KLASIFIKASI SENTIMEN, TOPIK, DAN DETAIL TOPIK DARI ULASAN KAI ACCESS MENGGUNAKAN MULTILAYER PERCEPTRON (MLP) DAN BIDIRECTIONAL LONG SHORT TERM MEMORY (BiLSTM)

Disusun Oleh:

N a m a : Nabiilah Nuur Ainii
NIM : 18523252

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
2022

HALAMAN PENGESAHAN DOSEN PEMBIMBING

KLASIFIKASI SENTIMEN, TOPIK, DAN DETAIL TOPIK DARI ULASAN KAI ACCESS MENGGUNAKAN MULTILAYER PERCEPTRON (MLP) DAN BIDIRECTIONAL LONG SHORT TERM MEMORY (BiLSTM)

TUGAS AKHIR JALUR MAGANG

Disusun Oleh:

N a m a	: Nabiilah Nuur Ainii
NIM	: 18523252

Yogyakarta, 22 Agustus 2022

Pembimbing,

(Arrie Kurniawardhani, S.Si., M.Kom.)

HALAMAN PENGESAHAN DOSEN PENGUJI

KLASIFIKASI SENTIMEN, TOPIK, DAN DETAIL TOPIK DARI ULASAN KAI ACCESS MENGGUNAKAN MULTILAYER PERCEPTRON (MLP) DAN BIDIRECTIONAL LONG SHORT TERM MEMORY (BiLSTM)

TUGAS AKHIR JALUR MAGANG

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 22 Agustus 2022

Tim Penguji

Arrie Kurniawardhani, S.Si., M.Kom. _____

Anggota 1

Irving Vitra Paputungan, S.T., M.Sc., Ph.D. _____

Anggota 2

Dr. Ahmad Luthfi, S.Kom., M.Kom. _____

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia

(Dr. Raden Teduh Dirgahayu, S.T., M.Sc.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Nabiilah Nuur Ainii

NIM : 18523252

Tugas akhir dengan judul:

KLASIFIKASI SENTIMEN, TOPIK, DAN DETAIL TOPIK DARI ULASAN KAI ACCESS MENGGUNAKAN MULTILAYER PERCEPTRON (MLP) DAN BIDIRECTIONAL LONG SHORT TERM MEMORY (BiLSTM)

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 22 Agustus 2022

(Nabiilah Nuur Ainii)

HALAMAN PERSEMBAHAN

Alhamdulillah saya panjatkan kepada Allah SWT karena dengan segala nikmat dan kesempatan yang telah diberikan kepada saya untuk menyelesaikan tugas akhir saya ini yang berjudul "Klasifikasi Sentimen, Topik, dan Detail Topik dari Ulasan KAI Access menggunakan Multilayer Perceptron (MLP) dan Bidireectional Long Short Term Memory (BiLSTM)" dengan segala kekurangannya. Segala puji bagi Allah SWT juga yang telah memberikan saya kesempatan untuk mendapatkan kehadiran serta dukungan dari orang-orang disekitar saya yang senantiasa menemani, memberikan semangat, dan do'a sehingga tugas akhir saya ini dapat diselesaikan dengan baik dan tepat waktu. Tugas akhir ini saya persembahkan untuk:

1. Kedua orang tua saya, Bapak Agung Susilo Putro dan Ibu Umiyati.
2. Keluarga tersayang lainnya yang turut mendukung saya selama ini.
3. Ibu Arrie Kurniawardhani, S.Si., M.Kom. selaku dosen pembimbing.
4. Program Studi Informatika FTI UII yang telah mewadahi saya selama 4 tahun terakhir dalam menuntut ilmu dan mengembangkan diri.
5. Rekan seperjuangan saya selama kuliah, teman-teman INSIGHT khususnya teman-teman terdekat saya di kelas E pada semester awal yang telah memberikan bantuan, dukungan, menemani, serta mewarnai masa perkuliahan saya empat tahun kebelakang ini.
6. Bangtan Seonyondan atau secara individu kepada Kim Namjoon, Kim Seokjin, Min Yoongi, Jung Hoseok, Park Jimin, Kim Taehyung, dan Jeon Jungkook yang sudah banyak memberikan dukungan yang sangat besar secara tidak langsung melalui lagu-lagunya.

Terima kasih banyak.

HALAMAN MOTO

"Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya"

– Al Baqarah ayat 286

"*Maybe I made a mistake yesterday, but yesterday's me is still me. Today I am who I am with all of my faults and mistakes. Tomorrow I might be a tiny bit wiser and that will be me too. These faults and mistakes are what I am, making up the brightest stars in the constellation of my life*."

– Kim Namjoon

KATA PENGANTAR

Assalamualaikum Warahmatullahi Wabarakatuh,

Puji dan syukur penulis haturkan kepada Allah SWT yang telah melimpahkan rahmat, taufiq, serta hidayah-Nya sehingga penulis dapat menyelesaikan tugas akhir jalur magang ini dengan lancar dan tepat waktu. Tidak lupa *shalawat* serta salam penulis junjungkan kepada Nabi Muhammad SWT yang telah membimbing dan menuntun umat muslim kepada jalan yang diridhai oleh Allah SWT dan menjadi rahmat bagi seluruh alam.

Tugas ini disusun sebagai bukti pelaksanaan kegiatan magang dan menjadi salah satu syarat kelulusan mahasiswa jalur akhir magang Program Studi Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia tahun 2022.

Adapun beberapa hambatan dan tantangan selama mengikuti aktivitas magang adalah penyesuaian kegiatan kerja dan adaptasi pada lingkungan tempat kerja, pembagian waktu dengan kegiatan lain diluar aktivitas magang, serta rasa malas dan bosan yang terkadang datang saat pengerjaan tugas magang yang bersifat repetitif yang dilakukan saat di rumah (*work from home*). Meskipun penulis mendapatkan kesulitan tersebut, penulis tidak berhenti untuk tetap menyelesaikan tugas tugas dan kewajibannya tersebut sampai kegiatan magang berakhir.

Selama pelaksanaan magang, penulisan laporan tengah, sampai penyelesaian tugas akhir ini penulis banyak dibantu, dibimbing, dan didukung oleh banyak pihak sehingga penulis merasa lebih ringan dalam pengerjaan tugas-tugas tersebut. Oleh karena itu, penulis ingin mengucapkan banyak terima kasih kepada:

1. Allah SWT atas segala rahmat dan hidayah-Nya yang hadir dalam segala langkah penulis selama hidup sebagai kekuatan salah satunya penyemangat untuk menuntut ilmu.
2. Orang tua dan keluarga yang senantiasa berdoa untuk kelancaran dan selalu memberikan semangat serta dukungannya setiap hari.
3. Bapak Raden Teduh Dirgahayu, S.T., M.Sc., selaku Ketua Program Studi Informatika Program Sarjana Fakultas Teknologi Industri Universitas Islam Indonesia.
4. Ibu Arrie Kurniawardhani, S.Si., M.Kom., selaku dosen pembimbing yang selalu membimbing penulis selama setahun kebelakang dalam kegiatan penjaluran magang hingga penyelesaian tugas akhir ini.
5. Bapak dan Ibu dosen Program Studi Informatika, yang telah memberikan ilmu yang luar biasa bermanfaat selama saya menempuh studi.

6. Bapak Dody Gunawan, S.T. selaku Vice President Divisi Enterprise Resource Planning (ERP) dan Ibu Salma Isra Lionara, S.T. selaku mentor yang telah membimbing penulis selama kegiatan magang di PT Kereta Api Indonesia (Persero).
7. Teman-teman terdekat saya ‘UwU’ dan Roosyidah Ulya yang selalu menjadi *support system* selama saya berkuliah.
8. Rekan-rekan seperjuangan, INSIGHT, yang telah memberikan bantuan, dukungan , dan berbagi selama pelaksanaan magang hingga pengerjaan tugas akhir.
9. Teman – teman dekat kuliah saya yang telah menemani saya sedari awal, teman – teman grup Pisang Goreng, rekan terdekat yang selalu mendukung dan menemani saya baik semasa perkuliahan hingga ujian pendadaran.
10. Rekan-rekan terdekat di himpunan, kepanitiaan, maupun asisten laboratorium yang senantiasa membuat 4 tahun masa kuliah saya begitu menyenangkan.
11. Pihak-pihak lainnya yang turut mendukung dan berkontribusi dalam penyusunan tugas akhir ini yang tidak dapat disebutkan satu persatu.

Hasil tugas ini masih jauh dari kata sempurna, namun penulis berharap hasil tugas ini dapat bermanfaat bagi siapapun yang membaca dikemudian hari.

Yogyakarta, 22 Agustus 2022

(Nabiilah Nuur Ainii)

SARI

Penulis melakukan kegiatan magang sebagai Data Scientist di PT Kereta Api Indonesia (Persero) atau sering disebut PT KAI, salah satu Badan Usaha Milik Negara (BUMN) yang bergerak di bidang transportasi yaitu kereta api. Beberapa segmen usaha yang dilakukan KAI adalah angkutan penumpang, angkutan barang, pendukung angkutan kereta api, pendapatan non angkutan, dan subsidi pemerintah. Kantor pusat PT KAI bertempat di Bandung. Sebagai Data Scientist Intern di PT KAI, penulis ditugaskan untuk mengerjakan dua proyek yang berkaitan dengan pengolahan data dengan masing-masing proyek menggunakan data yang berbeda.

Proyek KAI Access, proyek yang menggunakan data ulasan aplikasi KAI Access (aplikasi penjualan tiket kereta api daring miliki PT KAI), dipilih untuk dikaji lebih lanjut dalam laporan tugas akhir ini. Hasil akhir dari proyek ini adalah *prototype* yang sudah menggunakan model Machine Learning sehingga dapat mengklasifikasikan data ulasan KAI Access secara otomatis ke dalam beberapa label yaitu sentimen, topik, dan detail topik. *Prototype* klasifikasi ini dapat mengklasifikasikan data ulasan terkait KAI Access yang ditulis oleh pengguna secara langsung dalam jumlah besar. Selain itu, *dashboard* visualisasi hasil analisis data ulasan KAI Access juga dibuat sebagai hasil akhir dari proyek ini. Pada pembuatan model Machine Learning, proyek ini menggunakan Natural Language Processing (NLP) dengan metode Deep Learning. Klasifikasi *multi-label multi-class* ini dibuat dengan tiga model yang berbeda sesuai dengan label yang diklasifikasikan. Penyusunan model dilakukan menggunakan algoritma Multi-Layer Perceptron (MLP) dan Bidirectional Long Short Term Memory (BiLSTM). Proyek berhasil diselesaikan sesuai dengan produk hasil akhir yang diminta yaitu *dashboard* klasifikasi yang menggunakan model Machine Learning dengan akurasi masing – masing model sentimen, topik, dan detail topik sebesar 87,35%, 79,10%, dan 64,85% serta visualisasi *dashboard* yang lengkap dan mudah dipahami.

Banyak pengalaman yang didapatkan selama enam bulan menjalankan kegiatan magang di PT KAI, banyak hal-hal yang baru dilakukan pertama kali oleh penulis dari kegiatan magang ini. Pengalaman pertama bekerja di kantor, berkoordinasi dengan pegawai kantor, beradaptasi dengan lingkungan kerja di kantor, menggunakan *dataset* internal perusahaan, juga mempelajari *tools* baru saat pengerjaan proyek. Dari pengalaman-pengalaman tersebut, penulis bisa mengembangkan *soft skill* dan juga *hard skill* sehingga dapat menjadi lebih baik lagi.

Kata kunci: BiLSTM, KAI Access, Klasifikasi Multi-Label Multi-Class, MLP, Ulasan.

GLOSARIUM

Algoritma	Sebuah proses atau kumpulan aturan yang diikuti untuk perintah tertentu.
Akurasi	Ukuran sejauh mana data sesuai dengan wujud aktual.
Dataset	Kumpulan (set) data.
Training Data	Data yang dipakai sebagai bahan pelatihan algoritma untuk mempelajari pola dalam data.
Testing Data	Data yang dipakai sebagai data penguji yang akan diprediksi oleh model algoritma sehingga dapat diketahui performanya.
Epoch	Hiperparameter jumlah perulangan algoritma melakukan pengolahan *training data* pada proses pelatihan.
Hidden State	Masukan untuk langkah.
Hidden Layer	lapisan yang ada di antara lapisan *input* dan lapisan *output.*
Klasifikasi	Proses menyusun sebuah objek ke dalam kelompok atau golongan tertentu menurut standar atau ketetapan yang sudah ditentukan sebelumnya.
Layer	Komponen tingkatan pada model untuk menampung data mentah.

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I
PENDAHULUAN

1.1 Latar Belakang

PT Kereta Api Indonesia (PT KAI atau 'Perseroan') adalah salah satu Badan Usaha Milik Negara (BUMN) yang menyediakan, mengatur, dan mengurus jasa angkutan kereta api di Indonesia (KAI Company Profile, 2021). Namun, seiring dengan dinamika dunia usaha dan berkembangnya tuntutan pasar, saat ini PT KAI juga menyelenggarakan kegiatan usaha penunjang lainnya dengan memanfaatkan sumber daya yang dimilikinya.

Beberapa segmen usaha yang dilakukan KAI adalah angkutan penumpang, angkutan barang, pendukung angkutan kereta api, dan pendapatan non angkutan. Untuk menjalankan kelima segmen usaha tersebut, KAI memiliki beberapa anak perusahaan dan entitas asosiasi yaitu PT Reska Multi Usaha yang berfokus pada perusahaan induk khususnya usaha restoran kereta api, PT Railink yang juga anak perusahaan PT Angkasa Pura II dengan layanan kereta bandara, PT Kereta Commuter Indonesia yang mengelola KA Commuter Jabodetabek, PT Kereta Api Pariwisata yang menyediakan layanan kereta api pariwisata di Indonesia, PT Kereta Api Logistik yang khusus melayani distribusi logistik berbasis kereta api, dan PT Kereta Api Properti Manajemen dengan usaha inti di bidang konstruksi, properti, perdagangan serta perawatan infrastruktur perkeretaapian.

Divisi IT *Enterprise Resource Planning* (ERP) di bawah bidang *Strategic Planning & Business Development* PT KAI menangani proyek terkait data digital yang dimiliki oleh perusahaan (*Big Data*) serta bekerjasama dengan pihak ketiga untuk kepentingan data. Mulai dari pengambilan data, pengolahan, hingga analisisnya. Divisi ini juga aktif bekerja sama dengan divisi lain guna meningkatkan pelayanan perusahaan melalui data. Salah satu proyek besarnya adalah mengolah data dari aplikasi resmi PT KAI yaitu aplikasi KAI Access. Bidang ini mengolah data transaksi, data pengguna, dan data yang terdapat pada aplikasi tersebut.

Penulis magang pada divisi IT ERP turut serta dalam proyek dalam lingkup aplikasi KAI Access yaitu mengolah data ulasan aplikasi KAI Access di Play Store. Selain data ulasan KAI Access, penulis juga mengolah data lainnya yaitu data akun Twitter KAI121, akun Twitter resmi PT KAI. Jika sebelumnya tim pada divisi ERP hanya menggunakan data tersebut untuk keperluan analisis untuk mendukung *operational decision-making*, pada kegiatan magang ini penulis ditugaskan untuk proyek yang belum pernah dikerjakan oleh divisi ERP sebelumnya

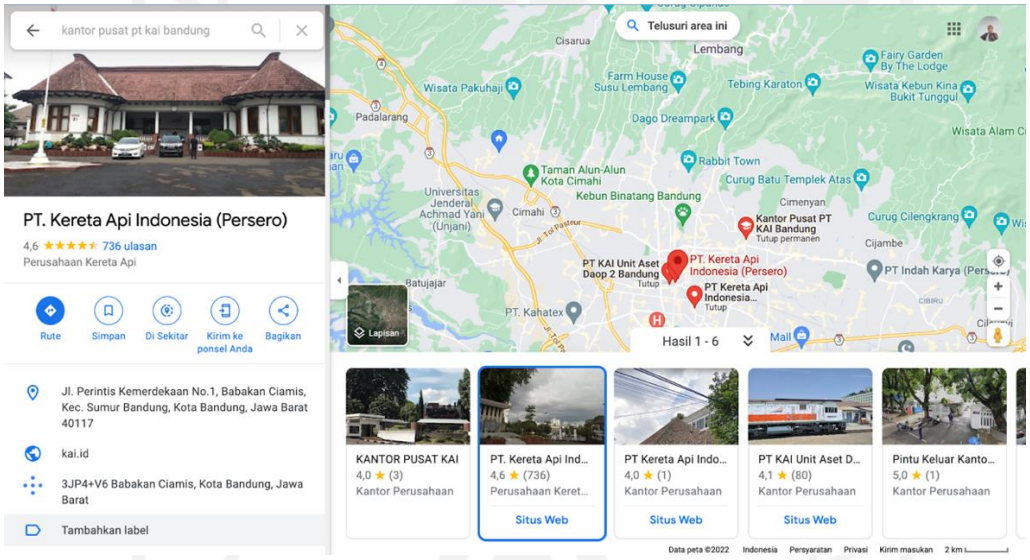
yaitu membuat model Machine Learning yang dapat mengklasifikasikan data-data tersebut. Selain itu, penulis juga ditugaskan untuk membuat analisis dan visualisasinya berdasarkan data KAI Access dan Twitter KAI121.

KAI Access dirilis sejak tahun 2014 dengan fitur yang terus berkembang hingga sekarang, dimana saat ini pemanfaatan teknologi pada pelayanan publik sudah menjadi hal yang umum. Hal ini menjadikan KAI Access sebagai layanan teknologi digital utama dalam pelayanan PT Kereta Api Indonesia. Ulasan terhadap sebuah aplikasi oleh pengguna sangat dibutuhkan karena dikenali sebagai sumber yang bernilai untuk memperbaiki maupun mengembangkan aplikasi, peningkatan nilai untuk pengguna, serta membantu pengembang aplikasi untuk lebih memahami pengguna (Abelein, Sharp, & Peach, 2013). Banyaknya jumlah ulasan yang diberikan oleh pengguna membuat pengembang aplikasi kesulitan dalam memilah dan mengkategorikan ulasan yang diberikan. Jika pengembang dapat mengidentifikasi dan mengklasifikasikan umpan balik dari pengguna secara cepat dan otomatis, maka pengembang dapat meningkatkan kualitas aplikasi dengan cepat dan meningkatkan kepuasan pengguna. Oleh karena itu, dibuatlah *protoype* yang dapat mengklasifikasikan data ulasan KAI Access dalam jumlah besar secara otomatis dan cepat. Data ulasan akan diklasifikasikan pada beberapa label yaitu sentimen, topik, dan detail topik yang setiap labelnya memiliki beberapa kelas untuk menjadi target klasifikasi data. Banyaknya label dan kelas di dalamnya dibuat untuk mendapatkan klasifikasi data yang lebih detail dan terperinci untuk memudahkan pengembang menganalisis *feedback* dari pengguna. *Prototype* ini dibangun dengan menggunakan model Machine Learning khususnya Multi-Layer Perceptron (MLP) dan Bidirectional LSTM (BiLSTM). Selain itu pemahaman tren ulasan oleh pengembang juga dibantu dengan pembuatan *dashboard* analisis data ulasan KAI Access.

Topik mengenai pengolahan data ulasan KAI Access mulai dari pembuatan model Machine Learning sampai pembuatan *dashboard* analisis data ulasan aplikasi KAI Access ini dipilih oleh penulis untuk dikaji pada tugas akhir ini karena selain data internal yang lebih mudah didapatkan, proyek ini memiliki data dengan detail yang lebih spesifik sebagai bahan analisis dibandingkan proyek lainnya. Terlebih, KAI Access menjadi aplikasi milik PT Kereta Api yang banyak digunakan saat ini sehingga hasil dari proyek ini dapat diterapkan secara langsung oleh tim KAI Access sebagai penunjang peningkatan layanan aplikasi KAI Access.

1.2 Ruang Lingkup

PT Kereta Api Indonesia (PT KAI atau ‘Perseroan’) atau dapat dilihat profilnya pada *website* PT Kereta api Indonesia (KAI, Situs Resmi PT Kereta Api Indonesia, 2017) memiliki sembilan Daerah Operasi (Daop) di Pulau Jawa yaitu Jakarta, Bandung, Cirebon, Semarang, Purwokerto, Yogyakarta, Madiun, Surabaya, dan Jember. PT KAI juga memiliki beberapa divisi regional yang tersebar di Pulau Sumatera yaitu Sumatera Utara dan Aceh, Sumatera Barat, Palembang, dan Tanjung Karang (KAI, PPD Daerah Operasional, 2022). Dari luasnya daerah operasional tersebut, PT KAI memiliki kantor pusat yang ada di Daerah Operasional 2 Bandung. Kantor Pusat PT KAI bertempat di Jalan Perintis Kemerdekaan nomor 1, Babakan Ciamis, Kec. Sumur Bandung, Kota Bandung pada Google Maps diperlihatkan pada Gambar 1.1. Halaman depan Kantor Pusat PT KAI diperlihatkan pada Gambar 1.2.



Gambar 1.1 Lokasi Kantor Pusat PT Kereta Api Indonesia di Bandung

Gambar 1.2 Halaman Depan Kantor Pusat PT KAI Bandung

Sumber: (Mayapasim1, 2012)

Permulaan perusahaan perkeretaapian dimulai pada tahun 1864, ketika pencangkulan pertama jalur kereta api Semarang - Yogyakarta yang dilakukan oleh Nederlandsch Indische Spoorweg Maatschappij (NISM). Selama 99 tahun perkeretaapian di Indonesia dipegang oleh pemerintahan Belanda hingga setelah Konferensi Meja Bundar (KMB) pada tahun 1949, Indonesia melakukan pengambilalihan aset – aset milik pemerintah Hindia Belanda termasuk NISM. Pada Tahun 1963, beberapa perkeretaapian di Indonesia dilakukan penggabungan dan berubah nama menjadi Perusahaan Negara Kereta Api (PNKA). Lalu pemerintah mengubah struktur PNKA pada tahun 1971 menjadi Perusahaan Jawatan Kereta Api (PJKA). Setelah itu pada tahun 1991, dalam rangka meningkatkan pelayanan jasa angkutan, PJKA berubah bentuk menjadi Perusahaan Umum Kereta Api (Perumka). Beberapa tahun kemudian, Perumka berubah menjadi perseroan terbatas yaitu PT Kereta Api (Persero) pada tahun 1998. Pada tahun 2011 nama perusahaan PT Kereta Api (Persero) berubah menjadi PT Kereta Api Indonesia (Persero) dengan meluncurkan logo baru (KAI Company Profile, 2021).

Walaupun PT Kereta api Indonesia (PT KAI) adalah perusahaan yang bergerak pada bidang jasa transportasi, namun PT KAI juga membutuhkan tenaga kerja di bidang informatika atau ilmu komputer. Contohnya seperti peran *software developer* yang diperlukan untuk pengembangan aplikasi KAI Access dan pengembangan *website* resmi PT KAI, *network engineer* yang mendesain, membangun, menjaga, serta mengelola jaringan komunikasi dan sistem perusahaan, hingga peran yang berhubungan dengan data seperti *data engineer* serta *data analyst* yang mengelola data perusahaan untuk manajemen dan peningkatan pelayanan perusahaan berdasarkan fakta dari kumpulan data.

Penulis melaksanakan kegiatan magang di PT Kereta Api Indonesia (PT KAI) pada divisi IT *Enterprise Resource Planning* (ERP), bertempat di Gedung E2 Kantor Pusat Kereta Api Indonesia yang diperlihatkan pada Gambar 1.3. Ditugaskan sebagai Data Scientist Intern di PT KAI bekerjasama dengan *data engineer* dan *data analyst*, penulis diberikan proyek yang berkaitan dengan pengolahan data. Tugas dari Data Scientist Intern PT KAI ini adalah mengumpulkan dan membersihkan data, membuat model Machine Learning, dan menganalisis data serta memvisualisasikan hasilnya.

Gambar 1.3 Gedung E2 Kantor Pusat Kereta Api Indonesia di Bandung

Pada awal pertemuan, penulis menadakan *kick-off meeting* bersama mentor untuk menentukan rencana kegiatan yang dilakukan oleh penulis selama magang dalam enam bulan ke depan termasuk proyek-proyek yang dikerjakan. Dari hasil *meeting* tersebut, ditentukan dua proyek yang penulis lakukan yaitu pengolahan data ulasan aplikasi KAI Access dan pengolahan data cuitan pada aplikasi Twitter. Masing-masing proyek yang dikerjakan memiliki durasi waktu pengerjaan selama tiga bulan. Sebelum langsung bersinggungan dengan data yang diolah, penulis serta mentor melakukan eksplorasi jurnal publik yang memiliki topik serupa dengan proyek yang direncanakan untuk menjadi referensi dalam pengerjaan proyek. *Timeline* pengerjaan kedua proyek tersebut ditampilkan pada Tabel 1.1. Awal pelaksanaan kegiatan magang pada 14 September 2021 dan pengerjaan proyek dimulai bulan September 2021 hingga Maret 2022. Proyek dinyatakan selesai apabila hasil produk proyek tersebut telah selesai pembuatannya, diserahkan, dan telah disetujui oleh mentor.

Tabel 1.1 *Timeline* Aktivitas Magang

No	Tanggal	Kegiatan	Lokasi
1	14 September 2021	*Planning Meeting*	Rumah
2	15 - 22 September 2021	Eksplorasi Referensi Metode Pengolahan Data	Rumah
3	23 September 2021 - 22 Oktober 2021 dan 29 Januari 2022 – 9 Maret 2022	Pengerjaan Proyek Data Twitter KAI121	Rumah dan Kantor

4	23 Oktober 2021 - 28 Januari 2022	Pengerjaan Proyek Data KAI Access	Rumah dan Kantor

1.2.1 Proyek Data KAI Access

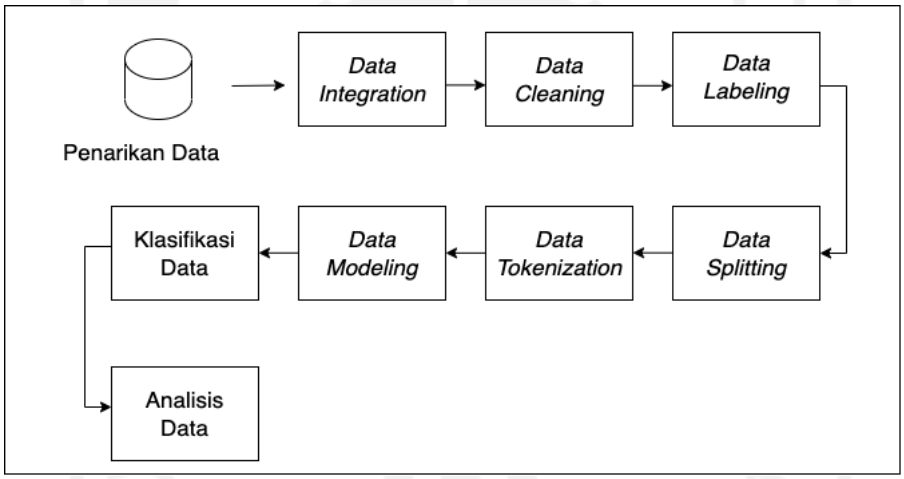
KAI Access adalah satu-satunya aplikasi resmi yang dikeluarkan oleh PT Kereta Api Indonesia untuk penjualan tiket kereta api secara daring. Aplikasi ini tidak hanya dibuat untuk penjualan tiket, beberapa fitur telah ditambahkan untuk kemudahan dan pelayanan kepada *customer* PT Kereta Api Indonesia (KAI, Situs Resmi PT Kereta Api Indonesia, 2017). Banyak transaksi yang dilakukan pada aplikasi ini, terutama semenjak pandemi Covid-19 berlangsung yang mengharuskan calon penumpang melakukan segala transaksi tiket kereta api secara daring yang salah satunya melalui aplikasi KAI Access. Pemesanan dan pembelian tiket, pengubahan jadwal keberangkatan, pembatalan tiket, transaksi KA Logistik, pemesanan makanan dan minuman di dalam kereta api, hingga *top up* dan pembayaran tagihan dengan *e-wallet* KAI Access yaitu KAIPay dapat dilakukan dalam aplikasi ini. Halaman utama aplikasi KAI Access diperlihatkan pada Gambar 1.4.

Gambar 1.4 Halaman Utama Aplikasi KAI Access

Penggunaan aplikasi terus meningkat sehingga diperlukannya peningkatan kualitas dari aplikasi ini untuk meningkatkan kepuasan pengguna. Oleh karena itu dengan memanfaatkan data ulasan KAI Access pada *platform* distribusi Play Store, penulis mengembangkan sistem klasifikasi sentimen, topik, dan detail topik data. Sistem klasifikasi sentimen, topik, dan detail

topik data ulasan KAI Access ini menggunakan Natural Language Processing (NLP) dengan model Machine Learning yang akan mempelajari data teks ulasan yang ditulis oleh pengguna.

Pengerjaan proyek pengolahan data ulasan KAI Access ini memakan waktu kurang lebih tiga bulan dengan *flow* pengerjaan mulai dari penggabungan data hingga analisis data ulasan KAI Access sebagaimana diilustrasikan pada Gambar 1.5. Pengerjaan dimulai dari penerimaan *dataset* ulasan KAI Access, *data integration, data cleaning, data labelling, data splitting, data tokenization, data modeling, klasifikasi data* atau pembuatan *prototype,* dan teakhir pembuatan *dashboard* analisis. Beberapa *tools* dipakai oleh penulis dalam pengerjaan proyek ini yaitu Google Colab, PyCharm, Google Sheet, dan TensorFlow. Penulis memakai arsitektur Multi-layer Perceptron (MLP) dan *Bidirectional Long Short Term Memory* (BiLSTM) untuk model proyek ini. Data ulasan KAI Access juga akan dianalisis dan divisualisasikan dalam bentuk *dashboard* menggunakan *tools* Tableau.



Gambar 1.5 *Flow* Pengolahan Data Proyek KAI Access dan Twitter KAI121

Selama pengerjaan proyek KAI Access, penulis bekerja bersama *data engineer* dan seorang *data analyst* yang juga berperan sebagai mentor ketika penulis melakukan kegiatan magang. mentor memiliki intensitas interaksi yang tinggi dengan penulis, membimbing dalam setiap proses pengolahan data serta membantu menghubungkan penulis dengan pihak lain yang terkait yaitu *data engineer*. Selain itu, mentor juga membantu penulis dalam pengerjaan proyek yaitu menentukan label data dan analisis dari data. *Data engineer* berperan menarik data ulasan KAI Access dari database dan disajikan dalam format CSV yang selanjutnya akan diolah oleh penulis.

Hasil dari proyek KAI Access ini adalah data dapat diklasifikasikan secara langsung dengan Machine Learning. Klasifikasi pada proyek ini dilakukan untuk tiga label yaitu

sentimen, topik, dan detail topik. Pengadaan label detail topik pada proyek KAI Access karena proyek ini membutuhkan hasil klasifikasi topik yang lebih detail untuk dapat mengidentifikasi keluhan pengguna pada sistem dengan spesifik lebih cepat. Selain klasifikasi, proyek ini juga menghasilkan *insight* dari analisis yang dilakukan dari data ulasan KAI Access ini. *Insight* yang divisualisasikan seperti jumlah data ulasan perbulan, versi yang paling banyak dipakai, jumlah dan rata-rata rating penilaian, dan jumlah data setiap kelas pada label sentimen, topik, dan detail topik.

1.2.2 Proyek Data Twitter KAI121

Twitter KAI121 adalah akun resmi layanan pelanggan PT Kereta Api Indonesia pada *platform* Twitter yang digunakan untuk membagikan informasi seputaran operasional kereta api secara keseluruhan setiap harinya. Akun Twitter KAI121 diperlihatkan pada Gambar 1.6. Pada akun ini masyarakat kerap memberikan pertanyaan terkait operasional kereta api dan kerap membagikan pengalaman penumpang dalam memakai jasa angkutan PT Kereta Api Indonesia yang dapat dilihat oleh seluruh pengguna Twitter.

Gambar 1.6 Akun Twitter KAI121

PT Kereta Api Indonesia ingin membuat analisis dan klasifikasi otomatis dengan data cuitan pada Twitter KAI121 yang nantinya akan dimanfaatkan untuk meningkatkan kualitas pelayanan pada Twitter KAI121 juga pelayanan operasional PT Kereta Api Indonesia berdasarkan tren keluhan, pertanyaan, dan pengalaman yang dibagikan oleh masyarakat di Twitter KAI121. Latar belakang tersebut membuat penulis mengembangkan proyek pembuatan model Machine Learning, Natural Processing Language (NLP), model yang dapat mempelajari data cuitan Twitter KAI121, melakukan klasifikasi secara *real time* dan hasilnya akan dijadikan bahan analisis untuk tim *big data* sebagai langkah peningkatan pelayanan PT Kereta Api Indonesia untuk masyarakat.

Pengerjaan proyek pengolahan data Twitter KAI121 ini memakan waktu kurang lebih tiga bulan dengan *flow* pengerjaan mulai dari penarikan data dari *platform* Twitter hingga analisis data Twitter KAI121 sebagaimana diilustrasikan pada Gambar 1.5. Pengerjaan dimulai dari penarikan data Twitter dengan Twitter API, *data integration, data cleaning, data labelling, data splitting, data tokenization, data modeling, klasifikasi data* atau pembuatan *prototype,* dan teakhir analisis data. Terdapat sedikit perbedaan dalam proses pengolahan data KAI Access dan Twitter KAI121, proyek data Twitter KAI121 ini tidak melibatkan *data engineer* untuk mengambilan data. Pengambilan data Twitter dilakukan sendiri oleh penulis, sedangkan data ulasan KAI Access diambil oleh *data engineer* perusahaan. Pengambilan data Twitter KAI121 menggunakan Twitter API membutuhkan waktu selama kurang lebih satu bulan karena terbatasnya jumlah data yang bisa diambil setiap harinya dengan API tersebut. Walaupun begitu, selama pengerjaan proyek ini penulis tetap dibimbing penuh oleh mentor yang juga membantu dalam menentukan label data dan analisis dari data.

Dalam pengerjaan proyek ini beberapa *tools* dipakai diantaranya tidak jauh berbeda dengan *tools* yang dipakai pada proyek data KAI Access yaitu Google Colab, PyCharm, Google Sheet, TensorFlow, dan Twitter Developer Access. Metode Machine Learning yang digunakan dalam klasifikasi dengan data Twitter KAI121 ini adalah model yang memakai arsitektur Multi-layer Perceptron (MLP). Data Twitter KAI121 juga akan dianalisis dan divisualisasikan dengan *tools* Tableau.

Hasil yang diperoleh dari proyek ini sedikit berbeda dengan hasil proyek KAI Access yaitu hanya menghasilkan klasifikasi sentimen dan topik serta *dashboard* analisis data Twitter KAI121, tanpa klasifikasi detail topik. *Insight* yang didapatkan dari analisis dan visualisasi data KAI121 seperti jumlah cuitan yang menyebutkan akun KAI121 perbulannya dan jumlah dari

masing-masing sentimen dan topik dari data tersebut juga akan dibuat dalam bentuk *dashboard*, serupa dengan salah satu produk akhir proyek KAI Access.

Tentu banyak pengalaman dan hal bermanfaat yang penulis peroleh dari kegiatan magang sebagai Data Scientist di PT KAI, dari segi teknis maupun non teknis. Dari segi non teknis, melalui kegiatan magang ini penulis mendapatkan pengalaman bekerja secara langsung di kantor dengan bekerjasama dengan pegawai kantor lainnya yang belum pernah dirasakan sebelumnya oleh penulis. Bekerjasama dengan pegawai lain dan bekerja pada lingkungan kerja yang sesungguhnya menuntut penulis untuk terus mengembangkan kemampuan berkomunikasi, adaptasi dengan lingkungan pekerjaan, dan manajemen waktu yang baik. Dari segi teknis, penulis mendapatkan pengalaman untuk mengolah *real dataset* milik perusahaan yang berbeda dengan data publik.

1.3 Tujuan

Mengembangkan tiga model untuk tiga klasifikasi yang berbeda yaitu klasifikasi sentimen, topik, dan detail topik yang masing-masing terdiri dari beberapa kelas untuk membangun *prototype* klasifikasi otomatis data ulasan KAI Access dengan cepat dan tepat serta membangun *dashboard* dari visualisasi data ulasan KAI Access.

1.4 Manfaat

Data ulasan aplikasi KAI Access yang diklasifikasikan pada klasifikasi label sentimen, topik, dan detail topik secara otomatis dapat digunakan oleh tim KAI Access untuk menganalisis permasalahan yang banyak dikeluhkan oleh pengguna pada laman ulasan aplikasi di Play Store. Dengan mengetahui masalah-masalah yang dikeluhkan oleh para pengguna dengan cepat, tim dapat secepat mungkin memperbaiki atau meningkatkan fitur aplikasi KAI Access sesuai dengan hasil klasifikasi tersebut. Hal ini dapat mengefisiensikan waktu dan memudahkan pekerjaan tim KAI Access.

KAI Access sebagai layanan teknologi digital utama dalam pelayanan PT Kereta Api Indonesia, kualitas pelayanan aplikasi ini tentu menjadi hal yang krusial untuk perusahaan. Dengan adanya sistem klasifikasi otomatis data ulasan KAI Access yang memudahkan tim KAI Access dalam mengevaluasi aplikasi, tentunya juga akan membantu perusahaan dalam meningkatkan kualitas pelayanan aplikasi ini. Diharapkan kepercayaan dan kenyamanan pengguna kepada perusahaan meningkat sejalan dengan meningkatnya kualitas pelayanan aplikasi KAI Access.

1.5 Sistematika penulisan

Laporan tugas akhir ini terdiri dari 5 bab, yaitu:

a. Bab I Pendahuluan

   Bab ini terdiri dari latar belakang, ruang lingkup, tujuan, dan manfaat dari kegiatan magang yang dilakukan oleh penulis selama enam bulan sebagai Data Scientist di PT Kereta Api Indonesia (Persero) serta sistematika penulisan tugas akhir.

b. Bab II Landasan Teori dan Tinjauan Pustaka

   Bab ini berisi teori dasar dari *data labeling*, *data preprocessing*, algoritma Multi-layer Preceptron (MLP) dan algoritma Bidirectional Long Short Term Memory (BiLSTM). Bab ini juga menjelaskan beberapa penelitian terdahulu yang berkaitan dengan teori-teori dasar pengetahuan dalam proyek yang dikerjakan.

c. Bab III Pelaksanaan Magang

   Menjelaskan mengenai manajemen, metodologi, implementasi dan hasil dari pengerjaan proyek pada kegiatan magang khususnya pada proyek data ulasan KAI Access.

d. Bab IV Refleksi Pelaksanaan Magang

   Menjelaskan mengenai relevansi metode-metode yang diterapkan pada proyek magang dengan kajian serta *gap* akademik dan menjelaskan mengenai manfaat, hambatan, serta tantangan yang didapatkan penulis selama kegiatan magang.

e. Bab V Penutup

   Berisi kesimpulan dari laporan tugas akhir dan saran perbaikan untuk kekurangan yang ada dalam pengerjaan proyek di kegiatan magang di PT KAI.

BAB II
LANDASAN TEORI DAN TINJAUAN PUSTAKA

2.1 Data Labeling

Data labeling adalah proses identifikasi data dan menambahkan tanda kepada data mentah untuk menyatakan pada model Machine Learning sebagai jawaban atau target dari apa yang diharapkan dari prediksinya (Qorita & Rahma, 2021). Sebuah label adalah elemen deskriptif yang dapat memberitahu model Machine Learning sehingga model dapat mempelajari dari contoh data. Data yang sudah dilabeli mempunyai karakteristik yang membantu model pembelajaran untuk menganalisis informasi dan mengidentifikasi pola untuk membuat prediksi yang akurat pada *input* baru. *Labeling* dapat dilakukan dengan manual oleh manusia secara langsung ataupun secara otomatis dengan mesin.

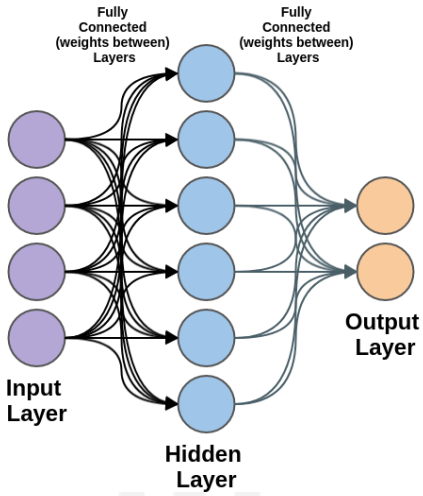
2.2 Data Preprocessing

Data mentah yang diambil dari sumber pada umumnya tidak memiliki kualitas yang bagus, data mentah ini cenderung tidak konsisten dan berantakan. Proses *data preprocessing* dapat membantu menyiapkan data untuk dapat diproses lebih baik di langkah proses selanjutnya. Proses ini dapat mengecek maupun meningkatkan kualitas data dari keakurasian, kelengkapan, konsistensi, aktualitas, kepercayaan, dan kemampuan interpretasi data tersebut (Allen & Cervo, 2015). Hal tersebut dilakukan dengan melakukan *data integration*, *cleaning*, *reduction*, dan *transformation*. *Data integration* yaitu proses menggabungkan beberapa *source* data menjadi satu *dataset*, *data cleaning* yaitu membersihkan bagian-bagian data yang kurang sempurna, *data reduction* yaitu mengurangi bagian-bagian dari data yang jumlahnya banyak dan tidak relevan dengan model dan analisis yang akan dilakukan, dan *data transformation* yaitu mengubah data ke dalam format yang sesuai untuk pengolahan data teks. Dengan data *preprocessing*, data ulasan mentah diubah menjadi format yang lebih dapat dipahami.

2.3 Multi-Layer Perceptron (MLP)

Jaringan saraf tiruan atau Artificial Neural Network (ANN) adalah sebuah teknik pengolahan informasi yang terinspirasi dari sistem kerja saraf pada otak manusia. Multi-layer Perceptron (MLP) adalah arsitektur ANN yang paling mendasar dan efektif dibandingkan algoritma lainnya dibandingkan dengan arsitektur ANN lainnya. Multi-layer Perceptron (MLP)

adalah arsitektur model yang berisi Perceptron Layer yang saling terhubung dan terdiri dari 3 tipe layer yaitu *input* dan *output layer* serta *hidden layer* diantaranya seperti yang diperlihatkan pada Gambar 2.1. *Input layer* menerima *input* untuk diproses dan hasil klasifikasi akan ditampilkan oleh *output layer* (S & P, 2019). Penggunaan jumlah *hidden layer* pada MLP tergantung kebutuhan, namum umumnya jumlah *hidden layer* pada MLP adalah berjumlah satu hingga tiga *hidden layer*. Banyaknya *layer* yang menyusun model inilah yang menjadikan MLP mempunyai kemampuan adaptasi dan pembelajaran mandiri yang kuat dan data dapat diproses lebih efektif (Samandianfard, et al., 2020). Permasalahan yang paling banyak diatasi oleh MLP adalah klasifikasi, pengenalan, prediksi, dan perkiraan.



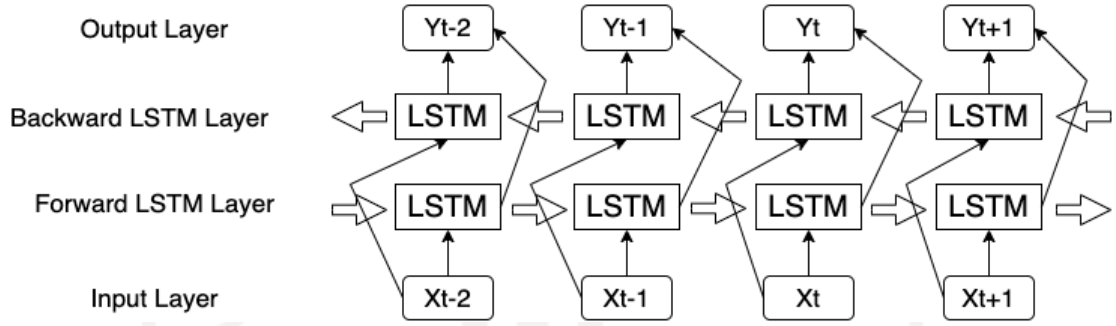
Gambar 2.1 Arsitektur Multi-layer Perceptron (MLP)
(Walters, 2019)

2.4 Bidirectional Long Short Term Memory (BiLSTM)

Bidirectional Long Short Term Memory (BiLSTM) adalah bentuk pengembangan dari arsitektur jaringan Long Short Term Memory (LSTM) yang merupakan hasil modifikasi dari arsitektur RNN dengan menambahkan *memory cell* yang dapat menyimpan informasi dalam waktu yang lama (Masnawi, 2018). LSTM mengkomputasi *hidden state* yang dapat merekam memori ketergantungan jangka panjang (*long-term dependencies memory*). Arsitektur LSTM juga baik dalam mengatasi data secara sekuensial. Seperti namanya, BiLSTM mengembangkan arsitektur LSTM dimana BiLSTM memiliki dua lapisan yang masing-masing memiliki arah proses yang saling berkebalikan. Lapisan pertama bergerak mundur dari urutan *input*nya atau *backward*, memahami atau memproses kata dari kata di akhir ke kata awal. Lapisan kedua bergerak maju atau *forward*, memahami kata sesuai dengan urutan maju dari awal ke akhir. Dengan kata lain, BiLSTM tidak hanya mendapatkan atau mempelajari dari informasi yang

diberikan dari *hidden state* sebelumnya, namun BiLSTM juga memungkinkan sebuah *hidden state* juga dapat mempelajari informasi dari *state* setelahnya. Dengan kemampuan BiLSTM ini, model dapat memahami perspektif kalimat secara lebih luas sehingga dapat memahami konteks kalimat dengan lebih cepat dan akurat.

Seperti yang diperlihatkan pada Gambar 2.2, *hidden layer* yang ada pada arsitektur BiLSTM lebih panjang dibandingkan dengan arsitektur LSTM karena BiLSTM memproses secara dua arah. *Input* dimasukkan pada *layer* pertama lalu diproses secara *forward*, lalu *input* juga masuk ke dalam layer *kedua* yang diproses secara *backward*, dan kedua informasi dari kedua *layer* yang berbeda arah tersebut digabungkan sebagai informasi yang dihasilkan menjadi *ouput*. Oleh karena itu, informasi yang diproses dengan BiLSTM ini dapat mengklasifikasikan dengan lebih akurat dibandingkan dengan algoritma LSTM yang hanya memproses satu arah.

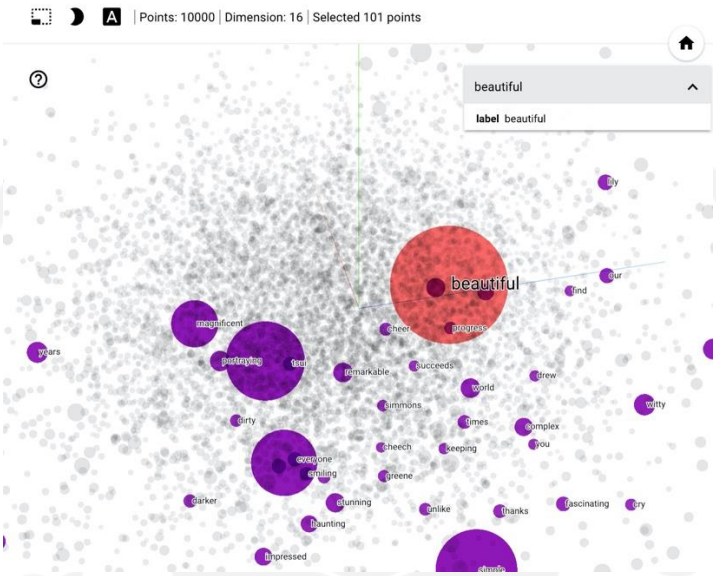


Gambar 2.2 Arsitektur Bidirectional Long Short Term Memory (BiLSTM)

2.5 Word Embedding

Word embedding adalah proses mengubah kata pada data *input* menjadi ruang vektor dengan dimensi yang tereduksi, jarak antara kata pada ruang vektor ditentukan dengan beberapa bentuk korelasi semantik (Venkatesh, Moffat, & Miranda, 2022). Setiap kata pada data menjadi vektor yang mempresentasikan sebuah titik pada ruang dengan dimensi tertentu. *Word embedding* membantu *neural network* mengelompokkan teks ke dalam vektor berdasarkan kata pada data tersebut dan kemiripan struktur data dengan data lainnya pada kelompok label data yang sama, gambaran pengelompokan kata dengan vektor divisualisasikan pada Gambar 2.3. Pada gambar tersebut, kata *beautiful*, *cheer*, dan *succeeds* memiliki vektor yang berdekatan dengan kata lain pada data latihnya, kata-kata tersebut memiliki korelasi semantik yang serupa. Contohnya adalah kata-kata yang banyak muncul pada teks dengan label yang sama atau dua kata yang seringkali muncul bersama, akan saling berdekatan vektornya begitu juga dengan label Positif dan Netral. *Layer* untuk melakukan *word embedding*,

Embedding Layer, kerap dipakai sebagai *layer* pertama pada *neural network* yang mengatasi permasalahan NLP seperti mesin penerjemah, pembuatan *caption*, *speech recognition* otomatis, dan tugas NLP lainnya (Goldbreg, 2017). Oleh karena itu, Embedding Layer adalah komponen yang penting dalam *neural network* untuk pemrosesan bahasa.

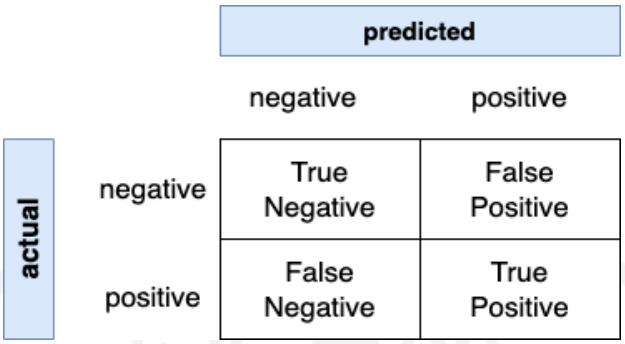


Gambar 2.3 Visualisasi Word Embedding TensorFlow
(TensorFlow, TensorFlow Word Embeddings, 2022)

2.6 Confusion Matrix

Confusion matrix adalah tabel yang digunakan untuk menentukan kinerja algoritma klasifikasi (Tripathy, Agrawal, & Rath, 2016). Confusion Matix memvisualisasikan dan merangkum kinerja algortima klasifikasi dengan beberapa istilah yaitu True Positive (TP), True Negative (TN), False Negative (FN), False Positive (FP) seperti yang ditunjukkan pada Gambar 2.4. True Positive (TP) mengartikan bahwa algoritma memprediksi kelas positif dan hal tersebut benar adanya (prediksi kelas positif, aktual kelas positif). True Negative (TN) berarti algoritma memprediksi data dikelompokkan ke dalam kelas negative dan itu benar (prediksi kelas negatif, aktual kelas negatif). False Positive (FP) berarti algoritma memprediksi data dikelompokkan pada kelas positif padahal data tersebut sebenarnya negatif (prediksi kelas positif, aktual kelas negatif). False Negative (FN) artinya algoritma mengelompokkan data ke dalam kelas negatif dan hal tersebut salah (prediksi kelas negatif, aktual kelas positif). Keempat parameter tersebut (TP, TN, FN, dan FP) dapat digunakan untuk mengevaluasi kinerja

klasifikasi menggunakan matriks evaluasi: akurasi, presisi, *recall*, dan F1 *score* (Dellia & Tjahyanto, 2017).



Gambar 2.4 Bagan Confusion Matrix

Akurasi adalah ukuran umum untuk mengevaluasi kinerja klasifikasi. Akurasi adalah rasio data yang diprediksi dengan benar dengan jumlah total data. Akurasi dapat dihitung dengan rumus yang ditampilkan pada persamaan (2.1).

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.1)$$

Presisi adalah rasio data yang diberi label dengan benar sebagai positif untuk semua data yang diprediksi positif, seperti yang ditunjukkan pada persamaan (2.2).

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

Recall adalah rasio data yang diberi label dengan benar sebagai positif untuk semua data yang benar-benar memiliki label positif. Nilai *recall* dapat dihitung seperti pada persamaan (2.3).

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

F1 score adalah nilai rata-rata presisi dan *recall*. Nilai F1 *score* yang lebih tinggi, nilai presisi dan penarikan yang lebih tinggi. F1 score dapat dihitung seperti pada persamaan (2.4).

$$\frac{1}{F1\ score} = \frac{1}{2}\left(\frac{1}{presisi} + \frac{1}{recall}\right) \quad (2.4)$$

2.7 Tinjauan Pustaka

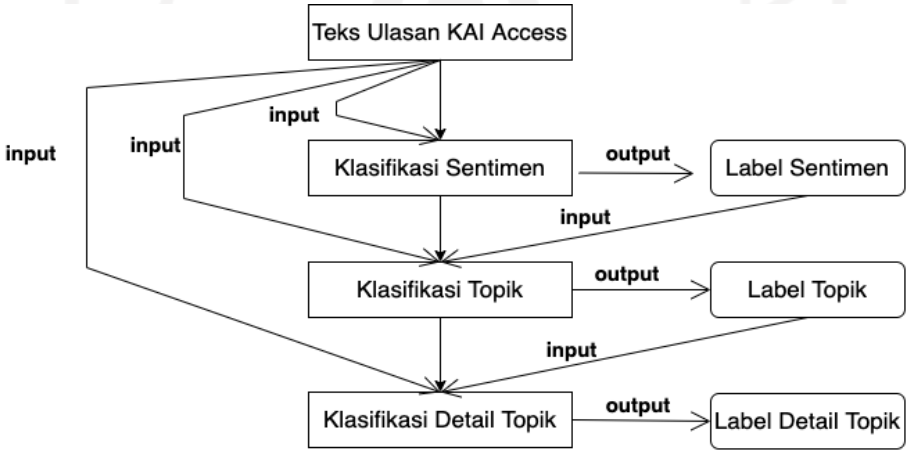
Identifikasi teks ulasan yang terperinci dengan menghasilkan beberapa label dapat mendukung evaluasi aplikasi dengan lebih baik. Klasifikasi dalam NLP dengan lebih dari satu label yang saling terkait dapat digunakan dengan klasifikasi *multi-label*. Klasifikasi yang memiliki target kelasnya lebih dari satu disebut *multi-class classification.* Tidak banyak penelitian yang serupa dengan klasifikasi pada data ulasan KAI Access ini, penelitian yang menerapkan klasifikasi *multi-class multi-label,* klasifikasi lebih dari satu label yang setiap labelnya memiliki jumlah kelas lebih dari dua. Salah satu penelitian yang serupa adalah penelitian klasifikasi *multi-class multi-label* dengan jumlah kelas yang besar (Dekel & Shamir, 2010). Penelitian ini membuat klasifikasi yang memiliki 1,5 juta label berupa kategori artikel Wikipedia dan pada setiap kategori tersebut memiliki beberapa kelas berupa *keyword* pengkategorian. Namun, penelitian ini menggunakan model tunggal untuk klasifikasi yang menghasilkan lebih dari satu label. Sedangkan, tingkat kepentingan setiap label dalam mengkarakterisasi semantik adalah relatif atau berbeda-beda (Zhang, Li, Liu, & Geng, 2017), seperti proyek data ulasan KAI Access ini yang setiap labelnya memiliki makna yang berbeda. Oleh karena itu, dalam penelitian ini dibuat arsitektur model yang berbeda berdasarkan kompleksitas setiap klasifikasi dan pelatihan model yang terpisah digunakan untuk setiap model.

Klasifikasi sentimen pada data teks telah banyak dilakukan oleh para peneliti mulai dari sentimen cuitan seseorang, ulasan, sampai teks percakapan. Berbagai pilihan arsitektur model Machine Learning dapat digunakan untuk membangun klasifikasi sentimen ini, salah satunya adalah arsitektur Multi-layer Perceptron (MLP). Terdapat penelitian yang menggunakan MLP sebagai arsitektur klasifikasi teks ulasan, penelitian yang dibuat oleh Livingstone et al (2019) ini melakukan klasifikasi sentimen pada dataset besar teks ulasan produk pada *platform* penjualan *online* Amazon. Arsitektur MLP digunakan pada penelitian ini dengan hasil akurasi yang tinggi yaitu pada angka 93,29% (Livingston, Tamil Selvi, Thabeetha, Grena, & Jenifer, 2019). Klasifikasi teks ulasan Amazon pada penelitian ini memiliki dua kelas sentimen, yaitu kelas Positif dan kelas Negatif.

Menentukan topik dan detail topik dari suatu data teks ulasan adalah hal yang cukup rumit bagi model untuk diklasifikasikan karena tidak terbatasnya kata, bahasa, dan panjang ulasan yang diberikan oleh pengguna. Terlebih, jika kelas yang ditentukan berjumlah banyak. Berbeda dengan model klasifikasi sentimen, model klasifikasi topik menggunakan arsitektur model BiLSTM. Digunakannya arstiketur ini karena LSTM itu sendiri mampu mengatasi

permasalahan NLP yang kompleks dengan baik (Md.Kowsher, et al., 2014), terlebih BiLSTM yang mempunyai efektifitas yang lebih baik karena memproses *input* data secara dua arah. Hal ini dibuktikan oleh penelitian yang membandingkan performa LSTM dan BiLSTM dalam identifikasi *cyberbullying* pada cuitan Twitter (Fadli & Hidayatullah, 2020). Hasil dari penelitian ini yaitu akurasi akhir arsitektur BiLSTM lebih unggul hampir 2% dibandingkan dengan akurasi akhir arsitektur LSTM. Klasifikasi topik termasuk klasifikasi yang kompleks karena model harus bisa menentukan data ulasan diklasifikasikan pada topik tertentu dan jumlah kelas yang cukup banyak. Salah satu klasifikasi *multi-class* dengan data teks telah dilakukan oleh Asghar et al.(2022). Pada penelitian tersebut, dilakukan pembuatan model klasifikasi pada data teks yang terdapat pada media sosial untuk menentukan apakah suatu data teks mengandung emosi sedih, senang, takut, malu, atau bersalah (Ashgar, et al., 2022). Klasifikasi emosi pada data media sosial ini dibangun dengan menggunakan metode BiLSTM dan mendapatkan hasil akurasi akhir sebesar 87,66%.

Berdasarkan dengan tinjauan beberapa jurnal di atas, pengerjaan proyek ini dilakukan dengan pelatihan model yang terpisah dan setiap model klasifikasi yang memiliki arsitektur yang berbeda yaitu MLP dan BiLSTM. Selain itu, hasil *prototype* yang diharapkan yaitu pada klasifikasi setiap label bergantung pada hasil klasifikasi label sebelumnya seperti yang diperlihatkan pada Gambar 2.5 guna mendukung keakuratan klasifikasi. Klasifikasi sentimen memakai teks ulasan sebagai data *input*, klasifikasi topik memakai data teks ulasan dan label sentimen sebagai data *input*, dan data *input* klasifikasi detail topik adalah teks ulasan, label sentimen, dan label topik.



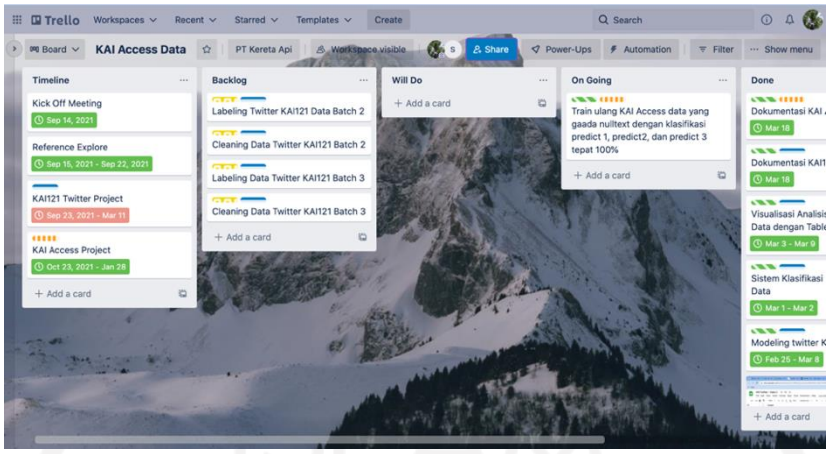
Gambar 2.5 Alur Klasifikasi Multi-Class Multi-Label Data Ulasan KAI Access

BAB III
PELAKSANAAN MAGANG

3.1 Manajemen Proyek

Penulis menjalani kegiatan magang sebagai Data Scientist PT KAI di tengah-tengah pandemi Covid-19 yang terjadi di Indonesia dan dunia. Oleh sebab itu, penulis tidak bisa melakukan kegiatan magang di kantor (*work from office*) setiap hari karena adanya pembatasan resmi dari pemerintah yang tidak memperbolehkan perusahaan memberlakukan Work From Office (WFO) seratus persen. Ditambah dengan kondisi diberlakukannya PPKM, sehingga penulis dijadwalkan untuk bekerja dari kantor di kantor pusat PT KAI Daop 2 Bandung hanya sekali dalam dua minggu. Selebihnya, dijadwalkan untuk bekerja dari rumah (*work from home*) dan melaporkan *progress* selama seminggu sekali secara virtual. Jumlah jam kerja setiap harinya adalah delapan jam, mulai bekerja dimulai dari jam 9 pagi dan selesai pada jam 5 sore.

Selama enam bulan penulis menjalankankan kegiatan magang, komunikasi adalah suatu hal yang sangat penting dalam pengerjaan proyek-proyek yang telah ditugaskan. penulis berkomunikasi aktif dengan mentor terkait teknis proyek maupun manajemen proyek. Manajemen proyek dilakukan dengan menggunakan *tools* Trello yang ditampilan pada Gambar 3.1. Penulis akan membuat daftar *task* yang dikerjakan dan memetakannya pada *card* sesuai *state*-nya. Trello dipilih karena *tools* ini cukup membantu dalam *managing progress* proyek dan mentor dapat memantau *progress* proyek secara langsung. Untuk komunikasi secara dua arah yang dilakukan saat bekerja dari rumah, penulis bersama mentor berkomunikasi khususnya berdiskusi dengan menggunakan *platform* komunikasi video *online* Google Meet atau Zoom dan saat bekerja dari kantor berdiskusi dilakukan secara langsung atau *face-to-face*. Untuk komunikasi tersurat, penulis dan mentor menggunakan media WhatsApp untuk respon yang lebih cepat.



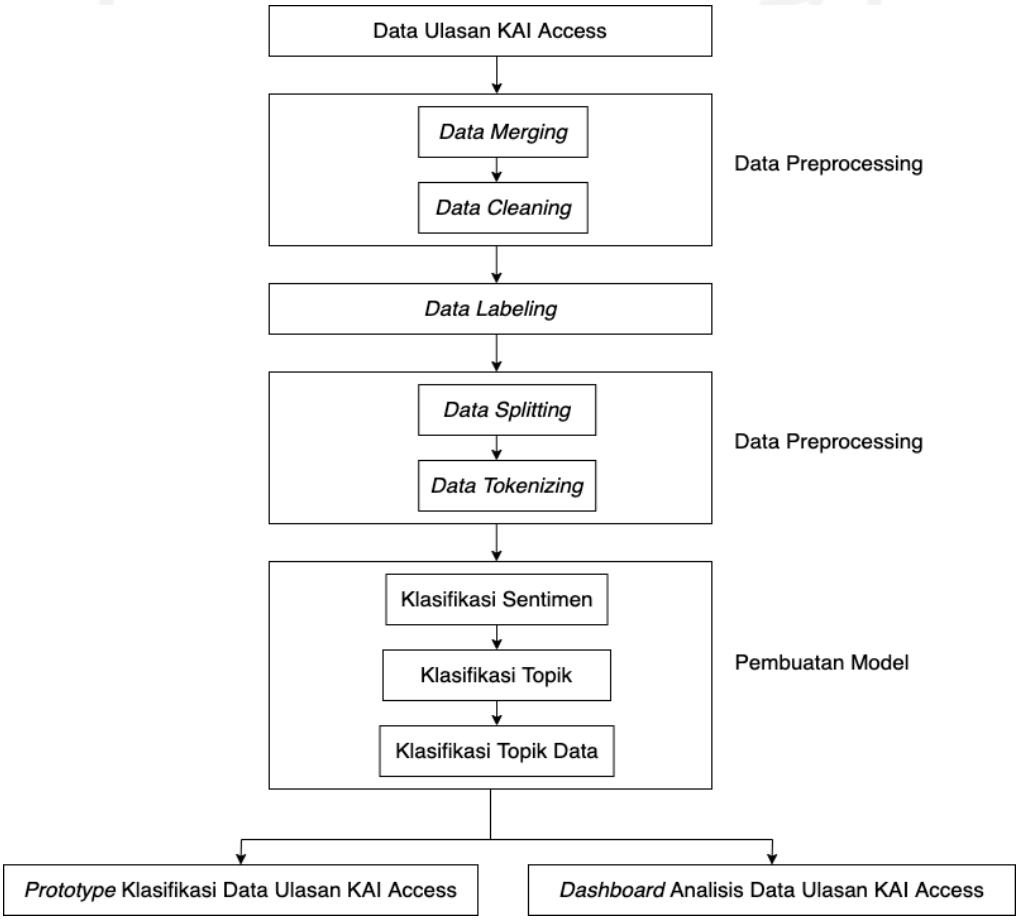
Gambar 3.1 Manajemen Proyek Data Ulasan KAI Access dengan Trello

Proyek yang ditugaskan kepada penulis selama kegiatan magang ditentukan dalam *kick-off meeting* yang dihadiri oleh penulis dan mentor. Proyek ditentukan berdasarkan kebutuhan perusahaan, ketersediaan data, dan durasi kegiatan magang. Secara keseluruhan, pengerjaan proyek dilakukan oleh penulis sendiri namun tetap dengan arahan, saran, dan bantuan mentor. Namun, pengambilan data perusahaan yang akan diolah oleh penulis menjadi pengecualian karena data diambil oleh *data engineer* yang memiliki akses ke dalam penyimpanan data internal perusahaan yang selanjutnya diberikan kepada penulis melalui perantara mentor. Sumber data yang dipakai ialah data ulasan KAI Access yang diambil dari *database* KAI Access dan data Twitter KAI121 yang ditarik oleh penulis menggunakan Twitter. Setiap satu langkah *progress* penulis telah diselesaikan, progress tersebut akan disampaikan kepada mentor untuk diberikan *feedback* dan keputusan apakah dapat dilanjut ke langkah berikutnya atau perlu direvisi. Jika diberikan revisi, mentor akan memberikan arahan dan penjelasan mengenai poin yang perlu direvisi serta penulis kembali melaporkan revisi yang telah diselesaikan dikemudian hari. Pertemuan antara penulis dan mentor dijadwalkan rutin selama sekali dalam seminggu. Dalam pertemuan rutin tersebut dilakukan pelaporan *progress* oleh penulis terkait proyek yang sedang dikerjakan penulis, khususnya *task* yang sedang dikerjakan maupun yang baru selesai dikerjakan.

Akhir proyek dilakukan dengan mengumpulkan *deliverables* produk proyek yang sebelumnya telah disetujui oleh mentor dan mengumpulkannya dalam satu *folder* pada Google Drive berisi seluruh dokumen pendukung lainnya seperti *dataset* yang digunakan, berkas *code* pengembangan produk, dan dokumentasi lengkap dari proyek tersebut. *Folder* ini dapat diakses oleh penulis juga mentor.

3.2 Metodologi

Pada pengerjaan proyek data ulasan KAI Access ini dilakukan beberapa proses dalam pengolahan sampai pengklasifikasian data. Gambar 3.2 menggambarkan alur pengerjaan proyek yang dilakukan pada proyek data KAI Access ini secara garis besar. Proses dimulai dari penggabungan data yang ada pada proses *data merging*, lalu data dibersihkan pada proses *data cleaning*. Sebelum proses *preprocessing* lainnya dilakukan, terlebih dahulu dilakukan *data labeling*. Setelah dilabeli, data dipisahkan menjadi beberapa bagian dan dilakukan *data tokenizing* agar data dapat dimasukkan ke dalam model. Setelah data siap, dilakukanlah *data modeling* untuk ketiga klasifikasi: klasifikasi sentimen, topik, dan detail topik. Terakhir yaitu pembuatan produk proyek ini yaitu pembuatan *prototype* dan *dashboard* analisis.

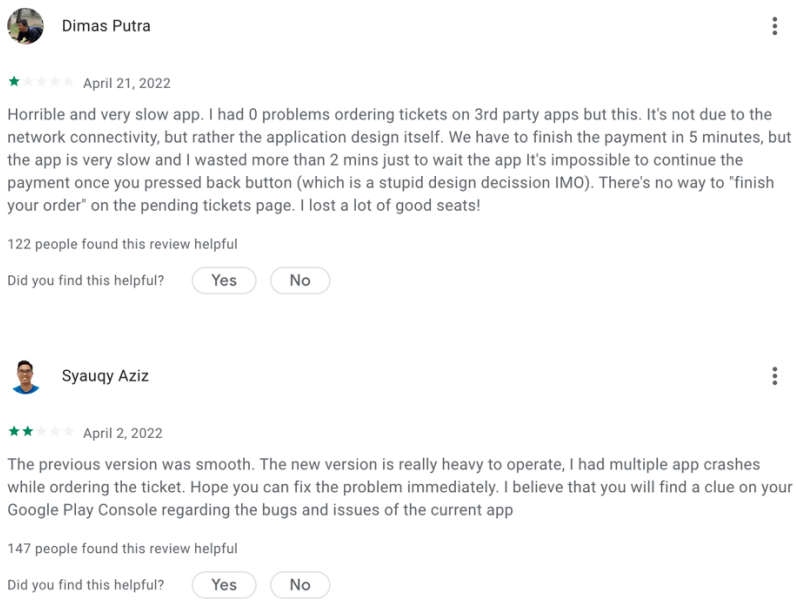


Gambar 3.2 Alur Pengerjaan Proyek KAI Access

3.2.1 Dataset

Data yang digunakan sebagai *input* model klasifikasi adalah data ulasan aplikasi KAI Access pada *platform* distribusi aplikasi Play Store yang dapat di akses pada https://play.google.com/store/apps/details?id=com.kai.kaiticketing. Data ulasan yang dapat

langsung diolah didapatkan dari tim KAI Access, internal perusahaan PT Kereta Api Indonesia. Data yang diberikan tersedia dalam bentuk berkas dengan format CSV dan data diberikan dengan bentuk beberapa pengelompokkan berkas yang berbeda sesuai dengan waktu (dalam bulan) diunggahnya data ulasan tersebut. Sehingga, jumlah baris data dalam masing-masing berkas CSV berbeda-beda, tergantung banyaknya jumlah ulasan yang diunggah pada bulan tersebut. Setiap berkas/bulan yang berbeda memiliki jumlah dan macam nama kolom yang sama yaitu terdapat kolom data teks ulasan dan kolom lainnya berupa detail atau data pendukung dari ulasan tersebut. Sampel data ulasan aplikasi KAI Access pada *platform* Play Store ditampilkan pada Gambar 3.3. Data ulasan yang ditampilkan pada Google Play Store adalah nama pengguna, waktu pengunggahan ulasan, *star* rating, dan teks ulasan. *Dataset* dibagi menjadi dua bagian, yaitu *training data* dan *testing data*.



Gambar 3.3 Contoh Ulasan Pada Laman Aplikasi KAI Access di Play Store

3.2.2 Data Preprocessing: Data Merging

Proses ini dilakukan pertama kali untuk memudahkan penulis melakukan *data labeling* karena *data labeling* lebih mudah dilakukan jika data sudah dikumpulkan menjadi satu kesatuan. Pada pengerjaan proyek ini, data mentah ulasan aplikasi KAI Access didapatkan dalam berkas yang terpisah sesuai dengan waktu bulan pengunggahan ulasan. *Data merging* dilakukan untuk menggabungkan beberapa sumber (berkas) data agar menjadi satu *dataset* besar yang mudah untuk diolah pada proses selanjutnya, *row* data akan bertambah seabnyak data yang akan digabungkan sedangkan kolomnya tidak berubah. Contoh sampel data ulasan

KAI Access dari berkas yang berbeda seperti pada Tabel 3.1, Tabel 3.2, Tabel 3.3 yang setiap tabelnya dari waktu bulan yang berbeda. Setelah dilakukan *data merging* menjadi satu berkas yang sama, hasil *dataset* seperti pada Tabel 3.4. Semua data pada semua waktu pengunggahan ulasan disatukan menjadi satu berkas.

Tabel 3.1 Sampel Data Ulasan KAI Access Bulan Januari 2021

Tanggal dan Waktu Ulasan	**Star Rating**	**Teks Ulasan**
2021-01-01T00:12:44Z	5	Kok bisa login ini gimana
2021-01-01T00:48:19Z	1	Pesanan hanya untuk pribadi mw pesan sekeluarga tidak bisa lebih dari 4

Tabel 3.2 Sampel Data Ulasan KAI Access Bulan Februari 2021

Tanggal dan Waktu Ulasan	**Star Rating**	**Teks Ulasan**
2021-02-02T04:33:46Z	4	Ini kenapa waktu mau pembayaran lewat Indomaret tapi nggak bisa kata Mbak nya error di tolok? Apa nya ini bermasalah? Padahal lagi butuh banget tiket
2021-02-02T04:53:23Z	1	Jadwal sering eror

Tabel 3.3 Sampel Data Ulasan KAI Access Bulan Maret 2021

Tanggal dan Waktu Ulasan	**Star Rating**	**Teks Ulasan**
2021-03-01T11:57:26Z	1	Mau daftar kq susah..ada tulisan email dan no.hp sudah terdaftar,aneh
2021-03-01T12:47:15Z	4	"TIDAK BISA GANTI NO TELPON/HP"

Tabel 3.4 Contoh Hasil Merging Data Ulasan KAI Access

Tanggal dan Waktu Ulasan	**Star Rating**	**Teks Ulasan**
2021-01-01T00:12:44Z	5	Kok bisa login ini gimana
2021-01-01T00:48:19Z	1	Pesanan hanya untuk pribadi mw pesan sekeluarga tidak bisa lebih dari 4
2021-01-01T01:07:39Z	2	mungkin bisa ditingkatkan agar pembayaran via link aja tidak harus nomor yg sama dg yg didaftarkan di kai access
2021-02-02T04:33:46Z	4	Ini kenapa waktu mau pembayaran lewat Indomaret tapi nggak bisa kata Mbak nya error di tolok? Apa nya ini bermasalah? Padahal lagi butuh banget tiket
2021-02-02T04:53:23Z	1	Jadwal sering eror
2021-03-01T11:57:26Z	1	Mau daftar kq susah..ada tulisan email dan no.hp sudah terdaftar,aneh

2021-03-01T12:47:15Z	4	"TIDAK BISA GANTI NO TELPON/HP"

3.2.3 Data Preprocessing: Data Cleaning

Data cleaning sama seperti *data merging*, dilakukan sebelum *data labeling*. Hal ini dikarenakan konteks data lebih dapat dipahami jika data dalam keadaan sudah bersih, tentunya memudahkan saat proses *data labeling*. Selain itu, urutan ini ditetapkan guna mengefisiensikan waktu karena terdapat beberapa baris data yang akan dihapus dalam *data cleaning* sehingga pada proses *data labeling*, penulis hanya melabeli data yang sudah pasti akan masuk sebagai *input* model.

Column Selection

Data ulasan aplikasi KAI Access memiliki banyak kolom sebagai informasi data ulasan. Tentunya, tidak memungkinkan semua kolom pada data dipakai sebagai *input* pada pelatihan model, karena tidak semua kolom relevan dengan pembelajaran model nantinya dan hanya akan menambah *noise*. Maka dari itu diperlukannya proses *column selection*, memilih kolom mana saja yang akan dipakai sebagai fitur *input* pelatihan model serta analisis data dan menyisihkan kolom lainnya. Selain untuk mempersingkat waktu pelatihan, *column selection* diperlukan untuk mengurangi kemungkinan *overfitting* karena *noise* yang disebabkan oleh data yang tidak relevan sehingga proses ini dapat meningkatkan akurasi yang diperoleh model karena mengurangi data yang bersifat *misleading*.

Filtering

Filtering adalah proses menghilangkan karakter-karakter yang tidak relevan pada konteks kalimat atau karakter-karakter yang berpotensi mengganggu proses pembelajaran model. Hal-hal yang dilakukan pada proses *filtering* ini yaitu seperti menghilangkan tanda baca, emoji, angka, teks yang terdiri dari satu huruf, spasi ganda, spasi pada awal kalimat, spasi garis baru, dan menghilangkan tautan yang terdapat pada data. Emoji dalam pemakaian umum memang menjadi karakter pendukung sentimen atau konteks sebuah teks, contoh umumnya emoji *thumbs up* menandakan teks tersebut memiliki sentimen yang baik dan sebaliknya emoji *thumbs down* identik dengan sentimen yang buruk. Namun setelah data ulasan KAI Access ditelaah, banyak pemakaian emoji yang tidak sesuai dengan konteks teks ulasannya. Ulasan sarkasme banyak didapatkan, contohnya banyak ulasan yang mengkritik buruk aplikasi KAI

Access namun menggunakan emoji *thumbs up*. Hal inilah yang menjadi alasan mengapa emoji dihapuskan dalam proses *filtering*.

Penghapusan *stopwords* pada proses *filtering* umum dilakukan pada pemrosesan data teks, namun pada pengerjaan proyek ini tidak dilakukan karena *stopwords* banyak membantu dalam pengklasifikasian sentimen, topik, dan detail topik sehingga jika *stopwords* dihapuskan, akan mengurangi akurasi dari model klasifikasi. Contoh kalimat dengan dan tanpa *stopwords* dijabarkan pada Tabel 3.5. Dengan kalimat pada tabel tersebut, terlihat pada Tabel 3.6 perbandingan hasil akurasi *testing data* dengan dan tanpa *stopwords*. *Testing* ini dilakukan menggunakan tiga *dataset* yang berbeda yaitu *dataset* A, *dataset* B, dan *dataset* C yang masing-masing memiliki 5 data di dalamnya.

Tabel 3.5 Contoh Kalimat Ulasan Dengan dan Tanpa Stopwords Data Prediksi A

	Kalimat 1	**Kalimat 2**	**Kalimat 3**	**Kalimat 4**	**Kalimat 5**
Dengan Stopwords	aplikasinya bagus sampai update terakhir tidak bisa melakukan apapun setelah memilih kereta aplikasinya terus crash saat waktunya masih berjalan jadi aku tidak pernah menyelesaikan pesanan saya saya bertanya tanya apa yang salah karena aplikasi saya yang lain berjalan dengan baik tolong perbaiki	proses mengubah nomor telepon saya sangat susah terkadang orang orang dapat kehilangan handphonenya dan harus mengganti nomornya segera	aplikasi ini berjalan sangat lambat saya tidak bisa menyelesaikan pemesanan tiket saya	selalu force close saat akan meproses pembayaran setelah memesan tiket tolong perbaiki	aplikasi ini sangat bagus saya pengguna yang sering memakai aplikasi ini semenjak saya sering naik kereta seperti 3 4 kali dalam seminggu
Tanpa Stopwords	aplikasinya bagus update apapun memilih kereta aplikasinya crash berjalan menyelesaikan pesanan salah aplikasi berjalan tolong perbaiki	proses mengubah nomor telepon susah terkadang orang orang kehilangan handphonenya mengganti nomornya	aplikasi berjalan lambat menyelesaikan pemesanan tiket	force close meproses pembayaran memesan tiket tolong perbaiki	aplikasi bagus pengguna memakai aplikasi semenjak kereta 3 4 kali seminggu

Tabel 3.6 Testing Perbandingan Keakuratan Model dengan dan Tanpa Stopwords

	Akurasi klasifikasi sentimen	**Akurasi klasifikasi topik**	**Akurasi klasifikasi**	**Frekuensi kesalahan klasifikasi**	**Frekuensi kesalahan klasifikasi**	**Frekuensi kesalahan klasifikasi**

			detail topik	pada Data A Saat Prediksi	pada Data B Saat Prediksi	pada Data C Saat Prediksi
Stopwords dihilangkan	99,82%	99,66%	98,79%	1/5	3/5	0
Dengan stopwords	99,90%	99,93%	98,51%	0	2/5	0

Handling Missing Values

Missing value adalah hal yang ditemukan hampir di setiap *dataset*. Tidak terkecuali pada proyek ini, ditemukan banyak *missing value* pada beberapa kolom tidak terkecuali pada kolom teks ulasan. *Missing value* yang ditemukan pada teks ulasan adalah hasil ulasan oleh pengguna yang hanya memberikan ulasan dengan *rating*, tidak disertai dengan teks ulasannya. Pengguna yang hanya memasukkan karakter tertentu seperti emoji pada teks ulasannya juga akan menjadi *missing value* karena pada proses *filtering*, karakter emoji dihapuskan. Dikarenakan proyek ini berfokus pada pengolahan data teks untuk klasifikasi ulasan, data yang memiliki teks ulasan dengan *missing values* dihapuskan dari *dataset.* Pada data yang memiliki *missing values* di fitur lain selain teks ulasan, data dengan *missing values* tersebut hanya diganti dengan kata 'unknown' untuk keperluan analisis data. Hal ini dikarenakan teks dengan *missing values* tidak akan memberikan manfaat terhadap pembelajaran dari model.

Case Folding

Case folding adalah proses mengubah seluruh huruf abjad pada teks menjadi seragam yaitu dengan mengubahnya menjadi huruf kecil. Proses ini diperlukan karena pada proses selanjutnya, proses *word tokenization*, perbedaan kecil seperti perbedaan *case* pada kata yang sama akan dianggap sebagai dua token yang berbeda. Misalnya dua kata yang sama namun memiliki perbedaan *case* pada salah satu hurufnya (contoh: 'data' dan 'Data', satu menggunakan huruf D kecil dan satu dengan huruf D kapital), pada *word tokenization* kedua kata ini akan memiliki token yang berbeda karena *case*-nya berbeda walaupun makna katanya sama. Hal ini akan menyebabkan berkurangnya akurasi pada model.

3.2.4 Data Labeling

Data Labeling diperlukan pada klasifikasi ini untuk membantu model pada setiap klasifikasi label untuk belajar mengenali target dari setiap data. Pada tahap ini setiap data ulasan aplikasi KAI Access diberi tiga label kelas yaitu sentimen, topik, dan detail topik untuk

memudahkan identifikasi permasalahan. Seluruh label dan kelas di dalamnya, jumlah dan macamnya ditentukan oleh penulis bersama mentor sebagai klien dan disesuaikan dengan kebutuhan perusahaan. *Data labeling* dilakukan secara manual oleh penulis, satu persatu. Jika ada konteks teks ulasan yang ambigu saat penulis melakukan *data labeling*, maka akan didiskusikan bersama mentor dan ditentukan bersama kelas dari data ambigu tersebut. Sembari melabeli data dengan label sentimen, topik, dan detail topik, dilakukan juga pembetulan kata yang berupa singkatan dan kesalahan pengetikan (*typo*) pada teks ulasan. Kata akronim juga akan diuraikan untuk membantu model memahami konteks data.

Label Sentimen

Label sentimen memiliki tiga kelas yaitu kelas Positif, Negatif, dan Netral. Sebagaimana hasil permintaan dan diskusi bersama klien, parameter dalam menentukan sentimen pada data ulasan adalah dengan menelaah pada kalimat di dalam datanya. Contoh pengelompokkan data pada setiap kelas sentimen ada pada Tabel 3.7. Sebuah data dikategorikan memiliki sentimen positif jika data menunjukkan kepuasan pada pelayanan pada aplikasi KAI Access. Selain itu, dikategorikan kelas Positif jika pada data ditemukan sebuah pujian dan kata-kata positif lainnya di luar konteks pelayanan KAI Access. Data ulasan dikategorikan pada kelas Negatif jika data ulasan menunjukkan ketidakpuasan/protes pengguna terhadap pelayanan KAI Access ataupun kalimat yang menyebutkan kata bersifat negatif maupun umpatan kasar. Sedangkan data yang dikategorikan pada kelas Netral tidak memiliki ciri pada kelas Positif maupun Negatif atau berada diantara keduanya tanpa menyebutkan kata-kata bersifat positif maupun negatif. Penentuan sentimen ini juga didukung dengan referensi penentuan anotasi sentimen berdasarkan *semantic-role based sentiment* yang ditulis pada penelitian di Canada (Mohammad, 2016).

Tabel 3.7 Contoh Pengelompokkan Data Ulasan KAI Access pada Label Sentimen

Label	Data
Positif	praktis dan mudah hemat waktu
	aku kasih bintang karena memang aplikasinya good
Netral	entah lah
	semoga covid ini cepat berlalu kangen naik ka
Negatif	lelet susah diklik balik ke awal mulu ngelag mending aplikasi yang dulu abis upgrade malah lelet
	aplikasi sangat jelek ancur bagus beli di shopee bukalapak

Label Topik

Kelas topik mencakup kategori pelayanan atau permasalahan yang dapat digali dari aplikasi KAI Access, sehingga hasil dari label ini dapat membantu pengembang dalam mengidentifikasi fitur dari aplikasi yang harus dievaluasi. Dari hasil kesepakatan bersama mentor, didapatkanlah tujuh kelas dalam label topik yaitu kelas Pemesanan, Pembayaran, Pembatalan, Registrasi-Login, Pengaturan, Error, dan Feedback. Contoh pengelompokkan topik data ada pada Tabel 3.8. Parameter data dapat dikategorikan pada kelas Pemesanan, Pembayaran, Pembatalan, Registrasi-Login, atau Pengaturan adalah dari konteks keseluruhan data ulasan, topik apa yang dibahas pada data ulasan tersebut yang termasuk dalam salah satu topik kelas tersebut. Sedangkan untuk kelas Error berisi ulasan yang mengeluhkan permasalahan pada aplikasi dengan topik selain pada kelas Pemesanan, Pembayaran, Pembatalan, Registrasi-Login, dan Pengaturan. Kelas Feedback yang berisi data ulasan yang hanya memberikan *feedback* buruk tanpa memberikan detail keluhannya atau keluhan terdapat diluar aplikasi KAI Access dan *feedback* baik yang diberikan pengguna untuk aplikasi KAI Access maupun diluar aplikasi KAI Access.

Tabel 3.8 Contoh Pengelompokkan Data Ulasan KAI Access pada Label Topik

Label	Data
Pemesanan	kok tampilan jadwalnya berbayang bayang gini ya
	ini kenapa ya kok tidak bisa pesan tiket kereta dan tidak bisa di buka tanggalnya tidak ada jadwalnya mau naik kereta aja susah banget
Pembayaran	mohon di tingkatkan lagi untuk pembayaran via debit selain bank mandiri dong misal bank swasta bca terima kasih
	aplikasinya error saya pesan tiket orang ketika mau membayar malah jadi kali lipat
Pembatalan	upgrade terus tapi menu untuk pembatalan tiket lokal tidak ada apa karena murah jadi harus ke stasiun ribet amat
	kenapa tiket tidak bisa di batalkan
Registrasi-Login	daftar akun aja error heran
	pas login lagi tulisannya email no hp kata sandi salah padahal tidak pernah di ubah males ngurusnya lagi
Pengaturan	kok tidak bisa ubah profil ganti nomor hp sih
	saya ingin merubah info akun saya tapi tidak bisa sangat buruk
Error	mohon di bantu kok sulit di buka akhir akhir ini
	setiap buka aplikasinya selalu yang muncul cuma layar putih mohon segera diperbaiki demi kenyamanan pelanggan
Feedback	bagus pas abis diupdate
	aku kasih bintang aja soal ya banyak yang komplain tapi tidak di respon

Label Detail Topik

Label detail topik yaitu topik yang lebih mendalam dari setiap kelas yang terdapat pada label topik dibuat untuk identifikasi permasalahan yang lebih spesifik dalam pelayanan KAI

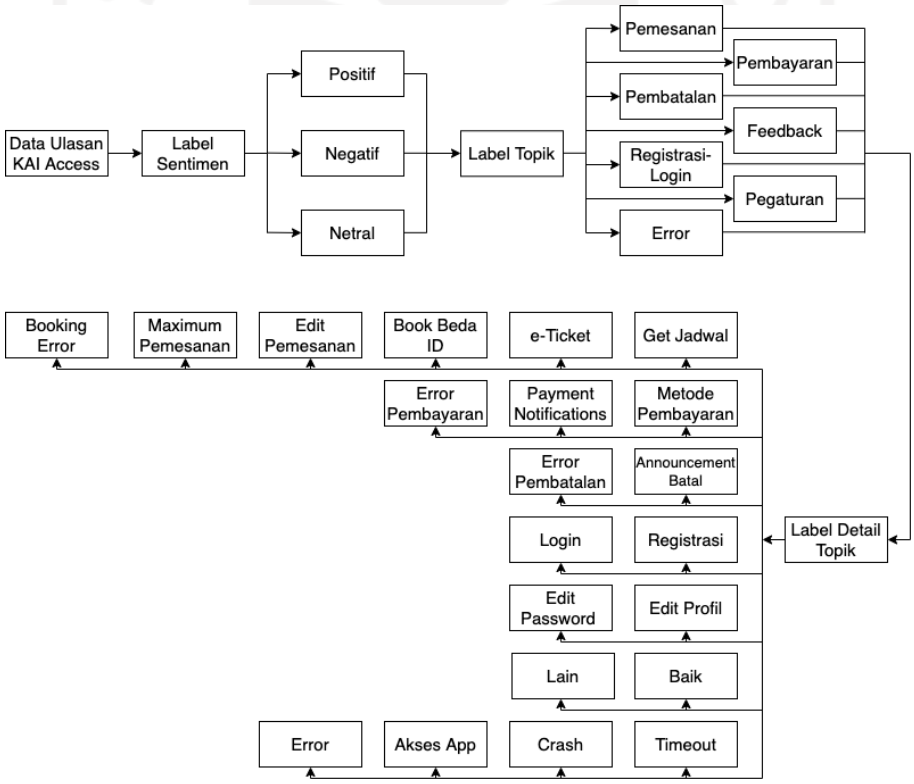
Access. Hasil diskusi penulis dan mentor menghasilkan kesepakatan bahwa label detail topik memiliki 21 kelas berdasarkan turunan label topik, yaitu kelas Pemesanan mempunyai detail topik Get Jadwal, Edit Pemesanan, Booking Berbeda ID, Maximum Pemesanan, e-Ticket, dan Error Pemesanan. Pada kelas Pembayaran memiliki detail kelas yaitu Metode Pembayaran, Payment Notification, dan Error Pembayaran. Kelas Pembatalan memiliki detail topik Error Pembatalan dan *Announcement* Pembatalan. Kelas Regitrasi-Login memiliki dua detail topik, yaitu Registrasi dan Login. Detail topik pada kelas Pengaturan yaitu Edit Password dan Edit Profil. Topik Error memiliki empat detail topik yaitu Timeout, Akses App, Crash, dan Error. Kelas topik terakhir yaitu Feedback memiliki detail topik baik dan lain. *Feedback* baik yang diberikan pada pengguna hanya akan masuk ke satu kelas detail topik yaitu kelas detail topik Baik dan tidak banyak bercabang lagi dikarenakan permintaan dari mentor yang berlaku juga sebagai klien. Mentor meminta untuk tidak terlalu menspesifikasikan ulasan dengan *feedback* positif. Hal ini dikarenakan klien ingin klasifikasi ulasan ini difokuskan untuk evaluasi kekurangan aplikasi, sehingga yang perlu didetailkan hanyalah *feedback* negatif. Secara keseluruhan kelas detail topik memiliki jumlah 21 label. Parameter data dapat dikategorikan pada label detail topik tertentu adalah data tersebut sebelumnya memiliki kelas topik apa serta konteks dari keseluruhan data ulasan, detail topik apa yang dibahas pada data ulasan tersebut. Contoh pengelompokkan detail topik data ada pada Tabel 3.9.

Tabel 3.9 Contoh Pengelompokkan Data Ulasan KAI Access pada Label Detail Topik

Topik	Label	Data
Pemesanan	Get Jadwal	mohon maaf teruntuk kai access ini kenapa ya kok saya nggak bisa pesan tiket kereta ka lokal kok nggak ada jadwal semuanya mohon dijawab admin
	Edit Pemesanan	untuk tiket kereta api lokal tidak bisa direschedule jadwal yang bisa cuma tiket kereta api non lokal antar kota perbaiki dong
	Book Beda ID	kalau mau beli tiket lagi penumpang pertama gabisa diubah tolong diperbaiki terima kasih
	Maximum Pemesanan	oh iya tambahin juga anggota maksimalnya dong paling tidak penumpang gitu hehe
	e-Ticket	setelah aplikasi saya diupdate tiket kereta yang terbeli malah hilang semua mohon dicek kembali dan segera diatasi masalah seperti ini
	Error Pemesanan	pesan tiket kok tidak bisa semua
Registrasi-Login	Registrasi	konfirmasi email tidak bekerja dengan baik
	Login	aplikasi nya tidak jelas wong email kata sandi nya bener tetap aja tidak bisa login payah
Pembayaran	Error Pembayaran	aplikasi banyak trouble tidak bisa dibayar tiketnya server down verifikasi tidak dikirim kirim
	Metode Bayar	pembayarannya kok tidak bisa via indomaret dan alfamart ya sekarang

	Payment Notifications	saya memesan tiket melalui aplikasi ini dan melakukan pembayaran menggunakan shopeepay tetapi tidak mendapat kode booking
Pembatalan	Announcement Batal	pembatalan kereta tidak ada pemberitahuan sama sekali
	Error Pembatalan	lebih praktis lagi kalau pembatalan tiket juga bisa dilakukan di aplikasi ini terutama tiket kereta lokal
Pengaturan	Edit Password	tidak bisa reset password
	Edit Profil	ganti no hp akun kok susahnya minta ampun apa harus telpon call center pt kai mohon solusinya
Feedback	Baik	sangat membantu
	Lain	saya pesan dari pasar senen ke lempuyangan tanggal mei
Error	Timeout	permintaan anda time out terus maksudnya gimana ya baru juga install
	Crash	pas dibuka warnanya putih semua jadi harus pinjem hp lain buat buka padahal sizenya termasuk kecil kalau dibandingkan dengan aplikasi olshop
	Akses App	kenapa aplikasi nya tidak bisa dibuka
	Error	agak lama load nya

Sehingga secara keseluruhan, proses pelabelan data ulasan KAI Access adalah sebagaimana diilustrasikan pada Gambar 3.4.



Gambar 3.4 Pengelompokkan Kelas Data KAI Access

3.2.5 Data Preprocessing: Data Splitting

Data splitting adalah proses membagi data menjadi bagian-bagian tertentu. Pada proyek ini, *data splitting* dilakukan untuk membagi data ulasan KAI Access ini menjadi dua bagian, yaitu *training data* dan *testing data*. *Training data* berisi data yang memiliki waktu unggah ulasan pada bulan Januari - September 2021, dan *testing data* berisi data dengan waktu unggah ulasan bulan Oktober 2021 – April 2022. Selain itu, *data splitting* juga dilakukan untuk memisahkan kolom data yang akan digunakan sebagai *input* yang akan dilatih serta fitur yang akan digunakan sebagai label/target data.

3.2.6 Data Preprocessing: Data Tokenization

Data tokenization adalah proses memisah setiap data ulasan aplikasi KAI Access menjadi kata individual yang disebut token, karena model hanya bisa memproses data dalam bentuk angka. Setiap token akan ditransformasikan menjadi token angka dimana setiap angka akan mewakilkan satu kata pada data yang sudah ditokenisasi. Maka, model akan memproses setiap token sebagai angka dan menyimpan sebuah struktur data *dictionary* berisi angka sebagai *key* dan kata sebagai *value*. Tokenisasi ini membantu model memahami konteks dan menginterpretasikan arti dengan menganalisa sekuen dari kata-kata. Level *data tokenization* yang digunakan dalam proyek ini yaitu *word tokenization*, membagi teks menjadi individual-individual kata.

Setelah seluruh kata memiliki token masing-masing, setiap data diubah menjadi *sequence* token berdasarkan *vocabulary* yang telah dibuat. *Sequence* berisi barisan token berdasarkan susunan kata data yang terdaftar pada *vocabulary*. *Sequence* setiap data memiliki panjang yang berbeda-beda sesuai dengan jumlah kata pada masing-masing data. Namun, model hanya akan menerima *input* dengan panjang yang sama pada setiap datanya. Maka dari itu, dilakukanlah *padding* yaitu mengubah setiap *sequence* data sehingga memiliki panjang yang sama. *Padding* akan mengisi *sequence* yang panjangnya kurang dari *maximum length* dengan token *default* [0]. Data *input* yang sudah dilakukan *padding* inilah yang akan menjadi *input* model.

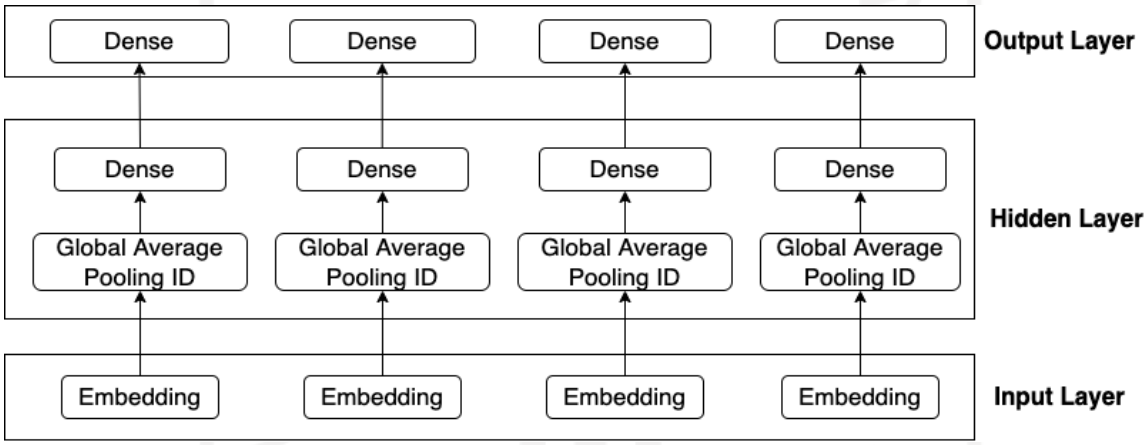
3.2.7 Data Modeling

Data ulasan aplikasi KAI Access yang sudah bersih kemudian akan dijadikan sebagai *input* model yang akan dilatih. Pada pengerjaan proyek ini, *output* yang diharapkan adalah hasil label dari tiga klasifikasi yaitu sentimen, topik, dan detail topik. Oleh karena itu, pada proses

ini dibuat tiga model yang berbeda untuk masing-masing klasifikasi dengan arsitektur yang berbeda-beda.

Model Klasifikasi Sentimen

Model klasifikasi sentimen data ulasan KAI Access ini menggunakan arsitektur Multi-Layer Perceptron (MLP) yang memiliki *input layer*, *output layer*, dan sebuah *hidden layer* seperti yang diperlihatkan pada Gambar 3.5. Embedding Layer sebagai *input layer*, Global Average Pooling 1D dan Dense Layer sebagai *hidden layer*, dan Dense Layer sebagai *output layer*. Referensi susunan *layer* ini didapatkan dari *website* TensorFlow *basic classification* (TensorFlow, Basic text classification, 2022).



Gambar 3.5 Bagan Arsitektur Model Klasifikasi Sentimen Data Ulasan KAI Access

Layer pertama atau *input layer* pada klasifikasi sentimen ini adalah Embedding Layer yang mengubah data *input* menjadi vektor. Embedding *Layer* memiliki tiga parameter yaitu pertama adalah parameter *input dimension* yang diisikan jumlah *vocabulary size* hasil *word tokenization*. Parameter kedua adalah jumlah *ouput dimension* yang diinginkan. Parameter terakhir pada Embedding Layer adalah *input length* yang diisi dengan variabel *maximum length* atau panjang dari *input sequences* yang dimasukkan ke dalam *layer*. Parameter *input dimension* diset dengan angka 4.100 karena jumlah *vocabulary* yang didapatkan dari *data tokenization* mendekati jumlah 4.100 (dibulatkan ke atas sampai angka ratusan). Parameter *output dimension* diset dengan angka 16 karena dengan 16 dimensi cukup untuk memproses data yang tidak terlalu besar dan biasanya jumlah dimensi adalah kelipatan delapan (TensorFlow, TensorFlow Word Embeddings, 2022). *Maximum length* pada *training data* ulasan KAI Access adalah 150, sehingga parameter *input length* klasifikasi sentimen didefinisikan dengan 150.

Hasil dari Embedding Layer akan menghasilkan *array* dua dimensi dimana ukuran barisnya sesuai dengan *input length* dan ukuran kolomnya sesuai dengan *output dimension*. Sehingga, untuk dapat dipelajari oleh model, Global Average Pooling 1D Layer digunakan pada *layer* pertama *hidden layer* untuk meratakan data *input* menjadi satu dimensi. Global Average Pooling 1D Layer meratakan *input* yang berupa 2 dimensi dengan cara mengambil nilai rata-rata dari *input* tersebut, dengan kata lain melakukan reduksi data dengan mengecilkan ukurannya (*down sampling*) tanpa menghilangkan maknanya (Peltarion, 2022). Oleh karena itu, *output shape* yang dihasilkan *pooling layer* ini lebih kecil sehingga dapat bekerja lebih cepat dalam memproses data.

Data yang sudah berukuran satu dimensi kemudian dimasukkan ke dalam Dense Layer untuk menambahkan *layer fully connected* atau *layer* yang terhubung dengan seluruh neuron, sebelum dimasukkan ke dalam *output layer*. Dense Layer memiliki dua parameter yaitu *units* dan *activation*. Angka pada parameter unit ini ditentukan dengan percobaan pada beberapa pilihan jumlah unit Dense Layer yaitu 16, 32, dan 64 unit untuk dicari hasil yang paling baik. *Activation* yang dipilih untuk Dense Layer ini yaitu aktivasi Rectified Linear Unit (ReLu), aktivasi yang hanya mengembalikan angka positif. Aktivasi ini dipilih karena dapat mempercepat proses konvergensi jika dibandingkan dengan aktivasi Tanh atau Sigmoid karena ReLu menonaktifkan neuron jika *output* dari transformasi linear kurang dari 0 (Gupta, 2020). Oleh karena itu, aktivasi ini bagus untuk digabungkan dengan teknik lainnya seperti Dropout. Aktivasi ReLu juga populer digunakan pada pemrosesan teks.

Dense Layer pada *output layer* digunakan untuk mengklasifikasikan data sesuai dengan kelasnya yang sudah ditentukan. Dense Layer ini juga mempunyai dua parameter yaitu *units* dan *activation*. Dikarenakan *layer* ini digunakan untuk klasifikasi, maka jumlah *units* yang digunakan adalah jumlah kelas pada klasifikasi yaitu 4 unit (jumlah kelas dihitung dari kelas 0, namun label kelas pada hasil *word tokenization* untuk klasifikasi sentimen ini dimulai dari 1 sehingga dianggap 4 kelas yaitu kelas 0, 1, 2, dan 3). Aktivasi yang dipilih yaitu Softmax, aktivasi yang digunakan untuk pengklasifikasian lebih dari dua kelas (*non-boolean*).

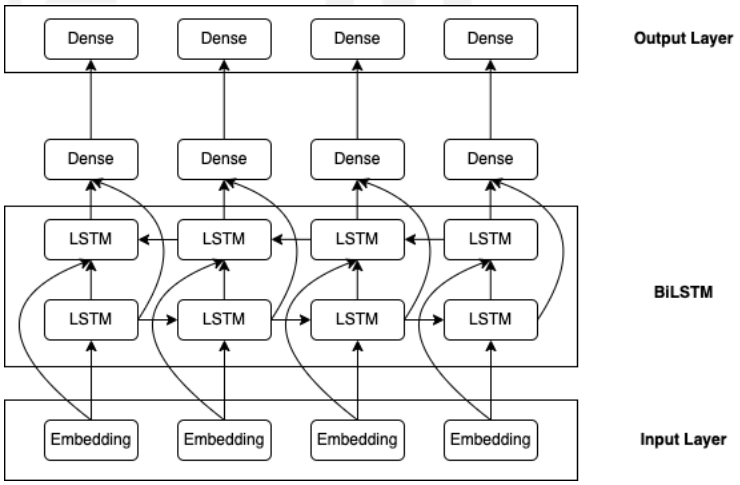
Fungsi yang digunakan pada arsitektur ini antara lain *optimizer* Adam, *optimizer* yang menerapkan algoritma penurunan gradien stokastik yang didasarkan pada estimasi adaptif (TensorFlow, Adam Optimizer, 2022). *Optimizer* ini dipilih selain karena kepopulerannya pada pembuatan model Machine Learning, *optimizer* ini secara komputasi sangat efisien menghabiskan hanya sedikit memori, dan cocok untuk data pada skala besar seperti data pada proyek ini (Kingma & Ba, 2014). Selain itu terdapat fungsi *loss* yaitu Sparse Categorical

Crossentropy, dipakainya *loss* ini karena klasifikasi yang dilakukan adalah *multi-class* (target lebih dari satu kelas). Pemilihan Sparse Categorical Crossentropy daripada Categorical Crossentropy sebagai fungsi *loss* karena data label pada proyek ini yang telah dikonversi ke dalam *array* berisi angka dalam bentuk *integer* (contoh: [1], [2], [3]) bukan hasil dari proses *encoding* (contoh: [1, 0, 0], [0, 1, 0], [0, 0, 1]).

Kemudian, model dilatih dengan jumlah *epoch* yang menghasilkan akurasi *training* yang baik. Dilakukan percobaan dengan beberapa jumlah *epoch* untuk klasifikasi sentimen ini, yaitu 30, 40, 50, dan 60 *epoch*.

Model Klasifikasi Topik

Berbeda dengan model klasifikasi sentimen, model klasifikasi topik menggunakan arsitektur model BiLSTM. Seperti yang telah dipaparkan pada Bab 2, BiLSTM dipilih karena terkenal baik dalam pengklasifikasian data teks dan mampu mempelajari data dengan lebih baik daripada LSTM. Terlebih, klasifikasi topik lebih sulit dilakukan dibandingkan dengan klasifikasi sentimen karena kelas pada label topik lebih banyak. Pada klasifikasi ini, selain teks ulasan, klasifikasi topik juga mengikutsertakan fitur sentimen sebagai *input* modelnya. Arsitektur model klasifikasi topik ini diperlihatkan pada Gambar 3.6. Embedding Layer sebagai *input layer*, BiLSTM dan Dense Layer sebagai *hidden layer*, dan Dense Layer sebagai *output layer*. Referensi susunan *layer* yang membentuk arsitektur ini didapatkan dari *website* TensorFlow RNN (TensorFlow, Text Classification With an RNN, 2022).



Gambar 3.6 Bagan Arsitektur Model Klasifikasi Topik dan Detail Topik Data Ulasan KAI Access

Model klasifikasi topik dengan BiLSTM ini memiliki Embedding Layer sebagai *input layer* dan Dense Layer sebagai *output layer*. Serupa dengan klasifikasi sentimen, Embedding Layer dan parameternya digunakan untuk mengelompokkan teks ke dalam beberapa vektor. Parameter *input length* pada Embedding Layer ini sama seperti pada klasifikasi sentimen yaitu mengambil *maximum length* pada data *input*, namun ditambah 1 jumlahnya (*maximum length*+1). Angka 1 ini adalah token label sentimen yang digabungkan dengan data teks sebagai *input* klasifikasi topik ini, sehingga *input length* data *input* lebih panjang 1 token dibandingkan dengan *input length* pada klasifikasi sentimen. Oleh karena itu, pada klasifikasi ini parameter *input length* pada Embedding Layer didefiniskan dengan angka 151. Untuk parameter lainnya pada Embedding Layer ini yaitu *input dimension* dan *output dimension*, didefinisikan sama seperti yang ada pada klasifikasi sentimen yaitu 4.100 untuk *input dimension* dan 16 untuk *output dimension*.

Setelah *input* dimasukkan ke dalam Embedding Layer, data masuk ke BiLSTM Layer untuk diproses secara dua arah: *forward* dan *backward* yang diperlihatkan oleh arah panah ke kanan dan kiri pada Gambar 3.6. Terdapat parameter pada BiLSTM Layer yaitu parameter *units*. Dilakukan beberapa percobaan untuk menentukan jumlah unit pada BiSTM Layer, diantaranya 16, 32, dan 64 unit. Masih pada *hidden layer*, data kemudian diproses pada Dense Layer. Dense Layer ini memiliki dua parameter, yaitu unit dan aktivasi. Pada Dense Layer ini, jumlah parameter unit ditentukan berdasarkan percobaan beberapa pilihan unit yaitu 16 dan 32. Parameter aktivasi yang digunakan pada Dense Layer ini adalah aktivasi ReLu. Terakhir, Dense Layer sebagai *output layer* dengan jumlah unit sebesar 8, sesuai dengan jumlah kelas yang akan diklasifikasikan dan digunakan aktivasi Softmax untuk mengklasifikasikan lebih dari satu kelas.

Tidak berbeda dengan klasifikasi sentimen, parameter fungsi *compile* yang digunakan pada model klasifikasi topik ini adalah Adam sebagai *optimizer* dan Sparse Categorical Crossentropy sebagai *loss* dengan alasan yang sama. Kemudian, model di latih dengan jumlah *epoch* yang lebih besar dari jumlah *epoch* klasifikasi sentimen karena klasifikasinya lebih kompleks daripada klasifikasi sentimen dan arsitektur yang dipakai juga berbeda. Jumlah *epoch* pada klasifikasi ini berjumlah 60 *epoch.* Selain karena klasifikasi yang lebih kompleks, dipilih jumlah 60 *epoch* karena jumlah Dense *unit* 16 dan 32 pada *epoch* ke-60 sudah cukup menghasilkan akurasi yang tinggi dan *loss* yang baik, sehingga tidak mencoba *epoch* yang lebih tinggi lagi. Selain itu, akurasi *training* pada jumlah *epoch* dibawah 50 masih belum stabil.

Model Klasifikasi Detail Topik

Serupa dengan klasifikasi topik, klasifikasi detail topik juga menggunakan arsitektur BiLSTM untuk membangun modelnya. Jumlah *output* kelas pada klasifikasi detail topik lebih banyak daripada klasifikasi topik, hal ini menjadikan klasifikasi detail topik lebih kompleks untuk diklasifikasikan dan cocok untuk dibuat menggunakan arsitektur BiLSTM karena BiLSTM dapat mengenali konteks dari data dengan lebih akurat. Selain teks ulasan, model klasifikasi detail topik ini juga mengikutsertakan label sentimen dan label topik sebagai *input* modelnya untuk mendukung akurasi klasifikasi ini. Arsitektur model ini diperlihatkan pada Gambar 3.6, arsitektur yang susunan *layer*nya serupa dengan susunan *layer* pada model klasifikasi topik.

Hal yang membedakan arsitektur model klasifikasi topik dan klasifikasi detail topik adalah jumlah unit parameter *input length* pada Embedding Layer. Terdapat selisih jumlah unit sebesar 1, dikarenakan pada klasifikasi detail topik panjang *input* bertambah satu karena label topik masuk sebagai *input* model. Selain itu, jumlah pada parameter unit di *hidden layer* – Dense Layer berbeda. Jumlah unit ini ditentukan dengan melakukan percobaan menggunakan tiga pilihan yaitu 16, 32, dan 64 unit. Pada *output layer* tentu saja juga berbeda dengan *output layer* pada klasifikasi topik, *ouput layer* klasifikasi detail topik memiliki jumlah unit sebanyak jumlah kelas label detail topik yaitu 22 unit.

Model ini di*compile* masih dengan parameter yang sama dengan klasifikasi sentimen dan topik yaitu Adam *optimizer*, Sparse Categorical Crossentropy *loss*. Model ini di*training* dengan jumlah yang serupa dengan klasifikasi topik. Jumlah *epoch* ditentukan dengan jumlah *epoch* yang memiliki akurasi *training* yang baik. Jumlah *epoch* untuk klasifikasi detail topik ini adalah 60 *epoch* dikarenakan jumlah *epoch* ini cukup untuk membuat akurasi *training* yang tinggi.

3.2.8 Evaluasi Model

Sebelum memasuki evaluasi model, *testing data* diklasifikasikan terlebih dahulu dengan model yang sudah di*training* pada proses *data modelling*. *Testing data* yang sudah dilabeli oleh model atau sudah mempunyai *predicted label* ini lah yang kemudian akan menjadi bahan evaluasi model dengan cara membandingkannya dengan *testing data* yang telah dilabeli manual oleh penulis atau *actual label*.

Evaluasi dilakukan untuk mengetahui seberapa baik performa model dari nilai-nilai *metrics* yang dihasilkan seperti akurasi serta nilai *recall*, presisi, dan *f1-score* dari masing-

masing kelas. Parameter umum dalam mengukur kualitas performa model adalah dengan *metric* akurasi, semakin tinggi akurasinya maka semakin baik performanya.

3.2.9 Prototype Klasifikasi Data Ulasan KAI Access

Prototype Klasifikasi Data Ulasan KAI Access ini adalah *prototype* sistem klasifikasi otomatis data ulasan KAI Access yang dibuat dengan model Machine Learning yang sudah di*training* sebelumnya. *Input prototype* ini adalah data mentah yang ingin diklasifikasikan (namun hanya kolom teks ulasan yang dipakai untuk klasifikasi) dan memiliki *output* kolom baru yaitu label sentimen, topik, dan detail topik pada tabel data mentah yang sama dengan yang dimasukkan seperti yang dijelaskan pada Tabel 3.10. *Prototype* ini ditujukan agar penggunanya dapat memasukkan berkas berisi data ulasan mentah, kemudian *prototype* yang berupa bentuk *code* ini di*running* untuk mengklasifikasikan data secara langsung dan cepat untuk memberikan label sentimen, topik, dan detail topik untuk masing-masing data ulasan sebagai keluarannya. Sehingga, pengguna dapat menggunakan data ulasan KAI Access yang sudah dilabeli sentimen, topik, dan detail topik untuk kebutuhannya, salah satunya untuk kebutuhan *development* aplikasi KAI Access.

Tabel 3.10 Data Input dan Ouput Prototype Data Ulasan KAI Access

Kolom Data Input Prototype	**Kolom Data Ouput Prototype**
Package Name	Package Name
App Version Code	App Version Code
App Version Name	App Version Name
Reviewer Language	Reviewer Language
Device	Device
Review Submit Date and Time	Review Submit Date and Time
Review Submit Millis Since Epoch	Review Submit Millis Since Epoch
Review Last Update Date and Time	Review Last Update Date and Time
Review Last Update Millis Since Epoch	Review Last Update Millis Since Epoch
Star Rating	Star Rating
Review Title	Review Title
Review Text	Review Text
Developer Reply Date and Time	Developer Reply Date and Time
Developer Reply Millis Since Epoch	Developer Reply Millis Since Epoch
Developer Reply Text	Developer Reply Text
Review Link	Review Link
	Sentimen
	Topik
	Detail Topik

Proses pembuatan *prototype* ini diawali dengan mengimpor *training data*, data yang akan diklasifikasikan (data mentah), tiga *trained model* yang sudah disimpan dalam format .h5, serta mengimpor *libraries* yang akan dipakai. Data mentah yang sudah diimpor kemudian dibersihkan untuk menghasilkan hasil klasifikasi yang lebih baik, lalu model yang sudah disimpan kemudian digunakan untuk mengklasifikasikan data mentah. Klasifikasi dibuat bertahap mulai dari klasifikasi sentimen, lalu *output* klasifikasi sentimen digabungkan dengan teks ulasan untuk menjadi *input* klasifikasi topik, *output* klasifikasi topik kemudian digabungkan dengan teks ulasan dan label sentimen untuk selanjutnya dijadikan *input* klasifikasi detail topik. Setelah selesai klasifikasi detail topik, data mentah akan mempunyai tiga kolom baru yaitu kolom label sentimen, topik, dan detail topik yang berisi hasil klasifikasi masing-masing label. Data yang dihasilkan dari *prototype* inilah yang berguna untuk kebutuhan pengguna.

Cara kerja pemakaian *prototype* oleh pengguna adalah cukup jalankan *code prototype* pada IDE Python, lalu masukkan lokasi data mentah yang ingin diklasifikasikan, dan pengguna hanya perlu menunggu hingga proses *running prototype* selesai kemudian data mentah yang sudah dimasukkan oleh pengguna akan mempunyai label sentimen, topik, dan detail topik.

3.2.10 Dashboard Analisis Data Ulasan KAI Access

Dashboard visualisasi analisis ini adalah *dashboard* yang menampilkan berbagai macam *insight* yang didapatkan penulis dari data ulasan KAI Access yang memiliki banyak fitur. *Training* dan *testing data* ulasan KAI Access yang telah digabungkan dan dilabeli manual oleh penulis menjadi bahan analisis yang kemudian visualisasikan dan ditampilkan pada *dashboard*. *Insight* yang digali untuk ditampilkan pada *dashboard* antara lain jumlah data ulasan setiap bulan, 5 versi aplikasi yang paling banyak digunakan, 5 versi aplikasi yang memiliki rata-rata *rating* paling tinggi, rata-rata *rating* pada setiap bulan, jumlah data pada setiap *rating*, dan jumlah data setiap kelas pada setiap label sentimen, topik, serta detail topik. Dengan berbagai fitur pada data yang dapat digunakan untuk analisis, dibuatlah berbagai visualisasi dari setiap *insight* yang ingin ditampilkan.

Setiap visualisasi memiliki satu *insight* atau hasil analisis yang bisa didapatkan dari satu atau lebih fitur yang ada pada data ulasan KAI Access. *Insight* divisualisasikan dalam berbagai *chart* atau tabel. *Chart* dipilih menyesuaikan dengan *insight* yang ingin ditampilkan karena *chart* harus memvisualisasikan *insight* dengan baik. Dengan *insight* yang sudah ditentukan

sebelumnya, dipilih empat jenis *chart* dan sebuah tabel untuk visualisasi delapan *insight* yang berbeda. *Insight* Jumlah data ulasan perbulan divisualisasikan dengan *line chart*, *insight* 5 versi aplikasi yang paling banyak dipakai divisualisasikan dengan *bar chart*, 5 versi aplikasi dengan rata-rata *rating* tertinggi dengan *bar chart*, rata-rata *rating* perbulan dengan *line chart*, jumlah data pada setiap *rating* denga menggunakan *treemaps chart*, jumlah data pada setiap kelas di label sentimen dengan *donut chart*, jumlah data pada setiap kelas di label topik dengan *bar chart*, dan *insight* jumlah data pada setiap kelas di label detail topik dengan tabel.

Selain pemilihan bentuk visualisasi, warna pada *chart* juga dipilih dengan pertimbangan untuk dapat mendukung visualisasi yang lebih baik. Visualisasi-visualisasi yang telah dibuat kemudian dikumpulkan dan dimuat pada sebuah *dashboard* yang diberi judul dan keterangan, tak lupa kumpulan visualisasi disusun dengan apik pada *dashboard* ini. *Filter* juga dibuat pada *dashboard* yang berguna untuk pengguna yang ingin melakukan *filtering data* sesuai dengan bulan diunggahnya ulasan. Tidak lupa pada bagian teratas *dashboard*, diberikan judul *dashboard* dan keterangan data seperti jumlah data keseluruhan.

3.3 Implementasi dan Hasil

3.3.1 Dataset

Dari data yang sudah diberikan dari tim internal perusahaan PT KAI, didapatkan 16 berkas berdasarkan waktu bulan penggunggahan ulasan berurut dari bulan Januari tahun 2021 hingga pertengahan bulan April tahun 2022. Pada rentang waktu tersebut, dikumpulkan data ulasan aplikasi KAI Access pada *platform* Play Store sebanyak 29.016 data. Didapatkan 674 data pada bulan Januari, 775 data pada bulan Februari, 3.473 pada bulan Maret, 4.331 pada bulan April, 3.128 data pada bulan Mei, 1.127 data pada bulan Juni, 1.159 data pada bulan Juli, 707 data pada bulan Agustus, dan 2.362 data pada bulan September. Sedangkan pada bulan Oktober 2021 didapatkan 3.670 data, pada bulan November 2021 sebanyak 1.560 data, pada bulan Desember 2021 sebanyak 1.304 data, pada bulan Januari 2022 sebanyak 1.110 data, pada bulan Februari 2022 sebanyak 841 data, pada bulan Maret 2022 sebanyak 2.033 data, dan pada bulan April 2022 sebanyak 762 data.

Dataset ulasan aplikasi KAI Access ini memiliki beberapa kolom yang cukup beragam terkait detail dari ulasan yang diunggah. *Dataset* ini tediri dari beberapa kolom yaitu *package name, app version code, app version name, reviewer language, device, review submit date and time, review submit millis since epoch, review last update date and time, review last update millis since epoch, star rating, review title, review text, developer reply date and time,*

developer reply millis since epoch, developer reply link, dan *review link*. Sampel data mentah ulasan aplikasi KAI Access diperlihatkan pada Gambar 3.7.

Package Name	App Version Code	App Version Name	Reviewer Language	Device	Review Submit Date and Time	Review Submit Millis Since Epoch
com.kai.kaiticketing	92	4.5.1	en	CPH1901	2021-10-01T00:02:38Z	1633046558581
com.kai.kaiticketing	81	4.4.1	id	1820	2021-10-01T00:08:33Z	1633046913770
com.kai.kaiticketing			en	beyond0	2021-10-01T00:13:08Z	1633047188455
com.kai.kaiticketing	92	4.5.1	id	merlin	2021-10-01T00:18:12Z	1633047492923
com.kai.kaiticketing	98	4.6.4	id	RMX1821	2021-10-01T00:18:37Z	1633047517952
com.kai.kaiticketing	98	4.6.4	id	tulip	2021-10-01T00:31:46Z	1633048306552
com.kai.kaiticketing	98	4.6.4	id	j7velte	2021-10-01T00:32:42Z	1633048362415
com.kai.kaiticketing	98	4.6.4	id	1601	2021-10-01T00:34:37Z	1633048477987
com.kai.kaiticketing	98	4.6.4	id	lancelot	2021-10-01T00:35:14Z	1633048514423
com.kai.kaiticketing	98	4.6.4	id	on7xelte	2021-10-01T00:38:58Z	1633048738748
com.kai.kaiticketing	84	4.4.3	id	a31	2021-10-01T00:55:22Z	1633049722368
com.kai.kaiticketing	98	4.6.4	en	olive	2021-10-01T00:57:21Z	1633049841903
com.kai.kaiticketing	98	4.6.4	id	A1601	2021-09-04T15:16:05Z	1630768565826

(a)

Review Last Update Date and Time	Review Last Update Millis Since Epoch	Star Rating	Review Title	Review Text
2021-10-01T00:02:38Z	1633046558581	5		
2021-10-01T00:08:33Z	1633046913770	5		
2021-10-01T00:13:08Z	1633047188455	5		
2021-10-01T00:18:12Z	1633047492923	4		
2021-10-01T00:18:37Z	1633047517952	1		sudah saya update dari dulu tapi kenapa kok masih muncul tulisan suruh update sih pas waktu buka??
2021-10-01T00:31:46Z	1633048306552	5		bagus dan membantu
2021-10-01T00:32:42Z	1633048362415	5		Pelayanannya bagus, nyaman, dan AC nya terasa segernya..👍👍👍, keep clean ya...
2021-10-01T00:34:37Z	1633048477987	4		
2021-10-01T00:35:14Z	1633048514423	5		
2021-10-01T00:38:58Z	1633048738748	5		Bagus
2021-10-01T00:55:22Z	1633049722368	5		
2021-10-01T00:57:21Z	1633049841903	5		
2021-10-01T01:01:43Z	1633050103306	3		Update september 2021 malah error terus ketika masuk halaman pilih kursi,tlg dperbaiki min

(b)

Developer Reply Date and Time	Developer Reply Millis Since Epoch	Developer Reply Text	Review Link
			http://play.google.com/console/developers/7684598256376423497/app/4972691018625421883/user-feedback
			http://play.google.com/console/developers/7684598256376423497/app/4972691018625421883/user-feedback
			http://play.google.com/console/developers/7684598256376423497/app/4972691018625421883/user-feedback
			http://play.google.com/console/developers/7684598256376423497/app/4972691018625421883/user-feedback
			http://play.google.com/console/developers/7684598256376423497/app/4972691018625421883/user-feedback

(c)

Gambar 3.7 Detail Data Ulasan KAI Access Bagian A (a), Bagian B (b), Bagian C (c)

3.3.2 Data Preprocessing: Data Merging

Data yang sudah dilakukan *merging* menghasilkan satu *dataset* ulasan aplikasi KAI Access dengan jumlah data sebanyak 29.016 data. Data hasil *merging* tidak menyebabkan penambahan maupun pengurangan kolom karena setiap sumber berkas data yang digabungkan memiliki kolom yang sama persis. *Code* proses *data merging* ditampilkan pada Gambar 3.8. Hasil *data merging* ini diperlihatkan pada Gambar 3.9, terlihat dalam satu *dataset* terdapat seluruh data dari waktu bulan unggah yang berbeda. *Data merging* dilakukan menggunakan beberapa *library* pada Python yaitu Pandas dan Glob dan dilakukan menggunakan IDE PyCharm. Hasil data merging kemudian disimpan dalam satu *file* CSV baru.

```
os.chdir("/Users/nabilahnran/Documents/PyCharmDoc/reviewKAIAccess/raw_data")
extension = 'csv'
all_filenames = [i for i in glob.glob('raw*.{}'.format(extension))]
df = pd.concat([pd.read_csv(f, encoding='utf-16') for f in all_filenames ])
df.to_csv("combined_csv.csv", index=False, encoding='utf-8-sig')
```

Gambar 3.8 Code Data Merging Data Ulasan KAI Access

1	Package Name	App Version Code	App Version Name	Reviewer Language	Device	Review Submit Date and Time	Review Submit Millis Since Epoch	Review Last Update Date and Time	Review Last Update Millis Since Epoch
2	com.kai.kaiticketing	84.00.00	04.04.03	id	hermes	2021-01-01T00:12:	1,60946E+12	2021-01-01T00:12:44Z	1,60946E+12
3	com.kai.kaiticketing	84.00.00	04.04.03	id	2006	2021-01-01T00:48:	1,60946E+12	2021-01-01T00:48:19Z	1,60946E+12
676	com.kai.kaiticketing	66.00.00	04.02.00	en	lucye	2021-02-01T00:07:	1,61214E+12	2021-02-01T00:07:24Z	1,61214E+12
677	com.kai.kaiticketing	81.00.00	04.04.01	id	1606	2021-02-01T00:13:	1,61214E+12	2021-02-01T00:13:23Z	1,61214E+12
1451	com.kai.kaiticketing	38.00.00	1.4.4.3	en	a50s	2021-03-01T00:18:	1,61456E+12	2021-03-01T00:18:49Z	1,61456E+12
1452	com.kai.kaiticketing	85.00.00	04.04.04	id	ugg	2021-03-01T00:54:	1,61456E+12	2021-03-01T00:54:49Z	1,61456E+12
4924	com.kai.kaiticketing	84.00.00	04.04.03	id	X9009	2021-04-01T00:09:	1,61724E+12	2021-04-01T00:09:19Z	1,61724E+12
4925	com.kai.kaiticketing	85.00.00	04.04.04	id	1904	2021-04-01T00:28:	1,61724E+12	2021-04-01T00:28:46Z	1,61724E+12
9255	com.kai.kaiticketing	88.00.00	04.04.07	id	ginkgo	2021-05-01T00:19:	1,61983E+12	2021-05-01T00:19:59Z	1,61983E+12
9256	com.kai.kaiticketing	85.00.00	04.04.04	id	CPH1909	2021-05-01T00:25:	1,61983E+12	2021-05-01T00:25:54Z	1,61983E+12
12383	com.kai.kaiticketing	81.00.00	04.04.01	id	a50s	2021-06-01T00:30:	1,62251E+12	2021-06-01T00:30:48Z	1,62251E+12
12384	com.kai.kaiticketing			id	ginkgo	2021-06-01T01:37:	1,62251E+12	2021-06-01T01:37:08Z	1,62251E+12
13510	com.kai.kaiticketing			en	m20lte	2021-07-01T00:37:	1,6251E+12	2021-07-01T00:37:55Z	1,6251E+12
13511	com.kai.kaiticketing	90.00.00	04.04.09	id	1201	2021-07-01T01:08:	1,6251E+12	2021-07-01T01:08:14Z	1,6251E+12
14669	com.kai.kaiticketing			id	a71	2021-08-01T00:28:	1,62778E+12	2021-08-01T00:28:05Z	1,62778E+12
14670	com.kai.kaiticketing	91.00.00	04.05.00	id	Infinix-X680B	2021-08-01T01:23:	1,62778E+12	2021-08-01T01:23:32Z	1,62778E+12
15376	com.kai.kaiticketing	92.00.00	04.05.01	id	lavender	2021-09-01T01:44:	1,63046E+12	2021-09-01T01:44:03Z	1,63046E+12
15377	com.kai.kaiticketing	93.00.00	04.06.00	id	ASUS_X01BD_	2021-09-01T02:09:	1,63046E+12	2021-09-01T02:09:41Z	1,63046E+12

Gambar 3.9 Hasil Proses Data Merging Data Ulasan KAI Access

3.3.3 Data Preprocessing: Data Cleaning

Column Selection

Column selection ini dilakukan dengan menghapuskan kolom-kolom yang tidak dipakai melalui *code* Python dengan *library* Pandas fungsi *drop*. *Code column selection* ini diperlihatkan pada Gambar 3.10. Terdapat kolom-kolom yang diputuskan oleh penulis untuk dihapuskan yaitu *package name, app version code, reviewer language, review submit date and time, review submit millis since epoch, review last update millis since epoch, review title,*

developer reply date and time, developer reply millis since epoch, developer reply text, dan *review link.*

```
df = df.drop(['Package Name', 'App Version Code', 'reviewer languange',
              'Review Submit Date and Time', 'Review Submit Millis Since
               Epoch', 'Review Last Update Millis Since Epoch',
              'Review Title', 'Developer Reply Date and Time',
              'Developer Reply Millis Since Epoch', 'Developer Reply
               Text','Review Link',], axis =1)
```

Gambar 3.10 Code Column Selection Data Ulasan KAI Access

Fitur-fitur ini yang tidak dihapuskan adalah *app version name, device, review last update date and time, star rating, review text*. Data yang akan digunakan untuk pembuatan model adalah *review text*, dan data yang akan digunakan untuk pembuatan *dashboard* adalah *app version name, device, review last update date and time,* dan *star rating.* Sebelum memproses data lebih lanjut lagi, setiap kolom data dirubah namanya dengan mengganti spasi dengan *underscore* dan mengubah seluruh huruf menjadi *lower case*. Hal ini untuk memudahkan penulisan *code* di proses-proses selanjutnya. Proses pengubahan nama kolom data ini diperlihatkan pada Gambar 3.11. Gambar 3.12 memperlihatkan hasil keseluruhan kolom yang tersisa setelah proses *column selection.*

```
df = df.rename( columns={'App Version Name': 'app_ver_name',
                         'Reviewer Language' :'reviewer_language',
                         'Device':'device',
                         'Review Last Update Date and Time':
                         'review_month',
                         'Star Rating':'star_rating',
                         'Review Text':'review_text'})
```

Gambar 3.11 Code Rename Kolom Data Ulasan KAI Access

app_ver_name	device	review_month	star_rating	review_text
4.5.1	CPH1901	2021-10-01T00:02:38Z	5	
4.4.1	1820	2021-10-01T00:08:33Z	5	
	beyond0	2021-10-01T00:13:08Z	5	
4.5.1	merlin	2021-10-01T00:18:12Z	4	
4.6.4	RMX1821	2021-10-01T00:18:37Z	1	
4.6.4	tulip	2021-10-01T00:31:46Z	5	bagus dan membantu
4.6.4	j7velte	2021-10-01T00:32:42Z	5	Pelayanannya bagus, nyaman, dan AC nya terasa segernya..👍👍👍, keep clean ya...
4.6.4	1601	2021-10-01T00:34:37Z	4	
4.6.4	lancelot	2021-10-01T00:35:14Z	5	
4.6.4	on7xelte	2021-10-01T00:38:58Z	5	
4.4.3	a31	2021-10-01T00:55:22Z	5	
4.6.4	olive	2021-10-01T00:57:21Z	5	
4.6.4	A1601	2021-10-01T01:01:43Z	3	Update september 2021 malah error terus ketika masuk halaman pilih kursi,tlg dperbaiki min

Gambar 3.12 Hasil Proses Column Selection Data Ulasan KAI Access

Filtering

Proses *filtering* dilakukan pada Pycharm dengan *code* seperti yang ditampilkan pada Gambar 3.13. *Data filtering* dilakukan dengan membersihkan teks pada kolom *review text* karena data pada kolom inilah yang akan dijadikan data *input* model untuk klasifikasi, kolom lainnya dibiarkan apa adanya karena data sudah pada format seharusnya dan hanya akan dipakai untuk proses analisis. Proses ini menggunakan *library* pada Python yaitu Pandas. Setelah dilakukan *filtering*, data ulasan terlihat lebih bersih karena tidak ada karakter-karakter yang tidak diperlukan.

```
#Remove punctuation and emojis
df['review_text'] = df['review_text'].str.replace('[^\w\s]','')
#remove URLs
df['review_text'] = df['review_text'].replace(r'http\S+', '',
    regex=True).replace(r'www\S+', '', regex=True)
#remove newlines
df['review_text'] = df['review_text'].str.replace('\n',' ')
#replace two space to one
df['review_text'] = df['review_text'].str.replace('\s\s+' , ' ',
    regex = True)
#remove leading space
df['review_text'] = df['review_text'].replace('^ +| +$', '',
    regex = True)
#remove number
df['review_text'] = df['review_text'].str.replace('\d+', '')
#remove single letter
df['review_text'] = df['review_text'].str.replace(r'\b\w\b',
'').str.replace(r'\s+', ' ')
```

Gambar 3.13 Code Data Filtering Data Ulasan KAI Access

Pada Tabel 3.11 diberikan contoh sebuah teks sebelum dan sesudah dilakukan *filtering*. Emoji, koma, dan titik pada teks sebelum terhapus.

Tabel 3.11 Hasil Proses Filtering Data Ulasan KAI Access

Sebelum	Sesudah
bagus dan membantu	bagus dan membantu
Pelayanannya bagus, nyaman, dan AC nya terasa segernya..👍👍👍, keep clean ya...	Pelayanannya bagus nyaman dan AC nya terasa segernya keep clean ya
Update september 2021 malah error terus ketika masuk halaman pilih kursi,tolong diperbaiki min	Update september malah error terus ketika masuk halaman pilih kursi tolong diperbaiki min

Handling Missing Values

Dari 29.016 data pada *dataset*, 19.782 data diantaranya tidak memiliki isi teks ulasan. Sehingga, data yang memiliki teks ulasan atau data yang dapat diolah pada proses selanjutnya

didapatkan hanya berjumlah 9.234. Pada data untuk analisis terdapat beberapa *missing values* pada fitur *device* dan versi aplikasi. Untuk kepentingan analisis, jika terdapat *missing values* di kolom-kolom tersebut namun teks ulasannya tidak kosong, maka *missing values* pada kolom *device* dan versi aplikasi tersebut diganti dengan kata 'unknown' seperti yang diperlihatkan pada Gambar 3.14. Contoh data pada Gambar 3.12 yang sudah dilakukan *handling missing values* terdapat pada Gambar 3.15.

```
df.drop(df[df['review_text'].isnull()].index, inplace= True )
df.loc[(df.device.isnull()), 'device'] = 'unknown'
df.loc[(df.app_ver_name.isnull()), 'app_ver_name'] = 'unknown'
```

Gambar 3.14 Code Handling Missing Values Data Ulasan KAI Access

app_ver_name	device	review_month	star_rating	review_text
4.6.4	tulip	2021-10-01T00:31:46Z	5	bagus dan membantu
4.6.4	j7velte	2021-10-01T00:32:42Z	5	Pelayanannya bagus nyaman dan AC nya terasa segernya keep clean ya
4.6.4	A1601	2021-10-01T01:01:43Z	3	Update september 2021 malah error terus ketika masuk halaman pilih kursi tlg dperbaiki min

Gambar 3.15 Hasil Proses Handling Missing Values Data Ulasan KAI Access

Case Folding

Proses *case folding* pada *cleaning data* ini dilakukan dengan cukup sederhana dengan menggunakan *library* String seperti yang dapat dilihat pada Gambar 3.16. Tabel 3.12 menunjukkan contoh teks ulasan sebelum dan sesudah dilakukannya *case folding* pada data ulasan aplikasi KAI Access. Seluruh data teks pada data latih memiliki huruf kecil.

```
#Lowering Case
df['review_text'] = df['review_text'].str.lower()
print(df['review_text'])
```

Gambar 3.16 Code Case Folding Data Ulasan KAI Access

Tabel 3.12 Hasil Proses Case Folding Data Ulasan KAI Access

Sebelum	Sesudah
bagus dan membantu	bagus dan membantu
Pelayanannya bagus nyaman dan AC nya terasa segernya keep clean ya	pelayanannya bagus nyaman dan ac nya terasa segernya keep clean ya
Update september malah error terus ketika masuk halaman pilih kursi tlg dperbaiki min	update september malah error terus ketika masuk halaman pilih kursi tlg dperbaiki min

3.3.4 Data Labeling

Data labeling dilakukan secara manual oleh penulis dengan menggunakan *tools* Google Sheets. Setelah dilakukan *data labeling* pada keseluruhan data ulasan aplikasi KAI Access yang telah dibersihkan, terdapat beberapa data ulasan yang memuat beberapa topik bahasan di dalamnya, sehingga dilakukan pemisahan data sehingga data bertambah sebanyak 124 data. Oleh karena itu, jumlah data yang akan diproses semula berjumlah 9.234 menjadi 9.358. Hasil contoh pelabelan sentimen, topik, dan detail topik pada data ulasan KAI Access diperlihatkan pada Gambar 3.17.

A	B	C	D
Review Text	**label sentimen**	**label topik**	**label detail topik**
sudah saya update dari dulu tapi kenapa kok masih muncul tulisan suruh update sih pas waktu buka	negatif	error	error
bagus dan membantu	positif	feedback	baik
pelayanannya bagus nyaman dan ac nya terasa segernya keep clean ya	positif	feedback	baik
bagus	positif	feedback	baik
update september malah error terus ketika masuk halaman pilih kursi tolong diperbaiki min	negatif	pemesanan	errorpemesanan

Gambar 3.17 Hasil Proses Data Labeling Data Ulasan KAI Acces

Hasil *data labeling* menunjukkan pada klasifikasi sentimen, sebanyak 5.677 data memiliki kelas sentimen Positif, 2.935 memiliki sentimen Negatif, dan sisanya sebanyak 746 data adalah sentimen Netral. Pada klasifikasi topik, diperoleh kelas Feedback sebanyak 6.775 data, Pemesanan sebanyak 793 data, Registrasi-Login sebanyak 623 data, Pembayaran sebanyak 466 data, Error sebanyak 450 data, Pengaturan sebanyak 162 data, dan 89 data lainnya dilabeli dengan kelas topik Pembatalan. Pada klasifikasi detail topik, kelas Baik memiliki data sebanyak 5.459 data, kelas Lain sebanyak 1.315 data, kelas Login sebanyak 378 data, kelas Registrasi sebanyak 245, kelas Error pemesanan sebanyak 382 data, kelas Get Jadwal sebanyak 237 data, kelas Book Beda ID sebanyak 50 data, kelas e-Ticket sebanyak 49 data, kelas Edit Pemesanan sebanyak 47 data, kelas Maximum Pemesanan sebanyak 27 data, kelas Edit Profil sebanyak 155 data, kelas Edit Password sebanyak 7 data, kelas Metode Bayar sebanyak 218 data, kelas Error Pembayaran sebanyak 177 data, kelas Payment Notif sebanyak 73 data, kelas Error Pembatalan sebanyak 82 data, kelas Announce Batal sebanyak 7 data, kelas Error sebanyak 212 data, kelas Akses App sebanyak 88 data, Kelas Crash sebanyak 89 data, dan kelas Timeout sebanyak 61 data.

3.3.5 Data Preprocessing: Data Splitting

Data splitting pada proyek ini memisahkan data yang akan dipakai menjadi bahan data pembelajaran oleh mesin yang kemudian dibagi lagi menjadi data menjadi data *input* dan data target. Fitur yang digunakan sebagai *input* pembelajaran oleh model adalah data ulasan yang sebelumnya sudah dibersihkan (teks ulasan) dan tentunya label sentimen, topik, dan detail topik menjadi data target. *Input* dan label data dimasukkan ke dalam variabel yang berbeda seperti yang ditampilkan pada Gambar 3.18. Fitur lainnya akan diabaikan, tidak dimasukkan ke variabel, hanya akan digunakan untuk keperluan analisis data.

```
import pandas as pd
data = pd.read_csv('/content/cleaned_and_labeled_data.csv')
train_set = data

#sentiment, topic, dan detail_topic dibawah adalah kolom data sentimen,
topik, dan detail topik pada data
text_train = train_set['review_text']
sent_train = train_set['sentiment']
topic_train = train_set['topic']
detail_train = train_set['detail_topic']

df = pd.read_csv('GABUNGAN_FINAL_KAIACCESS.csv')

df_training = df.loc[df['review_month'] <= '2021-09-31']
df_training.to_csv('data_training.csv')
df_testing = df.loc[df['review_month'] > '2021-09-31']
df_testing.to_csv('data_testing.csv')
```

Gambar 3.18 Code Data Splitting Data Ulasan KAI Access

Star rating adalah kolom yang cukup berpotensi sebagai fitur pendukung dalam klasifikasi sentimen karena dapat menunjukkan tingkat kepuasan pengguna. Namun saat ditinjau pada *dataset* secara keseluruhan, banyak pengguna memberikan *star rating* yang tidak sesuai dengan ulasan yang dipaparkan pada fitur teks ulasan sehingga *star rating* tidak dapat menjadi fitur yang kredibel untuk membantu proses pembelajaran model. Namun, kolom ini berguna untuk bahan *dashboard* analisis sehingga kolom *star rating* tidak dihapuskan namun tidak dipakai pada pembelajaran model.

Selain itu, terdapat pembagian data menjadi dua bagian, yaitu *training data* dan *testing data*. Data pada bulan Januari 2021 hingga September 2021 digunakan untuk *training data* dan data pada bulan Oktober 2021 hingga April 2022 digunakan untuk *testing data*. Total keseluruhan data yang akan diproses sebagai *training data* adalah sebanyak 8.293 data, dan total data yang akan diproses sebagai *testing data* adalah sebanyak 1.065 data. *Data splitting* menjadi *training* dan *testing data* ini *code*nya dapat dilihat pada Gambar 3.18 dan hasil jumlah *training* dan *testing data* setelah proses *splitting* diperlihatkan pada Gambar 3.19.

	app_ver_name	device
1	app_ver_name	device
2	04.04.03	hermes
3	04.04.03	2006
4	04.04.01	ginkgo
5	04.02.00	K33a42
6	04.04.03	j5y17ltedx
7	04.00.01	ASUS_X00T_3
8288	04.06.05	CPH1729
8289	04.06.04	a20s
8290	04.06.05	CPH1803
8291	04.07.01	RMX1911
8292	04.06.04	m11q

	App Version Name	Device
1	App Version Name	Device
2	unknown	davinci
3	04.09.02	m22
4	04.09.02	rolex
5	04.09.02	CPH1909
6	unknown	citrus
7	unknown	dandelion
1063	04.08.00	alioth
1064	04.08.00	2023

(b)

Gambar 3.19 Training Data (a), Testing (b) Hasil Proses Data Splitting Data Ulasan KAI Access

3.3.6 Data Preprocessing: Data Tokenization

Teks ulasan dan label pada data *input* ditokenisasi dengan menggunakan Tokenizer TensorFlow. Setiap kata pada data dan label akan diubah menjadi *unique token*, setiap kata yang berbeda memiliki token yang berbeda dan hasil token dari keseluruhan kata pada data dimasukkan ke dalam *vocabulary token*. Selain mendapatkan token dari data *input*, TensorFlow Tokenizer juga menyediakan satu token yang dapat menjadi token pengganti pada kata yang tidak diketahui/tidak terdapat pada *vocabulary*. Token tersebut disebut OOV (*out of vocabulary*) *token*.

Parameter fungsi TensorFlow Tokenizer yang dipakai pada saat melakukan *tokenizing* teks ulasan adalah parameter *number of words* (parameter yang ditampilkan pertama) dan OOV *token*. *Number of words* adalah angka maksimum kata yang akan disimpan berdasarkan frekuensi kata pada data *input*. Pada proyek ini parameter tersebut diisi dengan angka 100.000, angka yang sangat besar, agar tidak membatasi jumlah *vocabulary* kata. OOV token juga dipakai untuk menggantikan kata yang tidak diketahui *vocabulary* saat modelnya dipakai untuk data baru nantinya. Namun pada fungsi *tokenizer* untuk label tidak menggunakan parameter apapun karena panjang *input tokenizer* label sudah mutlak/pasti berjumlah 1 dan tidak akan berubah. *Input tokenizer* label juga tidak mungkin ada diluar *vocabulary*, sehingga tidak memerlukan parameter OOV Token. Setiap kata pada data ulasan akan diubah menjadi *unique token* dengan fungsi fit_on_texts().

Data label memiliki tokenizernya sendiri untuk memiliki *vocabulary*nya sendiri, berbeda dengan *vocabulary* teks ulasan. Label yang semula dalam bentuk sebuah kata, pada *data tokenizing* diubah menjadi token agar dapat dipelajari bersama dengan token teks ulasan oleh

model. *Code word tokenizing* data ulasan KAI Access diperlihatkan pada Gambar 3.20 dengan contoh rincian *vocabulary* hasil tokenisasi kata dari data pada Tabel 3.12 adalah seperti yang terlihat pada Tabel 3.13.

```
from tensorflow.keras.preprocessing.text import Tokenizer
vocab_size = 100000
oov_tok = '<OOV_TOK>'


tokenizer = Tokenizer(vocab_size, oov_token = oov_tok)
tokenizer.fit_on_texts(text_train)
text_word_index = tokenizer.word_index

sent_tokenizer = Tokenizer()
sent_tokenizer.fit_on_texts(sent_train)
sent_word_index = sent_tokenizer.word_index

topic_tokenizer = Tokenizer()
topic_tokenizer.fit_on_texts(topic_train)
topic_word_index = topic_tokenizer.word_index

detail_tokenizer = Tokenizer()
detail_tokenizer.fit_on_texts(detail_train)
detail_word_index = detail_tokenizer.word_index
```

Gambar 3.20 Data Tokenizing Data Ulasan KAI Access

Tabel 3.13 Contoh Daftar Vocabulary Tokenisasi Data Teks Ulasan KAI Access

Key	Value		Key	Value		Key	Value
[1]	<OOV_TOK>		[10]	segernya		[19]	ketika
[2]	bagus		[11]	keep		[20]	masuk
[3]	dan		[12]	clean		[21]	halaman
[4]	membantu		[13]	ya		[22]	pilih
[5]	pelayanannya		[14]	update		[23]	kursi
[6]	nyaman		[15]	september		[24]	tolong
[7]	ac		[16]	malah		[25]	diperbaiki
[8]	nya		[17]	error		[26]	min
[9]	terasa		[18]	terus			

Setelah masing-masing kata memiliki tokennya masing-masing, dilakukanlah pembuatan *sequence* pada data yaitu mengubah data teks menjadi *array* token dengan susunan kata sesuai *vocabulary*. Pembuatan *sequence* ini dilakukan dengan menggunakan fungsi TensorFlow Keras Sequence fit_on_sequence() yang memiliki parameter data *input* yang akan disusun sebagai *sequence*.

Data yang sudah berbentuk *sequence* kemudian dilakukan *padding*, proses ini mengubah setiap data sehingga memiliki panjang yang sama. Proses ini dilakukan dengan menggunakan TensorFlow Keras Sequence fungsi pad_sequences() yang memiliki parameter *sequence* data, *padding type*, *truncating type*, dan *maxlen*. Penulis memilih *padding type post* yaitu membuat

isian tambahan token akan ditambahkan diakhir *sequence*. *Truncating type* juga diset *post* agar tidak mengurangi inti kalimat yang biasanya berada di awal kalimat jika panjang *sequence* melebihi *maxlen*. *Maxlen* diisi dengan panjang *maximum* dari data *input* (*maximum length*) agar tidak ada *sequence* yang terpotong/terkena *truncating*. *Sequence* dan *padding* dilakukan juga pada data token label (target), *sequence* label untuk menjadi parameter target pada fungsi *compile* dan label yang dilakukan *padding* digabungkan dengan teks ulasan sebagai *input* klasifikasi topik dan sentimen. Maka dari itu, label detail topik tidak dilakukan padding karena tidak digunakan sebagai *input* klasifikasi. Teks ulasan *Code* untuk membuat *sequence* dan *padding* ditampilkan pada Gambar 3.21. Contoh data teks ulasan yang telah diubah menjadi *sequence* token dan telah dilakukan *padding* dapat dilihat pada Tabel 3.14.

```
# ================TEXT SEQ & PAD================
text_train_seq = tokenizer.texts_to_sequences(text_train)
text_train_pad = pad_sequences(text_train_seq, padding=padding_type,
                               truncating=trunc_type, maxlen=max_length)

# ================SENT SEQ & PAD================
sent_train_seq = np.array(sent_tokenizer.texts_to_sequences(sent_train))
sent_train_pad = pad_sequences(sent_train_seq, padding=padding_type,
                               truncating=trunc_type, maxlen=1)

# ================TOPIC SEQ & PAD================
topic_train_seq =
np.array(topic_tokenizer.texts_to_sequences(topic_train))
topic_train_pad = pad_sequences(topic_train_seq, padding=padding_type,
                                truncating=trunc_type, maxlen=1)

# ================DETAIL TOPIC SEQ PAD================
detail_train_seq = np.array(detail_tokenizer.
                    texts_to_sequences(detail_train))
```

Gambar 3.21 Code Pembuatan Sequence dan Padding Data Ulasan KAI Access

Tabel 3.14 Contoh Hasil Sequence dan Padding Data Ulasan KAI Access

Sebelum	**Sesudah Diubah Menjadi Sequence**	**Sesudah Diubah Menjadi Padded Sequence**
bagus dan membantu	[[2], [3], [4]]	[[2], [3], [4], [0], [0], [0], [0], [0], [0], [0], [0], [0], [0], , [0], [0], … , [0]]
pelayanannya bagus nyaman dan ac nya terasa segernya keep clean ya	[[5], [2], [6], [3], [7], [8], [9], [10], [11], [12], [13]]	[[5], [2], [6], [3], [7], [8], [9], [10], [11], [12], [13], [0], [0], [0], [0], … , [0]]
update september malah error terus ketika masuk halaman pilih kursi tolong diperbaiki min	[[14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26]]	[[14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [0], [0], ... , [0]]

3.3.7 Data Modeling

Masing-masing klasifikasi pada setiap label dilakukan *training* secara berurut mulai dari klasifikasi sentimen, klasifikasi topik, dan terakhir klasifikasi detail topik. *Input* klasifikasi sentimen adalah teks ulasan, *input* klasifikasi topik adalah teks ulasan dan label sentimen. Tak lupa dengan klasifikasi detail topik, selain teks ulasan, label sentimen dan label topik juga dimasukkan sebagai *input training* model klasifikasi detail topik. *Data modeling* pada proyek ini seluruhnya menggunakan TensorFlow, yaitu *open-source framework* yang menyediakan banyak *library* untuk Machine Learning yang akan memudahkan dalam pengembangan model.

Model Klasifikasi Sentimen

Sesuai dengan yang telah ditentukan sebelumnya pada subbab 3.2.5, model klasifikasi sentimen ini dibangun dengan menggunakan arsitektur MLP. *Code* pembuatan model klasifikasi label sentimen ini diperlihatkan pada Gambar 3.22.

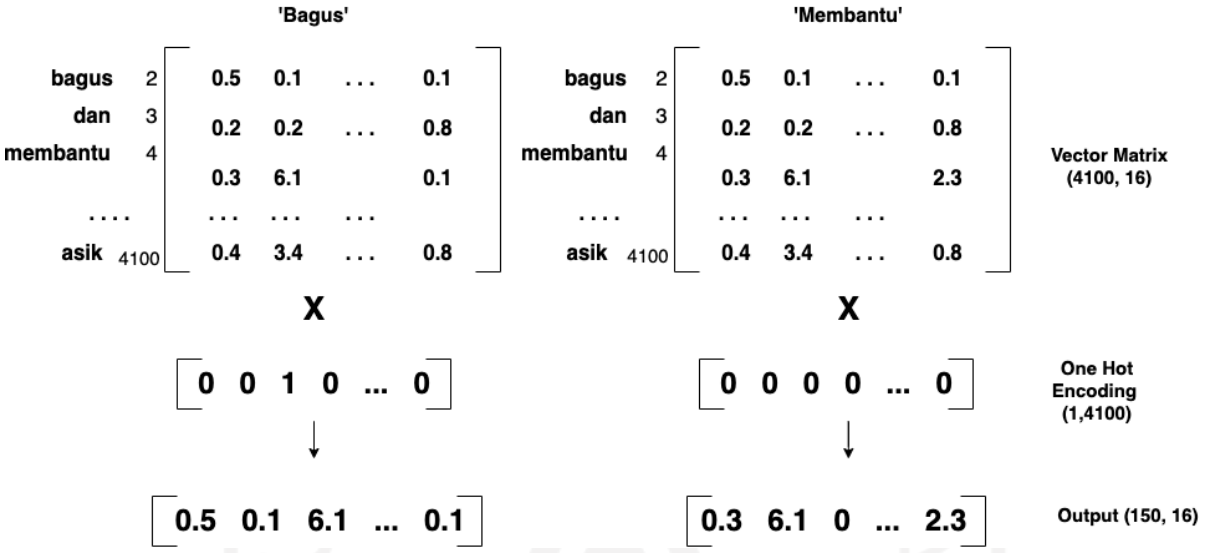
```
import tensorflow as tf
embedding_dim = 16
max_length = 150

ckpt_path = "detail/cp.ckpt"
cp_callback = tf.keras.callbacks.ModelCheckpoint(filepath=ckpt_path,
                                                 save_weights_only=True,
                                                 verbose=1)
sent_train_label_seq = np.array(sent_tokenizer.texts_to_sequences(sent_train))

sent_model = tf.keras.Sequential([
  tf.keras.layers.Embedding(4100, embedding_dim,input_length = max_length),
  tf.keras.layers.GlobalAveragePooling1D(),
  tf.keras.layers.Dense(32, activation= 'relu'),
  tf.keras.layers.Dense(4, activation = 'softmax')
])

sent_model.compile(optimizer = 'adam', loss = 'sparse_categorical_crossentropy',
                   metrics = ['sparse_categorical_accuracy'],
                   callback = cp_callback)

sent_hist = sent_model.fit(text_train_pad, sent_train_label_seq, epochs = 40,
                           callbacks= [cp_callback])
sent_model.save('sentiment_model.h5')
```

Gambar 3.22 Code Pembuatan Model dan Compiler Klasifikasi Sentimen Data Ulasan KAI Access

Pada *layer* pertama terdapat Embedding Layer yang mengubah setiap token kata menjadi vektor 16 dimensi. Pertama-tama, data yang sudah di*tokenize* dan sudah dalam bentuk *padded* seperti pada Tabel 3.14 akan dimasukkan pada Embedding Layer. Embedding Layer akan menghasilkan matriks berukuran (*input dimension/vocab size, output dimension)* yang artinya

matriks hasil Embedding Layer tersebut akan berbentuk (4100, 16) berisikan *random vektor* yang di*generate* oleh *layer* untuk setiap kata pada *vocabulary.*

Teknis penarikan vektor perkata dalam setiap data digambarkan pada Gambar 3.23. Setiap kata pada *vocabulary* mempunyai matriks *one hot encoding* (OHE) berdasarkan indeks kata tersebut pada *vocabulary*, matriks OHE setiap kata ini berurukuran (1, *vocab size*) atau (1, 4100). Untuk menarik vektor suatu kata pada matriks Embedding untuk menyusunnya sebagai *output* dari Embedding Layer, model mengalikan matriks vektor Embedding Layer yang berisikan vektor seluruh kata pada *vocabulary* dengan matriks OHE kata yang diinginkan, sehingga hasil yang didapatkan adalah vektor kata tersebut berukuran (1, *output dimension)* atau (1,16), untuk satu kata. Sehingga untuk satu baris data menghasilkan matriks berukuran (*maximum length*, *output dimension*) atau (150, 16).



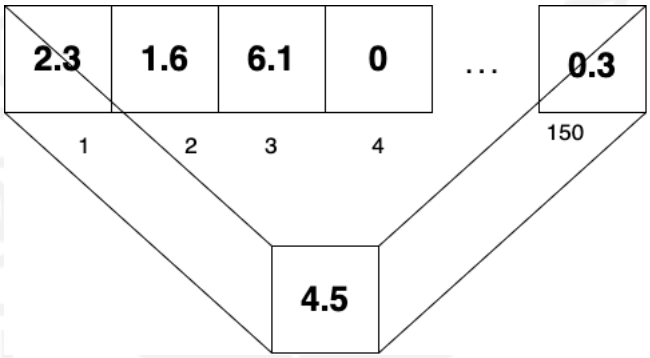
Gambar 3.23 Proses Penyusunan Vektor Output Embedding Layer

Pada Gambar 3.24 diperlihatkan hasil vektor kata ‘bagus’ dan ‘nyaman’. Angka pada kedua vektor serupa, seperti angka vektor ketiga pada kedua kata, dimana kedua angka memiliki jumlah yang berdekatan. Hal ini membuktikan, seperti pada hasil label, kedua kata tersebut memiliki vektor yang berdekatan karena keduanya sering muncul pada kelas Positif.

```
#Vektor kata 'bagus'
[ 0.01708297  0.0792605  -0.07980466 -0.07106113  0.08277994  0.02271034
 -0.05142977 -0.02761742 -0.00334069  0.01869947 -0.02971414 -0.03558976
  0.01099997  0.00938089  0.01310779  0.06343226]

#Vektor kata 'nyaman'
[ 0.04767221  0.01258586 -0.04343271 -0.00425109  0.07055457  0.01454767
 -0.01596894 -0.04215923  0.07603603  0.01256242 -0.06637429 -0.0785971
 -0.0139905   0.01185733  0.08927184 -0.00255792]
```

Gambar 3.24 Vektor Kata ‘Bagus’ dan ‘Nyaman’ pada Embedding Layer

Setelah diproses pada Embedding Layer, data diratakan bentuknya dengan Global Average Pooling 1D Layer. *Ouput* Embedding Layer yang berukuran (150, 16) diratakan dengan mengambil nilai rata-rata dari masing-masing kata (yang memiliki panjang 150) seperti yang diilustrasikan pada Gambar 3.25, sehingga hasil akhir hanya berukuran 16. Setelahnya, data diproses pada Dense Layer dengan 32 unit dan aktivasi ReLu. *Layer* ini memiliki 32 unit, dipilih dari hasil percobaan dengan beberapa pilihan unit lainnya yaitu 16 unit dan 64 unit yang hasil akurasi masing-masing unit tertera pada Tabel 3.15. Dense Layer dengan jumlah 32 unit menunjukkan hasil akurasi yang paling tinggi diantara kedua unit lainnya.



Gambar 3.25 Ilsutrasi Proses Global Average Pooling 1D

Tabel 3.15 Perbandingan Hasil Percobaan Jumlah Unit Dense Model Klasifikasi Sentimen

Jumlah Unit Dense	Akurasi Training
16	97, 80%
32	97, 97%
64	97, 88%

Layer terakhir sebagai *output layer* yaitu *layer* yang sama dengan sebelumya, yaitu Dense Layer. Dense Layer pada *layer* terakhir ini memiliki unit sesuai dengan jumlah kelas yaitu 4 unit, sehingga data akan terklasifikasikan pada salah satu kelas.

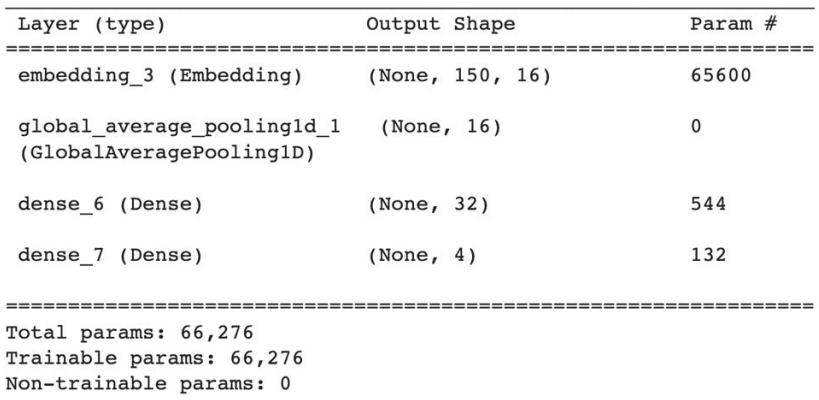
Sebelum model dilakukan *training*, TensorFlow mengharuskan pengguna untuk melakukan *compile* pada *neural network* yang sudah disusun. Fungsi *compile* yaitu menentukan beberapa parameter yang digunakan dalam *training model* yang akan dipakai. *Hyperparameter* yang dipakai pada proses *compile* ini adalah Adam *optimizer*, Sparse Categorical Crossentropy *loss*, dan Tensorflow Checkpoint. TensorFlow Callback Checkpoint digunakan pada klasifikasi ini untuk meningkatkan hasil akurasi, karena TensorFlow

Checkpoint menyimpan *model weights* pada setiap langkahnya (TensorFlow, Callbacks Model Check Point, 2022).

Model ini di*training* sebanyak 40 *epoch*. Penentuan jumlah *epoch* dilakukan dengan percobaan beberapa pilihan jumlah *epoch* diantara lain 30, 40, 50, dan 60 *epoch.* Model yang dilatih dengan 40 adalah model yang memiliki akurasi *training* paling baik dan *validation loss* yang stabil pada angka angka minimum seperti perbandingan yang dipaparkan pada Tabel 3.16. *Validation Loss* yang semakin tinggi sejalan dengan bertambahnya *epoch* menjadi salah satu indikasi terjadinya *overfitting* pada model, walaupun model memiliki akurasi yang meningkat tetapi model menjadi lebih kurang percaya diri dengan prediksinya. Dengan mencoba melakukan *training* pada model dengan 60 *epoch* sudah membuat *validation loss* meningkat dan akurasi *training* juga sudah tinggi, maka percobaan pada jumlah *epoch* yang lebih besar dari 60 tidak dilakukan. Setelah selesai dilatih, didapatkan hasil *training accuracy* sebesar 96,90%. Hasil *summary* model klasifikasi sentimen data ulasan KAI Access ini ditunjukkan pada Gambar 3.26.

Tabel 3.16 Perbandingan Hasil Percobaan Jumlah Epoch Model Klasifikasi Sentimen

Jumlah Epoch	Akurasi Training	Validation Loss
30	96,42	Stabil
40	97,38	Stabil
50	97,79	Meningkat
60	98,17	Meningkat



Gambar 3.26 Summary Model Label Sentimen Data Ulasan KAI Access

Model Klasifikasi Topik

Sesuai dengan yang telah ditentukan sebelumnya pada subbab 3.2.5, model klasifikasi topik proyek data ulasan KAI Access ini dibangun dengan menggunakan arsitektur BiLSTM.

Code pembuatan model klasifikasi topik data ulasan KAI Access secara keseluruhan ditampilkan pada Gambar 3.27.

```
topic_train_input = np.concatenate([text_train_pad, sent_train_pad], axis=1)

topic_model = tf.keras.Sequential([
                tf.keras.layers.Embedding(4100, embedding_dim, input_length =
                   max_length+1),
              tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(16)),
              tf.keras.layers.Dense(32, activation = 'relu'),
              tf.keras.layers.Dense(8, activation = 'softmax')
])
topic_model.compile(optimizer = 'adam', loss = 'sparse_categorical_crossentropy',
                    metrics = ['sparse_categorical_accuracy'])
topic_hist = topic_model.fit(topic_train_input, topic_train_seq, epochs = 60)
topic_model.save('topic_model.h5')
```

Gambar 3.27 Code Pembuatan Model dan Compiler Klasifikasi Topik Data Ulasan KAI Access

Pertama-tama, digabungkan *pad* token data teks ulasan dengan *pad* label sentimen untuk menjadi *input* model. Pada *layer* pertama, terdapat Embedding Layer sebagai *input layer*. Embedding Layer pada klasifikasi topik ini bekerja dengan sama seperti Embedding Layer pada klasifikasi sentimen, namun jumlah *maximum length* model ini berjumlah 151. Data *input* yang telah berupa token dalam bentuk *padded* akan masuk ke Embedding Layer untuk didapatkan vektor setiap kata pada kata *vocab*, berukuran (4100, 16). Diambil vektor kata tertentu pada setiap data untuk disusun menjadi keluaran Embedding Layer berukuran 2 dimensi (151, 16).

Setelah keluar dari Embedding Layer, data kemudian masuk ke BiLSTM Layer. Data *input* diproses secara dua arah dan hasil kedua proses tersebut digabungkan diakhir. *Input* masuk ke dalam BiLSTM Layer dengan jumlah unit yang ditentukan adalah 16. Unit pada BiLSTM umumnya adalah kelipatan 16, BiLSTM dengan jumlah unit 16 digunakan karena memiliki hasil yang paling baik dari beberapa pilihan unit lainnya yaitu 32 dan 64 unit. Hasil perbandingan jumlah unit *layer* BiLSTM ditampilkan pada Tabel 3.17. Hasil dengan menggunakan jumlah unit sebanyak 16 menghasilkan akurasi yang lebih tinggi dibandingkan hasil akurasi unit dengan jumlah 32 dan 64.

Tabel 3.17 Perbandingan Hasil Percobaan Jumlah Unit BiLSTM Model Klasifikasi Topik

Jumlah Unit BiLSTM	**Akurasi Training**
16	99,95%
32	99,79%
64	99,32%

Selanjutnya ada Dense *layer* yang memiliki jumlah unit 32 yang juga ditentukan dari perbandingan dengan unit lainnya yaitu jumlah unit 16. Model dengan jumlah unit pada Dense Layer sebanyak 32 unit menghasilkan akurasi yang lebih tinggi seperti yang dapat dilihat pada Tabel 3.18. Aktivasi yang digunakan pada *layer* ini sama dengan yang digunakan pada Dense Layer pada klasifikasi sentimen dengan alasan yang sama. Terakhir, pada Dense Layer terakhir sebagai *output layer*, parameter diisi dengan 8 unit sebagai jumlah kelas Topik dan menggunakan aktivasi Softmax.

Tabel 3.18 Perbandingan Hasil Percobaan Jumlah Unit Dense Model Klasifikasi Topik

Jumlah Unit Dense	**Akurasi Training**
16	99,75%
32	99,95%

Kemudian model klasifikasi topik ini di*compile* dengan Adam *optimizer*, Sparse Categorical Crossentropy *loss*, dan Tensorflow. Setelah itu, data dilatih dengan jumlah *epoch* 60 kali. Hasil akurasi yang didapatkan selama 60 *epoch* yaitu sebesar 99,95% pada *training*. Ringkasan model label topik ini diperlihatkan pada Gambar 3.28.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_2 (Embedding)      (None, 151, 16)           65600

bidirectional_2 (Bidirectio  (None, 32)               4224
nal)

dense_4 (Dense)              (None, 16)                528

dense_5 (Dense)              (None, 8)                 136

=================================================================
Total params: 70,488
Trainable params: 70,488
Non-trainable params: 0
```

Gambar 3.28 Summary Model Label Topik Data Ulasan KAI Access

Model Klasifikasi Detail Topik

BiLSTM adalah metode yang dipilih untuk menyusun model klasifikasi detail topik. Secara susunan arsitektur modelnya, model klasifikasi detail topik memiliki susunan *layer* yang sama seperti model klasifikasi topik. Embedding Layer sebagai *input layer* dan Dense Layer sebagai *output layer*, diantara *input* dan *output layer* terdapat Bidirectional LSTM Layer dan Dense Layer. Code pembuatan model klasifikasi dan *compiler* detail topik ini diperlihatkan pada Gambar 3.29.

```
detail_train_input = np.concatenate([text_train_pad, sent_train_pad,
```

```
                                    topic_train_pad], axis=1)

ckpt_path = "detail/cp.ckpt"
cp_callback = tf.keras.callbacks.ModelCheckpoint(filepath=ckpt_path,
                                                 save_weights_only=True,
                                                 verbose=1)

detail_model = tf.keras.Sequential([
                  tf.keras.layers.Embedding(4100, embedding_dim,
                     input_length = max_length+2),
               tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(32)),
               tf.keras.layers.Dense(32, activation = 'relu'),
               tf.keras.layers.Dense(22, activation = 'softmax')
])

detail_model.compile(optimizer = 'adam',
                     loss ='sparse_categorical_crossentropy',
                     metrics = ['sparse_categorical_accuracy'],
                     callback = cp_callback)
detail_hist = detail_model.fit(detail_train_input, detail_train_seq, epochs = 60,
                               callbacks = [cp_callback])
detail_model.save('detail_model.h5')
```

Gambar 3.29 Code Pembuatan Model dan Compiler Klasifikasi Detail Topik Data Ulasan KAI Access

Sebelum membuat model, *input* untuk model klasifikasi detail topik ini dibuat terlebih dahulu. *Input* ini dibuat dengan menggabungkan data teks ulasan, label sentimen, dan label topik. Dilanjutkan dengan pembuatan model, Embedding Layer pada *layer* pertama menerima data *input* berupa token lalu menghasilkan matriks vektor berukuran (4100, 16). Kemudian vektor setiap kata pada data disusun sebagai keluaran Embedding Layer. Sehingga, ukuran matriks pada *output* Embedding Layer adalah (150, 16) untuk sebuah data sesuai dengan ukuran *input length* dan dimensi yang dipilih. Dikarenakan data *input* telah ditambahkan dengan label sentimen dan topik, maka Embedding Layer akan membuat vektor lebih akurat lagi. Data dengan label sentimen atau topik yang sama akan memiliki kemungkinan mempunyai vektor yang dekat, relevan dengan label detail topik yang merupakan label yang mendetail dari setiap kelas pada label topik.

Selain itu, yang berbeda pada klasifikasi detail topik ini dibandingkan dengan klasifikasi topik, jika jumlah unit BiLSTM pada klasifikasi topik berjumlah 16, pada klasifikasi ini jumlah unit pada BiLSTM Layer sebanyak 32 unit. Hal ini ditentukan berdasarkan percobaan beberapa unit lainnya yaitu 32 dan 64 yang ditampilkan pada Tabel 3.19. BiLSTM Layer dengan jumlah unit 32 memiliki akurasi *training* yang lebih tinggi dibandingkan unit lainnya dari hasil percobaan. Pada *layer* terakhir terdapat *output layer* yaitu Dense Layer yang memiliki unit sebanyak 22, sesuai dengan banyaknya kelas untuk hasil klasifikasi detail topik pada data ulasan KAI Access.

Tabel 3.19 Perbandingan Hasil Percobaan Jumlah Unit BiLSTM Model Klasifikasi Detail Topik

Jumlah Unit BiLSTM	Akurasi Training
16	99,72%
32	99,73%
64	99,59%

Model klasifikasi detail topik ini di*compile* sama seperti model klasifikasi sebelumnya, klasifikasi sentimen dan topik, yaitu dengan Adam *optimizer*, Sparse Categorical Crossentropy *loss*, dan TensorFlow Checkpoint untuk meningkatkan akurasi model. Serupa dengan klasifikasi topik, hasil pembelajaran dengan jumlah 60 *epoch* cukup menghasilkan akurasi *training* yang bagus yaitu sebesar 99,73%. Ringkasan model label detail topik ini diperlihatkan pada Gambar 3.30.

```
Layer (type)                 Output Shape              Param #
=================================================================
embedding_8 (Embedding)      (None, 152, 16)           65600

bidirectional_8 (Bidirectio  (None, 64)                12544
nal)

dense_16 (Dense)             (None, 32)                2080

dense_17 (Dense)             (None, 22)                726

=================================================================
Total params: 80,950
Trainable params: 80,950
Non-trainable params: 0
```

Gambar 3.30 Summary Model Label Detail Topik Data Ulasan KAI Access

Ketiga model kemudian disimpan dengan menggunakan fungsi save_model() dari TensorFlow yang akan menyimpan model ke dalam sebuah *file*. Ketiga model disimpan dalam format h5. *File* ini akan digunakan untuk membuat *prototype* pada langkah selanjutnya.

3.3.8 Evaluasi Model

Predicted label pada *testing data* untuk *input* evaluasi model dibuat dengan memasukkan *testing data* ke dalam model yang sudah di*training* untuk ditentukan label datanya oleh model. Proses ini dilakukan menggunakan Python dengan *library* Numpy dan TensorFlow. Gambar 3.31, Gambar 3.32, dan Gambar 3.33 berturut-turut menampilkan *code* untuk membuat *predicted label* sentimen, topik, dan detail topik pada *testing dtaa.* Dengan ketiga *code* tersebut, kolom label sentimen, topik, dan detail topik pada setiap datanya ditambahkan secara otomatis

pada tabel *testing data*. Tabel *testing data* yang sudah mempunyai kolom sentimen, topik, dan detail topik hasil klasifikasi model kemudian disimpan dalam *file* CSV baru. Tabel *testing data* hasil pembuatan *predicted labels* ini diperlihatkan pada Gambar 3.34.

```
data_predict = pd.read_csv('data_testing.csv')

sent_test_text = np.array(data_predict['Review Text'])
sent_test_input = np.array(pad_sequences(
                            tokenizer.texts_to_sequences(sent_test_text),
                            padding=padding_type, truncating=trunc_type,
                            maxlen=max_length))
sprediction = sent_model.predict(sent_test_input)

slist_result = []
for row in range(len(sprediction)):
  sresult = sprediction[row].tolist().index(np.max(sprediction[row]))
  slist_result.append(sresult)
spre_result=[]

for s in slist_result:
  spre_result.append(sent_tokenizer.index_word[s])

data_predict['predict_sentiment'] = np.array(spre_result)
```

Gambar 3.31 Code Pembuatan Predicted Label Sentimen Testing Data Ulasan KAI Access

```
testing_text = np.array(data_predict['Review Text'])
testing_text = pad_sequences(tokenizer.texts_to_sequences(testing_text),
                             padding=padding_type, truncating=trunc_type,
                             maxlen=max_length)
testing_sent = np.array(data_predict['predict_sentiment'])
testing_sent = pad_sequences(sent_tokenizer
                             .texts_to_sequences(testing_sent),
                             padding=padding_type, truncating=trunc_type,
                              maxlen=1)
testing_input = np.concatenate([testing_text, testing_sent], axis=1)

predicting = topic_model.predict(np.array(testing_input))
listresult = []

for row in range(len(predicting)):
  result = predicting[row].tolist().index(np.max(predicting[row]))
  listresult.append(result)

preresult = []
for aa in listresult:
  preresult.append(topic_tokenizer.index_word[aa])

data_predict['predict_topic'] = np.array(preresult)
```

Gambar 3.32 Code Pembuatan Predicted Label Topik Testing Data Ulasan KAI Access

```
testing_text = np.array(data_predict['Review Text'])
testing_text = pad_sequences(tokenizer.texts_to_sequences(testing_text),
                             padding=padding_type, truncating=trunc_type,
                             maxlen=max_length)
testing_sent = np.array(data_predict['predict_sentiment'])
testing_sent = pad_sequences(sent_tokenizer
                             .texts_to_sequences(testing_sent),
```

```
                                padding=padding_type, truncating=trunc_type,
                                maxlen=1)
testing_topic = np.array(data_predict['predict_topic'])
testing_topic = pad_sequences(topic_tokenizer
                               .texts_to_sequences(testing_topic),
                               padding=padding_type,truncating=trunc_type,
                               maxlen=1)
testing_input = np.concatenate([testing_text, testing_sent,
                                 testing_topic], axis=1)

predicting = detail_model.predict(np.array(testing_input))
listresult = []

for row in range(len(predicting)):
  result = predicting[row].tolist().index(np.max(predicting[row]))
  listresult.append(result)

preresult = []

for aa in listresult:
  preresult.append(detail_tokenizer.index_word[aa])

data_predict['detail_topic_prediction'] = np.array(preresult)
```

Gambar 3.33 Code Pembuatan Predicted Label Detail Topik Testing Data Ulasan KAI Access

Review Text	Sentiment Prediction	Topic Prediction	Detail Prediction
aplikasi nya super lemot sudah daftar sampai ke pilihan pembayaran mau ganti metode pembayaran malah keluar mengisi ulang lagi dan lama banget kayak jaman punya laptop intel celeron tahun yang lalu tolong di tingkatkan lagi kecepatannya aplikasinya	negatif	registrasilogin	login
pesan udah mau bayar sistem tidak respon di update juga tetep gitu tolong dong kai access perbaiki sistemnya	negatif	pembayaran	paymentnotif
transfer saldo laporan masuk saldo berkurang tapi saldo di kai access tidak nambah	negatif	pembayaran	errorpembayaran
zaman sudah semakin maju semua bisa di akses dari gadget sudah bagus sih aplikasi ini tapi masih kurang sempurna karena kalau mau ubah jadwal pembatalan tidak bisa di akses untuk ka lokal masih harus datang ke service center ke stasiun padahal tinggal nanti uang nya dikembalikan via alat pembayaran yang digunakan saat pembayaran tiket dan seharusnya sudah mencakup semua lah	negatif	pemesanan	editpemesanan
tidak bisa edit nomor kalau mau pakai otp lebih baik ditaruhnya pas aktivasi aktivasi kaipay nya tidak bisa	negatif	registrasilogin	registrasi
tidak bisa di ragukan lagi kalian wajib banget download karna di sini tiketnya murah woy	negatif	feedback	lain
mau pilih metode pembayaran aplikasinya malah sering keluar keluar sendiri	negatif	pembayaran	errorpembayaran
aplikasi mau nyari kemana kok tidak bisa jancok	negatif	pemesanan	bookbedaid
sip	positif	feedback	baik
lebih baik lagi	positif	feedback	baik
masih tetep sama seperti yang kemaren sebelum di update tidak bisa login padahal password sudah sesuai dan aplikasi sudah di install ulang tolonglah diperbaiki agar masyarakat bisa menikmati manfaat kereta api apalagi sebentar lagi mau mudik lebaran	negatif	registrasilogin	login

Gambar 3.34 Tabel Testing Data Hasil Pembuatan Predicted Labels Data Ulasan KAI Access

Testing data yang memiliki *predicted label* dan *testing data* yang memiliki *actual label* hasil *labeling* manual oleh penulis dimasukkan untuk dilakukan perbandingan guna mengevaluasi model. Evaluasi dilakukan dengan menggunakan Python *library* untuk dapat mengevaluasi data dalam jumlah banyak. Evaluasi dilakukan dengan menggunakan *library* Python Pandas dan Sklearn Metrics dan proses ini dilakukan di Google Colab. Dengan fungsi

classification_result(), *output* yang akan dikeluarkan adalah nilai akurasi, presisi, *recall*, *f1-score*, dan *support*/frekuensi munculnya data untuk setiap kelas. *Code* proses evaluasi seperti yang ditampilkan pada Gambar 3.35.

```
from sklearn import metrics

#data testing yang sudah dilabeli secara manual
data = pd.read_csv('testing_labelled.csv')
#data testing yang sudah dilabeli secara otomatis oleh model
data_result = pd.read_csv('predicted_testing_labelled.csv')

Y_act = [data['sentiment'].to_list(), data['topic'].to_list(),
data['detail_topic'].to_list()]
Y_pred = [data_result['Sentiment Prediction'].to_list(), data_result['Topic
Prediction'].to_list(), data_result['Detail Prediction'].to_list()]

print(Y_act)
print(Y_pred)
def calc_metrics(label, num):
  print(metrics.classification_report(Y_act[num], Y_pred[num], digits=4))

calc_metrics('sentiment', 0)
calc_metrics('topic', 1)
calc_metrics('detail topic', 2)
```

Gambar 3.35 Code Proses Evaluasi Data Ulasan KAI Access

Klasifikasi Sentimen

Tabel 3.20 menunjukkan hasil performa model klasifikasi sentimen dengan MLP. Hasil ini menunjukkan kelas negatif memiliki nilai *recall*, presisi, dan *f1-score* yang tidak jauh berbeda dan dari masing-masing *metrics*, ketiganya memiliki nilai yang tinggi. Klasifikasi sentimen ini mendapat evaluasi akurasi yang cukup tinggi yaitu sebesar 87,35%.

Tabel 3.20 Hasil Evaluasi Model Klasifikasi Sentimen Data Ulasan KAI Access

Kelas	Precision	Recall	F1 Score	Jumlah Data	Akurasi
Negatif	0.9146	0.9470	0.9305	509	
Netral	0.6056	0.3282	0.4257	131	
Positif	0.8678	0.9532	0.9085	427	
Total				1067	0.8735

Label sentimen menjadi tolak ukur yang penting dalam kepuasan pengguna dalam penggunaan aplikasi. Dikarenakan klasifikasi teks ulasan ini ditujukan untuk peningkatan kualitas pelayanan berdasarkan ulasan ketidakpuasan, maka *error* dalam pengklasifikasian model akan lebih baik jika model memiliki hasil klasifikasi False Positive (FP) lebih besar daripada False Negative (FN) pada kelas Negatif. Maka dari itu, diharapkannya nilai *metrics recall* yang tinggi pada kelas negatif dan kelas netral serta presisi yang tinggi pada kelas Positif.

Hasil dari *evaluation metrics* pada tabel di atas, *recall* pada kelas Negatif memiliki nilai yang tinggi yaitu 94,7%, lebih tinggi dari nilai presisinya yang dimiliki yaitu sebesar 91,4%.

Hal ini sesuai dengan harapan, walaupun selisih dari kedua nilai stersebut tidak terlalu besar. Pada kelas Netral, hasil *metrics*nya berbanding terbalik dibandingkan dengan yang diharapkan karena hasilnya lebih besar nilai presisi dibandingkan dengan *recall* dan jumlah selisih keduanya cukup besar. Serupa dengan kelas Netral, kelas Positif tidak sesuai harapan karena yang diharapkan adalah presisinya yang lebih tinggi namun kenyataannya *recall* yang lebih tinggi. Namun secara keseluruhan, model sentimen ini memiliki nilai F1 Score yang tinggi untuk kelas Negatif dan Positif.

Klasifikasi Topik

Pada Tabel 3.21, ditunjukkan hasil performa model klasifikasi topik dengan BiLSTM dimana kelas Feedback, Pembayaran dan Pengaturan memiliki nilai evaluasi yang cukup tinggi. Akurasi dari evaluasi klasifikasi topik ini adalah sebesar 79,10%.

Tabel 3.21 Hasil Evaluasi Model Klasifikasi Topik Data Ulasan KAI Access

Kelas	Precision	Recall	F1 Score	Jumlah Data	Akurasi
Error	0.7037	0.4790	0.5700	119	
Feedback	0.8737	0.9509	0.9107	611	
Pembatalan	0.1795	0.4375	0.2545	16	
Pembayaran	0.9221	0.6961	0.7933	102	
Pemesanan	0.5981	0.4885	0.5378	131	
Pengaturan	0.6750	0.8182	0.7397	33	
Registrasi-Login	0.6379	0.6727	0.6549	55	
Total				1067	0.7910

Kelas Pemesanan, Pembayaran, Pembatalan, Pengaturan, dan Registrasi-Login adalah label yang memiliki ulasan yang spesifik pada bagian permasalahannya. Sehingga pada penggunaan klasifikasi ini dikemudian hari, ada kecenderungan pengembang akan memprioritaskan pengecekan kelompok ulasan dengan label Topik yang memiliki kelas Pemesanan, Pembayaran, Pembatalan, Pengaturan, dan Registrasi-Login yang sudah secara spefisik bidang permasalahannya. Oleh karena itu diharapkan kelas-kelas tersebut memiliki *recall* yang lebih besar dibandingkan dengan nilai presisinya seperti kelas Sentimen Negatif, sehingga kemungkinan ulasan pada kelas-kelas spesifik itu akan terlewat/terklasifikasi pada kelas lain lebih kecil. Namun pada hasil evaluasi, diantara kelas-kelas yang disebutkan sebelumnya, hanya kelas Pembatalan, Pengaturan, dan Registrasi-Login yang memiliki nilai *recall* yang lebih tinggi daripada nilai presisinya.

Klasifikasi Detail Topik

Performa model klasifikasi detail topik dengan BiLSTM terlihat pada Tabel 3.22. Kelas Baik dan Edit Profil memiliki nilai *metrices* yang tinggi. Sedangkan kelas Announce Batal dan Maximum Pemesanan memiliki nilai *metrics all zero* sejalan dengan jumlah frekuensi kelas kelas tersebut yang tidak banyak. Klasifikasi detail topik ini memiliki evaluasi akurasi sebesar 64,85%.

Tabel 3.22 Hasil Evaluasi Model Klasifikasi Detail Topik Data Ulasan KAI Access

Kelas	Precision	Recall	F1 Score	Jumlah Data	Akurasi
Akses App	0.1250	0.3750	0.1875	8	
Announce Batal	0.0000	0.0000	0.0000	1	
Baik	0.8384	0.9505	0.8910	404	
Book Beda ID	0.1667	0.1667	0.1667	6	
Crash	0.3000	0.1429	0.1935	21	
Edit Password	0.5000	1.0000	0.6667	1	
Edit Pemesanan	0.1000	0.1667	0.1250	6	
Edit Profil	0.6667	0.8125	0.7324	32	
Error	0.7442	0.3951	0.5161	81	
Error Pembatalan	0.3333	0.4667	0.3889	15	
Error Pembayaran	0.7143	0.3922	0.5063	51	
Error Pemesanan	0.5517	0.4211	0.4776	76	
E-ticket	0.4000	0.3333	0.3636	6	
Get Jadwal	0.3750	0.2571	0.3051	35	
Lain	0.5288	0.5340	0.5314	206	
Login	0.5806	0.4390	0.5000	41	
Maximum Pemesanan	0.0000	0.0000	0.0000	1	
Metode Bayar	0.6585	0.7500	0.7013	36	
Payment Notification	0.4167	0.2941	0.3448	17	
Registrasi	0.3704	0.7143	0.4878	14	
Timeout	0.2500	0.1111	0.1538	9	
Total				1067	0.6485

Hasil akhir pada evaluasi model ini adalah model klasifikasi Sentimen memiliki akurasi sebesar 87,35%, model klasifikasi Topik memiliki akurasi sebesar 79,10%, dan model klasifikasi Detail Topik memiliki akurasi sebesar 64,85%. Model klasifikasi pada penelitian ini dapat mengklasifikasikan dengan 2 dari 3 model memiliki akurasi di atas 75%untuk data ulasan KAI Access. Namun, dilihat dari hasil rata-rata nilai *f1-score* pada *evaluation metrics* ketiga klasifikasi ini, performa dari setiap model masih belum maksimal. Hal ini dikarenakan tidak meratanya jumlah data pada setiap kelas di setiap labelnya, membuat model tidak dapat mempelajari dengan baik data pada kelas yang memiliki jumlah data yang sedikit. Kemampuan mempelajari data yang dilakukan model pada data dengan kelas yang memiliki jumlah data

yang sedikit tidak sebaik mempelajari data pada kelas dengan jumlah data yang banyak, sehingga model cenderung lebih ahli memahami data pada kelas dengan jumlah data yang banyak. Kompeksitas teks ulasan yang tinggi dan terbatasnya jumlah *training data*, menyebabkan model belum bisa belajar secara maksimal.

3.3.9 Prototype Klasifikasi Data Ulasan KAI Access

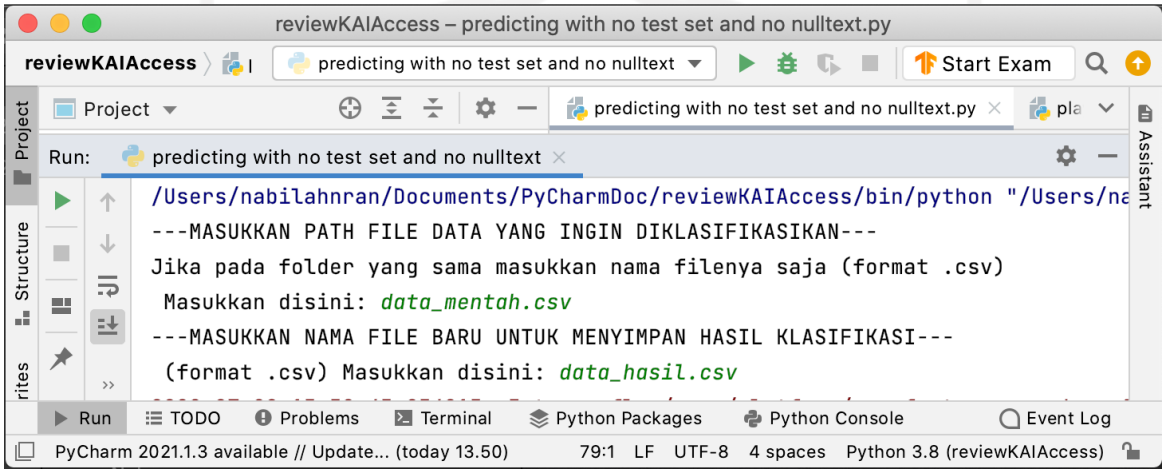
Pembuatan *Prototype* penulis buat dan jalankan pada IDE Python Pycharm. Diawali mengimpor *training data*, data mentah yang hendak diklasifikasikan, dan ketiga berkas model yang sudah disimpan dalam format h5. Kemudian, data mentah dibersihkan seperti proses *data cleaning* sebelumnya menggunakan *library* Pandas melakukan *filtering*, *handling missing values*, dan *case folding*. Data mentah tidak dipisahkan karena yang akan menjadi *input* klasifikasi hanya kolom *review text*, sehingga hanya diambil kolom tersebut. Sebelum melakukan klasifikasi pada data mentah, *tokenizer* dibuat kembali untuk proses *prototyping* ini. Hal ini dikarenakan *vocabulary* pada pembuatan model sebelumnya tidak bisa disimpan sehingga setiap menjalankan *prototype*, *vocabulary* baru dibuat Kembali dengan detail yang sama. TensorFlow Tokenizer memproses *training data* dan dibuatlah *vocabulary* kata dengan Tokenizer. Kemudian data mentah diubah menjadi *sequence* dan *padding* berdasarkan token yang telah dibuat sebelumnya memakai *training data*, lalu data mentah yang sudah berupa *padding* berisi token dimasukkan ke dalam model untuk diklasifikasikan. Seperti tujuan klasifikasi yang ingin dicapai sebagaimana yang ditampilkan pada Gambar 2.5, klasifikasi dilakukan berurut mulai dari klasifikasi sentimen, topik, dan detail topik dengan label sentimen menjadi *input* klasifikasi topik dan detail topik serta label topik ikut serta sebagai *input* klasifikasi detail topik. Alur proses *running prototype* ini ditulis di *code* dan ditampilkan pada Gambar 3.36. Isi detail masing-masing fungsi yang dipanggil pada proses *prototyping* ini dijabarkan pada halaman Lampiran.

```
#load data training
data_vocab = pd.read_csv('training_data.csv')
data_vocab.review_text=data_vocab.review_text.astype(str)
data_vocab.drop(columns=['app_ver_name', 'reviewer_language','device',
                         'review_month', 'star_rating'], inplace = True)

#load data mentah
predict_file = input('Masukkan path file data yang ingin diklasifikasikan\nJika
                      pada folder yang sama masukkan nama filenya saja
                     (format .csv) \n Masukkan disini: ')
data_predict = pd.read_csv(predict_file, delimiter=';')

#nama data hasil
result_file = input ('Masukkan nama file baru untuk menyimpan hasil klasifikasi
                     (format .csv)\n Masukkan disini: ')
```

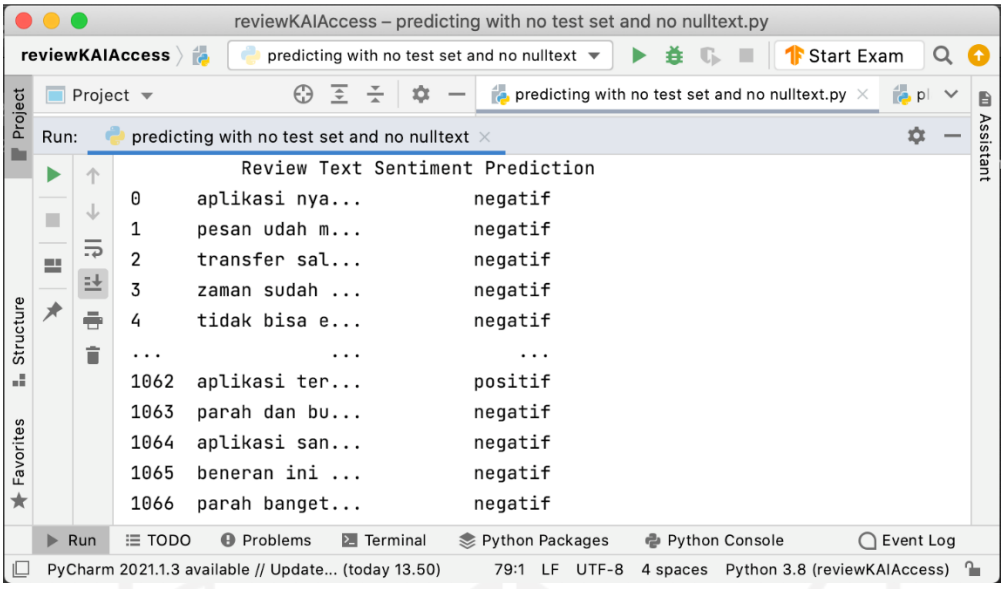
```
sent_model = load_model('sent_model.h5')
topic_model = load_model('topic_model.h5')
detail_model = load_model('detail_model.h5')

# preprocessing data mentah baru untuk dibersihkan
data_predict = preprocessing(data_predict)
# tokenizing data lama untuk list vocabulary token
tokenizer, text_word_index, sent_tokenizer, sent_word_index,
  topic_tokenizer, topic_word_index, detail_tokenizer,
  detail_word_index = process_vocab(data_vocab)
# data baru diklasifikasikan untuk label sentimen
data_predict = predict_sentiment(data_predict)
# data baru diklasifikasikan untuk label topik
data_predict = predict_topic(data_predict)
# data baru diklasifikasikan untuk data detail topik
data_predict = predict_detail(data_predict)
# data hasil klasifikasi secara keseluruhan disimpan pada file baru
data_predict.to_csv('data_hasil.csv')
```

Gambar 3.36 Main Code Penyusunan Prototype Klasifikasi Data KAI Access

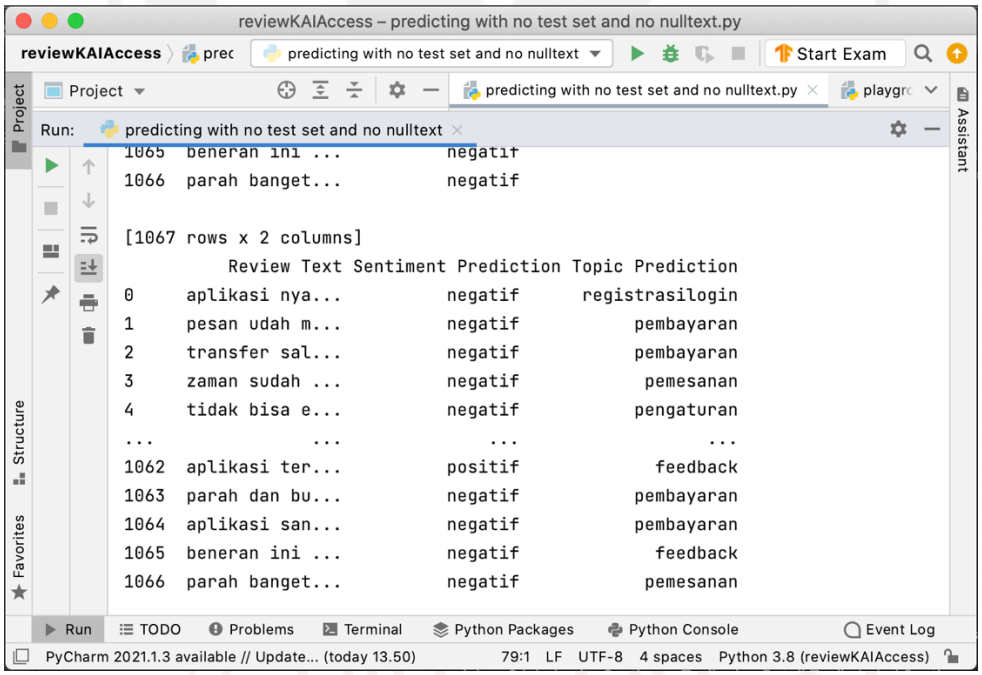
Proses klasifikasi sentimen, topik, dan detail topik dilakukan berurut dan memakan waktu *running* selama 11,89 detik pada data mentah yang berjumlah 1.067 data. Waktu *running* tersebut sudah termasuk *data tokenization training data*, *data cleaning* dan *data tokenization* data mentah, dan klasifikasi masing-masing label. *Output* dari *prototype* adalah data ulasan KAI Access awal beserta ketiga hasil klasifikasi labelnya yaitu kolom sentimen, topik, dan detail topik. Data hasil *running prototype* tersebut beserta semua otomatis disimpan ke dalam sebuah berkas CSV baru. Sehingga, berkas CSV baru hasil dari *prototype* ini memuat semua kolom yang sudah ada pada data mentah sebelumnya serta memuat kolom baru yaitu kolom label sentimen, topik, dan detail topik. Proses interaktif memasukkan data yang ingin diklasifikasikan hingga klasifikasi selesai oleh *prototype* dan nama berkas untuk menyimpan hasil klasifikasi diperlihatkan pada Gambar 3.37.



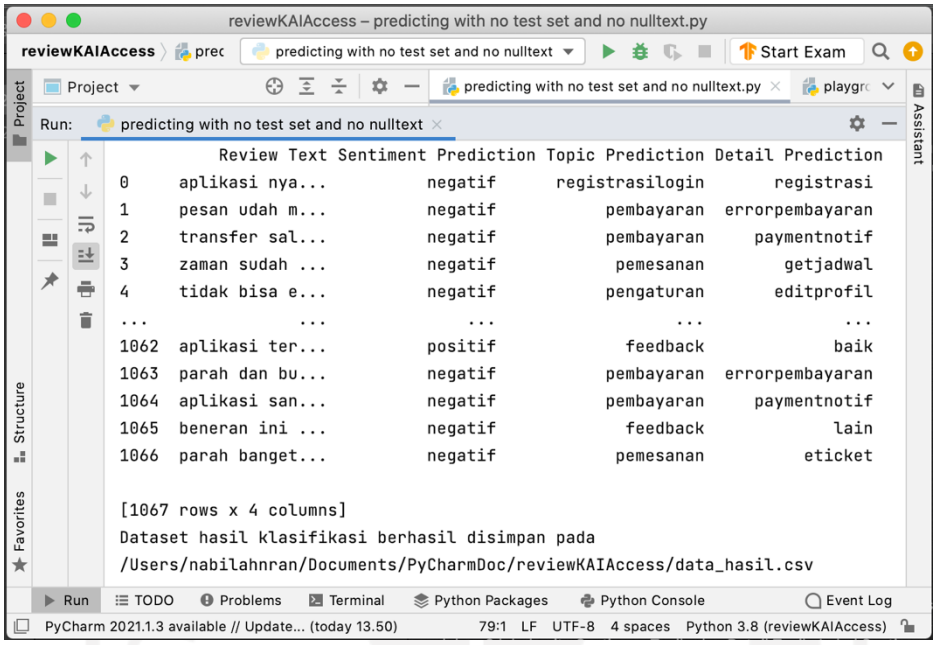
(a)


reviewKAIAccess – predicting with no test set and no nulltext.py
reviewKAIAccess
predicting with no test set and no nulltext
Start Exam
Project
predicting with no test set and no nulltext.py
Run:
predicting with no test set and no nulltext
Review Text Sentiment Prediction
0 aplikasi nya... negatif
1 pesan udah m... negatif
2 transfer sal... negatif
3 zaman sudah ... negatif
4 tidak bisa e... negatif
... ...
1062 aplikasi ter... positif
1063 parah dan bu... negatif
1064 aplikasi san... negatif
1065 beneran ini ... negatif
1066 parah banget... negatif
Run
TODO
Problems
Terminal
Python Packages
Python Console
Event Log
PyCharm 2021.1.3 available // Update... (today 13.50)
79:1 LF UTF-8 4 spaces Python 3.8 (reviewKAIAccess)
Structure
Favorites
Assistant

(b)


reviewKAIAccess – predicting with no test set and no nulltext.py
reviewKAIAccess
predicting with no test set and no nulltext
Start Exam
Project
predicting with no test set and no nulltext.py
Run:
predicting with no test set and no nulltext
1065 beneran ini ... negatif
1066 parah banget... negatif
[1067 rows x 2 columns]
Review Text Sentiment Prediction Topic Prediction
0 aplikasi nya... negatif registrasilogin
1 pesan udah m... negatif pembayaran
2 transfer sal... negatif pembayaran
3 zaman sudah ... negatif pemesanan
4 tidak bisa e... negatif pengaturan
... ...
1062 aplikasi ter... positif feedback
1063 parah dan bu... negatif pembayaran
1064 aplikasi san... negatif pembayaran
1065 beneran ini ... negatif feedback
1066 parah banget... negatif pemesanan
Run
TODO
Problems
Terminal
Python Packages
Python Console
Event Log
PyCharm 2021.1.3 available // Update... (today 13.50)
79:1 LF UTF-8 4 spaces Python 3.8 (reviewKAIAccess)
Structure
Favorites
Assistant

(c)

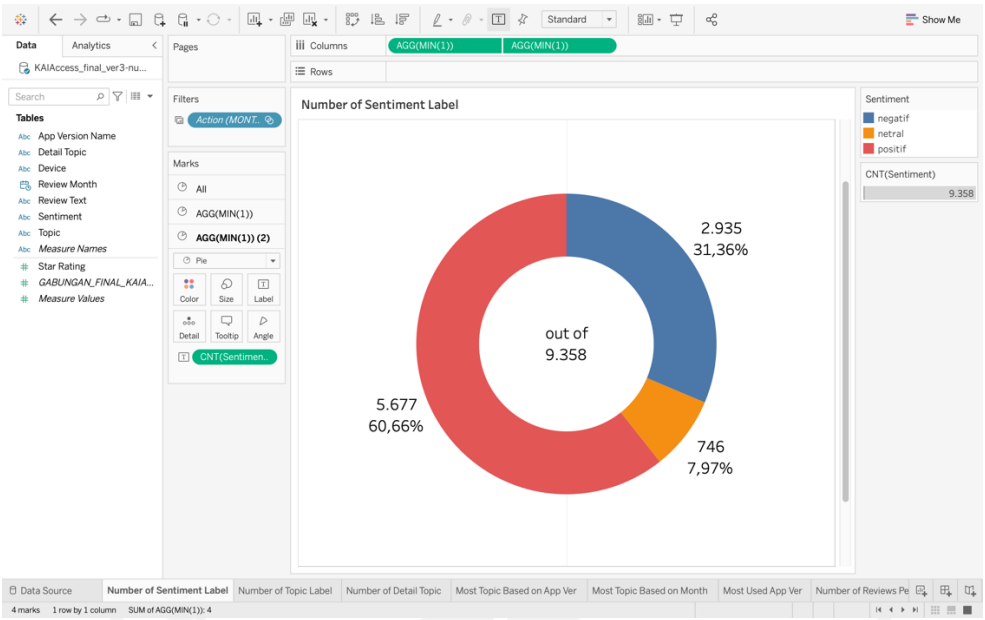


(d)

Gambar 3.37 Hasil Proyek Data Ulasan KAI Access: Prototype Klasifikasi Data Ulasan KAI Access, Interaksi Pemasukan Data Mentah dan Nama Berkas Hasil (a), Hasil Klasifikasi Sentimen (b), Hasil Klasifikasi Topik (c), Hasil Klasifikasi Detail Topik/Hasil Akhir Prototype (d)

3.3.10 Dashboard Analisis Data Ulasan KAI Access

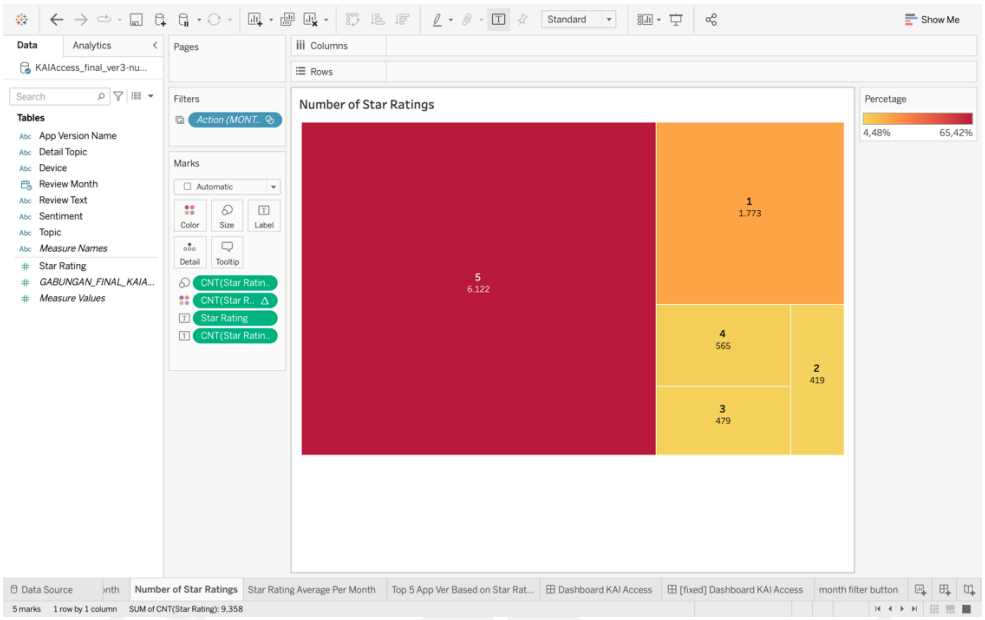
Analisis yang ditampilkan dalam bentuk *dashboard* ini dibuat dan ditampilkan dengan *tools* Tableau. Fitur-fitur data yang dipakai pada analisis ini adalah seperti hasil proses *column selection* pada Gambar 3.12 tanpa kolom *review text*, yaitu antara lain *review last update date and time, review text, app version name, star rating*, label sentimen, label topik, dan label detail topik. Data yang dipakai untuk analisis ini adalah gabungan dari *training data* dan *testing data*. Setiap *workbook* memiliki isi visualisasi dengan bentuk *chart* berbeda-beda sesuai dengan *insight* yang akan disampaikan. Pemilihan warna, ukuran, serta penambahan anotasi dipertimbangkan untuk dapat menyajikan visualisasi yang mudah dimengerti. Pada contoh Gambar 3.38, dilakukan pembuatan visualisasi analisis jumlah dan persentase data pada setiap kelas di label sentimen dengan menggunakan visual *donut chart*.



Gambar 3.38 Workbook Jumlah Data Setiap Kelas pada Label Sentimen Data Ulasan KAI Access

Setiap *insight* menggunakan kolom data yang berbeda-beda untuk dianalisis. *Insight* jumlah data ulasan setiap bulan menggunakan kolom data *review last update date*, *insight* 5 versi aplikasi yang paling banyak digunakan menggunakan kolom data *app version name*, *insight* 5 versi aplikasi yang memiliki rata-rata *rating* paling tinggi menggunakan kolom data *app version name* dan *star rating*, *insight* rata-rata *rating* pada setiap bulan menggunakan kolom data *star rating* dan *review last update date*, *insight* jumlah data pada setiap *rating* menggunakan kolom data *star rating*, dan *insight* jumlah data setiap kelas pada setiap label sentimen, topik, serta detail topik menggunakan kolom data label sentimen, topik, dan detail topik.

Setiap *workbook* yang memiliki data yang ingin di*highlight* divisualisasikan dengan perbedaan warna yang mencolok. Seperti pada *workbook* jumlah data pada setiap *rating*, *rating* dengan jumlah data yang paling banyak hingga paling sedikit divisualisasikan dengan gradasi warna merah dan kuning dengan merah menggambarkan jumlah yang paling banyak dan kuning menggambarkan jumlah paling sedikit seperti yang diperlihatkan Gambar 3.39. *Workbook* jumlah data pada label sentimen sebagai contoh lain, digunakan warna yang mencolok dan benar-benar berbeda pada masing-masing kelas sentimen seperti pada Gambar 3.38 sehingga terlihat perbedaan proporsi pada setiap bagian.



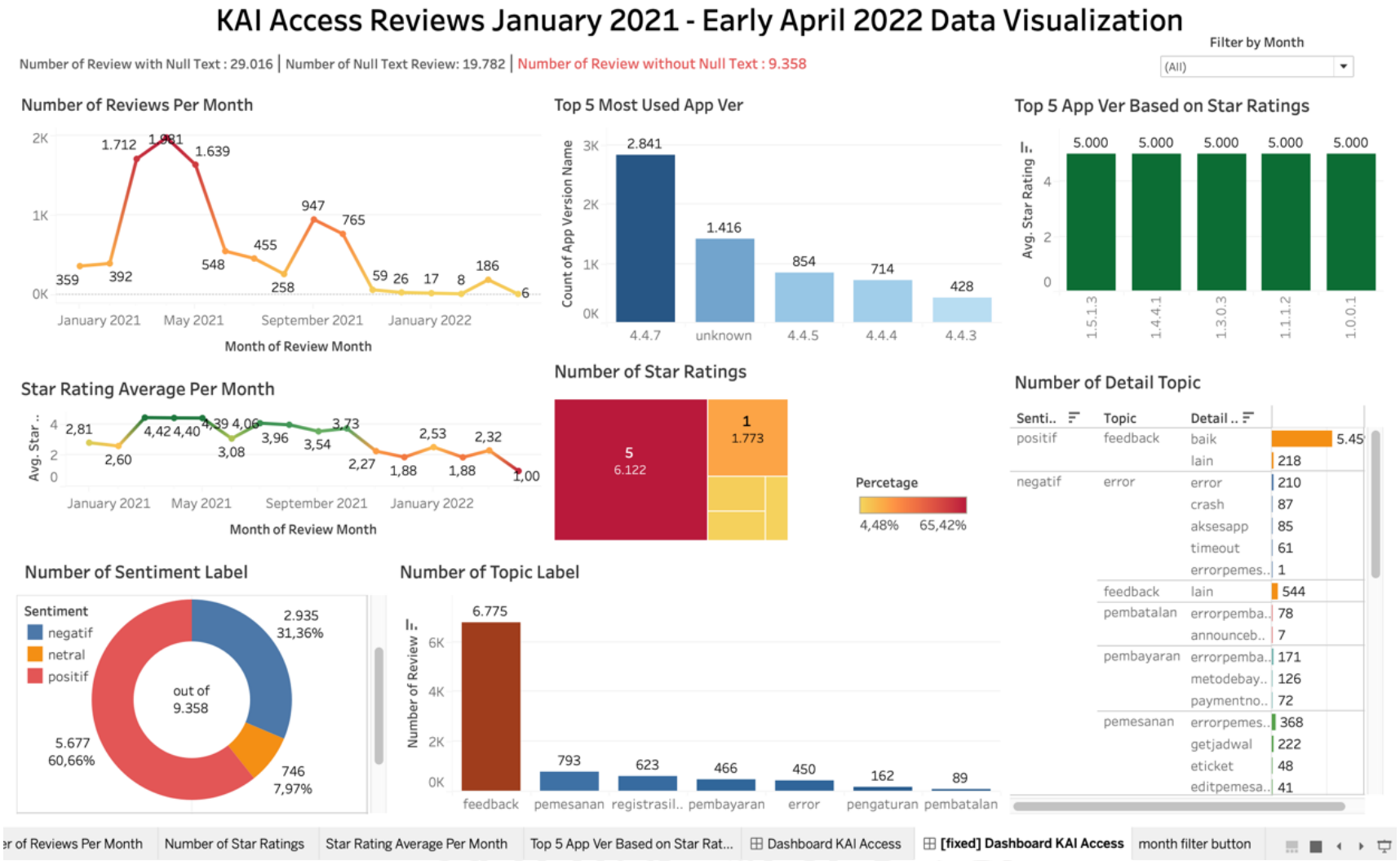
Gambar 3.39 *Workbook* Jumlah Data pada Setiap *Rating* Data Ulasan KAI Access

Visualisasi hasil analisis/*insight* pada *workbook* – *workbook* yang didapatkan dari data ulasan KAI Access inilah yang kemudian disajikan dalam bentuk *dashboard*. *Dashboard* ini menampilkan kumpulan informasi yang bermanfaat dari data ulasan KAI Access yang ringkas dan dapat dipahami oleh orang awam sekalipun.

Workbook-worbook berisi visualisasi disusun dengan apik pada halaman *dashboard*, dipastikan seluruh visualisasi dari setiap *workbook* dapat dilihat dengan mudah dan diperlihatkan dengan jelas. *Dashboard* diatur untuk satu tampilan penuh dan tidak dapat di *scroll* sesuai dengan permintaan klien (mentor). *Workbook* yang di dalamnya disusun dengan mempertimbangkan ukuran objek setiap *insight* pada *dashboard*, *insight* harus terlihat jelas dan dapat dipahami dengan cepat. Selain itu, *insight* jumlah data setiap kelas pada setiap label sentimen, topik, serta detail topik diletakkan di satu sisi yang sama agar memudahkan klien memahami analisis label bersamaan.

Selain itu, pada *dashboard* data ulasan KAI Access ini, dibuat *filter* data berdasarkan waktu dalam rentang waktu bulan yang dibuat dalam menu *dropdown*. Sehingga, tampilan *chart-chart* dalam *dashboard* dapat menampilkan analisis data hanya pada bulan tertentu sesuai yang diinginkan. Hal ini membantu klien dalam memahami analisis untuk rentang waktu perbulan. Hasil akhir analisis dalam bentuk visualisasi pada *dashboard* diperlihatkan pada Gambar 3.40. *Layout* yang dipakai pada *dashboard* ini adalah jenis *automatic layout* (ukuran *dashboard* menyesuaikan ukuran *screen* statis, tidak dapat di*scroll*) atas permintaan klien, sehingga ruang untuk menampilkan visualisasi terbatas. Agar seluruh *insight* yang telah

ditentukan cukup pada *dashboard*, Penyusunan visualisasi-visualisasi *insight* pada sebuah halaman *dashboard* ini berdasarkan penyesuaian bentuk *chart*. Selain itu, korelasi antar visualisasi *insight* sehingga yang memiliki korelasi diletakkan pada posisi yang berdekatan.



Gambar 3.40 Hasil Proyek Data Ulasan KAI Access: *Dashboard* Visualisasi Analisis

Dashboard yang diperlihatkan pada Gambar 3.40 dapat membantu analisis yang mendukung evaluasi perusahaan. Terdapat delapan visualisasi dari delapan *insight* yang berbeda sehingga dapat dibuat beberapa analisis dari setiap *insight* ataupun korelasi antara visualisasi - visualisasi tersebut.

Chart jumlah ulasan pada setiap bulan (*number of reviews per month)* pada kiri atas *dashboard* menunjukkan jumlah ulasan yang didapatkan pada setiap bulannya sejak Januari 2021 hingga awal April 2022. Hasil dari visualisasi *chart* ini, jumlah ulasan aplikasi KAI Access pada Play Store yang terbanyak adalah pada bulan April 2021 kemudian diikuti oleh bulan Maret dan Mei 2021. Hal ini berkemungkinan besar disebabkan oleh memuncaknya pembelian tiket kereta api dalam rangka mudik hari raya besar Idul Fitri pada bulan Mei 2021, sehingga banyak pengguna yang baru mengunduh dan memberikan ulasannya. Reservasi tiket kereta api bisa dilakukan sejak 90 hari sebelum tanggal keberangkatan, sehingga diasumsikan pada bulan Maret hingga April 2021 memiliki jumlah ulasan yang tinggi karena banyak pengguna yang menggunakan aplikasi pada saat itu untuk reservasi tiket mudik, begitu pula dengan pembelian tiket kereta api kembali ke daerah masing-masing pasca Idul Fitri. Tingginya angka ulasan pada bulan Mei 2021 juga menunjukkan bahwa banyak pula pengguna yang memesan tiket mendekati hari keberangkatan.

Disebelahnya, terdapat *chart* 5 versi aplikasi yang paling banyak digunakan oleh pengguna dalam *range* waktu tersebut. Hal ini dapat membantu data analis untuk mengetahui seberapa banyak pengguna pada masing-masing versi aplikasi, terlebih untuk mengetahui pengguna yang masih belum memakai aplikasi versi terbaru. Aplikasi versi terbaru tentu berisi perbaikan-perbaikan dari versi sebelumnya dan fitur pendukung yang lebih baik dari sebelumnya. Pengguna yang tidak menggunakan versi terbaru memiliki kemungkinan lebih besar mendapatkan *error* dan ketidakpuasan dibandingkan dengan versi terbaru, sehingga memunculkan ulasan yang buruk terhadap aplikasi. Versi yang paling banyak dipakai adalah versi 4.4.7, dimana versi ini bukan versi yang terbaru (versi paling baru pada saat pengambilan data terakhir adalah versi 4.9.5). Hal ini berkorelasi dengan analisis *chart* jumlah ulasan pengguna perbulan. Dikarenakan banyak pengguna yang mengunduh pada Maret – Mei 2021 dengan kondisi liburan Idul Fitri, maka banyak pengguna yang mmemiliki versi aplikasi pada masa itu, yaitu versi 4.4.7 yang diunggah oleh *developer* pada bulan Maret 2021 dan tidak memperbaruinya setalah itu.

Sebelah kanan atas *dashboard* terdapat *chart* 5 versi aplikasi yang memiliki rata-rata *star rating* paling tinggi. *Chart* ini digunakan data analis untuk mengukur kepuasan dari setiap versi

yang digunakan pengguna, khususnya 5 versi dengan kepuasan pengguna yang paling tinggi. 5 versi aplikasi dengan jumlah rata-rata *star rating* tersebut adalah versi aplikasi pada awal pengoperasian aplikasi ini, yaitu versi 1.x. Namun setelah dianalisis korelasinya dengan jumlah ulasan pada versi aplikasi tersebut, kelima versi tersebut memiliki jumlah ulasan kurang dari 5 ulasan. Sehingga, variasi dari *star rating* versi ini kecil.

Chart rerata *rating* perbulan (*star rating average per-month*) merupakan *chart* yang menampilkan rata-rata *rating* aplikasi KAI Access di setiap bulan. Ditampilkannya *chart* ini yaitu untuk melihat performa aplikasi secara keseluruhan berdasarkan penilaian pengguna pada berdasarkan bulan. Pada visualisasi *chart* tersebut, bulan Maret 2021 memiliki rerata *rating* paling tinggi diantara bulan lain yaitu dengan rerata 4,42 dari total *rating* 5 yang kemudian diikuti oleh bulan April dan Mei sebagai jumlah rerata *rating* tertinggi.

Di tengah *dashboard* terdapat visualisasi berbentuk *treemaps* untuk *insight* jumlah dari masing-masing *star rating* (*number of star rating*). *Chart* ini memvisualisasikan jumlah ulasan pada masing-masing *star rating* (angka bulat dari 1-5). Hasil visualisasi ini menunjukkan bahwa *star rating* 5, atau *star rating* tertinggi, memiliki jumlah yang paling banyak dibandingkan *rating* lainnya yaitu dengan jumlah 6.122 data. Hal ini menandakan evaluasi aplikasi yang bagus, karena 65,42% dari data ulasan keseluruhan memiliki *rate* maksimal atau dapat diartikan dengan kepuasan pengguna yang tinggi terhadap aplikasi. Tentunya hal ini perlu kembali ditinjau oleh tim data analis untuk diverifikasi kesesuaian *rating*nya dengan isi ulasannya.

Pada bagian kiri bawah *dashboard*, terdapat *chart* jumlah label sentimen atau lebih tepatnya jumlah ulasan di setiap kelas pada label sentimen. *Chart* ini dibuat dengan menggunakan *donut chart* sehingga dapat membandingkan besaran persentase setiap kelas pada keseluruhan data. Pada *chart* yang telah divisualisasikan ini, diperlihatkan bahwa kelas sentimen Positif memiliki persentase yang paling besar diantara kedua kelas lainnya dengan hasil persentase sebesar 60,66%. Karena sentimen erat kaitannya dengan kepuasan pengguna, hal ini menandakan dari keseluruhan data ulasan KAI Access yang diambil, lebih dari setengahnya memiliki pengalaman yang baik terhadap pelayanan aplikasi KAI Access. Karena sentimen yang erat kaitannya dengan kepuasan pengguna yang dapat direpresentasikan dari *star rating*, maka terdapat korelasi antara *chart* jumlah label sentimen dengan jumlah ulasan pada *star rating*. Oleh karena itu, variabel *star rating* berbanding lurus atau berkorelasi positif dengan tingkat kepuasan pengguna yang diwakilkan dengan sentimen, sehingga semakin besar angka *star rating* maka semakin baik pula sentimen pengguna pada ulasan. Maka dapat dilihat

korelasinya dengan jumlah tertinggi pada *chart* jumlah label sentimen adalah kelas Positif dan jumlah tertinggi pada *chart* jumlah *star rating* adalah *star rating* 5.

Chart di sebelah kanan *chart* jumlah label sentimen, terdapat *chart* jumlah label topik atau jumlah ulasan pada setiap kelas label topik. *Chart* untuk label topik ini dibuat dengan menggunakan *bar chart* dengan tujuh bar pada sumbu X yang masing masing mewakili setiap kelas pada label topik. Pada hasil visualisasi *chart* ini ditemukan bahwa kelas Feedback mendapatkan jumlah ulasan paling tinggi yaitu sebanyak 6.775 ulasan, sedangkan jumlah pada masing-masing kelas lain tidak lebih dari 1.000 ulasan. Terdapat kausalitas pada *chart* label sentimen dan *chart* label topik dikarenakan ulasan dengan kelas sentimen Positif sudah pasti memiliki kelas topik Feedback. Hal ini ditunjukkan dari jumlah kelas Positif paling tinggi pada *chart* jumlah label sentimen dan jumlah kelas Feedback pada *chart* jumlah label topik juga paling tinggi.

Chart terakhir yaitu *chart* label detail topik. Seperti yang telah dijelaskan pada subbab 3.2.4, label detail topik adalah bentuk pecahan yang lebih mendetail dari setiap kelas pada label topik. *Chart* ini menunjukkan banyaknya jumlah ulasan pada setiap kelas pada label detail topik. Dikarenakan banyaknya kelas pada label detail topik, *insight* label detail topik ini ditampilkan pada bentuk tabel dan jumlah ulasannya divisualisasikan dengan *bar chart* di dalam tabel. Dari hasil visualisasi, didapatkan jumlah ulasan terbanyak pada label detail topik adalah pada kelas Baik. *Chart* ini memiliki kausalitas dengan dua *chart* sebelumnya, *chart* label sentimen dan *chart* label topik. Hal ini dikarenakan pada kedua *chart* tersebut yang tertinggi adalah kelas Positif dan kelas Feedback dan jika ulasan tersebut memiliki kelas detail topik Baik, sudah pasti memiliki kelas topik Feedback dan kelas sentimen Positif.

Analisis – analisis ini dihasilkan dari Penulis sendiri dengan kemampuan dan pengalaman sendiri tanpa arahan dari pihak yang lebih ahli pada bidang analisis. Sehingga, *dashboard* dan analisisnya akan diberikan pada pihak *data analyst* perusahaan yang lebih ahli dan berpengalaman, untuk kemudian dapat dianalisis lebih dalam sebagai bahan evaluasi layanan aplikasi KAI Access.

BAB IV
REFLEKSI PELAKSANAAN MAGANG

4.1 Relevansi Akademik

Penulis mendapatkan banyak pengalaman dan pembelajaran dari segi akademik selama enam bulan melaksanakan kegiatan yang tentunya berdampak baik pada penulis. Beberapa pembelajaran terkait akademik yang didapatkan penulis antara lain *data labeling*, pembuatan *prototype*, dan pembuatan *dashboard* data ulasan KAI Access.

4.1.1 Data Labeling

Data labeling pada proyek data KAI Access ini cukup kompleks, tidak hanya dilabeli oleh satu label seperti yang banyak dilakukan, namun tiga label sekaligus dalam terhadap satu data. Label tersebut adalah sentimen, topik, dan detail topik. Ketiga label ini ditentukan atas permintaan mentor, yang ingin mengidentifikasi sentimen dan topik pada data ulasan serta ingin memudahkan identifikasi tren masalah pada data dengan menambah detail topik pada data.

Label sentimen, topik, dan detail topik ini tidak terdapat pada data mentah ulasan KAI Access, sehingga harus dilakukan secara manual menggunakan Google Sheets, membaca dan memahami teks ulasan satu persatu dan melabeli data tersebut dengan ketiga label. Rincian kelas dan parameter klasifikasi label ditentukan penulis bersama dengan mentor yang berperan sebagai Data Analyst dan lebih mengetahui urgensi pengelompokkan ulasan. Awalnya penulis dan mentor memiliki pendapat yang berbeda. Namun setelah menjelaskan sudut pandang terhadap pendapat masing-masing dan mempertimbangkan untuk hasil akhir yang baik, akhirnya keputusan akhir penentuan label dan kelasnya disepakati bersama dengan mengambil pendapat terbaik. Selama kegiatan *data labeling*, penulis sering kali bertanya dan berdiskusi kepada mentor karena banyaknya data yang ambigu untuk dikelompokkan kedalam suatu kelas. Penulis mendapatkan bantuan untuk menentukan kelas serta mendapat saran dari mentor untuk menangani data yang ambigu tersebut dengan melihat data dari sudut pandang yang lebih jauh yaitu sudut pandang pengguna.

Data labeling adalah proses yang memakan waktu paling lama pada proyek ini, memakan waktu total sektiar satu bulan setengah hanya untuk *data labeling* data ulasan KAI Access. Hal ini dikarenakan data ulasan yang bebas ditulis oleh pengguna aplikasi KAI Access terkadang

sulit untuk dipahami pengelompokannya sehingga banyak kebingungan dalam penentuan macam-macam kelasnya. Perubahan label secara total juga pernah dilakukan ditengah-tengah proses *data labeling* karena setelah ditinjau kelas label tidak sesuai dengan tujuan awal. Bagi penulis, proses inilah yang paling sulit. Maka dari itu, *data labeling* memberikan pembelajaran yang banyak bagi penulis mulai dari menentukan parameter yang sesuai dan adil untuk kelas label serta memahami data dan mengelopokkannya.

4.1.2 Pembuatan Prototype

Prototype klasifikasi data ulasan KAI Access ini adalah hasil akhir proyek ini, hasil akhir dari seluruh proses yang sebelumnya sudah dilakukan mulai dari pengumpulan data hingga pembuatan model klasifikasi. *Prototype* inilah yang akan digunakan oleh perusahaan dalam membantu mengklasifikasikan data ulasan kedepannya.

Penulis sebelumnya belum pernah membuat *prototype* untuk klasifikasi data seperti ini, maka dari itu penulis secara mandiri membangun *prototype* dengan bantuan berbagai referensi di internet. Menyusun alur *prototype*, melalui trial and error, hingga dapat bekerja dengan baik. Oleh karena itu, melalui kegiatan magang ini, pengalaman baru didapatkan oleh penulis yaitu membuat *prototype* dengan menggunakan model yang sudah disimpan.

4.1.3 Pembuatan Dashboard

Diakhir proyek, penulis ditugaskan untuk membuat *dashboard* yang memvisualisasikan hasil analisis dari data ulasan KAI Access. *Dashboard* ini diminta oleh mentor dibuat agar mentor juga pegawai lain yang akan melihat analisis dari data ulasan yang sudah dikumpulkan dan melihat tren dari analisisnya. Tidak ada arahan untuk menentukan apa saja *insight* yang harus digali oleh penulis, mentor menyerahkan penuh hal tersebut kepada penulis untuk mendapat *insight* penting sebanyak-banyaknya. *Data analyst* pada PT KAI sehari-hari membuat *dashboard* menggunakan *tools* Tableau, maka penulis pun membuatnya menggunakan Tableau.

Pembelajaran lainnya yang didapat yaitu belajar visualisasi khususnya *dashboard* pada Tableau. penulis hanya berpengalaman untuk melakukan visualisasi menggunakan *library* yang ada pada Python serta Google Data Studio. Sehingga, pada proyek ini, penulis dibimbing dan diarahkan oleh mentor untuk membuat *dashboard* visualisasi analisis data dengan menggunakan Tableau. Cukup sulit untuk mempelajari *tools* yang belum pernah dipakai, namun lambat laun penulis mulai terbiasa dan nyaman menggunakan Tableau. Banyak

kelebihan yang dimiliki Tableau dibandingkan dengan visualisasi menggunakan *library* Python ataupun Google Data Studio. Perbandingan tersebut dijelaskan pada Tabel 4.1.

Tabel 4.1 Perbandingan Tools dalam Analisis dan Visualisasi Data

Perbandingan	Tableau	Google Data Studio	Python Library
Visualisasi Data	Memiliki bayak opsi visualisasi	Visualisasi yang simpel	Memiliki banyak opsi visualisasi dari beberapa *library* (mengimpor beberapa *library* yang berbeda)
Platform	*Desktop* (mengunduh terlebih dahulu)	Web	*Dekstop* (Impor melalui IDE)
Kemampuan Integrasi	Terhubung dengan data *connectors* dan *database* serta berkas lokal	Terhubung dengan banyak *public dataset* Google dan berkas lokal	Terhubung dengan berkas lokal
Pembaharuan *Dashboard*	*Real-time dashboard* dan dapat menangani data yang kompleks	*Real-time dashboard* namun menjadi lambat ketika menangani data yang besar	-
Kemampuan dibagikan dan kolaborasi	Memiliki lebih banyak opsi kolaborasi melalui Cloud, lokal, atau *server* Tableau	Serupa dengan Google *tools* lainnya yang menyediakan fitur kolaborasi serta tersedia juga data *blending*.	-

4.2 Pembelajaran Magang

Banyak sekali manfaat yang penulis dapatkan selama kegiatan magang dikarenakan ini pertama kalinya penulis melakukan kegiatan magang. Hal yang paling berkesan bagi penulis adalah pengalaman mengolah *dataset* internal perusahaan yang lebih kompleks jika dibandingkan dengan *dataset* publik yang pernah penulis olah, maka penulis banyak belajar mengolah data perusahaan beserta dengan kompleksitasnya dengan banyak arsitektur. Banyak arsitektur model yang dapat menjawab permasalahan yang didapatkan pada perusahaan, salah satunya MLP dan BiLSTM untuk pengklasifikasian data ulasan KAI Access. Adapun *tools* yang dipakai dalam pengerjaan proyek ini namun belum pernah penulis gunakan dalam proses mengolah data sebelumnya, sehingga penulis dapat mempelajari *tools* baru tersebut yaitu Tableau. Penulis juga mendapatkan pengalaman bagaimana menghasilkan suatu solusi dari permintaan *stakeholder*, dalam proyek ini yaitu pembuatan *prototype*. Mentor menjelaskan tujuan yang diharapkan dapat dicapai dengan membuat sesuatu menggunakan Machine Learning dan akhirnya penulis memutuskan untuk membuat *prototype* klasifikasi data KAI Access.

Pada awal kegiatan *planning meeting*, mentor berharap model klasifikasi yang dibuat dapat langsung diaplikasikan secara otomatis kepada data ulasan yang masuk setiap waktu secara *real-time*. Hal tersebut bisa dilakukan jika dengan memasukkan model klasifikasi kedalam arsitektur sistem KAI Access. Namun karena kurangnya pengalaman penulis dalam model *deployment* dan *role data scientist* secara umum tidak mencakup hal tersebut, maka hal ini menjadi kendala yang akhirnya diputuskan hanya membuat *prototype* untuk mengklasifikasikan berkas data yang dimasukkan manual oleh pengguna. Namun akan memungkinkan kedepannya jika PT KAI telah mempunyai pegawai dengan *role machine learning/*AI *engineer*, proyek ini akan dikembangkan sebagaimana harapan di awal proyek.

Target penyelesaian tugas-tugas yang sudah penulis susun pada awal kegiatan magang mengalami ketidaksesuaian karena terjadinya keterlambatan pengerjaan pada suatu tugas. Tugas tersebut adalah *data labeling*. *Data labelling* memakan waktu yang lebih lama dibandingkan dengan yang sudah ditargetkan karena tidak konsistennya parameter dalam pembuatan kelas-kelas untuk klasifikasi serta beberapa data yang bersifat ambigu sehingga perlu waktu lama untuk menentukan kelas klasifikasinya. Karena hambatan ini, penulis sempat merombak kelas-kelas data kembali dari awal dan juga melabeli data dari awal. Hal ini juga dikarenakan pengalaman penulis yang belum pernah melabeli data yang memiliki tingkat kompleksitas data seperti data KAI Access ini sebelumnya.

BAB V
PENUTUP

5.1 Kesimpulan

Proyek data ulasan KAI Access ini berhasil mengembangkan tiga arsitektur model yang berbeda untuk tiga klasifikasi yang berbeda yaitu sentimen, topik, dan detail topik atau dapat disebut *multi-class multi-label classification*. Arsitektur MLP untuk klasifikasi sentimen ulasan KAI Access memiliki akurasi di atas 80% yaitu sebesar 87,53%, model dapat mengklasifikasikan sentimen negatif, positif, ataupun netral yang ada pada data ulasan cukup baik. Algoritma BiLSTM untuk klasifikasi topik dengan akurasi sebesar 79,10%, data dapat terklasifikasikan ke dalam kelompok topiknya masing-masing dengan cukup baik. Terakhir, akurasi model klasifikasi detail topik dengan algoritma BiLSTM berada pada angka 64,85% yang mana lebih rendah dibandingkan dengan dua klasifikasi sebelumnya.

Dashboard visualisasi data ulasan KAI Access pun berhasil dibuat dengan menampilkan delapan buah visualisasi dari *insight* yang berbeda yang cukup menjelaskan performa aplikasi berdasarkan *feedback* dari pengguna. Kemudian dihasilkan analisis-analisis yang dapat membantu sebagai pendukung evaluasi aplikasi KAI Access.

Secara keseluruhan, hasil produk telah memenuhi tujuan dilakukannya proyek ini yaitu telah dikembangkannya tiga model klasifikasi yaitu klasifikasi sentimen, topik, dan detail topik. *Prototype* yang dibangun juga telah dapat mengklasifikasikan secara otomatis dan cepat namun terdapat kekurangan dalam keakuratan pengklasifikasiannya yang belum sempurna disebabkan oleh beberapa kekurangan seperti tidak meratanya jumlah data pada setiap kelas dan arsitektur model yang dibuat cukup sederhana.

5.2 Saran

Hasil produk dari proyek klasifikasi data ulasan KAI Access memiliki kekurangan yaitu hasil klasifikasi yang tidak seluruhnya tepat, masih terdapat banyak data yang hasil klasifikasinya tidak sesuai kelas yang seharusnya. Hal ini dikarenakan kompleksitas klasifikasi yang tinggi, *layer* penyusun model masih sederhana, serta tidak meratanya jumlah data latih pada setiap kelas. Maka dari itu diharapkan jika proyek ini dilanjutkan kedepannya, hal yang perlu ditingkatkan dari proyek ini adalah persebaran jumlah datanya pada setiap kelas merata agar model dapat mempelajari data dengan baik. Harapan lainnya jika proyek ini

dikembangkan adalah peningkatan arsitektur model yang lebih dalam, yang memungkinkan hasil akurasi yang lebih baik pada klasifikasi yang kompleks ini. Selain itu, pada *prototyping* proyek ini, *vocabulary* untuk digunakan dalam *tokenize* data mentah dalam *prototype* masih dihasilkan dengan *generate* ulang secara manual jika *prototype* itu dijalankan. Jikalau proyek ini dapat dikembangkan, diharapkan *vocabulary training data* dapat disimpan dalam sebuah *file* dan pada proses *prototyping* hanya perlu dimasukkan. Hal ini berguna untuk mengurangi waktu *running prototype*.

DAFTAR PUSTAKA

Abelein, U., Sharp, H., & Peach, B. (2013). Does Involving User in Software Development Really Influence System Success? *IEEE Software*, 23.

Allen, M., & Cervo, D. (2015). *Multi-Domain Master Data Management.* Morgan Kaufmann Publisher Inc.

Ashgar, M. Z., Lajis, A., Alam, M. M., Rahmat, M. K., Nasir, H. M., Ahmad, H., . . . Albogamy, F. R. (2022). A Deep Neural Network Model for the Detection and Classification of Emotions from Textual Content. *Hindawi Complexity*.

Dekel, O., & Shamir, O. (2010). Multiclass-Multilabel Classification with More Classes than Examples. *Thirteenth International Conference on Artificial Intelligence and Statistics*.

Dellia, P., & Tjahyanto, A. (2017). Tax Complaints Classification on Twitter Using Text Mining. *IPTEK Journal of Science*.

Fadli, H. F., & Hidayatullah, A. F. (2020). Identifikasi Cyberbullying pada Media Sosial Twitter Menggunakan Metode LSTM dan BiLSTM. *Automata Informatika Universitas Islam Indonesia*.

Goldbreg, Y. (2017). Neural Network Methods for Natural Language Processing. *Synthesis Lectures on Human Language Technologies*.

Gupta, D. (2020, January 30). *Fundamentals of Deep Learning – Activation Functions and When to Use Them?* Diambil kembali dari Analytics Vidhya: https://www.analyticsvidhya.com/blog/2020/01/fundamentals-deep-learning-activation-functions-when-to-use-them/#:~:text=ReLU%20stands%20for%20Rectified%20Linear,neurons%20at%20the%20same%20time.

Indonesia, P. K. (2022). *Google Play Store*. Diambil kembali dari KAI Access: https://play.google.com/store/apps/details?id=com.kai.kaiticketing&hl=en&gl=US

KAI Company Profile, K. (2021). *KAI Company Profile*. Diambil kembali dari PT Kereta Api Indonesia: https://www.kai.id/static/company-profile/company_profile_2021.pdf

KAI Heritage, K. (2017). *Sejarah Perkeretaapian*. Diambil kembali dari PT Kereta Api Indonesia: https://heritage.kai.id/page/sejarah-perkeretapian

KAI, P. (2017). *Situs Resmi PT Kereta Api Indonesia*. Diambil kembali dari PT Kereta Api Indonesia: https://www.kai.id/corporate/page/10

KAI, P. (2022). *PPD Daerah Operasional*. Diambil kembali dari E-PPID PT Kereta Api Indonesia: https://ppid.kai.id/?_it8tnz=T1RBeE1EQXdNREF3&_8dnts=Y0hCcFpBPT0=&_8ith=TlRBPQ==

Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. *Cornell University Arxiv*.

Livingston, S. J., Tamil Selvi, B. S., Thabeetha, M., Grena, C. P., & Jenifer, C. S. (2019). A Neural Network Based Approach for Sentimental Analysis on Amazon Product Reviews. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*.

Mayapasim1. (2012). *Kantor Pusat PT. KAI Bandung*. Diambil kembali dari Ahmed Kreatif: https://mayapasim1.wordpress.com/kantor-pusat-pt-kai- bandung/

Md.Kowsher, Tahabilder, A., Sanjid, M. I., Prottasha, N. J., Uddin, M., Hossain, M., & Jilani, M. K. (2014). LSTM-ANN & BiLSTM-ANN: Hybrid deep learning models for enhanced classification accuracy. *Procedia Computer Science*.

Mohammad, S. M. (2016). A Practical Guide to Sentiment Annotation: Challenges and Solutions. *National Research Council Canada*, 179.

Peltarion. (2022). *Global Average Pooling 1D*. Diambil kembali dari Knowledeg Center Peltarion: https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/blocks/global-average-pooling-1d

Qorita, A. K., & Rahma, F. (2021). Analisis Sentimen Berdasarkan Aspek pada Tempat Wisata di Daerah Istimewa Yogyakarta Menggunakan Metode Multinomial Naïve Bayes. *Automata UII*.

S, A., & P, C. (2019). Chapter Fourteen - Energy-efficient edge based real-time healthcare support system. *Elseview*.

Samandianfard, S., Hashemi, S., Kargar, K., Izadyar, M., Mostafaeipour, A., Mosavi, A., . . . Shamshirband, S. (2020). Wind speed prediction using a hybrid model of the multi-layer perceptron and whale optimization algorithm. *Elsevier*.

TensorFlow. (2022). *Adam Optimizer*. Diambil kembali dari TensorFlow: https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/Adam

TensorFlow. (2022). *Basic text classification*. (TensorFlow) Dipetik July 9, 2022, dari https://www.tensorflow.org/tutorials/keras/text_classification#create_the_model

TensorFlow. (2022). *Callbacks Model Check Point*. Diambil kembali dari TensorFlow: https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/ModelCheckpoint

TensorFlow. (2022). *TensorFlow Word Embeddings*. Diambil kembali dari TensorFlow: https://www.tensorflow.org/text/guide/word_embeddings

TensorFlow. (2022). *Text Classification With an RNN*. Diambil kembali dari TensorFlow: https://www.tensorflow.org/text/tutorials/text_classification_rnn#create_the_model

Tripathy, A., Agrawal, A., & Rath, S. K. (2016). Classification of sentiment reviews using n-gram machine learning approach. *Expert System with Application*.

Venkatesh, S., Moffat, D., & Miranda, E. R. (2022). Word Embeddings for Automatic Equalization in Audio Mixing. *Journal of Audio Engineering EPSRC*.

Walters, A. G. (2019). *Classify Sentences via a Multilayer Perceptron (MLP)*. Diambil kembali dari Austin G. Walters: https://austingwalters.com/classify-sentences-via-a-multilayer-perceptron-mlp/

Zhang, M. L., Li, Y. K., Liu, X. Y., & Geng, X. (2017). Binary Relevance for Multi-Label Learning. *Higher Education Press and Springer-verlag Berlilne Heidelberg*.

Zhou, Q., Zhang, Z., & Wu, H. (2018). NLP at IEST 2018: BiLSTM-Attention and LSTM-Attention via Soft Voting Emotion Classification. *Computational Approches to Subjectivity, Sentiment and Social Media Analysis*.

LAMPIRAN

```
import pandas as pd
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences

oov_tok = '<OOV_TOK>'
vocab_size = 100000
padding_type = 'post'
trunc_type = 'post'
max_length = 150
test_portion = 0.2
embedding_dim = 16


data_vocab = pd.read_csv('training_data.csv')
data_vocab.review_text=data_vocab.review_text.astype(str)
data_vocab.drop(columns=['app_ver_name', 'reviewer_language','device',
'review_month', 'star_rating'],
                inplace = True)
#load data mentah
predict_file = input('Masukkan path file data yang ingin diklasifikasikan\nJika
pada folder yang '
                     'sama masukkan nama filenya saja (format .csv) \n Masukkan
disini: ')
data_predict = pd.read_csv(predict_file, delimiter=';')

#nama data hasil
result_file = input ('Masukkan nama file baru untuk menyimpan hasil klasifikasi
(format .csv)\n '
                     'Masukkan disini: ')

#load saved model
sent_model = load_model('nonull_model_sent-nonumber.h5')
topic_model = load_model('nonull_model_topic-nonumber.h5')
detail_model = load_model('nonull_model_detail-nonumber.h5')


#data cleaning function
def preprocessing(data_predict):
    # Cleaning punctuation, space, emoji, capital letter
    # Remove punctuation and emojis
    data_predict['Review Text'] = data_predict['Review Text']
                                            .str.replace('[^\w\s]', '')
    # Lowering Case
    data_predict['Review Text'] = data_predict['Review Text'].str.lower()
    # remove URLs
    data_predict['Review Text'] = data_predict['Review Text']
                                            .replace(r'http\S+', '',
                                                     regex=True)
                                            .replace(r'www\S+', '', regex=True)
    # remove newlines
    data_predict['Review Text'] = data_predict['Review Text']
                                            .str.replace('\n', ' ')
    # replace two space to one
    data_predict['Review Text'] = data_predict['Review Text']
                                            .str.replace('\s\s+', ' ',
                                                         regex=True)
    # remove leading space
    data_predict['Review Text'] = data_predict['Review Text']
                                            .replace('^ +| +$', '', regex=True)
    #remove number
    data_predict['Review Text'] = data_predict['Review Text']
```

```
                                                .str.replace('\d+', '')
    #remove single letter
    data_predict['Review Text'] = data_predict['Review Text']
                                                .str.replace(r'\b\w\b', '')
                                                .str.replace(r'\s+', ' ')
    print(data_predict[data_predict['Review Text'].isnull()])
    #remove data that have no review text
    data_predict = data_predict.drop(data_predict[data_predict['Review Text']
                                                                .isnull()].index)
    # manipulate NaN values
    data_predict.loc[(data_predict['App Version Name'].isnull()),
                                    'App Version Name'] = 'unknown'
    data_predict.loc[(data_predict.Device.isnull()), 'device'] = 'unknown'

    return data_predict


#tokenizing training data
def process_vocab(data_vocab):
    train_set = data_vocab
    text_train = train_set['review_text']
    sent_train = train_set['sentiment']
    topic_train = train_set['topic']
    detail_train = train_set['detail_topic']

    tokenizer = Tokenizer(vocab_size, oov_token=oov_tok)
    tokenizer.fit_on_texts(text_train)
    text_word_index = tokenizer.word_index

    sent_tokenizer = Tokenizer()
    sent_tokenizer.fit_on_texts(sent_train)
    sent_word_index = sent_tokenizer.word_index

    topic_tokenizer = Tokenizer()
    topic_tokenizer.fit_on_texts(topic_train)
    topic_word_index = topic_tokenizer.word_index

    detail_tokenizer = Tokenizer()
    detail_tokenizer.fit_on_texts(detail_train)
    detail_word_index = detail_tokenizer.word_index

    print("setences dict:")
    print(text_word_index)
    print("sentiment dict:")
    print(sent_word_index)
    print("topic dict:")
    print(topic_word_index)
    print("detail dict:")
    print(detail_word_index)

    return tokenizer, text_word_index, sent_tokenizer, sent_word_index,\
           topic_tokenizer, topic_word_index, detail_tokenizer, detail_word_index


#data tokenizing and sentiment classification
def predict_sentiment(data_predict):
    input = np.array(data_predict['Review Text'])
    prediction = sent_model
                 .predict(np.array(pad_sequences(tokenizer
                                                 .texts_to_sequences(input),
                                                 padding=padding_type,
                                                 maxlen=max_length,
                                                 truncating=trunc_type)))
    list_result = []
    for row in range(len(prediction)):
        result = prediction[row].tolist().index(np.max(prediction[row]))
        list_result.append(result)
```

```
    pre_result = []
    for s in list_result:
        pre_result.append(sent_tokenizer.index_word[s])

    data_predict['Sentiment Prediction'] = np.array(pre_result)
    print(data_predict[['Review Text', 'Sentiment Prediction']])
    return data_predict

#data tokenizing and topic classification
def predict_topic(data_predict):
    input_text = np.array(data_predict['Review Text'])
    input_text = pad_sequences(tokenizer.texts_to_sequences(input_text),
                                        padding=padding_type,maxlen=max_length,
                                        truncating=trunc_type)
    input_sent = np.array(data_predict['Sentiment Prediction'])
    input_sent = pad_sequences(sent_tokenizer
                               .texts_to_sequences(input_sent), maxlen=1)
    input = np.concatenate([input_text, input_sent], axis=1)
    prediction = topic_model.predict(np.array(input))
    list_result = []
    for row in range(len(prediction)):
        result = prediction[row].tolist().index(np.max(prediction[row]))
        list_result.append(result)
    pre_result = []
    for s in list_result:
        pre_result.append(topic_tokenizer.index_word[s])

    data_predict['Topic Prediction'] = np.array(pre_result)
    print(data_predict[['Review Text', 'Sentiment Prediction',
                        'Topic Prediction']])
    return data_predict


#data tokenizing and detail topic classification
def predict_detail(data_predict):
    input_text = np.array(data_predict['Review Text'])
    input_text = pad_sequences(tokenizer.texts_to_sequences(input_text),
                               padding=padding_type, maxlen=max_length,
                               truncating=trunc_type)
    input_sent = np.array(data_predict['Sentiment Prediction'])
    input_sent = pad_sequences(sent_tokenizer.texts_to_sequences(input_sent),
                               maxlen=1, padding=padding_type,
                               truncating=trunc_type)
    input_topic = np.array(data_predict['Topic Prediction'])
    input_topic = pad_sequences(topic_tokenizer.texts_to_sequences(input_topic),
                                maxlen=1, padding=padding_type,
                                truncating=trunc_type)
    input = np.concatenate([input_text, input_sent, input_topic], axis=1)
    prediction = detail_model.predict(np.array(input))
    list_result = []
    for row in range(len(prediction)):
        result = prediction[row].tolist().index(np.max(prediction[row]))
        list_result.append(result)
    preresult = []

    for aa in list_result:
        preresult.append(detail_tokenizer.index_word[aa])

    data_predict['Detail Prediction'] = np.array(preresult)
    print(data_predict[['Review Text', 'Sentiment Prediction',
                        'Topic Prediction', 'Detail Prediction']])
    return data_predict


if __name__ == '__main__':
    # preprocessing data mentah baru untuk dibersihkan
    data_predict = preprocessing(data_predict)
```

```
    # tokenizing data lama untuk list vocabulary token
    tokenizer, text_word_index, sent_tokenizer, sent_word_index, topic_tokenizer,
topic_word_index, detail_tokenizer, detail_word_index = process_vocab(data_vocab)
    # data baru diklasifikasikan untuk label sentimen
    data_predict = predict_sentiment(data_predict)
    # data baru diklasifikasikan untuk label topik
    data_predict = predict_topic(data_predict)
    # data baru diklasifikasikan untuk data detail topik
    data_predict = predict_detail(data_predict)
    data_predict.to_csv(result_file)
```

UNIVERSITAS ISLAM INDONESIA