

---

BI Engineer

# Technical Requirements Document (TRD) Dashboard Department Marketing & Sales

---

*Project/Initiative*

*Month February 2022*

*Version 0.3*

*Superstore, Pltd. adalah salah satu perusahaan retail yang terletak di US. Perusahaan ini menjual berbagai kebutuhan sehari-hari seperti furniture dan sebagainya. Dengan cabang yang sudah tersebar di berbagai tempat di US. Superstore, Pltd. menunjukkan pertumbuhan yang sangat bagus dari tahun ke tahun.*

## 1 Document Revisions

Date	Version Number	Document Changes
05/04/2023	0.1	Initial Draft

## 2 Approvals

Role	Name	Title	Signature	Date
User	Subroto Nugraha	Head Department Marketing & Sales		
BI Engineer	Ahmad Rizky	Ahmad Rizky		09/04/2023
BI Engineer	Hedrin S. Sitorus	Hedrin S. Sitorus		09/04/2023
BI Engineer	Yogi Kusuma S	Yogi Kusuma S		09/04/2023

## 3 Introduction

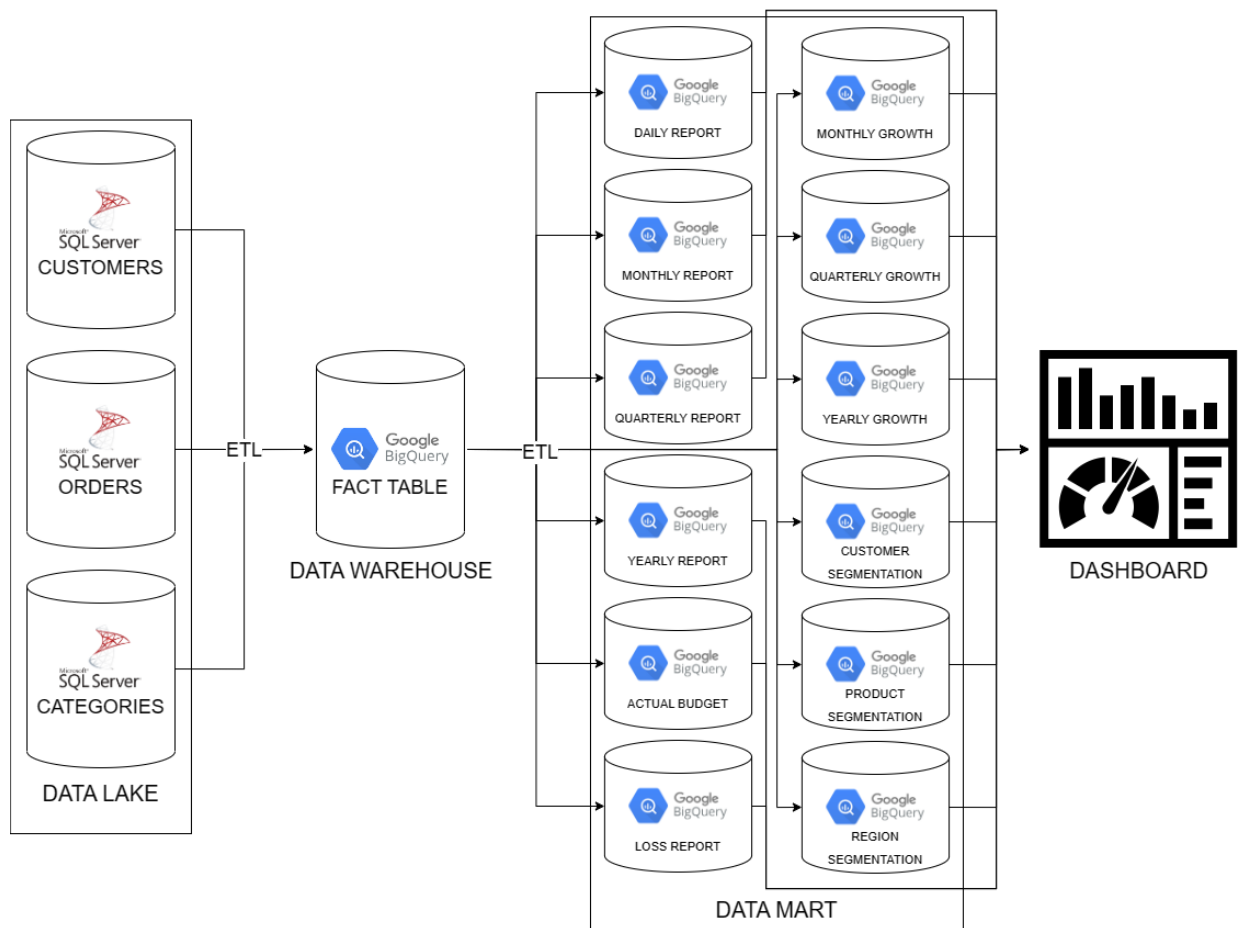
### 3.1 Project Overview

proyek ini adalah proyek dashboard Business Intelligence yang berfungsi untuk memberikan visibilitas kepada Department Marketing & Sales atas monitoring penjualan produk, performa cabang, pencapaian target penjualan secara berkala (daily, monthly, quarterly dan yearly). proyek ini diharapkan untuk menggantikan proses konsolidasi report manual yang selama ini dilakukan oleh Departemen Marketing & Sales serta menjadi dashboard yang dapat diakses oleh seluruh *branch manager* yang bertugas di Superstore untuk memonitoring dan mengevaluasi kinerja toko mereka.

### 3.2 Project Objectives

Objective yang tuju dari pembuatan dashboard ini adalah supaya dapat memberi gambaran end-to-end kepada Department Marketing & Sales atas penjualan produk di semua regional serta dapat melakukan monitoring penjualan secara daily, monthly, quarterly dan yearly.

### 3.3 Technical Design Diagram



#### 3.3.1 Components

- Connection & Location

Data Source atau sumber data untuk proyek ini terletak pada Data Lake perusahaan yang berada pada RDBMS SQL Server. Data Lake ini dapat diakses pada server perusahaan dengan IP Address 34.101.194.202 pada port 1433. Ada 3 database dari Data Lake yang

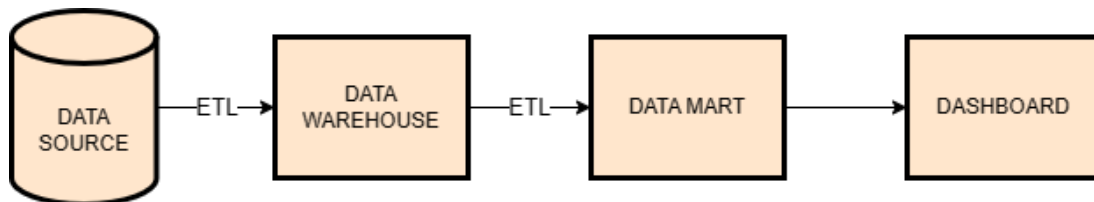
---

digunakan pada proyek ini, yaitu database Customers (berisi identitas customer), Categories (berisi detail produk), dan Orders (berisi riwayat pesanan).

- **Server & Storage**

Data Warehouse yang digunakan untuk menyimpan data adalah Google BigQuery yang diakses dengan akun Google perusahaan. Proyek BigQuery yang digunakan diberi nama "bie1-kel2".

- **Logical Data Flow**



### 3.3.2 Attachment & Link

Data Source yang digunakan pada proyek ini juga sudah diekspor dan diletakkan pada Google Drive yang hanya bisa diakses dengan email perusahaan dari tiap karyawan.

## 3.4 Technical Specification

### 3.4.1 Application Server

Berikut ini adalah spesifikasi teknis untuk application server yang digunakan dalam proyek ini:

- Operating System: Windows Server 2016 atau yang lebih baru
- Application Memory: Minimal 8 GB RAM (Disarankan 16 GB RAM atau lebih)
- Application CPU: Minimal 2 CPU Core (Disarankan 4 CPU Core atau lebih)
- Network: Koneksi internet yang stabil
- Software: Microsoft SQL Server Integration Services (SSIS) 2016 atau yang lebih baru

### 3.4.2 Database Server

- Deskripsi Penggunaan Database: Digunakan untuk membangun dan mengoperasikan data warehouse menggunakan BigQuery
- Tanggal Permintaan: 8 April 2023
- Tanggal Butuh: 10 April 2023
- Justifikasi Kebutuhan Database: Kebutuhan database diperlukan untuk membangun dan mengoperasikan data warehouse menggunakan BigQuery. BigQuery dipilih karena kemampuannya dalam menangani volume data yang besar dan fitur-fitur analisis data yang lengkap.

### 3.4.3 Server Request

- Deskripsi Penggunaan Server: Digunakan untuk menjalankan SSIS Package dan menyimpan data sementara selama proses ETL
- Tanggal Permintaan: 3 April 2023
- Tanggal Butuh: 5 April 2023

- Justifikasi Kebutuhan Server: Kebutuhan server diperlukan untuk menjalankan SSIS Package dan menyimpan data sementara selama proses ETL. Server dengan spesifikasi yang cukup diperlukan untuk memastikan kinerja SSIS Package optimal.

Spesifikasi Server yang Dibutuhkan:

- Sistem Operasi: Windows Server 2019
- RAM: Minimal 32 GB
- CPU: Intel Xeon Processor
- Storage: Minimal 1 TB
- Hak Akses: Admin dan Developer
- Nama Server: Server\_xxx

#### 3.4.4 Database Request

No	Process Name	Data	Formula
1	tabel fakta 'Fact_Sales' pada data warehouse	Fact_Sales	<pre>CREATE TABLE Fact_Sales (     row_id INT64,     order_id STRING,     id_waktu_order STRING,     id_waktu_shipping STRING,     id_shipping STRING,     id_produk STRING,     id_customer STRING,     id_location INT64,     sales FLOAT64,     quantity INT64,     discount FLOAT64,     profit FLOAT64,     created_time TIMESTAMP,     modified_time TIMESTAMP );</pre>

2	tabel dimensi 'dimensi_customer' pada data warehouse	dimensi_customer	<pre>CREATE TABLE dimensi_customer (     id_customer STRING NOT NULL,     customer_name STRING,     customer_segment STRING,     PRIMARY KEY (id_customer) );</pre>
3	tabel dimensi 'dimensi_waktu' pada data warehouse	dimensi_waktu	<pre>CREATE TABLE dimensi_waktu (     id_waktu STRING NOT NULL,     tanggal DATE,     tahun INT64,     bulan INT64,     nama_bulan STRING,     hari_dalam_bulan INT64,     hari_dalam_tahun INT64,     minggu_dalam_bulan INT64,     minggu_dalam_tahun INT64,     kuartal INT64,     semester INT64,     PRIMARY KEY (id_waktu) );</pre>
4	tabel dimensi 'dimensi_shipping' pada data warehouse	dimensi_shipping	<pre>CREATE TABLE dimensi_shipping (     id_shipping STRING NOT NULL,     ship_mode STRING,     PRIMARY KEY (id_shipping) );</pre>

5	tabel dimensi 'dimensi_location_cabang' pada data warehouse	dimensi_location_cabang	<pre> CREATE TABLE dimensi_location_cabang (     id_location INT64 NOT NULL,     country STRING,     region STRING,     state STRING,     city STRING,     PRIMARY KEY (id_location) ); </pre>
6	tabel dimensi 'dimensi_produk' pada data warehouse	dimensi_produk	<pre> CREATE TABLE dimensi_produk (     id_produk STRING NOT NULL,     nama_produk STRING,     category STRING,     sub_category STRING,     PRIMARY KEY (id_produk) ); </pre>
7	tabel metric 'daily_report.' untuk Daily reporting	daily_report	<pre> CREATE OR REPLACE TABLE biel-kel2.yogi_dwh_superstore.dail y_report AS SELECT     tahun,     hari_dalam_tahun AS hari,     COUNT(sales) AS total_sales,     SUM(sales - (sales * discount)) AS sales_bersih FROM     biel-kel2.yogi_dwh_superstore.Fact _Sales AS Fact_Sales LEFT JOIN     biel-kel2.yogi_dwh_superstore.dime nsi_produk AS dimensi_produk ON dimensi_produk.id_produk = Fact_Sales.id_produk </pre>

			<pre> LEFT JOIN biel-kel2.yogi_dwh_superstore.dime nsi_waktu AS dimensi_waktu  ON dimensi_waktu.id_waktu = Fact_Sales.id_waktu_order  GROUP BY hari_dalam_tahun, tahun  ORDER BY tahun, hari_dalam_tahun ASC </pre>
8	tabel metric 'monthly_report' untuk Monthly reporting	monthly_report	<pre> create or replace table `biel-kel2.yogi_dwh_superstore.mon thly_report` as  select      tahun,      bulan,      count(sales) as total_sales,      sum(sales-(sales * discount)) as sales_bersih  from `biel-kel2.yogi_dwh_superstore.Fac t_Sales` as Fact_Sales  left join `biel-kel2.yogi_dwh_superstore.dim ensi_produk` as dimensi_produk      on dimensi_produk.id_produk = Fact_Sales.id_produk  left join `biel-kel2.yogi_dwh_superstore.dim ensi_waktu` as dimensi_waktu      on dimensi_waktu.id_waktu = Fact_Sales.id_waktu_order  group by bulan, tahun  order by tahun, bulan asc </pre>



9	tabel metric 'kuartal_report' untuk Quarterly report	kuartal_report	<pre> CREATE OR REPLACE TABLE `biel-kel2.yogi_dwh_superstore.kua rtal_report` AS  SELECT      tahun,      kuartal,      COUNT(sales) AS total_sales,      SUM(sales - (sales * discount)) AS sales_bersih  FROM `biel-kel2.yogi_dwh_superstore.Fac t_Sales` AS Fact_Sales  LEFT JOIN `biel-kel2.yogi_dwh_superstore.dim ensi_produk` AS dimensi_produk     ON dimensi_produk.id_produk = Fact_Sales.id_produk  LEFT JOIN `biel-kel2.yogi_dwh_superstore.dim ensi_waktu` AS dimensi_waktu     ON dimensi_waktu.id_waktu = Fact_Sales.id_waktu_order  GROUP BY kuartal, tahun  ORDER BY tahun, kuartal ASC </pre>
---	---	----------------	---

10	tabel metric 'years_report' untuk Yearly reporting	years_report	<pre> create or replace table biel-kel2.yogi_dwh_superstore.year s_report as  select  tahun,  count(sales) as total_sales,  sum(sales-(sales * discount)) as sales_bersih  from biel-kel2.yogi_dwh_superstore.Fact _Sales as Fact_Sales  left join biel-kel2.yogi_dwh_superstore.dime nsi_produk as dimensi_produk on dimensi_produk.id_produk = Fact_Sales.id_produk  left join biel-kel2.yogi_dwh_superstore.dime nsi_waktu as dimensi_waktu on dimensi_waktu.id_waktu = Fact_Sales.id_waktu_order  group by tahun  order by tahun asc </pre>
11	tabel metric 'actual_vs_budget' untuk Actual vs Budget	'actual_vs_budget'	<pre> CREATE OR REPLACE TABLE `biel-kel2.yogi_dwh_superstore.act ual_vs_budget` AS  SELECT      SUM(actual_value) AS total_actual_value,      SUM(budget_value) AS total_budget_value,      ((SUM(actual_value)/SUM(budget_val ue)) * 100) AS percentage  FROM      (SELECT </pre>

			<pre> tahun,  sales_bersih AS actual_value,  (LAG(sales_bersih) OVER (ORDER BY tahun) * 1.1) AS budget_value  FROM  `biel-kel2.yogi_dwh_superstore.yea rs_report`); </pre>
12	tabel metric 'monthly_growth' untuk Current Month vs Last Month Growth	monthly_growth	<pre> CREATE OR REPLACE TABLE `biel-kel2.yogi_dwh_superstore.mon thly_growth` AS  SELECT  tahun,  bulan,  SUM(sales) as sales,  ROUND(((SUM(sales) - LAG(SUM(sales)) OVER (ORDER BY tahun, bulan))/LAG(SUM(sales)) OVER (ORDER BY tahun, bulan)) * 100, 2) AS growth_percentage  FROM  `yogi_dwh_superstore.Fact_Sales` AS fact_sales  LEFT JOIN  `yogi_dwh_superstore.dimensi_waktu ` AS dimensi_waktu  ON fact_sales.id_waktu_order = dimensi_waktu.id_waktu  GROUP BY tahun, bulan  ORDER BY tahun, bulan; </pre>

13	tabel metric 'kuartal_growth' untuk Current Quarter vs Last Quarter Growth	kuartal_growth	<pre> CREATE OR REPLACE TABLE `biel-kel2.yogi_dwh_superstore.kua rtal_growth` AS  SELECT      tahun,      kuartal,      SUM(sales) as sales,      ROUND(((SUM(sales) - LAG(SUM(sales)) OVER (ORDER BY tahun, kuartal))/LAG(SUM(sales)) OVER (ORDER BY tahun, kuartal)) * 100, 2) AS growth_percentage  FROM `yogi_dwh_superstore.Fact_Sales` AS fact_sales  LEFT JOIN `yogi_dwh_superstore.dimensi_waktu ` AS dimensi_waktu  ON fact_sales.id_waktu_order = dimensi_waktu.id_waktu  GROUP BY tahun, kuartal  ORDER BY tahun, kuartal; </pre>
14	tabel metric 'yearly_growth' untuk Current Year vs Last Year Growth	yearly_growth	<pre> CREATE OR REPLACE TABLE `biel-kel2.yogi_dwh_superstore.yea rly_growth` AS  SELECT      tahun,      SUM(sales) as sales,      ROUND(((SUM(sales) - LAG(SUM(sales)) OVER (ORDER BY tahun))/LAG(SUM(sales)) OVER (ORDER BY tahun)) * 100, 2) AS growth_percentage  FROM `yogi_dwh_superstore.Fact_Sales` AS fact_sales </pre>

			<pre>LEFT JOIN `yogi_dwh_superstore.dimensi_waktu ` AS dimensi_waktu  ON fact_sales.id_waktu_order = dimensi_waktu.id_waktu  GROUP BY tahun  ORDER BY tahun;</pre>
--	--	--	--

15	tabel metric 'loss_report' untuk Loss reporting	loss_repor	<pre> create or replace table `biel-kel2.yogi_dwh_superstore.loss_report` as  select     tahun,     kuartal,     bulan,     city,     dimensi_produk.id_produk as id_produk,     nama_produk,     category,     sub_category,     sum(profit) as loss  from `yogi_dwh_superstore.Fact_Sales` as fact_sales  left join `yogi_dwh_superstore.dimensi_waktu` as dimensi_waktu  on fact_sales.id_waktu_order = dimensi_waktu.id_waktu  left join `yogi_dwh_superstore.dimensi_produk` as dimensi_produk  on fact_sales.id_produk = dimensi_produk.id_produk  left join `yogi_dwh_superstore.dimensi_location_cabang` as dimensi_location_cabang  on fact_sales.id_location =dimensi_location_cabang.id_location  where profit&lt;0  group by </pre>
----	---	------------	---

			tahun, kuartal, bulan, city, id_produk, nama_produk, category, sub_category order by tahun, kuartal, bulan
16	tabel metric 'segmentasi_customer' untuk Segmentasi Customer	segmentasi_custome r	create or replace table `biel-kel2.yogi_dwh_superstore.segmentasi_customer` as select tahun, kuartal, dimensi_customer.id_customer as id_customer, customer_name, customer_segment, case when sum(sales)<200 then 'Bronze' when sum(sales)<500 then 'Silver' else 'Gold' end as segmentasi from `yogi_dwh_superstore.Fact_Sales` as fact_sales left join `yogi_dwh_superstore.dimensi_waktu` as dimensi_waktu

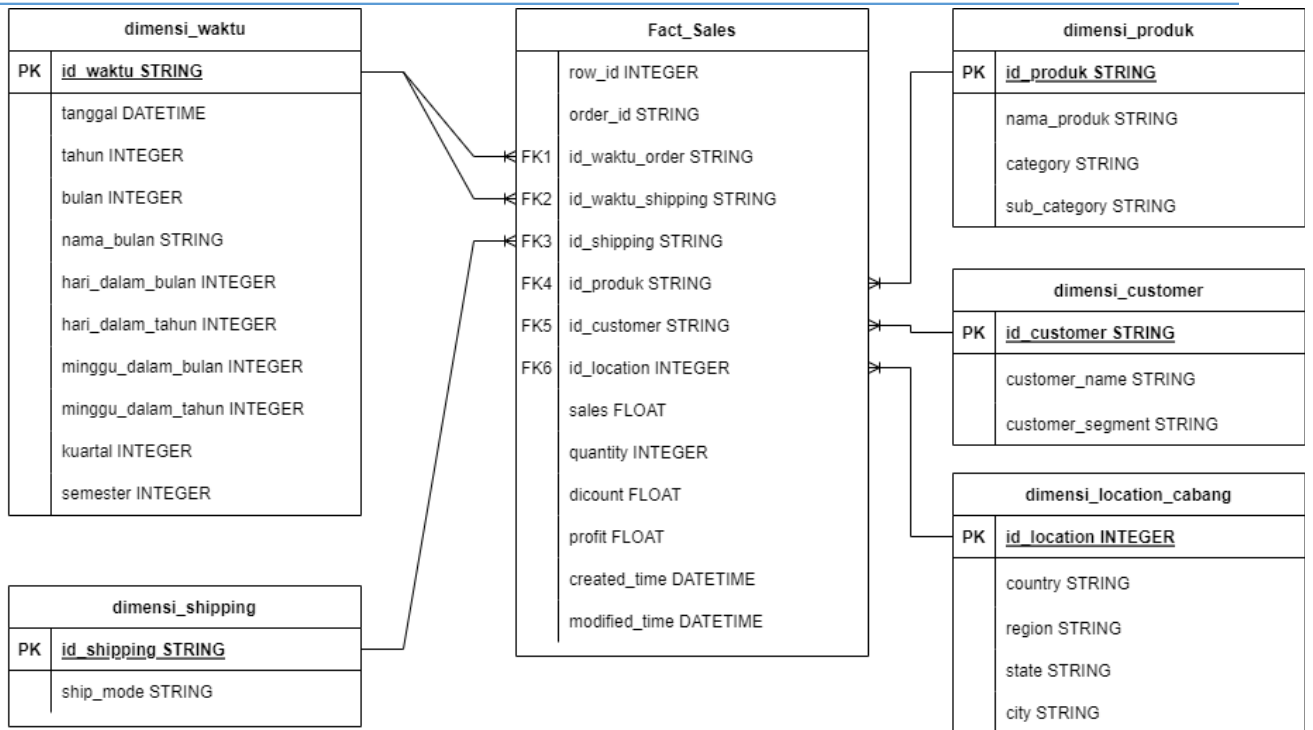
			<pre>on fact_sales.id_waktu_order = dimensi_waktu.id_waktu  left join `yogi_dwh_superstore.dimensi_custome mer` as dimensi_customer  on fact_sales.id_customer = dimensi_customer.id_customer  group by tahun, kuartal,dimensi_customer.id_custom er,customer_name,customer_segment  order by tahun, kuartal asc</pre>
--	--	--	--



17	tabel metric 'segmentasi_products' untuk Segmentasi Product	segmentasi_products	<pre> create or replace table `biel-kel2.yogi_dwh_superstore.segmentasi_products` as  select     tahun,     bulan,     city,     region,     dimensi_produk.id_produk as id_produk,     nama_produk,     category,     sub_category,     sum(quantity) as quantity,     case         when sum(quantity)&lt;5 then '3rd Product'         when sum(quantity)&lt;10 then '2nd Product'         else '1st Product'     end as segmentasi  from `yogi_dwh_superstore.Fact_Sales` as fact_sales  left join `yogi_dwh_superstore.dimensi_waktu` as dimensi_waktu  on fact_sales.id_waktu_order = dimensi_waktu.id_waktu  left join `yogi_dwh_superstore.dimensi_produk` as dimensi_produk  on fact_sales.id_produk = dimensi_produk.id_produk  left join `yogi_dwh_superstore.dimensi_locat </pre>
----	--	---------------------	---

			<pre>ion_cabang` as dimensi_location_cabang on fact_sales.id_location =dimensi_location_cabang.id_locati on group by tahun,     bulan,     city,     region,     dimensi_produk.id_produk,     nama_produk,     category,     sub_category order by tahun, bulan asc</pre>
--	--	--	--

18	tabel metric 'segmentasi_cabang' untuk Segmentasi Toko Cabang	segmentasi_cabang	<pre> create or replace table `biel-kel2.yogi_dwh_superstore.segmentasi_cabang` as  select     tahun,     bulan,     city,     sum(sales) as sales,     case         when sum(sales)&lt;1000 then 'Kategori 1'         when sum(sales)&lt;2000 then 'Kategori 2'         else 'Kategori 3'     end as segmentasi  from `yogi_dwh_superstore.Fact_Sales` as fact_sales  left join `yogi_dwh_superstore.dimensi_waktu` as dimensi_waktu  on fact_sales.id_waktu_order = dimensi_waktu.id_waktu  left join `yogi_dwh_superstore.dimensi_location_cabang` as dimensi_location_cabang  on fact_sales.id_location =dimensi_location_cabang.id_location  group by tahun,     bulan,     city  order by tahun, bulan asc </pre>
----	---	-------------------	---



daily_report		monthly_report		kuartal_report		years_report	
	tahun INTEGER		tahun INTEGER		tahun INTEGER		tahun INTEGER
	hari INTEGER		bulan INTEGER		kuartal INTEGER		total_sales INTEGER
	total_sales INTEGER		total_sales INTEGER		total_sales INTEGER		sales_bersih FLOAT
	sales_bersih FLOAT		sales_bersih FLOAT		sales_bersih FLOAT		

actual_vs_budget	
	total_actual_value FLOAT
	tital_budget_value FLOAT
	percentage FLOAT

monthly_growth	
	tahun INTEGER
	bulan INTEGER
	sales FLOAT
	growth_percentage FLOAT

kuartal_growth	
	tahun INTEGER
	kuartal INTEGER
	sales FLOAT
	growth_percentage FLOAT

yearly_growth	
	tahun INTEGER
	sales FLOAT
	growth_percentage FLOAT

segmentasi_customer	
	tahun INTEGER
	kuartal INTEGER
	id_customer STRING
	customer_name STRING
	customer_segment STRING
	segmentasi STRING

segmentasi_products	
	tahun INTEGER
	bulan INTEGER
	city STRING
	region STRING
	id_produk STRING
	nama_produk STRING
	category STRING
	sub_category STRING

segmentasi_cabang	
	tahun INTEGER
	bulan INTEGER
	city STRING
	sales FLOAT
	segmentasi STRING

loss_report	
	tahun INTEGER
	kuartal INTEGER
	bulan INEGER
	city STRING
	id_produk STRING
	nama_produk STRING
	category STRING
	sub_category STRING
	loss FLOAT

#### 3.4.5 Access Request

- Head Departement Marketing & Sales pusat: Full access untuk semua region dan kota.
- PIC Departement Marketing & Sales pusat: Full access untuk semua region dan kota.
- Semua Manager di masing-masing region: Hanya dapat mengakses data untuk region yang bersangkutan.
- Semua Branch Manager di masing-masing kota: Hanya dapat mengakses data untuk kota yang bersangkutan.

### 3.5 Resources Requirement

#### 3.5.1 Department & Staff

No	Departemen	Staff
1	Head Departemen Marketing & Sales	Subroto Nugraha
2	BI Engineer	Ahmad Rizky
3	BI Engineer	Hedrin S. Sitorus
4	BI Engineer	Yogi Kusuma
5	BI Analyst	Andi Wijaya

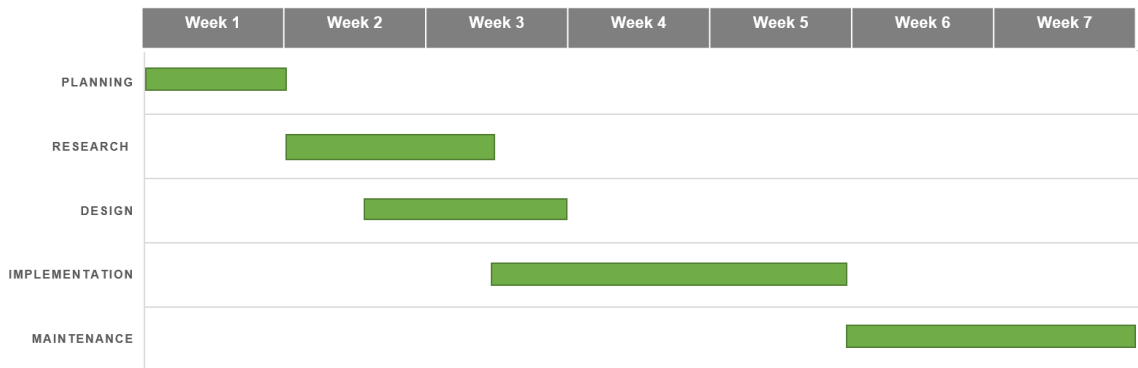
#### 3.5.2 Technical Dependencies

- ETL data dari Data Lake ke Data Warehouse dilakukan tiap jam 8 pagi.
- Data identitas customer lengkap dan valid.
- Data detail produk lengkap, tapi ada 32 data yang tidak valid. Ke-32 data tersebut tidak valid, karena memiliki id produk yang sama dengan 32 item lainnya (1 id merujuk pada 2 item berbeda). Setelah dikonfirmasi kepada user, ternyata untuk setiap id, produk lama adalah produk yang valid. Oleh karena itu, untuk setiap id, data dengan index yang lebih besar bisa dihapus.
- Data riwayat order valid, tetapi ada 4 data yang tidak lengkap. Keempat data order tersebut dinyatakan tidak lengkap, karena tidak ada id produk yang terekam (null). Setelah dikonfirmasi, keempat data tersebut ternyata terekam pada saat terjadi bug pada sistem pencatatan order, dimana sistem memperbolehkan tagihan ter-generate meskipun customer tidak memilih produk apapun.

#### 3.5.3 Non-Technical Dependencies

Anomali pada data telah dikomunikasikan terlebih dahulu kepada user (Head of Marketing & Sales Department)

### 3.6 Timeline & Deadlines



### 3.7 Assumptions

- Proyek ini akan memberikan visibilitas kepada Departemen Pemasaran & Penjualan terhadap monitoring penjualan produk, performa cabang, pencapaian target penjualan secara berkala (daily, monthly, quarterly, dan yearly). Asumsi ini menyiratkan bahwa proyek ini akan menghasilkan dashboard Business Intelligence (BI) yang dapat menggantikan proses konsolidasi report manual yang selama ini dilakukan oleh Departemen Pemasaran & Penjualan.
- Proyek ini diharapkan dapat diakses oleh seluruh branch manager yang bertugas di Superstore untuk memonitoring dan mengevaluasi kinerja toko mereka. Asumsi ini mengindikasikan bahwa proyek ini akan menyediakan akses ke dashboard BI kepada seluruh branch manager, sehingga mereka dapat memantau dan mengevaluasi kinerja toko mereka dengan mudah.
- Proyek ini akan membantu Departemen Pemasaran & Penjualan dalam mengambil keputusan yang lebih baik berdasarkan data yang dihasilkan dari dashboard BI. Asumsi ini mengimplikasikan bahwa proyek ini diharapkan dapat menyediakan informasi yang relevan dan akurat kepada Departemen Pemasaran & Penjualan, sehingga mereka dapat membuat keputusan yang lebih cerdas dan efektif dalam mengelola penjualan produk, performa cabang, serta pencapaian target penjualan.
- Proyek ini akan mengurangi waktu dan usaha yang diperlukan untuk melakukan proses konsolidasi report manual oleh Departemen Pemasaran & Penjualan. Asumsi ini menyiratkan bahwa proyek ini akan menghemat waktu dan usaha yang sebelumnya diperlukan untuk melakukan proses konsolidasi report manual, karena dashboard BI yang dihasilkan akan otomatis menggabungkan data yang diperlukan dalam satu tampilan yang mudah diakses.
- Proyek ini akan meningkatkan efisiensi dan efektivitas kerja Departemen Pemasaran & Penjualan serta branch manager dalam memantau dan mengevaluasi kinerja penjualan. Asumsi ini mengindikasikan bahwa dashboard BI yang dihasilkan akan membantu meningkatkan efisiensi dan efektivitas kerja Departemen Pemasaran & Penjualan serta branch manager, karena mereka dapat mengakses data penjualan secara real-time dan membuat keputusan yang lebih cepat dan tepat berdasarkan informasi yang disediakan.
- Proyek ini akan menghadapi tantangan dalam mengintegrasikan data dari berbagai sumber yang berbeda, seperti sistem penjualan, sistem pencatatan cabang, dan sistem pencapaian target penjualan. Asumsi ini mengimplikasikan bahwa proyek ini akan memerlukan upaya dalam mengintegrasikan data dari berbagai sumber yang berbeda untuk menghasilkan dashboard BI yang komprehensif dan akurat.

- Proyek ini akan memerlukan investasi dalam pengembangan dan implementasi teknologi BI yang sesuai, seperti perangkat lunak BI, perangkat keras, serta infrastruktur IT yang diperlukan untuk menjalankan dashboard BI. Asumsi ini mengindikasikan bahwa proyek ini akan memerlukan anggaran dan sumber daya yang cukup untuk memastikan pengembangan

### 3.8 Project Cost

Komponen Biaya	Deskripsi	Detail tools	Estimasi Biaya
Tenaga Kerja	Biaya untuk tim pengembangan dashboard dan implementasi	-	Rp.5,000,000
Perangkat Lunak	Biaya lisensi perangkat lunak BI dan infrastruktur IT	Big Query	Rp.4,482,285
		Visual Studio Code	-
		Visual Studio	-
		Excel	-
		Microsoft SQL Server Management Studio 2019	-
Perangkat Keras	Biaya perangkat keras untuk server dan penyimpanan data	-	Rp. 3,000,000
Pelatihan	Biaya pelatihan untuk pengguna dan administrator dashboard	-	Rp.1,000,000
Pengelolaan Proyek	Biaya pengelolaan proyek, koordinasi, dan dokumentasi	-	Rp.2,000,000

Total Biaya		-	Rp.15,482,285
-------------	--	---	---------------

## 4 Appendices

### 4.1 List of Acronyms

- BI: Business Intelligence
- RDBMS: Relational Database Management System
- SQL: Structured Query Language
- IP: Internet Protocol
- CPU: Central Processing Unit
- ERD: Entity Relationship Diagram
- ETL: Extract, Transform, Load

### 4.2 Glossary of Terms

- **Technical Requirements Document (TRD)**: Dokumen yang berisi spesifikasi teknis yang diperlukan untuk proyek atau inisiatif tertentu.
- **Dashboard**: Papan kontrol yang menampilkan data dalam bentuk grafik, tabel, atau metrik lainnya untuk memberikan visibilitas dan pemahaman tentang kinerja suatu proyek atau sistem.
- **Data Lake**: Repositori data yang menyimpan berbagai jenis data dalam volume besar, baik struktur maupun tidak terstruktur, untuk dianalisis dan diproses.
- **Data Warehouse**: Tempat penyimpanan data yang dirancang khusus untuk menganalisis data bisnis dan memudahkan akses terhadap data tersebut.
- **Google BigQuery**: Layanan manajemen data dan analisis data yang ditawarkan oleh Google Cloud Platform.
- **Google Drive**: Layanan penyimpanan dan berbagi file yang disediakan oleh Google.
- **Access Request**: Permintaan izin akses ke sistem atau aplikasi yang diajukan oleh pengguna kepada pihak yang berwenang.
- **ETL (Extract, Transform, Load)**: Proses pengambilan data dari sumber, mengubah format dan struktur data, dan memuat data ke tempat penyimpanan data yang baru.
- **ERD (Entity Relationship Diagram)**: Diagram yang digunakan untuk menggambarkan struktur relasi antara entitas atau objek dalam suatu sistem basis data.
- **Non-valid**: Data yang tidak memenuhi syarat atau kriteria yang ditetapkan dalam konteks tertentu.

### 4.3 Related Documents

Folder Google Drive berisi data source yang telah diekstrak ke dalam bentuk CSV dapat diakses pada pranala berikut: [Data Source](#).