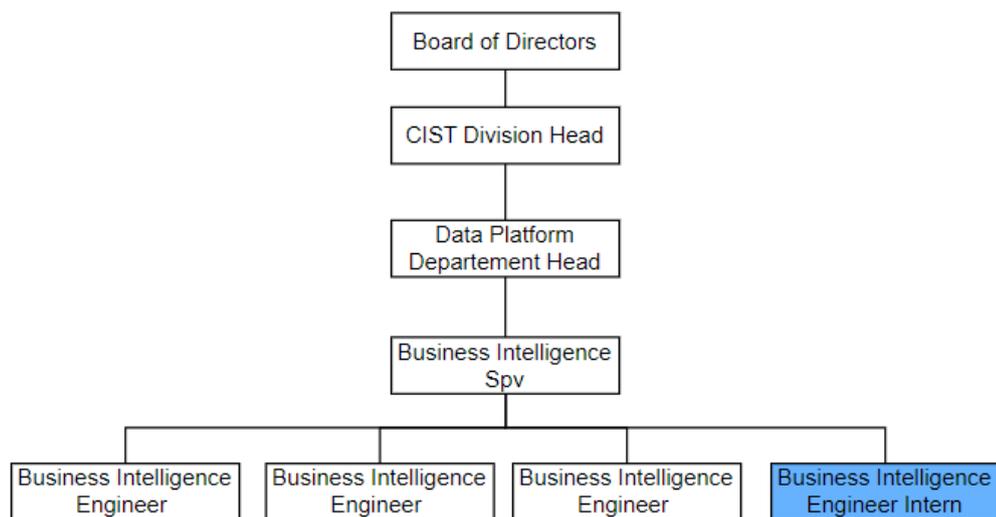


BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Dalam pelaksanaan kegiatan kerja magang di PT Serasi Autoraya, mahasiswa menempati kedudukan sebagai *Intern* pada departemen *Data Platform*. Pada departemen *Data Platform* tersebut, mahasiswa berada pada tim *Business Intelligence*.



Gambar 3. 1 Kedudukan Pekerja Magang

Gambar 3. 1 merupakan bagan yang menggambarkan kedudukan pekerja magang di PT Serasi Autoraya. Pada tim *Business Intelligence* terdapat satu supervisor yang memimpin dan mengkoordinasikan tim dengan tiga karyawan dibawahnya. Pekerja magang juga dipimpin oleh *supervisor* dan juga dibimbing oleh mentor yang merupakan salah satu anggota tim *Business Intelligence*.

Untuk alur koordinasi pengerjaan proyek, pekerja magang akan membuat perencanaan proyek dibantu dengan mentor magang. Setelah terkumpul beberapa topik yang akan diajukan, pekerja magang melakukan konsultasi dengan mentor untuk meminta arahan dan saran yang dapat dikembangkan. Setelah didapat proyek

yang memungkinkan, pekerja magang melakukan konsultasi dengan mentor dan *supervisor* tentang proyek yang akan diajukan.

Setelah mendapat masukan dan saran dari *supervisor*, dilakukan perbaikan terhadap rencana proyek. Rencana proyek yang sudah jadi disusun dalam bentuk Power Point singkat. Kemudian Power Point tersebut dipresentasikan ke kepala departemen *Data Platform* untuk mendapat persetujuan. Setelah persetujuan didapat, maka proyek tersebut dapat dilakukan sesuai dengan rencana yang telah dirancang.

3.2 Tugas dan Uraian Kerja Magang

Kegiatan kerja magang di PT Serasi Autoraya yang dilaksanakan selama 5 bulan dari 8 Januari 2024 sampai dengan 7 Juni 2024 akan dijelaskan secara singkat pada Tabel 3. 1 berikut. Kemudian kegiatan magang yang dilakukan akan dijelaskan secara rinci setelahnya.

Tabel 3. 1 Rincian Kerja Magang

No.	Kegiatan	Mulai	Selesai
1	Kegiatan Onboarding: Pengenalan Perusahaan dan alur kerja tim		
1.a	Perkenalan perusahaan & alur kerja departemen.	08/01/2024	09/01/2024
1.b	Pemberian akses akun pendukung pekerjaan	09/01/2024	10/01/2024
1.c	Pembelajaran tools yang akan dipakai.	09/01/2024	12/01/2024
2	Pembuatan dokumentasi User Access Matrix dan Role Permission Matrix.		
2.a	Pembuatan User Access Matrix (UAM).	15/01/2024	19/01/2024
2.b	Pembuatan Role Permission Matrix (RPM).	22/01/2024	23/01/2024
2.c	Pembaruan dashboard Monitor.	24/01/2024	31/01/2024
3	Pengajuan proposal proyek magang.		
3.a	Pembuatan PPT proposal proyek magang.	15/01/2024	26/01/2024
3.b	Presentasi proyek yang akan dikerjakan selama magang.	29/01/2024	29/01/2024
4	Melakukan data scraping dari review pada App Store dan Play Store untuk aplikasi TracToGo dan Ibid Auction.		
4.a	Melakukan data scraping pada App Store	01/02/2024	02/02/2024
4.b	Melakukan data scraping pada Play Store	01/02/2024	02/02/2024
5	Melakukan pembersihan pada hasil data scraping dan labelling untuk sentiment.		

No.	Kegiatan	Mulai	Selesai
5.a	Melakukan pelabelan sentiment manual untuk seluruh data ulasan.	05/02/2024	06/02/2024
5.b	Membuat koding untuk membersihkan data ulasan.	07/02/2024	15/02/2024
6	Membuat pemodelan prediksi sentiment menggunakan <i>machine learning</i>.		
6.a	Membuat word vectorizer fasttext untuk mengoptimalkan model machine learning.	19/03/2024	28/03/2024
6.b	Membuat model machine learning untuk klasifikasi ulasan berdasarkan sentiment.	15/02/2024	04/03/2024
6.c	Melakukan ekspor data yang sudah siap untuk dipakai.	04/03/2024	04/03/2024
7	Pembuatan dashboard analisis sentiment menggunakan aplikasi Power BI.		
7.a	Penyambungan power BI dengan data hasil klasifikasi.	05/03/2024	06/03/2024
7.b	Pembuatan dashboard analisis sentiment.	06/03/2024	08/03/2024
8	Migrasi Dashboard dan code python ke cloud.		
8.a	Melakukan upload kode python pada databricks.	11/03/2024	11/03/2024
8.b	Melakukan testing kode pada databricks.	12/03/2024	22/04/2024
8.c	Melakukan penyambungan data hasil klasifikasi pada cloud ke power BI.	22/04/2024	24/04/2024
8.d	Melakukan publish power BI Dashboard.	25/04/2024	25/04/2024
9	Pembuatan Data Dictionary		
9.a	Request akses ke server	30/04/2024	02/05/2024
9.b	Pembuatan dokumen data dictionary	03/05/2024	07/06/2024
10	Review magang		
10.a	Melakukan Presentasi hasil kerja selama setengah periode pertama.	18/03/2024	18/03/2024
10.b	<i>Review End-Intern dan Exit Interview</i>	06/06/2024	06/06/2024

Berdasarkan Tabel 3. 1 yang merupakan uraian kerja magang secara singkat, terdapat kegiatan pembuatan UAM dan RPM, pembuatan dashboard analisis sentiment, juga pembuatan data dictionary sebagai *business support*. Selain itu pekerja magang juga menjalani review bersama mentor sebanyak dua kali pada saat pertengahan kegiatan magang dan akhir kegiatan magang.

3.2.1 Kegiatan *Onboarding* : Pengenalan Perusahaan dan alur kerja tim

Pada hari pertama kegiatan magang di PT Serasi Autoraya merupakan kegiatan *onboarding* untuk pekerja magang. Kegiatan yang dilakukan pertama

kali merupakan pertemuan dengan perwakilan tim HRD yang merekrut dan membicarakan beberapa hal tentang kebutuhan dan pekerjaan di kantor nantinya. Setelah itu tim HRD mempertemukan pekerja magang dengan kepala departemen Data Platform untuk melakukan pengenalan perusahaan dan tim yang akan dimasuki nantinya.

3.2.1.1 Perkenalan perusahaan & alur kerja departemen.

Pada pertemuan pertama dengan kepala departemen Data Platform, dijelaskan tentang pekerjaan apa saja yang akan dilakukan selama magang di PT Serasi Autoraya sebagai Business Intelligence Engineer Intern.

Setelah itu, para pekerja magang diperkenalkan dengan anggota-anggota yang terlibat dalam Departemen Data Platform. Mulai dari tim *Data Engineer*, *Artificial Intelligence Engineer*, *Business Intelligence Engineer*, dan *Data Operation*. Setiap anggota memiliki peran yang unik dalam memastikan operasi yang lancar dan efisien dari departemen ini.

Pekerja magang memiliki kesempatan untuk bertemu dengan anggota departemen yang akan menjadi mentor dan supervisi selama masa magang berlangsung. Mentor memberikan arahan dan bimbingan yang dapat membantu pemahaman yang lebih dalam tentang tugas dan tanggung jawab pekerja magang.

Mentor memberikan penjelasan yang komprehensif tentang alur kerja departemen. Penjelasannya menguraikan bagaimana data masuk ke sistem, diproses, dianalisis, dan digunakan untuk menghasilkan wawasan yang bermanfaat bagi perusahaan. Penjelasan ini membantu saya memahami kontribusi masing-masing tim dari tim *data engineer* sampai *data operation* dalam menciptakan lingkungan kerja yang terintegrasi.

Dimulai dari tim *Data Engineer*, tim ini bertanggung jawab atas pengelolaan infrastruktur data, pemrosesan data mentah menjadi bentuk yang lebih terstruktur, dan memastikan ketersediaan dan integritas data yang nantinya akan digunakan oleh tim *AI Engineer* maupun *BI Engineer*.

Selanjutnya tim *AI Engineer*, tim ini fokus pada pengembangan dan implementasi model kecerdasan buatan untuk menghasilkan wawasan yang dapat diambil dari data perusahaan.

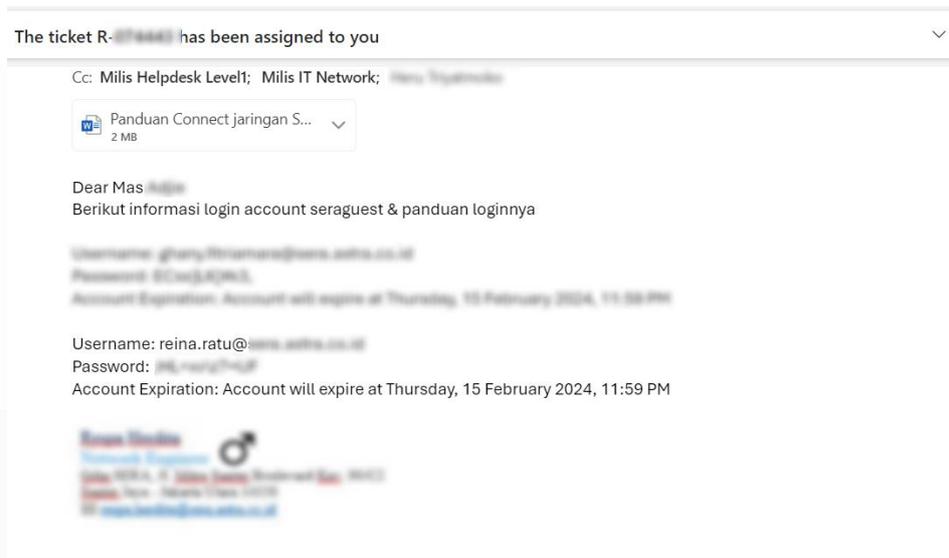
Lalu terdapat tim *Business Intelligence Engineer*, tim ini fokus pada pembuatan dashboard, laporan, dan alat visualisasi data untuk memberikan wawasan yang mudah dipahami kepada user dari department lainnya.

Terakhir tim *Data Operation*, tim ini bertanggung jawab atas manajemen operasional data, seperti pengelolaan database oleh *database administrator*.

3.2.1.2 Pemberian akses akun pendukung pekerjaan

Akses yang diberikan kepada pekerja magang terdapat beberapa aspek, seperti akses wifi dan akun Microsoft yang dapat digunakan dalam beberapa aplikasi Microsoft seperti Microsoft Azure, Power BI, Outlook, dan Microsoft Teams.

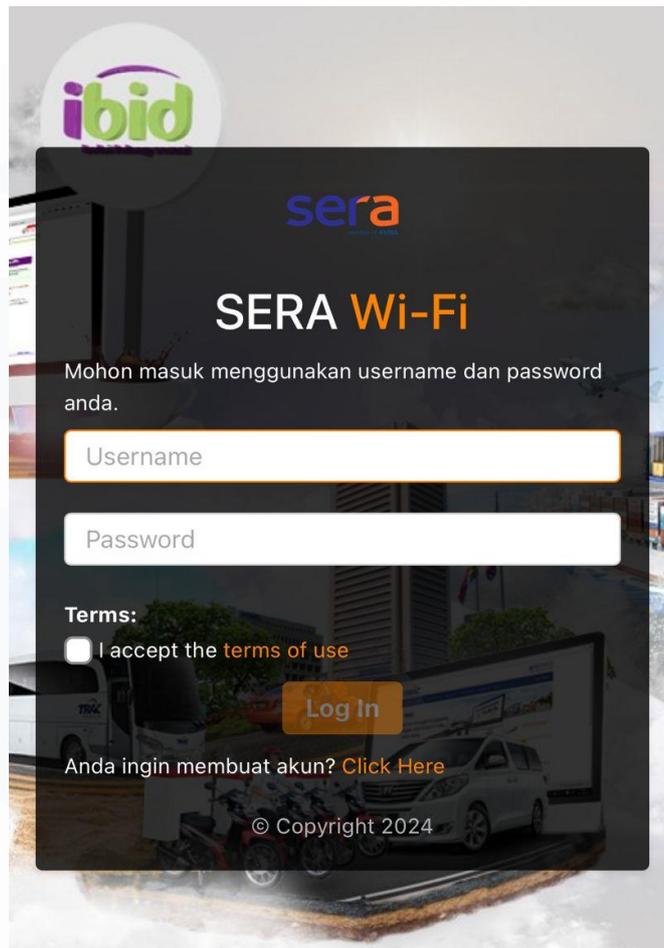
Request Akses wifi kantor dibantu oleh mentor dengan membuat form email yang akan dikirim ke tim helpdesk PT Serasi Autoraya. Setelah *request* diterima, tim helpdesk akan memberikan akun dan password untuk login pada halaman login wifi.



Gambar 3. 2 Pemberian Akun Wifi

Gambar 3. 2 merupakan balasan berupa email dari *request* wifi yang diminta. Email berisikan username akun, password, dan tanggal expired akun yang sudah diberikan. Akun yang diberikan dapat digunakan selama sebulan. Setelah masa aktif akun sudah habis, pekerja magang dapat melakukan *request* perpanjangan masa aktif wifi kembali melalui email ke helpdesk.





Gambar 3. 3 Tampilan Halaman Login Wifi

Gambar 3. 3 merupakan tampilan dari halaman login wifi pada PT Serasi Autoraya. Untuk mendapatkan akses wifi, maka akan digunakan akun yang sudah diberikan melalui email oleh tim helpdesk.

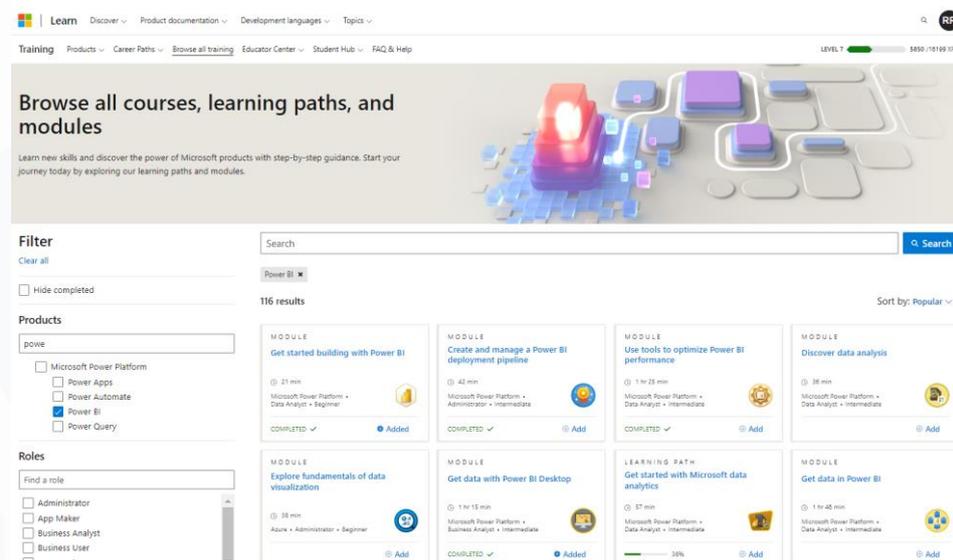
3.2.1.3 Pembelajaran tools yang akan dipakai.

Sebelum memulai pekerjaan sebagai seorang BI engineer Intern, pekerja magang melakukan pembelajaran terhadap tools yang akan digunakan selama magang berlangsung.



Gambar 3. 4 Power BI

Gambar 3. 4 merupakan logo dari tools utama, yaitu Power BI, yang akan digunakan selama magang di PT Serasi Autoraya sebagai seorang *Business Intelligence Engineer Intern*. Pembelajaran Power BI dilakukan dengan mentor dan juga melalui platform Microsoft Learn seperti pada Gambar 3. 5. Hal ini termasuk pemahaman tentang bagaimana memanfaatkan fitur-fitur dalam Power BI untuk menghasilkan visualisasi data yang informatif dan mudah dimengerti.



Gambar 3. 5 Tampilan Microsoft Learn

Selain mempelajari visualisasi pada Power BI, dipelajari juga DAX query yang merupakan bahasa yang digunakan dalam Power BI untuk melakukan manipulasi dan analisis data. Pemahaman yang kuat tentang

DAX query dapat memungkinkan untuk menggali lebih dalam ke dalam dataset, menerapkan logika bisnis kompleks, dan menghasilkan wawasan yang berharga bagi tim dan user dalam perusahaan. Dengan demikian, hal tersebut dapat meningkatkan efektivitas analisis data dan membuat keputusan yang lebih baik didukung oleh data.

Langkah berikutnya setelah mempelajari Power BI dan DAX query adalah melakukan latihan menggunakan data dummy. Latihan ini dapat membantu dalam mempraktikkan penggunaan Power BI dan penerapan DAX query dalam proyek yang akan dilakukan nantinya. Melalui latihan tersebut, keterampilan pekerja magang akan lebih terlatih, juga dapat memperdalam pemahaman tentang fungsi dari tools yang akan digunakan dalam proyek nantinya. Dengan demikian, pekerja magang dapat memberikan kontribusi yang berarti dalam mencapai tujuan departemen dan perusahaan secara keseluruhan.

Selain menggunakan Power BI Desktop, pekerja magang juga menggunakan beberapa *tools* pendukung lainnya seperti Microsoft Excel dan Databricks.



Gambar 3. 6 Logo Aplikasi Microsoft Excel

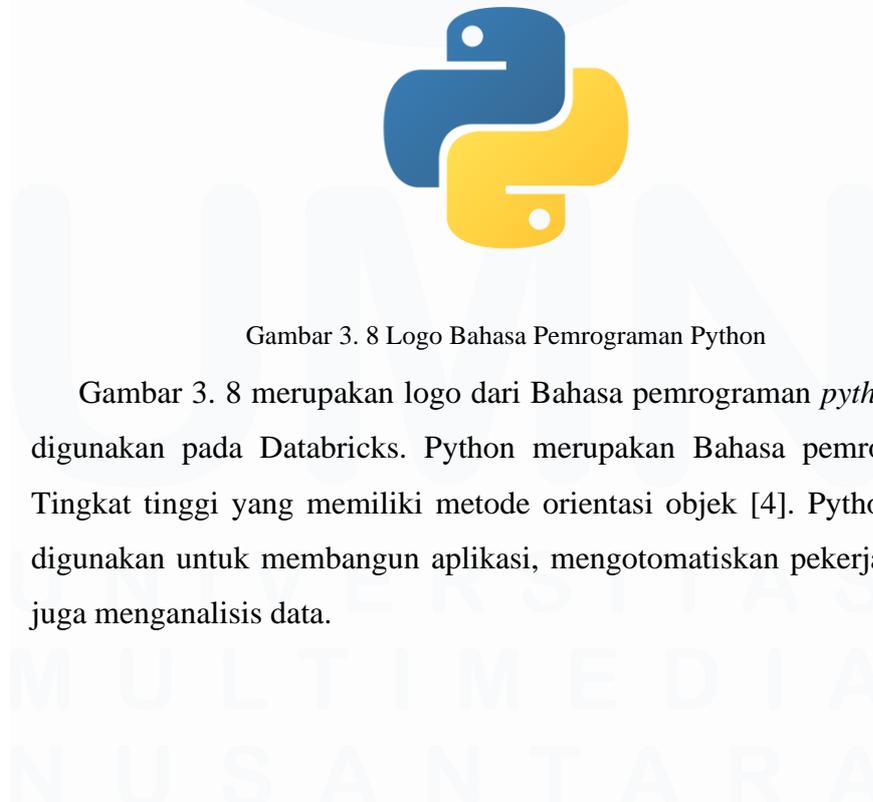
Gambar 3. 6 merupakan logo dari aplikasi Microsoft Excel yang digunakan selama kegiatan magang berlangsung. Microsoft Excel merupakan aplikasi dari Microsoft Office yang berupa program spreadsheet [2]. Microsoft Excel ini dapat digunakan untuk menganalisis data,

membuat rumus dan perhitungan pada data yang berbentuk lembaran tabel (sreadsheet).



Gambar 3. 7 Logo Aplikasi Databricks

Gambar 3. 7 merupakan logo dari aplikasi Databricks yang digunakan selama kegiatan magang berlangsung. Databricks merupakan platform yang dapat digunakan untuk melakukan analitik di perusahaan dengan skala besar yang terintegrasi dengan penyimpanan *cloud* [3].



Gambar 3. 8 Logo Bahasa Pemrograman Python

Gambar 3. 8 merupakan logo dari Bahasa pemrograman *python* yang digunakan pada Databricks. Python merupakan Bahasa pemrograman Tingkat tinggi yang memiliki metode orientasi objek [4]. Python dapat digunakan untuk membangun aplikasi, mengotomatiskan pekerjaan, dan juga menganalisis data.

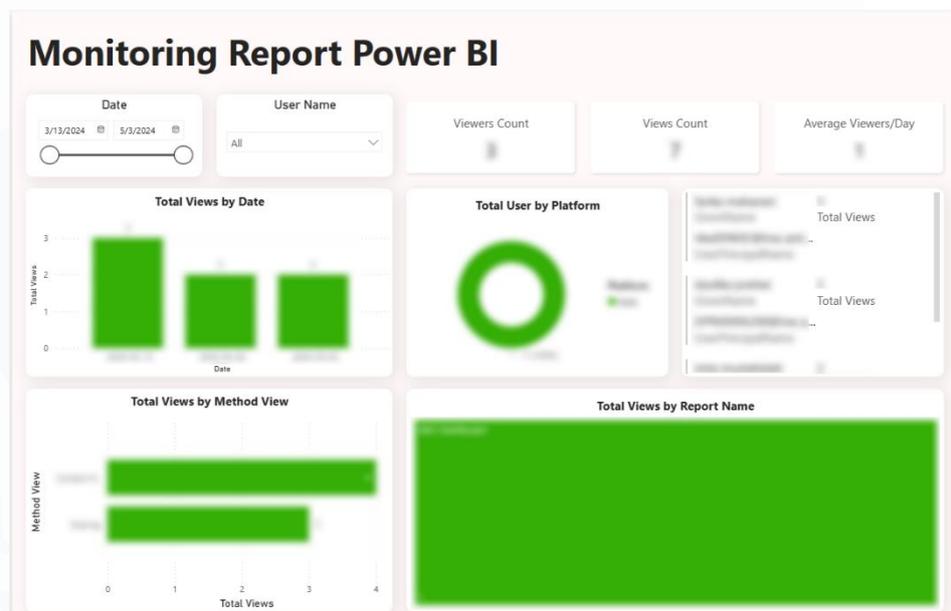
No.	Role	Standard/Custom	Notes	PERMISSION																
				Can view and edit the dashboard																
1	Admin																			
2	Member																			
3	Contributor																			
4	Viewer																			

Gambar 3. 10 Role Permission Matrix (RPM)

Gambar 3. 10 merupakan hasil *Role Permission Matrix* yang telah selesai dibuat. Pada tabel tersebut terdapat beberapa *permission* yang dapat dilakukan oleh masing-masing role. Berdasarkan *permission* tersebut, tiap role memiliki akses yang berbeda-beda yang dibagi menjadi 4 kriteria. Kriteria tersebut merupakan CRUD: *Create (C)*, *Read (R)*, *Update (U)*, dan *Delete (D)*.

3.2.2.3 Pembaruan dashboard monitor

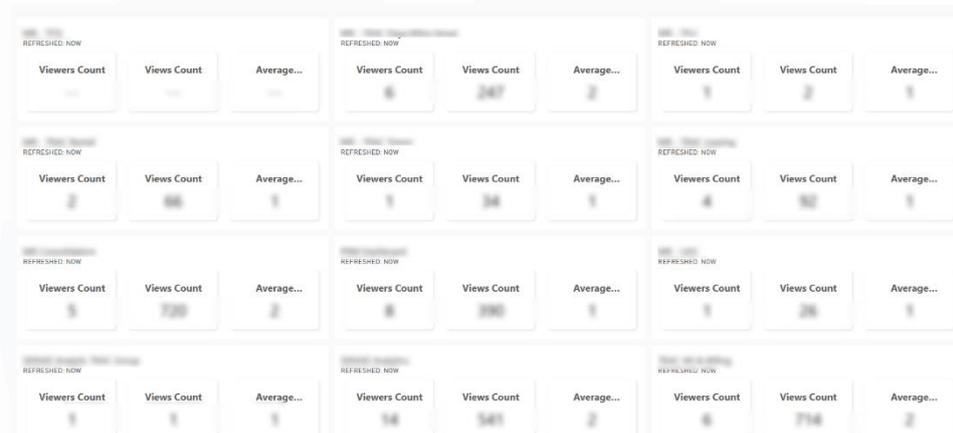
Dashboard monitor yang akan dibuat akan digunakan untuk memonitor penggunaan dashboard atau *workspace* yang sudah dibuat. Sebelum membuat dashboardnya, akan dibuat report penggunaan dari masing masing *workspace*.



Gambar 3. 11 Usage Report Workspace

Gambar 3. 11 merupakan tampilan dari salah satu report penggunaan *workspace*. Pada report tersebut terdapat informasi mengenai viewers dari

workspace yang digunakan, yaitu berupa jumlah, tanggal, maupun user yang mengaksesnya. Data tersebut berasal dari metrics model pada workspace power BI.



Gambar 3. 12 *Dashboard Monitoring Usage Workspace*

Gambar 3. 12 merupakan *dashboard monitoring* dari penggunaan semua workspace yang ada. Data *viewers* diambil dari hasil *usage report workspace* yang sudah dibuat sebelumnya.

3.2.3 Pengajuan proposal proyek magang

Untuk memenuhi tugas magang, pekerja magang diberi tugas untuk mempersiapkan proyek yang akan dikerjakan selama magang. Proyek dibuat berdasarkan pengamatan pekerja magang selama melakukan magang di Perusahaan.

3.2.3.1 Pembuatan PPT proposal proyek magang

Untuk mengajukan proyek magang, pekerja magang membuat power point sederhana tentang proyek yang akan dikerjakan. Berikut merupakan power point yang diajukan.





Dashboard Sentiment Analysis Playstore TracToGo dan IBID

Gambar 3. 13 PPT Pengajuan Proyek Magang (1)

Gambar 3. 13 merupakan judul dari proyek yang diajukan untuk dilakukan selama magang berlangsung. Proyek akan membuat dashboard analisis sentiment dari ulasan *mobile approtation* dari *business unit* SERA yaitu TracToGo dan IBID. TracToGo merupakan aplikasi *mobile* yang dapat digunakan user untuk melakukan penyewaan kendaraan, seperti mobil, bis, dan motor. Sedangkan IBID merupakan aplikasi *mobile* yang dapat digunakan user untuk mengikuti lelang yang diadakan oleh salah satu *business unit* SERA yaitu IBID.

U M N
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

Background

Analisis sentimen dapat digunakan untuk meningkatkan kualitas aplikasi yang juga akan meningkatkan pengalaman pengguna, mengidentifikasi area untuk perbaikan agar dapat bersaing dengan kompetitor.

Goals

Menyediakan informasi rekomendasi yang dapat diimplementasikan untuk tim developer TracToGo dan IBID berdasarkan analisis sentimen.

Gambar 3. 14 PPT Pengajuan Proyek Magang (2)

Pada Gambar 3. 14 dijelaskan alasan dibuatnya dashboard analisis sentimen untuk aplikasi TracToGo dan IBID tersebut, yaitu untuk meningkatkan kualitas aplikasi yang akan juga meningkatkan pengalaman pengguna. Tujuan yang akan dicapai adalah adanya rekomendasi yang dapat membantu tim developer aplikasi mobile maupun tim layanan untuk meningkatkan kualitasnya.

Technology & Implementation



Bahasa pemrograman python untuk pengumpulan data, pengolahan data, dan pemodelan. Lalu Power BI sebagai alat untuk visualisasi dan pembuatan dashboard.

Gambar 3. 15 PPT Pengajuan Proyek Magang (3)

Pada Gambar 3. 15 terdapat aplikasi yang akan digunakan selama proyek berlangsung. Bahasa pemrograman python akan digunakan untuk mengolah data ulasan dengan menggunakan aplikasi Visual Studio Code

maupun Azure Databricks. Lalu aplikasi Power BI Desktop akan digunakan untuk pembuatan dashboard analisis sentiment yang akan menjadi hasil akhir dari proyek ini. Aplikasi Power BI digunakan karena SERA menggunakan aplikasi Power BI sebagai alat untuk proyek-proyek sebelumnya. Sehingga *user* yang akan menggunakan dashboard analisis sentiment akan lebih familiar dengan cara penggunaan aplikasi Power BI dan juga pengelolaan lisensi dan akses *user* akan lebih mudah dijalani.

3.2.3.2 Presentasi proyek yang akan dikerjakan selama magang

Setelah dibuatnya perencanaan dan Power Point untuk pengajuan proyek magang, Presentasi dilakukan kepada kepala departemen Data Platform. Presentasi dilakukan untuk meminta persetujuan dari kepala departemen Data Platform bahwa proyek yang diajukan dapat dilakukan selama melakukan magang di PT Serasi Autoraya.

3.2.4 Melakukan *data scraping* dari review pada App Store dan Play Store untuk aplikasi TracToGo dan Ibib Auction

Hal pertama yang dilakukan dalam pembuatan dashboard analisis sentiment adalah pengambilan data yang akan digunakan. Data yang akan digunakan berasal dari ulasan aplikasi *mobile* dari Play Store dan App Store untuk aplikasi TracToGo dan Ibib.

3.2.4.1 Melakukan *data scraping* pada Play Store

Penarikan data ulasan dari Play Store memerlukan library yang sudah tersedia.

```

from google_play_scraper import app
import pandas as pd
import numpy as np
from google_play_scraper import Sort, reviews_all
from google_play_scraper import Sort, reviews

```

Gambar 3. 16 Kode *Import Library* untuk *Scraping* di Play Store

Gambar 3. 16 merupakan kode untuk mengimpor *library* yang diperlukan untuk melakukan *scraping* data dari Google Play Store. Sebelum dilakukan *import library* tersebut, harus dipastikan bahwa sudah dilakukan *install library* yang diperlukan sebelumnya.

```

# SCRAPE SEMUA REVIEWS SEKALIGUS

result_trac = reviews_all(
    'com.trac.tractogo',
    sleep_milliseconds=0, # defaults to 0
    lang='id', # defaults to 'en'
    country='id', # defaults to 'us'
    sort=Sort.NEWEST, # defaults to Sort.MOST_RELEVANT , you can use Sort.
                    NEWEST to get newst reviews
)

```

Gambar 3. 17 Kode untuk *Scraping Ulasan TracToGo* di Play Store

Gambar 3. 17 merupakan kode untuk melakukan *scraping ulasan* aplikasi TracToGo dari Google Play Store. Pada proses dilakukan akses ke halaman ulasan aplikasi dan mengekstrak informasi yang diperlukan, seperti peringkat, ulasan, dan waktu pembuatan ulasan tersebut. Hasil data ulasan TracToGo dari *Play Store* yang berhasil diambil adalah sebesar 398 baris yang berisi ulasan dari tanggal 28 Maret 2021 sampai 22 Januari 2024.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

```
p_trac = pd.DataFrame(np.array(result_trac), columns=['review'])
p_trac = p_trac.join(pd.DataFrame(p_trac.pop('review').tolist()))
p_trac.head()
```

Gambar 3. 18 Kode untuk Mengubah Data Menjadi Data Frame

Setelah melakukan scraping data, langkah selanjutnya adalah mengubahnya menjadi bentuk data yang lebih mudah diolah, yaitu Data Frame. Dalam kode yang ditunjukkan di Gambar 3. 18, data yang telah diambil dari ulasan TracToGo diubah menjadi Data Frame.

reviewId	userName	userImage	content	score	thumbsUpCount	reviewCreatedVersion	at	replyContent	repliedAt	appVersion
0	lh.googleusercontent.com		Dafarnya susah	2	0	None	2024-01-22 18:11:28			None
1	lh.googleusercontent.com			5	0					
2	lh.googleusercontent.com			5	0					
3	lh.googleusercontent.com			5	0					
4	lh.googleusercontent.com			2	0			Hi		

Gambar 3. 19 Hasil Scraping Data TracToGo di Play Store

Gambar 3. 19 merupakan hasil dari *data scraping* ulasan TracToGo dari Google Play Store yang telah diubah menjadi Data Frame. Tabel tersebut berisi informasi review id, username, user image, content (ulasan), score (rating), at (tanggal pembuatan ulasan), dan sebagainya.

```
p_trac.to_excel("p_trac.xlsx", index = False)
```

Gambar 3. 20 Kode untuk Ekspor Data Frame ke File Excel

Setelah data diubah menjadi Data Frame, langkah selanjutnya adalah mengekspornya ke format file Excel. Kode dalam Gambar 3. 20 menggambarkan proses untuk mengekspor Data Frame ke file Excel.

```

result_ibid = reviews_all(
    'com.sera.ibid',
    sleep_milliseconds=0, # defaults to 0
    lang='id', # defaults to 'en'
    country='id', # defaults to 'us'
    sort=Sort.NEWEST, # defaults to Sort.MOST_RELEVANT , you can use Sort.
    NEWEST to get newst reviews
)

```

Gambar 3. 21 Kode untuk Scraping Ulasan Ibid di Play Store

Sama seperti langkah sebelumnya, Gambar 3. 21 menunjukkan kode untuk melakukan scraping ulasan aplikasi Ibid dari Google Play Store. Hasil data ulasan IBID dari *Play Store* yang berhasil diambil adalah sebesar 1791 baris yang berisi ulasan dari tanggal 16 Mei 2019 sampai 31 Januari 2024.

```

p_ibid = pd.DataFrame(np.array(result_ibid),columns=['review'])
p_ibid = p_ibid.join(pd.DataFrame(p_ibid.pop('review').tolist()))
p_ibid.head()

```

Gambar 3. 22 Kode untuk Mengubah Data Ulasan Ke Data Frame

Setelah data ditarik, maka data ulasan Ibid tersebut akan diubah juga ke bentuk *data frame* menggunakan kode seperti di Gambar 3. 22.

reviewId	userName	userImage	content	score	thumbsUpCount	reviewCreatedVersion	at	replyContent	repliedAt	appVersion
0		!tgoogleusercontent	admin nya pelawannya	5	0	1.15.5	2024-01-31 17:09:44			
1		!tgoogleusercontent	i	2	0					
2		!tgoogleusercontent		5	0					
3		!tgoogleusercontent		5	0					
4		!tgoogleusercontent	i n	3	0					

Gambar 3. 23 Hasil Scraping Data Ibid di Play Store

Gambar 3. 23 merupakan hasil dari ulasan aplikasi Ibid di Play Store yang sudah berbentuk *data frame*.

```
p_ibid.to_excel("p_ibid.xlsx", index = False)
```

Gambar 3. 24 Kode untuk Ekspor Data Frame ke File Excel

Setelah data diubah menjadi Data Frame, data ini juga akan diekspor ke format file Excel menggunakan kode pada Gambar 3. 24.

3.2.4.2 Melakukan data scraping pada App Store

Untuk melakukan penarikan data ulasan pada App Store diperlukan library yang berbeda.

```
from app_store_scraper import AppStore
import pandas as pd
import numpy as np
```

Gambar 3. 25 Kode Import Library untuk Scraping di App Store

Gambar 3. 25 merupakan kode untuk mengimpor library yang diperlukan untuk melakukan scraping data dari App Store. Seperti pada penarikan data ulasan pada Play Store sebelumnya, langkah ini penting untuk memungkinkan akses ke halaman web dan pengambilan informasi.

```
a_reviews = AppStore('id', 'tractogo-rental-mobil-bus', '1459840738')
a_reviews.review()
```

Gambar 3. 26 Kode untuk Scraping Ulasan TracToGo di App Store

Setelah library yang dibutuhkan diimpor, maka akan dilakukan penarikan data ulasan pada App Store untuk aplikasi TracToGo. Gambar 3. 26 merupakan kode untuk melakukan hal tersebut. Hasil data ulasan TracToGo dari *App Store* yang berhasil terambil adalah sebesar 391 baris yang berisi ulasan dari tanggal 23 September 2019 sampai 1 Februari 2024.

```

a_trac = pd.DataFrame(np.array(a_reviews.reviews), columns=['review'])
a_trac = a_trac.join(pd.DataFrame(a_trac.pop('review').tolist()))

a_trac.head()

```

Gambar 3. 27 Kode untuk Mengubah Data Menjadi Data Frame

Setelah data diambil dari ulasan TracToGo di App Store, langkah selanjutnya adalah mengubahnya menjadi bentuk Data Frame dengan menggunakan kode seperti pada Gambar 3. 27.

	date	review	rating	isEdited	title	userName	developerResponse
0	2023-	Saya sangat puas	5	False	Aplikasinya		
1	2024-		5	False			
2	2022-		2	False			
3	2023-		5	False			
4	2023-		5	False			

Gambar 3. 28 Hasil Scraping Data TracToGo di App Store

Gambar 3. 28 merupakan hasil dari *data scraping* ulasan TracToGo dari App Store yang telah diubah menjadi Data Frame. Tabel tersebut berisi date (tanggal pembuatan ulasan), *review*, rating, username, dan sebagainya.

```

a_trac.to_excel("a_trac.xlsx", index = False)

```

Gambar 3. 29 Kode untuk Ekspor Data Frame ke File Excel

Setelah data diubah menjadi Data Frame, langkah selanjutnya adalah mengekspornya ke format file Excel dengan menggunakan kode seperti di Gambar 3. 29.

```

a_reviews = AppStore('id', 'ibid-balai-lelang-astra', '1501501381')
a_reviews.review()

```

Gambar 3. 30 Kode untuk Scraping Ulasan Ibid di App Store

Selanjutnya adalah melakukan scraping ulasan aplikasi Ibid dari App Store dengan menggunakan kode yang ditunjukkan di Gambar 3. 30. Hasil data ulasan IBID dari *App Store* yang berhasil terambil adalah sebesar 30

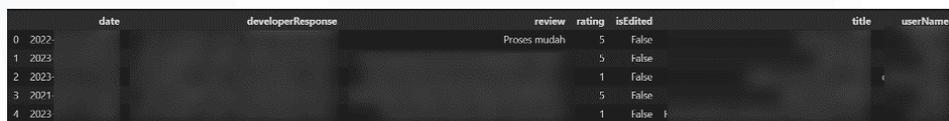
baris yang berisi ulasan dari tanggal 11 Oktober 2020 sampai 22 Oktober 2023.

```
a_ibid = pd.DataFrame(np.array(a_reviews.reviews), columns=['review'])
a_ibid = a_ibid.join(pd.DataFrame(a_ibid.pop('review').tolist()))

a_ibid.head(5)
```

Gambar 3. 31 Kode untuk Mengubah Data Menjadi Data Frame

Pada data ulasan ini juga diubah bentuknya menjadi *data frame* dengan menggunakan kode pada Gambar 3. 31.



	date	developerResponse	review	rating	isEdited	title	userName
0	2022-		Proses mudah	5	False		
1	2023-			5	False		
2	2023-			1	False		
3	2021-			5	False		
4	2023-			1	False		

Gambar 3. 32 Hasil Scraping Data Ibid di App Store

Gambar 3. 32 merupakan hasil dari *data scraping* ulasan Ibid pada App Store yang sudah diubah menjadi bentuk *data frame*.

```
a_ibid.to_excel("a_ibid.xlsx", index = False)
```

Gambar 3. 33 Kode untuk Ekspor Data Frame ke File Excel

Data ini juga perlu diekspor ke bentuk file Excel untuk dilakukan pelabelan manual dengan menggunakan kode seperti pada Gambar 3. 33.

3.2.5 Melakukan pembersihan pada hasil *data scraping* dan labelling untuk sentiment

Setelah data mentah didapatkan, hal yang akan dilakukan selanjutnya adalah pelabelan sentiment secara manual dan pembersihan data agar hasil klasifikasi sentiment yang akan dilakukan memiliki akurasi yang baik.

3.2.5.1 Melakukan pelabelan sentiment manual untuk seluruh data ulasan

Pelabelan data dilakukan untuk training model machine learning yang akan digunakan. Pelabelan ini dilakukan secara manual dengan

menggunakan file excel dari hasil sraping data ulasan pada App Store dan Play Store.

Sentiment	Count (manual_p_trac)	Count (manual_a_trac)	Count (manual_p_ibid)	Count (manual_a_ibid)
negative	42	37.0	254	7.0
neutral	1	0.0	71	0.0
positive	355	354.0	1466	23.0

Gambar 3. 34 Hasil Pelabelan Ulasan Secara Manual

Gambar 3. 34 merupakan hasil perhitungan dari data ulasan yang sudah diberi label sentimen. Hasil pelabelan memiliki jumlah yang berbeda-beda tiap data ulasan dari sumber dan aplikasi yang berbeda.

3.2.5.2 Membuat koding untuk membersihkan data ulasan

Pembersihan data ulasan dilakukan dengan membuat kode yang dapat membersihkan data dari berbagai hal seperti tanda baca, kata-kata slang, dan hal lainnya yang tidak diperlukan dalam training model *machine learning* nantinya.

```
## Kamus Kata Slang
# Baca kamus slang dari file Excel
kamus_slang = pd.read_excel('combined_slang_words.xlsx')

# Ubah kamus slang ke dalam bentuk dictionary
slang_dict = dict(zip(kamus_slang['Kunci'], kamus_slang['Nilai']))
```

Gambar 3. 35 Kode untuk Impor Kamus *Slang Word*

Gambar 3. 35 merupakan kode untuk mengimpor kamus yang akan digunakan untuk mengubah *slang words* ke bentuk yang lebih baku. Kamus tersebut tersedia di github yang dapat digunakan dengan bebas (https://github.com/louisowen6/NLP_bahasa_resources).

```

def cleaningText(text):
    text = re.sub('[0-9]+', '', text) # Menghapus angka
    text = text.replace('\n', ' ') # Mengganti baris baru dengan spasi
    # text = text.translate(str.maketrans('', '', string.punctuation)) # Menghapus semua tanda baca
    text = text.translate(str.maketrans(string.punctuation, ' ' * len(string.punctuation))) # Mengganti semua tanda baca dengan spasi
    text = text.strip() # Menghapus spasi di awal dan akhir teks
    text = re.sub(r'\s+', ' ', text) # Mengganti beberapa spasi berturut-turut dengan satu spasi
    return text

def replaceSlang(text, slang_dict):
    words = text.split()
    for i in range(len(words)):
        if words[i] in slang_dict:
            words[i] = slang_dict[words[i]]
    return ' '.join(words)

def casefoldingText(text): # Converting all the characters in a text into lower case
    text = text.lower()
    return text

def tokenizingText(text): # Tokenizing or splitting a string, text into a list of tokens
    text = word_tokenize(text)
    return text

```

Gambar 3. 36 Kode untuk Membersihkan Data Ulasan (1)

Gambar 3. 36 merupakan kode untuk membersihkan data ulasan yang berisi beberapa fungsi. Pertama terdapat fungsing `cleaningText()` yang dapat menghapus angka, mengganti baris baru dengan spasi, menghapus tanda baca, dan memperbaiki spasi yang tidak pas. Lalu terdapat fungsi `replaceSlang()` yang berfungsi untuk mengubah kata-kata *slang* dengan kata yang lebih baku. Lalu terdapat fungsi `casefoldingText()` yang berfungsi untuk mengubah semua huru menjadi huruf kecil. Lalu terdapat fungsi `tokenizingText()` yanh berfungsi untuk mengubah data ulasan menjadi berbentuk token-token.



```

def filteringText(text): # Remove stop words
    list_stopwords = stopwords.words('indonesian')
    list_stopwords.extend(['ibid'])
    listStopwords = set(list_stopwords)
    filtered = []
    for txt in text:
        if txt not in listStopwords:
            filtered.append(txt)
    text = filtered
    return text

def stemmingText(text): # Reducing a word to its word stem that addixes to suffixes
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()
    text = [stemmer.stem(word) for word in text]
    return text

def toSentence(list_words): # Convert list of words into sentence
    sentence = ' '.join(word for word in list_words)
    return sentence

```

Gambar 3. 37 Kode untuk Membersihkan Data Ulasan (2)

Gambar 3. 37 merupakan kode lanjutan dari fungsi yang akan digunakan untuk pembersihan data sebelumnya. Pada Gambar 3.34 terdapat kode fungsi `filteringText()` yang akan digunakan untuk menghapus *stopwords* menggunakan kamus sastra *stopwords Indonesian*. Lalu terdapat fungsi `stemmingText` yang berfungsi untuk membuat kata berimbuhan menjadi kata dasar. Lalu terakhir terdapat fungsi `toSentence()` untuk membuat data yang tadinya berbentuk token menjadi satu kembali menjadi bentuk *sentence*.

```

df['text_clean'] = df['Review'].apply(cleaningText)
df['casefolding'] = df['text_clean'].apply(casefoldingText)
df['replace_slang'] = df['casefolding'].apply(lambda x: replaceSlang(x, slang_dict))

df['text_preprocess'] = df['replace_slang'].apply(tokenizingText)
df['remove_stopword'] = df['text_preprocess'].apply(filteringText) # Remove stop words
df['stemming'] = df['remove_stopword'].apply(stemmingText)

df['Review_Clean'] = df['stemming'].apply(toSentence)

```

Gambar 3. 38 Kode untuk Membersihkan Data Ulasan (3)

Gambar 3. 38 merupakan kode untuk menggunakan fungsi yang telah dibuat sebelumnya. Hasil dari pembersihan data tersebut, terdapat data yang dihilangkan karena adanya *remove duplicate*. Data ulasan dari aplikasi Tractogo yang tadinya berjumlah 789 baris menurun menjadi 776

baris. Lalu untuk data ulasan aplikasi IBID yang tadinya berjumlah 1.821 baris menjadi 1.119 baris.

3.2.6 Membuat pemodelan prediksi sentiment menggunakan *machine learning*

Setelah data sudah bersih dan siap digunakan, maka akan dilajut dengan pemodelan klasifikasi sentiment.

3.2.6.1 Membuat word vectorizer fasttext untuk mengoptimalkan model machine learning

Pada tahap ini akan dibuat word vectorizer menggunakan FastText untuk membuat data mejadi betuk vector dan siap untuk dilakukan training model *machine learning*. Word vectorizer atau word embedding merupakan teknik natural language processing (NLP) yang memungkinkan kata-kata diubah menjadi vektor kontinu dengan panjang yang ditentukan, sehingga tidak terbatas oleh kosakata yang lebih banyak [5]. Hal tersebut dapat membuat mesin untuk memahami hubungan atarkata dalam data teks untuk melakukan klasifikasi teks, analisis sentiment, dan pemrosesan NLP lainnya.

Salah satu word vectorizer yang dapat digunakan adalah fasttext. FastText adalah metode pembelajaran mesin yang digunakan untuk membangun representasi vektor kata [6]. Metode ini dikembangkan oleh Facebook AI Research dan merupakan pengembangan dari model Word2Vec. FastText memperhitungkan informasi *sub-word* dalam representasi vektor kata, sehingga dapat menangkap makna kata yang tidak terdapat dalam kamus.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

```

import pandas as pd
import numpy as numpy

from tqdm.auto import tqdm

from nltk.tokenize import word_tokenize
from gensim.models import FastText

```

Gambar 3. 39 Kode Import Library untuk Training Model Fasttext

Gambar 3. 39 merupakan kode untuk mengimpor library yang akan digunakan untuk melakukan training model fasttext.

```

# Data Play Store

p_ibid = pd.read_excel("manual_p_ibid.xlsx")
p_ibid.head(3)

# Data App Store

a_ibid = pd.read_excel("manual_a_ibid.xlsx")
a_ibid.head()

```

Gambar 3. 40 Kode untuk Load Data Ulasan

Sebelum melakukan training, data ulasan akan diimpor menggunakan pandas seperti pada Gambar 3. 40. Data yang akan digunakan pertama merupakan data ulasan dari aplikasi Ibid baik dari Play Store maupun App Store.

```

df = pd.concat([p_ibid, a_ibid], ignore_index=True)

```

Gambar 3. 41 Kode untuk Menggabungkan Data Ulasan Menjadi Satu Data Frame

Dari data ulasan ibid pada Play Store dan App store tersebut, akan digabung dengan menggunakan pandas concat dengan kode pada

Gambar 3. 41. Setelah itu data akan dibersihkan seperti pada kode yang dijelaskan di sub-subbab 3.2.5.2. Setelah itu data yang akan digunakan untuk training adalah data yang sudah dibersihkan.

```
sentences = [word_tokenize(Review.lower()) for Review in tqdm(df.Review)]
```

Gambar 3. 42 Kode untuk Mengubah Data Menjadi Berbentuk Token.

Gambar 3. 42 merupakan kode untuk mengubah data ulasan yang sudah dibersihkan ke dalam bentuk token agar dapat dilakukan training menggunakan fasttext.

```
model = FastText(sentences, vector_size=128, window=5, min_count=3, workers=4, epochs=1000, sg=0, hs=0)
```

Gambar 3. 43 Kode untuk Training Model Fasttext

Gambar 3. 43 merupakan kode untuk melakukan training model fasttext. Data yang digunakan menggunakan variabel sentences yang telah diubah menjadi token pada tahap sebelumnya. Model dibuat untuk menjadi 128 vector untuk tiap katanya.

```
model.save("ibid_review.ft")
```

Gambar 3. 44 Kode untuk Ekpor Model Fasttext

Setelah model selesai di-*training*, maka model akan disimpan dengan menggunakan kode seperti di Gambar 3. 44. Hal ini dilakukan agar model dapat dipakai di kemudian hari tanpa harus melakukan *training* lagi.

```
model = FastText.load("ibid_review.ft")
```

Gambar 3. 45 Kode untuk Load Model Fasttext

Untuk menggunakan model yang sudah disimpan tadi, kode pada Gambar 3. 45 dapat digunakan untuk melakukan *load model* menggunakan pickle.

```
def norm_sent_vector(sentence, w2v_model):
    if len(sentence) == 0:
        return np.zeros(w2v_model.vector_size)
    vecs = [w2v_model[word] for word in sentence]
    vecs = [vec / np.linalg.norm(vec) for vec in vecs if np.linalg.norm(vec) >
            0]
    sent_vec = np.mean(vecs, axis=0)
    return sent_vec
```

Gambar 3. 46 Kode untuk Penggunaan Model Fasttext (1)

Pada Gambar 3. 46 terdapat kode untuk penggunaan model fasttext sebagai *word vectorizer*. Pada gambar tersebut terdapat fungsi yang dapat digunakan pada kode selanjutnya.

```
vecs2 = [norm_sent_vector(sentence, ft) for sentence in df.Review_Clean]
vecs2 = np.array(vecs2)
```

Gambar 3. 47 Kode untuk Penggunaan Model Fasttext (2)

Pada Gambar 3. 47 terdapat kode untuk penggunaan model fasttext. Kode tersebut akan menggunakan fungsi yang sudah dibuat sebelumnya.

3.2.6.2 Membuat model machine learning untuk klasifikasi ulasan berdasarkan sentiment

Setelah data ulasan berbentuk vektor seperti yang diperlukan untuk melakukan *training* model *machine learning*, selanjutnya akan dilakukan *training* model *machine learning* tersebut untuk klasifikasi sentiment pada ulasan aplikasi.

```
X = vecs2
y = df['Sentiment']
```

Gambar 3. 48 Kode untuk Menentukan Variabel Independen dan Dependen

Pertama akan variabel independent dan dependen akan ditentukan dengan kode pada Gambar 3. 48. Variabel independen merupakan data ulasan yang sudah diubah menjadi vector menggunakan fasttext yaitu variabel `vecs2`. Lalu variabel dependennya merupakan label sentiment yang sudah dibuat secara manual pada tahap sebelumnya.

```
# Handling imbalanced using SMOTE
smote = SMOTE()
x_sm,y_sm = smote.fit_resample(X,y)
```

Gambar 3. 49 Kode untuk Mengatasi Data yang Tidak Seimbang

Data yang tidak seimbang adalah kondisi di mana kelas-kelas dalam data memiliki frekuensi yang sangat berbeda. Hal ini dapat menyebabkan masalah dalam *training* model karena model akan cenderung menjadi bias terhadap kelas yang memiliki data lebih banyak. Gambar 3. 49 merupakan kode untuk mengatasi data tidak seimbang dengan menggunakan teknik oversampling dengan menggunakan SMOTE untuk menyeimbangkan kelas-kelas dalam data.

```
X_train , X_test , y_train , y_test = train_test_split(x_sm ,y_sm ,
    test_size=0.3,random_state=3)

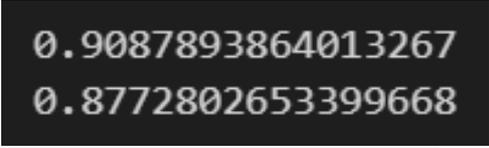
#random forest
random_forest_classifier = RandomForestClassifier()
random_forest_classifier.fit(X_train,y_train)
random_forest_classifier_prediction = random_forest_classifier.predict(X_test)
print(accuracy_score(random_forest_classifier_prediction,y_test))

# SVM
svm = SVC()
svm.fit(X_train,y_train)
svm_prediction = svm.predict(X_test)
print(accuracy_score(svm_prediction,y_test))
```

Gambar 3. 50 Kode untuk Training Model Machine Learning

Setelah data ulasan diseimbangkan, maka selanjutnya akan dilakukan *training* model *machine learning*. Pada Gambar 3. 50 dilakukan *split data train* dan *data test*. Lalu model yang digunakan adalah *random forest* dan *support vector machine (SVM)*.

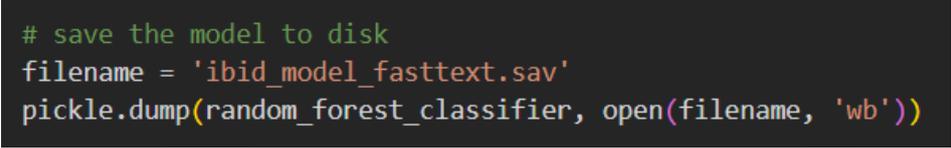
UNIVERSITAS
MULTIMEDIA
NUSANTARA



```
0.9087893864013267
0.8772802653399668
```

Gambar 3. 51 Hasil Akurasi Model Machine Learning

Gambar 3. 51 merupakan hasil akurasi dari *training random forest* dan *SVM*. Dikarenakan oleh hasil akurasi dari model *random forest* lebih tinggi, maka model *random forest* akan digunakan sebagai model klasifikasi sentiment nantinya.



```
# save the model to disk
filename = 'ibid_model_fasttext.sav'
pickle.dump(random_forest_classifier, open(filename, 'wb'))
```

Gambar 3. 52 Kode untuk Menyimpan Model Machine Learning

Setelah model dengan akurasi terbaik dipilih, maka model tersebut perlu disimpan agar dapat dengan mudah digunakan di kemudian hari dengan menggunakan kode seperti di Gambar 3. 52. Kemudian kode pada Gambar 3. 40 sampai dengan Gambar 3. 52 akan dilakukan juga untuk data ulasan dari aplikasi TracToGo. Jadi tiap aplikasi masing-masing memiliki model fasttext untuk *word vectorizer* dan model *machine learning* untuk klasifikasi sentimennya masing-masing.

3.2.6.3 Melakukan ekspor data yang sudah siap untuk dipakai

Setelah model *random forest* untuk klasifikasi disimpan, maka untuk menggunakannya model tersebut perlu di-*load* dengan menggunakan `pickle.load()`.

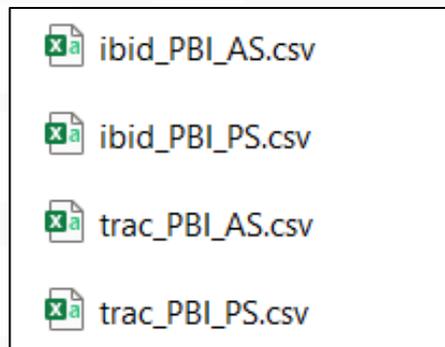
Gambar 3. 53 Kode untuk Penggunaan Model Machine Learning

Gambar 3. 53 merupakan kode untuk melakukan *load model machine learning*. Kemudian model yang sudah di-*load* akan digunakan untuk klasifikasi dengan menggunakan `model.predict()`. Kemudian hasil klasifikasi tersebut akan dimasukkan ke kolom baru bernama “Sentiment”.

```
ibid_PBI_AS = df[['Store', 'Date', 'Rating', 'Title', 'Review', 'Review_Clean',  
                'Sentiment']]  
ibid_PBI_AS.to_csv("ibid_PBI_AS.csv")
```

Gambar 3. 54 Kode untuk Ekspor Data yang Sudah Siap Digunakan

Setelah data ulasan tersebut diberi sentimentnya dengan menggunakan model yang sudah di-*training*, maka data akan diekspor ke bentuk csv untuk digunakan di Power BI Desktop seperti di Gambar 3. 54. Sebelum diekspor, akan dipilih kolom mana saja yang akan digunakan untuk pembuatan dashboard analisis sentiment.



Gambar 3. 55 File Hasil Klasifikasi Menggunakan *Machine Learning*

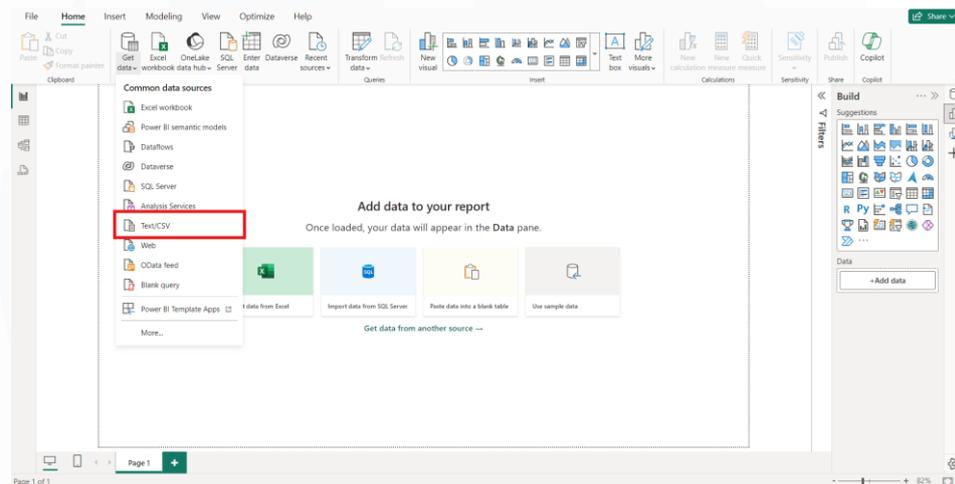
Setelah data ulasan Ibid di ekspor, maka kode pada Gambar 3. 53 dan juga Gambar 3. 54 harus dilakukan lagi untuk semua data yang akan digunakan pada Power BI Desktop. Data yang akan digunakan terdapat 4 buah termasuk aplikasi Ibid pada Play Store dan App Store maupun aplikasi TracToGo pada kedua *store* tersebut juga sehingga menghasilkan data seperti pada Gambar 3. 55.

3.2.7 Pembuatan dashboard analisis sentiment menggunakan aplikasi Power BI

Pembuatan dashboard atau visualisasi dari analisis sentiment pada proyek ini akan menggunakan Power BI Desktop sebagai alatnya. Berikut merupakan tahapan yang dilakukan dalam pembuatan dashboard analisis sentimen.

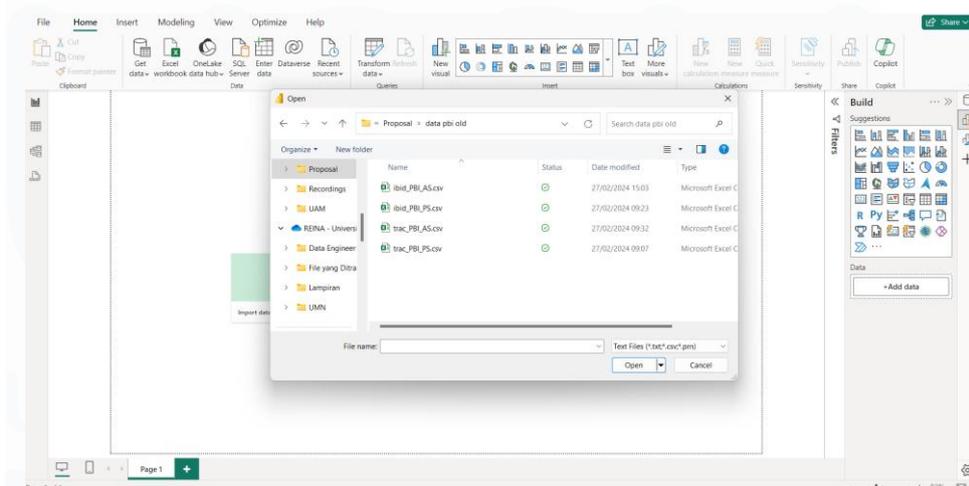
3.2.7.1 Penyambungan power BI dengan data hasil klasifikasi

Pada tahap ini Power BI akan dihubungkan dengan data hasil klasifikasi. Data tersebut disimpan dalam format CSV seperti yang ada di tahapan ekspor hasil klasifikasi.



Gambar 3. 56 Mengambil Data dari File CSV (1)

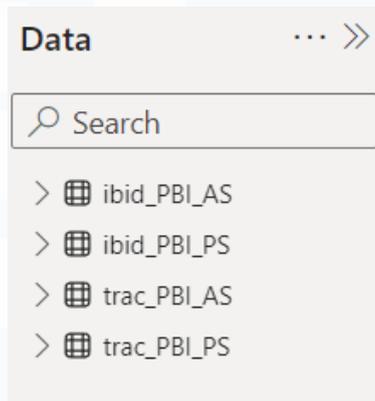
Gambar 3. 56 merupakan salah satu tahapan dari pengambilan data dari file csv ke Power BI Desktop. Setelah klik pada tombol “*Get Data*”, maka akan muncul dropdown seperti di gambar. Setelah itu, untuk mengambil data dari file csv klik Text/CSV.



Gambar 3. 57 Mengambil Data dari File CSV (2)

Setelah klik pada Text/CSV maka akan muncul tab seperti di Gambar 3. 57. Pada tab tersebut, klik salah satu file csv yang akan digunakan. Jika

sudah klik “*Open*” lalau akan muncul *tab* untuk *load*. Setelah muncul *tab* tersebut klik tombol “*Load*” dan data akan terimpor. Setelah itu akan dilakukan hal tersebut sampai semua data yang dibutuhkan sudah di impor.

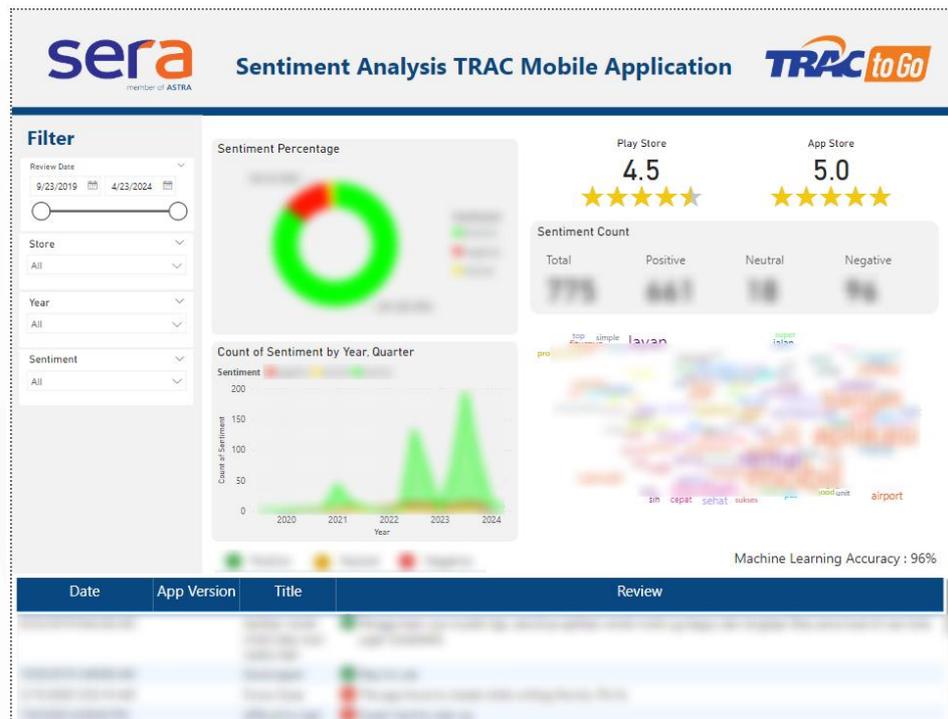


Gambar 3. 58 Hasil Data pada Power BI Desktop Setelah CSV di *Import*

Setelah semua data tersedia dalam Power BI Desktop, maka pada menu Data akan terlihat seperti pada Gambar 3. 58 yang memiliki empat tabel untuk tiap dataset ulasan pada tiap aplikasi mobile dan masing-masing Play Store dan App Store.

3.2.7.2 Pembuatan dashboard analisis sentiment

Setelah data yang diperlukan tersedia dalam Power BI Desktop, maka pembuatan dashboard akan dilanjutkan dengan membuat visualisasi yang tepat untuk analisis sentiment.



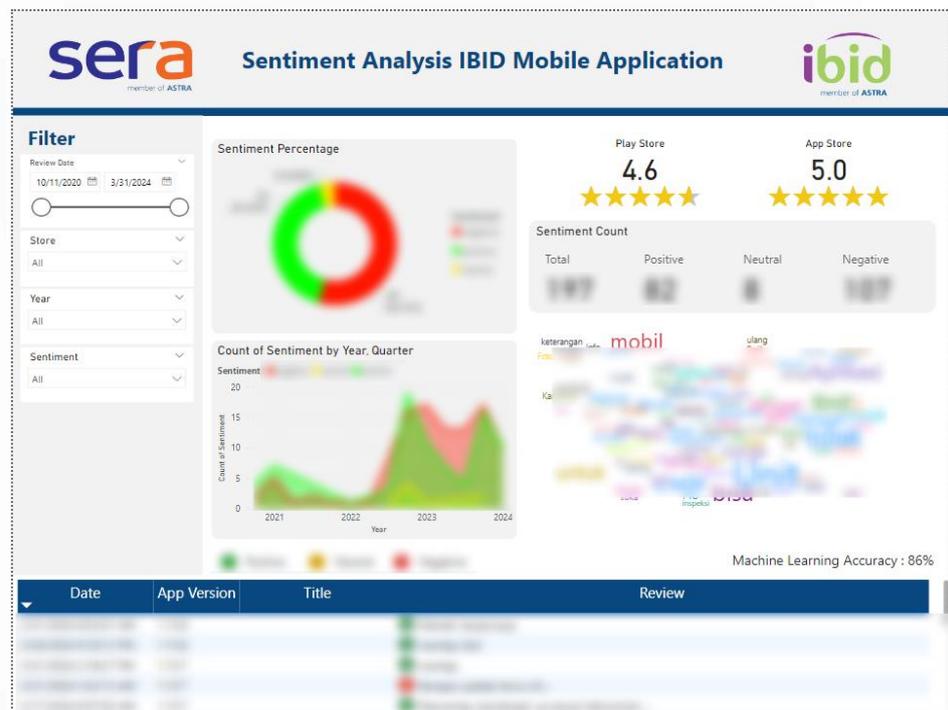
Gambar 3. 59 Hasil Dashboard Analisis Sentimen untuk TracToGo

Gambar 3. 59 merupakan hasil dashboard analisis sentiment dari aplikasi mobile TracToGo. Pada Dashboard tersebut terdapat header yang berisi judul untuk memudahkan user mengetahui apa yang ada pada dashboard visualisasi tersebut, juga terdapat logo dari PT Serasi Autoraya, dan juga logo dari aplikasi TracToGo. Lalu pada bagian visualisasi terdapat *donut chart* yang merupakan visualisasi dari persentase masing-masing kelas sentiment. Lalu terdapat juga *area chart* yang merupakan visualisasi dari tiap kelas sentiment pada tiap tahunnya.

Kemudian terdapat juga rating dari aplikasi tersebut pada masing-masing *store*. Selanjutnya terdapat visualisasi *card* yang berisi jumlah ulasan keseluruhan dan berdasarkan masing-masing kelas sentiment. Lalu juga terdapat *wordcloud* untuk visualisasi kata-kata yang terdapat pada ulasan. Pada *wordcloud* tersebut, semakin banyak kata tersebut muncul, maka semakin besar visualisasi dari katanya.

Lalu untuk melihat hasil ulasan lebih *detail*, terdapat tabel yang berisi tanggal, versi aplikasi, title ulasan, dan ulasan pada bagian bawah

visualisasi. Di samping kolom ulasan, masing-masing data mempunyai symbol bulat berwarna yang merepresentasikan sentimen dari masing-masing ulasan, yaitu merah untuk negatif, kuning untuk netral, dan hijau untuk positif. Terakhir, terdapat bagian *filter* yang dapat digunakan untuk mem-*filter* semua visualisasi pada dashboard tersebut. *Filter* tersebut dapat digunakan untuk mem-*filter* tanggal, store, tahun, dan sentiment.



Gambar 3. 60 Hasil Dashboard Analisis Sentimen untuk Ibid

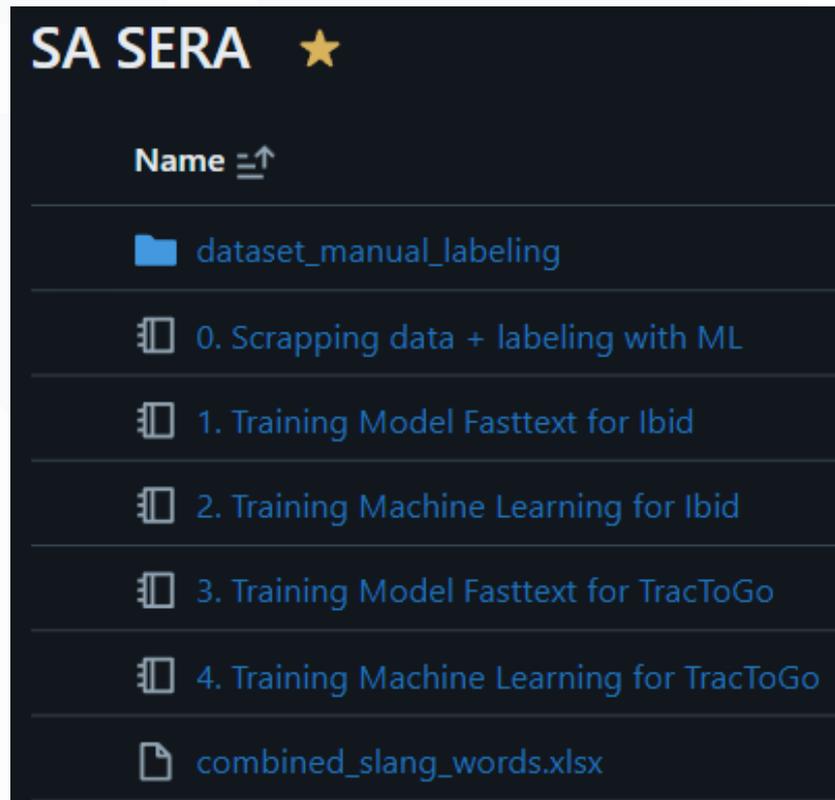
Gambar 3. 60 merupakan hasil dashboard analisis sentiment untuk aplikasi Ibid. Isi visualisasi pada dashboard hanya berbeda sedikit saja dengan dashboard aplikasi TracToGo. Perbedaannya terletak pada header, pada dashboard Ibid ini judul disesuaikan dengan isi dashboard, dan juga logo yang digunakan merupakan logo dari aplikasi Ibid.

3.2.8 Migrasi Dashboard dan code python ke cloud

Pada tahap ini, kode python dan dashboard yang sudah dibuat akan dipindahkan ke cloud. Untuk kode python akan di *upload* dia Databricks dan dashboard Power BI akan di publish ke workspace SERA.

3.2.8.1 Melakukan upload kode python pada databricks

Pada tahap ini dilakukan pengunggahan notebook berisi kode Python yang terkait dengan scraping data, pemrosesan data, dan pembuatan model ke Databricks. Databricks adalah platform cloud yang sering digunakan untuk analisis data dan pemrosesan big data.



Gambar 3. 61 File yang Di Upload di Databricks

Gambar 3. 61 merupakan file yang diunggah ke dalam Databricks dengan folder yang diberi nama “SA SERA”. Pada folder tersebut terdapat folder lainnya Bernama dataset_manual_labeling yang berisi data mentah yang sudah diberi label secara manual. Lalu terdapat file dengan nama “0. Scraping data + labeling with ML” yang berisi kode untuk melakukan scraping data dan pelabelan dengan model yang sudah di training sebelumnya. Jadi jika SERA ingin melakukan klasifikasi sentiment dengan data terbaru, dapat melakukan klik “Run All” pada notebook tersebut dan data pada Blob Storage akan terbaharui selama 30 hari terakhir. Lalu untuk membuat data pada dashboard Power BI menampilkan data terbaru, dapat

dilakukan refresh semantic model pada workspace dimana dashboard akan di-*publish* nantinya.

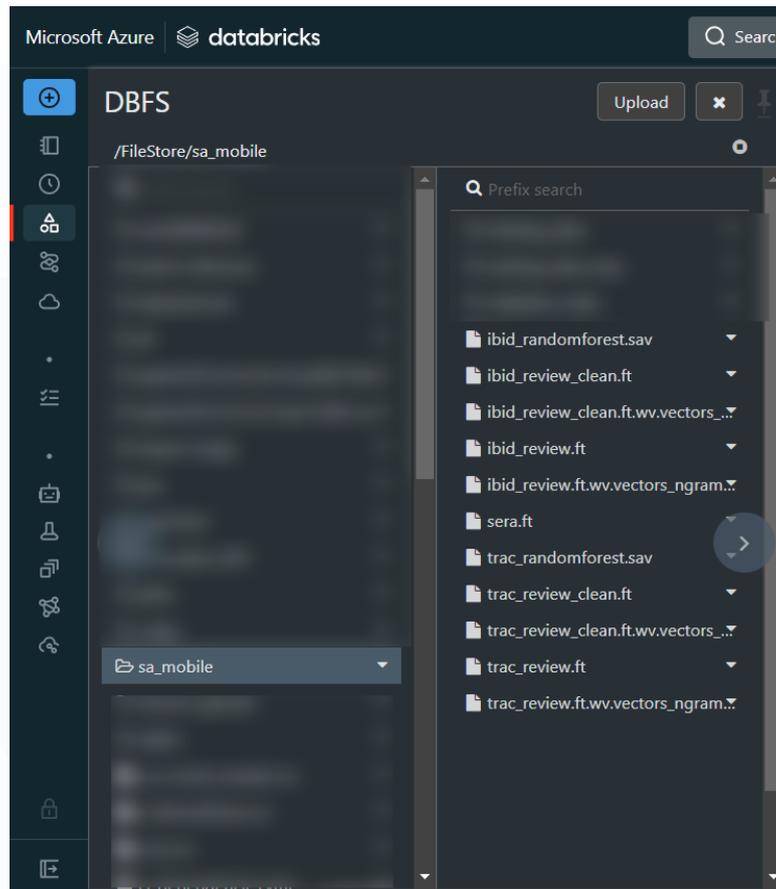
Lalu pada file “1. Training Model Fasttext for Ibid” berisi kode untuk melakukan training model fasttext untuk aplikasi Ibid. Pada file “2. Training Machine Learning for Ibid” terdapat kode untuk training model *machine learning* untuk klasifikasi ulasan pada aplikasi Ibid.

Kemudian pada file “3. Training Model Fasttext for TracToGo” berisi kode untuk melakukan training model fasttext untuk aplikasi TracToGo. Pada file “4. Training Machine Learning for TracToGo” terdapat kode untuk training model *machine learning* untuk klasifikasi ulasan pada aplikasi TracToGo.

Kemudian file “combined_slang_words.xlsx” berisi kamus yang akan digunakan pada fungsi pengubah *slang words* menjadi kata yang lebih baku pada proses pembersihan data.

3.2.8.2 Melakukan testing dan perbaikan kode pada databricks

Setelah kode diunggah, langkah selanjutnya adalah melakukan *testing* dan perbaikan kode jika diperlukan. Setelah kode dijalankan kembali di Databricks, terdapat beberapa hal yang harus diubah. Seperti path pada file, dan harus melakukan instalasi *library* yang tidak ada pada *environment* yang digunakan.



Gambar 3. 62 Tempat Model Fasttext dan Model *Machine Learning* di Simpan

Gambar 3. 62 menunjukkan tempat di mana model Fasttext dan model machine learning disimpan setelah pengujian dan perbaikan dilakukan. Model Fasttext dan model machine learning tersebut disimpan di Databricks File Systems (DBFS).

```

sc._jsc.hadoopConfiguration().set("...",
)
DeltaTable_Path = "wasbs://...blob.core.windows.net/..."

#####

DbSilver = "..._silver"

data_path = ""+DeltaTable_Path +"/trac_playstore"
(df_p_trac_final
.write.mode("overwrite")
.option("overwriteSchema", "true")
.option("path", data_path)
.saveAsTable(""+DbSilver+".trac_playstore")
)

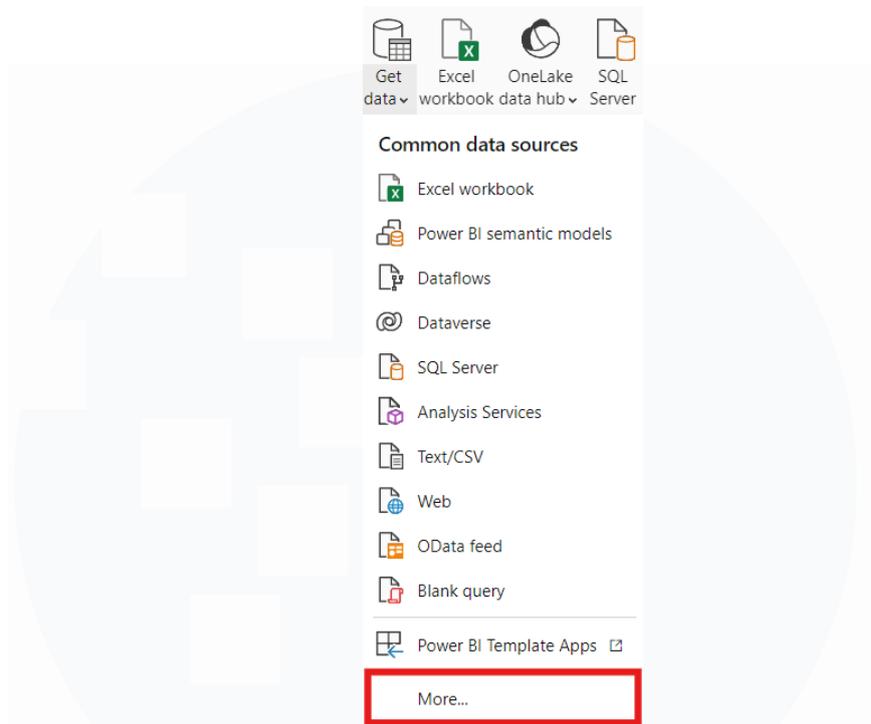
```

Gambar 3. 63 Kode untuk Menyimpan Hasil Data Scraping dan Klasifikasi Sentimen

Hasil data scraping dan sudah diberikan sentimen menggunakan model klasifikasi akan disimpan di *Blob Storage* dengan bentuk *Delta Table*. Gambar 3. 63 merupakan kode untuk menyimpan data tersebut ke *Blob Storage*, data tersebut kemudian akan digunakan pada visualisasi Power BI nantinya.

3.2.8.3 Melakukan penyambungan data hasil klasifikasi pada cloud ke power BI

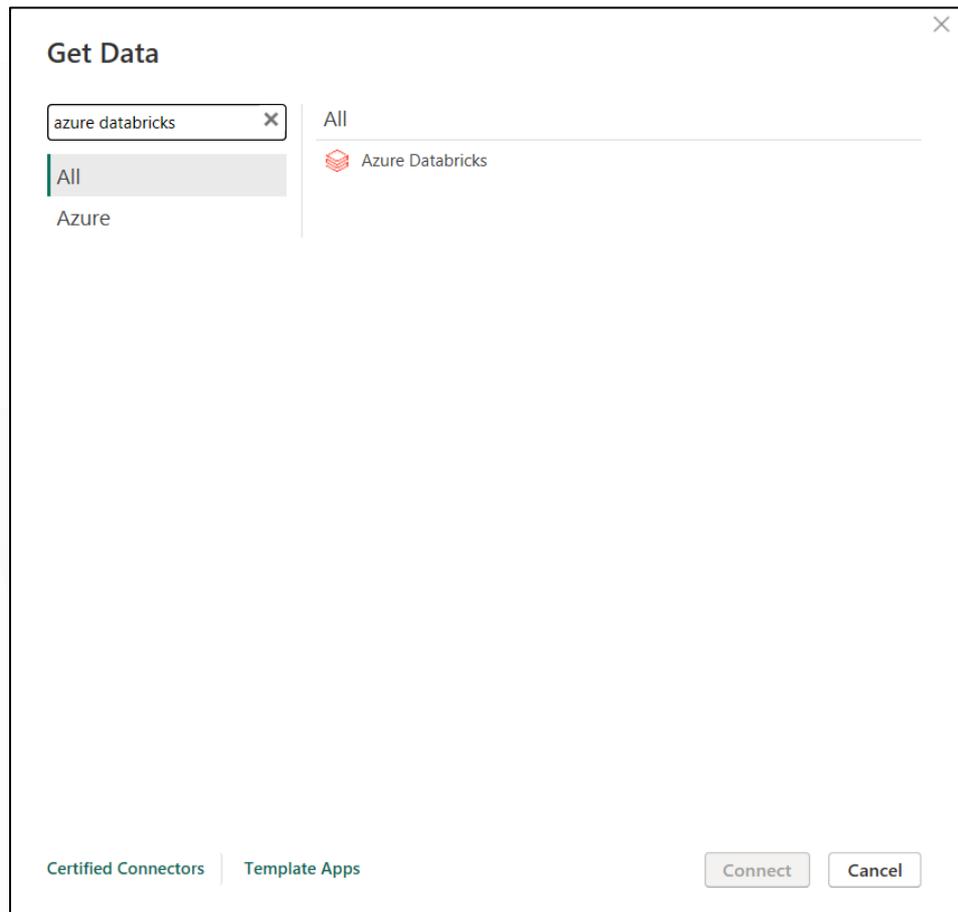
Pada tahap akan dilakukan penyambungan data hasil klasifikasi yang telah diproses dan disimpan di *Blob Storage* ke aplikasi Power BI untuk pembuatan dashboard. Gambar 3.60 dan 3.61 menunjukkan tahapan penyambungan Power BI dengan Azure Databricks, yang merupakan integrasi antara Power BI dan layanan cloud dari Microsoft.



Gambar 3. 64 Tahapan Koneksi Power BI dengan Azure Databricks (1)

Untuk membuat koneksi Power BI dengan *Blob Storage* hal pertama yang perlu dilakukan adalah klik tombol *Get Data* pada menu *Home*, lalu klik *More* seperti pada Gambar 3. 64.





Gambar 3. 65 Tahapan Koneksi Power BI dengan Azure Databricks (2)

Setelah *More* di klik maka akan muncul tampilan seperti pada Gambar 3. 65. Pada tampilan tersebut perlu untuk mencari pada kolom search dengan kata kunci “Azure Databricks”. Setelah pilihan tersebut di klik, maka dapat di klik tombol *connect* untuk menyambungkan.

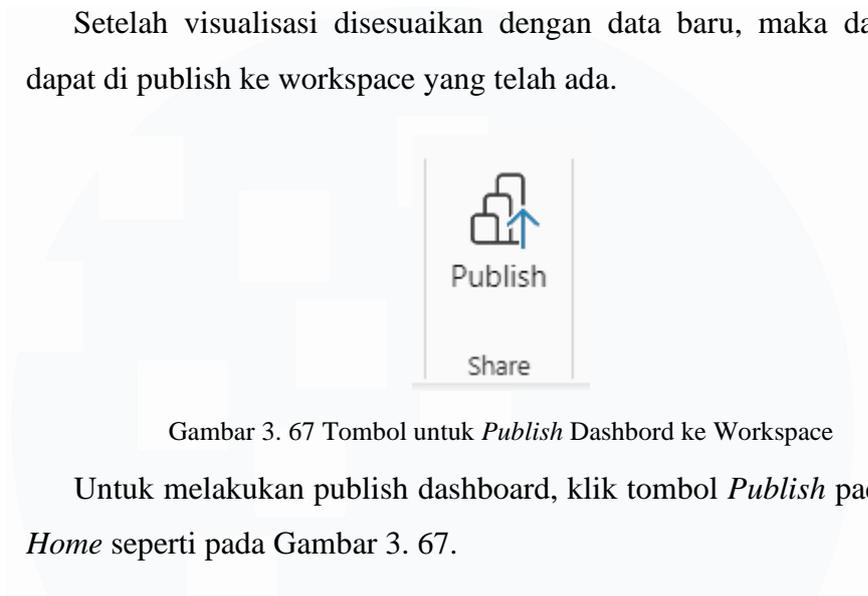
Gambar 3. 66 Tahapan Koneksi Power BI dengan Azure Databricks (3)

Setelah itu akan muncul tampilan seperti pada Gambar 3. 66. Pada halaman tersebut diminta *server hostname* dan *HTTP Path* untuk menyambungkan ke Blob Storage. Lalu *data connectivity mode* yang digunakan adalah *import*. Setelah klik *OK* user diperlukan untuk melakukan *login* menggunakan akun Perusahaan. Setelah itu data akan muncul pada menu *Data* pada Power BI.

Jika Power BI sudah tersambung dengan Blob Storage, maka hanya diperlukan penyesuaian pada visualisasinya, yaitu dengan mengubah sumber data yang tadinya berasal dari CSV ke data dari *Blob Storage*.

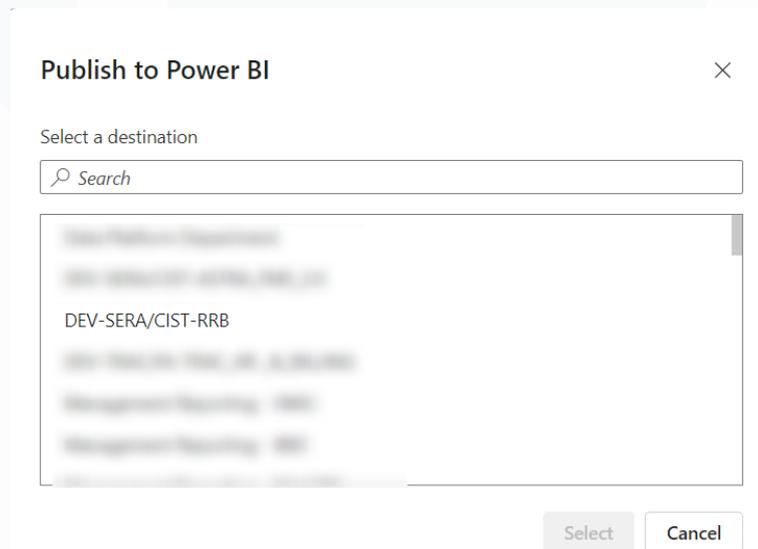
3.2.8.4 Melakukan publish power BI Dashboard

Setelah visualisasi disesuaikan dengan data baru, maka dashboard dapat di publish ke workspace yang telah ada.



Gambar 3. 67 Tombol untuk *Publish* Dashboard ke Workspace

Untuk melakukan publish dashboard, klik tombol *Publish* pada menu *Home* seperti pada Gambar 3. 67.



Gambar 3. 68 Workspace yang Digunakan untuk *Publish* Dashboard

Setelah klik tombol *publish* akan muncul tampilan seperti pada Gambar 3. 68. Pada tampilan tersebut akan muncul *workspace* yang terdapat pada perusahaan. Untuk dashboard analisis sentiment ini akan disimpan pada workspace yang dibuat untuk pekerja magang. Setelah *workspace* dipilih dan klik *select*, maka *dashboard* akan muncul pada *workspace* dan dapat dibuka secara daring.

3.2.9 Pembuatan Data Dictionary

Data dictionary yang dibuat merupakan dokumen yang berisi penjelasan dari data yang digunakan dalam aplikasi yang terdapat di PT Serasi Autoraya. Dokumen ini nantinya dapat digunakan untuk mengetahui data apa saja yang digunakan untuk keperluan bisnis. Pembuatan data dictionary ini termasuk ke pekerjaan *business support* yang akan membantu bisnis berjalan dengan lancar kedepannya.

3.2.9.1 Request akses ke server

Langkah pertama yang dilakukan dalam proses pembuatan *Data Dictionary* adalah meminta akses server database yang digunakan untuk tiap aplikasi yang akan dibuat *data dictionary*-nya. Permintaan ini dilakukan melalui salah satu database administrator yang mempunyai akses ke server database yang akan digunakan. Persetujuan pemberian akses server database tidak hanya melibatkan satu orang, tetapi dibutuhkan persetujuan beberapa tingkatan karyawan.

3.2.9.2 Pembuatan dokumen data dictionary

Setelah akses ke server database diberikan, langkah selanjutnya adalah membuat dokumen *Data Dictionary*. Dokumen ini berisi daftar lengkap dari semua data yang digunakan dalam aplikasi, termasuk deskripsi data, format, jenis data, jumlah karakter, sumber data, dan informasi penting lainnya yang berkaitan dengan setiap elemen data.

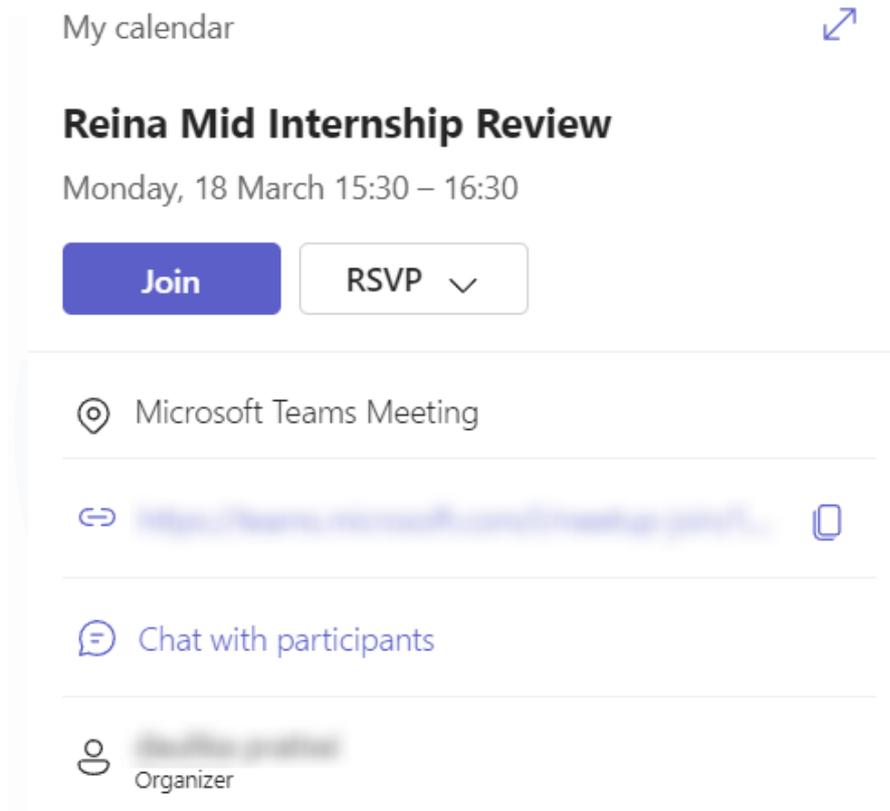
Pembuatan *data dictionary* ini dilakukan berdasarkan *Entity Relationship Diagram (ERD)* pada tiap aplikasi yang akan dibuat *data dictionary*-nya seperti yang terdapat pada Gambar 3. 69 dan juga Gambar 3. 70. ERD tersebut didapatkan dari tim proyek pembuatan tiap aplikasinya, sehingga diperlukan untuk menghubungi tim yang membuat tiap aplikasinya.

3.2.10 Review Magang

Review magang dilakukan selama dua kali selama magang berlangsung, yaitu pada saat pertengahan masa magang dan saat hari terakhir masa magang. Review dilakukan bersama dengan *supervisor* dan mentor kegiatan magang. Review magang ini bertujuan untuk mengontrol kegiatan magang agar tetap sesuai rencana dan memberikan pekerja magang ruang untuk berdiskusi tentang magang yang sedang dilakukan.

3.2.10.1 Melakukan Presentasi hasil kerja selama setengah periode pertama.

Pada kegiatan ini pekerja magang mempresentasikan hasil kerjanya selama setengah periode pertama magang. Presentasi ini bertujuan untuk memberikan *update* kepada *supervisor* dan mentor terkait kemajuan proyek, pencapaian yang telah dicapai, hambatan yang dihadapi, serta rencana untuk kegiatan berikutnya. Ini juga merupakan kesempatan bagi peserta magang untuk berbagi pengalaman, pengetahuan yang diperoleh, dan pelajaran yang telah dipelajari.



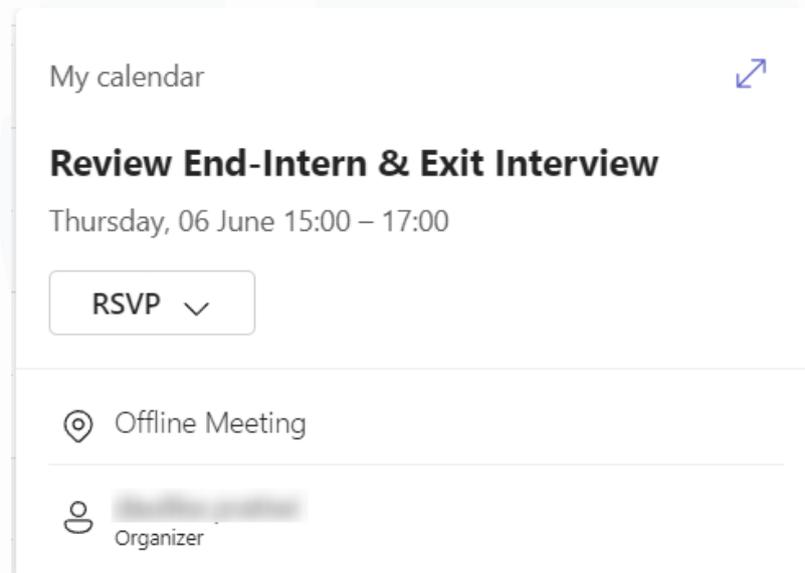
Gambar 3. 71 Undangan untuk Mid Review Internship

Gambar 3. 71 merupakan undangan untuk meeting *mid review internship* yang dilaksanakan pada 18 Maret 2024. Meeting tersebut dilakukan melalui platform Microsoft Teams secara daring. Selain bersama mentor dan *supervisor* magang, meeting juga dilakukan bersama *supervisor* dari tim *Data Engineer* dan *AI Engineer* untuk meminta pendapat mereka terhadap pekerjaan *scraping data* dan pembuatan model *machine learning* untuk klasifikasi sentimen.

3.2.10.2 Review *End-Intern* dan *Exit Interview*

Review End-Intern akan dilakukan beberapa hari sebelum masa magang berakhir. Hal ini dilakukan dengan tujuan mendapatkan umpan balik tentang kinerja selama kegiatan magang berlangsung. Umpan balik dapat berupa diskusi tentang pencapaian dan hal-hal dapat diperbaiki oleh peserta magang untuk kedepannya.

Exit interview akan dilakukan dengan tujuan untuk memastikan bahwa data-data perusahaan yang tersimpan pada perangkat kerja pekerja magang sudah dikembalikan. Pekerja magang akan mengisi formulir tentang data-data perusahaan yang dikembalikan. Hal ini dilakukan agar data perusahaan tidak tersebar ke luar perusahaan.



Gambar 3. 72 Undangan untuk *End-Internship Review* dan *Exit Interview*

Gambar 3. 72 merupakan undangan untuk *meeting review end-internship* yang dilaksanakan pada 6 Juni 2024. *Meeting* tersebut dilakukan melalui platform Microsoft Teams secara daring bersama dengan mentor dan *supervisor* dari tim *Business Intelligence*.

3.3 Kendala yang Ditemukan

Pada saat kegiatan magang berlangsung di PT Serasi Autoraya sebagai *Business Intelligence Engineer Intern*, terdapat kendala yang dialami baik secara teknis maupun non-teknis berikut:

- a. Saat pembuatan User Access Matrix (UAM) diperlukan data posisi hingga divisi pada tiap user. Tetapi untuk mengisi bagian tersebut diperlukan untuk mengisi manual untuk tiap user yang memerlukan waktu yang tidak sebentar.

- b. Saat melakukan training model *machine learning* untuk klasifikasi sentiment ulasan, akurasi yang didapatkan tidak tinggi. Sehingga hasil klasifikasi memiliki performa yang kurang baik. Tetapi Perusahaan memerlukan hasil klasifikasi yang memiliki performa baik agar dapat melakukan analisis dengan tepat sesuai data yang ada.
- c. Pada saat melakukan migrasi kode python, databricks tidak dapat membaca file model *word vectorizer fasttext* yang sudah dilakukan *training* pada *visual studio code*. Kemudian file tersebut sudah dicoba di storage yang berbeda untuk mengetes apakah hal tersebut akan berpengaruh. Tetapi hal tersebut tidak berpengaruh sehingga dibutuhkan solusi lainnya agar dapat menggunakan model fasttext di Databricks.

3.4 Solusi atas Kendala yang Ditemukan

Berdasar pada kendala-kendala yang dialami selama kegiatan magang di PT Serasi Autoraya berlangsung, berikut merupakan Solusi yang dibuat untuk menghadapi kendala-kendala tersebut:

- a. Untuk mengatasi masalah pertama, pekerja magang meminta solusi pada mentor dan supervisor yang menghasilkan Solusi bahwa terdapat file karyawan aktif yang memiliki data yang diperlukan untuk mengisi UAM. Data tersebut dapat digunakan secara lebih cepat dengan menggunakan fungsi vlookup pada Microsoft Excel.
- b. Untuk mengatasi hasil akurasi yang kurang baik, maka digunakan hyperparameter untuk mendapatkan parameter terbaik dalam sebuah model. Lalu karena hasil akurasi masih kurang baik, dibuat perubahan metode word vectorizer atau word embedding yang tadinya menggunakan TF-IDF menjadi menggunakan Fasttext.
- c. Untuk mengatasi masalah file model fasttext tidak bisa dibaca pada Databricks, didapatkan Solusi untuk melakukan training ulang pada Databricks. Setelah model di *training* ulang di Databricks, masalah tetap muncul karena file tidak terbaca. Lalu diperoleh Solusi bahwa model dapat berjalan dengan baik saat file disimpan di *Databricks File System (DBFS)*.

Kemudian setelah mendapat Solusi tersebut, maka semua model yang akan digunakan dalam analisis sentiment akan disimpan pada DBFS tersebut.

