

## BAB III

### PELAKSANAAN KERJA MAGANG

#### 3.1 Kedudukan dan Koordinasi

Kegiatan magang dilaksanakan di perusahaan Staffinc Group dengan penempatan pada divisi *Engineering*, tepatnya pada posisi *Data Operation* di tim Data. Dalam program ini, posisi yang dijalani adalah sebagai *Data Operation Intern*, dengan ruang lingkup tanggung jawab yang difokuskan pada pengelolaan serta optimalisasi alur data perusahaan. Selama periode magang, bimbingan dan arahan diberikan secara langsung oleh mentor internal, Bapak Doni Kurniawan, yang menjabat sebagai *Data Warehouse Officer*, serta berperan sebagai pemberi tugas, pengarah teknis, dan pengatur jadwal kerja harian.

Divisi Data Engineering memegang peran sentral dalam pengelolaan keseluruhan data perusahaan, termasuk data pekerja *outsourcing*, data klien, serta data operasional internal dari berbagai divisi seperti *Finance*, *People*, *Business*, dan *Legal*. Mengingat *volume* dan kompleksitas data yang tinggi, dibutuhkan dokumentasi dan arsitektur pengolahan data yang sistematis agar proses berjalan efektif dan efisien.

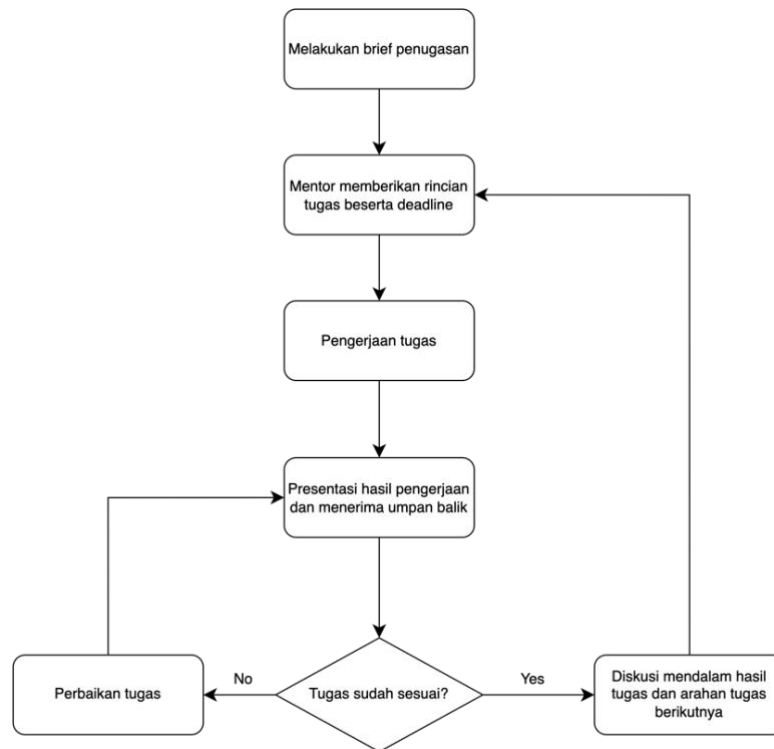
Salah satu proyek utama selama masa magang adalah pemantauan data pipeline dan penyusunan arsitektur alur data (*data flow*) beserta dokumentasi teknis sebagai referensi standar pengelolaan data lintas divisi. Proyek ini bertujuan untuk mencegah redundansi dan kesalahan dalam pengolahan data. Proyek berhasil diselesaikan dalam waktu tiga minggu, dengan hasil berupa dokumentasi alur data yang kini dapat dimanfaatkan oleh tim *Data Engineering* maupun divisi internal lainnya untuk memahami dan menavigasi

data perusahaan dengan lebih mudah. Dalam pengerjaannya, digunakan *Python API* dan *Google Sheets API* untuk mengotomatisasi dan mempermudah proses dokumentasi.

Selain penyusunan dokumentasi *data flow*, tanggung jawab lain selama masa magang mencakup pengembangan *dashboard* interaktif menggunakan *Metabase* berdasarkan permintaan dari divisi-divisi internal. *Dashboard* ini menyajikan visualisasi data dalam bentuk laporan mingguan, bulanan, hingga kuartalan untuk mendukung pengambilan keputusan yang lebih cepat dan berbasis data. Proses pengembangan dilakukan melalui koordinasi dengan masing-masing divisi guna memahami kebutuhan metrik dan indikator yang diperlukan, dengan komunikasi difasilitasi melalui platform *Slack* sebagai kanal utama antar tim. Kehadiran mahasiswa magang dalam proyek ini terbukti efektif, di mana selama satu minggu penuh tidak terdapat permintaan *dashboard* yang tertunda, serta membantu meringankan beban tim data yang sebelumnya hanya terdiri dari satu *Data Warehouse Engineer* dan satu *Data Analyst Officer*.

Seluruh proses kerja selama program magang dilaksanakan secara kolaboratif melalui komunikasi lintas divisi, baik melalui pertemuan langsung di kantor maupun koordinasi daring menggunakan platform komunikasi seperti *Slack*, serta pertemuan virtual melalui *Zoom* atau *Google Meet*. Evaluasi terhadap hasil kerja dilakukan secara berkala oleh mentor dan tim, untuk memastikan bahwa setiap *output* yang dihasilkan telah sesuai standar dan dapat dimanfaatkan langsung dalam kegiatan operasional perusahaan.

Terdapat pula alur kerja yang harus diikuti selama pelaksanaan program magang di Staffinc, yang dijelaskan melalui bagan berikut:



Gambar 3.1 Alur kerja Data Operation Intern di Staffinc Group

Pada gambar 3.1 dijelaskan bagan alur yang menggambarkan alur kerja selama program magang yang memungkinkan terjadinya komunikasi intensif serta pemberian arahan dan masukan dari mentor yang juga merangkap sebagai supervisor. Adapun penjelasan dari alur kerja tersebut adalah sebagai berikut:

1. Arahan dan diskusi awal akan diberikan terkait tugas atau pekerjaan yang akan dijalankan, agar pemahaman terhadap proyek dapat diperoleh secara menyeluruh.
2. Tugas spesifik beserta tenggat waktu akan ditugaskan oleh mentor. Pada tahap ini, proses pengerjaan dilakukan secara mandiri, namun

tetap didukung dengan ruang diskusi apabila ditemukan kendala dalam penyelesaiannya.

3. Setelah tugas selesai dikerjakan, akan dilakukan presentasi atau penjelasan mengenai hasil kerja beserta *output* yang dihasilkan, kemudian akan diberikan evaluasi dan umpan balik oleh mentor.
4. Jika hasil tugas telah memenuhi kriteria yang ditentukan, diskusi lanjutan akan dilakukan untuk meninjau hasil kerja serta kemungkinan pengembangan selanjutnya.
5. Apabila masih terdapat hal-hal yang perlu direvisi, maka tugas tersebut akan ditinjau ulang dan disempurnakan agar kualitasnya sesuai dengan standar yang ditetapkan.

Melalui pelaksanaan program magang ini, pengalaman kerja nyata dalam bidang *data operation* dan *data visualization* di lingkungan perusahaan teknologi berbasis layanan *outsourcing* dapat diperoleh. Keterlibatan dalam proyek-proyek berskala perusahaan turut memberikan pemahaman yang lebih mendalam mengenai pentingnya integrasi sistem data dalam mendukung keberhasilan operasional dan pengambilan keputusan strategis perusahaan.

### **3.2 Tugas dan Uraian Kerja Magang**

Selama pelaksanaan program magang di Staffinc Group, peserta magang ditempatkan pada posisi *Data Operation Intern* dengan lingkup tanggung jawab yang berfokus pada pengelolaan data operasional perusahaan. Penugasan yang diberikan mencakup sejumlah kegiatan teknis seperti penanganan alur data (*data pipeline*) dalam *proses ETL (Extract, Transform, Load)*, penyusunan dokumentasi arsitektur data perusahaan, serta pembuatan

visualisasi data interaktif untuk mendukung kebutuhan analisis di berbagai divisi internal.

Tugas-tugas yang diberikan disesuaikan dengan kebutuhan aktual perusahaan dan berkaitan langsung dengan pengolahan data dalam lingkup bisnis *outsourcing*. Penugasan ini tidak hanya memberikan pengalaman kerja yang relevan, tetapi juga menjadi bagian dari kontribusi peserta magang dalam mendukung pelaksanaan proyek-proyek strategis perusahaan. Berikut rincian tugas-tugas dan uraian kerja magang yang dilakukan selama program magang di Staffinc Group:

Tabel 3.1 Rincian Tugas Praktik Kerja Magang

No	Kegiatan	PIC	Tanggal	
			Mulai	Selesai
<b>On Boarding Session</b>				
A	<i>1-on-1 with Direct Leaders/Manager</i>	<i>Supervisor, Mentor</i>	17 Februari 2025	17 Februari 2025
B	<i>Set up</i> keperluan pekerjaan ( <i>VPN, Internal Databases, Accessibility, etc</i> )	<i>Mentor</i>	17 Februari 2025	17 Februari 2025
<b>Data Documentation</b>				
A	Mengenal dan mempelajari struktur data dan sistem <i>database</i> perusahaan melalui <i>Table Plus MariaDB</i>	<i>Mentor</i>	24 Februari 2025	08 April 2025
B	Membuat Architecture Data Flow Staffinc	<i>Mentor</i>	20 Februari 2025	25 Februari 2025
C	Mendokumentasikan Alur dan Proses Data dari AWS DMS, Apache Airflow, dan Pentaho	<i>Mentor</i>	27 Februari 2025	21 Maret 2025
D	Membuat dokumentasi relasi table beserta diagramnya	<i>Mentor</i>	24 Maret 2025	27 Maret 2025
<b>Dashboard and Data Internal Request</b>				
A	EWA Monthly Report Dashboard	<i>Mentor</i>	08 April 2025	23 Mei 2025
B	Client Contract Management Monitoring Dashboard	<i>Mentor</i>	22 Mei 2025	27 Mei 2025
C	Metabase Internal Usage Dashboard	<i>Mentor</i>	26 Mei 2025	02 Juni 2025
D	Attendance Report All Project Dashboard	<i>Mentor</i>	28 Mei 2025	05 Juni 2025

No	Kegiatan	PIC	Tanggal	
			Mulai	Selesai
E	Specify Client Attendance and Overtime Raw Data	<i>Mentor</i>	04 Juni 2025	12 Juni 2025
<b>Handling Data Pipeline</b>				
A	Brief task handling data pipeline	<i>Mentor</i>	15 April 2025	15 April 2025
B	Membuat data pipeline terbaru di Staffinc	<i>Mentor</i>	16 April 2025	06 Mei 2025
<b>Sailors Program Events</b>				
A	Pengenalan lingkungan perusahaan dan <i>office tour</i> di Staffinc Group	<i>People Team</i>	17 Februari 2025	17 Februari 2025
B	<i>Town Hall</i> Divisi EOP ( <i>Engineering, Operation, Product</i> )	<i>Supervisor</i>	27 Februari & 29 April 2025	27 Februari & 29 April 2025
C	First Month Catch Up Session	<i>People Team, Mentor</i>	13 Maret 2025	14 Maret 2025
D	<i>Town Hall Quartil 1 Staffinc</i>	<i>Stakeholders, Supervisor, People Team</i>	20 Maret 2025	20 Maret 2025
E	<i>Chemistry Event Vol.1 Learning with Leader: Staffinc Plus</i>	<i>People Team</i>	09 April 2025	09 April 2025
D	<i>Chemistry Event Vol.2 Personal Branding</i>	<i>People Team</i>	30 April 2025	30 April 2025
F	<i>Graduation Sailors</i>	<i>People Team</i>	14 Mei 2025	14 Mei 2025

Uraian dari tugas-tugas yang telah dilaksanakan, sebagaimana tercantum pada Tabel 3.1, disajikan di bawah ini:

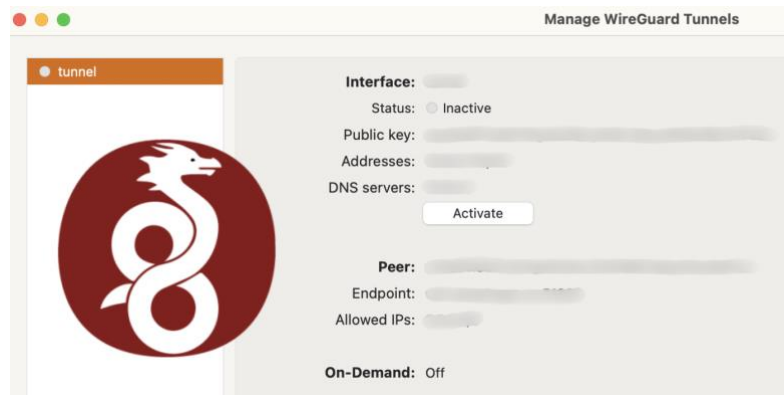
### 3.2.1 On Boarding Session

Pada hari pertama pelaksanaan magang, sesi *On Boarding* dilaksanakan pada tanggal 17 Februari 2025 bertempat di Gedung Ventura lantai 8, yang merupakan kantor pusat Staffinc Group. Dalam sesi ini, seluruh peserta program magang *SAILORS Intern* dikumpulkan di *event hall* untuk diberikan pembekalan terkait sistem kerja perusahaan, pengenalan lingkungan kerja, serta tur kantor. Selain itu, dilakukan juga

sesi pengenalan antara peserta magang yang mencakup divisi penempatan dan peran masing-masing.

Setelah sesi pembekalan oleh tim *People* diselesaikan, peserta diarahkan untuk bergabung dengan *user* dan *leader* dari divisi masing-masing. Bagi peserta yang ditempatkan pada divisi Data, dilakukan pengenalan dengan seluruh anggota tim serta pemaparan mengenai peran tiap anggota. Bimbingan langsung diberikan oleh Bapak Doni Kurniawan selaku *Data Warehouse Officer*. Selain pengenalan, pada sesi tersebut juga diberikan pengarahan mengenai perangkat lunak dan sistem yang akan digunakan selama periode magang.

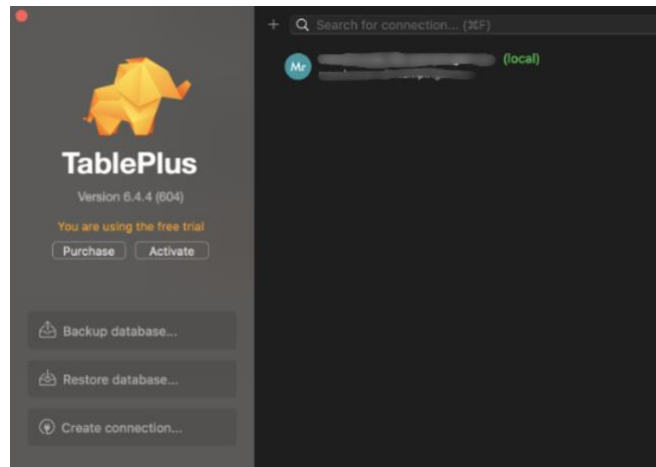
Langkah awal yang dilakukan adalah pengelolaan *WireGuard tunnel* untuk menyambungkan perangkat ke jaringan *VPN* perusahaan. Proses dimulai dengan pemasangan aplikasi *WireGuard* pada perangkat, dilanjutkan dengan konfigurasi tunnel berdasarkan data *VPN* perusahaan. Setelah konfigurasi selesai, koneksi dilakukan ke server *VPN* yang berbasis di Singapura agar akses terhadap database internal perusahaan dapat dilakukan secara aman dan terintegrasi. Ilustrasi keberhasilan koneksi *VPN* ditampilkan pada Gambar 3.2.



Gambar 3.2 Set up VPN Server from WireGuard

Setelah koneksi VPN *WireGuard* berhasil tersambung, langkah selanjutnya adalah pemasangan dan konfigurasi *TablePlus*, yaitu aplikasi pengelola basis data yang dipilih karena sifatnya yang ringan, fleksibel, dan mendukung berbagai jenis database seperti *PostgreSQL* dan *MariaDB*. Tampilan awal *TablePlus* setelah konfigurasi koneksi database dapat dilihat pada Gambar 3.3. Penggunaan *TablePlus* diprioritaskan karena struktur basis data di perusahaan menggunakan kombinasi dua sistem tersebut, dengan *MariaDB* sebagai basis data utama. Oleh karena itu, penguasaan terhadap *TablePlus* menjadi penting untuk menunjang efisiensi kerja tim Data.





Gambar 3.3 Set up MariaDB from Table Plus

Pada sesi pengenalan tersebut juga diberikan arahan untuk mempelajari sejumlah perangkat kerja (*tools*) utama lain yang digunakan dalam tim Data. Beberapa di antaranya adalah *AWS DMS*, *Apache Airflow*, *Google BigQuery*, *Pentaho Data Integration*, serta *Metabase*. Akses terhadap perangkat-perangkat tersebut diberikan secara bertahap sesuai kebutuhan tugas yang diberikan dalam masing-masing fase proyek.

Sebagai tahap awal, dilakukan pembelajaran mengenai penggunaan *AWS DMS* (*Amazon Web Services – Data Migration Service*), yaitu layanan migrasi data dari *AWS* yang memungkinkan pemindahan data antar basis data berbeda secara aman dan efisien tanpa mengganggu operasional aplikasi yang sedang berjalan [5]. Logo layanan *AWS* dapat dilihat pada Gambar 3.4. Pemahaman terhadap layanan ini dibutuhkan karena digunakan dalam proses migrasi dan sinkronisasi data dalam pekerjaan yang akan dilakukan.



Gambar 3.4 Logo aplikasi AWS DMS

Selanjutnya, dilakukan pula pengenalan terhadap *Apache Airflow*, yaitu platform manajemen alur kerja berbasis pemrograman yang digunakan untuk mendefinisikan, menjadwalkan, dan memantau alur kerja data secara sistematis. Dengan dukungan bahasa Python, *Airflow* memungkinkan pembuatan skenario alur data yang kompleks secara efisien dan terintegrasi. Tools ini digunakan oleh perusahaan karena bersifat *open source*, fleksibel, serta mendukung proses orkestrasi data berskala besar [6]. Logo aplikasi Apache Airflow ditampilkan pada Gambar 3.5. Oleh karena itu, penguasaan terhadap penggunaan *Apache Airflow* juga ditargetkan sebagai salah satu kompetensi yang perlu dikuasai selama masa magang berlangsung.



Gambar 3.5 Logo aplikasi Airflow

*Google BigQuery* juga termasuk salah satu perangkat yang diminta untuk dipelajari selama sesi *onboarding* bagi peserta magang. *Google BigQuery* merupakan *cloud-based data warehouse* yang dikembangkan

oleh *Google*, dan dirancang untuk menganalisis data dalam skala besar dengan tingkat kecepatan dan efisiensi yang tinggi. Melalui *Google BigQuery*, proses *query* terhadap data yang tersimpan di *Google Cloud* dapat dilakukan tanpa keharusan untuk mengelola infrastruktur secara langsung. Karena berbasis cloud, platform ini memiliki arsitektur serverless, sehingga pengelolaan infrastruktur fisik tidak lagi diperlukan. Dengan segala keunggulan tersebut, *Google BigQuery* digunakan sebagai salah satu data source utama dalam lingkungan perusahaan [7]. Gambar 3.6 menampilkan logo resmi dari *platform Google BigQuery*.



Gambar 3.6 Logo aplikasi Big Query

Selain *tools* berbasis *data warehouse*, digunakan pula perangkat lunak Pentaho sebagai salah satu alat utama dalam pengelolaan data yang mencakup integrasi data, OLAP, pelaporan, pembuatan *dashboard*, *data mining*, serta proses ETL (*Extract, Transform, Load*). Pentaho dipilih karena kemampuannya dalam mendukung *operational automation*, seperti proses *auto export data*, pengiriman pesan *WhatsApp blast*, serta *email blast* melalui integrasi API [8]. Karena frekuensi penggunaannya yang tinggi dalam kegiatan operasional harian perusahaan, pemahaman mendalam mengenai cara kerja Pentaho dianggap sangat penting dan harus

dikuasai oleh peserta magang. Logo aplikasi Pentaho Data Integration dapat dilihat pada Gambar 3.7.



Gambar 3.7 Logo aplikasi Pentaho Data Integration

Platform lain yang juga banyak digunakan di lingkungan perusahaan adalah Metabase, sebuah sistem *business intelligence* (BI) yang bersifat *user-friendly* dan *open source*. Penggunaan Metabase memungkinkan visualisasi dan analisis data tanpa memerlukan kemampuan pemrograman yang kompleks. Berbagai pertanyaan mengenai data dapat diajukan oleh pengguna, dan hasilnya dapat disajikan dalam berbagai bentuk visual seperti diagram batang, tabel, atau grafik interaktif lainnya [9]. Gambar 3.8 menunjukkan logo aplikasi *Metabase* yang digunakan dalam perusahaan. Oleh karena itu, penguasaan terhadap Metabase juga dianggap penting dalam rangka mendukung pelaksanaan tugas yang berkaitan dengan analisis dan visualisasi data selama masa magang berlangsung.



Gambar 3.8 Logo aplikasi Metabase

## 3.2.2 Data Documentation

### 3.2.2.1 Mempelajari Struktur Data Perusahaan

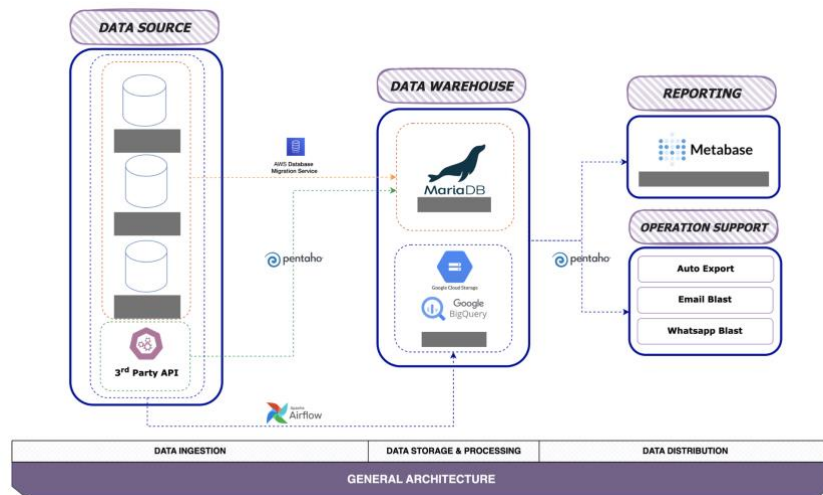
Proyek dokumentasi alur data merupakan sebuah proyek pendokumentasian alur data di perusahaan dalam bentuk dokumentasi menggunakan *Google Sheet*. Pada tahapan awal dilakukan pemahaman terhadap struktur data perusahaan dilakukan dengan menggunakan aplikasi *TablePlus* sebagai alat utama untuk mengakses *database* internal perusahaan. Akses terhadap basis data perusahaan diberikan melalui konfigurasi VPN dengan menggunakan *WireGuard*, agar koneksi terhadap server *database* perusahaan dapat dilakukan secara aman dan terkontrol.

*Database* utama yang digunakan perusahaan adalah *MariaDB*, yang menjadi sumber data utama dalam sebagian proses operasional perusahaan. Maka dari itu, proses familiarisasi terhadap struktur tabel, jenis data, serta relasi antar table dalam *MariaDB* dilakukan terlebih dahulu. Pemahaman ini menjadi dasar dalam mendokumentasikan alur dan struktur data secara menyeluruh.

### 3.2.2.2 Membuat Arsitektur *Data Flow* Staffinc

Setelah memahami basis data perusahaan, tahapan selanjutnya adalah penyusunan arsitektur alur data perusahaan. Arsitektur ini disusun untuk menggambarkan perjalanan data dari sumber awal (*data source*) hingga ke titik akhir masing-masing

datanya memiliki peranan masing-masing. Gambar 3.9 berikut merupakan gambaran arsitektur *data flow* Staffinc.



Gambar 3.9 Data Architecture Staffinc

Arsitektur data flow Staffinc dibagi menjadi tiga bagian utama yaitu *Data Ingestion*, *Data Storage and Processing*, dan *Data Distribution* dapat dilihat di Gambar 3.9 diatas. Pada bagian *Data Ingestion* dilakukan pemindahan data dari beberapa sumber data ke dalam penyimpanan *database* baik yang bersifat lokal (*MariaDB*) maupun *cloud storage* (*Google Cloud Storage* dan *Google Big Query*). Sumber data utama berasal dari aplikasi internal perusahaan yang mencakup informasi pekerja, klien, serta rincian data lainnya yaitu *Staffinc Work*. Data tersebut dimasukkan ke dalam *MariaDB* menggunakan bantuan *AWS DMS (Data Migration Service)*. Selain itu, data dari sumber data external seperti *API WhatsApp* dan *Gmail* juga dimasukkan melalui proses

integrasi yang dijalankan menggunakan bantuan *Pentaho* melalui fitur *export API*. Seluruh proses pemindahan data ini dikendalikan dan dijadwalkan secara otomatis menggunakan *Apache Airflow*, sebelum akhirnya disimpan ke dalam penyimpanan *cloud* berupa *Google Cloud Storage* dan *Google Big Query*.

Pada tahapan *Data Storage and Processing*, data yang telah dikumpulkan disimpan dalam sistem penyimpanan lokal maupun *cloud*, dan digunakan sebagai acuan dalam proses bisnis perusahaan. Proses ini memastikan bahwa data tersimpan dengan aman, terstruktur, dan siap untuk digunakan dalam berbagai kebutuhan analisis dan operasional.

Tahapan terakhir, yaitu *Data Distribution* dilakukan dengan memanfaatkan *Pentaho* sebagai alat bantu utama. Data yang telah disimpan kemudian didistribusikan untuk kebutuhan analisis visual menggunakan *Metabase*, seperti pembuatan *dashboard*, penyajian *raw data* pengguna, dan laporan lainnya. Selain itu, data juga digunakan dalam proses *Operational Support*, seperti *auto export data*, pengiriman *email blast*, serta pengiriman pesan melalui *WhatsApp* yang dilakukan secara otomatis menggunakan *Pentaho API*.

### **3.2.2.3 Mendokumentasikan Alur dan Proses Data**

Dokumentasi selanjutnya difokuskan pada *tools* dan *software* utama yang digunakan dalam pengelolaan dan pemrosesan data perusahaan, yaitu *AWS DMS*, *Apache Airflow*,

dan *Pentaho*. Berikut rincian pengerjaan dokumentasi masing-masing *tools* diatas:

**a) AWS DMS (Data Migration Services)**

Pendokumentasian pada *AWS DMS* dilakukan untuk mencatat proses migrasi data antar basis data, termasuk konfigurasi sumber dan target. Informasi yang didokumentasikan mencakup *Endpoint Name*, *Type*, *Engine*, *HostName*, *UserName*, dan *Database Name*, serta detail migrasi data seperti *Task Title*, *Source*, *Target*, *Tables*, dan *Pairing At*. Hasil dokumentasi *AWS DMS* dapat dilihat dalam Gambar 3.10 dibawah ini.

The image displays two screenshots of AWS DMS documentation. The top screenshot is a table with columns: Endpoint Name, Type, Engine, HostName, UserName, and db\_name. The bottom screenshot is a table with columns: Task Title, Source, Target, Tables, and Pairing At, with rows numbered 1 through 18.

Gambar 3.10 AWS DMS Documentation



Proses dokumentasi dilakukan secara manual dengan mengakses langsung platform *Amazon Web Services (AWS)* untuk memperoleh data yang diperlukan, kemudian data tersebut dicatat dan disusun ke dalam *Google Sheet*. Tujuan dari dokumentasi ini adalah untuk memudahkan pelacakan konfigurasi *DMS* yang aktif dan memastikan konsistensi dalam proses migrasi data di lingkungan perusahaan.

#### **b) Apache Airflow**

Dokumentasi disusun ke dalam beberapa bagian utama yang diklasifikasikan ke dalam beberapa lembar kerja (*sheet*) dalam *Google Sheet*. Masing-masing *sheet* mewakili area spesifik dari konfigurasi dan pelaksanaan *workflow*, seperti *connection*, *dags\_master*, *dags\_task\_gcs*, *dags\_task\_bq*, *unscheduled*, dan *not\_in\_airflow*. Khusus untuk bagian *not\_in\_airflow*, berisi data-data penting yang tersimpan di *MariaDB* namun belum dimasukkan ke dalam sistem penjadwalan *Airflow*, sehingga dijadikan acuan untuk pengembangan *pipeline* di masa mendatang. Tiap entri dalam sheet mencakup informasi seperti *DAG Name*, *task name*, *path file*, asal tabel (*origin table*), serta *field partition* untuk integrasi dengan *BigQuery (bq\_partition field)*. Gambar 3.11 berikut adalah contoh isian dokumentasi *Airflow dags\_master*.

	A	B	C	D
1		dags_name	cron_schedule	status
2	1	4,1_sambutan	0 1 ***	active
3	2	4,1_dibayar	0 5 ***	active
4	3	4,1_hutang	30 0 ***	active
5	4	4,1_internat	0 1 ***	active
6	5	4,1_sampingan	30 1 ***	active
7	6	4,1_sampingan	0 1 ***	active
8	7	4,1_sampingan_minggu	0 2 ***	active
9	8	4,1_sampingan_midi	0 6 ***	active
10	9	4,1_sandi	15 0 ***	active
11	10	4,1_sampingan_agensi_master_data	0 8,14,20 ***	active

Gambar 3.11 Airflow DAGs Master

Proses pengumpulan data untuk dokumentasi ini dilakukan secara otomatis menggunakan metode *Python API Scraping* yang dapat dilihat pada Gambar 3.12. Dengan menggunakan *API* yang telah dikonfigurasi untuk mengakses *Airflow* dan *Google Sheet*, data dari sistem *Airflow* diambil secara langsung. Setelah proses autentikasi berhasil, sistem akan mengambil daftar *DAG (Directed Acyclic Graph)* yang tersedia dengan memanfaatkan permintaan *'requests.get'*. Data yang diperoleh kemudian disesuaikan formatnya agar selaras dengan struktur dokumentasi yang diinginkan.

```

import requests
import gspread
from google.oauth2.service_account import Credentials

# --- Konfigurasi API Airflow (Disesuaikan) ---
AIRFLOW_URL = "https://airflow. ...."
AIRFLOW_USERNAME = "chyntia"
AIRFLOW_PASSWORD = " ...."

# --- Konfigurasi Google Sheets (Disesuaikan)---
SERVICE_ACCOUNT_FILE = " .... json"
GOOGLE_SHEET_ID = " ...."

# Autentikasi Google Sheets
creds = Credentials.from_service_account_file(SERVICE_ACCOUNT_FILE, scopes=["https://www.googleapis.com/auth/spreadsheets"])
client = gspread.authorize(creds)
spreadsheet = client.open_by_key(GOOGLE_SHEET_ID)

# --- Buat atau Ambil Sheet Baru untuk Airflow DAGs ---
SHEET_NAME = "Airflow DAGs"
try:
    sheet = spreadsheet.worksheet(SHEET_NAME)
except gspread.exceptions.WorksheetNotFound:
    sheet = spreadsheet.add_worksheet(title=SHEET_NAME, rows="100", cols="4")

```

Gambar 3.12 Scraping Airflow with Python API

Penyaringan data juga dilakukan menggunakan logika kondisional *if-else* seperti pada Gambar 3.13 di bawah untuk hanya menyertakan jadwal yang masih aktif. Data yang telah diformat selanjutnya diunggah ke dalam *Google Sheet* setelah dilakukan inisialisasi header agar tampilan data lebih rapi dan terbaca. Apabila proses berhasil dijalankan tanpa kendala, akan muncul notifikasi sistem berupa pernyataan bahwa “Dokumentasi *DAG Airflow* berhasil diperbarui di *sheet*”.

```

# --- Ambil Daftar DAG dari Airflow ---
response = requests.get(f"{AIRFLOW_URL}/dags", auth=(AIRFLOW_USERNAME, AIRFLOW_PASSWORD))

if response.status_code != 200:
    print("✗ Gagal mengambil daftar DAG:", response.status_code, response.text)
    exit()

dags = response.json().get("dags", [])

# --- Format Data Sesuai Sheet ---
data_list = []
for index, dag in enumerate(dags, start=1):
    dag_id = dag["dag_id"]

    # Ambil nilai cron schedule (handle format dict atau string)
    schedule = dag.get("schedule_interval", {})
    if isinstance(schedule, dict) and "value" in schedule:
        schedule = schedule["value"]
    elif isinstance(schedule, str):
        schedule = schedule
    else:
        schedule = "Not Available"

    is_active = "active" if dag.get("is_active", False) else "inactive"

    # Tambahkan ke list
    data_list.append([index, dag_id, schedule, is_active])

# --- Update Google Sheet dengan Data Baru ---
header = ["num", "dags_name", "cron_schedule", "status [active/inactive]"]
sheet.clear() # Hapus isi sheet sebelum menulis data baru
sheet.append_row(header) # Tambahkan header
sheet.append_rows(data_list) # Tambahkan data

print(f"✔ Dokumentasi DAG Airflow berhasil diperbarui di sheet '{SHEET_NAME}' di Google Sheets!")

```

Gambar 3.13 Request DAGs Task with Python API

Selain dari sistem *Airflow*, proses dokumentasi juga melibatkan pengambilan data dari *repository* lokal perusahaan yang disimpan dalam *GitHub*. File yang terdapat dalam direktori lokal dibaca secara otomatis menggunakan Python, kemudian dilakukan pemisahan *path file* berdasarkan masing-masing *DAG* dan *task*. Hasil ekstraksi data tersebut kemudian dimasukkan ke dalam *sheet dags\_task\_gcs* dan *dags\_task\_bq* untuk memberikan gambaran yang lebih komprehensif mengenai struktur dan hubungan antar tugas dalam alur kerja yang berjalan di lingkungan perusahaan. Gambar 3.14 berikut merupakan salah satu dokumentasi *dags\_task* pada *Airflow*

no	dag_name	task_name	path	level_1	level_2	level_3	level_4	origin_table	bq_partition_id
1	...	...	...	...	...	...	...	...	...
2	...	...	...	...	...	...	...	...	...
3	...	...	...	...	...	...	...	...	...
4	...	...	...	...	...	...	...	...	...
5	...	...	...	...	...	...	...	...	...
6	...	...	...	...	...	...	...	...	...
7	...	...	...	...	...	...	...	...	...
8	...	...	...	...	...	...	...	...	...
9	...	...	...	...	...	...	...	...	...
10	...	...	...	...	...	...	...	...	...
11	...	...	...	...	...	...	...	...	...
12	...	...	...	...	...	...	...	...	...
13	...	...	...	...	...	...	...	...	...
14	...	...	...	...	...	...	...	...	...
15	...	...	...	...	...	...	...	...	...
16	...	...	...	...	...	...	...	...	...
17	...	...	...	...	...	...	...	...	...
18	...	...	...	...	...	...	...	...	...
19	...	...	...	...	...	...	...	...	...
20	...	...	...	...	...	...	...	...	...

Gambar 3.14 Dokumentasi dags\_task\_gcs Airflow

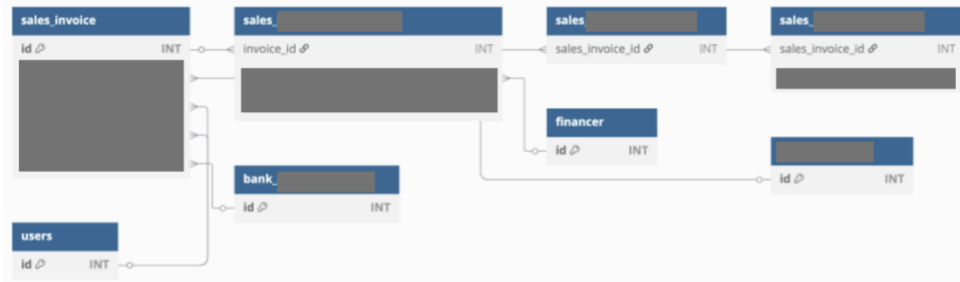
**c) Pentaho Data Integration**

Pendokumentasian Pentaho Data Integration dilakukan dengan mencatat daftar jobs aktif yang berjalan berdasarkan ‘cron\_list’ yang tersedia pada server. Masing-masing job diperiksa untuk memperoleh informasi transformasi data yang dijalankan secara otomatis.

Setiap Job yang aktif kemudian didokumentasikan ke dalam Google Sheet, termasuk rincian seperti job name, schedule, path, path level, dan type. Setiap Job juga di berikan rincian proses nya mencakup Job atau Transformation apa saja yang termasuk di dalam Job tersebut. Pada gambar 3.15 berikut merupakan salah satu contoh hasil dokumentasi job di Pentaho.



dan *payment*. Pada Gambar 3.16 berikut merupakan salah satu diagram. Relasi yang dibuat untuk melihat relasi antara tabel dalam *view table Sales*.



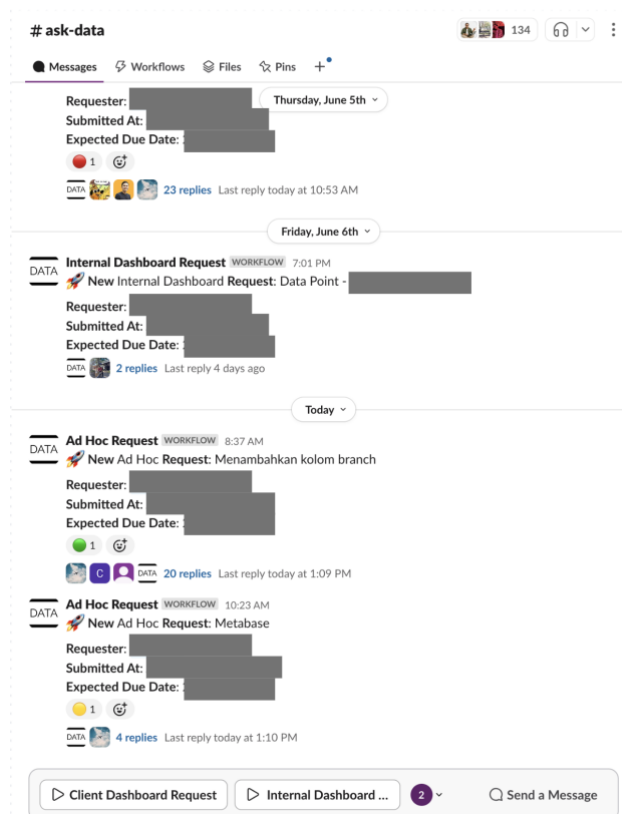
Gambar 3.16 Diagram Relasi Sales View Table

Diagram ini dibuat untuk memudahkan anggota tim dalam menelusuri sumber data dan memahami konteks hubungan antar tabel. Dokumentasi ini juga menjadi referensi penting dalam pengembangan *query SQL* maupun pembuatan *pipeline data*, agar proses pengambilan dan pengolahan data menjadi lebih akurat dan efisien.

### 3.2.3 Dashboard and Data Internal Request

Tim Data di Staffinc memiliki tanggung jawab utama dalam menyelesaikan berbagai permintaan data dari divisi-divisi internal perusahaan, baik dalam bentuk *dashboard* maupun data mentah (*raw data*), sesuai dengan kebutuhan masing-masing. Permintaan tersebut umumnya disampaikan melalui *channel Slack* perusahaan bernama *#ask-data*, di mana para pengguna atau staff dari berbagai divisi akan mengajukan tiket permintaan, seperti pembuatan *dashboard* untuk klien, *dashboard internal*, pembaruan visualisasi yang telah ada, maupun permintaan *ad-hoc* lainnya kepada tim data. Selain melalui *Slack*, beberapa permintaan juga dapat disampaikan secara langsung dalam forum rapat

internal sebagai bagian dari proyek tertentu yang memerlukan dukungan visualisasi data khusus. Gambar 3.17 berikut merupakan contoh tampilan *channel #ask-data* di Staffinc yang terdiri dari tiket-tiket permintaan divisi lain kepada divisi data untuk dibuatkan permintaan dan juga menjadi forum untuk diskusi rincian tugas permintaan yang diberikan.



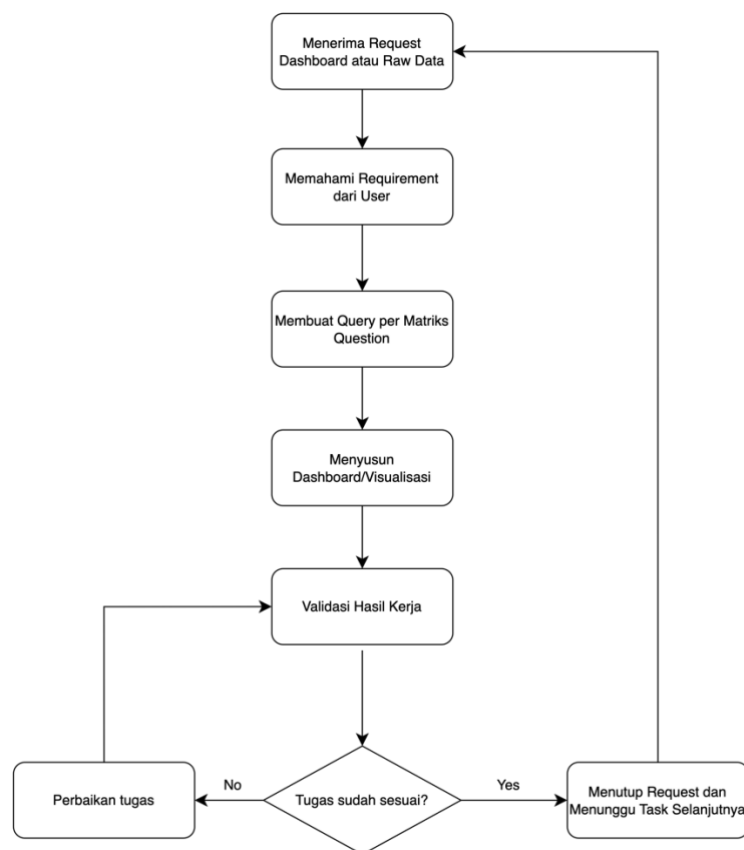
Gambar 3.17 Channel #ask-data Internal Slack

Selama program magang, beberapa permintaan internal telah diberikan kepada mahasiswa untuk diselesaikan secara mandiri. Proses pengerjaan dilakukan melalui *platform Metabase*, sebuah perangkat lunak visualisasi data yang digunakan perusahaan untuk membangun *dashboard*



interaktif serta menyajikan informasi dalam bentuk visual. Platform ini memungkinkan pengguna untuk menganalisis dan menyampaikan data dalam berbagai bentuk grafik dan tabel, serta mempermudah pemantauan kinerja berdasarkan indikator tertentu.

Alur penugasan internal untuk pembuatan *dashboard* dilakukan berdasarkan tahapan yang tergambar dalam diagram alur kerja pada Gambar 3.18 berikut.



Gambar 3.18 Alur kerja request dashboard internal

Alur kerja diagram pada Gambar 3.18 di atas dijelaskan sebagai berikut, setelah tugas diterima, langkah pertama dimulai dengan memahami kebutuhan dan tujuan dari *dashboard* yang diminta oleh user. Proses ini mencakup analisis terhadap informasi yang dibutuhkan, indikator yang akan ditampilkan, serta format visualisasi yang sesuai. Selanjutnya, dilakukan pembuatan *query* untuk setiap metrik atau pertanyaan yang diperlukan, yang mencakup pembuatan *table view* pada *database* serta proses perhitungan data yang relevan.

Hasil dari *query* tersebut kemudian disusun dalam bentuk kumpulan *card question* yang diatur menjadi sebuah *dashboard* di *Metabase*, dengan visualisasi yang disesuaikan berdasarkan jenis data dan kebutuhan user. Setelah *dashboard* selesai disusun, hasilnya dikirimkan kepada *user* untuk divalidasi. Apabila ditemukan revisi atau penyesuaian, perbaikan akan dilakukan mahasiswa sesuai arahan *user*, dan jika telah sesuai, maka proses dianggap selesai sambil menunggu penugasan berikutnya. Selama masa magang, total enam *request dashboard internal* sudah dikerjakan oleh mahasiswa yang menjadikan pembelajaran nyata dalam pembuatan *dashboard* di perusahaan untuk digunakan sebagai pengambilan keputusan dalam bisnis.

Rincian tugas dalam pengerjaan *request dashboard internal* yang diberikan akan dijelaskan pada bagian selanjutnya dari laporan ini.

### **3.2.3.1 EWA Monthly Report Dashboard**

EWA atau *Early Withdrawal Access* merupakan fitur penarikan gaji lebih awal yang ditawarkan oleh Staffinc kepada para pekerja dalam

program *outsourcing*. Melalui fitur ini, pekerja diberikan akses untuk menarik sebagian dari gaji bulanan mereka sebagai bentuk pinjaman atau kasbon. Setiap transaksi penarikan *EWA* dikenakan biaya administrasi, yang kemudian tercatat sebagai pendapatan perusahaan. Karena *EWA* berkontribusi terhadap penghasilan Staffinc, maka diperlukan pembuatan *dashboard* pelaporan yang dapat menyajikan rincian transaksi secara komprehensif. Saat ini, laporan tersebut belum tersedia dalam bentuk *dashboard*, sehingga tim data menginisiasi proyek pembuatan *EWA Monthly Report Dashboard* guna menyediakan laporan yang dapat memantau transaksi secara berkala.

Proyek *dashboard* ini dikerjakan secara mandiri sebagai bagian dari pelatihan pembuatan *dashboard* dan pendalaman penggunaan *platform Metabase*. Tahapan awal dimulai dengan memahami proses bisnis *EWA* melalui sesi *briefing* langsung dengan tim bisnis yang menangani program *EWA*. Selanjutnya, dilakukan eksplorasi dan pemahaman terhadap struktur dan isi database yang berkaitan dengan *EWA*. Berdasarkan hasil pemahaman tersebut, disusun sebuah *one-pager* berisi daftar metrik utama yang akan dijadikan dasar pembuatan *card* untuk visualisasi *dashboard* yang ditampilkan pada Gambar 3.19 di bawah ini.

# One Pager

EWA MONTHLY REPORT DASHBOARD	
<b>Background</b>	<p>Tim bisnis <b>belum mempunyai laporan transaksi EWA per-bulan</b>, yang mana menyebabkan susahnya monitoring transaksi EWA.</p> <p>Tim bisnis <b>membutuhkan laporan transaksi</b> dan juga eligibilitas dari masing-masing worker terhadap benefit EWA ini.</p>
<b>Objective</b>	Menyediakan laporan EWA yang akurat dan dapat diakses setiap bulan untuk mendukung keputusan operasional dan strategi bisnis.
<b>Metrics</b>	<ul style="list-style-type: none"><li>● # of transaction (per month)</li><li>● # of transaction (daily)</li><li>● # of eligible worker</li><li>● # of active agent</li><li>● % of eligible worker</li><li>● # of whitelisted client</li><li>● % of active user per eligible user</li><li>● Avg # of transaction per user</li><li>● Total GMV</li><li>● Total admin fee</li><li>● Monthly cohort</li></ul>
<b>Tools</b>	Metabase TablePlus

Gambar 3.19 One Pager EWA Dashboard

Setelah mempunyai acuan *metrics* yang akan dibuat, selanjutnya dilakukan pembuatan *query* per *metrics* sebagai berikut:

**a) # of Transaction (per month)**

Metrik pertama adalah *Number of Transactions per Month*, yang bertujuan menampilkan jumlah transaksi penarikan EWA setiap bulan. Data transaksi diolah melalui proses *inner join* ke beberapa tabel yang relevan seperti *cycle*, *user*, dan *agent*. Filter tambahan disediakan untuk memudahkan user dalam menyaring

data berdasarkan tanggal transaksi serta status keaktifan pekerja/agent. Data kemudian dikelompokkan dan diurutkan secara kronologis berdasarkan bulan untuk menampilkan tren transaksi. Selain itu, kolom *transaction status* seperti ‘disbursed’, ‘error’, dan ‘pending’ turut ditambahkan untuk memantau status eksekusi setiap transaksi. Query dari metrik ini ditampilkan pada Gambar 3.20 berikut ini.

```

EWA - # of Transaction per Year and Month BQ

BigQuery Start Date End Date T Agent Status

1 SELECT
2   FORMAT_TIMESTAMP('%Y-%m', TIMESTAMP(wr.created_at)) AS month,
3   COUNT(DISTINCT wr.transaction_id) AS total_transactions
4 FROM staging.fd_internal_v3_ewa_withdrawal_requests AS wr
5 INNER JOIN staging.fd_internal_v3_ewa_cycles AS ec
6   ON wr.ewa_cycle_id = ec.id
7 INNER JOIN staging.fd_internal_v3_ancillary_users AS au
8   ON ec.ancillary_user_id = au.id
9 INNER JOIN staging.fd_kerjaan_agent AS ag
10  ON au.agent_id = ag.id
11 INNER JOIN staging.fd_internal_v3_client AS c
12  ON ag.client_id = c.id
13 WHERE 1=1
14    [[AND ag.status = {{agent_status}}]]
15    [[AND DATE(wr.created_at) BETWEEN DATE({{start_date}}) AND DATE({{end_date}})]]
16 GROUP BY
17   month
18 ORDER BY
19   month ASC;
20

```

Gambar 3.20 Query Number of Transaction per Month

### b) # of Transaction (daily)

Selanjutnya adalah metrik *Number of Transactions per Day*, yang memiliki tujuan serupa dengan sebelumnya namun difokuskan pada pengelompokan harian. Pada Gambar 3.21 dijelaskan *query* yang digunakan pada dasarnya sama dengan transaksi bulanan, hanya saja bagian *timestamp* diubah menjadi *Day*. Status transaksi juga disertakan untuk memberikan konteks proses penarikan pada tiap hari.

EWA - # of Transaction per Day BQ

```
BigQuery v T Agent Status Start Date End Date
1 SELECT
2   DATE(wr.created_at) AS day,
3   COUNT(*) AS total_transactions
4 FROM staging.fd_internal_v3_ewa_withdrawal_requests AS wr
5 INNER JOIN staging.fd_internal_v3_ewa_cycles AS ec
6   ON wr.ewa_cycle_id = ec.id
7 INNER JOIN staging.fd_internal_v3_ancillary_users AS au
8   ON ec.ancillary_user_id = au.id
9 INNER JOIN staging.fd_kerjaan_agent AS ag
10  ON au.agent_id = ag.id
11 INNER JOIN staging.fd_internal_v3_client AS c
12  ON ag.client_id = c.id
13 WHERE 1=1
14   [[AND ag.status = {{status}}]]
15   [[AND DATE(wr.created_at) BETWEEN DATE({{start_date}}) AND DATE({{end_date}})]]
16 GROUP BY
17   day
18 ORDER BY
19   day;
20 |
```

Gambar 3.21 Query Number of Transaction per Day

### c) # of Eligible Workers

Metrik *Number of Eligible Workers* digunakan untuk mengetahui jumlah pekerja yang memenuhi syarat untuk melakukan penarikan EWA. Data ini diperoleh dari jumlah *ancillary\_user* yang memiliki akses terhadap modul ‘EWA’ pada tabel *ancillary\_access\_controls*, yang menunjukkan bahwa agen atau pekerja tersebut terverifikasi untuk mengikuti program EWA. Data ini diambil dengan menggunakan *query* sebagaimana ditampilkan pada Gambar 3.22 di bawah ini.

**EWA - # of Eligible Worker**

BigQuery

```

1 SELECT
2   COUNT(DISTINCT ac.ancillary_user_id) AS number_of_eligible_agents
3 FROM staging.fd_internal_v3_ewa_withdrawal_requests AS wr
4 INNER JOIN staging.fd_internal_v3_ewa_cycles AS ec
5   ON wr.ewa_cycle_id = ec.id
6 INNER JOIN staging.fd_internal_v3_ancillary_users AS au
7   ON ec.ancillary_user_id = au.id
8 INNER JOIN staging.fd_internal_v3_ancillary_access_controls AS ac
9   ON au.id = ac.ancillary_user_id
10 INNER JOIN staging.fd_kerjaan_agent AS ag
11   ON au.agent_id = ag.id
12 WHERE 1=1
13   AND ac.module_name = 'EWA'
14   [[AND ag.status = {{agent_status}}]]
15   [[AND DATE(wr.created_at) BETWEEN DATE('{{start_date}}') AND DATE('{{end_date}}')]]
16 |

```

Gambar 3.22 Query Number of Eligible Worker

**d) # of active agent**

Metrik *Number of Active Agents* disusun untuk menampilkan jumlah agen yang masih berstatus aktif di bawah naungan Staffinc. Perhitungan ini dilakukan secara sederhana dengan membaca entri dalam tabel *agent* yang memiliki status ‘active’ sebagaimana pada Gambar 3.23 di bawah ini.

**EWA - # of Active Agent BQ**

BigQuery

```

1 SELECT
2   COUNT(*) AS number_of_active_agents
3 FROM
4   staging.fd_kerjaan_agent
5 WHERE
6   status = 'active'|

```

Gambar 3.23 Query Number of Active Agent

**e) % of Eligible Workers**

*Percentage of Eligible Workers* dihitung dengan membandingkan jumlah pekerja yang memenuhi syarat dengan jumlah total pekerja aktif. Fungsi `SAFE_DIVIDE` digunakan dalam query ini untuk membagi nilai *eligible* dengan nilai total, dan hasilnya disimpan dalam kolom baru bertajuk *percentage\_eligible\_agents* dengan rincian query seperti pada Gambar 3.24 di bawah ini.

**EWA - % of Eligible Worker**

```

1 SELECT
2   SAFE_DIVIDE(eligible.count, total.count) AS percentage_eligible_agents
3 FROM (
4   SELECT COUNT(DISTINCT ac.ancillary_user_id) AS count
5   FROM staging.fd_internal_v3_ewa_withdrawal_requests AS wr
6   INNER JOIN staging.fd_internal_v3_ewa_cycles AS ec
7     ON wr.ewa_cycle_id = ec.id
8   INNER JOIN staging.fd_internal_v3_ancillary_users AS au
9     ON ec.ancillary_user_id = au.id
10  INNER JOIN staging.fd_internal_v3_ancillary_access_controls AS ac
11    ON au.id = ac.ancillary_user_id
12  INNER JOIN staging.fd_kerjaan_agent AS ag
13    ON au.agent_id = ag.id
14  WHERE ac.module_name = 'EWA'
15        [[AND ag.status = {{agent_status}}]]
16        [[AND DATE(wr.created_at) BETWEEN DATE('{{start_date}}') AND DATE('{{end_date}}')]]
17 ) AS eligible,
18 (
19   SELECT COUNT(DISTINCT id) AS count
20   FROM staging.fd_kerjaan_agent
21   WHERE status = 'active'
22        [[AND status = {{agent_status}}]]
23 ) AS total
24

```

Gambar 3.24 Query Percentage of Eligible Workers

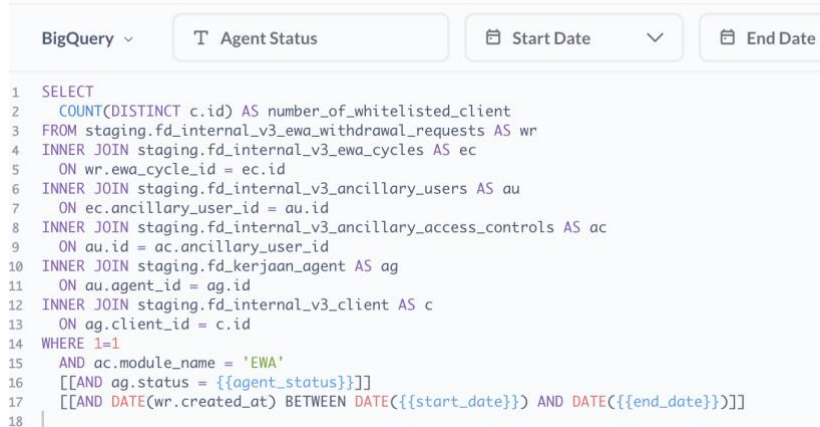
#### f) # of Whitelisted Clients

Metrik *Number of Whitelisted Clients* menyajikan informasi mengenai klien yang diperbolehkan untuk menyediakan fitur EWA bagi para pekerjanya. Penilaian terhadap kelayakan klien didasarkan pada sejumlah kriteria tertentu, salah satunya adalah faktor keamanan pekerjaan seperti kurir paket.



Klien yang dinilai tidak memenuhi standar keamanan biasanya tidak dianjurkan untuk mengaktifkan fitur ini. Pada gambar 3.25 dijelaskan *query* yang digunakan mirip dengan metrik *eligible workers*, yakni dengan memeriksa apakah pekerja dari klien tersebut memiliki akses terhadap modul *EWA*.

EWA - # of Whitelisted Client



```
1 SELECT
2   COUNT(DISTINCT c.id) AS number_of_whitelisted_client
3 FROM staging.fd_internal_v3_ewa_withdrawal_requests AS wr
4 INNER JOIN staging.fd_internal_v3_ewa_cycles AS ec
5   ON wr.ewa_cycle_id = ec.id
6 INNER JOIN staging.fd_internal_v3_ancillary_users AS au
7   ON ec.ancillary_user_id = au.id
8 INNER JOIN staging.fd_internal_v3_ancillary_access_controls AS ac
9   ON au.id = ac.ancillary_user_id
10 INNER JOIN staging.fd_kerjaan_agent AS ag
11   ON au.agent_id = ag.id
12 INNER JOIN staging.fd_internal_v3_client AS c
13   ON ag.client_id = c.id
14 WHERE 1=1
15   AND ac.module_name = 'EWA'
16   [[AND ag.status = {{agent_status}}]]
17   [[AND DATE(wr.created_at) BETWEEN DATE({{start_date}}) AND DATE({{end_date}})]]
18
```

Gambar 3.25 Query Number of Whitelisted Client

**g) % of active user per eligible user**

*Percentage of Active Users per Eligible User* adalah metrik yang menampilkan persentase pekerja aktif di antara mereka yang telah memenuhi syarat untuk mengikuti *EWA*. Perhitungan dilakukan kembali menggunakan fungsi *SAFE\_DIVIDE*, dengan membandingkan jumlah pekerja aktif dan jumlah pekerja yang *eligible*, dan hasilnya dicatat dalam kolom *percent\_active\_user\_from\_eligible* dan ditampilkan seperti pada Gambar 3.26 di bawah ini.

### EWA - % Active User per Eligible Worker

```
BigQuery T Agent Status Start Date End Date
1 SELECT
2   SAFE_DIVIDE(
3     active_ewa_user.total_active_user,
4     eligible.total_eligible_user
5   ) AS percent_active_user_from_eligible
6
7 FROM (
8   SELECT COUNT(DISTINCT ac.ancillary_user_id) AS total_eligible_user
9   FROM staging.fd_internal_v3_ewa_withdrawal_requests AS wr
10  INNER JOIN staging.fd_internal_v3_ewa_cycles AS ec
11    ON wr.ewa_cycle_id = ec.id
12  INNER JOIN staging.fd_internal_v3_ancillary_users AS au
13    ON ec.ancillary_user_id = au.id
14  INNER JOIN staging.fd_internal_v3_ancillary_access_controls AS ac
15    ON au.id = ac.ancillary_user_id
16  INNER JOIN staging.fd_kerjaan_agent AS ag
17    ON au.agent_id = ag.id
18  WHERE ac.module_name = 'EWA'
19    [[AND ag.status = {{agent_status}}]]
20    [[AND DATE(wr.created_at) BETWEEN DATE('{{start_date}}') AND DATE('{{end_date}}')]]
21 ) AS eligible,
22 (
23   SELECT COUNT(DISTINCT ac.ancillary_user_id) AS total_active_user
24   FROM staging.fd_internal_v3_ewa_withdrawal_requests AS wr
25   INNER JOIN staging.fd_internal_v3_ewa_cycles AS ec
26     ON wr.ewa_cycle_id = ec.id
27   INNER JOIN staging.fd_internal_v3_ancillary_users AS au
28     ON ec.ancillary_user_id = au.id
29   INNER JOIN staging.fd_internal_v3_ancillary_access_controls AS ac
30     ON au.id = ac.ancillary_user_id
31   INNER JOIN staging.fd_kerjaan_agent AS ag
32     ON au.agent_id = ag.id
33   WHERE ac.module_name = 'EWA'
34     AND wr.status = 'disbursed'
35     [[AND ag.status = {{agent_status}}]]
36     [[AND DATE(wr.created_at) BETWEEN DATE('{{start_date}}') AND DATE('{{end_date}}')]]
37 ) AS active_ewa_user
38
```

Gambar 3.26 Query Percentage of Active User per Eligible Workers

#### h) Avg # of transaction per user

Untuk metrik *Average Number of Transactions per User*, dilakukan perhitungan rata-rata transaksi yang dilakukan oleh masing-masing pekerja. Metrik ini berguna bagi tim bisnis untuk mengevaluasi kebiasaan transaksi pengguna serta mengidentifikasi potensi pengembangan fitur tambahan. Perhitungan dilakukan menggunakan fungsi *AVG* terhadap transaksi pengguna, dan data dikelompokkan serta diurutkan berdasarkan bulan untuk memberikan visualisasi yang lebih jelas.

Pada Gambar 3.27 di bawah ini ditampilkan penggunaan *query* untuk mendapatkan matrik *average transaction per user*.

#### EWA - Avg # of transaction per user

```
BigQuery v T Agent Status Start Date End Date
1 WITH transactions_per_agent_per_month AS (
2   SELECT
3     ag.id AS agent_id,
4     FORMAT_TIMESTAMP('%Y-%m', wr.created_at) AS month,
5     COUNT(wr.transaction_id) AS total_transactions
6   FROM staging.fd_internal_v3_ewa_withdrawal_requests AS wr
7   INNER JOIN staging.fd_internal_v3_ewa_cycles AS ec
8     ON wr.ewa_cycle_id = ec.id
9   INNER JOIN staging.fd_internal_v3_ancillary_users AS au
10    ON ec.ancillary_user_id = au.id
11  INNER JOIN staging.fd_kerjaan_agent AS ag
12    ON au.agent_id = ag.id
13  WHERE 1=1
14    [[AND ag.status = {{agent_status}}]]
15    [[AND DATE(wr.created_at) BETWEEN DATE('{{start_date}}') AND DATE('{{end_date}}')]]
16  GROUP BY agent_id, month
17 )
18
19 SELECT
20   month,
21   ROUND(AVG(total_transactions), 2) AS avg_transaction_per_user
22 FROM transactions_per_agent_per_month
23 GROUP BY month
24 ORDER BY month
25
```

Gambar 3.27 Query Average of Transaction per User

#### i) Total GMV

Metrik *Total GMV (Gross Merchandise Value)* dihitung untuk mengetahui total nilai transaksi yang terjadi dalam periode tertentu. Dalam konteks *EWA*, perhitungan ini digunakan untuk mengukur pendapatan keseluruhan yang dihasilkan dari fitur *EWA*. Proses ini dilakukan dengan menjumlahkan kolom *amount\_final* pada setiap transaksi seperti ditunjukkan pada Gambar 3.28 berikut ini.

### EWA - Total GMV

```
BigQuery v T Agent Status Start Date End Date
1 SELECT
2   SUM(wr.amount_final) AS total_gmv
3 FROM staging.fd_internal_v3_ewa_withdrawal_requests AS wr
4 INNER JOIN staging.fd_internal_v3_ewa_cycles AS ec
5   ON wr.ewa_cycle_id = ec.id
6 INNER JOIN staging.fd_internal_v3_ancillary_users AS au
7   ON ec.ancillary_user_id = au.id
8 INNER JOIN staging.fd_kerjaan_agent AS ag
9   ON au.agent_id = ag.id
10 INNER JOIN staging.fd_internal_v3_client AS c
11   ON ag.client_id = c.id
12 WHERE 1=1
13   [[AND ag.status = {{agent_status}}]]
14   [[AND DATE(wr.created_at) BETWEEN DATE({{start_date}}) AND DATE({{end_date}})]]
15
```

Gambar 3.28 Query Total GMV

### j) Total Admin Fee

*Total Admin Fee* merupakan salah satu metrik penting karena mencerminkan jumlah pendapatan bersih perusahaan dari biaya administrasi yang dikenakan pada setiap transaksi *EWA*. Nilai ini diperoleh dengan menjumlahkan seluruh entri pada kolom *admin\_fee* seperti pada Gambar 3.29 di bawah ini.

### EWA - Total Admin Fee

```
BigQuery v Start Date End Date T Agent Status
1 SELECT
2   SUM(wr.admin_fee) AS total_admin_fee
3 FROM staging.fd_internal_v3_ewa_withdrawal_requests AS wr
4 INNER JOIN staging.fd_internal_v3_ewa_cycles AS ec
5   ON wr.ewa_cycle_id = ec.id
6 INNER JOIN staging.fd_internal_v3_ancillary_users AS au
7   ON ec.ancillary_user_id = au.id
8 INNER JOIN staging.fd_kerjaan_agent AS ag
9   ON au.agent_id = ag.id
10 INNER JOIN staging.fd_internal_v3_client AS c
11   ON ag.client_id = c.id
12 WHERE 1=1
13   [[AND ag.status = {{agent_status}}]]
14   [[AND DATE(wr.created_at) BETWEEN DATE({{start_date}}) AND DATE({{end_date}})]]
15
```

Gambar 3.29 Query Total Admin Fee

### k) Monthly Cohort

Metrik *Monthly Cohort* digunakan untuk mengelompokkan pengguna atau pekerja yang memiliki kesamaan karakteristik berdasarkan bulan pertama kali mereka melakukan transaksi penarikan EWA. Dalam konteks ini, pengelompokan diperlukan untuk memantau konsistensi pengguna dalam menggunakan fitur *EWA* dari bulan ke bulan. Pada Gambar 3.29 berikut menjelaskan *query* yang disusun dengan mengidentifikasi *first\_transaction* dan transaksi bulanan lainnya berdasarkan *user\_id*, sehingga dapat terlihat pola perilaku transaksi setiap pekerja sejak bulan pertama mereka melakukan penarikan.

EWA - Monthly Cohort

```

BigQuery v
1 WITH first_txn AS (
2   SELECT
3     ec.ancillary_user_id,
4     DATE_TRUNC(MIN(DATE(wr.created_at)), MONTH) AS cohort_month
5   FROM staging.fd_internal_v3_ewa_withdrawal_requests wr
6   JOIN staging.fd_internal_v3_ewa_cycles ec ON wr.ewa_cycle_id = ec.id
7   WHERE wr.status = 'disbursed'
8   GROUP BY ec.ancillary_user_id
9 )
10
11 monthly_txn AS (
12   SELECT
13     ec.ancillary_user_id,
14     DATE_TRUNC(DATE(wr.created_at), MONTH) AS txn_month
15   FROM staging.fd_internal_v3_ewa_withdrawal_requests wr
16   JOIN staging.fd_internal_v3_ewa_cycles ec ON wr.ewa_cycle_id = ec.id
17   WHERE wr.status = 'disbursed'
18 )
19
20 cohort_activity AS (
21   SELECT
22     f.cohort_month,
23     m.txn_month,
24     DATE_DIFF(m.txn_month, f.cohort_month, MONTH) AS month_num,
25     m.ancillary_user_id
26   FROM first_txn f
27   JOIN monthly_txn m USING (ancillary_user_id)
28 )
29
30 SELECT
31   cohort_month,
32   month_num,
33   COUNT(DISTINCT ancillary_user_id) AS active_user_count
34 FROM cohort_activity
35 GROUP BY cohort_month, month_num
36 ORDER BY cohort_month, month_num
37

```

Gambar 3.29 Query Monthly Cohort

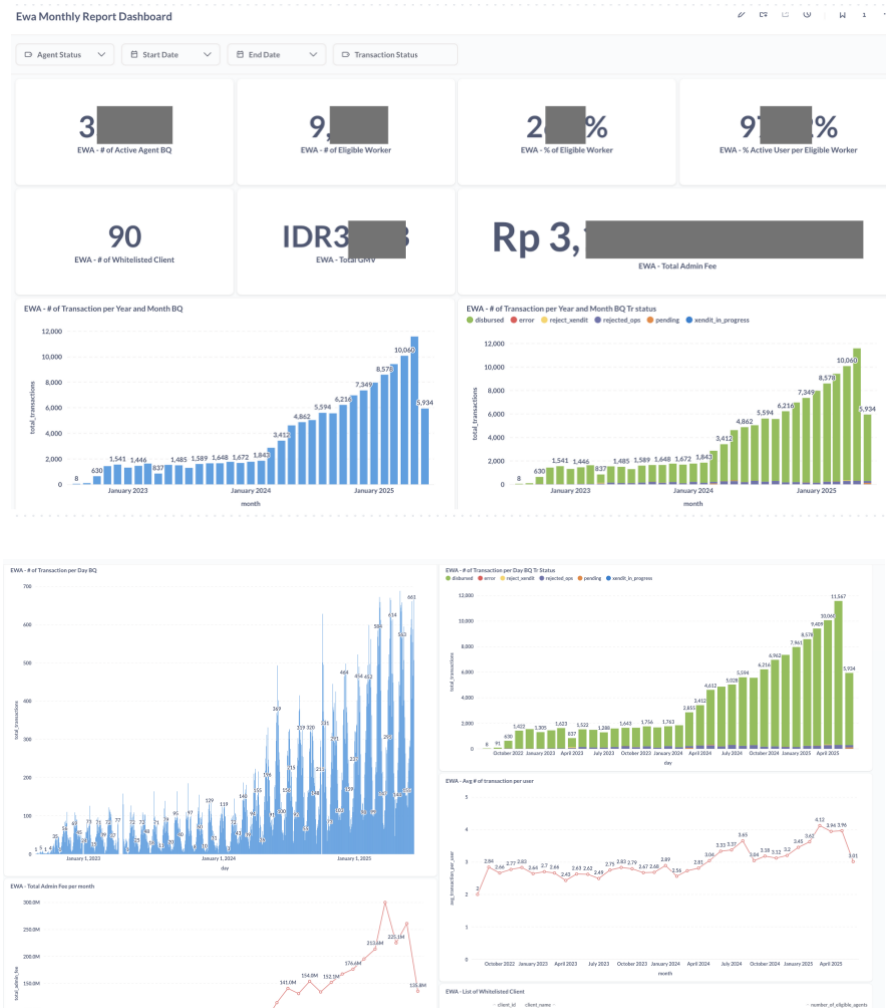
Hasil dari metrik ini ditampilkan pada Gambar 3.30 berikut dalam bentuk *pivot table* yang mencakup *cohort\_month* sebagai baris, dan bulan-bulan berikutnya sebagai kolom. Tabel ini menampilkan angka ‘1’ jika pekerja melakukan penarikan pada bulan tertentu, dan ‘0’ jika tidak melakukan penarikan. Dengan format ini, tim bisnis dapat dengan mudah mengamati retensi dan perilaku penggunaan fitur EWA oleh para pekerja dari waktu ke waktu.

cohort_month	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
August 1, 2022	4	2	1	1	1	1	1	1								
September 1, 2022	30	15	14	7	4	5	3	2	2	1	1	1	1	1	1	1
October 1, 2022	211	172	155	113	99	105	72	67	63	27	25	24	19	21	18	14
November 1, 2022	326	250	197	138	127	61	86	90	74	78	70	66	64	52	52	47
December 1, 2022	128	73	65	63	37	51	44	36	29	29	27	22	22	20	16	18
January 1, 2023	114	70	67	37	53	47	38	36	34	27	28	18	23	19	22	19
February 1, 2023	165	127	62	92	76	62	58	48	45	30	27	24	16	12	10	12
March 1, 2023	123	38	61	47	45	42	37	32	22	18	16	15	15	12	9	5
April 1, 2023	68	41	30	22	16	16	16	13	11	9	10	10	9	7	8	5
May 1, 2023	184	110	88	67	57	55	44	40	34	31	33	22	23	10	11	11

Gambar 3.30 Hasil Monthly Cohort

Seluruh metrik yang telah dijelaskan kemudian dikompilasi ke dalam satu *dashboard* komprehensif bertajuk ‘*EWA Monthly Report Dashboard*’, yang dirancang secara interaktif menggunakan *Metabase*. *Dashboard* ini dilengkapi dengan beberapa filter tambahan seperti rentang tanggal (*date range*), status agen (*agent status*), dan status transaksi (*transaction status*) untuk memudahkan tim bisnis dalam memantau performa serta mengelola operasional program *EWA* secara lebih efisien dan

informatif. Gambar 3.31 berikut merupakan tampilan akhir dari *dashboard EWA Monthly Report* yang telah dikembangkan.



Gambar 3.31 EWA Monthly Report Dashboard

### 3.2.3.2 Metabase Internal Usage Dashboard

Dashboard *Metabase Internal Usage* dikembangkan sebagai alat pelaporan yang menampilkan penggunaan Metabase secara internal di lingkungan perusahaan. Informasi yang disajikan mencakup data pengguna (user), peran masing-masing pengguna (role), waktu login terakhir (*last login*), serta memberikan wawasan tambahan melalui dua metrik utama yaitu *Most Viewed Dashboard (Top 10)* dan *Most Viewed Card (Top 10)*. Melalui dashboard ini, ringkasan penggunaan Metabase oleh internal perusahaan dapat dipantau secara lebih efektif, sehingga mendukung pengelolaan akses dan optimalisasi penggunaan Metabase.

#### **3.2.3.3 Attendance Report All Project Dashboard**

Dashboard *Attendance Report All Project* berfungsi untuk memantau data kehadiran para pekerja di seluruh proyek dan klien yang berada di bawah naungan perusahaan. Data yang ditampilkan meliputi informasi detail seperti tanggal, hari, ID agen, nama agen, cabang, nama klien, posisi, shift kerja, waktu *clock in* dan *clock out*, durasi jam kerja, serta status kehadiran pekerja. Selain itu, dashboard ini juga dilengkapi dengan ringkasan hasil absensi (*summary attendance*) untuk memudahkan tim bisnis dalam memahami dan mengevaluasi pola kehadiran secara menyeluruh.

#### **3.2.3.4 Specify Client Attendance and Overtime Raw Data**

Sebagai tindak lanjut dari kebutuhan tim bisnis, dibuat dashboard dan data mentah (*raw data*) yang disesuaikan untuk masing-masing klien guna memantau absensi dan lembur pekerja

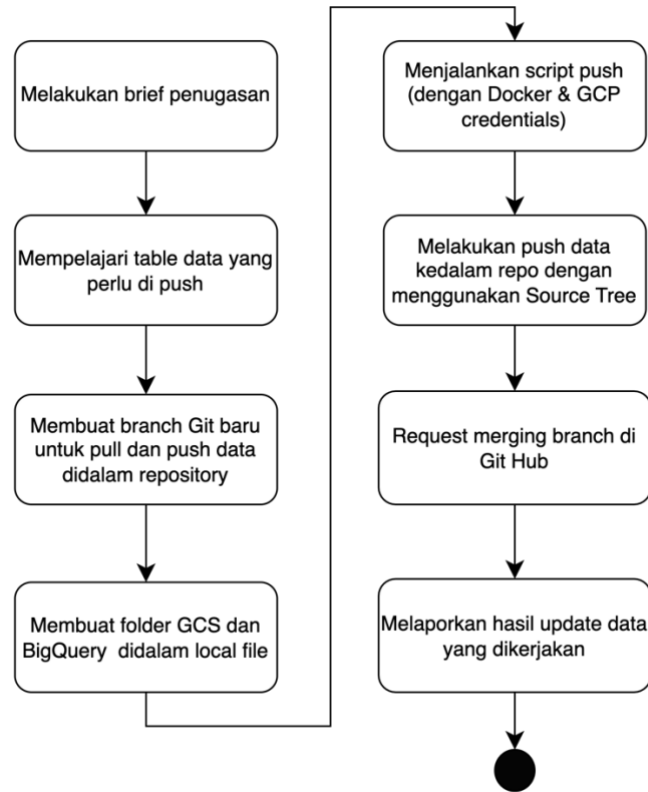


secara spesifik. Permintaan ini muncul karena setiap unit bisnis yang menangani klien tertentu membutuhkan laporan yang lebih terperinci agar proses pemantauan dan pengelolaan data kehadiran dapat dilakukan dengan lebih optimal. Selain itu, dashboard ini juga memungkinkan tim bisnis untuk memberikan akses pelaporan yang relevan secara langsung kepada masing-masing klien sesuai kebutuhan.

### 3.2.4 Handling Data Pipeline

Pada bagian tugas ini, ruang lingkup pekerjaan mulai memasuki ranah data pipeline, yang meliputi proses pengembangan, pemantauan, serta pemeliharaan pipeline data. Salah satu tugas utama yang diberikan adalah melakukan push data dari *database warehouse* ke *Google Cloud Storage (GCS)* dan *BigQuery* dengan memanfaatkan *tools* seperti *Git Hub*, *Sourcetree*, dan *Docker*.

Dalam pengerjaan tugas *handling data pipeline* dilakukan alur kerja yang sedikit berbeda dan ditampilkan pada alur diagram Gambar 3.32 berikut:



Gambar 3.32 Alur Pengerjaan Data Pipeline

Berdasarkan diagram alur kerja pada Gambar 3.32 diatas, berikut adalah tahapan-tahapan proses pengerjaan tugas *data pipeline*:

1. Proses pengelolaan *data pipeline* diawali dengan melakukan *brief* penugasan dari mentor atau atasan. Pada tahap ini, informasi mengenai tabel yang perlu diproses serta tujuan *push* data ke *cloud* akan dijelaskan secara umum.
2. Setelah *brief* diterima, tahap berikutnya adalah mempelajari struktur tabel data yang akan di *push*. Pemeriksaan dilakukan untuk memastikan

kelengkapan data, memahami atribut penting seperti *timestamp*, dan menilai dependensi antar tabel.

3. Setelah memahami data yang dibutuhkan, dibuat *branch* baru pada Git untuk mengisolasi pekerjaan dan menjaga kestabilan *main repository*. *Branch* ini akan digunakan sebagai jalur pengembangan lokal sebelum perubahan di *push* ke repository utama.
4. Selanjutnya dilakukan pembuatan folder dan file konfigurasi untuk *Google Cloud Storage (GCS)* dan *BigQuery* di dalam direktori lokal. Struktur file ini disesuaikan dengan kebutuhan tabel yang akan ditambahkan ke *pipeline*.
5. Tahap berikutnya adalah membuat dan menjalankan *script push* data ke *GCS* dan *BigQuery* menggunakan *Docker*. Eksekusi dilakukan dari lokal dengan akses kredensial yang telah dikonfigurasi, serta menggunakan *environment* yang sesuai dengan standar *pipeline* perusahaan.
6. Setelah proses *push* berhasil dilakukan, data yang sudah dimasukkan ke dalam *repository* kemudian dipush ke *remote repository* menggunakan *SourceTree*. *Tool* ini digunakan untuk mengelola *version control* secara visual dan lebih terstruktur.
7. Setelah data berhasil di *push* ke *remote branch*, langkah selanjutnya adalah melakukan *request merge* ke *main branch* melalui *GitHub*. Proses ini membutuhkan *review* dari tim atau mentor sebagai bentuk validasi akhir sebelum data dianggap resmi masuk ke sistem produksi.

8. Tahapan terakhir adalah melaporkan hasil pekerjaan yang telah dilakukan kepada mentor. Laporan berisi konfirmasi bahwa data telah berhasil ditambahkan ke *cloud* dan sudah tersimpan di dalam *repository* utama.

### **3.2.5 Sailors Program Events**

Pada program magang *SAILORS* di Staffinc, terdapat beberapa event rutin yang di adakan oleh *People Team* untuk melihat perkembangan para mahasiswa magang selama bekerja di Staffinc. Rangkaian kegiatan ini, dilaksanakan secara bertahap diluar pekerjaan utama dari masing-masih divisi mahasiswa bekerja dengan beberapa tujuan seperti untuk meningkatkan pemahaman mahasiswa magang tentang perusahaan, mendapatkan pengalaman dan wawasan baru, serta menjalin kebersamaan antara peserta magang lainnya dan juga pekerja di Staffinc. Beberapa kegiatan yang dilakukan diantaranya adalah:

#### **3.2.5.1 On Boarding Session with People Team**

*On boarding session with People Team* dilaksanakan pada tanggal 17 Februari 2025 yang merupakan hari pertama para mahasiswa magang melaksanakan praktik magang. Dalam sesi ini, mahasiswa dikumpulkan dengan para peserta magang lainnya didalam *event hall* Staffinc pada pukul 09.00 WIB untuk memulai kegiatan. Rangkaian kegiatan yang dilakukan mulai dari sambutan dari *People Team*, perkenalan masing-masing peserta magang agar lebih mengenal teman-teman seperjuangan. Pada gambar 3.33

berikut ditampilkan hasil dokumentasi *on boarding session* bersama *People Team* dan juga *CEO Staffinc*.



Gambar 3.33 SAILORS On Boarding

Kemudian sesi utama yaitu penyambutan peserta magang dan pengenalan perusahaan oleh *CEO (Chief Executive Officer)* Staffinc yang mana diisi dengan sejarah berdirinya Staffinc Group yang semula merupakan sebuah perusahaan bernama Sampingan pada tahun 2018 yang merupakan sebuah bisnis kecil yang kemudian menjadi Staffinc Group sekarang. Setelah sesi ini oleh *CEO*, dilanjutkan kembali oleh *People Team* untuk membahas *SAILORS Handbook* yang mana merupakan sebuah jurnal yang berisi info-info dasar perusahaan serta kolom pengembangan diri peserta magang selama mengikuti program magang di Staffinc.

Sesi *on boarding* ini diakhiri dengan *office tour* Dimana para peserta magang diperkenalkan dan dibawa mengelilingi kantor Staffinc yang terdiri dari 2 lantai dan mengenalkan ruangan-

ruangan yang akan digunakan oleh peserta magang saat bekerja nantinya. Setelah melaksanakan sesi ini, *People Team* kemudian mengarahkan para peserta magang kepada masing-masing divisi untuk kemudian melanjutkan sesi bersama *user* divisi masing-masing peserta magang.

### **3.2.5.2 Mini Town Hall Divisi EOP (Engineering, Operation, Product)**

*Town hall* merupakan sebuah rangkaian acara rutin dari Staffinc yang dilaksanakan setiap kuartil di perusahaan guna untuk mengetahui perkembangan yang ada didalam perusahaan. Sesi ini dibagi menjadi 2 rangkaian besar yaitu *Town Hall Quartil* dan juga *Mini Town Hall Division*. Pada kegiatan ini, mahasiswa berkesempatan mengikuti *Mini Town Hall* divisi *EOP (Engineering, Operation, Product)* pada tanggal 27 Februari 2025 yang merupakan rangkaian acara update divisi EOP selama kuartil awal tahun 2025. Pada kegiatan ini, mahasiswa mendapatkan wawasan apa saja yang sudah dikerjakan, sedang dikerjakan, serta akan dikerjakan dari masing-masing tim divisi EOP.

### **3.2.5.3 First Month Catch Up Session**

Pada sesi ini, peserta magang akan dipertemukan dengan penanggung jawab dari *People Team* dan akan dilakukan sesi *sharing* pengalaman bekerja selama satu bulan di Staffinc. Pada sesi ini, mahasiswa didampingi kak Hanifa selaku *People Acquisition* untuk berbagi pengalaman selama mengikuti 1 bulan magang. Kak Hanifa menanyakan apa saja yang sudah dikerjakan

selama satu bulan kebelakang, apa kebutuhan yang perlu di sediakan oleh perusahaan, hingga kendala-kendala yang terjadi selama satu bulan bekerja disana. Pada sesi ini, mahasiswa sangat menghargai kesempatan yang diberikan sehingga dapat bertukar pikiran selama kurang lebih 1 jam dengan *People Team*. Dengan sesi ini, diharapkan dari kedua belah pihak memiliki ekspektasi masing-masing yang perlu dipenuhi tanpa adanya tekanan ataupun paksaan sehingga melancarkan kegiatan magang di perusahaan.

#### **3.2.5.4 Town Hall Quartil 1 Staffinc**

Pada kegiatan *Town Hall Quartil 1 Staffinc* yang dilaksanakan pada tanggal 20 Maret 2025 yang ditampilkan pada Gambar 3.34 di bawah ini, mahasiswa berkesempatan mengikuti rangkaian acara besar perusahaan yang dilaksanakan bertepatan dengan bulan Ramadhan sehingga diikuti dengan rangkaian kegiatan lain selain *quarterly update* masing-masing divisi di perusahaan juga dilaksanakan acara lain seperti *mini games*, *kultum*, dan juga ditutup dengan acara buka bersama di perusahaan. Pada kegiatan ini, mahasiswa dapat melihat perkembangan dari masing-masing divisi, serta kebersamaan para pekerja di Staffinc Group yang mana memiliki suasana kekeluargaan yang sangat erat dan asik.



Gambar 3.34 Town Hall Quartil 1 Staffinc

### 3.2.5.5 Chemistry Event Vol.1 Learning with Leader: Staffinc Plus

*Chemistry Event* merupakan rangkaian *event* yang disusun oleh *People Team* untuk meningkatkan wawasan para peserta magang selama bekerja di Staffinc. Program ini juga memungkinkan peserta magang untuk berinteraksi, bermain, serta bekerja bersama peserta magang lain untuk meningkatkan solidaritas dan melatih kerja sama tim pada masing-masing individu di *SAILORS* Program.

Pada *Chemistry Event Vol.1* yang dilaksanakan pada tanggal 09 April 2025 seperti pada Gambar 3.35 di bawah, dilakukan sesi belajar bersama dengan salah satu Staffinc *Leader* tepatnya divisi Staffinc *Plus* yang membawakan tema '*How to do a better communication*', dimana dijelaskan bahwa dalam menyampaikan suatu informasi, kita diperlukan untuk mengetahui tujuan dan metode penyampaian kita, *audiens* yang ingin kita sampaikan informasinya, serta harus mengetahui bagaimana alur cara kita akan menyampaikannya. Dengan dilengkapi sesi praktik bersama



tim, menjadikan sesi ini sebuah pengalaman baru yang dilakukan selama magang.



Gambar 3.35 Chemistry Event Vol.1

### 3.2.5.6 Chemistry Event Vol.2 Learning with Leader: Personal Branding

Pada *event* ini diberikan tips dan trik untuk melakukan *personal branding* yang kuat sesuai dengan bidang masing-masing. Diberikan bahwa dengan memahami diri sendiri, maka kita bisa berbagi informasi kepada orang lain mulai dari hal terkecil tapi akan sangat berpengaruh kedepannya. Gambar 3.36 menampilkan masing-masing peserta *SAILORS Staffinc* yang mengikuti *event chemistry volume 2* yang dilaksanakan pada 30 April 2025.



Gambar 3.36 Chemistry Event Vol.2

### 3.2.5.7 SAILORS Graduation

Pada event ini diadakan sesi *sharing* dan diskusi pengalaman magang selama menjadi *SAILORS*, menyaksikan penugasan ‘*A Day in My Life as SAILORS*’, sesi *awarding* “*Best SAILORS*” dan juga “*Best Video*”. Acara ditutup dengan pembuatan video testimoni bersama *People Team*. Event ini memberikan apresiasi atas kontribusi peserta selama program berlangsung dan menjadi penutup yang berkesan bagi seluruh peserta magang. Gambar 3.37 merupakan dokumentasi bersama teman-teman *SAILORS* dan juga *People Team* dengan mengenggam piagam penghargaan penyelesaian program *SAILORS* di Staffinc.



Gambar 3.37 Graduation SAILORS

### **3.2.6 Analisis *People, Technology, dan Organization (PTO)***

Dalam pelaksanaan program magang sebagai *Data Operations Intern* di Staffinc Group, sejumlah aspek yang relevan dengan pendekatan *People, Technology, dan Organization (PTO)* telah diidentifikasi. Ketiga aspek ini memiliki peran yang signifikan dalam mendukung kelancaran proses kerja harian, pengelolaan data, serta kolaborasi lintas divisi.

#### **3.2.6.1 People: Kolaborasi dan Komunikasi dalam Tim Data**

Selama kegiatan magang, pola komunikasi yang dominan antara tim data dengan divisi lainnya dilakukan melalui platform Slack. Sebuah channel khusus telah disediakan untuk menerima berbagai permintaan data, seperti permintaan pembuatan dashboard, laporan, maupun data mentah. Komunikasi dilakukan

secara threaded, sehingga dokumentasi percakapan dapat tertata dengan rapi dan mudah ditelusuri.

Melalui praktik ini, dapat terlihat bahwa budaya kerja yang diterapkan bersifat kolaboratif dan terbuka. Permintaan dari pengguna ditindaklanjuti secara responsif oleh tim data. Namun, pendekatan ini juga menuntut pengelolaan waktu dan prioritas yang baik untuk menghindari penumpukan permintaan secara bersamaan.

### 3.2.6.2 Technology: Otomatisasi, Integrasi, dan Visualisasi Data

Selama program magang, telah dilakukan sejumlah kegiatan teknis yang mencerminkan pemanfaatan teknologi dalam operasional data, antara lain:

1. Penyusunan dokumentasi arsitektur alur data yang mencakup *AWS*, *Airflow*, *GCS*, dan *BigQuery*.
2. Penggunaan skrip *Python* untuk melakukan otomasi, seperti pengambilan data dari *Airflow API* ke *Google Sheets*, serta pemrosesan data dari folder lokal secara otomatis.
3. Pengembangan *data pipeline* dari *database* lokal *MariaDB* ke *GCS* dan *BigQuery* melalui repositori *GitHub*.
4. Pembuatan *dashboard* laporan unit bisnis menggunakan platform *Metabase*.

Teknologi-teknologi tersebut berperan penting dalam meningkatkan efisiensi, otomasi, dan skalabilitas sistem pengelolaan data di Staffinc Group. Selain itu, dokumentasi alur

data juga membantu dalam memastikan integrasi yang menyeluruh antar sistem dan meminimalkan kesalahan dalam proses pengolahan.

### **3.2.6.3 Organization: Struktur dan Budaya Perusahaan**

Selama masa magang, partisipasi dalam berbagai kegiatan organisasi telah dilakukan, termasuk *Mini Townhall* Divisi EOP, *Townhall* Kuartal, dan *Chemistry Event*. Kegiatan-kegiatan tersebut menunjukkan bahwa struktur organisasi di Staffinc mendorong transparansi, komunikasi terbuka, serta peningkatan keterlibatan anggota tim.

Kegiatan seperti *First Month Catch-Up* yang difasilitasi oleh *People Team* menunjukkan perhatian perusahaan terhadap pengalaman dan perkembangan peserta magang. Budaya organisasi yang terbuka dan suportif tersebut memberikan ruang bagi pembinaan, kolaborasi, serta pertumbuhan profesional secara menyeluruh.

## **3.3 Kendala yang Ditemukan**

Kendala yang dialami selama mengikuti program magang di Staffinc ini adalah sebagai berikut:

1. Kegiatan magang dilaksanakan secara *hybrid* yang menimbulkan beberapa kendala seperti keterbatasan interaksi secara langsung dengan supervisor sehingga dapat mempengaruhi kekurangan pemahaman materi dan tugas yang disampaikan.

2. Beberapa tools baru memiliki tingkat kesulitan yang lumayan untuk mahasiswa yang sebelumnya belum pernah dipergunakan di perkuliahan seperti *Apache Airflow*, *Metabase*, *Github*, dan juga *Docker* sehingga menimbulkan kesulitan untuk mengerjakan tugas yang diberikan.
3. Dalam proyek membuat *Dashboard Monthly Report EWA* belum memiliki *dashboard* awal yang kemungkinan menjadi acuan, sehingga mahasiswa mengalami beberapa kesulitan seperti penyesuaian permintaan dari tim bisnis terhadap metrik yang ada didalam *dashboard* serta beberapa data yang sulit di kumpulkan dalam *database* yang besar.

### 3.4 Solusi atas Kendala yang Ditemukan

Solusi yang didapatkan untuk menyelesaikan kendala yang didapatkan selama program magang adalah sebagai berikut:

1. Rutin melakukan pertemuan virtual bersama mentor untuk pembahasan tugas agar lebih dimengerti lewat *Zoom* atau *Google Meet*. Staffinc juga menyediakan platform komunikasi *Slack* yang bisa digunakan untuk bertukar pesan, file, maupun dokumen-dokumen lain untuk mempermudah proses kerja *hybrid* selama program magang berlangsung.
2. Mempelajari penggunaan *tools* baru dengan otodidak dan bertanya kepada mentor dan tim data lainnya. Melakukan penugasan secara langsung agar lebih familiar menggunakan *tools* yang digunakan.
3. Mempelajari matriks yang dibutuhkan oleh tim bisnis dan dicoba divisualisasikan sesuai kebutuhan tim bisnis.