

BAB III

PELAKSANAAN KERJA

3.1 Kedudukan dan Koordinasi

Bagian Kedudukan dan Koordinasi disajikan sebagai satu subbab utama yang kemudian diuraikan kembali ke dalam dua subsubbab. Bagian Kedudukan menjelaskan posisi serta peran yang dijalankan selama program magang dalam struktur organisasi perusahaan. Sementara itu, bagian Koordinasi memaparkan alur komunikasi dan mekanisme supervisi yang diterapkan selama penyelesaian tugas. Pengantar ini memberikan gambaran awal mengenai ruang lingkup kegiatan magang dan menjadi dasar untuk memahami pembahasan yang disajikan pada subsubbab selanjutnya.

3.1.1. Kedudukan

Posisi *Business Intelligence & Data Warehouse Modelling Intern* di PT. Bumi Serpong Damai Tbk. berada di bawah Departemen *BI & Visualization*, tepatnya pada *section BI & Data Warehouse (DWH) Modelling*. *Section* ini dipimpin oleh satu *Section Head* yang menjadi penanggung jawab utama selama program magang berlangsung. Selain itu, terdapat juga satu karyawan pada *section* yang berperan sebagai penanggung jawab harian dan memberikan arahan teknis terkait tugas yang diberikan.

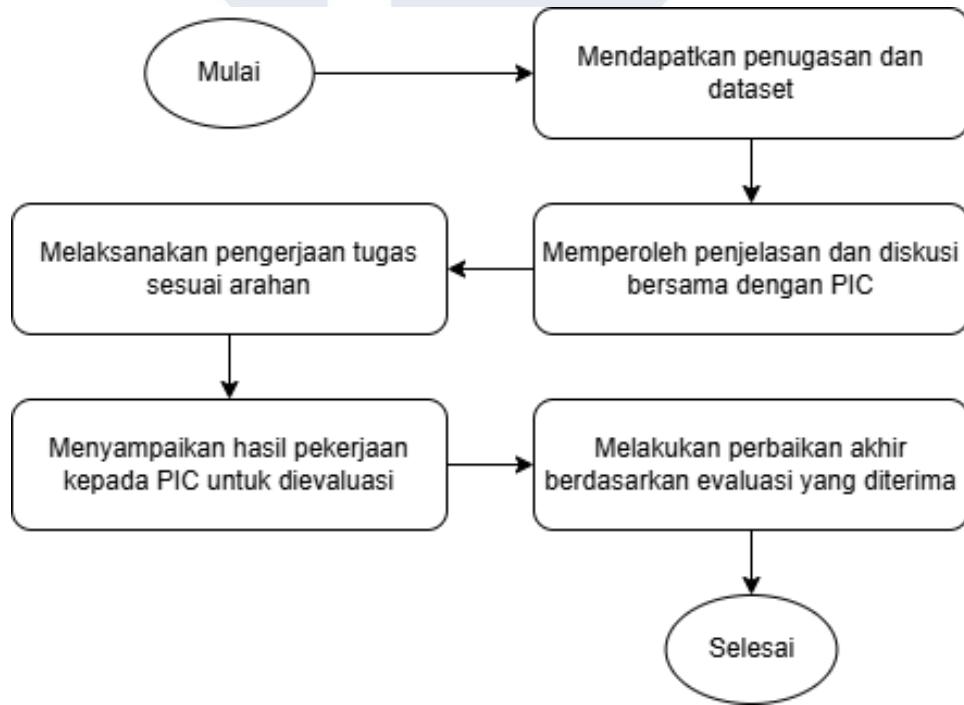
Penugasan dapat berasal langsung dari *Section Head* maupun dari karyawan yang bertanggung jawab menyesuaikan dengan kebutuhan pekerjaan. Setiap tugas yang diberikan karyawan selalu berada dalam sepenuhnya *Section Head*, sehingga alur supervisi tetap terstruktur dan transparan. Seluruh hasil pekerjaan tetap dilaporkan kepada *Section Head* sebagai bentuk pelaporan akhir dan validasi kualitas pekerjaan.

Koordinasi kerja berlangsung melalui komunikasi rutin, baik secara langsung maupun melalui media internal perusahaan. Mekanisme komunikasi yang terbuka memastikan setiap instruksi dapat dipahami secara

tepat dan potensi kesalahan dapat diminimalkan. Lingkungan kerja yang kooperatif dan profesional mendukung kelancaran proses supervisi dan penyelesaian tugas selama masa magang.

3.1.2. Koordinasi

Pelaksanaan tugas selama magang mengikuti mekanisme pendampingan yang berlaku di *section BI & DWH Modelling* di bawah Departemen *Business Intelligence & Visualization* PT. Bumi Serpong Damai Tbk. Proses dimulai ketika *Person in Charge* (PIC) yang terdiri dari *Section Head* dan satu staf pendamping menyampaikan penugasan beserta dataset yang akan digunakan. PIC akan menjelaskan terlebih dahulu terkait penugasan yang diberikan dan memberikan ruang diskusi bersama dengan PIC untuk memastikan pemahaman terhadap konteks penugasan. Pada tahap ini, ruang lingkup dan target *output* pekerjaan ditetapkan agar proses pengerjaan berjalan terarah.



Gambar 3.1 Bagan Alur Koordinasi

Pekerjaan kemudian dilaksanakan sesuai arahan yang telah diberikan. Selama proses pengerjaan, komunikasi tetap berlangsung secara berkala untuk memastikan setiap kendala dapat segera disampaikan. Setelah tugas

mencapai tahap penyelesaian, hasil kerja disampaikan kembali kepada PIC untuk melalui proses evaluasi. Masukan yang diberikan menjadi dasar untuk melakukan revisi akhir agar *output* pekerjaan memenuhi tujuan yang diinginkan. Setiap *output* yang telah direvisi selalu dikonfirmasi kembali kepada *Section Head* sebagai penanggung jawab utama dalam struktur koordinasi.

Secara keseluruhan, proses koordinasi ini mengikuti bagan alur pada Gambar 3.1, yang menggambarkan seluruh tahapan mulai dari penerimaan tugas, diskusi awal, penggerjaan, evaluasi oleh PIC, hingga revisi dan penyelesaian akhir. Alur tersebut memastikan setiap tugas dilaksanakan secara sistematis, terdokumentasi, dan sesuai ekspektasi departemen.

3.2 Tugas yang Dilakukan

Program magang selama enam bulan di Departemen *Business Intelligence & Visualization* memberikan kesempatan untuk terlibat secara langsung dalam proses pengolahan dan pemodelan data di *Section Business Intelligence (BI) & Data Warehouse (DWH) Modelling*. Pada posisi *Business Intelligence & Data Warehouse Modelling Intern*, fokus pekerjaan berada pada pembuatan dan pengembangan *query*, perancangan model data, serta penyusunan struktur data yang mendukung kebutuhan *Data Warehouse*. Selain itu, pekerjaan juga mencakup pembuatan dan penyempurnaan *dashboard* berdasarkan kebutuhan *section*. Seluruh aktivitas dilaksanakan berdasarkan arahan *Person in Charge* (PIC) yang memberikan bimbingan teknis sekaligus melakukan validasi terhadap hasil pekerjaan. Setiap tugas yang diberikan dipastikan selaras dengan standar *section* sehingga dapat mendukung kebutuhan analitik yang sedang berjalan. Inisiatif kerja juga menjadi aspek penting agar proses penyelesaian tugas tidak sepenuhnya bergantung pada instruksi PIC dan setiap hasil akhir dapat memenuhi kebutuhan analitik dengan tepat. Seluruh rangkaian kegiatan dan hasil pekerjaan ini bertujuan mengaplikasikan pemahaman konseptual menjadi solusi analitik yang terukur dan berdampak nyata bagi operasional departemen. Oleh karena itu, sebagai bentuk pertanggungjawaban atas durasi magang, Tabel 3.1 berikut menyajikan realisasi kegiatan secara rinci sesuai rencana kerja pada Tabel 1.2.

Selain itu, keberhasilan dalam menjalankan seluruh tanggung jawab juga sangat didukung oleh penguasaan dan integrasi berbagai perangkat lunak utama yang membentuk ekosistem *data warehousing* perusahaan. Setiap tools memiliki peran spesifik yang berkontribusi pada efisiensi alur kerja dari awal hingga akhir. *SQL Server Management Studio* (SSMS) merupakan *interface* utama yang digunakan untuk berinteraksi dengan *database* dan mengimplementasikan logika pemodelan data. Selama masa magang, SSMS menjadi lingkungan kerja primer untuk pembuatan, pengembangan, dan validasi *query SQL* yang kompleks. Tugas *intern* berpusat pada penulisan *View* harmonisasi data yang berfungsi sebagai *staging* area logis sebelum data dimuat ke dalam *Data Warehouse*. *Microsoft Fabric* berfungsi sebagai *platform cloud* terpusat untuk pengelolaan alur kerja data otomatis (*Pipeline*). *Tool* ini mengambil alih eksekusi *query* yang telah divalidasi di SSMS, memastikan seluruh proses *Extract, Transform, dan Load* (ETL) berjalan sesuai jadwal tanpa campur tangan manual. Fabric mengelola alur data dari sumber hingga pemuatan data akhir ke tabel target DWH. Melalui pemahaman Fabric, *intern* turut memastikan proses pengolahan data berskala besar di lingkungan *cloud* dapat berlangsung secara efisien, terjadwal, dan terstruktur. *Power BI* adalah *tool* utama untuk visualisasi data dan pengembangan *Business Intelligence*. Setelah data berhasil dimodelasi dan tersimpan di DWH, *intern* menggunakan *Power BI* untuk menghubungkan diri ke *database* dan menyajikan *output* dalam bentuk *dashboard* analitik yang interaktif. Peran ini mencakup penyempurnaan *dashboard* yang sudah ada maupun pembuatan *dashboard* baru, termasuk *dashboard monitoring* internal guna menyajikan wawasan bisnis yang relevan dan mudah dipahami oleh *user* dari berbagai departemen. Terakhir, *Microsoft Excel* memiliki peran penting sebagai *tool* pendukung untuk dokumentasi dan manajemen proyek. *Excel* digunakan untuk menyusun Dokumentasi *Mapping* secara rinci. Dokumentasi ini mencakup pelacakan asal-usul, tipe data, dan transformasi yang terjadi pada setiap kolom data. Selain itu, *Excel* dipakai untuk membuat rekapitulasi dan *Progress Tracking* proyek migrasi. *Spreadsheet* ini memuat kalkulasi kuantitatif untuk mengukur persentase kemajuan migrasi per skema, sehingga memberikan update status proyek yang akurat dan terstruktur bagi tim DWH.

Tabel 3.1 *Timeline Realisasi Agenda BI & DWH Modelling Intern*

No.	Aktivitas	Minggu ke-	Tanggal Mulai Aktivitas	Tanggal Akhir Aktivitas	Keterangan
1.	Mendukung proyek migrasi dan pengembangan <i>Data Warehouse</i> perusahaan menggunakan <i>SQL Server Management Studio (SSMS)</i>.	5 Juli – 5 Januari	31 Juli 2025	30 Januari 2026	Mendukung proyek migrasi dan pengembangan <i>Data Warehouse</i> dengan merancang dan menstandarisasi struktur data, sambil mendokumentasikan pemetaan data dan ERD.
1.1	Membantu tim dengan berpartisipasi dalam proyek migrasi sesuai dengan tugas yang diberikan.	5 Juli – 5 Januari	31 Juli 2025	30 Januari 2026	Membantu <i>Migration Project</i> dalam hal membuat <i>View</i> , <i>Table</i> , standarisasi tipe data, dan konversi <i>casting</i> .
1.2	Menyusun serta memperbarui dokumentasi terkait proyek migrasi.	2 Agustus – 5 Januari	12 Agustus 2025	30 Januari 2026	Membuat dan melengkapi <i>Document Mapping</i> pada <i>Microsoft Excel</i> berdasarkan <i>Migration Project</i> yang sudah dikerjakan.
1.3	Membuat <i>Entity Relationship Diagram (ERD)</i> .	2 Oktober – 1 November	10 Oktober 2025	05 November 2025	Membuat dan merevisi ERD berdasarkan <i>Migration Project</i>

No.	Aktivitas	Minggu ke-	Tanggal Mulai Aktivitas	Tanggal Akhir Aktivitas	Keterangan
					yang sudah dikerjakan menggunakan <i>tools draw.io</i> .
2.	Mengembangkan dan mengoptimalkan sistem <i>Business Intelligence</i> berbasis <i>Power BI</i>.	4 Juli – 3 September	28 Juli 2025	16 September 2025	Membuat dan memperbarui tampilan <i>Dashboard</i>.
2.1	Membuat <i>dashboard</i> interaktif sesuai dengan kebutuhan departemen.	2 – 3 September	12 September 2025	16 September 2025	Membuat <i>dashboard</i> yang akan digunakan untuk memantau <i>pipeline</i> menggunakan <i>Power BI</i> .
2.2	Melakukan perbaikan dan penyesuaian pada <i>dashboard</i> yang sudah ada.	4 Juli – 3 Agustus	28 Juli 2025	18 Agustus 2025	Memperbarui tampilan (<i>Make Over</i>) <i>Dashboard</i> yang sudah ada sebelumnya sesuai dengan referensi desain dari <i>Supervisor</i> .
3.	Mendukung kebutuhan section <i>BI & DWH Modelling</i> dengan menggunakan berbagai tools yang tidak terbatas pada <i>SSMS</i> dan <i>Power BI</i>.	4 Juli – 2 September	28 Juli 2025	11 September	Mencatat progress dari proyek migrasi yang sudah dilakukan dan mempelajari <i>pipeline</i> yang digunakan.

No.	Aktivitas	Minggu ke-	Tanggal Mulai Aktivitas	Tanggal Akhir Aktivitas	Keterangan
3.1	Membuat rekapitulasi sistem yang sudah dimigrasi menggunakan <i>Microsoft Excel</i> .	2 September	10 September 2025	11 September 2025	Membuat kalkulasi <i>sheet</i> tentang progres dari <i>Migration Project</i> yang sudah dilakukan dengan menggunakan <i>Microsoft Excel</i> .
3.2	Mempelajari alur kerja pipeline data yang digunakan oleh <i>section</i> .	4 Juli – 3 Agustus	28 Juli 2025	15 Agustus 2025	<i>Sharing knowledge</i> terkait <i>pipeline</i> , proses, dan hal-hal lain dalam <i>tools Microsoft Fabric</i> .

UMN
 UNIVERSITAS
 MULTIMEDIA
 NUSANTARA

3.3 Uraian Pelaksanaan Kerja

Bagian ini merupakan penjelasan secara detail mengenai pekerjaan yang dilakukan. Subbab Uraian Pelaksanaan Kerja dibagi menjadi tiga subsubbab, yaitu Proses Pelaksanaan, Kendala yang Ditemukan, dan Solusi atas Kendala yang Ditemukan. Subsubbab Proses Pelaksanaan menguraikan seluruh pekerjaan yang dilakukan selama program magang berlangsung secara rinci. Lalu, subsubbab Kendala yang Ditemukan membahas terkait seluruh kendala yang terjadi, sedangkan subsubbab Solusi atas Kendala yang Ditemukan menjelaskan solusi yang dilakukan untuk menghadapi hambatan-hambatan tersebut.

3.3.1 Proses Pelaksanaan

Bagian Proses Pelaksanaan ini dibagi ke dalam tiga aktivitas utama. Setiap aktivitas kemudian dibagi lagi ke dalam lingkup aktivitas yang lebih kecil mengikuti Tabel 1.2 agar penjelasan dapat dilakukan secara detail. Beberapa aktivitas juga akan dijelaskan dengan menggunakan metode *Extract, Tranform, and Load* (ETL) karena urutan pengeraannya yang lengkap dan sesuai dengan metode tersebut. Selain itu, pelaksanaan seluruh rangkaian tugas dilakukan dengan pendekatan yang variatif, baik secara mandiri maupun kolaboratif. Aktivitas seperti pembuatan dan penyesuaian *dashboard*, serta pembuatan rekapitulasi dan pendalaman terkait alur kerja *pipeline* dikerjakan secara individual. Sementara itu, proses penggeraan proyek migrasi dilakukan bersama dengan tim ,dimana kontribusi diberikan lebih spesifik pada bagian penyesuaian *query*, pembuatan dokumentasi proyek migrasi, serta pembuatan *Entity Relationship Diagram* (ERD). Penjelasan lebih lanjut akan diuraikan pada subbab berikut.

3.3.1.1. Mendukung proyek migrasi dan pengembangan *Data Warehouse* perusahaan menggunakan *SQL Server Management Studio* (SSMS).

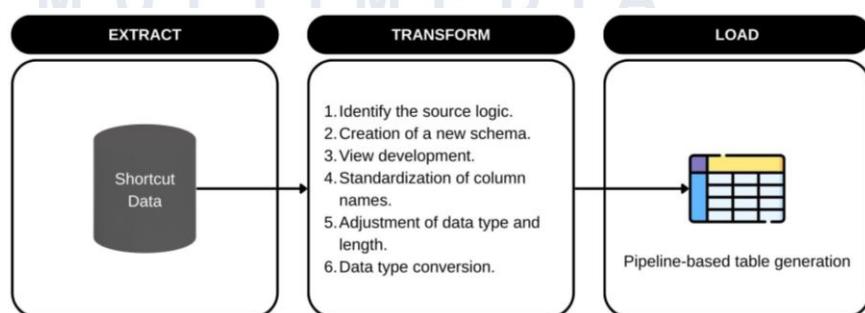
Proyek migrasi dan pengembangan *Data Warehouse* (DWH) perusahaan merupakan kegiatan utama selama magang sebagai BI dan DWH *Modelling Intern*. Proyek penting ini dimulai karena

perusahaan perlu menata ulang dan membuat struktur data mereka menjadi lebih baik dan terpisah. Sebelumnya, proses pembersihan data (*Data Quality* atau DQ) dan proses perancangan DWH masih tercampur menjadi satu, sehingga perlu dipisahkan agar alur kerja data lebih efisien. Tujuan migrasi ini adalah membangun DWH baru yang lebih terstruktur. Alur kerja data dibuat sangat jelas, mulai dari data mentah diambil oleh tim *Data Engineer*, kemudian dibersihkan dan dicek kualitasnya oleh tim *Data Quality*, barulah data tersebut dimodelkan oleh tim *Data Warehouse* agar tim *Business Intelligence* dapat membuat *report* sesuai kebutuhan departemen.

Keterlibatan dalam proyek ini berada pada bagian DWH *Modelling*. Perusahaan memiliki banyak sekali tabel Dimensi (Dim) dan Fakta (Fact) dari berbagai *schema* yang sudah digunakan dalam koneksi DWH yang lama, dan seluruh tabel tersebut harus dipindahkan atau dimigrasi ke dalam koneksi DWH yang baru. Seluruh pekerjaan perancangan struktur DWH baru ini disebut lingkungan *development*, dan prosesnya dilakukan menggunakan aplikasi *SQL Server Management Studio* (SSMS). Prinsip utama dalam *development* ini adalah semua struktur dan skema DWH baru yang dibuat harus mengikuti dan menyesuaikan dengan apa yang sudah ada di *database production*, *database* yang sedang berjalan saat ini. Hal ini memastikan bahwa migrasi tidak membuat struktur yang sama sekali baru, melainkan menata ulang apa yang sudah ada agar lebih rapi dan sesuai standar pemodelan data *modern*. Pekerjaan *development* ini dilakukan secara paralel dengan *database production* yang lama tetap digunakan untuk kegiatan bisnis sehari-hari, sementara DWH baru sedang dibangun. Fokus tugas utama ini adalah memastikan semua tabel Dim dan Fact yang dimigrasi terimplementasi dengan benar, siap digunakan, dan selaras dengan lingkungan *production* tanpa mengganggu operasional yang sedang berjalan.

1. Membantu tim dengan berpartisipasi dalam proyek migrasi sesuai dengan tugas yang diberikan.

Peran sebagai DWH *Modelling Intern* secara langsung berkontribusi pada implementasi detail dari *Migration Project* atau proyek migrasi *Data Warehouse* (DWH). Aktivitas ini mengharuskan pemahaman yang baik mengenai langkah-langkah migrasi alur data dari awal hingga akhir. Pekerjaan ini berfokus pada pembangunan arsitektur data baru di lingkungan *development* sambil memastikan bahwa pemindahan data dari struktur lama di sistem *production* berjalan lancar dan logis. Proses ini diawali dengan menerima pengarahan awal (*Briefing*) yang terperinci dari *Person in Charge* atau PIC, mengenai tabel dan skema spesifik yang menjadi target pemindahan. Setelah target dan lingkup kerja ditentukan, proses migrasi teknis yang dilakukan Tim DWH *Modelling* dibagi menjadi tiga langkah utama yang saling terkait dan berurutan berikut. Selain itu, Gambar 3.2 juga menunjukkan gambar dari tiga alur langkah utama yang lebih jelas. Selama kurang lebih empat bulan, enam skema dengan sekitar 106 tabel Dimensi dan *Fact* berhasil dibantu proses migrasinya. Skema tersebut mencakup berbagai jenis data, seperti data *customer*, transaksi, member, *event*, karyawan, *cluster*, penghuni, properti, dan lainnya. Penjelasan di setiap tahap ETL menggunakan salah satu contoh *query* buatan dari sekian banyak *query* migrasi yang telah dikerjakan. Hal ini diwakilkan dengan tabel Dim_Employee agar familiar di setiap pembacanya.



Gambar 3.2 Alur ETL

a. Extract

Tahap Pengambilan Data atau *Extract* berfungsi untuk memastikan data sumber yang sudah terjamin kualitasnya tersedia di lingkungan kerja Tim DWH. Data yang sudah divalidasi dan dibersihkan oleh Tim *Data Quality* (DQ) tersimpan dalam *Warehouse Development* mereka tersendiri. Seluruh tim data yang terlibat dalam proyek ini memiliki akses ke *Microsoft Fabric Tools*. Tools ini sangat penting karena dapat memfasilitasi mekanisme *shortcut* atau pintasan data yang efisien antar *Workspace Database Development* setiap tim. Secara spesifik dalam lingkungan *development*, *Microsoft Fabric* memungkinkan data yang berasal dari *Warehouse* Tim *Data Quality* (DQ) untuk ditarik dan dijadikan sumber data di *Lakehouse* Tim DWH *Modelling*. Setelah data sumber ini siap, langkah selanjutnya adalah memahami logika *query* dari suatu tabel yang berlaku di *production*. Sebagai contoh, salah satu tabel *dummy* yang ingin dimigrasi adalah tabel “az.Dim_Employee” dari skema “az”. Pada lingkungan *production*, *query* untuk membuat tabel ini disimpan dalam sebuah *Stored Procedure* bernama “dbo.sp_az_DataAZ”.

```
SELECT  
    | EmployeeId  
    , EmployeeName  
    , EmployeeRoles  
    , HiredDate  
    , LastCutOffData = @LastCutOffData  
  
FROM [dbo].[TBL_AZ_Employee];
```

Gambar 3.3 Dummy *Stored Procedure* Dim_Employee

Berdasarkan Gambar 3.3 yang menunjukkan potongan *query* dari *Stored Procedure* tersebut, sumber tabel yang digunakan

untuk membuat tabel Dim_Employee pada lingkungan *production* adalah “dbo.TBL_AZ_Employee”. Oleh karena itu, tabel sumber inilah yang akan ditarik ke dalam *lakehouse* lingkungan *development* Tim DWH dari *warehouse* Tim DQ menggunakan *Microsoft Fabric*, spesifiknya fitur *Shortcut*. Langkah pertama dimulai dengan mengakses *Workspace* Tim DWH dan memilih *Lakehouse* tujuan yang akan digunakan untuk menampung data lingkungan *development*. Mahasiswa kemudian memilih opsi *New Shortcut* pada folder tabel untuk menginisiasi pembuatan koneksi langsung ke sumber data tanpa melakukan penggandaan data secara fisik. Melalui pemilihan sumber internal pada *Microsoft OneLake*, mahasiswa dapat mencari dan memilih tabel “dbo.TBL_AZ_Employee” dari *Warehouse* Tim DQ. Setelah konfigurasi diselesaikan, tabel tersebut akan muncul secara otomatis di dalam *Lakehouse* tujuan sebagai objek referensi yang siap digunakan untuk proses transformasi data lebih lanjut dalam proyek migrasi ini.

b. *Transform*

Logika *query* yang diekstrak dari *Stored Procedure* (SP) Gambar 3.3 di lingkungan *production* ini, kemudian digunakan sebagai dasar untuk proses transformasi data. Pekerjaan dimulai dengan membuat skema baru untuk mengelompokkan tabel secara terstruktur, dalam hal ini skema tersebut ialah “az”. Langkah selanjutnya adalah melakukan penyesuaian *query* lama yang ada di dalam SP ke dalam *View* baru, misalnya “az.view_Dim_Employee” di *Database Development* Tim DWH. Penyesuaian ini berarti *query* harus dimodifikasi mengikuti standar nama kolom, tipe data, dan panjang data yang sudah ditetapkan dalam *Lakehouse* yang merupakan tarikan data dari *Warehouse Development* Tim DQ. Pada *lakehouse*, tabel “dbo.TBL_AZ_Employee” berubah nama menjadi

“dbo.AZ_Employee” dan memiliki kolom *EmployeeID* bertipe data *bigint*, *EmployeeName* bertipe *varchar(8000)*, *EmployeeRole* bertipe *varchar(1000)*, dan *HireDate* bertipe *datetime2(6)*. Penyesuaian dalam *query* dilakukan menggunakan fungsi *CAST* untuk mengonversi tipe data setiap kolom. Kolom *LastCutOffData* yang ada pada Gambar 3.3 tidak diimplementasikan pada pembuatan *view* ini karena kolom tersebut akan dibuat secara otomatis melalui *pipeline* di *Microsoft Fabric* setelah proses *Transform* selesai. Selain itu, ditambahkan juga kolom *key* sebagai kunci utama untuk setiap tabel Dimensi. Tabel Fakta tidak ditambahkan kolom *key* karena kebutuhannya berbeda dengan tabel jenis Dimensi. Maka, *query view* yang sudah dibuat akan seperti pada Gambar 3.4.

```
-- Membuat schema az
CREATE SCHEMA az

-- Membuat view view_Dim_Employee
CREATE VIEW az.view_Dim_Employee AS
SELECT
    | EmployeeKey    = CAST(NULL AS BIGINT)
    , EmployeeID    = CAST(EmployeeID AS BIGINT)
    , EmployeeName   = CAST(EmployeeName AS VARCHAR(8000))
    , EmployeeRole   = CAST(EmployeeRole AS VARCHAR(1000))
    , HireDate       = CAST(HireDate AS DATETIME2(6))

FROM [lakehouse].[dbo].[AZ_Employee];
```

Gambar 3.4 *Query Create View Dim_Employee*

c. Load

Setelah proses pembuatan *view* “az.view_Dim_Employee” selesai, langkah berikutnya adalah mengisi data tersebut ke dalam tabel “az.Dim_Employee” di *Database Development Warehouse*. Proses pengisian ini dilakukan menggunakan *pipeline* di *Microsoft Fabric* yang bekerja secara otomatis. Secara teknis, *pipeline* ini dijalankan dengan cara menghubungkan data asal, yaitu *view* ke dalam tabel tujuan,

kemudian menekan tombol *Run* untuk memulai prosesnya. Saat pipeline berjalan, sistem tidak hanya memindahkan data, tetapi juga secara otomatis menambahkan kolom baru bernama *LastCutOffData*. Selain itu, *pipeline* ini berfungsi untuk memastikan bahwa semua data yang masuk sudah benar, misalnya dengan menyesuaikan tipe data agar tidak terjadi eror. Pemantauan jalannya proses ini dilakukan melalui monitoring secara manual guna memastikan seluruh data berpindah secara sempurna tanpa ada hambatan teknis. Tabel akhir yang terbentuk pun telah berada dalam kondisi rapi serta akurat, sehingga siap digunakan untuk memenuhi kebutuhan data tim secara optimal.

Tabel-tabel yang telah dibuat selanjutnya akan dilakukan validasi kualitas datanya, dengan cara membandingkan hasil tabel tersebut di lingkungan *Development* dan *Production*. Proses validasi akan dibagi ke dalam dua bagian, yaitu pengecekan jumlah data (*Row Count*) dan pengecekan nilai data (*Value Check*). Kebenaran dari suatu pengecekan jumlah data ditandai dengan jumlah data yang lebih sedikit atau sama di lingkungan *Development* daripada *Production*. Kondisi ini normal terjadi karena titik *cut off data* tertentu yang digunakan untuk *Development*, sementara data *Production* masih terus bertambah secara *real time* dari aktivitas bisnis yang sedang berjalan. Namun, apabila jumlah data di *Development* lebih banyak, tidak menutup kemungkinan bahwa jumlah data memang sudah benar. Anomali ini perlu dicek lebih lanjut ke Tim DQ untuk memastikan kebenarannya. Selain itu, *Value Check* dilakukan secara *random* pada beberapa *record* spesifik, biasanya menggunakan kolom ID karena sifatnya yang unik. Kemudian, hasilnya akan dibandingkan *value* dari ID tersebut yang ada di lingkungan *Development* sama dengan *Production*. Hal tersebut dilakukan untuk memastikan bahwa setiap kolom menghasilkan nilai yang konsisten dan akurat dengan sumber aslinya.

2. Menyusun serta memperbarui dokumentasi terkait proyek migrasi.

Selain melaksanakan tugas implementasi teknis dalam proses data, tanggung jawab penting lainnya dalam proyek migrasi *Data Warehouse* adalah menyusun dan memperbarui dokumentasi data secara menyeluruh. Dokumentasi ini berfungsi sebagai acuan utama yang menjamin keseragaman dan kemudahan pelacakan informasi bagi tim yang terlibat. Proses dokumentasi dilakukan menggunakan aplikasi *Microsoft Excel* dan diorganisir berdasarkan struktur skema *database* DWH. Setiap file *Excel* didedikasikan untuk satu skema tertentu, misalnya, semua tabel yang termasuk dalam skema “az” akan memiliki satu file dokumentasi. Lebih lanjut, di dalam file tersebut, setiap tabel yang dimigrasi seperti tabel “Dim_Employee” akan memiliki lembar kerja atau *sheet*-nya tersendiri. Isi utama dari dokumentasi ini adalah pencatatan rinci mengenai asal-usul data dan perubahan yang terjadi pada setiap kolom. Rincian yang dicatat mencakup metadata penting dari sumber hingga target. Dokumentasi ini merinci kolom tersebut diambil dari Tabel Sumber (*TableName*) mana di Lakehouse dan bagaimana Nama Kolom (*FieldName*) aslinya. Kolom ini juga berfungsi mencatat Nama Kolom (*FieldName*) dan Tabel Tujuan (*TableName*) yang digunakan di *Warehouse* DWH baru, serta membandingkan apakah nama kolom tersebut sama atau berbeda dengan yang ada di *Lakehouse*. Data yang dicatat juga meliputi Tipe Data (*DataType*) dan Panjang (*Length*) yang diimplementasikan pada tabel *Warehouse* target. Lebih lanjut, terdapat kolom *Usage* yang berfungsi sebagai indikator yang menunjukkan apakah kolom tersebut digunakan atau tidak. Apabila di dalam *query* terdapat *query* terkait join dengan tabel Dim, maka pencatatan akan dilakukan pada kolom *Dimension*. Sedangkan, kolom terakhir yaitu *Note* digunakan untuk mencatat *query* yang unik dari suatu kolom. Seluruh detail pencatatan ini memastikan bahwa setiap

penyesuaian yang dilakukan selama proses migrasi dapat dilacak dan dipahami dengan mudah oleh tim DWH maupun yang lainnya. Contoh struktur dan format detail dokumentasi ini dapat dilihat pada Gambar 3.5.

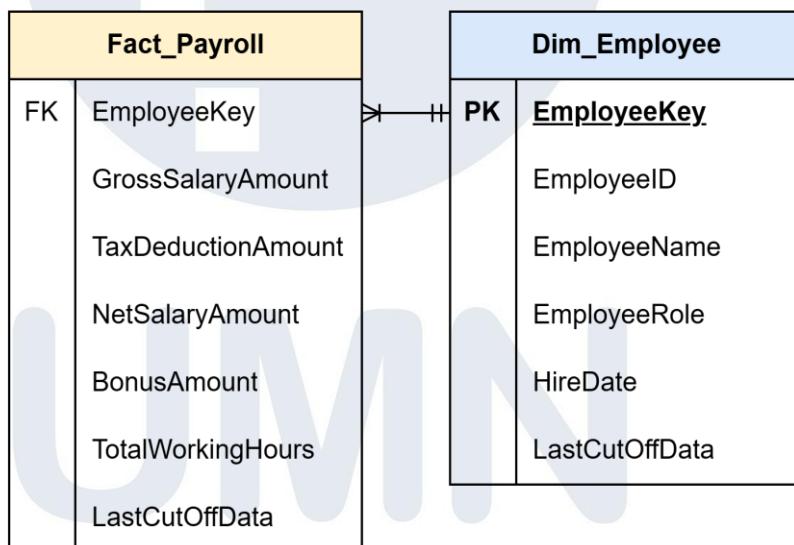
Table Name : Dim_Employee																
No	PK / FK	ColumnName	ColumnDesc	Usage	TableName	FieldName	Source DB : lakehouse Schema : dbo			Target DB : warehouse Schema : az			Dimension			Note
							TableName	FieldName	DataType	Length	TableName	FieldName	KeyRelation			
1	PK	EmployeeKey	*auto generate key*	<input checked="" type="checkbox"/>			Dim_Employee	EmployeeKey	bigint							
2		EmployeeID		<input checked="" type="checkbox"/>	AZ_Employee	EmployeeID	Dim_Employee	EmployeeID	bigint							
3		EmployeeName		<input checked="" type="checkbox"/>	AZ_Employee	EmployeeName	Dim_Employee	EmployeeName	varchar	8000						
4		EmployeeRole		<input checked="" type="checkbox"/>	AZ_Employee	EmployeeRole	Dim_Employee	EmployeeRole	varchar	1000						
5		HireDate		<input checked="" type="checkbox"/>	AZ_Employee	HireDate	Dim_Employee	HireDate	datetime2	6						
6		LastCutOffData	cut off date data	<input checked="" type="checkbox"/>			Dim_Employee	LastCutOffData	datetime2	6						

Gambar 3.5 Documentation Mapping dari az.Dim_Employee

3. Membuat *Entity Relationship Diagram* (ERD).

Setelah proses dokumentasi *mapping* dan penyesuaian data dalam proyek migrasi DWH selesai, langkah penting berikutnya adalah visualisasi model data melalui *Entity Relationship Diagram* (ERD). ERD berfungsi sebagai peta yang menjelaskan secara visual bagaimana tabel-tabel data saling terhubung dan berinteraksi. Pembuatan ERD ini dilakukan menggunakan *tools* visual *draw.io*. Berbeda dengan ERD tradisional yang menggambarkan seluruh *database*, dalam konteks DWH *Modelling*, fokus pembuatan ERD terdapat pada setiap tabel fakta secara individual. Artinya, jika dalam satu skema terdapat lima tabel *Fact*, maka akan dibuat lima diagram ERD yang terpisah, di mana masing-masing diagram menunjukkan relasi tabel fakta tersebut dengan dimensi-dimensi yang relevan. Sebagai contoh implementasi, terdapat sebuah table *Fact* dalam skema “az”, yaitu “Fact_Payroll”. Tabel ini berfungsi untuk mencatat dan mengukur seluruh aktivitas terkait pembayaran gaji karyawan. Keterkaitan “Fact_Payroll” dengan tabel “Dim_Employee” bersifat wajib karena tabel *Fact* ini menggunakan kolom *EmployeeKey* sebagai *Foreign Key* yang merujuk pada *Primary Key* di “Dim_Employee”. Oleh karena itu, *query* untuk tabel *Fact*

membutuhkan koneksi dengan tabel dimensi tersebut untuk dapat mengambil konteks detail dari setiap karyawan. Hubungan yang diperlukan dalam *query* ini harus direpresentasikan dalam Diagram ERD sebagai relasi *one-to-many* antara keduanya. Tabel “Fact_Payroll” dirancang untuk menyimpan berbagai kolom yang saling berkaitan, meliputi *EmployeeKey*, *GrossSalaryAmount*, *TaxDeductionAmount*, *NetSalaryAmount*, *BonusAmount*, dan *TotalWorkingHours*. Gambar 3.6 menyajikan tampilan ERD yang dihasilkan secara visual, di mana terlihat bahwa setiap baris data gaji di “Fact_Payroll” diidentifikasi dan dihubungkan secara unik melalui *EmployeeKey*, memungkinkan analisis untuk mengetahui rincian gaji, potongan, dan bonus untuk setiap karyawan secara rinci.



Gambar 3.6 Contoh ERD dari *Fact_Payroll*

3.3.1.2. Mengembangkan dan mengoptimalkan sistem *Business Intelligence* berbasis Power BI.

Tahap akhir dari pemodelan *Data Warehouse* (DWH) adalah penyajian data yang siap untuk analisis, yang diwujudkan melalui pengembangan sistem *Business Intelligence* (BI). Meskipun Tim BI memiliki unit khusus yang bertanggung jawab penuh atas laporan, fokus tugas ini ada pada dua aspek penting. Pertama, pembuatan

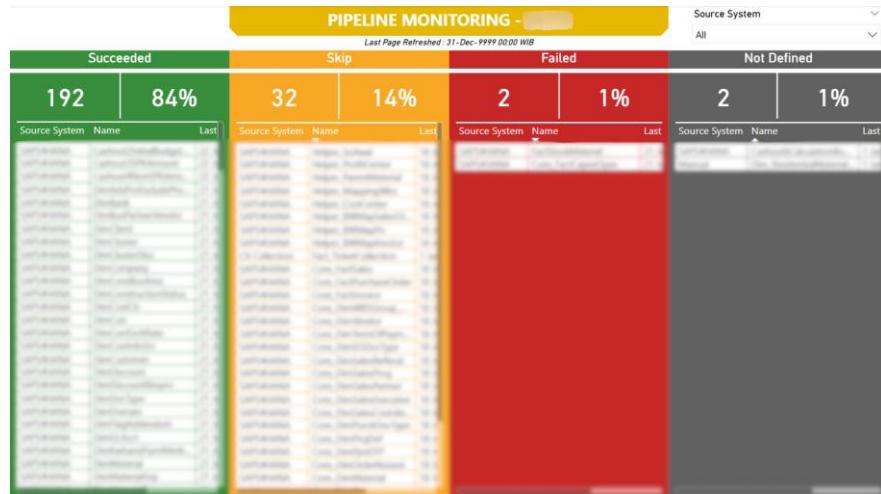
dashboard baru yang ditujukan untuk keperluan internal Tim DWH sendiri. Kedua, melakukan *make-up dashboard*, yaitu proses estetika dan penyempurnaan tampilan pada *dashboard* yang sudah ada. Penataan visual (*make-up*) pada *dashboard* sangatlah penting. Tujuannya adalah untuk memastikan *dashboard* tidak hanya berfungsi secara fungsional, tetapi juga mengikuti standar visual yang telah ditetapkan. Hal ini menjamin hasil akhir berupa laporan dan visualisasi data yang terstandar, informatif, dan secara optimal mendukung pengambilan keputusan.

1. Membuat *dashboard* interaktif sesuai dengan kebutuhan departemen.

Keterlibatan peserta magang tidak hanya mencakup proyek migrasi. Namun, juga meluas pada pengembangan sistem *Business Intelligence* (BI) yang berfokus pada alat *monitoring* esensial untuk keperluan internal Tim DWH. Tugas ini berupa pembuatan *dashboard* interaktif menggunakan *Power BI* yang berfungsi memvisualisasikan dan melacak status pelaksanaan proses data atau *pipeline* pada setiap tabel DWH. *Dashboard* ini secara spesifik memonitor status yang berhasil, gagal, terlewat (*skip*), atau tidak tedefinisi (*not defined*) dari setiap eksekusi *pipeline*. *Dashboard monitoring* ini disusun dalam tiga halaman yang saling terhubung. Halaman pertama merupakan halaman ringkasan yang menyajikan tampilan sekilas performa *pipeline* secara keseluruhan. Halaman pertama dibuat dengan menyesuaikan referensi yang diberikan oleh PIC. Halaman kedua merupakan halaman detail dari tabel-tabel yang sudah ditampilkan dalam halaman pertama. Tidak hanya itu, halaman kedua juga menyediakan informasi rinci mengenai status dan waktu eksekusi tabel di lingkungan DWH. Sedangkan, halaman ketiga berfungsi sebagai validasi silang untuk mengecek status tabel sumber di Tim DQ dari sisi *lakehouse* Tim DWH.

Proses diawali dengan tahap persiapan data yang krusial menggunakan sumber data metadata dan operasional. Data yang ditarik untuk *dashboard monitoring* ini bukanlah data bisnis, melainkan data *log* dan rekaman status eksekusi *pipeline* dari *database* perusahaan. Penarikan data secara spesifik dilakukan pada tabel-tabel internal yang menyimpan informasi mengenai waktu mulai, waktu berakhir, dan status akhir dari setiap proses *pipeline* yang dijalankan. Status *pipeline* menjadi kolom yang paling penting karena tujuan pembuatan *dashboard* yang ingin menyajikan hasil dari status tersebut. Data ini ditarik baik dari lingkungan DWH yang langsung dihubungkan dengan *tools Power BI*. Sumber data ini harus ditarik secara lengkap untuk memastikan *dashboard* memiliki visibilitas penuh terhadap seluruh data yang dibutuhkan. Data yang sudah dipersiapkan sebelumnya, kemudian akan langsung digunakan ke dalam proses visualisasi dan desain *dashboard*. Halaman pertama ditunjukkan oleh Gambar 3.7. Pada halaman pertama ini, informasi mengenai *pipeline* tabel mana saja yang sudah berjalan dengan berhasil, gagal, terlewat, ataupun tidak terdefinisi dapat diketahui dengan segera. Selain status tersebut, halaman ini juga dilengkapi dengan informasi ID dan nama tabel setiap sistem yang dimiliki oleh perusahaan. Visualisasi pada *dashboard* bertujuan untuk memudahkan pengguna mengetahui total dan persentase dari setiap tabel di masing-masing status. Oleh karena itu, setiap status dipisahkan dengan warna yang kontras. Warna hijau mewakili status *pipeline* yang berhasil dijalankan, warna kuning mewakili status *pipeline* yang terlewat, warna merah mewakili status *pipeline* yang gagal dijalankan, sedangkan warna abu-abu mewakili status *pipeline* yang tidak berjalan sebagaimana mestinya dan memerlukan penelusuran lebih lanjut. *Last Page Refreshed* memberikan informasi kepada pengguna kapan *pipeline monitoring* tersebut di *refresh*. Lebih lanjut, halaman ini juga dilengkapi dengan filter yang memungkinkan

pengguna untuk menganalisis status *pipeline* secara spesifik berdasarkan setiap sistem yang terlibat.

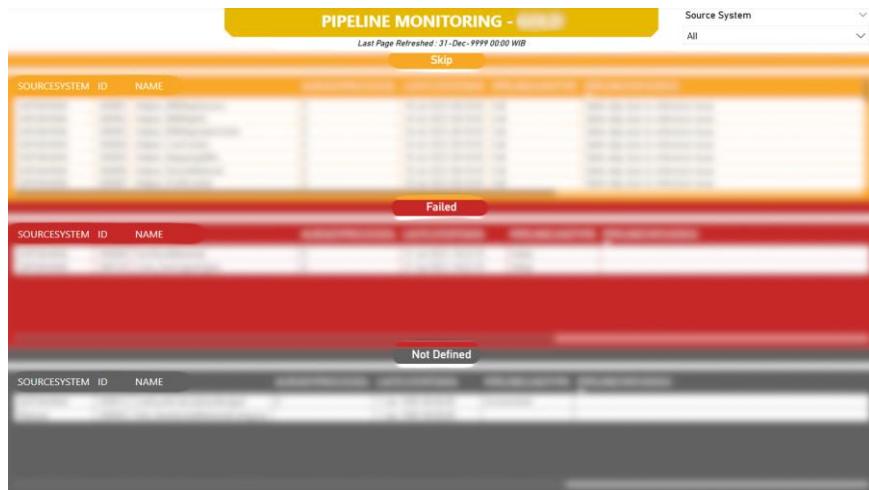


Gambar 3.7 Halaman Pertama *Dashboard*

Gambar 3.8 merupakan hasil tangkapan layar dari *dashboard* halaman yang kedua. Halaman kedua ini seperti ekspansi dengan halaman pertama, dimana halaman kedua menyajikan informasi yang lebih spesifik untuk setiap statusnya. Pada halaman ini, tabel dengan status *pipeline* berhasil akan diabaikan dan dianggap selesai. Sedangkan, tabel dengan status *pipeline skip*, gagal, dan *not defined* akan ditelusuri lebih lanjut melalui kolom-kolom yang mendefinisikan kapan tabel *pipeline* dari tabel tersebut terakhir berjalan dan berhenti, apakah status dari tabel tersebut aktif untuk digunakan atau *deactivate*, dan lain sebagainya. Fungsi filter dan *Last Page Refreshed* di halaman kedua ini memiliki fungsi yang sama dengan halaman pertama.

Berbeda dengan halaman pertama dan kedua yang masih saling berkaitan, halaman ketiga yang ditunjukkan oleh Gambar 3.9, dirancang untuk menyajikan informasi status tabel yang berada di *warehouse* Tim DQ. Halaman ini sangat penting karena proses *pipeline* di tim DWH hanya akan berjalan jika tabel sumber di tim sebelumnya juga sudah berhasil *loading*. Fungsi utama dari halaman ini adalah sebagai validasi silang atau *cross-validation* dan alat

penelusuran masalah untuk tabel-tabel yang berstatus selain *success*. Maksudnya, apabila di halaman pertama dan kedua banyak ditemukan tabel yang berstatus *failed*, *skip*, atau *not defined*, maka *dashboard* ini akan memudahkan tim DWH untuk segera menelusuri apakah kegagalan tersebut disebabkan oleh masalah pada sumber datanya.



Gambar 3.8 Halaman Kedua *Dashboard*

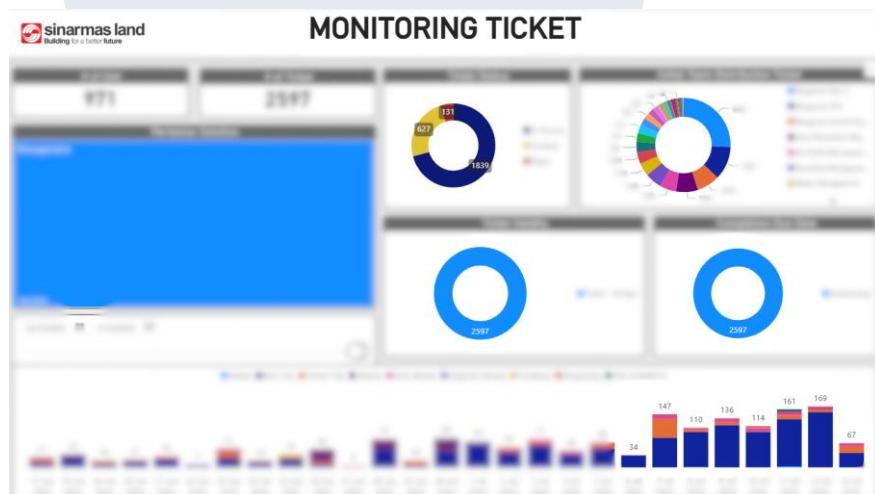


Gambar 3.9 Halaman Ketiga *Dashboard*

2. Melakukan perbaikan dan penyesuaian pada *dashboard* yang sudah ada.

Selain pengembangan *Dashboard Monitoring Pipeline*, tanggung jawab lain yang dijalankan adalah melakukan perbaikan dan penyesuaian visual (*Make-up*) pada salah satu *dashboard Business*

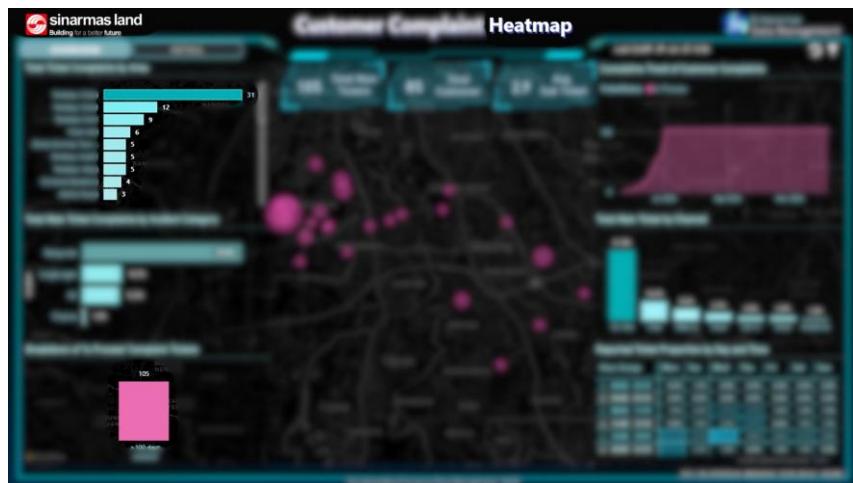
Intelligence (BI) yang sudah ada. Tugas ini bukan sekadar mengubah estetika, tetapi merupakan bagian integral dari strategi standardisasi visual perusahaan. Sebelum penyesuaian dilakukan, *dashboard* yang ada seringkali memiliki ketidakstabilan dalam penggunaan warna, ukuran *font*, dan *icon*, yang berpotensi membingungkan pengguna akhir saat beralih dari satu laporan ke laporan lainnya. Oleh karena itu, tugas utama di sini adalah memastikan *dashboard* yang ditangani memiliki tema visual yang seragam dan menjadikannya lebih profesional, serta mudah dicerna oleh seluruh pemangku kepentingan. Pekerjaan penyesuaian ini menuntut ketelitian tinggi terhadap detail desain dan *user experience (UX)* agar fungsionalitas laporan tetap optimal.



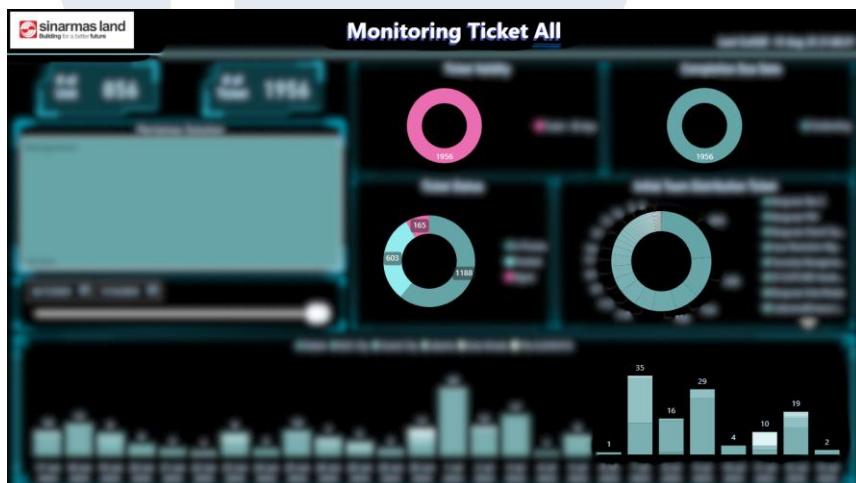
Gambar 3.10 *Dashboard* sebelum penyesuaian visual

Proses penyesuaian visual ini dilakukan dengan mengacu pada referensi visual yang sudah ditetapkan tim BI. Secara spesifik, penyesuaian ini meliputi perancangan ulang penggunaan skema warna, pemilihan tipe dan ukuran *font* yang konsisten, penataan ulang tata letak (*layout*) visual, hingga penggantian ikon-ikon agar selaras dengan tema dari referensi yang diberikan. Misalnya, semua grafik utama harus menggunakan palet warna yang sama, dan semua *Key Performance Indicator (KPI)* harus menggunakan *icon* dengan gaya yang identik. Hasil dari kegiatan *make-up* ini sangat penting. Hal ini

memastikan bahwa *dashboard* tidak hanya fungsional secara data, tetapi juga mengikuti standar visual yang ada.



Gambar 3.11 Visual *dashboard* yang menjadi referensi



Gambar 3.12 Hasil *dashboard* setelah penyesuaian dengan referensi

Gambar 3.10 merupakan hasil tangkapan layar dari *dashboard* yang akan di *make-up*. Sebelum proses penyesuaian visual, *dashboard* masih memiliki warna-warna *chart* yang beragam, seperti *pie chart* warna-warni, *full* biru, dan *bar chart* biru oren. *Dashboard* tersebut juga masih menggunakan tema yang terang dengan *full background* berwarna putih dan *font* hitam. Visualisasi *dashboard* Gambar 3.10 sangat berbanding terbalik dengan *dashboard* referensi yang ditunjukkan oleh Gambar 3.11. *Dashboard* Gambar 3.11

menggunakan tiga komponen warna utama, yaitu pink, biru kehijauan atau *teal*, dan biru. Tema yang digunakan juga cenderung gelap dengan *full background* hitam dan warna *font* putih. Setelah dilakukan *make-up dashboard*, Gambar 3.12 menunjukkan hasil *dashboard* Gambar 3.10 yang sudah disesuaikan dengan tema dan warna dari Gambar 3.11. Proses pun selesai dan *dashboard* dapat digunakan langsung oleh tim BI dan DWH.

3.3.1.3. Mendukung kebutuhan section BI & DWH *Modelling* dengan menggunakan berbagai tools yang tidak terbatas pada SSMS dan Power BI.

Dukungan terhadap section *BI* dan DWH *Modelling* tidak terbatas pada pemodelan skema data, tetapi juga mencakup aktivitas krusial yang berhubungan dengan manajemen proyek, monitoring infrastruktur, dan pelacakan kemajuan dari proyek yang sedang dilakukan. Tugas-tugas ini memerlukan penggunaan beragam tools di luar SQL Server Management Studio (SSMS) dan Power BI. Secara umum, pekerjaan difokuskan pada dua area utama, yaitu Manajemen Progres Migrasi dan Pemahaman Infrastruktur Data. Pada hal manajemen, dilakukan rekapitulasi dan pemantauan kemajuan (*Progress Tracking*) proyek migrasi *Data Warehouse* secara keseluruhan menggunakan Microsoft Excel, di mana Excel berfungsi sebagai *checklist* dinamis dan alat kalkulasi metrik persentase kemajuan migrasi per skema. Sementara itu, dalam aspek infrastruktur, dilakukan pembelajaran mendalam terhadap alur kerja *pipeline* data tim menggunakan Microsoft Fabric. Pemahaman pipeline ini memastikan data DWH ditransformasi secara akurat dan otomatis, serta sekaligus menjadi dasar untuk monitoring status eksekusi *pipeline*. Aktivitas pemahaman dan monitoring infrastruktur di Microsoft Fabric ini sangat penting karena *Data Warehouse* bertindak sebagai sumber data yang mengonsolidasikan data dari berbagai sistem operasional, yang kemudian disajikan dalam laporan

real-time di Power BI untuk mendukung analisis bisnis dan pengambilan keputusan strategis.

1. Membuat rekapitulasi sistem yang sudah dimigrasi menggunakan Microsoft Excel.

Salah satu tugas penting lainnya yang dilakukan untuk mendukung manajemen proyek adalah membuat rekapitulasi dan pemantauan perkembangan atau *Progress Tracking* dari proyek migrasi *Data Warehouse* secara keseluruhan menggunakan *Microsoft Excel*. Tugas ini bertujuan memberikan arah yang jelas dan terukur mengenai status migrasi setiap skema data, serta memastikan seluruh tim dapat memantau capaian yang sudah diraih. Langkah awalnya adalah melakukan pencatatan mendetail terhadap semua tabel yang terlibat. Setiap skema memerlukan pencatatan daftar lengkap namanya tabel sumber yang harus dimigrasi, mencakup semua tabel *Fact* maupun Dimensi. Seiring berjalannya proyek, *spreadsheet Excel* ini kemudian berfungsi sebagai *checklist* dinamis, di mana dilakukan pembaruan status secara berkala untuk mencatat tabel mana saja yang telah berhasil dimigrasi dan diimplementasikan ke dalam struktur DWH yang baru. Lebih dari sekadar *checklist*, *spreadsheet* ini juga dikembangkan untuk melakukan kalkulasi metrik kemajuan proyek. Tugas rekapitulasi ini melibatkan perhitungan status secara kuantitatif. Contohnya, untuk setiap skema, dihitung jumlah tabel asli yang harus dipindahkan, jumlah tabel yang sudah berhasil dimigrasi, dan kemudian dikonversi menjadi persentase kemajuan untuk skema tersebut. Selain itu, dibuat pula perhitungan total keseluruhan yang menunjukkan total persentase kemajuan keseluruhan proyek migrasi dari semua skema yang terlibat. Penggunaan *Microsoft Excel* di sini sangat efektif karena kemampuannya dalam memvisualisasikan data dan melakukan perhitungan cepat, yang pada akhirnya mampu menyediakan *update* rekapitulasi yang akurat dan terbaru. Tabel 3.2 merupakan contoh tabel *dummy* yang dibuat untuk memperjelas tugas

dari subsubbab ini. Tabel tersebut terdiri dari kolom Nama *Schema*, Nama Tabel, dan Jenis Tabel. Seluruh nama *schema* dan tabel merupakan nama buatan atau *dummy*. Sedangkan, Tabel 3.3 merupakan tabel perhitungan dari rekapitulasi kemajuan migrasi Tabel 3.2. Tabel perhitungan ini menunjukkan bagaimana dan berapa banyak progress tabel yang sudah dimigrasi untuk setiap skemanya.

Tabel 3.2 Contoh Tabel Rekapitulasi

Nama Schema	Nama Tabel	Jenis Tabel
az	Dim_TerminationReason	Dim
	Dim_Location	
	Dim_JobTitle	
	Dim_Employee	
	Dim_Department	
	Fact_TimeSheet	
	Fact_Payroll	
	Fact_HRRecruitment	
	Fact_EmployeeTraining	
	Fact_Attendance	
by	Dim_SalesChannel	Dim
	Dim_Product	
	Dim_Customer	
	Dim_Time	
	Fact_SalesTarget	Fact
	Fact_SalesOrder	

Tabel 3.3 Perhitungan Rekapitulasi Progres Migrasi

No	Nama Schema	Jumlah Tabel		Persentase
		Done	All	
1.	az	8	10	80%
2.	by	4	6	67%

No	Nama Schema	Jumlah Tabel		Percentase
		Done	All	
Total		12	16	75%

2. Mempelajari alur kerja pipeline data yang digunakan oleh *section*.

Pemahaman mendalam terhadap alur kerja data *pipeline* adalah keahlian mendasar dalam peran DWH *Modelling*. *Pipeline* yang dikembangkan oleh tim berfungsi sebagai jembatan krusial yang menghubungkan data sumber yang telah bersih dengan model *Data Warehouse* yang siap untuk dianalisis. Secara keseluruhan, alur kerja *pipeline* ini sepenuhnya diimplementasikan dan diatur dalam *Microsoft Fabric*. *Fabric* berperan sebagai platform terpusat yang menjalankan semua proses otomatis. *Pipeline DWH Modelling* memiliki fokus yang sangat spesifik, yakni mengambil data yang sudah divalidasi dan bersih dari sistem *upstream* atau *warehouse* Tim DQ dan menerapkan logika pemodelan dimensional ataupun *fact*. Alur kerja DWH ini dimulai secara otomatis segera, setelah data sumber di *lakehouse* tim DWH siap. Hal ini dikarenakan data yang ada di dalam *lakehouse* tim DWH merupakan data yang diambil dari *warehouse* tim DQ. Melalui *pipeline* ini, serangkaian langkah *query* dijalankan yang mengambil data dari *View* yang sudah terdefinisi, misalnya “az.view_Dim_Employee” untuk menciptakan tabel permanen dimensi atau fakta di *Data Warehouse development*, dalam hal ini menjadi “az.Dim_Employee”. Selain itu, kolom metadata seperti *LastCutOffData* juga ditambahkan secara otomatis selama eksekusi *pipeline*. Proses ini mencakup penyesuaian penamaan kolom, konversi tipe data, penggunaan fungsi *CAST*, dan penambahan *Key dimensi* yang semuanya telah didefinisikan dalam *View* tersebut. Setelah logika transformasi berhasil dieksekusi, *pipeline* kemudian melanjutkan dengan memuat data yang sudah termodelasi ke tabel

target. Selain pemrosesan data inti, *pipeline* DWH juga terintegrasi dengan mekanisme *error handling* dan *auditing internal Fabric* untuk secara otomatis mencatat status eksekusi yang berhasil, gagal, terlewat, atau tidak terdefinisi, dan waktu pemrosesan, yang memastikan integritas data terjamin sehingga setiap anomali dapat segera dideteksi oleh tim.

3.3.2 Kendala yang Ditemukan

Kendala yang ditemui selama pelaksanaan proyek merupakan salah satu bagian dari proses pembelajaran dan pengembangan diri di lingkungan profesional. Tantangan-tantangan ini memberikan pemahaman yang lebih dalam mengenai kompleksitas pengolahan dan pemodelan data di lingkungan *Data Warehouse* (DWH) yang nyata. Selama pelaksanaan tugas sebagai BI & DWH *Modelling Intern*, terdapat dua kendala utama yang ditemukan pada fase awal dan implementasi teknis, yang secara signifikan memengaruhi kecepatan adaptasi dan efisiensi kerja tim, yaitu tantangan terkait adaptasi teknologi baru dan isu administratif terkait akses kerja. Berikut penjelasan secara rinci terkait dua kendala tersebut.

1. Kurangnya familiaritas terhadap *Microsoft Fabric*

Kendala teknis pertama yang dihadapi adalah kebutuhan mendesak untuk beradaptasi dengan lingkungan kerja berbasis *cloud* yang sepenuhnya baru. Latar belakang akademis sebelumnya membekali dengan dasar-dasar *Extract, Transform, and Load* (ETL) menggunakan *tools* seperti *Pentaho*. Namun, proyek migrasi DWH mengharuskan transisi penuh menggunakan platform *Microsoft Fabric*.

Perpindahan dari *tools* ETL tradisional ke *Fabric* yang berbasis *cloud* menimbulkan sedikit tantangan di fase awal pembelajaran karena *Fabric* digunakan untuk seluruh alur *pipeline* dan mekanisme *shortcut* data, yang seluruhnya berbeda dengan konsep lokal. Meskipun tugas implementasi logika inti seperti membuat *View*

harmonisasi seperti misalnya az.view_Dim_Employee dilakukan menggunakan SSMS, proses eksekusi akhir, yaitu pembuatan tabel permanen dari *View* dan menjalankan *pipeline* secara keseluruhan, seluruhnya dikendalikan oleh *Fabric*. Oleh karena itu, waktu yang diperlukan untuk menyiapkan dan menguji *pipeline* awal menjadi lebih lama. Hal ini disebabkan harus melalui tahap *troubleshooting* integrasi yang intensif antara *View* yang telah divalidasi di SSMS dengan *pipeline* yang bertanggung jawab mengeksekusi dan memuat data akhir di *Fabric*. Proses penyiapan integrasi awal inilah yang menuntut waktu adaptasi dan *debugging*, meskipun setelah *pipeline* berhasil berjalan, proses *Load* data untuk banyak tabel akan berlangsung sangat cepat. Pemahaman mendalam diperlukan tidak hanya pada logika *query SQL* yang dibuat di SSMS, tetapi juga pada aspek alur *pipeline* yang dikelola langsung di *Fabric*. Meskipun tantangan ini teratasi seiring berjalannya waktu, namun pada fase awal proyek, efisiensi kerja sempat sedikit tertunda karena proses adaptasi teknologi yang mendalam.

2. Keterlambatan email

Kendala kedua yang bersifat administratif dan sangat umum di lingkungan perusahaan besar adalah keterlambatan dalam memperoleh akses dan *privilege* yang dibutuhkan untuk bekerja. Pekerjaan BI & DWH Modelling sangat bergantung pada akses ke berbagai *resource* data dan kolaborasi. Pada fase awal, proses pembuatan akun email perusahaan mengalami keterlambatan. Ketiadaan email resmi menghambat kemampuan untuk menerima undangan ke *Microsoft Teams*, yang merupakan saluran komunikasi dan kolaborasi utama tim, sehingga menyulitkan koordinasi harian dan berbagi update data penting.

Namun, hambatan terbesar muncul dari keterlambatan perolehan hak akses (*privilege*) ke *Workspace* dan *Database*. Memulai tugas

inti migrasi, seperti membuat *View* harmonisasi atau menjalankan *query* validasi, diperlukan berbagai macam akses baik *read* maupun *write* ke *Database Development* tim lain dan khususnya tim DWH. Penundaan perolehan akses ini secara tidak langsung menghambat semua pekerjaan teknis. Selama periode menunggu *privilege* diaktifkan, pekerjaan inti DWH *Modelling* terpaksa ditangguhkan atau hanya dapat dilakukan melalui bekerja sama dengan anggota tim lain yang sudah memiliki akses. Situasi ini menyebabkan efektivitas kerja terganggu, karena implementasi teknis tidak dapat dimulai secara mandiri sebelum otorisasi akses yang dibutuhkan berhasil diperoleh.

3. Adanya deskripsi pekerjaan yang tidak dilaksanakan

Rencana kerja awal bagi posisi BI & DWH *Modeling Intern* mencakup delapan poin deskripsi pekerjaan utama yang dirancang untuk mendukung siklus pengembangan data seperti pada Lampiran H. Namun, realisasi di lapangan menunjukkan adanya kendala pada poin kelima, yaitu tugas pemantauan *pipeline* harian atau *monitoring daily pipeline job*. Kendala ini disebabkan oleh adanya pembagian fungsi yang sangat spesifik di dalam struktur departemen, di mana tugas tersebut berada di bawah otoritas *Section Head* yang berbeda dari posisi peserta magang. Tanggung jawab operasional pemantauan tersebut secara resmi ditugaskan kepada peserta magang lain yang berada di bawah pengawasan *Section Head* tersebut. Situasi ini mengakibatkan satu poin tugas tidak dapat diimplementasikan oleh mahasiswa guna menghindari tumpang tindih tanggung jawab serta menjaga alur koordinasi tetap sesuai dengan struktur organisasi.

3.3.3 Solusi atas Kendala yang Ditemukan

Setiap proyek profesional, termasuk dalam lingkungan *section Business Intelligence (BI) & Data Warehouse (DWH)* pasti memiliki tantangan dan hambatan saat melakukan suatu pekerjaan. Namun,

menghadapi dan menyelesaikan kendala inilah yang menjadi indikator penting dari proses belajar dan pengembangan diri. Kendala yang dihadapi cukup beragam, mulai dari masalah teknis seperti adaptasi terhadap *tools* baru hingga masalah administratif yang berkaitan dengan prosedur internal perusahaan. Situasi ini menuntut inisiatif untuk bersikap proaktif, tangkas, dan mampu menyesuaikan strategi kerja. Solusi yang diterapkan berfokus pada dua hal utama, yaitu pendekatan proaktif untuk mempercepat penguasaan teknologi atau *tools* baru dan penyesuaian strategis dalam menghadapi keterbatasan prosedural perusahaan. Pendekatan ini secara efektif mengubah potensi penghambat menjadi peluang sehingga mampu menciptakan prosedur kerja yang lebih efisien. Oleh karena itu, langkah-langkah selanjutnya didesain untuk memastikan efisiensi kerja tim dapat tercapai, sekaligus memanfaatkan setiap kendala yang ditemukan sebagai pembelajaran berharga dalam lingkungan kerja profesional. Berikut penjelasan lebih lanjut terkait solusi yang dilakukan atas kendala-kendala pada subbab 3.3.2.

1. Mempelajari cara penggunaan *Microsoft Fabric* secara mandiri

Mengatasi tantangan terkait adaptasi terhadap lingkungan teknologi baru, *Microsoft Fabric*, solusi yang diterapkan berfokus pada pembelajaran mandiri yang proaktif dan inisiatif *transfer knowledge*. Menyadari bahwa platform *cloud* ini berbeda dengan tools ETL tradisional yang sempat digunakan sebelumnya, langkah awal dilakukan dengan mendedikasikan waktu di jam kerja untuk eksplorasi dan belajar secara mandiri (*self-study*). Sumber daya daring seperti tutorial video di *YouTube* dan dokumentasi resmi *Microsoft* menjadi acuan utama agar fungsionalitas inti *Fabric*, terutama mekanisme alur pipeline, dapat dipahami. Selanjutnya, proses *transfer* pengetahuan dipercepat melalui diskusi dan kolaborasi aktif dengan rekan *intern BI & DWH Modelling* lain yang sudah lebih berpengalaman *menggunakan* tools tersebut. Melalui sesi

sharing informal ini, banyak praktik dan pengetahuan yang dapat dipelajari dengan cepat. Kombinasi antara inisiatif belajar mandiri yang terstruktur dan pemanfaatan pengetahuan kolektif tim terbukti sangat efektif dalam memperpendek waktu yang dibutuhkan untuk mencapai pemahaman terkait *Microsoft Fabric* tersebut sehingga sesuai dengan kebutuhan dan tujuan pemahaman.

2. Menghubungi *Supervisor* terkait email dan akses

Menghadapi kendala yang bersifat administratif terkait keterlambatan email dan *privilege* akses sistem, solusi yang diterapkan adalah meningkatkan koordinasi yang proaktif dan menerapkan penyesuaian strategi kerja yang adaptif. Walaupun permintaan email perusahaan dan hak akses telah diajukan sebelum magang dimulai, proses otorisasi internal memang membutuhkan waktu. Guna memperlancar proses dan memastikan kebutuhan pekerjaan menjadi prioritas, langkah strategis yang diambil adalah menjalin komunikasi intensif dengan *Supervisor* secara berkala untuk memantau perkembangan status permintaan email dan akses. Upaya koordinasi yang cepat ini membawa hasil, di mana email perusahaan yang vital berhasil diterima dalam tiga hari kerja setelah dimulainya program magang sehingga memungkinkan kolaborasi segera melalui *Microsoft Teams* dan *tools* lainnya. Sambil menanti aktivasi *privilege* penuh seperti akses ke *Workspace* dan *Database Development*, waktu dialokasikan secara bijaksana untuk menyelesaikan tugas-tugas lain yang tidak bergantung pada akses *database*. Strategi kerja yang fleksibel ini memastikan produktivitas tetap berjalan optimal di awal proyek. Pendekatan adaptif ini menjadi pelajaran penting mengenai bagaimana mengatasi penundaan yang disebabkan oleh prosedur administratif, sehingga implementasi teknis dapat segera dilanjutkan secara mandiri setelah semua otorisasi sistem berhasil diperoleh.

3. Penyesuaian fokus tugas dan koordinasi tim

Penyelesaian atas perbedaan rencana kerja ini dilakukan dengan mendiskusikan kembali pembagian tugas bersama pembimbing lapangan. Mahasiswa diputuskan untuk menyesuaikan fokus pekerjaan agar sejalan dengan aturan yang sudah berjalan di perusahaan, mengingat pemantauan *pipeline* memang sudah menjadi tanggung jawab peserta magang lain. Langkah ini diambil untuk memastikan tidak ada pengerajan tugas yang ganda dan agar kerja sama tim tetap rapi. Melalui penyesuaian tersebut, mahasiswa bisa lebih fokus mengerjakan tujuh poin tugas lainnya yang menjadi prioritas, seperti membantu proyek migrasi, merancang model data, dan mengelola proses *ingestion* data. Maka dari itu, tujuan magang tetap tercapai dengan hasil yang maksimal dan sesuai dengan kebutuhan tim saat ini.

