilan data. Adapun *software* yang digunakan untuk menghimpun ulasan (*review*) di Google Play Store adalah AppFollow. Meskipun diperlukan *browser* dan koneksi internet untuk dapat mengakses AppFollow, pemilihan *software* tersebut mempertimbangkan kemudahan dalam proses *scraping*, penerapan fitur *filter* yang bervariasi, serta *output*-nya dapat langsung disimpan dalam format Microsoft Excel File (.xlsx), *comma-separated values* (.csv) atau sebagai *input* bagi *application program interface* (API). Selanjutnya adalah menyiapkan *software* untuk proses *text mining* dan *word clouding*, yakni Jupyter Notebook dengan bahasa pemrograman Python 3. Alasan pemilihan Jupyter Notebook di antaranya fleksibilitas yang baik saat menulis, mengubah dan menjalankan *coding*, kemudahan dalam memanggil (*import*) *library modules* yang berkaitan dengan proses *cleaning* dan

*mining* seperti *pandas*, *matplotlib*, dan *wordcloud*, serta kompatibilitas terhadap beberapa bahasa pemrograman yang mumpuni. Sedangkan bahasa pemrograman Python dipilih karena sifatnya *general-purpose*, serta *syntax* yang sederhana sehingga mudah dipelajari oleh berbagai kalangan. Contoh *interface* dari *software* AppFollow dan Jupyter Notebook ditampilkan pada Gambar A.1 dan A.2 di halaman lampiran.

### 3.2.2 Tahap Pengumpulan Data

Sumber data berupa ulasan pengguna aplikasi KAI Access terdapat di *website* Google Play Store. Untuk dapat memperoleh dan menyimpan kumpulan ulasan tersebut, langkah awal yang dilakukan adalah membuka aplikasi AppFollow melalui *browser*. Gambar A.3 menampilkan *home interface* dari AppFollow, dengan cara penambahan aplikasi yang ingin diteliti melalui AppFollow ditampilkan pada Gambar A.4 di halaman lampiran. Kemudian akan muncul *pop-up* seperti beserta opsi “Manage Review” untuk menampilkan ulasan pengguna KAI Access yang ditampilkan pada Gambar A.5.

Bila opsi “Manage Reviews” dipilih, maka *interface* dari *Rating & Reviews* akan ditampilkan seperti pada Gambar A.6. Melalui halaman tersebut, seluruh ulasan akan ditampilkan beserta rangkuman statistik terkait jumlah *rating* dan ulasan yang terhitung sejak aplikasi tersebut didaftarkan ke Google Play Store. Selain itu, berbagai fitur *filter* dapat diterapkan bila ingin menampilkan *rating* dan/atau ulasan dengan kriteria tertentu, misalnya ulasan dengan rentang tanggal 1 Januari 2018 hingga 31 Desember 2019 atau ulasan dengan *rating* 3 ke bawah. Dikarenakan dalam penelitian ini hanya menggunakan data *rating* dan ulasan, maka *filter* yang diterapkan yakni “Filter by Reviews” dengan cara menekan tombol “Reviews”.

Selanjutnya yakni mengambil dan menyimpan kumpulan ulasan dengan cara menekan tombol “Export” yang berada di kanan-atas halaman (dapat dilihat pada Gambar A.6). Kemudian akan terdapat *pop-up* yang menawarkan opsi penyimpanan *file* seperti pada Gambar A.7, dengan opsi yang dipilih adalah “All

Reviews (using current filters)” karena di tahap sebelumnya telah diterapkan “Filter by Reviews”. Untuk format *file* disimpan dalam bentuk *comma-separated values* (csv) dengan alasan mempermudah proses *text mining*, karena data *csv* hanya disimpan dalam bentuk *text file* saja (tidak ada *formatting*, *formula* dan sebagainya). *File* tersebut akan menjadi *input* bagi tahapan selanjutnya, yakni *text preprocessing*.

### 3.2.3 Tahap Pengolahan Data

Penelitian ini menjalankan dua proses utama, yakni *text mining* dan analisis permasalahan menggunakan *fishbone diagram*. Untuk tahap pertama yakni *text mining*, seluruh sub-prosesnya dijalankan melalui *software* Jupyter Notebook, sehingga langkah awal yang dilakukan yakni membuka lembar kerja baru dengan cara menekan opsi “New” dan memilih “Notebook Python 3”, seperti yang ditunjukkan pada Gambar A.8. Dalam tahap ini, rangkaian sub-proses yang dilakukan meliputi *data inputting*, *tokenization*, *stop word removal*, dan *word clouding*. Setiap sub-proses memerlukan modul (*module*) tersendiri karena setiap modul umumnya memiliki fungsi analisis yang berbeda.

*Data inputting* adalah proses memasukkan data ke dalam lembar kerja, dengan modul yang digunakan adalah *pandas*. Langkah ini bertujuan agar susunan data mengikuti ketentuan pemrograman. Contoh *data inputting* terhadap data ulasan (*review*) ditampilkan pada Gambar A.9, dengan format data berupa *data frame*. Setelah itu, dilakukan *tokenization* atau penguraian kalimat menjadi kumpulan kata. Proses ini bertujuan untuk mempermudah pembentukan pola atau *trend* kata-kata yang mengandung makna. Pada penelitian ini, *tokenization* dilakukan bersamaan dengan *case folding* atau pengubahan karakter besar (*uppercase*) menjadi karakter kecil (*lowercase*) supaya penulisan *coding* pada Jupyter menjadi lebih efisien. Gambar A.10 menampilkan cara *tokenization* dan *case folding* melalui Jupyter.

Langkah berikutnya yakni *stop word removal* dengan menggunakan modul Natural Language Toolkit (NLTK). Dalam NLTK terdapat *stop word* Bahasa

Indonesia yang terdiri dari kata dasar seperti “tapi” dan “harus”, serta kata berimbuhan seperti “tetapi” dan “seharusnya”. Setiap kata yang terdaftar dalam *stop word database* akan dihilangkan dari dokumen karena dianggap tidak memiliki makna khusus. Selain dari *stop word database* yang dimiliki NLTK, *stop word* yang tidak terdapat atau belum terdaftar dalam *database* ditambahkan terlebih dahulu, dengan sebagian dari *stop word* yang dimaksud berupa kata maupun singkatan kata yang informal atau tidak lazim dalam Kamus Besar Bahasa Indonesia (KBBI), seperti “gabisa”, “gk”, “klo” dan sebagainya. Dalam penelitian ini, penambahan *stop word* diterapkan melalui modul *wordcloud* dikarenakan proses penambahannya lebih mudah dibandingkan dengan modul NLTK. Adapun contoh *stop word removal* ditunjukkan pada Gambar A.11, dengan penerapan *stop word* tambahan dilakukan bersamaan dengan proses *word clouding*.

Proses selanjutnya yaitu eliminasi *emoji* yang terdapat dalam ulasan. *Emoji* atau ekspresi seseorang dalam wujud gambar atau *icon* tidak dapat dikenali oleh sistem *machine learning* sehingga dapat mengurangi efisiensi kinerja serta akurasi sistem yang dikarenakan *emoji* akan direpresentasikan sebagai *null values*. Adapun contoh penerapan proses ini pada Jupyter Notebook ditunjukkan pada Gambar A.12.

Setelah ulasan bersih dari *emoji*, langkah berikutnya yakni klasifikasi ulasan dengan cara pemberian label (*data labeling*). Tahapan ini bertujuan untuk menyaring ulasan dengan sentimen negatif yang dapat diketahui berdasarkan *rating* yang diberikan bersamaan dengan ulasan. Dengan demikian, proses klasifikasi berguna untuk mengurangi informasi yang *irrelevance* seperti ulasan dengan sentimen positif. Pelabelan dilakukan dengan menggunakan data *rating* pada ulasan sebagai acuan pelabelan, dengan ketentuan yakni bila ulasan memiliki *rating*  $< 3$  (kurang dari tiga) maka ulasan tersebut dianggap sebagai ulasan negatif. Jenis label yang digunakan yaitu *boolean*, dengan pemberian label “0” (nol) untuk ulasan positif dan label “1” (satu) untuk ulasan negatif. Dari pelabelan tiap ulasan, kemudian dilakukan proses pemilahan ulasan, dimana ulasan negatif akan digunakan untuk proses berikutnya yakni pemilahan ulasan berdasarkan bahasa.

Pemilahan berdasarkan bahasa dilakukan supaya informasi yang terkandung dalam ulasan berbahasa Inggris tidak tertutupi oleh informasi dari ulasan berbahasa Indonesia, yang direpresentasikan dalam *word cloud*.

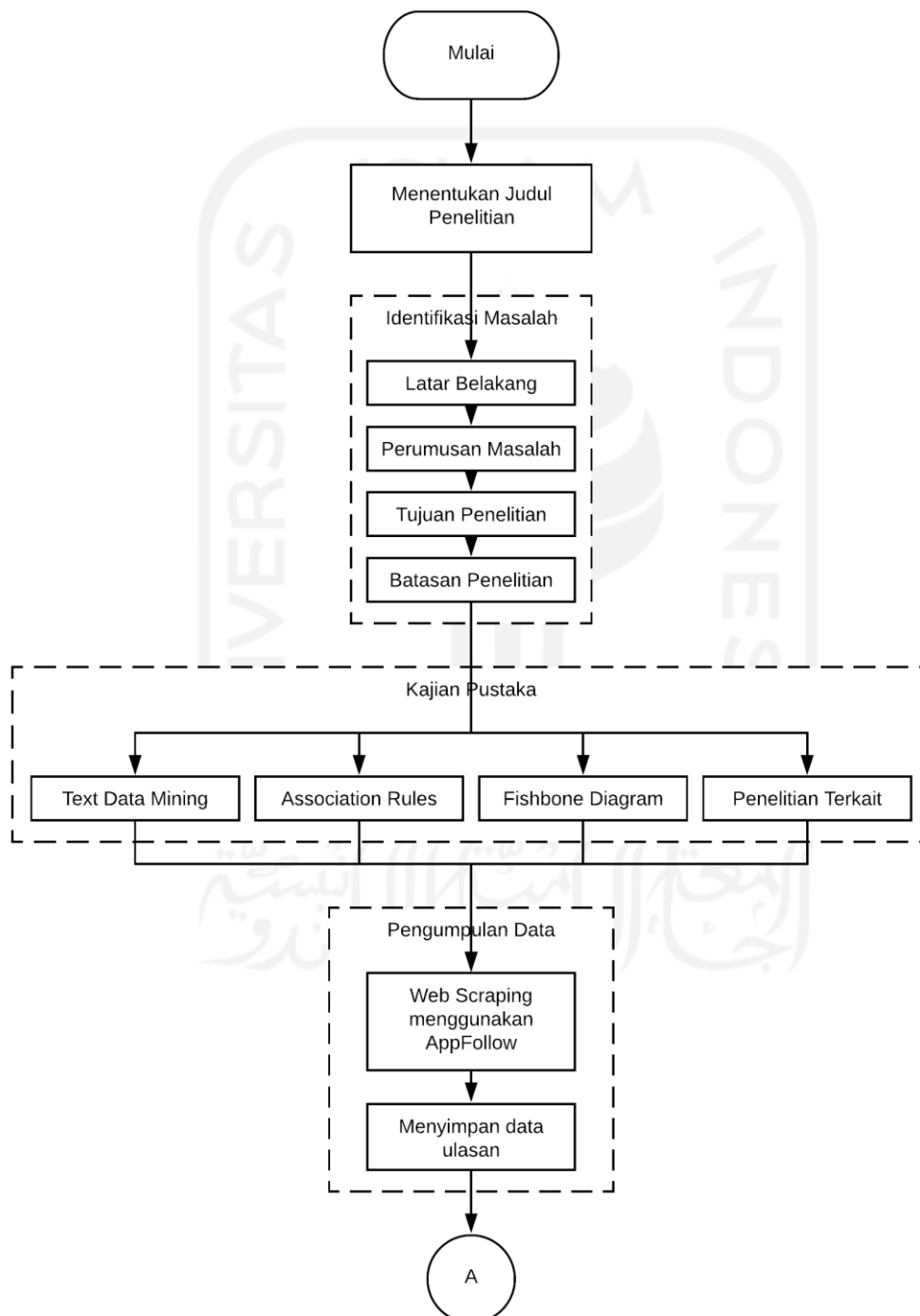
Setelah proses pemilahan selesai, maka tahapan berikutnya yakni *word clouding* atau representasi kata-kata yang sering muncul dalam kumpulan ulasan. Dalam penelitian ini, tahap *word clouding* akan menghasilkan *output* yang mengandung informasi tentang keluhan apa saja yang paling sering diutarakan oleh pengguna aplikasi. Proses ini dilakukan menggunakan algoritma *coding* yang sama untuk ulasan berbahasa Inggris maupun yang berbahasa Indonesia, namun dilakukan secara terpisah. Penerapan *word clouding* dilakukan dengan cara seperti pada Gambar A.13. Selanjutnya, hasil dari wordcloud disusun dalam *frame* menggunakan modul *matplotlib*. Contoh dari *output* proses *word clouding* ditampilkan pada Gambar A.14. Adapun kata yang dicetak besar merupakan kata yang sering muncul pada data yang dianalisis.

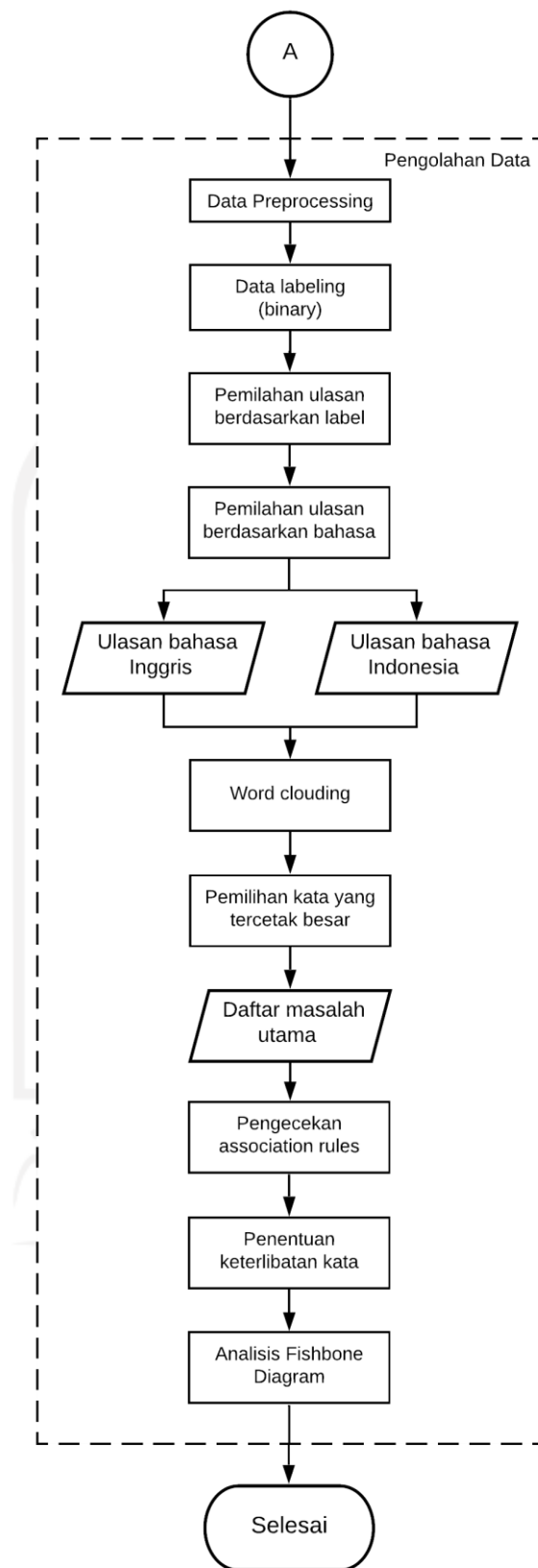
Dari *word cloud* kemudian akan diketahui beberapa kata kunci permasalahan yang tercetak dengan *font* besar. Kata-kata tersebut merupakan bagian dari *voice of customer* (VOC) yang kemudian akan diteliti akar permasalahannya dengan cara mengecek asosiasi kata. Dengan demikian, akan diketahui kata apa saja yang berkaitan dengan kata kunci permasalahan. Dalam konteks *data mining*, tahap asosiasi kata lebih dikenal dengan istilah *association rules*, dengan analisisnya dapat dilakukan menggunakan *module mlxtend* pada Python. Contoh *output* dari proses asosiasi kata ditampilkan pada Gambar A.15. Setelah dilakukan pengecekan *association rules*, maka kata-kata yang teridentifikasi berkaitan dengan kata kunci akan menjadi akar/penyebab permasalahan yang selanjutnya disusun dalam *fishbone diagram* (akan dijelaskan pada Bab V). Karena PT. KAI bergerak di bidang jasa, dengan KAI Access sebagai salah satu wujud pelayanannya, maka jenis *fishbone diagram* yang dipilih yakni 4Ps (*Policies, Procedures, People, dan Process technology*).



### 3.3 Diagram Alir

Adapun garis besar dari tahap-tahap penelitian dapat dilihat pada Gambar 3.1 berikut ini.





Gambar 3.1 Diagram alir penelitian

## BAB IV

### PENGUMPULAN DAN PENGOLAHAN DATA

#### 4.1 Pengumpulan Data

Adapun data yang digunakan untuk penelitian ini adalah ulasan pengguna aplikasi KAI Access yang bersumber dari Google Play Store, dengan bantuan aplikasi AppFollow untuk proses *scraping*. Gambar 4.1 dan 4.2 menunjukkan *screenshot* rekapan ulasan dengan total ulasan yang diperoleh sebanyak 11.768 ulasan. Adapun interval waktu perilisan ulasan yakni dimulai dari 16 Juli 2014 hingga 10 April 2020.

	A	B	C	D	E	F	G	H	I
1	Date	Author	Rating	Review					
2	4/10/2020 0:22	erma novriawati		1 saya tidak bisa add tiket di aplikasiku karena pesan tiket lewat agen. s					
3	4/9/2020 23:22	dwi aprilia		5 Good					
4	4/9/2020 22:39	Galih Mahardika		1 Tidak bisa melakukan pembatalan jika pemesanan tiket di marketplace					
5	4/9/2020 13:31	Titik Sendyningtiyas		1 Rubbish app. I can't even log in. How dare KAI recommend this kinda a					
6	4/9/2020 7:42	beauty psp		2 Aplikasinya bagus. Tapi pembayaran hanya bisa pakai LinkAja dan apli					
7	4/9/2020 0:58	Ysudahi Prasetyo		1 Dimana menu "my trip" untuk pembatalan tiket???					
8	4/8/2020 3:17	den alvan		1 Tidak mengerti sama app ini, sudah dapat konfirmasi akun aktif di ema					
9	4/7/2020 14:11	Ve Ve		1 KAI acces sangat menyestatkan, 😞😞😞 sudah ada login, auto updat					
10	4/7/2020 5:57	Sriyati Djamari		1 Cant log in, and called to KAI call centre to cancel the tickets during co					
11	4/7/2020 3:42	dwiyan panduwijaya		1 Aplikasi rusak					
12	4/6/2020 11:24	ozi ziqin		5 good respon					
13	4/6/2020 5:55	E Ooi		1 Managed to register but can only get errors when trying to login.. App					
14	4/6/2020 5:29	Heri Setyoadi		1 Ribet, mending via web					
15	4/5/2020 13:48	sylvia utami		1 tidak bisa login kenapa kah? semua orang butuh refund urgent kenapa					
16	4/5/2020 0:16	Nicko Redano		4 Getting better					
17	4/4/2020 11:49	Lucia Lukito		2 Tidak ada menu My Trips untuk pembatalan tiket. bagaimana nih?					
18	4/4/2020 5:18	Ayub Angga Direja		1 Andai Pak Jonan masih di KAI, bakal ngamuk beliau pas mencoba canc					
19	4/3/2020 12:53	koko woke		3 Sudah bisa , Mohon di Buat petunjuk pembatalan banyak user yang bir					
20	4/2/2020 18:07	Jorise Manise		3 Menu Pembatalan tiket yang beli dari toko online bagaimana mana?					
21	4/2/2020 15:14	Tunamayoo		1 Tidak mendapat email konfirmasi, sudah cek spam, dll tidak ada email					
22	4/2/2020 10:04	Ramen Sans		5 Mantap					

Gambar 4.1 Hasil *review scraping* dalam format csv (bagian 1)

	A	B	C	D	E	F	G	H	I
11750	7/16/2014 8:47	Bambang Suliastowo		5 Usefull app					
11751	7/16/2014 8:44	Rad. Vanjosh		5 Selamat ya PT KAI untuk aplikasi ini, semoga bermanfaat bagi seluruh					
11752	7/16/2014 8:37	Achmad Sinteljak		5 Siiiip					
11753	7/16/2014 8:37	arief setiawan		5 Official apps for Train Ticket Booking... Faster to connect compared to					
11754	7/16/2014 8:32	Fauzi Ikhwan		5 Akhirnya keluar juga aplikasi resmi dari KAI. Nice apps:)					
11755	7/16/2014 8:26	Bramantiyo Marjuki		5 great					
11756	7/16/2014 6:38	Muhammad Hendro Gunawan		5 easy, simply, usefull....grab it fast buddy!					
11757	7/16/2014 5:58	Andika Daniarsa		5 Gud job					
11758	7/16/2014 4:12	jjo jjo		5 mantap....					
11759	7/16/2014 4:00	Rizal Rahardian		5 Akhirnya ada aplikasi resmi KAI di android n ios kalau bisa d platform v					
11760	7/16/2014 3:54	Awan R. Setiawan		5 Mohon dibantu saya sangat senang sekali ada app ini, tetapi kq tidak b					
11761	7/16/2014 3:33	erlangga adriadi		5 Akhirnya keluar juga di android. Keep up you'r good work IT team.					
11762	7/16/2014 3:16	jevi tri		5 Mantap bravo.....Loved it					
11763	7/16/2014 2:54	Tatang Hamdani		5 Welcome KAI access, semoga menjadi bagian dari perubahan dan berm					
11764	7/16/2014 2:48	icegoldensnow boardYT		5 Semakin mudah membeli ticket kereta api.... Mantapp!!					
11765	7/16/2014 2:30	Awal Arifianto		5 Bagus sekali untuk pemesanan tiket kereta api					
11766	7/16/2014 2:23	Dias Azhar Pradana		5 Senang sekali akhirnya ada aplikasi resmi dari KAI untuk reservasi tiket					
11767	7/16/2014 2:21	Susi Susilawati		5 Aplikasi keren nih bisa pesen tiket dmn aja					
11768	7/16/2014 2:11	bagit airlangga		5 Aplikasi yang sangat bagus. Membantu untuk memesan tiket secara ce					
11769	7/16/2014 2:02	Yogi Wibisono		5 Good Job Sir					
11770									
11771									

Gambar 4.2 Hasil *review scraping* dalam format csv (bagian 2)

## 4.2 Pengolahan Data

Hasil ulasan yang diperoleh masih mengandung karakter dan kata-kata tidak bermakna, sehingga perlu dilakukan *text preprocessing* agar data lebih “bersih”. Melalui tahapan proses yang telah dijelaskan pada Bab III, maka hasil dari tiap proses ditampilkan berurutan di bawah ini.

### 4.2.1 Data Inputting

Data ulasan yang terkumpul dalam format csv dimasukkan ke dalam lembar kerja Jupyter menggunakan modul pandas seperti yang ditunjukkan pada Gambar 4.3 berikut ini.

**Read CSV data**

```
In [1]: import pandas as pd

In [2]: df = pd.read_csv('C:\COLLEGE\College\8th Semester\TA\KAIA_Review_Strc_Cmpl.csv',
                        usecols=['Rating', 'Review'])
```

Gambar 4.3 *Data inputting* menggunakan modul pandas

### 4.2.2 Tokenization dan Stop Word Removal

Proses *tokenization* dilakukan supaya pengecekan kata pada tahap *stop word removal* dapat dilakukan dengan akurat. Adapun penerapan algoritma *tokenization* beserta *stop word removal* pada Python ditunjukkan pada Gambar 4.4 berikut ini.

```

Text preprocessing

In [3]: import unicodedata
import nltk
from unidecode import unidecode
from nltk.corpus import stopwords

In [4]: stopword_id = set(stopwords.words('indonesian'))

Tokenize and remove stopwords

In [5]: sentence = ""
for val in df.Review:
    val = str(val)
    kata1 = val.split()
    for i in range (len(kata1)):
        kata1[i] = kata1[i].lower()
    sentence += ' '.join([word for word in kata1 if word not in stopword_id])+'|'

```

Gambar 4.4 *Tokenization* dan *stop word removal*

Pada tahap ini juga dilakukan *case folding* yang bertujuan untuk mengubah karakter huruf besar menjadi kecil agar proses *stop word removal* berjalan lancar. *Case folding* dilakukan karena seluruh *stop word database* yang terdapat dalam modul NLTK berukuran karakter kecil sehingga NLTK tidak dapat mengenali suatu karakter atau kata berukuran besar. Sebagai contoh yakni kata “saya” dan “Saya” akan dianggap berbeda oleh NLTK. Penerapan *case folding* pada Gambar 4.4 yakni melalui *syntax* `lower()`.

### 4.2.3 Emoji Removal

Tahap berikutnya adalah eliminasi *emoji* yang dilakukan seperti pada Gambar 4.5 berikut ini.

```

Remove emojis

In [6]: def deEmojify(inputString):
        return inputString.encode('ascii', 'ignore').decode('ascii')
        no_emoji = deEmojify(sentence)

```

Gambar 4.5 Proses eliminasi emoji

Dari proses di atas, maka akan diperoleh ulasan yang lebih bersih dari karakter tak bermakna. Setelah data dibersihkan, kemudian kumpulan kata hasil *tokenization* digabungkan kembali sesuai dengan susunan yang berdasarkan kalimat awal. Perintah “append” digunakan dalam menjalankan proses tersebut. Contoh penerapan “append” beserta perbandingan antara ulasan bersih dan ulasan awal (mentah) ditunjukkan pada Gambar 4.6 di bawah ini.

```

Append clean review to dataframe

In [7]: newSentence = []
        temp = no_emoji.split('|')
        for each in temp:
            newSentence.append([each])
        print(newSentence)
...

In [8]: clean_text = pd.DataFrame(newSentence, columns=['Clean_Review'])

In [9]: new_df = pd.concat([df,clean_text], axis=1)
        new_df

Out[9]:

```

	Rating	Review	Clean_Review
0	1.0	saya tidak bisa add tiket di aplikasiku karena...	add tiket aplikasiku pesan tiket agen. ini. to...
1	5.0		Good good
2	1.0	Tidak bisa melakukan pembatalan jika pemesanan...	pembatalan pemesanan tiket marketplace lain. ...
3	1.0	Rubbish app. I can't even log in. How dare KAI...	rubbish app. i can't even log in. how dare kai...
4	2.0	Aplikasinya bagus. Tapi pembayaran hanya bisa ...	aplikasinya bagus. pembayaran pakai linkaja ap...
...	...	...	...
11764	5.0	Senang sekali akhirnya ada aplikasi resmi dari...	senang aplikasi resmi kai reservasi tiket onli...
11765	5.0	Aplikasi keren nih bisa pesen tiket dmn aja	aplikasi keren nih pesen tiket dmn aja
11766	5.0	Aplikasi yang sangat bagus. Membantu untuk mem...	aplikasi bagus. membantu memesan tiket cepat.
11767	5.0		Good Job Sir good job sir
11768	NaN		NaN

```

11769 rows x 3 columns

```

Gambar 4.6 Perbandingan ulasan bersih dengan ulasan awal (*raw*)

#### 4.2.4 Data Labeling

Langkah selanjutnya yakni melabeli data berdasarkan *rating* masing-masing ulasan, dimana ulasan dengan *rating* lebih dari 3 (tiga) diberi label “0” dan ulasan dengan *rating* kurang dari 3 (tiga) diberi label “1”. Pelabelan ini bertujuan untuk memilah ulasan dengan *rating* lebih dari 3 (tiga) agar tidak diikutsertakan dalam proses *word cloud*. Pada tahap ini juga disertakan proses eliminasi *null values* yang kemungkinan terdapat dalam data ulasan. Gambar 4.7 menunjukkan cara pelabelan menggunakan Python.

**Review labeling based on rating**

```
In [10]: new_df["bad_review"] = new_df["Rating"].apply(lambda x: 1 if x < 3 else 0)
new_df["bad_review"].value_counts(normalize=False)
```

```
Out[10]: 1    6662
         0    5107
         Name: bad_review, dtype: int64
```

**Drop missing values**

```
In [11]: new_df.dropna(inplace=True)
new_df
```

```
Out[11]:
```

	Rating	Review	Clean_Review	bad_review
0	1.0	saya tidak bisa add tiket di aplikasiku karena...	add tiket aplikasiku pesan tiket agen. ini. to...	1
1	5.0	Good	good	0
2	1.0	Tidak bisa melakukan pembatalan jika pemesanan...	pembatalan pemesanan tiket marketplace lain. ...	1
3	1.0	Rubbish app. I can't even log in. How dare KAl...	rubbish app. i can't even log in. how dare kai...	1
4	2.0	Aplikasinya bagus. Tapi pembayaran hanya bisa ...	aplikasinya bagus. pembayaran pakai linkaja ap...	1
...	...	...	...	...
11763	5.0	Bagus sekali untuk pemesanan tiket kereta api	bagus pemesanan tiket kereta api	0
11764	5.0	Senang sekali akhirnya ada aplikasi resmi dari...	senang aplikasi resmi kai reservasi tiket onli...	0
11765	5.0	Aplikasi keren nih bisa pesen tiket dmn aja	aplikasi keren nih pesen tiket dmn aja	0
11766	5.0	Aplikasi yang sangat bagus. Membantu untuk mem...	aplikasi bagus. membantu memesan tiket cepat.	0
11767	5.0	Good Job Sir	good job sir	0

11768 rows x 4 columns

Gambar 4.7 Proses pelabelan ulasan berdasarkan *rating*

#### 4.2.5 Pemilahan Ulasan Berdasarkan Label

Langkah berikutnya yakni pemilahan ulasan berdasarkan label, dimana ulasan dengan label “0” tidak diikutsertakan dalam tahapan berikutnya. Proses ini ditampilkan pada Gambar 4.8 berikut ini, dimana akan diketahui terdapat 6.662 ulasan dengan *rating* di bawah tiga.

**Remove reviews labeled '0'**

```
In [13]: new_df1 = new_df[new_df['bad_review'].isin(['0']) == False]
new_df1
```

Out[13]:

	Rating	Review	Clean_Review	bad_review
0	1.0	saya tidak bisa add tiket di aplikasiku karena...	add tiket aplikasiku pesan tiket agen. ini. to...	1
2	1.0	Tidak bisa melakukan pembatalan jika pemesanan...	pembatalan pemesanan tiket marketplace lain. ...	1
3	1.0	Rubbish app. I can't even log in. How dare KAl...	rubbish app. i can't even log in. how dare kai...	1
4	2.0	Aplikasinya bagus. Tapi pembayaran hanya bisa ...	aplikasinya bagus. pembayaran pakai linkaja ap...	1
5	1.0	Dimana menu "my trip" untuk pembatalan tiket???	dimana menu "my trip" pembatalan tiket???	1
...	...	...	...	...
11698	1.0	sudah instal.mau sign up tapi g bisa. msalahan...	instal.mau sign up g bisa. msalahnyaa isi bio...	1
11703	1.0	Tidak bisa mengambil data ke server..	mengambil data server..	1
11721	1.0	Payah, pembayarannya sangat terbatas	payah, pembayarannya terbatas	1
11723	1.0	Apa ada masalah dengan sistem registrasinya. S...	sistem registrasinya. registrasi	1
11728	1.0	Masih banyak bug yang perlu diperbaiki, a.l: i...	bug diperbaiki, a.l: info ketersediaan "tersed...	1

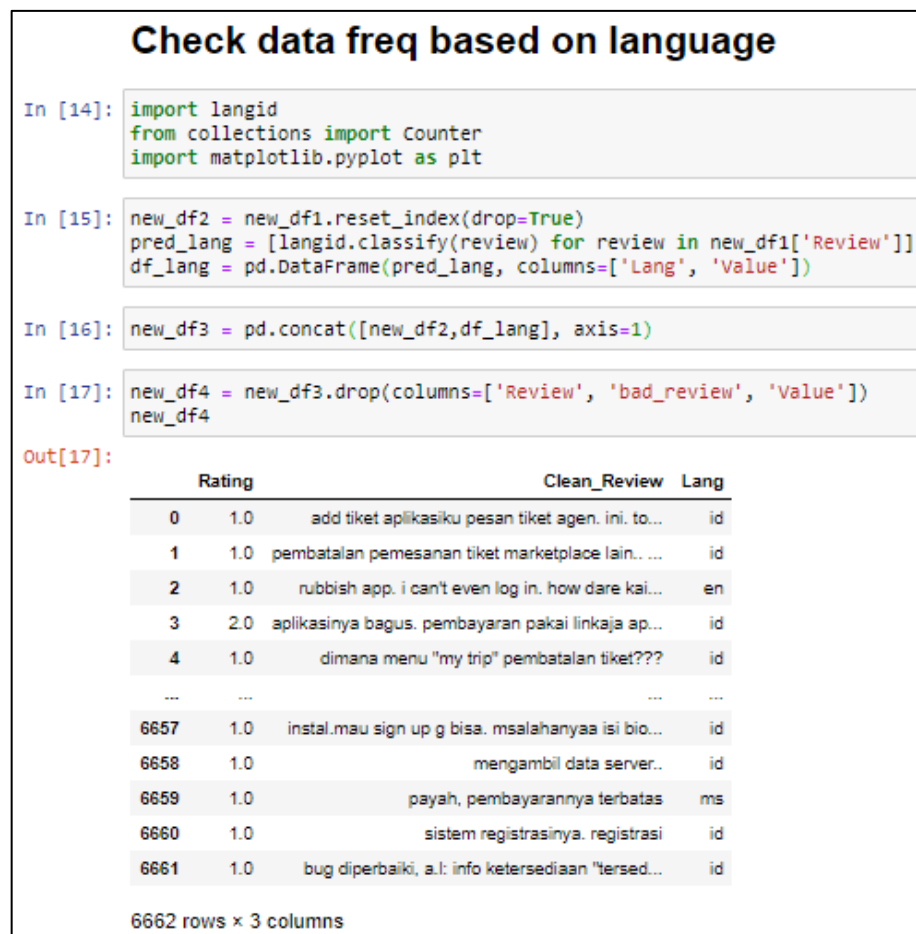
6662 rows x 4 columns

Gambar 4.8 Proses pemilahan ulasan berdasarkan label

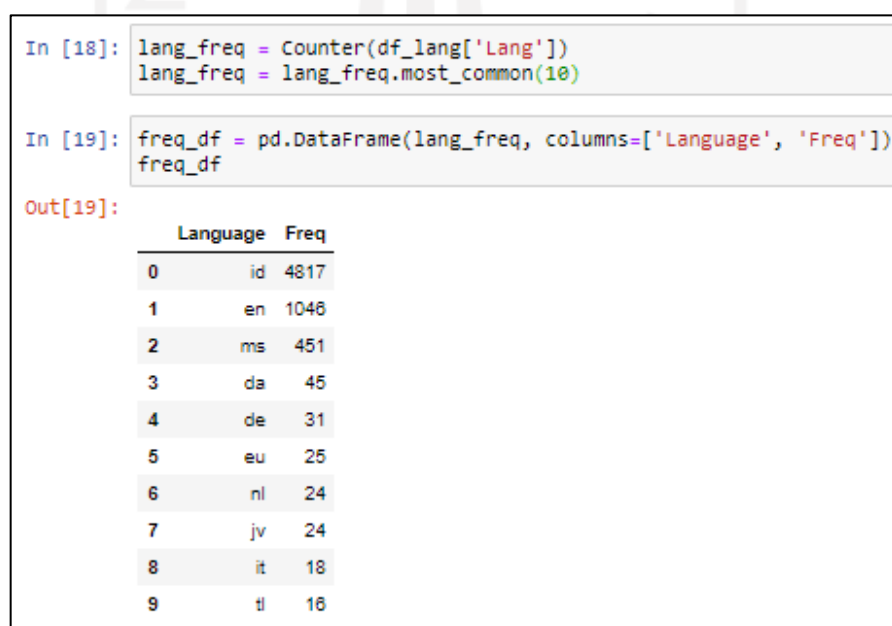
#### 4.2.6 Pemilahan Ulasan Berdasarkan Bahasa

Dikarenakan pada data ulasan terdapat ulasan berbahasa Indonesia dan Inggris, maka kedua data tersebut akan tetap diproses namun perlu dipilah terlebih dahulu. Sebelum dipilah, tiap data ulasan diprediksi serta diberi label agar mudah dalam pemilahannya. Dalam penelitian ini, *module* yang digunakan untuk memprediksi bahasa serta memberi label bahasa yakni *langid*. Proses ini ditampilkan pada Gambar 4.9, sedangkan untuk mengetahui frekuensi penggunaan bahasa maka dilakukan perhitungan menggunakan *module* *counter* seperti yang ditampilkan pada Gambar 4.10. Dari Gambar 4.10 dapat diketahui bahwa dari 6.662 ulasan, diprediksi jumlah ulasan berbahasa Indonesia (berlabel “id”) sebanyak 4.817 buah, untuk ulasan berbahasa Inggris (berlabel “en”) sebanyak 1.046 buah, serta terdapat ulasan berbahasa Jawa (berlabel “jv”) sebanyak 24 buah dan ulasan berbahasa Malay/Melayu (berlabel “ms”) sebanyak 451 buah. Selain keempat macam bahasa tersebut, terdapat ulasan yang salah diprediksi oleh *module* *langid* sebanyak 324 buah. Untuk menghindari kesalahan atau penurunan akurasi dalam proses berikutnya, maka ulasan dengan bahasa selain Inggris, Indonesia, Jawa, dan Melayu tidak diikutsertakan. Dengan demikian, data ulasan yang akan diproses lebih lanjut adalah ulasan berbahasa Inggris (EN) serta gabungan dari ulasan berbahasa Indonesia (ID), Melayu (MS), dan Jawa (JV).





Gambar 4.9 Proses prediksi dan pelabelan ulasan berdasarkan bahasa



Gambar 4.10 Proses pengecekan frekuensi penggunaan bahasa

Langkah berikutnya adalah pemilahan ulasan berdasarkan bahasa, yakni bahasa Inggris (EN) dan gabungan bahasa Indonesia, Melayu, dan Jawa (ID – MS – JV). Adapun proses pemilihannya masing-masing ditampilkan pada Gambar 4.11 dan 4.12 berikut ini.

**EN reviews**

```
In [20]: df_en = new_df4[new_df4['Lang'].isin(['en']) == True]
df_en
```

Out[20]:

	Rating	Clean_Review	Lang
2	1.0	rubbish app. i can't even log in. how dare kai...	en
7	1.0	cant log in, and called to kai call centre to ...	en
9	1.0	managed to register but can only get errors wh...	en
18	1.0	this apps is being quarantened because of covi...	en
22	1.0	i can't login more times	en
...	...	...	...
6497	2.0	what a pity. it's good to make a ticket reser...	en
6502	1.0	can't access server	en
6528	1.0	system error. jadwal maret. please fix it...	en
6584	1.0	couldnt proceed to payment or apps crash in th...	en
6611	1.0	always asking for update even though it's alre...	en

1046 rows × 3 columns

Gambar 4.11 Pemilahan ulasan berlabel EN

**ID, MS, and JV reviews**

```
In [21]: df_id = new_df4[new_df4['Lang'].isin(['id', 'ms', 'jv']) == True]
df_id
```

Out[21]:

	Rating	Clean_Review	Lang
0	1.0	add tiket aplikasiku pesan tiket agen. ini. to...	id
1	1.0	pembatalan pemesanan tiket marketplace lain... ..	id
3	2.0	aplikasinya bagus. pembayaran pakai linkaja ap...	id
4	1.0	dimana menu "my trip" pembatalan tiket???	id
5	1.0	mengerti app ini, konfirmasi akun aktif email,...	id
...	...	...	...
6657	1.0	instal.mau sign up g bisa. masalahnyaa isi bio...	id
6658	1.0	mengambil data server..	id
6659	1.0	payah, pembayarannya terbatas	ms
6660	1.0	sistem registrasinya. registrasi	id
6661	1.0	bug diperbaiki, a.l: info ketersediaan "tersed...	id

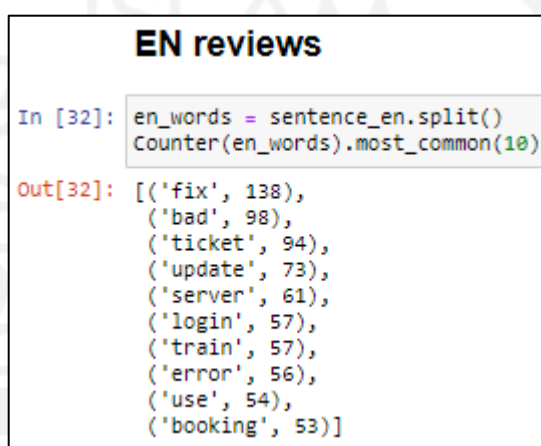
5292 rows × 3 columns

Gambar 4.12 Pemilahan ulasan berlabel ID – MS – JV





yang tercetak besar, yakni “diperbaiki”, “error”, “login”, dan “update”. Karena kata “login”, “update”, dan “error” muncul di kedua *word cloud*, serta kata “fix” memiliki makna yang sama dengan kata “diperbaiki”, maka dapat dijadikan satu entitas sehingga mengurangi informasi yang berlebihan (*redundancy*). Dengan demikian, akan terdapat delapan kata kunci permasalahan yakni “bad”, “email”, “error”, “fix”, “login”, “ticket”, “time”, dan “update”. Sebagai perbandingan, perhitungan kata dengan frekuensi terbanyak juga dilakukan, dengan proses perhitungannya ditunjukkan pada Gambar 4.17 dan 4.18 berikut ini.



```

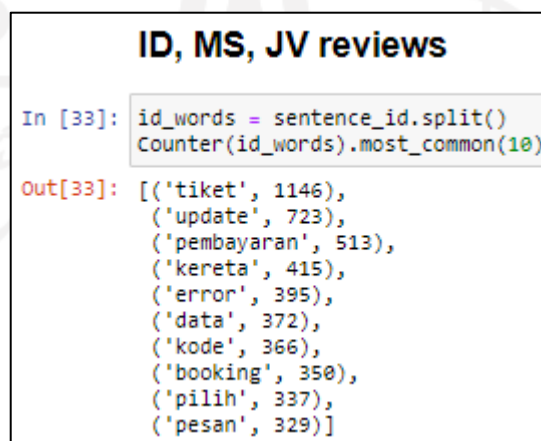
EN reviews

In [32]: en_words = sentence_en.split()
          Counter(en_words).most_common(10)

Out[32]: [('fix', 138),
          ('bad', 98),
          ('ticket', 94),
          ('update', 73),
          ('server', 61),
          ('login', 57),
          ('train', 57),
          ('error', 56),
          ('use', 54),
          ('booking', 53)]

```

Gambar 4.17 Perhitungan frekuensi kata berlabel EN



```

ID, MS, JV reviews

In [33]: id_words = sentence_id.split()
          Counter(id_words).most_common(10)

Out[33]: [('tiket', 1146),
          ('update', 723),
          ('pembayaran', 513),
          ('kereta', 415),
          ('error', 395),
          ('data', 372),
          ('kode', 366),
          ('booking', 350),
          ('pilih', 337),
          ('pesan', 329)]

```

Gambar 4.18 Perhitungan frekuensi kata berlabel ID – MS – JV

Hasil dari perhitungan frekuensi kata menunjukkan bahwa 6 (enam) dari 8 (delapan) kata kunci permasalahan masuk ke dalam sepuluh besar (*top 10*) kata dengan frekuensi terbanyak, sedangkan kata “email” dan “time” berada di luar sepuluh besar.

Kata kunci yang diperoleh akan digunakan sebagai “dampak” atau *consequent* yang dikeluhkan oleh pengguna KAI Access, atau dalam konteks diagram sebab-akibat (*fishbone diagram*) kata kunci tersebut akan menjadi “kepala ikan”. Untuk mengetahui penyebab permasalahan tiap kata kunci, maka dilakukan pengecekan asosiasi kata (*association rules*) menggunakan algoritma apriori. Proses pengecekan dilakukan terpisah berdasarkan kedua jenis bahasa pada ulasan, namun *coding* yang digunakan relatif sama, serta hasilnya digunakan secara bersamaan supaya memperkaya informasi yang akan diperoleh. Langkah pertama dalam algoritma apriori yakni menentukan *unique words* atau kamus kata dari data ulasan, dengan penerapannya melalui Python menggunakan *module* *mlxtend* seperti yang ditunjukkan pada Gambar 4.19 dan 4.20.

```

EN reviews

In [33]: import numpy as np
         from mlxtend.frequent_patterns import apriori, association_rules

In [34]: apriori_en = ''
         for val in df_en.Clean_Review:
             val = str(val)
             kotoba1 = val.split()
             apriori_en += ' '.join([word for word in kotoba1 if word not in custom_stopword_en])+'|'

In [35]: bun1 = []
         header = []
         kotoba1_spl = apriori_en.split('|')
         for each in kotoba1_spl:
             newWord = []
             kotoba1_splHeaderIndex = 0
             kotoba1_splHeader = []
             eachWord = each.split()
             for word in eachWord:
                 newWord.append(' '.join(e for e in word if e.isalnum()))
                 kotoba1_splHeader.append('header' + str(kotoba1_splHeaderIndex))
                 kotoba1_splHeaderIndex += 1
             if(len(header) < len(kotoba1_splHeader)):
                 header = kotoba1_splHeader
             bun1.append(newWord)

```

Gambar 4.19 Tahap penentuan *unique words* (langkah pertama)

```

In [36]: table_en = pd.DataFrame(bun1)
         table_en
...

In [37]: pd.DataFrame(bun1).to_csv('Split Bun EN.csv', header=header, index=None)

In [38]: words_table_en = pd.read_csv('Split Bun EN.csv')
         uniqueWordEN = words_table_en.header0.unique()
         pd.DataFrame(uniqueWordEN)

Out[38]:
          0
0  rubbish
1    cant
2  managed
3    apps
4   login
...     ...
404  ocd
405  layout
406  omg
407  pitty
408  system

409 rows x 1 columns

```

Gambar 4.20 Tahap penentuan *unique words* (langkah kedua)

Selanjutnya yakni menerapkan *one hot encoding* atau pengecekan “eksistensi” kata pada data ulasan yang mengacu pada *unique words*, dengan implementasi proses tersebut di Python ditunjukkan pada Gambar 4.21, serta *output* proses yang ditampilkan pada Gambar 4.22. Tahap ini berlaku untuk ulasan berbahasa Inggris maupun Indonesia – Melayu – Jawa.

```

In [39]: # One Hot Encoding
         encoded_vals_en = []
         for index, row in table_en.iterrows():
             labels = {}
             uncommons = list(set(uniqueWordEN) - set(row))
             commons = list(set(uniqueWordEN).intersection(row))
             for uc in uncommons:
                 labels[uc] = 0
             for com in commons:
                 labels[com] = 1
             encoded_vals_en.append(labels)
         encoded_vals_en[0]
         ohe_df_en = pd.DataFrame(encoded_vals_en)

```

Gambar 4.21 Penerapan *one hot encoding*

```
In [57]: ohe_df_en
```

```
Out[57]:
```

	NaN	fail	services	diupdate	said	unstable	hate	needs	disappointed	butud	...	installed	wth	convenience	log	need	rubbish	shame	trip	
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	1	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	1	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1042	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
1043	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
1044	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
1045	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
1046	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0

Gambar 4.22 Hasil proses *one hot encoding* dari ulasan berlabel EN

Setelah proses *one hot encoding* dilakukan, langkah berikutnya adalah menganalisis *association rules* atau hubungan antar kata. Proses penentuan *association rules* ditunjukkan pada Gambar 4.23, kemudian *output* dari proses ditampilkan pada Gambar 4.24 hingga 4.26 untuk sumber ulasan berbahasa Inggris, serta Gambar 4.27 hingga 4.30 untuk sumber ulasan berbahasa Indonesia – Melayu – Jawa. Untuk *association rules* bahasa Inggris diperoleh sebanyak 98 buah, sedangkan untuk yang berbahasa Indonesia – Melayu – Jawa diperoleh *rules* sebanyak 164 buah. *Association rules* yang digunakan mengacu pada parameter *lift* dengan nilai  $\geq 1$  yang menyatakan bahwa *rules* valid, dengan korelasi antar kata yang semakin kuat bila nilai *lift* semakin besar.

```
In [42]: rules_en = association_rules(freq_items_en, metric="lift", min_threshold=1)
```

```
In [66]: ar_lift_en = rules_en.sort_values(by='lift', ascending=False)
```

...

```
In [67]: ar_lift_en1 = ar_lift_en.drop(columns=['antecedent support', 'consequent support', 'leverage', 'conviction'])
```

Gambar 4.23 Proses analisis *association rules*



	antecedents	consequents	support	confidence	lift
95	(old)	(better, version)	0.011461	0.400000	23.266667
94	(better, version)	(old)	0.011461	0.666667	23.266667
6	(choose)	(seat)	0.013372	0.666667	18.864865
7	(seat)	(choose)	0.013372	0.378378	18.864865
97	(version)	(old, better)	0.011461	0.235294	17.596639
92	(old, better)	(version)	0.011461	0.857143	17.596639
96	(better)	(old, version)	0.011461	0.300000	16.531579
93	(old, version)	(better)	0.011461	0.631579	16.531579
91	(payment)	(linkaja)	0.010506	0.261905	13.057823
90	(linkaja)	(payment)	0.010506	0.523810	13.057823
44	(old)	(version)	0.018147	0.633333	13.001961
45	(version)	(old)	0.018147	0.372549	13.001961
10	(offline)	(you)	0.019102	0.338983	12.675545
11	(you)	(offline)	0.019102	0.714286	12.675545
46	(old)	(better)	0.013372	0.466667	12.215000
47	(better)	(old)	0.013372	0.350000	12.215000
49	(server)	(busy)	0.033429	0.443038	11.893866
48	(busy)	(server)	0.033429	0.897436	11.893866
67	(train)	(local)	0.023878	0.403226	11.410201
66	(local)	(train)	0.023878	0.675676	11.410201
89	(version)	(better)	0.017192	0.352941	9.238235
88	(better)	(version)	0.017192	0.450000	9.238235
86	(updated)	(version)	0.010506	0.423077	8.685520
87	(version)	(updated)	0.010506	0.215686	8.685520
75	(account)	(email)	0.010506	0.250000	7.270833
74	(email)	(account)	0.010506	0.305556	7.270833
40	(buy)	(ticket)	0.023878	0.757576	7.145782
41	(ticket)	(buy)	0.023878	0.225225	7.145782
21	(ticket)	(booked)	0.010506	0.099099	6.917117
20	(booked)	(ticket)	0.010506	0.733333	6.917117
2	(schedule)	(train)	0.011461	0.342857	5.789862
3	(train)	(schedule)	0.011461	0.193548	5.789862
77	(account)	(login)	0.012416	0.295455	4.549131
76	(login)	(account)	0.012416	0.191176	4.549131
1	(this)	(fix)	0.010506	0.578947	4.490058
0	(fix)	(this)	0.010506	0.081481	4.490058
9	(connection)	(offline)	0.010506	0.244444	4.337853
8	(offline)	(connection)	0.010506	0.186441	4.337853
23	(ticket)	(book)	0.019102	0.180180	3.849972
22	(book)	(ticket)	0.019102	0.408163	3.849972
63	(train)	(book)	0.010506	0.177419	3.790981
62	(book)	(train)	0.010506	0.224490	3.790981

Gambar 4.24 Association rules pada ulasan berlabel EN (bagian 1)

5	(schedule)	(fix)	0.015282	0.457143	3.545397
4	(fix)	(schedule)	0.015282	0.118519	3.545397
16	(check)	(ticket)	0.010506	0.354839	3.346992
17	(ticket)	(check)	0.010506	0.099099	3.346992
85	(tickets)	(fix)	0.011461	0.428571	3.323810
84	(fix)	(tickets)	0.011461	0.088889	3.323810
42	(local)	(ticket)	0.011461	0.324324	3.059167
43	(ticket)	(local)	0.011461	0.108108	3.059167
26	(ticket)	(try)	0.012416	0.117117	2.919562
27	(try)	(ticket)	0.012416	0.309524	2.919562
19	(ticket)	(train)	0.017192	0.162162	2.738448
18	(train)	(ticket)	0.017192	0.290323	2.738448
56	(work)	(fix)	0.012416	0.342105	2.653216
57	(fix)	(work)	0.012416	0.096296	2.653216
13	(fix)	(offline)	0.019102	0.148148	2.629002
12	(offline)	(fix)	0.019102	0.338983	2.629002
73	(fix)	(problem)	0.011461	0.088889	2.515315
72	(problem)	(fix)	0.011461	0.324324	2.515315
31	(ticket)	(use)	0.012416	0.117117	2.502482
30	(use)	(ticket)	0.012416	0.265306	2.502482
32	(version)	(ticket)	0.012416	0.254902	2.404346
33	(ticket)	(version)	0.012416	0.117117	2.404346
68	(fix)	(book)	0.014327	0.111111	2.374150
69	(book)	(fix)	0.014327	0.306122	2.374150
81	(version)	(update)	0.011461	0.235294	2.324084
80	(update)	(version)	0.011461	0.113208	2.324084
35	(ticket)	(booking)	0.011461	0.108108	2.309983
34	(booking)	(ticket)	0.011461	0.244898	2.309983
54	(fix)	(connection)	0.012416	0.096296	2.240494
55	(connection)	(fix)	0.012416	0.288889	2.240494
24	(app)	(ticket)	0.010506	0.234043	2.207591
25	(ticket)	(app)	0.010506	0.099099	2.207591
38	(fix)	(ticket)	0.029608	0.229630	2.165966
39	(ticket)	(fix)	0.029608	0.279279	2.165966
71	(app)	(fix)	0.010506	0.234043	1.815130
70	(fix)	(app)	0.010506	0.081481	1.815130
59	(time)	(fix)	0.014327	0.227273	1.762626
58	(fix)	(time)	0.014327	0.111111	1.762626
65	(train)	(fix)	0.012416	0.209677	1.626165
64	(fix)	(train)	0.012416	0.096296	1.626165
82	(update)	(fix)	0.021012	0.207547	1.609644
83	(fix)	(update)	0.021012	0.162963	1.609644
79	(login)	(fix)	0.013372	0.205882	1.596732

Gambar 4.25 Association rules pada ulasan berlabel EN (bagian 2)

78	(fix)	(login)	0.013372	0.103704	1.596732
15	(ticket)	(time)	0.010506	0.099099	1.572072
14	(time)	(ticket)	0.010506	0.166667	1.572072
29	(ticket)	(update)	0.016237	0.153153	1.512749
28	(update)	(ticket)	0.016237	0.160377	1.512749
51	(server)	(bad)	0.010506	0.139241	1.301650
50	(bad)	(server)	0.010506	0.098214	1.301650
36	(bad)	(ticket)	0.014327	0.133929	1.263272
37	(ticket)	(bad)	0.014327	0.135135	1.263272
61	(error)	(fix)	0.010506	0.159420	1.236393
60	(fix)	(error)	0.010506	0.081481	1.236393
53	(server)	(fix)	0.011461	0.151899	1.178059
52	(fix)	(server)	0.011461	0.088889	1.178059

Gambar 4.26 Association rules pada ulasan berlabel EN (bagian 3)

	antecedents	consequents	support	confidence	lift
99	(close)	(force)	0.012280	0.585586	34.825893
98	(force)	(close)	0.012280	0.730337	34.825893
153	(data, server)	(mengambil)	0.017381	0.647887	32.973727
156	(mengambil)	(data, server)	0.017381	0.884615	32.973727
82	(koneksi)	(internet)	0.010769	0.407143	16.325812
83	(internet)	(koneksi)	0.010769	0.431818	16.325812
104	(lokal)	(ka)	0.014548	0.306773	15.175224
105	(ka)	(lokal)	0.014548	0.719626	15.175224
155	(data)	(mengambil, server)	0.017381	0.260623	14.675306
154	(mengambil, server)	(data)	0.017381	0.978723	14.675306
114	(data)	(mengambil)	0.019082	0.286119	14.561805
115	(mengambil)	(data)	0.019082	0.971154	14.561805
157	(server)	(data, mengambil)	0.017381	0.238342	12.490535
152	(data, mengambil)	(server)	0.017381	0.910891	12.490535
12	(mengambil)	(server)	0.017759	0.903846	12.393932
13	(server)	(mengambil)	0.017759	0.243523	12.393932
29	(terbaru)	(versi)	0.012847	0.507463	11.884956
28	(versi)	(terbaru)	0.012847	0.300885	11.884956
11	(server)	(sibuk)	0.010202	0.139896	11.051813
10	(sibuk)	(server)	0.010202	0.805970	11.051813
136	(pembayaran)	(metode)	0.010013	0.112766	9.182619
137	(metode)	(pembayaran)	0.010013	0.815385	9.182619
84	(verifikasi)	(kode)	0.011147	0.500000	8.428344
85	(kode)	(verifikasi)	0.011147	0.187898	8.428344
6	(nomor)	(hp)	0.013981	0.308333	7.189464
7	(hp)	(nomor)	0.013981	0.325991	7.189464
160	(kereta, tiket)	(lokal)	0.010769	0.337278	7.112402
161	(lokal)	(kereta, tiket)	0.010769	0.227092	7.112402
34	(email)	(password)	0.011525	0.231939	6.935898
35	(password)	(email)	0.011525	0.344633	6.935898
124	(kursi)	(pilih)	0.023994	0.390769	6.650616
125	(pilih)	(kursi)	0.023994	0.408360	6.650616
97	(seat)	(pilih)	0.013603	0.387097	6.588113
96	(pilih)	(seat)	0.013603	0.231511	6.588113
121	(kereta)	(lokal)	0.024561	0.305164	6.435198
120	(lokal)	(kereta)	0.024561	0.517928	6.435198
81	(password)	(login)	0.015114	0.451977	6.102848
80	(login)	(password)	0.015114	0.204082	6.102848
30	(akun)	(email)	0.013225	0.285714	5.750136
31	(email)	(akun)	0.013225	0.266160	5.750136
159	(lokal, tiket)	(kereta)	0.010769	0.459677	5.711438
162	(kereta)	(lokal, tiket)	0.010769	0.133803	5.711438

Gambar 4.27 Association rules pada ulasan berlabel ID – MS – JV (bagian 1)

2	(bayar)	(link)	0.010769	0.239496	5.609076
3	(link)	(bayar)	0.010769	0.252212	5.609076
86	(pembayaran)	(linkaja)	0.017570	0.197872	5.570948
87	(linkaja)	(pembayaran)	0.017570	0.494681	5.570948
14	(data)	(server)	0.026828	0.402266	5.516050
15	(server)	(data)	0.026828	0.367876	5.516050
139	(link)	(pembayaran)	0.019649	0.460177	5.182376
138	(pembayaran)	(link)	0.019649	0.221277	5.182376
26	(update)	(terbaru)	0.016437	0.117886	4.656504
27	(terbaru)	(update)	0.016437	0.649254	4.656504
116	(lokal)	(beli)	0.010769	0.227092	4.277566
117	(beli)	(lokal)	0.010769	0.202847	4.277566
106	(kereta)	(jadwal)	0.015492	0.192488	4.227553
107	(jadwal)	(kereta)	0.015492	0.340249	4.227553
32	(login)	(email)	0.014359	0.193878	3.901878
33	(email)	(login)	0.014359	0.288973	3.901878
132	(tiket)	(beli)	0.041564	0.204082	3.844143
133	(beli)	(tiket)	0.041564	0.782918	3.844143
79	(tiket)	(pembelian)	0.012091	0.059369	3.832210
78	(pembelian)	(tiket)	0.012091	0.780488	3.832210
91	(kode)	(booking)	0.015114	0.254777	3.725235
90	(booking)	(kode)	0.015114	0.220994	3.725235
95	(tiket)	(saldo)	0.010580	0.051948	3.273346
94	(saldo)	(tiket)	0.010580	0.666667	3.273346
72	(akun)	(login)	0.010769	0.232653	3.141410
73	(login)	(akun)	0.010769	0.145408	3.141410
148	(pesan)	(tiket)	0.038164	0.623457	3.061184
149	(tiket)	(pesan)	0.038164	0.187384	3.061184
53	(tiket)	(pesen)	0.015681	0.076994	2.953127
52	(pesen)	(tiket)	0.015681	0.601449	2.953127
1	(bayar)	(pembayaran)	0.011714	0.260504	2.933721
0	(pembayaran)	(bayar)	0.011714	0.131915	2.933721
118	(pembayaran)	(lokal)	0.012091	0.136170	2.871510
119	(lokal)	(pembayaran)	0.012091	0.254980	2.871510
22	(pembayaran)	(gagal)	0.012091	0.136170	2.699434
23	(gagal)	(pembayaran)	0.012091	0.239700	2.699434
45	(update)	(versi)	0.014548	0.104336	2.443587
44	(versi)	(update)	0.014548	0.340708	2.443587
122	(lokal)	(tiket)	0.023427	0.494024	2.425667
123	(tiket)	(lokal)	0.023427	0.115028	2.425667
50	(pemesanan)	(tiket)	0.016437	0.488764	2.399841
51	(tiket)	(pemesanan)	0.016437	0.080705	2.399841
20	(orang)	(tiket)	0.015492	0.443243	2.176333

Gambar 4.28 Association rules pada ulasan berlabel ID – MS – JV (bagian 2)

21	(tiket)	(orang)	0.015492	0.076067	2.176333
100	(2)	(tiket)	0.011525	0.438849	2.154756
101	(tiket)	(2)	0.011525	0.056586	2.154756
158	(lokal, kereta)	(tiket)	0.010769	0.438462	2.152854
163	(tiket)	(lokal, kereta)	0.010769	0.052876	2.152854
61	(1)	(tiket)	0.014170	0.421348	2.068828
60	(tiket)	(1)	0.014170	0.069573	2.068828
55	(tiket)	(stasiun)	0.010958	0.053803	1.977646
54	(stasiun)	(tiket)	0.010958	0.402778	1.977646
126	(penumpang)	(tiket)	0.011714	0.397436	1.951418
127	(tiket)	(penumpang)	0.011714	0.057514	1.951418
146	(kereta)	(tiket)	0.031929	0.396714	1.947871
147	(tiket)	(kereta)	0.031929	0.156772	1.947871
112	(pembayaran)	(kode)	0.010202	0.114894	1.936726
113	(kode)	(pembayaran)	0.010202	0.171975	1.936726
9	(server)	(error)	0.013981	0.191710	1.831625
8	(error)	(server)	0.013981	0.133574	1.831625
128	(cek)	(tiket)	0.011714	0.358382	1.759660
129	(tiket)	(cek)	0.011714	0.057514	1.759660
49	(ini)	(update)	0.011336	0.237154	1.700890
48	(update)	(ini)	0.011336	0.081301	1.700890
140	(pembayaran)	(kereta)	0.011903	0.134043	1.665463
141	(kereta)	(pembayaran)	0.011903	0.147887	1.665463
102	(pas)	(tiket)	0.014359	0.337778	1.658495
103	(tiket)	(pas)	0.014359	0.070501	1.658495
57	(tiket)	(lagi)	0.013036	0.064007	1.652640
56	(lagi)	(tiket)	0.013036	0.336585	1.652640
92	(booking)	(tiket)	0.022860	0.334254	1.641194
93	(tiket)	(booking)	0.022860	0.112245	1.641194
151	(tiket)	(ini)	0.015681	0.076994	1.610797
150	(ini)	(tiket)	0.015681	0.328063	1.610797
5	(tiket)	(bayar)	0.014359	0.070501	1.567905
4	(bayar)	(tiket)	0.014359	0.319328	1.567905
88	(linkaja)	(tiket)	0.011336	0.319149	1.567027
89	(tiket)	(linkaja)	0.011336	0.055659	1.567027
76	(bikin)	(tiket)	0.010391	0.305556	1.500283
77	(tiket)	(bikin)	0.010391	0.051020	1.500283
145	(tiket)	(link)	0.013036	0.064007	1.499076
144	(link)	(tiket)	0.013036	0.305310	1.499076
63	(diperbaiki)	(error)	0.013792	0.156317	1.493476
62	(error)	(diperbaiki)	0.013792	0.131769	1.493476
74	(akun)	(tiket)	0.013225	0.285714	1.402862
75	(tiket)	(akun)	0.013225	0.064935	1.402862

Gambar 4.29 Association rules pada ulasan berlabel ID – MS – JV (bagian 3)

39	(login)	(update)	0.014170	0.191327	1.372210
38	(update)	(login)	0.014170	0.101626	1.372210
142	(pembayaran)	(tiket)	0.024750	0.278723	1.368537
143	(tiket)	(pembayaran)	0.024750	0.121521	1.368537
58	(susah)	(tiket)	0.011903	0.269231	1.321928
59	(tiket)	(susah)	0.011903	0.058442	1.321928
71	(tiket)	(error)	0.027773	0.136364	1.302839
70	(error)	(tiket)	0.027773	0.265343	1.302839
18	(bisa)	(tiket)	0.012847	0.264591	1.299149
19	(tiket)	(bisa)	0.012847	0.063080	1.299149
111	(tiket)	(bagus)	0.010391	0.051020	1.292110
110	(bagus)	(tiket)	0.010391	0.263158	1.292110
108	(jadwal)	(tiket)	0.011903	0.261411	1.283532
109	(tiket)	(jadwal)	0.011903	0.058442	1.283532
43	(update)	(kursi)	0.010391	0.074526	1.213738
42	(kursi)	(update)	0.010391	0.169231	1.213738
24	(gagal)	(tiket)	0.012280	0.243446	1.195323
25	(tiket)	(gagal)	0.012280	0.060297	1.195323
68	(kereta)	(error)	0.010013	0.124413	1.188662
69	(error)	(kereta)	0.010013	0.095668	1.188662
135	(tiket)	(masuk)	0.010013	0.049165	1.188269
134	(masuk)	(tiket)	0.010013	0.242009	1.188269
17	(server)	(tiket)	0.017570	0.240933	1.182984
16	(tiket)	(server)	0.017570	0.086271	1.182984
65	(tiket)	(diperbaiki)	0.020971	0.102968	1.167049
64	(diperbaiki)	(tiket)	0.020971	0.237687	1.167049
41	(update)	(booking)	0.010769	0.077236	1.129306
40	(booking)	(update)	0.010769	0.157459	1.129306
130	(data)	(tiket)	0.014925	0.223796	1.098843
131	(tiket)	(data)	0.014925	0.073284	1.098843
67	(error)	(pembayaran)	0.010013	0.095668	1.077383
66	(pembayaran)	(error)	0.010013	0.112766	1.077383
36	(update)	(diperbaiki)	0.013225	0.094851	1.075045
37	(diperbaiki)	(update)	0.013225	0.149893	1.075045
47	(tiket)	(update)	0.029473	0.144712	1.037890
46	(update)	(tiket)	0.029473	0.211382	1.037890

Gambar 4.30 *Association rules* pada ulasan berlabel ID – MS – JV (bagian 4)

Dari kedua hasil tersebut, kemudian dicari kata yang berhubungan dengan kata kunci permasalahan. Tabel 4.1 menunjukkan kata-kata yang berhasil dihimpun.

Tabel 4.1 *Association rules* terkait kata kunci permasalahan

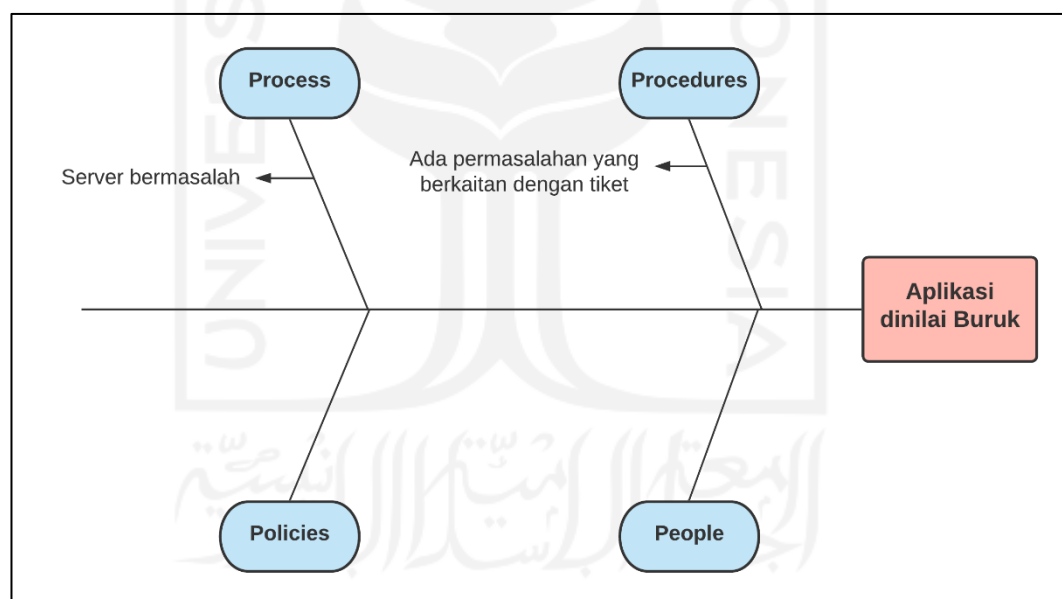
<i>Consequent</i>	<i>Antecedent</i>	<i>Support</i>	<i>Confidence</i>	<i>Lift</i>
Bad	Server	0.0105	0.1392	1.3016
	Ticket	0.0143	0.1351	1.2632
Email	Account	0.0105	0.25	7.2708
	Password	0.0115	0.3446	6.9359
Error	Server	0.014	0.1917	1.8316
	Diperbaiki	0.0138	0.1563	1.4934
	Tiket	0.0277	0.1363	1.3028
	Kereta	0.01	0.1244	1.1886
	Pembayaran	0.01	0.1127	1.0773

<i>Consequent</i>	<i>Antecedent</i>	<i>Support</i>	<i>Confidence</i>	<i>Lift</i>
Fix/Perbaiki	Schedule	0.0152	0.4571	3.5454
	Ticket	0.0114	0.4285	3.3238
	Work	0.0124	0.3421	2.6532
	Offline	0.0191	0.339	2.629
	Problem	0.0114	0.3243	2.5153
	Connection	0.0124	0.2889	2.2405
	App	0.0105	0.234	1.8151
	Time	0.0143	0.2272	1.7626
	Update	0.021	0.2075	1.6096
	Login	0.0133	0.2059	1.5967
	Error	0.0105	0.1594	1.2364
Server	0.0114	0.1519	1.178	
Login	Account	0.0124	0.2954	4.5491
	Password	0.0151	0.452	6.1028
	Email	0.0143	0.289	3.9019
	Update	0.0141	0.1016	1.3722
Ticket	Buy	0.0239	0.7576	7.1457
	Booked	0.0105	0.7333	6.9171
	Book	0.0191	0.4082	3.845
	Check	0.0105	0.3548	3.347
	Version	0.0124	0.2549	2.4043
	Time	0.0105	0.1667	1.572
	Update	0.0162	0.1604	1.5127
	Kereta Lokal	0.0107	0.2271	7.1124
	Saldo	0.0106	0.6667	3.2733
	Orang	0.0155	0.4432	2.1763
	Bayar	0.0143	0.3193	1.5679
	LinkAja	0.0113	0.3191	1.567
	Bikin	0.0104	0.3055	1.5002
	Akun	0.0132	0.2857	1.4028
	Jadwal	0.0119	0.2614	1.2835
	Gagal	0.0123	0.2434	1.1953
	Server	0.0176	0.2409	1.183
Data	0.0149	0.2238	1.0988	
Time	(Sudah disebutkan di "antecedent")			
Update	Kursi	0.0104	0.1692	1.2137
	Booking	0.0107	0.1574	1.1293

Untuk kata kunci “time”, hanya terdapat dua *rules* yang telah menjadi *antecedent*, masing-masing bagi *consequent* “fix” dan “ticket” sehingga tidak perlu disebutkan kembali.

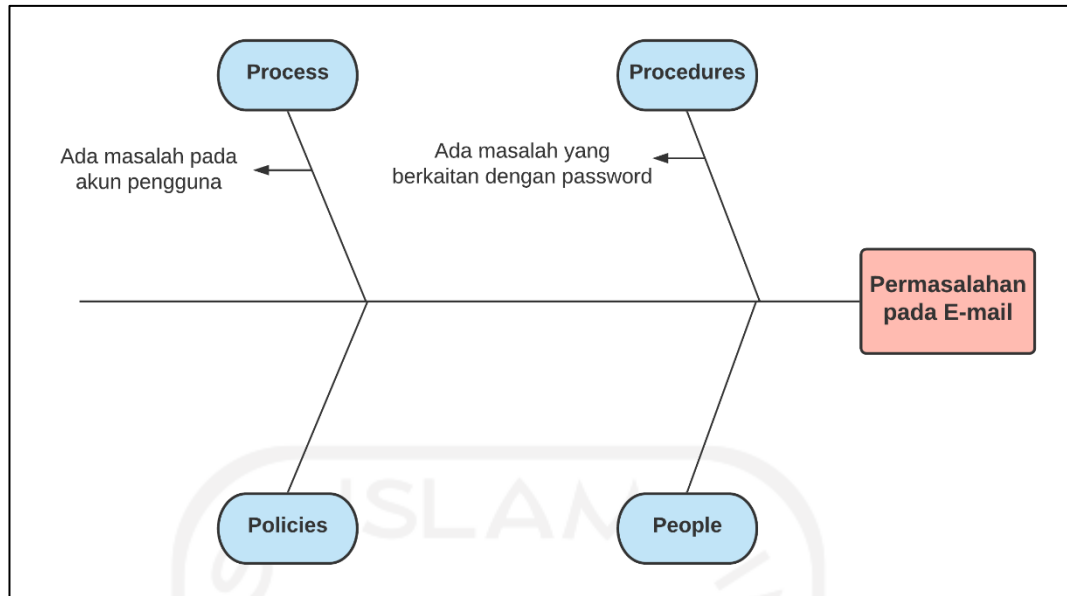
#### 4.2.8 Analisis Akar Masalah dengan *Fishbone Diagram*

Dari hasil *association rules* pada Tabel 4.1, tahap berikutnya yakni penyusunan *fishbone diagram*, dengan *consequent* (dampak yang dikeluarkan) berperan sebagai “kepala ikan”, serta *antecedent* berperan sebagai “tulang ikan” atau penyebab permasalahan. Karena PT. KAI merupakan industri jasa, maka jenis diagram yang digunakan yakni 4Ps *fishbone diagram*. Gambar 4.31 hingga 4.37 menunjukkan pengkategorian tiap permasalahan, dengan analisis akar permasalahan yang dibahas pada Bab V.

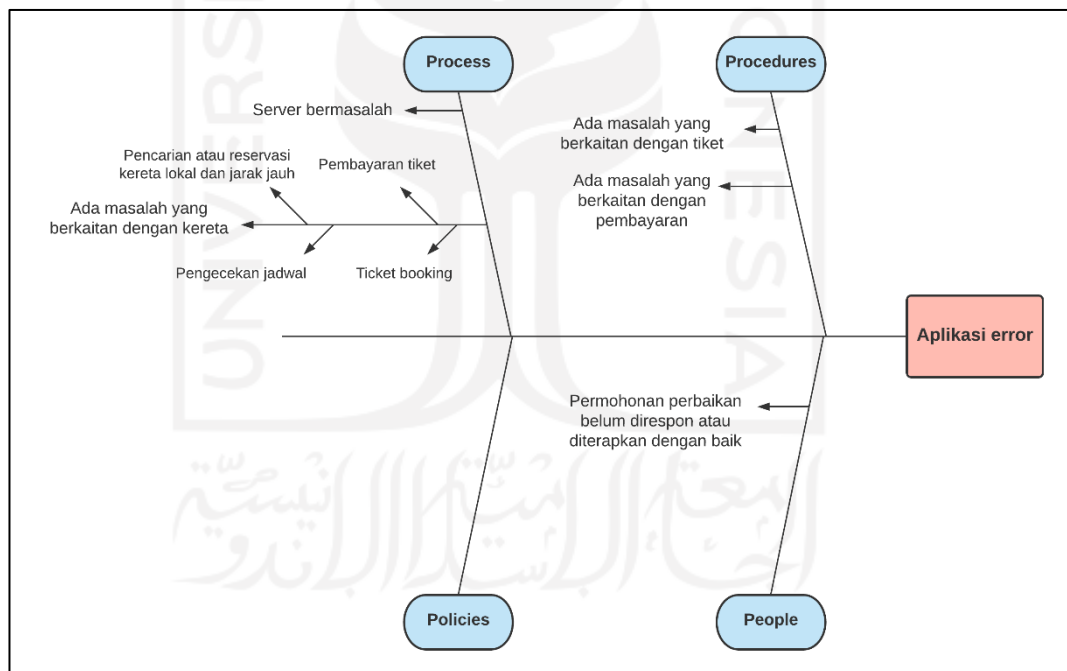


Gambar 4.31 *Fishbone diagram* untuk kata kunci “bad”

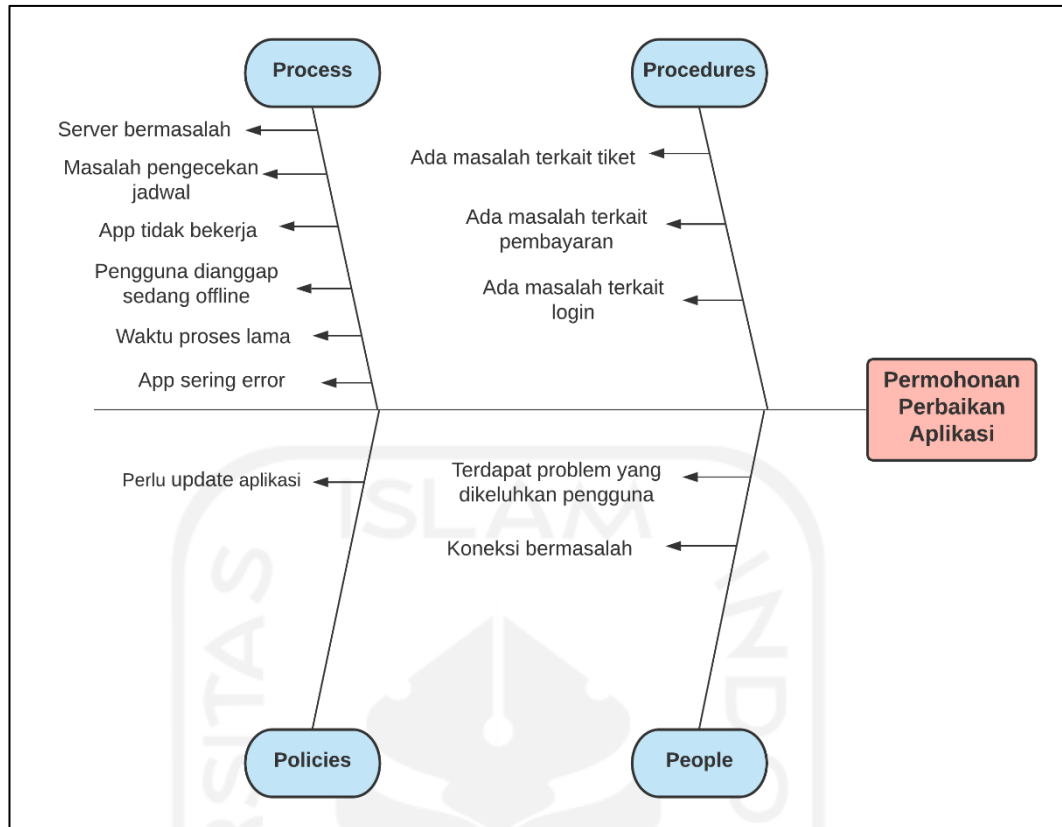




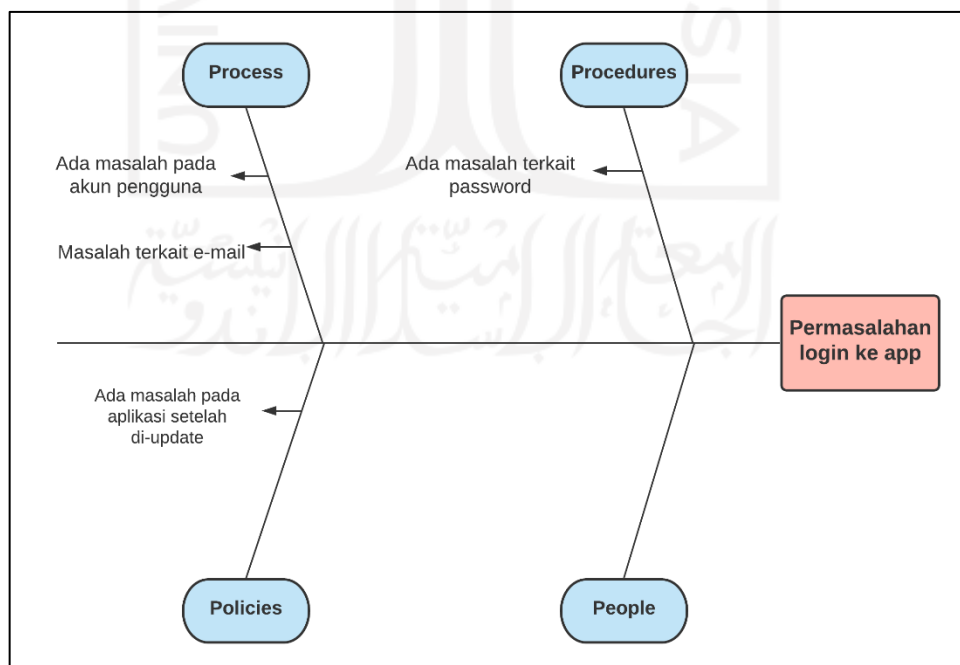
Gambar 4.32 *Fishbone diagram* untuk kata kunci "email"



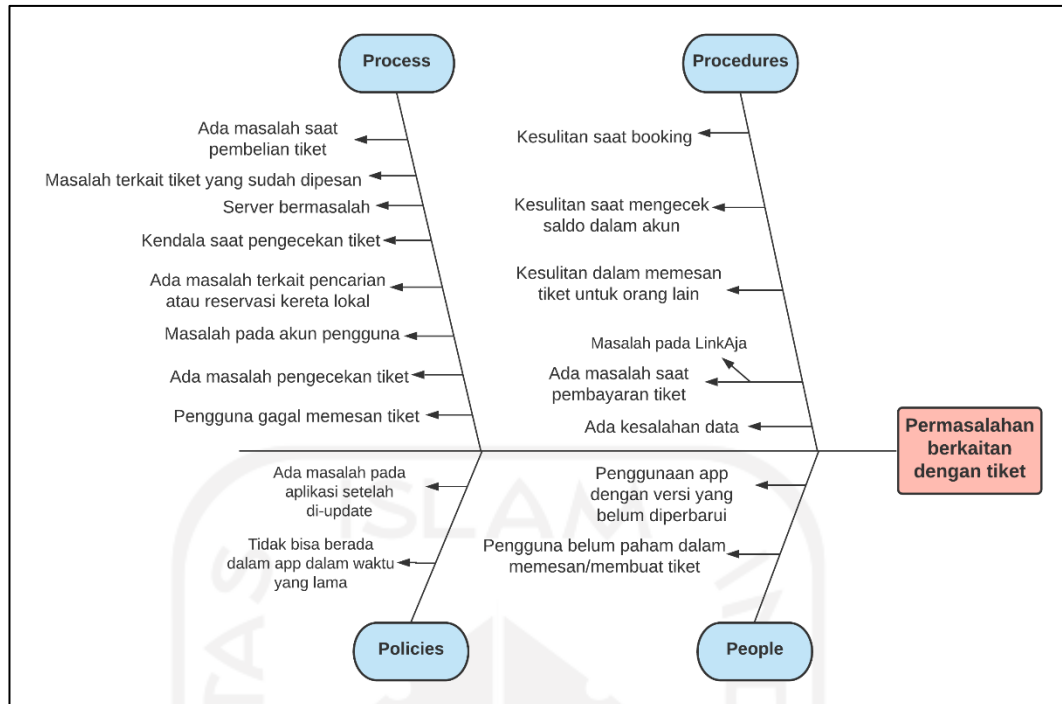
Gambar 4.33 *Fishbone diagram* untuk kata kunci "error"



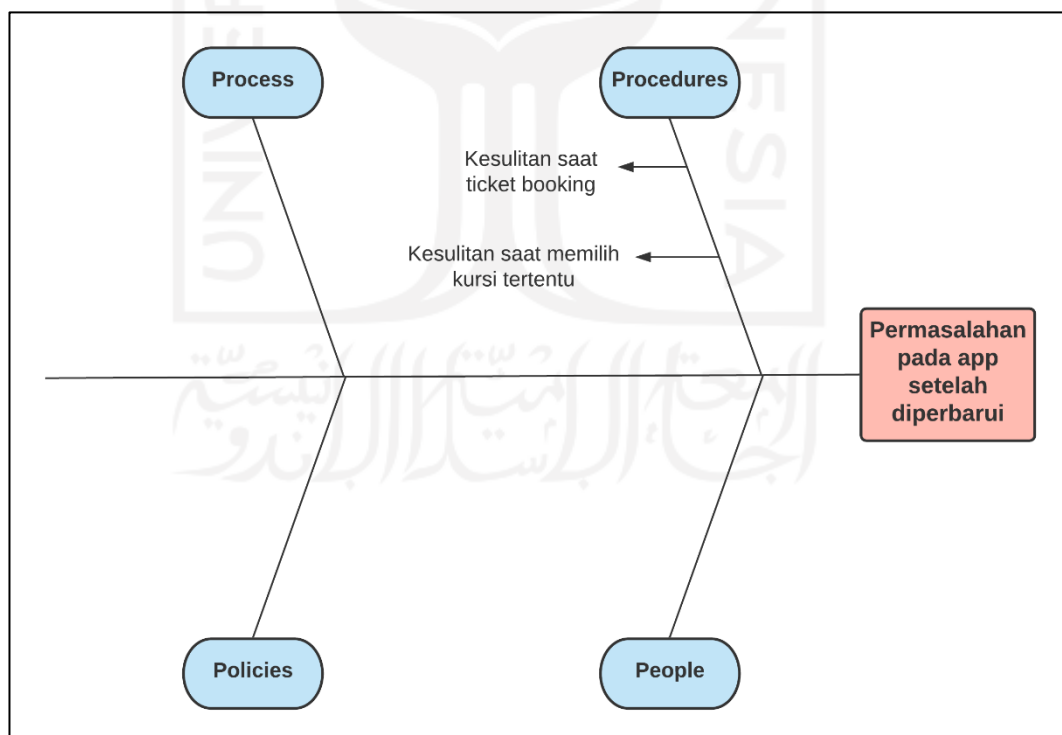
Gambar 4.34 *Fishbone diagram* untuk kata kunci “fix”



Gambar 4.35 *Fishbone diagram* untuk kata kunci “login”



Gambar 4.36 *Fishbone diagram* untuk kata kunci "ticket"



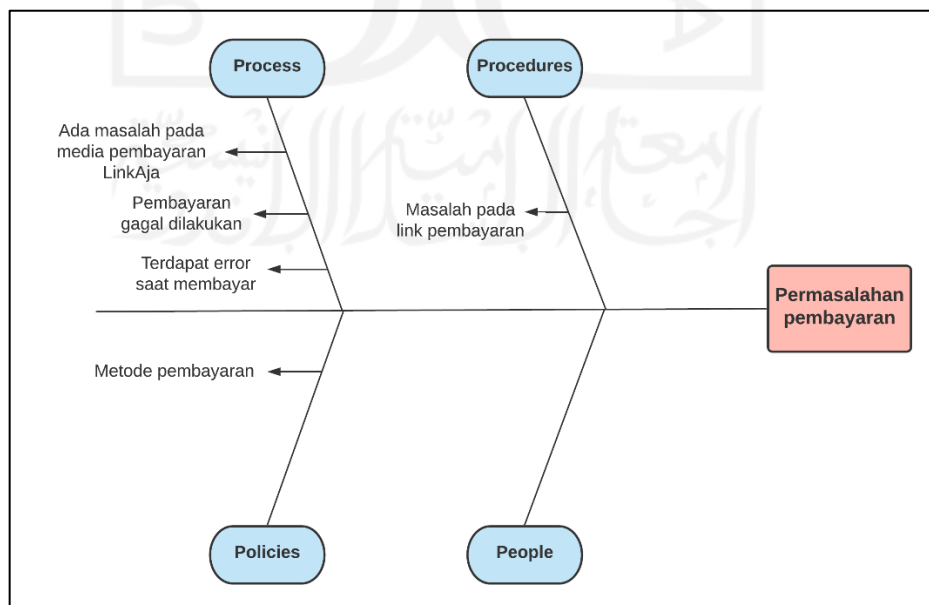
Gambar 4.37 *Fishbone diagram* untuk kata kunci "update"

Pada Gambar 4.33 dan 4.36 terdapat penyebab permasalahan yang memiliki cabang penyebab lanjutan, yakni permasalahan terkait kereta dan pembayaran tiket. Cabang penyebab yang tercantum pada *fishbone* didukung dengan beberapa *association rules*, yang ditampilkan pada Tabel 4.2 berikut ini.

Tabel 4.2 *Association rules* pendukung cabang penyebab lanjutan

<i>Consequent</i>	<i>Antecedent</i>	<i>Support</i>	<i>Confidence</i>	<i>Lift</i>
Masalah terkait kereta	Local	0.0105	0.5238	1.3016
	Schedule	0.0143	0.8154	1.2632
	Book	0.0105	0.4602	7.2708
	Ticket	0.0115	0.0957	6.9359
	Pembayaran	0.014	0.2397	1.8316
Masalah saat pembayaran tiket	LinkAja	0.0105	0.6757	13.0578
	Metode	0.01	0.3428	9.1826
	Link	0.0196	0.2245	5.1824
	Error	0.01	0.1621	1.0774
	Gagal	0.0121	0.134	2.6994

Untuk *consequent* “masalah pembayaran tiket”, *antecedent* pada *fishbone diagram* (mengacu pada Gambar 4.36) dijabarkan lebih jelas pada Gambar 4.38 di bawah ini.



Gambar 4.38 *Fishbone diagram* untuk kata kunci “pembayaran”

## BAB V

### PEMBAHASAN

#### 5.1 *Word Cloud* dan *Association Rules*

Berdasarkan hasil dari *word clouding*, dapat terlihat bila kata kunci dari keluhan utama para pengguna aplikasi KAI Access adalah “bad”, “email”, “error”, “fix”, “login”, “ticket”, “time”, dan “update”. Kemudian setelah meninjau *association rules*, untuk data ulasan berbahasa Inggris (EN) diperoleh *rules* sebanyak 98 buah serta untuk gabungan ulasan berbahasa Indonesia, Melayu dan Jawa (ID – MS – JV) sebanyak 164 buah. Dari kumpulan *rules* tersebut, terdapat 45 *rules* yang berhubungan dengan kata kunci permasalahan.

#### 5.2 *Fishbone Diagram*

Untuk masing-masing permasalahan yang teridentifikasi, adapun penyebab-penyebabnya mengacu pada *rules* yang diperoleh yakni sebagai berikut:

##### 5.2.1 Masalah Aplikasi dinilai Buruk (*Bad*)

Terdapat dua penyebab utama aplikasi KAI Access dinilai buruk oleh pengguna, yakni *server* yang bermasalah dan terdapat beberapa kendala yang ada hubungannya dengan tiket. Terkait penyebab yang disebutkan terakhir, kendala yang dimaksud beserta saran perbaikan akan dijelaskan lebih lanjut pada subbab 5.2.6 terkait masalah tiket. Sedangkan untuk mengatasi masalah pada *server*, perlu dilakukan pengecekan *server* secara berkala supaya layanan KAI Access dapat berjalan tanpa adanya masalah yang lebih besar yang disebabkan oleh *server*.

### 5.2.2 Masalah *E-mail*

Penyebab masalah yang berkaitan dengan *e-mail* di antaranya yakni adanya permasalahan pada akun pengguna aplikasi serta *password* pengguna. Adapun penyelesaian masalah yang dapat dilakukan yakni perlu dilakukan diskusi antara pihak penyelenggara KAI Access dan pengguna aplikasi, baik secara langsung maupun melalui media komunikasi jarak jauh, sehingga kedua belah pihak dapat menyelaraskan informasi dan kemudian dapat mengetahui letak kesalahan dari permasalahan tersebut.

### 5.2.3 Masalah *Error*

Berdasarkan *rules* yang diperoleh, terdapat lima penyebab permasalahan *error* pada aplikasi. Yang pertama adalah terdapat masalah pada *server*, dengan saran perbaikan yakni perlu dilakukannya pengecekan *server* secara berkala. Penyebab kedua yang teridentifikasi adalah permohonan perbaikan yang belum memenuhi ekspektasi dari para pengguna, yang mana diperlukan komunikasi yang lancar dan berkelanjutan antara pihak KAI Access dengan pengguna aplikasi, dengan harapan perbaikan aplikasi di masa yang akan datang dapat menjawab kebutuhan pengguna aplikasi selaku *end-user*. Kemudian penyebab kelima dari permasalahan *error* yakni terdapat permasalahan terkait kereta, yang mana permasalahan yang dimaksud meliputi masalah pembayaran tiket, pemesanan tiket, pengecekan jadwal kereta, serta pencarian atau reservasi kereta lokal dan jarak jauh. Adapun saran perbaikan untuk keempat masalah tersebut relatif sama, yakni melakukan komunikasi terhadap pengguna terkait rincian masalah tersebut, kemudian melakukan pengecekan sistem informasi dan perbaikan bila diperlukan. Selanjutnya, penyebab keempat dan kelima yakni terdapat permasalahan yang berkaitan dengan tiket dan pembayaran. Untuk masalah terkait tiket akan dijelaskan lebih lanjut pada subbab 5.2.6, sedangkan untuk masalah pembayaran akan dijelaskan lebih lanjut pada subbab 5.2.8.

#### 5.2.4 Masalah Permohonan Perbaikan Aplikasi (*Fix*)

Terdapat cukup banyak penyebab dari adanya permohonan perbaikan, di antaranya terdapat masalah yang berhubungan dengan tiket, pembayaran, dan *login*. Untuk masalah terkait tiket akan dijelaskan pada subbab 5.2.6, sedangkan untuk masalah pembayaran akan dijelaskan pada subbab 5.2.8, dan untuk masalah *login* akan dijelaskan pada subbab 5.2.5. Penyebab berikutnya adalah adanya hambatan koneksi antara pengguna dengan KAI Access, yang mana salah satu penyebab umumnya dikarenakan internet yang bermasalah. Masalah ini dapat diselesaikan dengan beberapa cara, yakni mengecek sinyal pada *smartphone* dan kuota paket data (khususnya bagi pengguna aplikasi), melakukan *restart* pada perangkat jaringan internet seperti *router*, hingga menghubungi penyedia layanan internet bila diperlukan. Kemudian untuk masalah pada *server*, perlu dilakukan pengecekan *server* secara berkala agar layanan dapat berjalan tanpa adanya masalah yang lebih besar yang disebabkan oleh *server*. Penyebab berikutnya yakni aplikasi yang seringkali *error* hingga berhenti bekerja (*force close*), serta adanya masalah saat pengecekan jadwal. Kedua penyebab tersebut dapat ditangani dengan melakukan pengecekan hingga perbaikan sistem informasi pada aplikasi. Selanjutnya yakni keluhan terkait lamanya waktu proses, yang mana dapat disebabkan dari koneksi internet atau sistem informasi (*internal*) aplikasi.

#### 5.2.5 Masalah Login

Adapun penyebab yang diketahui dari *association rules* untuk permasalahan *login* di antaranya yakni terdapat masalah pada *e-mail*, akun aplikasi KAI Access hingga *password* pengguna. Penyebab lainnya yakni muncul masalah pada aplikasi setelah pengguna melakukan *update* aplikasi. Adapun penyelesaian masalah yang dapat dilakukan untuk penyebab-penyebab tersebut yakni melakukan diskusi antara pihak penyelenggara KAI Access dan pengguna aplikasi, dengan harapan kedua belah pihak dapat menyelaraskan informasi dan kemudian dapat mengetahui letak kesalahan dari permasalahan tersebut.

### 5.2.6 Masalah Tiket

Pada permasalahan ini, terdapat banyak keluhan yang dapat dikelompokkan berdasarkan solusi perbaikan yang dapat dilakukan. Untuk keluhan masalah terkait aplikasi pasca-pembaruan, akun pengguna, tidak bisanya mengakses atau berada di dalam aplikasi dalam waktu yang lama, kemudian pengecekan, pemesanan hingga pembelian tiket kereta, diperlukan diskusi atau komunikasi antara pihak KAI Access dan pengguna aplikasi dengan harapan kedua belah pihak dapat menyelaraskan informasi dan kemudian dapat mengetahui letak kesalahan dari permasalahan tersebut. Selain itu, pengecekan dan perbaikan sistem informasi dapat dilakukan sebagai tindak lanjut atas diskusi yang telah dilakukan. Pembahasan lebih lanjut terkait masalah pembayaran dapat dilihat pada subbab 5.2.8. Penyebab lain dari permasalahan tiket yakni *server* yang bermasalah, dengan solusi yang dapat dilakukan berupa pengecekan *server* secara berkala. Penyebab berikutnya yakni adanya kesalahan data dan kesulitan saat pengecekan saldo dalam aplikasi. Kedua penyebab tersebut perlu didiskusikan dengan pihak KAI Access karena menyangkut privasi pengguna. Kemudian penyebab terakhir yang teridentifikasi adalah terdapat pengguna aplikasi yang belum memahami cara pembuatan tiket (mencakup pencarian hingga pembayaran), yang mana pihak KAI Access dapat melakukan komunikasi dengan pengguna yang mengeluhkan hal tersebut, hingga melakukan survey terkait proses pembuatan tiket yang dipahami atau diinginkan pengguna layanan kereta bila diperlukan.

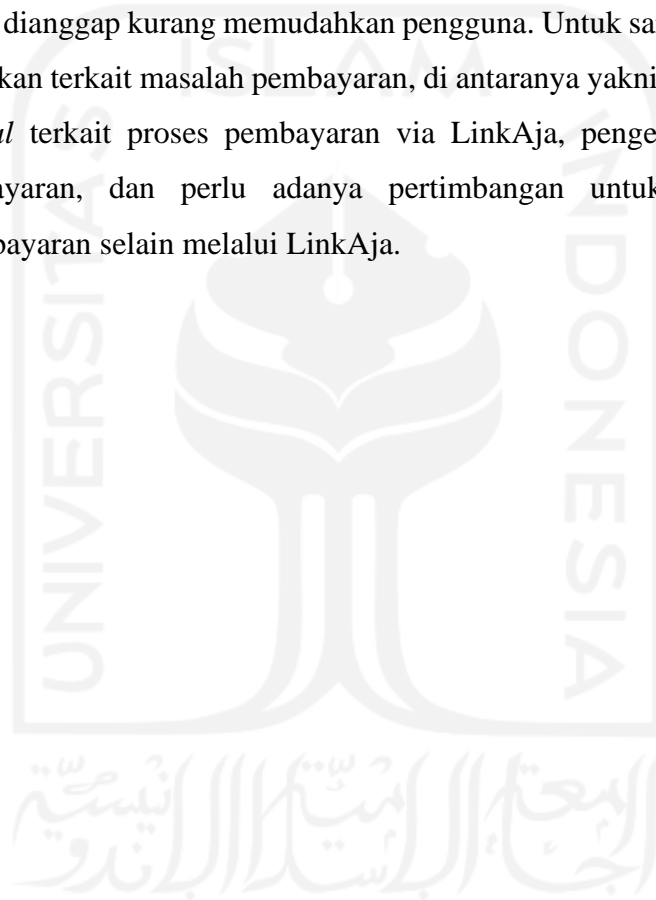
### 5.2.7 Masalah pada Aplikasi Setelah Pembaruan (*Update*)

Beberapa penyebab yang diketahui berdasarkan *association rules* untuk permasalahan ini di antaranya terdapat kesulitan saat memesan tiket dan memilih kursi kereta. Untuk keluhan tersebut, diperlukan komunikasi antara pihak KAI Access dan pengguna aplikasi yang selanjutnya pihak KAI Access dapat melakukan pengecekan dan perbaikan sistem informasi. Selain itu, diperlukan pula pengecekan koneksi internet oleh kedua belah pihak.



### 5.2.8 Masalah Pembayaran

Dari beberapa permasalahan yang telah dibahas di subbab sebelumnya, salah satu penyebab yang cukup sering muncul yakni adanya masalah terkait dengan pembelian atau pembayaran tiket. Adapun penyebab lanjutan yang teridentifikasi berdasarkan *association rules*, di antaranya terdapat masalah *internal* pada metode pembayaran LinkAja, pembayaran seringkali gagal dilakukan, munculnya *error* saat proses pembayaran, *link* pembayaran yang bermasalah, dan metode pembayaran yang dianggap kurang memudahkan pengguna. Untuk saran perbaikan yang dapat dilakukan terkait masalah pembayaran, di antaranya yakni perlu adanya perbaikan *internal* terkait proses pembayaran via LinkAja, pengecekan sistem informasi pembayaran, dan perlu adanya pertimbangan untuk menambah opsi/metode pembayaran selain melalui LinkAja.



## BAB VI

### PENUTUP

#### 6.1 Kesimpulan

Berdasarkan tujuan dari penelitian terhadap aplikasi KAI Access, maka dapat ditarik kesimpulan sebagai berikut.

1. Dari hasil analisis ulasan, maka diperoleh delapan kata kunci permasalahan atau keluhan utama pengguna aplikasi KAI Access, yakni permasalahan “bad”, “email”, “error”, “fix”, “login”, “ticket”, “time”, dan “update”. Kemudian berdasarkan analisis *association rules*, untuk data ulasan berbahasa Inggris (EN) diperoleh *rules* sebanyak 98 buah, serta untuk gabungan ulasan berbahasa Indonesia, Melayu dan Jawa (ID – MS – JV) sebanyak 164 buah. Dari kumpulan *rules* tersebut, terdapat 45 *rules* yang berhubungan dengan kata kunci permasalahan.
2. Setelah meninjau *association rules*, untuk tiap kata kunci permasalahan, terdapat penyebab atau akar permasalahan yang sering muncul atau dikeluhkan pengguna. Beberapa di antaranya yakni adanya masalah yang berkaitan dengan pencarian, pemesanan hingga pembayaran tiket, kemudian terdapat keluhan terkait *server*, serta masalah terkait akun pengguna aplikasi. Untuk beberapa penyebab tersebut, solusi penyelesaian masalah yang dapat dilakukan yakni menggelar diskusi atau komunikasi antara pihak penyelenggara KAI Access dan pengguna aplikasi, baik secara langsung maupun melalui media komunikasi jarak jauh, sehingga kedua belah pihak dapat menyelaraskan informasi dan kemudian dapat mengetahui letak kesalahan dari permasalahan

tersebut. Selanjutnya, pengecekan hingga perbaikan *server* ataupun sistem informasi internal aplikasi dapat dilakukan sebagai tindak lanjut atas diskusi yang telah dilakukan.

## 6.2 Saran

Saran mengenai penelitian selanjutnya maupun pengembangan yang dapat dilakukan berdasarkan penelitian ini di antaranya melakukan *in-depth problem solving* dengan penerapan beberapa bidang kelimuan untuk tiap permasalahan sehingga solusi perbaikan aplikasi KAI Access dapat diberikan secara *detail*. Kemudian pengembangan penelitian lainnya yakni penggunaan ulasan-ulasan dengan *rating* 3 (tiga) yang cenderung memiliki sentimen positif, negatif ataupun netral. Dengan analisis ulasan berbasis *lexicon*, maka ulasan dengan *rating* 3 (tiga) dapat dimanfaatkan untuk menambah data untuk analisis permasalahan.

## DAFTAR PUSTAKA

- Abu-Shanab, E., & Harb, Y. (2019). E-Government Research Insights: Text Mining Analysis. *Electronic Commerce Research and Applications*.
- Anicas, M. (2014). *How To Troubleshoot Common HTTP Error Codes* (online): <https://www.digitalocean.com/community/tutorials/how-to-troubleshoot-common-http-error-codes> (diakses tanggal 21 Oktober 2020).
- Badan Pusat Statistik. (2020). *Jumlah Penumpang Kereta Api, 2006-2020 (Ribu Orang)* (online): <https://www.bps.go.id/linkTableDinamis/view/id/815> (diakses tanggal 21 Oktober 2020).
- Baenanda, L. (2019). *Mengenal Lebih Jauh Revolusi Industri 4.0* (online): <https://binus.ac.id/knowledge/2019/05/mengenal-lebih-jauh-revolusi-industri-4-0/> (diakses tanggal 1 April 2020).
- Bansal, S. (2020). *Supervised and Unsupervised learning* (online): <https://www.geeksforgeeks.org/supervised-unsupervised-learning/> (diakses tanggal 20 Oktober 2020).
- Bloomenthal, A. (2019). *Electronic Commerce (e-commerce)* (online): <https://www.investopedia.com/terms/e/ecommerce.asp> (diakses tanggal 1 April 2020).
- Budianto, A. (2019). *Pengguna Aplikasi KAI Access Capai 3,8 Juta Akun* (online): <https://ekbis.sindonews.com/berita/1466099/34/pengguna-aplikasi-kai-access-capai-38-juta-akun> (diakses tanggal 3 April 2020).
- Debora, Y. (2019). *Sejarah Revolusi Industri dari 1.0 hingga 4.0* (online): <https://tirto.id/sejarah-revolusi-industri-dari-10-hingga-40-dhhu> (diakses tanggal 1 April 2020).
- Desra. (2019). *7 Jenis e-Commerce yang Perlu Anda Ketahui* (online): <https://www.jurnal.id/id/blog/jenis-e-commerce-untuk-kembangkan-bisnis/> (diakses tanggal 3 April 2020).

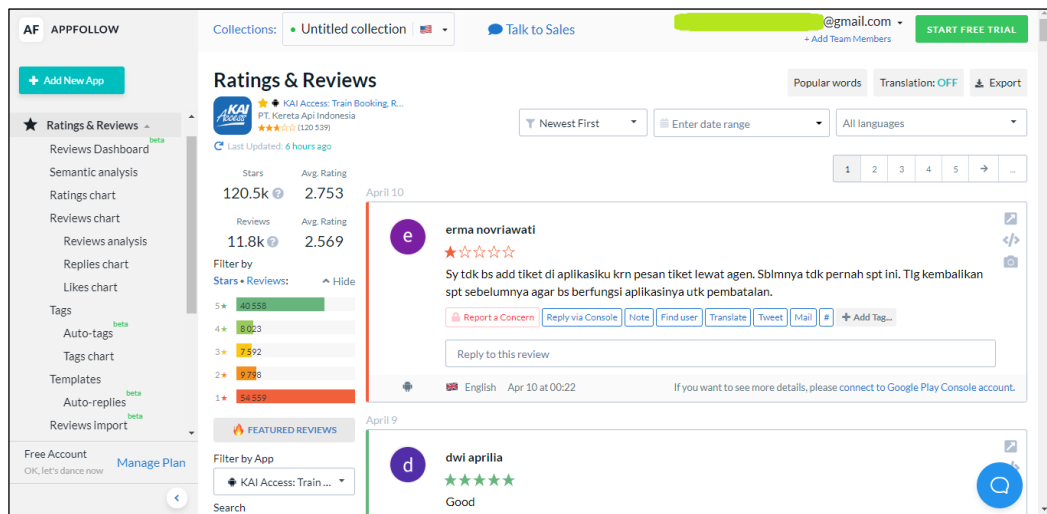
- Dewi, L. C., Meiliana, & Chandra, A. (2019). Social Media Web Scraping using Social Media Developers API and Regex. *4th International Conference on Computer Science and Computational Intelligence 2019 (ICCSICI)* (hal. 444-449). Jakarta: Procedia Computer Science.
- Drury, B., & Roche, M. (2019). A survey of the applications of text mining for agriculture. *Computers and Electronics in Agriculture*, 1-12.
- Fayyad, U., Piatetsky-shapiro, & Smyth. (1996). *From Data Mining to Knowledge Discovery: An Overview*. California: The MIT Press.
- Gokhberg, L., Kuzminov, I., Khabirova, E., & Thurner, T. (2019). Advanced text-mining for trend analysis of Russia's Extractive Industry. *Futures*.
- Gupta, S. (2018). *Sentiment Analysis: Concept, Analysis and Applications* (online): <https://towardsdatascience.com/sentiment-analysis-concept-analysis-and-applications-6c94d6f58c17> (diakses tanggal 1 April 2020).
- Hahsler, M. (2005). Introduction to arules – A computational environment for mining association rules and frequent item sets. *Journal of Statistical Software*, 1-3.
- Hajaj, Y (2020). *Introduction to Supervised, Semi-supervised, Unsupervised and Reinforcement Learning* (online): <https://www.baeldung.com/cs/machine-learning-intro> (diakses tanggal 13 Januari 2021).
- Halvey, M., & Keane, M. T. (2007). An Assessment of Tag Presentation Techniques. *World Wide Web Conference Committee*. Alberta: Wayback Machine.
- Han, J., Pei, J., & Kamber, M. (2011). *Data Mining: Concepts and Techniques*. Massachusetts: Morgan Kaufmann Publisher.
- Hill, N., Roche, G., & Allen, R. (2007). *Customer Satisfaction: The Customer Experience Through the Customer's Eyes*. Aldershot: Gower Publishing.
- Kemp, S., & Moey, S. (2019). *Digital 2019 Spotlight: E-Commerce in Indonesia* (online): <https://datareportal.com/reports/digital-2019-ecommerce-in-indonesia> (diakses tanggal 1 April 2020).
- Kiran, D. R. (2017). *Total Quality Management*. Andhra Pradesh: BSP Books Pvt.
- Kotler, P., Bowen, J., & Makens, J. (2013). *Marketing for hospitality and tourism*. London: Pearson Education.

- Kumar B, S., & Karthika. (2014). A Survey on Text Mining Process and Techniques. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Vol. 3 Issue 7*, 2279-2284.
- Li, J., Lowe, D., Wayment, L., & Huang, Q. (2020). Text mining datasets of  $\beta$ -hydroxybutyrate (BHB) supplement products' consumer online reviews. *Data in Brief*, 1-8.
- Martaon, A. T. (2019). *Pengguna KAI Access Bertambah 6.600 Orang per Hari* (online): <https://www.medcom.id/nasional/peristiwa/GNIYgOBb-pengguna-kai-access-bertambah-6-600-orang-per-hari> (diakses tanggal 3 April 2020).
- Moro, A., Joanny, G., & Moretti, C. (2020). Emerging technologies in the renewable energy sector: A comparison of expert review with a text mining software. *Futures*.
- Nayoan, A. (2020). *Apa itu Web Scraping? Pengertian, Teknik, dan Manfaatnya* (online): <https://www.niagahoster.co.id/blog/web-scraping/> (diakses tanggal 4 April 2020).
- Nisbet, R., Elder, J., & Miner, G. (2009). *Handbook of Statistical Analysis and Data Mining Applications*. London: Academic Press.
- Nugroho, K. S. (2019). *Dasar Text Preprocessing dengan Python* (online): <https://medium.com/@ksnugroho/dasar-text-preprocessing-dengan-python-a4fa52608ffe> (diakses tanggal 4 April 2020).
- Oliver, R. L. (1981). A cognitive model of the antecedents and consequences of satisfaction decisions. *Journal of Marketing Research, Vol. XVII*.
- Pejic-Bach, M., Bertoncel, T., Mesko, M., & Krstic, Z. (2019). Text mining of industry 4.0 job advertisement. *International Journal of Information Management*.
- PT. Kereta Api Indonesia (Persero) (2020). *Sekilas KAI - Company Profile* (online): [https://kai.id/corporate/about\\_kai/](https://kai.id/corporate/about_kai/) (diakses tanggal 13 Januari 2021).
- Russell, J. P. (2012). *The ASQ Auditing Handbook, Fourth Edition*. Milwaukee: ASQ Quality Press.

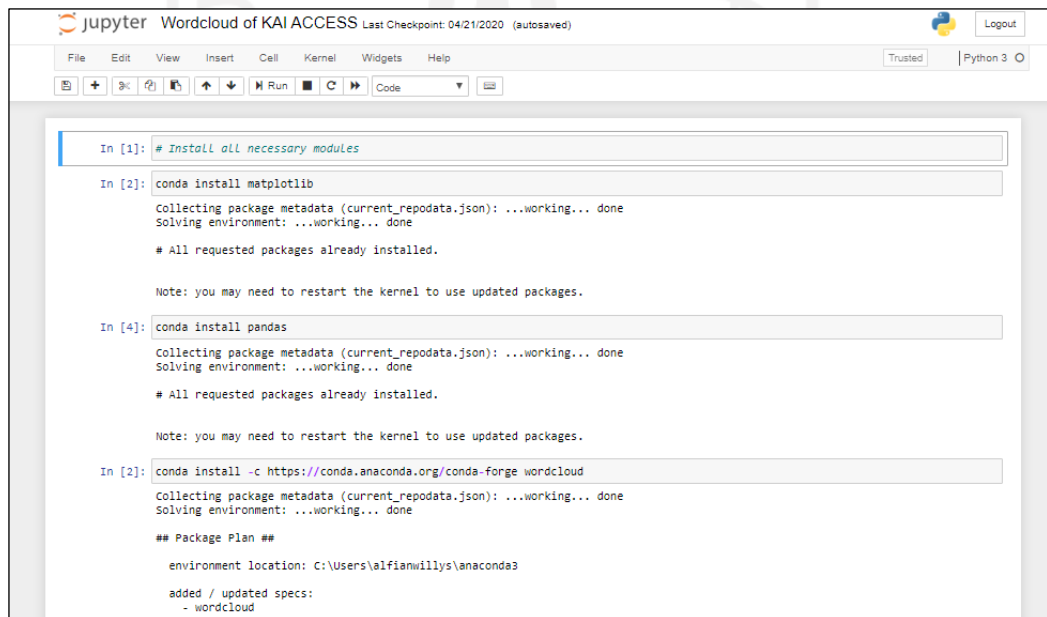
- SAS. (2020). *Data Mining: What is it and why it matters* (online):  
[https://www.sas.com/en\\_id/insights/analytics/data-mining.html#](https://www.sas.com/en_id/insights/analytics/data-mining.html#) (diakses tanggal 4 April 2020).
- Sethi, A. (2020). *Supervised Learning vs. Unsupervised Learning – A Quick Guide for Beginners* (online):  
<https://www.analyticsvidhya.com/blog/2020/04/supervised-learning-unsupervised-learning/> (diakses tanggal 20 Oktober 2020).
- Tague, N. R. (2005). *The Quality Toolbox, Second Edition* (online):  
<https://asq.org/quality-resources/quality-assurance-vs-control> (diakses tanggal 2 April 2020).
- Villeneuve, H., & O'Brien, W. (2019). Listen to the guests: Text-mining Airbnb reviews to explore indoor environmental quality. *Building and Environment*, 6-8.
- Wu, C., Kuo, C., Su, C., Wang, S., & Dai, H. (2019). Using text mining to extract depressive symptoms and to validate the diagnosis of major depressive disorder from electronic health records. *Journal of Affective Disorders*.
- Yildiz, T. (2019). Examining The Concept of Industry 4.0 Studies Using Text Mining and Scientific Mapping Method. *3rd World Conference on Technology, Innovation and Entrepreneurship (WOCTINE)*.
- Zhao, Y. (2013). *R and Data Mining*. San Diego: Elsevier Inc.

# LAMPIRAN

## Lampiran 1. Tahapan Penelitian

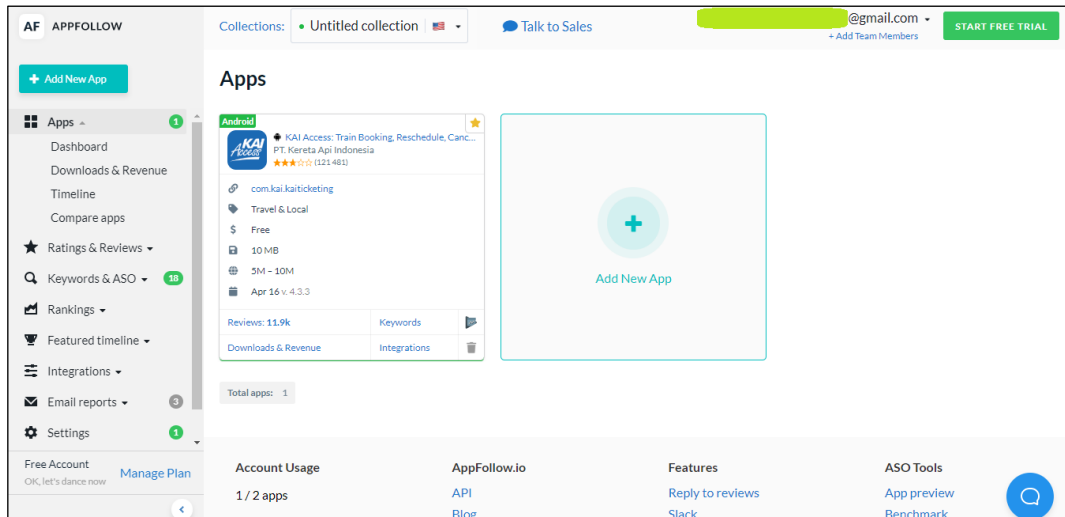


Gambar A.1 Interface aplikasi AppFollow

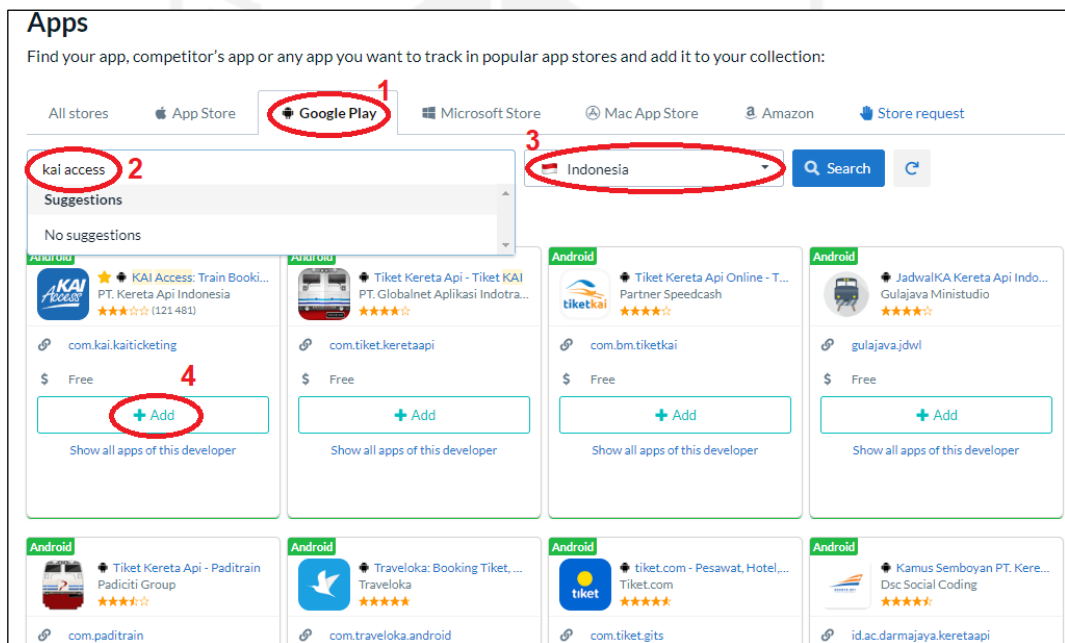


Gambar A.2 Interface Jupyter Notebook

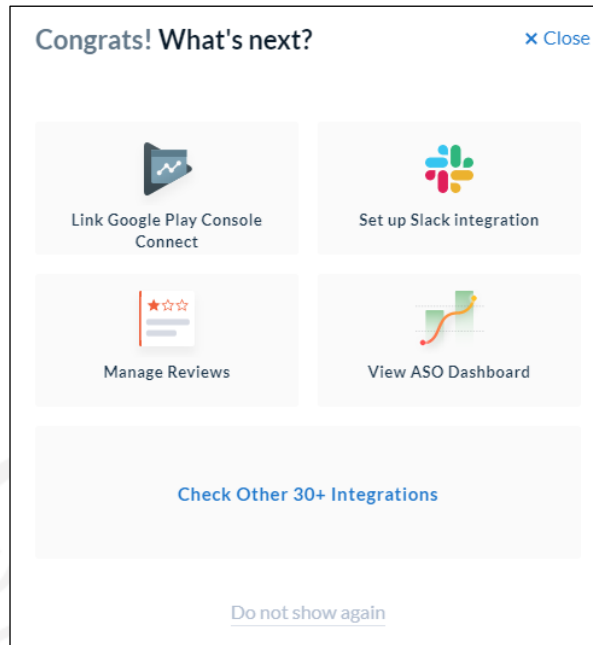




Gambar A.3 Home interface pada AppFollow



Gambar A.4 Langkah pencarian aplikasi KAI Access



Gambar A.5 Pilihan "Manage Reviews" pada *pop-up*

**Ratings & Reviews** Popular words Translation: OFF Export

**KAI Access** ★ **KAI Access: Train Booking R...**  
 PT. Kereta Api Indonesia  
 ★★★★★ (120.539)

Last Updated: 6 hours ago

Stars: 120.5k ⓘ Avg. Rating: 2.753

Reviews: 11.8k ⓘ Avg. Rating: 2.569

Filter by  
 Stars **Reviews** ⌵ Hide

Stars	Count
5★	2.893
4★	1.006
3★	1.215
2★	1.456
1★	5.209

**FEATURED REVIEWS**

Filter by App  
 KAI Access: Train ...

Search

**April 10**

**e** **erma novriawati**  
 ★★★★★  
 Sy tdk bs add tiket di aplikasiku krn pesan tiket lewat agen. Sblmnya tdk pernah spt ini. Tlg kembalikan spt sebelumnya agar bs berfungsi aplikasinya utk pembatalan.

[Report a Concern](#) [Reply via Console](#) [Note](#) [Find user](#) [Translate](#) [Tweet](#) [Mail](#) <#> [+ Add Tag...](#)

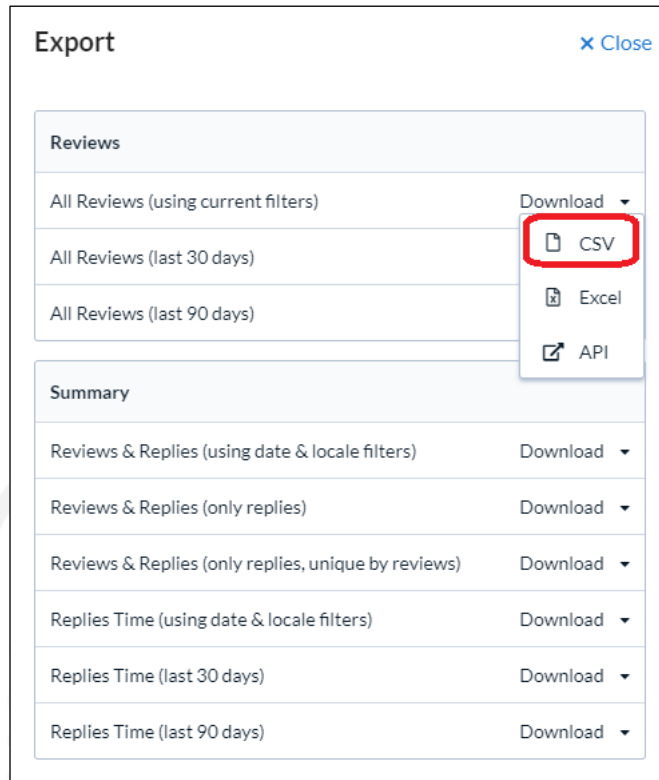
Reply to this review

English Apr 10 at 00:22 If you want to see more details, please connect to Google Play Console account.

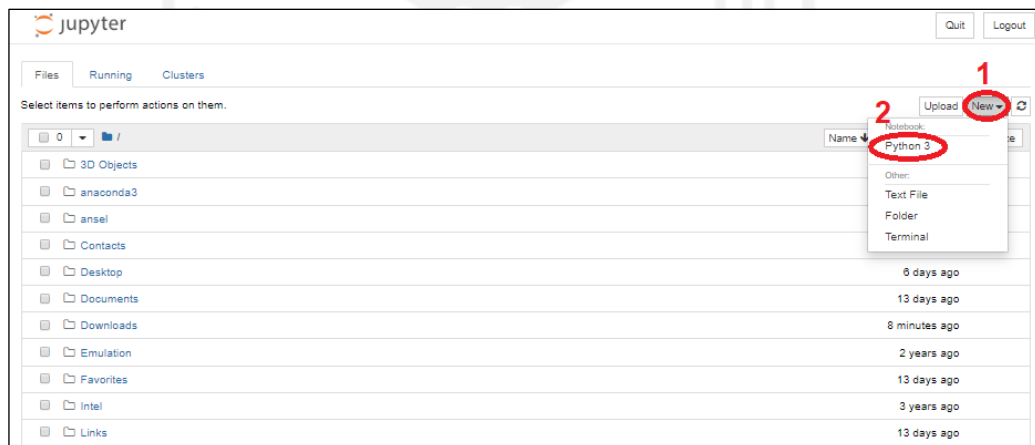
**April 9**

**d** **dwi aprilia**  
 ★★★★★  
 Good

Gambar A.6 Opsi "Filter by Reviews"



Gambar A.7 Download file as csv pada AppFollow



Gambar A.8 Membuka lembar kerja baru pada Jupyter

```

In [1]: import pandas as pd

In [2]: df = pd.read_csv('C:\COLLEGE\College\8th Semester\TA\KAIA_Review_Strc_Cmpl.csv', usecols = ['Review'], squeeze = True)

In [3]: df
Out[3]: 0      Sy tdk bs add tiket di aplikasiku krn pesan ti...
        1      Good
        2      Tidak bisa melakukan pembatalan jika pemesanan...
        3      Rubbish app. I can't even log in. How dare KAI...
        4      Aplikasinya bagus. Tapi pembayaran hanya bisa ...
        ...
        11769     Bagus sekali untuk pemesanan tiket kereta api
        11770     Senang sekali akhirnya ada aplikasi resmi dari...
        11771     Aplikasi keren nih bisa pesen tiket dmn aja
        11772     Aplikasi yg sangat bagus. Membantu untuk memes...
        11773     Good Job Sir
        Name: Review, Length: 11774, dtype: object

```

Gambar A.9 Contoh *data inputting* menggunakan pandas

```

In [4]: sentence = ''

In [7]: for val in df:
        val = str(val)
        tokens = val.split()
        for i in range(len(tokens)):
            tokens[i] = tokens[i].lower()
        sentence += ' '.join(tokens)+' '

In [ ]:

```

Gambar A.10 Proses *tokenization* dan *case folding*

```

In [3]: # Apply stopwords
        stop_word = set(stopwords.words('indonesian'))

In [11]: # Add customized stopwords to WordCloud
        custom_stopword = ['aplikasi', 'aplikasinya', 'app', 'ga', 'gak', 'ya', 'nya', 'udah', 'kai', 'kai access',
                            'saja', 'aja', 'pt kai', 'ini', 'kali', 'tolong', 'banget', 'muncul', 'access', 'lagi', 'bisa',
                            'nih', 'sama', 'gk', 'gabisa', 'sih', 'pt', 'saya', 'klo', 'ada', 'ngga'] + list(STOPWORDS)

```

Gambar A.11 Penerapan *stop word removal*

```

In [6]: # Remove any Emojis
        import unicodedata
        from unidecode import unidecode
        def deEmojify(inputString):
            return inputString.encode('ascii', 'ignore').decode('ascii')

In [7]: no_emoji = deEmojify(sentence)

```

Gambar A.12 Tahap eliminasi emoji menggunakan modul “unicodedata”



## Lampiran 2. *Source Code* untuk Proses Pengolahan Data

### Read csv data

```
import pandas as pd

df = pd.read_csv('C:\COLLEGE\College\8th
                Semester\TA\KAIA_Review_Strc_Cmpl.csv', usecols=['Rating',
                'Review'])
```

### Text preprocessing

```
import unicodedata
import nltk
from unidecode import unidecode
from nltk.corpus import stopwords

stop_word_id = set(stopwords.words('indonesian'))
```

### Tokenize and remove stopwords

```
sentence = ""
for val in df.Review:
    val = str(val)
    kata1 = val.split()
    for i in range (len(kata1)):
        kata1[i] = kata1[i].lower()
    sentence += ' '.join([word for word in kata1 if word not in stop_word_id])+'
```

### Remove Emojis

```
def deEmojify(inputString):
    return inputString.encode('ascii', 'ignore').decode('ascii')
no_emoji = deEmojify(sentence)
```

**Append clean review to dataframe**

```

newSentence = []
temp = no_emoji.split('|')
for each in temp:
    newSentence.append([each])

clean_text = pd.DataFrame(newSentence, columns=['Clean_Review'])

```

```

new_df = pd.concat([df,clean_text], axis=1)

```

**Review labeling based on rating**

```

new_df["bad_review"] = new_df["Rating"].apply(lambda x: 1 if x < 3 else 0)
new_df["bad_review"].value_counts(normalize=False)

```

**Drop missing values**

```

new_df.dropna(inplace=True)

```

**Remove reviews labeled '0'**

```

new_df1 = new_df[new_df['bad_review'].isin(['0']) == False]

```

**Check data frequency based on language**

```

import langid
from collections import Counter
import matplotlib.pyplot as plt

new_df2 = new_df1.reset_index(drop=True)
pred_lang = [langid.classify(review) for review in new_df1['Review']]
df_lang = pd.DataFrame(pred_lang, columns=['Lang', 'Value'])

new_df3 = pd.concat([new_df2,df_lang], axis=1)

new_df4 = new_df3.drop(columns=['Review', 'bad_review', 'Value'])

```

```
lang_freq = Counter(df_lang['Lang'])
lang_freq = lang_freq.most_common(10)
```

```
freq_df = pd.DataFrame(lang_freq, columns=['Language', 'Freq'])
```

### **Separate data between EN and ID-MS-JV language**

#### **# EN Reviews**

```
df_en = new_df4[new_df4['Lang'].isin(['en']) == True]
```

#### **# ID-MS-JV Reviews**

```
df_id = new_df4[new_df4['Lang'].isin(['id', 'ms', 'jv']) == True]
```

### **Apply wordcloud and customized stopwords**

#### **# EN Reviews**

```
from wordcloud import WordCloud, STOPWORDS
```

```
stopword_en = set(stopwords.words('english'))
```

```
custom_stopword_en = ['aplikasi', 'aplikasinya', 'app', 'app.', 'apps', ',', '!', '!',
                       'application', 'kai', 'kai access', 'really', 'pt kai',
                       'access', 'pt', 'always', 'even', 'please', 'make',
                       'cant', 'this', 'want', 'need', 'still', 'last', 'new'] +
                       list(stopword_en) + list(STOPWORDS)
```

```
sentence_en = ""
```

```
for val in df_en.Clean_Review:
```

```
    val = str(val)
```

```
    wrds = val.split()
```

```
    sentence_en += ''.join([word for word in wrds if word not in
                             custom_stopword_en])+'
```



```

wordcloud_en = WordCloud(width=800, height=800,
                          background_color='white',
                          stopwords=custom_stopword_en,
                          min_font_size=10, font_path='arial').generate(sentence_en)

plt.figure(figsize=(10,10), facecolor=None)
plt.imshow(wordcloud_en)
plt.axis('off')
plt.tight_layout(pad=0)

# ID, MS, and JV reviews
custom_stopword_id = ['aplikasi', 'aplikasinya', 'app', 'ga', 'gak', 'ya', 'nya', 'udah',
                      'kai', 'kai access', 'saja', 'aja', 'pt kai', 'ini', 'kali', 'tolong',
                      'banget', 'muncul', 'access', 'lagi', 'bisa', 'nih', 'sama', 'gk',
                      'gabisa', 'sih', 'pt', 'saya', 'klo', 'ada', 'ngga', 'please',
                      'terima', 'kasih', 'terima kasih', 'mohon', 'bisa', 'pakai']
+ list(STOPWORDS)

sentence_id = ""
for val in df_id.Clean_Review:
    val = str(val)
    kata2 = val.split()
    sentence_id += ''.join([word for word in wrds if word not in
                           custom_stopword_id])+' '

wordcloud_id = WordCloud(width=800, height=800,
                          background_color='white',
                          stopwords=custom_stopword_id,
                          min_font_size=10, font_path='arial').generate(sentence_id)

```

```
plt.figure(figsize=(10,10), facecolor=None)
plt.imshow(wordcloud_en)
plt.axis('off')
plt.tight_layout(pad=0)
```

### **Check the word frequencies**

#### **# EN reviews**

```
en_words = sentence_en.split()
Counter(en_words).most_common(10)
```

#### **# ID reviews**

```
id_words = sentence_id.split()
Counter(id_words).most_common(10)
```

### **Check the association rules**

#### **# EN reviews**

```
import numpy as np
from mlxtend.frequent_patterns import apriori, association_rules

apriori_en = "
for val in df_en.Clean_Review:
    val = str(val)
    kotoba1 = val.split()
    apriori_en += ''.join([word for word in kotoba1 if word not in
                           custom_stopword_en])+'
```

```
bun1 = []
header = []
kotoba1_spl = apriori_en.split('|')
for each in kotoba1_spl:
    newWord = []
    kotoba1_splHeaderIndex = 0
```

```

kotoba1_splHeader = []
eachWord = each.split()
for word in eachWord:
    newWord.append(".join(e for e in word if e.isalnum()))
    kotoba1_splHeader.append('header' + str(kotoba1_splHeaderIndex))
    kotoba1_splHeaderIndex += 1
if(len(header) < len(kotoba1_splHeader)):
    header = kotoba1_splHeader
bun1.append(newWord)

table_en = pd.DataFrame(bun1)

pd.DataFrame(bun1).to_csv('Split Bun EN.csv', header=header, index=None)

words_table_en = pd.read_csv('Split Bun EN.csv')
uniqueWordEN = words_table_en.header0.unique()
pd.DataFrame(uniqueWordEN)

encoded_vals_en = []
for index, row in table_en.iterrows():
    labels = {}
    uncommons = list(set(uniqueWordEN) - set(row))
    commons = list(set(uniqueWordEN).intersection(row))
    for uc in uncommons:
        labels[uc] = 0
    for com in commons:
        labels[com] = 1
    encoded_vals_en.append(labels)
encoded_vals_en[0]
ohe_df_en = pd.DataFrame(encoded_vals_en)

```

```

freq_items_en = apriori(ohe_df_en, min_support=0.01, use_colnames=True)
freq_items_en.head(10)

rules_en = association_rules(freq_items_en, metric="lift", min_threshold=1)

ar_lift_en = rules_en.sort_values(by='lift', ascending=False)

ar_lift_en1 = ar_lift_en.drop(columns=['antecedent support', 'consequent support',
                                     'leverage', 'conviction'])

with pd.option_context('display.max_rows', 12000):
    print(ar_lift_en1)

# ID, MS, and JV reviews
apriori_id = ""
for val in df_id.Clean_Review:
    val = str(val)
    kotoba2 = val.split()
    apriori_id += ''.join([word for word in kotoba2 if word not in
                           custom_stopword_id])+'|'

bun2 = []
header = []
kotoba2_spl = apriori_id.split("|")
for each in kotoba2_spl:
    newWord = []
    kotoba2_splHeaderIndex = 0
    kotoba2_splHeader = []
    eachWord = each.split()
    for word in eachWord:
        newWord.append(''.join(e for e in word if e.isalnum()))
        kotoba2_splHeader.append('header' + str(kotoba2_splHeaderIndex))

```

```

    kotoba2_splHeaderIndex += 1
    if(len(header) < len(kotoba2_splHeader)):
        header = kotoba2_splHeader
    bun2.append(newWord)

table_id = pd.DataFrame(bun2)

pd.DataFrame(bun2).to_csv('Split Bun ID.csv', header=header, index=None)

words_table_id = pd.read_csv('Split Bun ID.csv')
uniqueWordID = words_table_id.header0.unique()
pd.DataFrame(uniqueWordID)

encoded_vals_id = []
for index, row in table_id.iterrows():
    labels = {}
    uncommons = list(set(uniqueWordID) - set(row))
    commons = list(set(uniqueWordID).intersection(row))
    for uc in uncommons:
        labels[uc] = 0
    for com in commons:
        labels[com] = 1
    encoded_vals_id.append(labels)
encoded_vals_id[0]
ohe_df_id = pd.DataFrame(encoded_vals_id)

freq_items_id = apriori(ohe_df_id, min_support=0.01, use_colnames=True)
freq_items_id.head(10)

rules_id = association_rules(freq_items_id, metric="lift", min_threshold=1)

ar_lift_id = rules_id.sort_values(by='lift', ascending=False)

```

```
ar_lift_id1 = ar_lift_id.drop(columns=['antecedent support', 'consequent support',  
                                     'leverage', 'conviction'])
```

```
with pd.option_context('display.max_rows', 12000):  
    print(ar_lift_id1)
```

