

STUDI KASUS PEMBANGUNAN DATA WAREHOUSE HASIL MERGER PHILIP MORRIS INTERNATIONAL DAN ALTRIA

Amir Salim¹, Muhammad Giat², Aryan Dafi Maulana³, Varian Avila Faldi⁴

Program Studi Teknik Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Padjadjaran

¹amir21001@mail.unpad.ac.id, ²muhammad21019@mail.unpad.ac.id,
³aryan21001@mail.unpad.ac.id, ⁴varian21001@mail.unpad.ac.id

Abstrak

Pemanfaatan *Data Warehouse* dalam pemanfaatan bisnis sangatlah memiliki dampak positif bagi banyak perusahaan. Gudang data/*Data warehouse* membentuk sebuah sistem yang bertugas mengarsipkan sekaligus melakukan analisis data historis untuk menunjang keperluan informasi pada sebuah bisnis maupun organisasi. Dengan dibentuknya *Data Warehouse*, pihak bisnis/organisasi dapat lebih mudah membuat suatu keputusan atau menganalisis keadaan organisasi dikarenakan *Data Warehouse* dapat membentuk pengetahuan yang penting dimanfaatkan dalam aspek organisasi (*Business Intelligence*). Philip Morris International dan Altria adalah dua perusahaan rokok yang cukup besar pada industri rokok. Pada tahun 2019, kedua perusahaan ini melakukan *merger* atau penggabungan dibawah nama Phillip Morris. Tentu dalam proses *merger* dari dua perusahaan ini, terdapat penggabungan banyak komponen salah satunya adalah data yang menjadi poin penting pada setiap bisnis. Penelitian ini memiliki tujuan untuk membuat Skema rancangan *Data Warehouse* bagi kedua perusahaan ini menggunakan Skema *Star*. Dengan pembentukan Skema *Star*, diharapkan Philip Morris International dapat memiliki pengetahuan akan transaksi, kapasitas gudang dan pabrik, dan hubungan transaksi dan gudang terhadap penjualan dan kapasitas sehingga dapat memberikan pengetahuan akan logistik, penyebaran, dan penjualan Phillip Morris International.

Kata kunci: *Data Warehouse, Skema Bintang, analisis transaksi, warehouse capacity, analisis produksi*

Abstract

The use of Data Warehouses in business has had a positive impact on many companies. A data warehouse forms a system that is tasked with archiving and analyzing historical data to support the information needs of a business or organization. By establishing a Data Warehouse, businesses/organizations can more easily make decisions or analyze the state of the organization because the Data Warehouse can form knowledge that is important for use in organizational aspects (Business Intelligence). Philip Morris International and Altria are two fairly large cigarette companies in the cigarette industry. In 2019, these two companies merged or merged under the name Phillip Morris. Of course, in the merger process of these two companies, there was a merger of many components, one of which was data which is an important point in every business. This research aims to create a Data Warehouse design scheme for these two companies using the Star Scheme. With the formation of the Star Scheme, it is hoped that Philip Morris International will be able to have knowledge of the largest and smallest transactions in each warehouse, the largest and smallest capacities in each warehouse - the largest and smallest production in what factories, the relationship between transactions and warehouses on sales and capacity so that it can provide knowledge of Phillip Morris International logistics, deployment and sales.

Keywords: *Data Warehouse, Star Schema, transaction analysis, warehouse capacity, production analysis*

I. PENDAHULUAN

Merger antara Philip Morris International (PMI) dan Altria Group memiliki tujuan yang mungkin meliputi penguatan pasar global dan diversifikasi produk. Dengan bergabung, kedua perusahaan dapat memperkuat posisi mereka di pasar tembakau global serta menciptakan portofolio produk yang lebih luas untuk menghadapi perubahan preferensi konsumen dan regulasi yang berkembang. Selain itu, merger tersebut dapat mencapai efisiensi operasional dengan mengurangi tumpang tindih dalam fungsi bisnis dan menghasilkan penghematan biaya. Melalui gabungan kekuatan dalam penelitian dan pengembangan (R&D), mereka dapat mempercepat inovasi produk dan mengembangkan produk-produk baru yang memenuhi permintaan pasar yang berkembang, seperti produk dengan risiko berkurang. Selain itu, merger dapat memberikan akses ke pasar yang sebelumnya tidak terjangkau oleh salah satu perusahaan secara independen serta membantu menghadapi tantangan regulasi yang ketat dengan memiliki lebih banyak sumber daya dan pengaruh untuk berinteraksi dengan regulator. Meskipun demikian, rencana merger ini masih bersifat spekulatif, dan tujuan pastinya mungkin berubah tergantung pada kondisi bisnis dan strategi kedua perusahaan.

Philip Morris International (PMI) adalah salah satu perusahaan rokok terkemuka di dunia dengan sejarah panjang yang dimulai pada tahun 1847 oleh pendirinya, Philip Morris, di London, Inggris. Perusahaan ini dikenal atas merek-merek rokok terkenalnya seperti Marlboro, L&M, dan Parliament. PMI memiliki kehadiran global yang kuat di lebih dari 180 negara di seluruh dunia, dengan operasi yang meliputi Eropa, Asia, Amerika Latin, Afrika, dan Timur Tengah. Selain menghasilkan rokok konvensional, PMI telah melakukan transformasi besar-besaran dengan beralih ke produk dengan risiko berkurang seperti rokok elektronik dan produk tembakau tanpa asap sebagai bagian dari upayanya untuk menyesuaikan diri dengan perubahan preferensi konsumen dan regulasi yang berkembang. Perusahaan terus melakukan inovasi dalam upaya untuk memperluas portofolio produknya dan mempertahankan posisinya sebagai pemimpin dalam industri tembakau global.

Altria Group adalah salah satu perusahaan terkemuka dalam industri tembakau di Amerika Serikat. Didirikan pada tahun 1985, Altria memiliki sejarah yang panjang dalam menghasilkan dan mendistribusikan produk tembakau, termasuk

merek rokok terkenal seperti Marlboro, Virginia Slims, dan Black & Mild. Perusahaan ini juga memiliki saham mayoritas di perusahaan rokok elektronik terkemuka, Juul Labs Inc. Selain itu, Altria memiliki kepentingan dalam industri minuman keras melalui kepemilikan saham di Anheuser-Busch InBev, salah satu produsen bir terbesar di dunia. Altria juga berinvestasi dalam industri cannabis melalui perusahaan Cronos Group. Dengan kehadiran luasnya dalam berbagai sektor, Altria terus beradaptasi dengan perubahan dalam preferensi konsumen dan regulasi industri, sambil tetap menjadi pemain utama dalam industri tembakau dan produk konsumen di Amerika Serikat.

II. LANDASAN TEORI

A. DATA WAREHOUSE

Menurut W.H. Inmon (2006), data warehouse adalah suatu koleksi data yang bisa digunakan untuk menunjang pengambilan keputusan manajemen, yang berorientasi subjek (topik), terpadu, time variant, dan tidak mudah berubah[1]. Secara umum, data Warehouse adalah suatu sistem yang dirancang untuk menyimpan dan mengelola data secara terpusat dari berbagai sumber yang berbeda, termasuk data operasional, historis, dan eksternal. Tujuan utamanya adalah untuk menyediakan akses yang mudah dan cepat terhadap data yang relevan bagi keperluan analisis dan pengambilan keputusan dalam suatu organisasi. Data Warehouse memungkinkan penyimpanan data dalam format yang terstruktur, terintegrasi, dan mudah dimengerti, sehingga memfasilitasi proses analisis yang lebih mendalam.

B. DATA CLEANING

Data Cleaning yang juga disebut sebagai pembersihan data, merupakan tahap di mana data yang tidak lengkap, mengandung kesalahan, dan tidak konsisten dibersihkan dari kumpulan data, sehingga data yang sudah bersih dan relevan dapat digunakan kembali untuk proses analisis guna menemukan wawasan baru[2]. Tujuan dari proses ini adalah untuk memastikan bahwa data yang digunakan dalam analisis atau pemrosesan lebih akurat, konsisten, dan dapat diandalkan. Proses pembersihan data ini melibatkan langkah-langkah seperti deteksi dan koreksi kesalahan pengetikan, penanganan nilai yang hilang atau tidak lengkap, standarisasi format data, dan eliminasi duplikat.

C. DATA INTEGRATION

Data Integration adalah proses penggabungan data dari berbagai sumber yang berbeda menjadi satu set data yang terpadu, konsisten, dan bermakna. Tujuannya adalah untuk menciptakan pandangan menyeluruh tentang bisnis atau organisasi dengan menggabungkan informasi dari berbagai sistem atau sumber data yang berbeda. Proses ini dapat melibatkan transformasi data, penyesuaian format, dan penyatuan struktur data yang berbeda untuk memungkinkan analisis yang lebih holistik dan mendalam.

D. DATA TRANSFORMATION

Proses mengubah data ke dalam bentuk yang dapat digunakan dalam perhitungan suatu algoritma[3]. Proses ini berguna untuk analisis, pelaporan, atau integrasi data. Tujuannya adalah untuk menghasilkan data yang lebih berguna, relevan, atau sesuai dengan kebutuhan spesifik suatu aplikasi atau proses bisnis.

E. STAR SCHEMA

Skema bintang adalah sebuah struktur dalam pemodelan multidimensi yang terdiri dari satu tabel fakta yang terhubung ke beberapa tabel dimensi. Tabel fakta berisi data yang terukur atau dapat dihitung, seperti penjualan atau pendapatan, sedangkan tabel dimensi berisi informasi yang mendeskripsikan data fakta tersebut, seperti waktu, lokasi, atau produk.

Dalam skema bintang, setiap dimensi tidak memiliki tingkatan atau sub-dimensi tambahan. Artinya, struktur dimensi sederhana dan tidak kompleks. Setiap dimensi berdiri sendiri sebagai entitas terpisah tanpa adanya relasi hierarki di dalamnya.

Kinerja query pada skema bintang lebih cepat dibandingkan dengan model pemodelan multidimensi lainnya[4]. Hal ini mungkin disebabkan oleh struktur yang sederhana dan langsung sehingga memudahkan sistem database dalam melakukan akses dan pengolahan data. Dengan demikian, skema bintang menjadi pilihan yang populer dalam pembangunan sistem analisis data dan business intelligence karena efisiensinya dalam kinerja query.

III. METODOLOGI

A. Proses Pengumpulan dan pengolahan data

Dalam studi kasus ini, proses pengumpulan data untuk pembangunan data warehouse hasil merger Philip Morris International (PMI) dan Altria Group,

Inc. melibatkan ekstraksi data dari berbagai sumber internal perusahaan. Pertama, data pabrik dari PMI dan Altria dikumpulkan, yang mencakup informasi seperti ID pabrik, negara/negara bagian, kota, dan kapasitas produksi tahunan. Sumber data ini diperoleh dari Laporan Tahunan PMI 2022 dan Laporan Tahunan Altria 2022. Selanjutnya, data gudang dari kedua perusahaan juga diekstrak, yang meliputi detail seperti ID gudang, negara/negara bagian, kota, dan kapasitas gudang. Sumber data gudang berasal dari Laporan Operasional PMI 2022 dan Laporan Operasional Altria 2022.

Selain itu, data transaksi penjualan dari PMI dan Altria juga dikumpulkan untuk diintegrasikan ke dalam data warehouse. Data ini meliputi informasi seperti ID transaksi, tahun, negara/negara bagian, jumlah produk, total penjualan, dan ID gudang terkait. Sumber data transaksi penjualan diperoleh dari Laporan Keuangan PMI Q4 2022 dan Laporan Keuangan Altria Q4 2022. Data akan diolah menggunakan proses ELT (*Extract, Load, Transform*), sehingga data yang awalnya terpisah pada 2 perusahaan, dapat menjadi satu dan diolah menjadi Data Warehouse.

B. Proses Pembentukan Skema Data Warehouse

Terdapat 2 Skema Data Warehouse yang cukup umum dalam pembentukan Data Warehouse yakni *snowflake* dan *Star*. Kedua skema ini memiliki kelebihan dan kekurangannya masing-masing. Pengujian skema *star* dan *snowflake* dengan melihat dari performa yang diujikan yaitu *Row Length*, *Response time*, *Block Size* dan *cpu time* pada penerimaan Mahasiswa baru Unsiq Jawa Tengah di Wonosobo lebih dimenangkan oleh *star schema*, jika dilihat dari eksekusi baris dan eksekusi waktu pemanggilan dan Pengujian Query dalam skenario yang berbeda akan mendapatkan hasil yang berbeda pula, skema *star* lebih memerlukan waktu pemanggilan yang singkat dibandingkan *snowflake*[5].

Data Warehouse yang dibentuk memiliki tujuan untuk melakukan analisis terkait logistik dan data transaksi perusahaan internasional yang tentu memiliki banyak komponen, sehingga Penulis menentukan skema *Star* lebih cocok dimanfaatkan untuk kasus *merger* perusahaan PMI dan Altria. Tabel yang dibentuk melalui proses ELT, akan diolah menjadi tabel dimensi yang akan membentuk tabel fakta pada skema *Star*.

IV. HASIL DAN ANALISIS

A. Data Collecting

Pada tahap *data collecting*, penulis menggunakan data laporan tahunan yang dapat diakses melalui Website official bari dari Altria atau phillip Morris International. Kumpulan data yang diambil adalah :

1. Laporan tahunan *Philip Morris International 2022*
2. Laporan tahunan Altria 2022
3. Laporan operasional *Philip Morris International 2022*
4. Laporan operasional Altria 2022
5. Laporan keuangan *Philip Morris International Q4 2022*
6. Laporan keuangan Altria 2022

Setelah itu, kumpulan dari data yang diambil dari 4 laporan tersebut, akan disusun menjadi beberapa file CSV, dan akan ditampilkan dalam bentuk tabel pada Python dengan di import menggunakan Library Pandas.

1. Proses *Data Collecting* gudang Philip Morris

```
# Read dataset gudang_phillip.csv
table_gudang_p_df = pd.read_csv('https://raw.githubusercontent.com/rhmadgiatt/Data-warehouse/main/gudang_phillip.csv')
table_gudang_p_df.head()
```

	GudangID	Negara	Kota	Kapasitas (Ton)
0	PMI-W001	Indonesia	Surabaya	25000.0
1	PMI-W002	Rusia	Moskow	20000.0
2	PMI-W003	Jerman	Hamburg	18000.0
3	PMI-W004	Italia	Napoli	15000.0
4	PMI-W005	Spanyol	Barcelona	22000.0

2. Proses *Data Collecting* gudang Altria

```
# Read dataset gudang_altria.csv
table_gudang_al_df = pd.read_csv('https://raw.githubusercontent.com/rhmadgiatt/Data-warehouse/main/gudang_altria.csv')
table_gudang_al_df.head()
```

	GudangID	Negara Bagian	Kota	Kapasitas (Ton)
0	ALT-W001	Virginia	Norfolk	12000
1	ALT-W002	Pennsylvania	Philadelphia	10000
2	ALT-W003	Illinois	Joliet	8000
3	ALT-W004	North Carolina	Greensboro	14000
4	ALT-W005	Kentucky	Lexington	11000

3. Proses *Data Collecting* Pabrik Philip Morris

```
# Read dataset pabrik_phillip.csv
table_pabrik_p_df = pd.read_csv('https://raw.githubusercontent.com/rhmadgiatt/Data-warehouse/main/pabrik_phillip.csv')
table_pabrik_p_df.head()
```

	PabrikID	Negara	Kota	KapasitasProduksi (Ton/Tahun)
0	PMI-F001	Indonesia	Karangwang	150000
1	PMI-F002	Rusia	St. Petersburg	120000
2	PMI-F003	Jerman	Berlin	100000
3	PMI-F004	Italia	Bologna	80000
4	PMI-F005	Spanyol	Madrid	90000

4. Proses *Data Collecting* Pabrik Altria

```
# Read dataset pabrik_altria.csv
table_pabrik_al_df = pd.read_csv('https://raw.githubusercontent.com/rhmadgiatt/Data-warehouse/main/pabrik_altria.csv')
table_pabrik_al_df.head()
```

PabrikID	Negara Bagian	Kota	KapasitasProduksi (Ton/Tahun)	
0	ALT-F001	Virginia	Richmond	60000
1	ALT-F002	Pennsylvania	Wilkes-Barre	50000
2	ALT-F003	Illinois	Chicago	40000
3	ALT-F004	North Carolina	Winston-Salem	55000
4	ALT-F005	Kentucky	Louisville	45000

5. Proses *Data Collecting* transaksi PMI

```
# Read dataset transaksi_phillip.csv
table_transaksi_p_df = pd.read_csv('https://raw.githubusercontent.com/rhmadgiatt/Data-warehouse/main/transaksi_phillip.csv')
table_transaksi_p_df.head()
```

TransaksiID	Tahun	Negara	JumlahProduk	TotalPenjualan (USD)	GudangID
0	PMI-T001	2022	Indonesia	25000000	PMI-W001
1	PMI-T002	2022	Rusia	18000000	PMI-W002
2	PMI-T003	2022	Jerman	12000000	PMI-W003
3	PMI-T004	2022	Italia	10000000	PMI-W004
4	PMI-T005	2022	Spanyol	15000000	PMI-W005

6. Proses *Data Collecting* transaksi Altria

```
# Read dataset transaksi_altria.csv
table_transaksi_al_df = pd.read_csv('https://raw.githubusercontent.com/rhmadgiatt/Data-warehouse/main/transaksi_altria.csv')
table_transaksi_al_df.head()
```

TransaksiID	Tahun	Negara Bagian	JumlahProduk	TotalPenjualan (USD)	GudangID
0	ALT-T001	2022	Virginia	10000000	ALT-W001
1	ALT-T002	2022	Pennsylvania	8000000	ALT-W002
2	ALT-T003	2022	Illinois	6000000	ALT-W003
3	ALT-T004	2022	North Carolina	9000000	ALT-W004
4	ALT-T005	2022	Kentucky	7000000	ALT-W005

B. Data Cleaning

Dalam studi kasus ini, proses Data Cleaning, Data Integration, dan Data Transformation dilakukan dengan menggunakan bahasa pemrograman Python dan library-library yang tersedia di dalamnya.

Dalam tahap data cleaning, beberapa langkah penting dilakukan untuk memastikan kualitas data yang akan diintegrasikan ke dalam data warehouse.

1. Penghapusan Data Duplikat

- Melakukan pengecekan data duplikat pada setiap tabel menggunakan metode `uplicated()` pada library Pandas.
- Menghapus baris-baris yang merupakan data duplikat dari setiap tabel menggunakan metode `drop_duplicates()`.

Cleaning Data

Pada tahap cleaning kita akan melakukan beberapa tahapan agar data lebih mudah dipahami dan lebih bersih untuk diolah

- Menghapus Data Duplicate

```
# Check for duplicate data table_gudang_p_df
duplicate_gudang_p = table_gudang_p_df.duplicated()

# Print how many duplicates are in table_gudang_p_df
print("Jumlah data duplikat pada tabel Gudang Philip: ", duplicate_gudang_p.sum())
```

Jumlah data duplikat pada tabel Gudang Philip: 4

```
# Delete duplicate data table_gudang_p_df
table_gudang_p_df = table_gudang_p_df.drop_duplicates()

# Success message
print("Duplicates removed successfully!")
```

Duplicates removed successfully!

```
# Print sum of duplicates
print("Jumlah data duplikat pada tabel Gudang Philip: ", table_gudang_p_df.duplicated().sum())
```

Jumlah data duplikat pada tabel Gudang Philip: 0

```
# Print how many duplicates are in table_pabrik_al_df
print("Jumlah data duplikat pada tabel pabrik Altria: ", table_pabrik_al_df.duplicated().sum())
```

Jumlah data duplikat pada tabel pabrik Altria: 5

```
# Delete duplicates
table_pabrik_al_df = table_pabrik_al_df.drop_duplicates()

# Success message
print("Duplicates removed successfully!")
```

Duplicates removed successfully!

```
# Print sum of duplicates
print("Jumlah data duplikat yang tersisa: ", table_pabrik_al_df.duplicated().sum())
```

Jumlah data duplikat yang tersisa: 0

Langkah ini dilakukan untuk menghindari redundansi data dan menjaga konsistensi informasi dalam data warehouse.

2. Penanganan Missing Value

- Melakukan pengecekan missing value atau data yang hilang pada setiap tabel menggunakan metode `isnull().sum()`.
- Menghapus baris-baris yang mengandung missing value dari setiap tabel menggunakan metode `dropna()`.

```
• Missing value

Untuk mengatasi missing value baris dari data akan di drop

# Check missing values in table_gudang_p_df
missing_values = table_gudang_p_df.isnull().sum()

# Print missing values in table_gudang_p_df
print("Jumlah data yang hilang pada tabel Gudang Philip: ", missing_values.sum())

Jumlah data yang hilang pada tabel Gudang Philip: 8

# Drop missing values
table_gudang_p_df = table_gudang_p_df.dropna()

# Success message
print("Data hilang berhasil dihapus!")

Data hilang berhasil dihapus!

# Print sum of missing values
print("Jumlah data yang hilang yang tersisa: ", table_gudang_p_df.isnull().sum().sum())

Jumlah data yang hilang yang tersisa: 0
```

Langkah ini dipilih untuk menjaga integritas data dan menghindari analisis yang bias akibat data tidak lengkap.

3. Penyeragaman Nama Kolom

- Mengubah nama kolom pada tabel tertentu agar seragam dengan menggunakan metode `rename()`.

Dalam kasus ini, nama kolom 'Negara Bagian' pada tabel Gudang, Pabrik, dan Transaksi Altria diubah menjadi 'Negara' agar sesuai dengan nama kolom pada tabel Philip Morris International.

```
• Mengubah nama kolom pada tabel Gudang, Pabrik, dan Transaksi

# Rename column 'Negara Bagian' to 'Negara'
table_gudang_al_df = table_gudang_al_df.rename(columns={'Negara Bagian': 'Negara'})
table_pabrik_al_df = table_pabrik_al_df.rename(columns={'Negara Bagian': 'Negara'})
table_transaksi_al_df = table_transaksi_al_df.rename(columns={'Negara Bagian': 'Negara'})

# Success message
print("Nama kolom berhasil diubah!")

Nama kolom berhasil diubah!
```

Penyeragaman nama kolom memudahkan proses integrasi data dari sumber yang berbeda.

C. Data Integration

Setelah proses cleaning data selesai, langkah selanjutnya adalah melakukan integrasi data dari berbagai sumber. Dalam studi kasus ini, data dari Philip Morris International dan Altria Group, Inc.

Melakukan penggabungan data dari Philip Morris International dan Altria Group, Inc. untuk tabel Gudang, Pabrik, dan Transaksi menggunakan operasi `concat()` pada library Pandas.

Hasil dari proses ini adalah tiga tabel baru yang mencakup data dari kedua perusahaan, yaitu `gudang_df`, `pabrik_df`, dan `transaksi_df`.

Integrasi data ini penting untuk mendapatkan gambaran yang utuh dari operasi gabungan kedua perusahaan setelah merger.

```
Integrasi Data

• Menggabungkan Dataset untuk Tabel Gudang, Pabrik, dan Transaksi

# Merge table_gudang_p_df and table_gudang_al_df
gudang_df = pd.concat([table_gudang_p_df, table_gudang_al_df])

# Merge table_pabrik_p_df and table_pabrik_al_df
pabrik_df = pd.concat([table_pabrik_p_df, table_pabrik_al_df])

# Merge table_transaksi_p_df and table_transaksi_al_df
transaksi_df = pd.concat([table_transaksi_p_df, table_transaksi_al_df])

# Success message
print("Data berhasil digabungkan!")

Data berhasil digabungkan!

# Show merged data table_gudang_p_df and table_gudang_al_df
print(gudang_df)

   GudangID      Negara      Kota  Kapasitas (Ton)
0  PMI-W001  Indonesia  Surabaya    25000.0
1  PMI-W002      Rusia  Moscow    20000.0
2  PMI-W003     Jerman  Hamburg    18000.0
3  PMI-W004     Italia  Napoli    15000.0
4  PMI-W005     Spanyol  Barcelona  22000.0
..      ...
95 ALT-W096  West Virginia  Huntington     70.0
96 ALT-W097  South Carolina  Greenville     60.0
97 ALT-W098     Hawaii  Pearl City     50.0
98 ALT-W099     Alaska  Fairbanks     40.0
99 ALT-W100  District of Columbia  Washington D.C.     30.0

[186 rows x 4 columns]
```

```
# Show merged data table_pabrik_p_df and table_pabrik_al_df
print(pabrik_df)

   PabrikID      Negara      Kota  KapasitasProduksi (Ton/Tahun)
0  PMI-F001  Indonesia  Karawang    150000
1  PMI-F002      Rusia  St. Petersburg    120000
2  PMI-F003     Jerman  Berlin    100000
3  PMI-F004     Italia  Bologna    80000
4  PMI-F005     Spanyol  Madrid    90000
..      ...
98 ALT-F096  Delaware  Dover     40
99 ALT-F097  West Virginia  Huntington     30
100 ALT-F098  South Carolina  Greenville     20
101 ALT-F099     Hawaii  Pearl City     15
102 ALT-F100     Alaska  Fairbanks     10

[199 rows x 4 columns]

# Show merged data table_transaksi_p_df and table_transaksi_al_df
print(transaksi_df)

   TransaksiID  Tahun      Negara  JumlahProduk  \
0  PMI-T001    2022  Indonesia    2500000
1  PMI-T002    2022      Rusia    1800000
2  PMI-T003    2022     Jerman    1200000
3  PMI-T004    2022     Italia    1000000
4  PMI-T005    2022     Spanyol    1500000
..      ...
95  ALT-T096    2022  West Virginia     7000
96  ALT-T097    2022  South Carolina     6000
97  ALT-T098    2022     Hawaii     5000
98  ALT-T099    2022     Alaska     4000
99  ALT-T100    2022  District of Columbia     3000

   TotalPenjualan (USD)  GudangID
0    2500000000  PMI-W001
1    1800000000  PMI-W002
2    1200000000  PMI-W003
3    1000000000  PMI-W004
4    1500000000  PMI-W005
..      ...
95     700000  ALT-W096
96     600000  ALT-W097
97     500000  ALT-W098
98     400000  ALT-W099
99     300000  ALT-W100

[200 rows x 6 columns]
```

D. Data Transformation

1. Pencarian Transaksi Terbesar dan Terkecil di Gudang
 - Melakukan merge antara tabel `transaksi_df` dan `gudang_df` berdasarkan kolom `GudangID` menggunakan metode `merge()`.

- Mencari transaksi terbesar dengan melihat nilai maksimum pada kolom JumlahProduk dari data hasil merge.
- Mencari transaksi terkecil dengan melihat nilai minimum pada kolom JumlahProduk dari data hasil merge.

```

Transformasi Data
• Mencari Tahu Transaksi Terbesar dan Terkecil di Gudang

# Merge the transaction and warehouse data based on GudangID
merged_data = pd.merge(transaksi_df, gudang_df, on='GudangID')

# Find the Largest transaction
largest_transaction = merged_data[merged_data['JumlahProduk'] == merged_data['JumlahProduk'].max()]

# Print the largest transaction
print("Transaksi Terbesar: \n\n", largest_transaction)

Transaksi Terbesar:
  TransaksiID Tahun Negara_x JumlahProduk TotalPenjualan (USD) \
68 PHI-T073 2022 Amerika Serikat 3000000 3000000000

  GudangID Negara_y Kota Kapasitas (Ton)
68 PHI-W073 Amerika Serikat Chicago 25000.0

# Find the fewest transactions
fewest_transaction = merged_data[merged_data['JumlahProduk'] == merged_data['JumlahProduk'].min()]

# Print fewest transaction
print("Transaksi Terkecil: \n\n", fewest_transaction)

Transaksi Terkecil:
  TransaksiID Tahun Negara_x JumlahProduk \
184 ALT-T100 2022 District of Columbia 3000

  TotalPenjualan (USD) GudangID Negara_y Kota \
184 300000 ALT-W100 District of Columbia Washington D.C.

  Kapasitas (Ton)
184 30.0

```

2. Pencarian Kapasitas Gudang Terbesar, Terkecil, dan Rata-rata

- Mencari kapasitas gudang terbesar dengan menggunakan metode max() pada kolom Kapasitas (Ton) dari tabel gudang_df.
- Mencari kapasitas gudang terkecil dengan menggunakan metode min() pada kolom Kapasitas (Ton) dari tabel gudang_df.
- Mencari kapasitas gudang rata-rata dengan menggunakan metode mean() pada kolom Kapasitas (Ton) dari tabel gudang_df.

```

• Mencari Tahu Kapasitas Terbesar dan Terkecil Gudang

# Find largest warehouse capacity
kapasitas_terbesar = gudang_df["Kapasitas (Ton)"].max()

# Find smallest warehouse capacity
kapasitas_terkecil = gudang_df["Kapasitas (Ton)"].min()

# Find the average warehouse capacity
kapasitas_rata_rata = gudang_df["Kapasitas (Ton)"].mean()

print("Kapasitas Gudang Terbesar:", kapasitas_terbesar)
print("Kapasitas Gudang Terkecil:", kapasitas_terkecil)
print("Kapasitas Gudang Rata-rata:", kapasitas_rata_rata)

Kapasitas Gudang Terbesar: 25000.0
Kapasitas Gudang Terkecil: 30.0
Kapasitas Gudang Rata-rata: 7800.725806451613

```

3. Pencarian Jumlah Produksi Terbesar, Terkecil, dan Rata-rata di Pabrik

- Mencari jumlah produksi terbesar dengan menggunakan metode max() pada kolom

KapasitasProduksi (Ton/Tahun) dari tabel pabrik_df.

- Mencari jumlah produksi terkecil dengan menggunakan metode min() pada kolom KapasitasProduksi (Ton/Tahun) dari tabel pabrik_df.
- Mencari rata-rata jumlah produksi dengan menggunakan metode mean() pada kolom KapasitasProduksi (Ton/Tahun) dari tabel pabrik_df.

```

• Mencari Tahu Jumlah Produksi Terbesar, Terkecil, dan Rata-rata di Pabrik

# Find the Largest production
produksi_terbesar = jumlah_produksi.max()

# Find the smallest production
produksi_terkecil = jumlah_produksi.min()

# Find the average production
rata_rata_produksi = jumlah_produksi.mean()

print("Jumlah Produksi Terbesar:", produksi_terbesar)
print("Jumlah Produksi Terkecil:", produksi_terkecil)
print("Rata-rata Jumlah Produksi:", rata_rata_produksi)

Jumlah Produksi Terbesar: 150000
Jumlah Produksi Terkecil: 10
Rata-rata Jumlah Produksi: 30558.442211055277

```

4. Konversi Satuan Ton menjadi Kilogram

- Menambahkan kolom baru Kapasitas (KG) pada tabel gudang_df dengan mengonversi nilai pada kolom Kapasitas (Ton) menjadi kilogram (dikalikan dengan 1000).
- Menambahkan kolom baru KapasitasProduksi (KG/Tahun) pada tabel pabrik_df dengan mengonversi nilai pada kolom KapasitasProduksi (Ton/Tahun) menjadi kilogram (dikalikan dengan 1000).

- Konversi Satuan Ton Menjadi KG

```
# Convert Ton to KG in the warehouse table
gudang_df['Kapasitas (KG)'] = gudang_df['Kapasitas (Ton)'] * 1000

# Convert Ton to KG in the factory table
pabrik_df['KapasitasProduksi (KG/Tahun)'] = pabrik_df['KapasitasProduksi (Ton/Tahun)'] * 1000

# Print the updated data
print(" Data Hasil Konversi Gudang : ""\n", gudang_df)
print(" Data Hasil Konversi Pabrik : ""\n", pabrik_df)
```

```
Data Hasil Konversi Gudang :
GudangID      Negara      Kota      Kapasitas (Ton) \
0  PMI-W001      Indonesia  Surabaya  25000.0
1  PMI-W002      Rusia      Moscow    20000.0
2  PMI-W003      Jerman     Hamburg   18000.0
3  PMI-W004      Italia     Napoli    15000.0
4  PMI-W005      Spanyol   Barcelona 22000.0
..          ...
95 ALT-W096      West Virginia  Huntington 70.0
96 ALT-W097      South Carolina  Greenville 60.0
97 ALT-W098      Hawaii      Pearl City 50.0
98 ALT-W099      Alaska      Fairbanks 40.0
99 ALT-W100      District of Columbia  Washington D.C. 30.0
```

```
Kapasitas (KG)
0      25000000.0
1      20000000.0
2      18000000.0
3      15000000.0
4      22000000.0
..          ...
95      70000.0
96      60000.0
97      50000.0
98      40000.0
99      30000.0
```

[186 rows x 5 columns]

```
Data Hasil Konversi Pabrik :
PabrikID      Negara      Kota      KapasitasProduksi (Ton/Tahun) \
0  PMI-F001      Indonesia  Karawang  150000
1  PMI-F002      Rusia      St. Petersburg 120000
2  PMI-F003      Jerman     Berlin    100000
3  PMI-F004      Italia     Bologna   80000
4  PMI-F005      Spanyol   Madrid    90000
..          ...
98 ALT-F096      Delaware  Dover     40
99 ALT-F097      West Virginia  Huntington 30
100 ALT-F098      South Carolina  Greenville 20
101 ALT-F099      Hawaii      Pearl City 15
102 ALT-F100      Alaska      Fairbanks 10
```

```
KapasitasProduksi (KG/Tahun)
0      150000000
1      120000000
2      100000000
3      80000000
4      90000000
..          ...
98      40000
99      30000
100     20000
101     15000
102     10000
```

[199 rows x 5 columns]

5. Konversi Mata Uang (USD ke IDR) pada Tabel Transaksi

- Mendefinisikan nilai tukar mata uang dari USD ke IDR.
- Menambahkan kolom baru TotalPenjualan (IDR) pada tabel transaksi_df dengan mengonversi nilai pada kolom TotalPenjualan (USD) menjadi IDR (dikalikan dengan nilai tukar yang telah didefinisikan).

- Konversi Mata Uang (USD ke IDR) Pada Tabel Transaksi

```
# Define the exchange rate from USD to IDR
exchange_rate = 16040

# Convert the 'TotalPenjualan (USD)' column to IDR
transaksi_df['TotalPenjualan (IDR)'] = transaksi_df['TotalPenjualan (USD)'] * exchange_rate

# Display the updated table
print(transaksi_df)
```

```
TransaksiID Tahun      Negara      JumlahProduk \
0  PMI-T001  2022      Indonesia  2500000
1  PMI-T002  2022      Rusia      1800000
2  PMI-T003  2022      Jerman     1200000
3  PMI-T004  2022      Italia     1000000
4  PMI-T005  2022      Spanyol   1500000
..          ...
95 ALT-T096  2022      West Virginia  7000
96 ALT-T097  2022      South Carolina  6000
97 ALT-T098  2022      Hawaii      5000
98 ALT-T099  2022      Alaska      4000
99 ALT-T100  2022      District of Columbia  3000
```

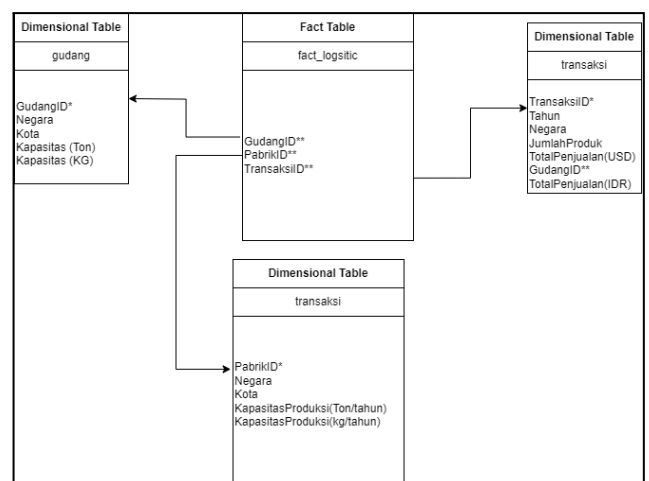
```
TotalPenjualan (USD) GudangID TotalPenjualan (IDR)
0      2500000000 PMI-W001  4010000000000
1      1800000000 PMI-W002  2887200000000
2      1200000000 PMI-W003  1924800000000
3      1000000000 PMI-W004  1604000000000
4      1500000000 PMI-W005  2406000000000
..          ...
95      7000000 ALT-W096  11228000000
96      6000000 ALT-W097  9624000000
97      5000000 ALT-W098  8020000000
98      4000000 ALT-W099  6416000000
99      3000000 ALT-W100  4812000000
```

[200 rows x 7 columns]

Langkah-langkah di atas dilakukan untuk memastikan kualitas data, integrasi data dari sumber yang berbeda, serta transformasi data sesuai dengan kebutuhan analisis dan pelaporan dalam studi kasus pembangunan data warehouse hasil merger Philip Morris International dan Altria Group, Inc.

E. Skema Data Warehouse

Setelah selesai dilakukan proses *Data Transformation*, maka selanjutnya akan dibuat proses pembentukan skema Data Warehouse, yakni Star. Terdapat 3 tabel dimensional yang digunakan untuk membentuk tabel fakta yang memberikan informasi terkait logistik dan pengaruhnya akan transaksi. Yakni tabel fakta gudang, pabrik, dan transaksi.



V. KESIMPULAN

Dalam kesimpulan jurnal ini, dapat disimpulkan bahwa pemanfaatan Data Warehouse memiliki dampak positif yang signifikan bagi perusahaan, terutama dalam mendukung pengambilan keputusan dan analisis keadaan organisasi melalui Business Intelligence. Melalui pembentukan Skema Star dalam Data Warehouse untuk Philip Morris International dan Altria setelah merger pada tahun 2019, diharapkan perusahaan dapat memperoleh pemahaman yang mendalam mengenai transaksi, kapasitas gudang, dan produksi di berbagai lokasi pabrik. Dengan demikian, pengetahuan yang dihasilkan dari analisis data ini diharapkan dapat meningkatkan efisiensi dalam logistik, penyebaran produk, dan penjualan bagi Philip Morris International. Penelitian ini menunjukkan pentingnya implementasi teknologi informasi, seperti Data Warehouse, dalam mendukung strategi bisnis dan pengambilan keputusan di era yang semakin digital ini.

REFERENSI

- [1] Han, J. and Kamber, M. (2006) Data Mining: Concepts and Techniques. 2nd Edition, Morgan Kaufmann Publishers, San Francisco.
- [2] Zai, C. (2022). Implementasi Data Mining Sebagai Pengolahan Data. *Jurnal Portal Data*, 2(3).
- [3] Saputro, I. W., & Sari, B. W. (2020). Uji Performa Algoritma Naïve Bayes untuk Prediksi Masa Studi Mahasiswa. *Creative Information Technology Journal*, 6(1), 1-11.
- [4] Dini Nurmalasari, M. S. Z., & Hanifah, P. (2019). Analisis kinerja pemodelan data star schema pada data perpustakaan. *Jurnal Komputer Terapan Vol*, 5(2), 44-53.
- [5] Khomsatun, K., Kusriani, K., & Suyoto, A. (2020). Perbandingan Model Skema Star Dan Snowflake Data Akademik (Studi Kasus UNSIQ Jawa Tengah Di Wonosobo). *Jurnal Informa: Jurnal Penelitian dan Pengabdian Masyarakat*, 6(2), 7-15.