

DATA ENGINEER

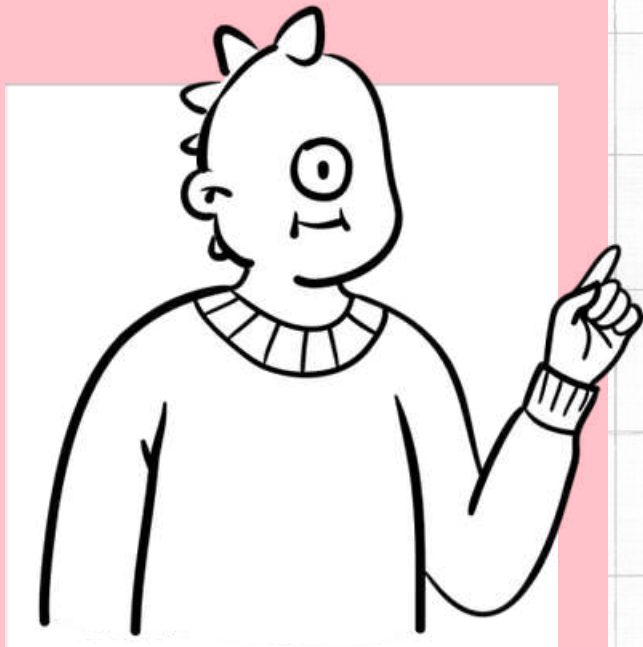
Learning Progress Review (Week 3)

Git & GitHub, Data Warehouse, Business Intelligence

Oleh : MIT TEAM

Fast Track Data Engineer Scholarship Batch 1 DigitalSkola
Batch 1

MEET OUR TEAM MIT



**MIFTAHUL
ARIFIN**

[www.linkedin.com/in/
miftah-arif](https://www.linkedin.com/in/miftah-arif)



MEITY KUMALASARI

[https://www.linkedin.co
m/in/meity-kumalasari-
b27603100/](https://www.linkedin.com/in/meity-kumalasari-b27603100/)



**MICHAEL
WIRYASEPUTRA**

[https://www.linkedin.co
m/in/michael-
wiryaseputra](https://www.linkedin.com/in/michael-wiryaseputra)



MARCEL KURNIAWAN

[https://www.linkedin.co
m/in/marcelkurniawan](https://www.linkedin.com/in/marcelkurniawan)



**MAULANA AHMAD
MALIKI**

[https://www.linkedin.co
m/in/lanaahm/](https://www.linkedin.com/in/lanaahm/)

Interested in knowing more about us?

Please Insert the link below the name into your browser

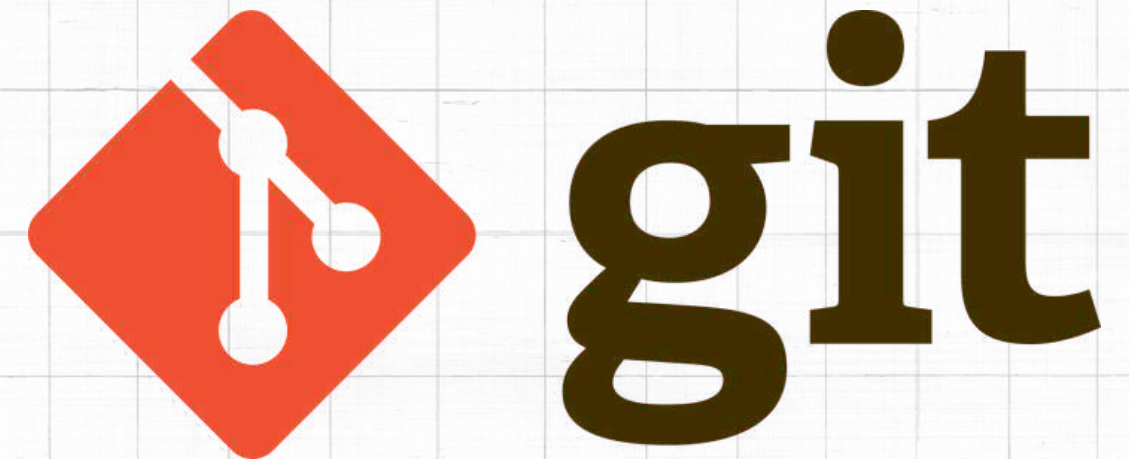


GIT & GITHUB

- Git Perkenalan & Tujuan Pemakaian Git
- Terminologi Git
- GitHub
- Membuat Repository
- Perintah Dasar Git
- Branching
- Conflict

GIT

- Git merupakan software berbasis **Version Control System (VCS)** yang bertugas untuk **mencatat perubahan seluruh file** atau **repository** suatu project.
- Developer software biasa menggunakan Git untuk **distributed revision** (VCS terdistribusi), hal ini bertujuan untuk **menyimpan database tidak hanya ke satu tempat**. Namun semua orang yang terlibat dalam penyusunan kode dapat menyimpan database ini.

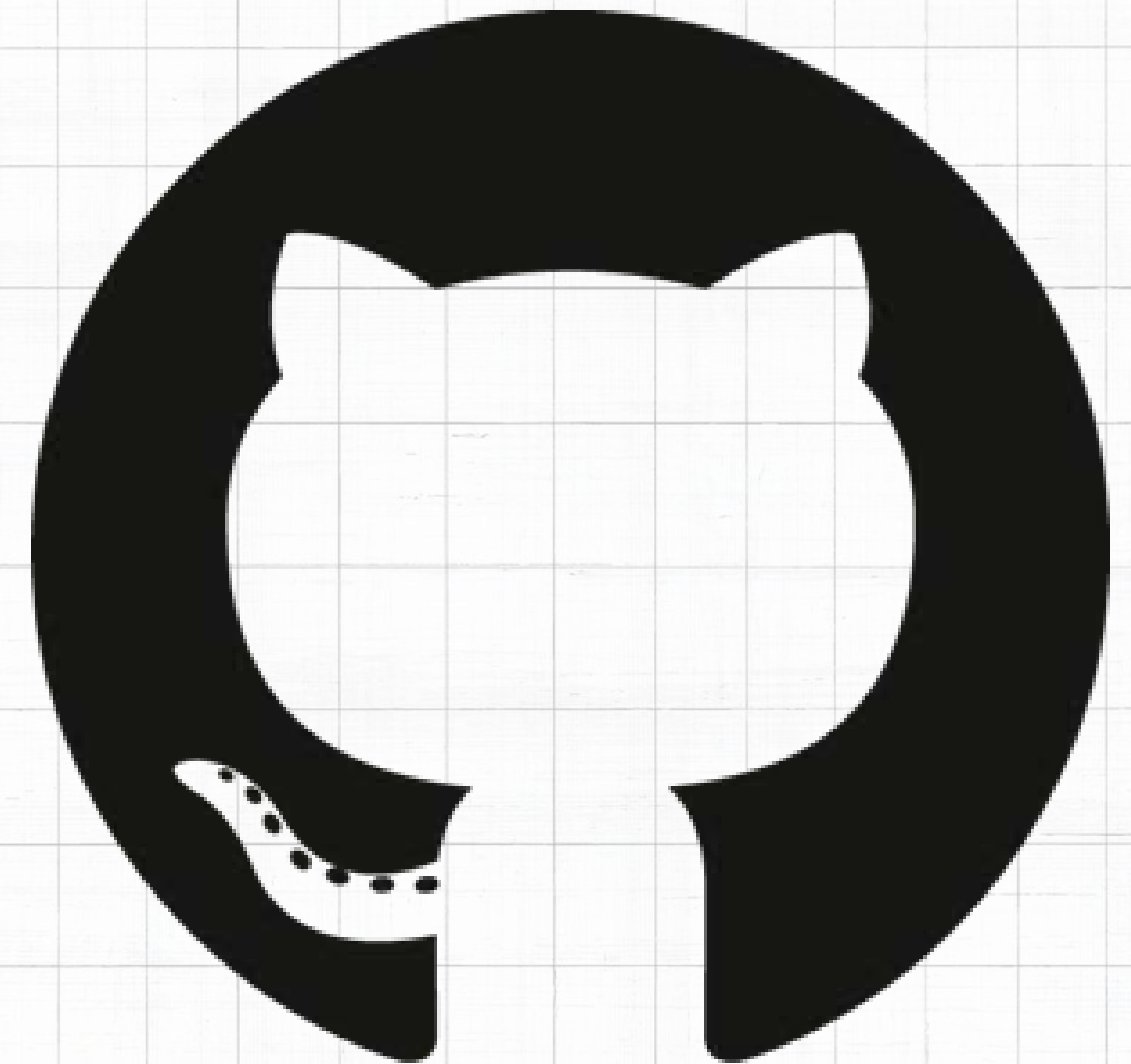


Terminology / Istilah GIT

- **Repository (Repo):** Tempat penyimpanan semua file dan riwayat perubahan dari proyek (local repository & remote repository).
- **Commit:** Rekaman perubahan yang dibuat ke file dalam repository.
- **Branch:** Salinan independen dari kode proyek yang memungkinkan pengembang bekerja pada fitur atau perbaikan baru tanpa mengganggu kode utama. Branch default biasanya disebut "main" atau "master".
- **Clone:** Proses membuat salinan lokal dari repository jarak jauh sehingga pengembang bisa bekerja dengan kode tersebut di komputer mereka sendiri.
- **Pull:** Perintah untuk mengambil pembaruan dari repository jarak jauh dan langsung menggabungkannya ke branch lokal.
- **Push:** Perintah untuk mengirimkan perubahan yang telah dikomit dari branch lokal ke branch di repository jarak jauh.
- **Fetch:** Perintah untuk mengambil pembaruan dari repository jarak jauh ke repository lokal tanpa menggabungkannya secara otomatis.
- **Snapshot:** Keadaan atau gambar keseluruhan dari semua file dalam repository pada saat tertentu.
- **Staged:** Perubahan yang telah dipilih untuk dimasukkan dalam commit berikutnya.

GitHub

- GitHub merupakan **manajemen project, sistem versioning code**, sekaligus platform **jaringan sosial** bagi para **developer** seluruh dunia.
- Github memungkinkan untuk programmer melakukan **penyelesaian proyek secara kolaborasi**.



Membuat Repositori

Dalam membuat repositori, kita dapat menggunakan dua cara yaitu melalui **platform github.com** ataupun menggunakan **git command melalui CLI**. Untuk cara yang kedua ini dilakukan dengan cara menghubungkan repository yang sudah ada di local ke github.

Melalui Platform github.com

1. Login ke akun **GitHub** yang sesuai.
2. Klik tombol "+" di pojok kanan atas, lalu pilih "**New repository**".
3. Isi nama **repository** (misalnya, my-repo).
4. Tambahkan deskripsi jika diinginkan (**opsional**).
5. Pilih visibilitas repository: **Public** atau **Private**.
6. Pilih opsi untuk menginisialisasi repository dengan **README** jika diinginkan.
7. Klik "**Create repository**".

Melalui git command (git init)

1. Menghubungkan **git repository** yang Sudah ada di **local** ke **GitHub**. (`git add remote origin "https://github.com/user_name/dari-local.git"`)
2. Cek dengan "**git remote -v**"

Perintah dasar git

```
git init : Untuk memulai proyek baru dengan Git

git clone url-repositori : Untuk mengklon repositori yang sudah ada

git status : Untuk melihat status perubahan pada repositori

git add nama-file : Untuk menambahkan file atau perubahan ke staging area

git commit -m "Pesan commit" : Untuk menyimpan perubahan ke dalam repositori

git log : Untuk melihat riwayat commit di repositori

git branch nama-cabang : Untuk membuat cabang baru:

git checkout nama-cabang : Untuk berpindah ke cabang lain

git merge nama-cabang : Untuk menggabungkan cabang lain ke cabang saat ini

git branch -d nama-cabang : Untuk menghapus cabang yang sudah tidak diperlukan

git push origin nama-cabang : Untuk mengirim perubahan ke repositori remote

git pull : Untuk mengambil perubahan terbaru dari repositori remote

git diff : Untuk melihat perbedaan antara perubahan saat ini dan commit terakhir
```


Git Branching



```
# membuat branch
git branch database_department
# memeriksa branch aktif
git branch
# memindahkan branch dalam git
git checkout database_department
# push branch
git push -u origin database_department
```

Git Branch berfungsi untuk menambahkan fitur baru, memperbaiki bugs, mengerjakan proyek secara bersamaan dengan pengguna lain tanpa harus bertabrakan & mengganggu proses pengerjaan masing masing

Git Conflict

```
● ● ●  
  
# buat file example.txt di branch utama (master)  
# buat perubahan file example.txt di branch develop  
  
# melakukan checkout di branch master  
git checkout master  
  
# merge dari branch develop ke branch master  
git merge develop  
  
# Output (Conflict)  
Auto-merging example.txt  
CONFLICT (content): Merge conflict in example.txt  
Automatic merge failed; fix conflicts and then commit the  
result.
```

Git Conflict terjadi ketika dua atau lebih perubahan yang bertentangan dilakukan pada file yang sama di dua cabang yang berbeda. Ini bisa terjadi ketika dua atau lebih pengembang melakukan perubahan pada bagian yang sama dari file atau ketika pengembang mencoba untuk menggabungkan dua cabang yang memiliki perubahan yang berbeda pada bagian yang sama dari file



DATA WAREHOUSE

- Pengenalan Data Warehouse
- Data Warehouse Architecture
- Data Warehouse Component

Data Warehouse

Data warehouse adalah sistem penyimpanan data yang dirancang untuk mengumpulkan, menyimpan, dan mengelola data dari berbagai sumber yang berbeda untuk analisis dan pelaporan bisnis.

Beberapa komponen basic yang dapat kita temukan dalam data warehouse :

1. Data Source

Sumber data adalah tempat asal data yang akan dimasukkan ke dalam data warehouse. Sumber data ini bisa berasal dari berbagai sistem operasional seperti **Database Transaksional, File Flat, dan Data Eksternal**.

2. ETL component

- **Extract:** Pengambilan data dari berbagai sumber data.
- **Transform:** Proses pembersihan dan pengolahan data untuk memastikan konsistensi, integritas, dan kualitas data. Ini termasuk mengubah format data, menghapus duplikasi, dan melakukan perhitungan yang diperlukan.
- **Load:** Memasukkan data yang telah diolah ke dalam database data warehouse.

Data Warehouse

3. Physical target database

Physical database atau basis data fisik adalah representasi fisik dari data yang disimpan di dalam sistem manajemen basis data (DBMS). Ini mencakup semua struktur penyimpanan yang digunakan untuk menyimpan data, termasuk file, blok, halaman, dan lokasi memori yang digunakan oleh DBMS untuk mengelola data.

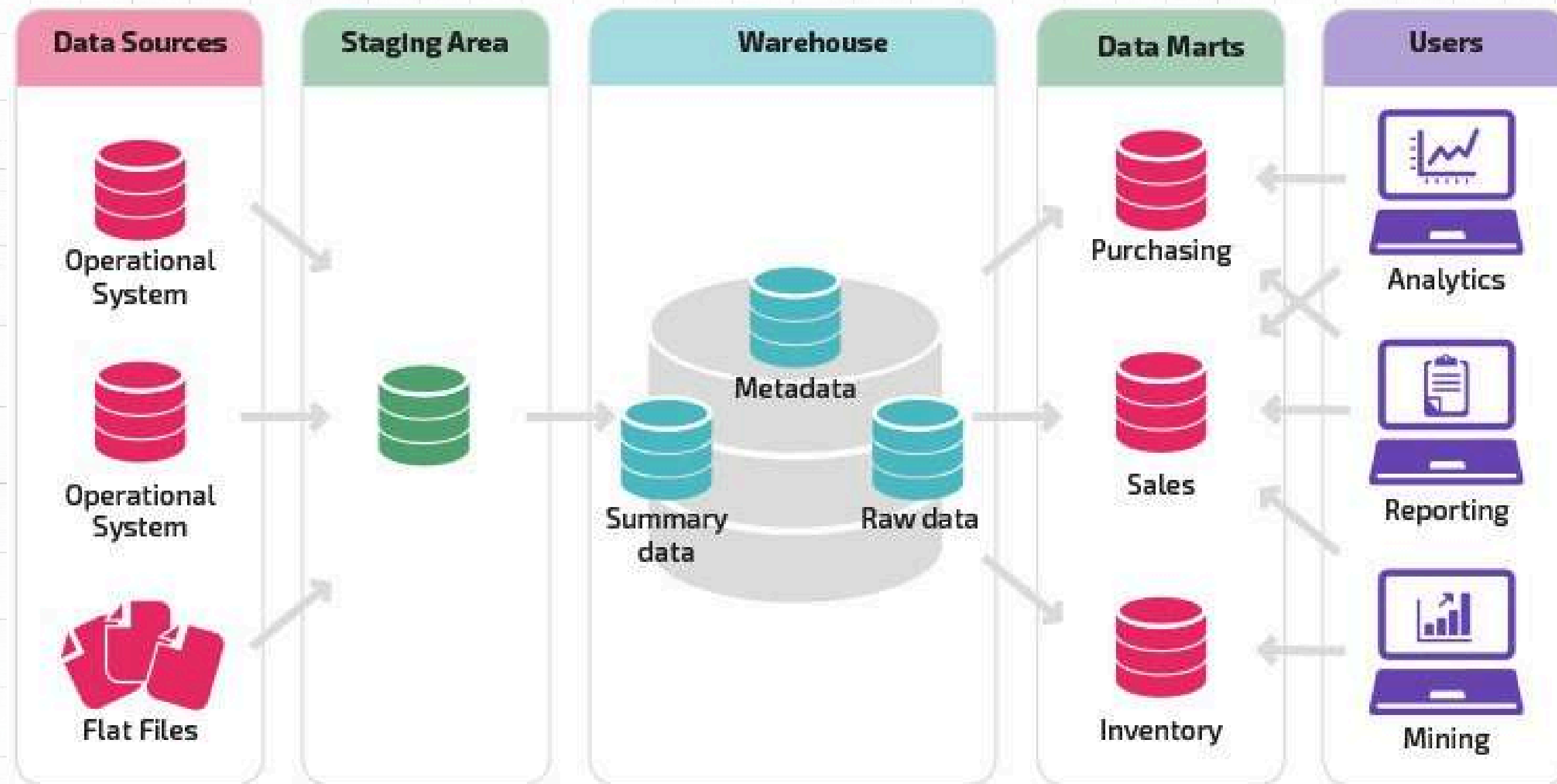
4. Data Warehouse Schema Scripts

adalah sekumpulan instruksi atau perintah yang ditulis dalam bahasa pemrograman database (seperti SQL) untuk membuat dan mengatur struktur skema dalam data warehouse.

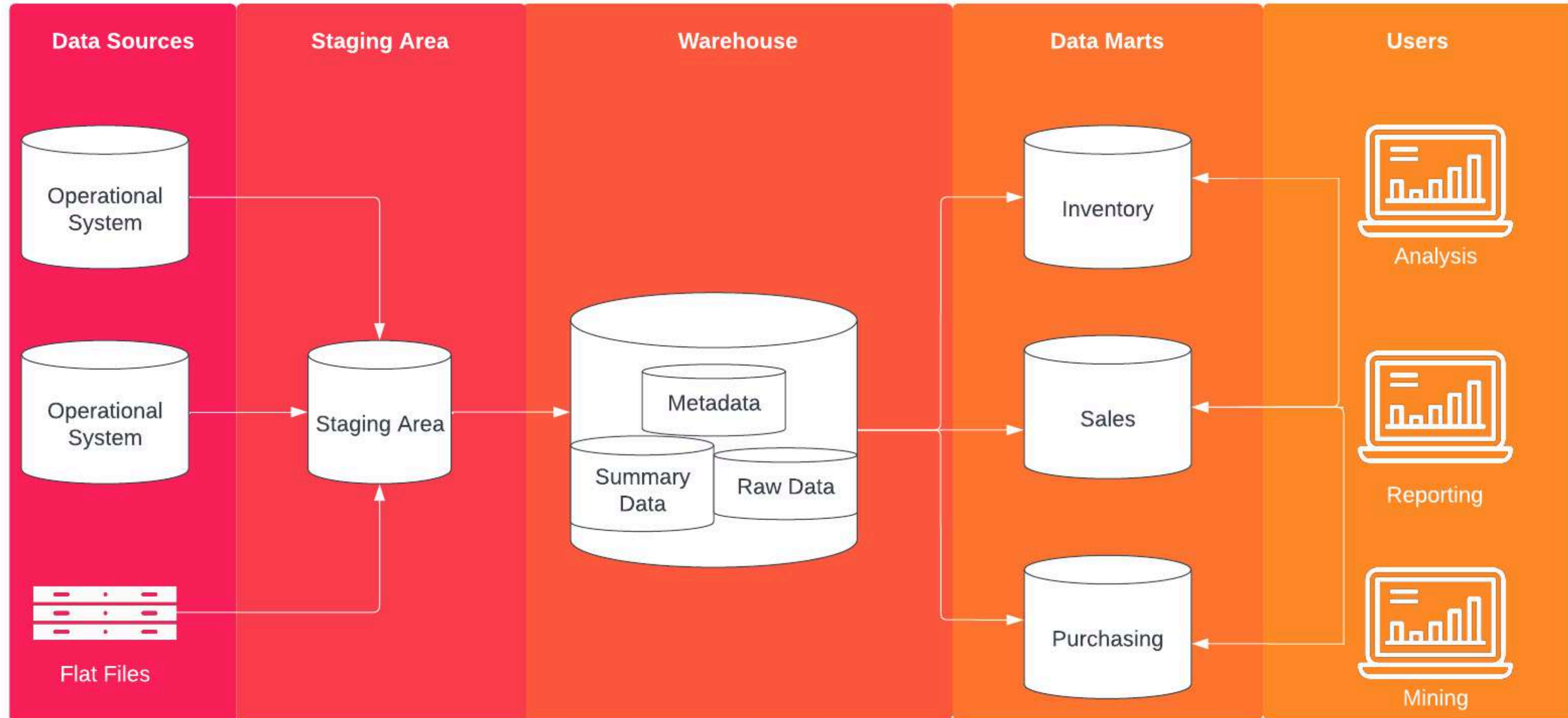
5. Metadata

Metadata adalah data yang memberikan informasi tentang data lainnya. Ini adalah deskripsi atau definisi tentang data, struktur data, konten, dan cara data tersebut disimpan, diatur, dan dikelola dalam suatu sistem.

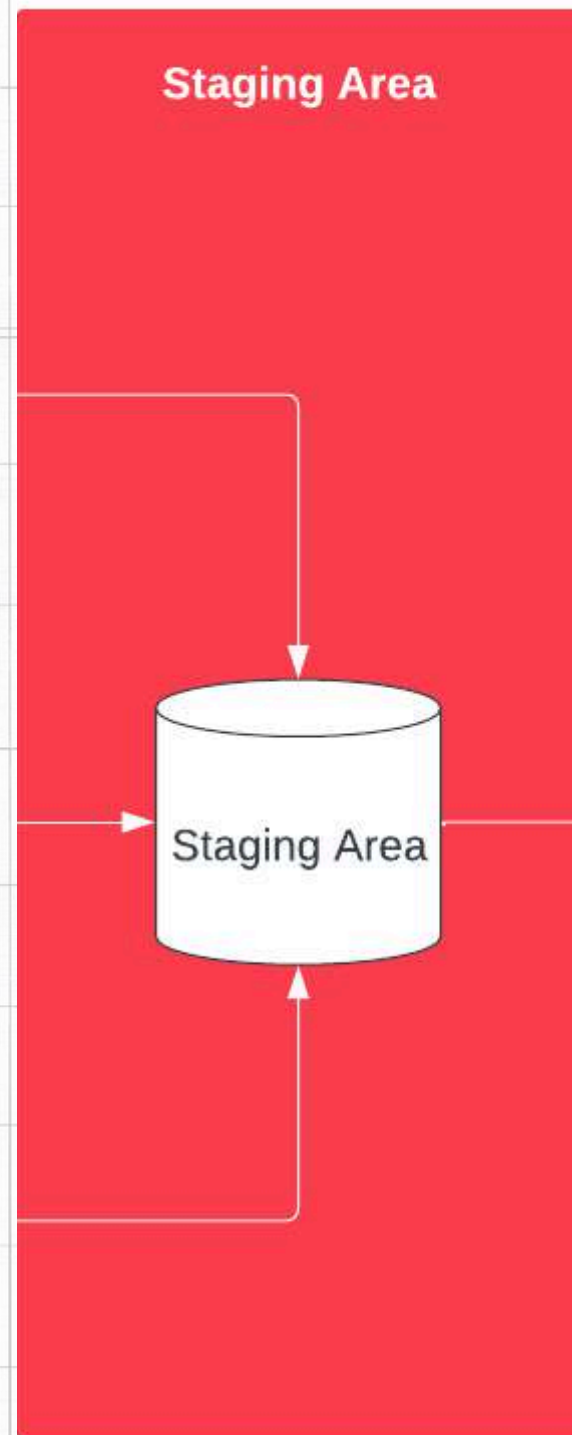
Data Warehouse Architecture



Data Warehouse Component



Data Warehouse Component



Staging Area

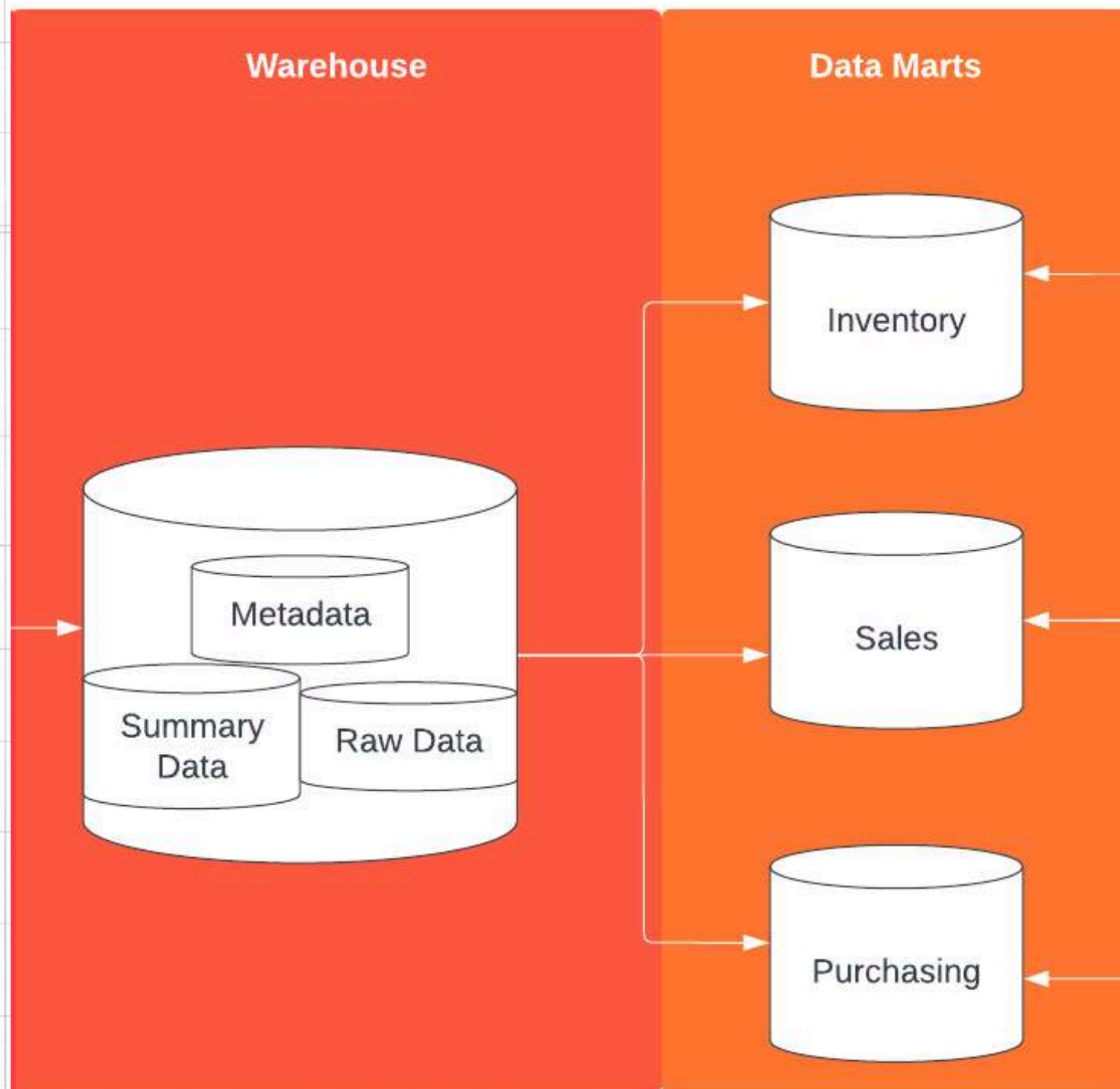
Staging area adalah tempat penyimpanan sementara data sebelum masuk ke dalam data warehouse. Di sini, data yang diekstraksi dari berbagai sumber diubah dan dibersihkan.

Proses ETL yang terjadi di staging area mencakup :

- Data Validation**
- Data Cleaning**
- Data Transforming**
- Data Aggregating**
- Data Loading**

Staging Area termasuk ke dalam Data Lake

Data Warehouse Component



Warehouse dan Data Mart

Ini adalah tempat utama penyimpanan data dalam data warehouse. Database ini biasanya dioptimalkan untuk query dan analisis, bukan untuk transaksi operasional. Data yang disimpan dalam data warehouse bersifat terstruktur dan disusun dalam bentuk skema yang memungkinkan akses yang cepat dan efisien.

Data mart adalah subset dari data warehouse yang disesuaikan untuk kebutuhan spesifik departemen atau unit bisnis tertentu, seperti penjualan, pemasaran, atau keuangan. Data mart mempermudah akses data bagi pengguna yang memerlukan analisis khusus tanpa harus mengakses seluruh data warehouse.

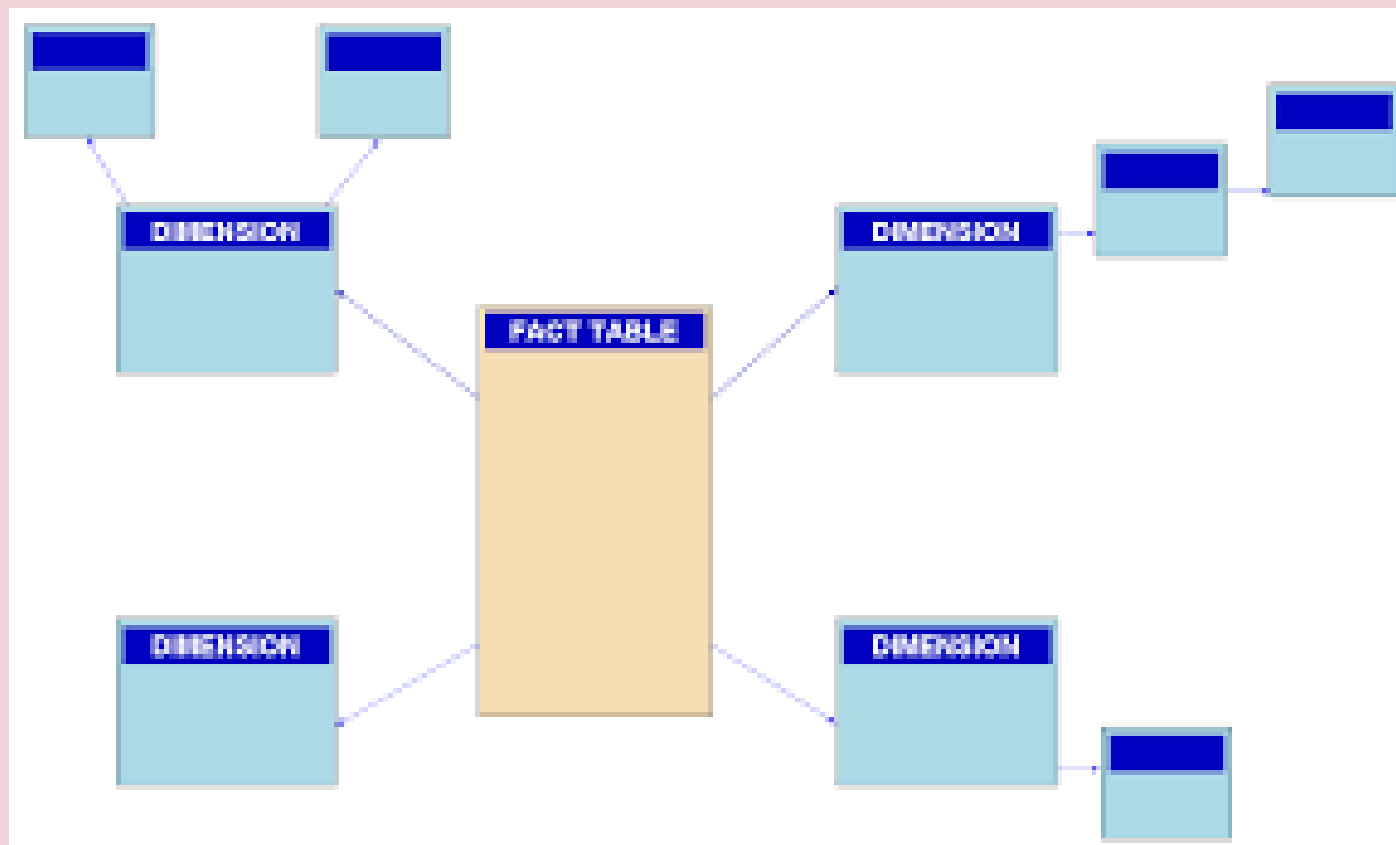
Data Warehouse Component



Users and Analytics Tools

Ini adalah alat-alat yang digunakan untuk mengakses, menganalisis, dan melaporkan data dari data warehouse. Beberapa contoh alat tersebut meliputi:

- SQL Query Tools: Alat untuk menulis dan menjalankan query SQL.
- Business Intelligence (BI) Tools: Alat untuk membuat laporan, dasbor, dan visualisasi data, seperti Tableau, Power BI, dan Google Data Studio.
- OLAP (Online Analytical Processing): Alat untuk analisis multidimensi yang memungkinkan pengguna melakukan analisis data yang kompleks secara interaktif.



DATA Model of Data Warehouse

- Data Warehouse Schema
- Fact Table and Dimension Table
- Star Schema and Snowflake Schema
- Case Study

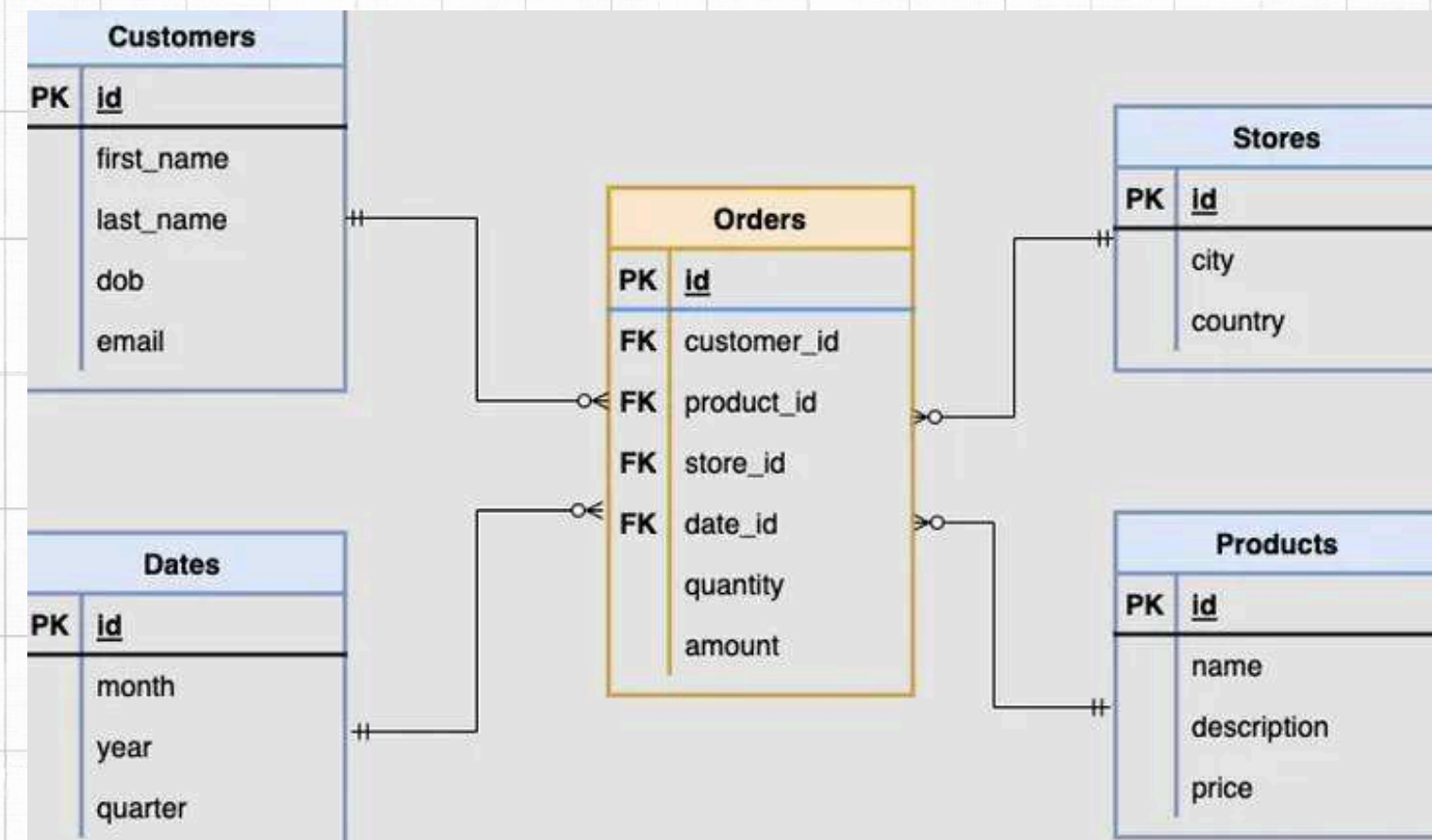
Data Warehouse Schema

Untuk membuat data warehouse, terlebih dahulu kita perlu mengelompokkan data tersebut menjadi 3 komponen, yaitu:

- **Metadata:** membantu dalam mengelola, melacak, dan memonitor data dengan lebih efisien
- **Raw Data:** Data 'murni' hasil tangkapan dari sensor digital yang sama sekali belum disentuh oleh kompresi atau pun interpolasi apapun
- **Summary Data:** sekumpulan ringkasan sejumlah data pada sebuah item barang

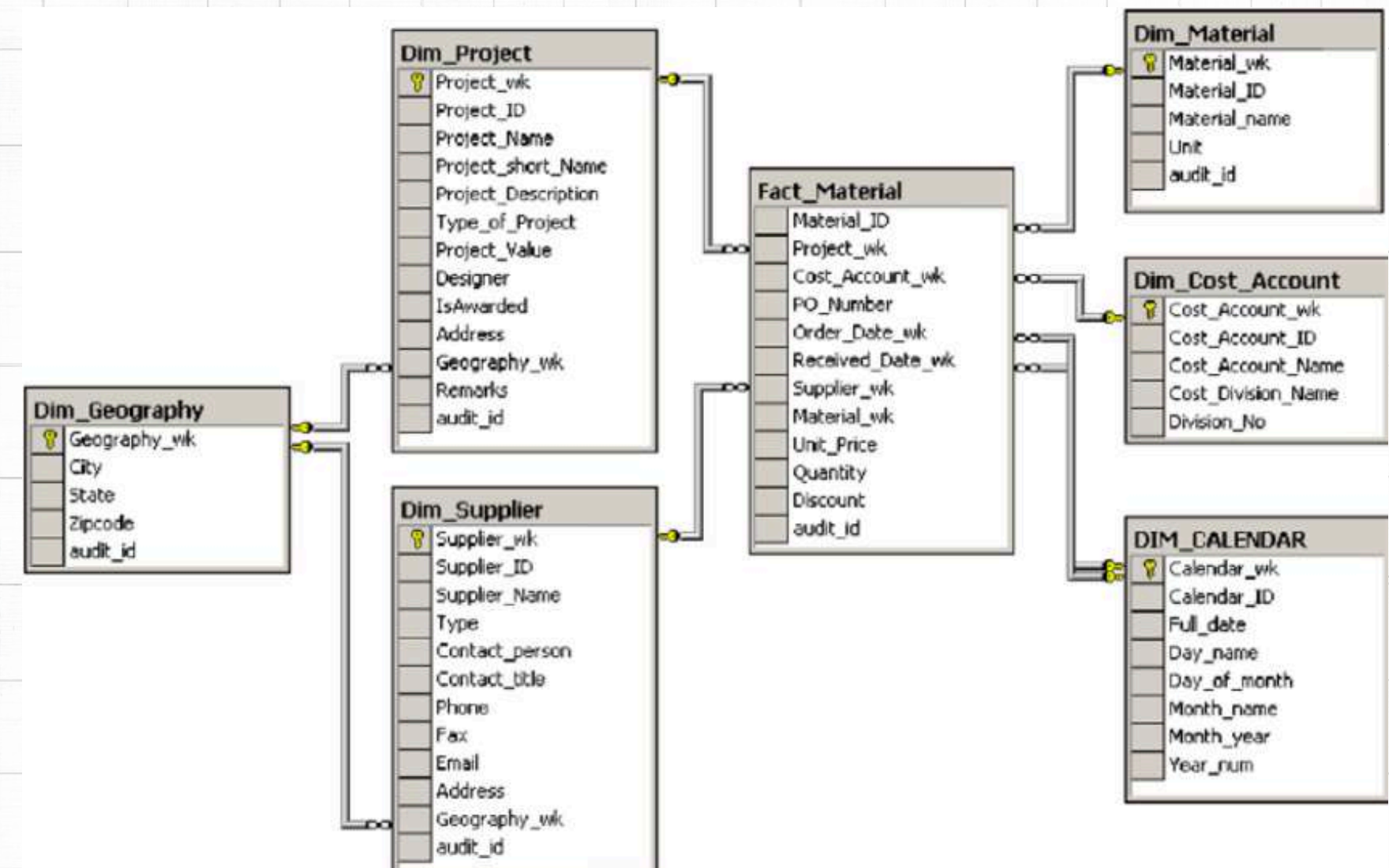
Fact Table

- **Fact Table** adalah *essential table* dari setiap *dimensional model* dan terdiri dari *business measurement*.
- Fact Table berisi *gabungan dari semua foreign key* dari dimensional table
- Fact Table berbentuk *metrics* yang dihasilkan dari *business proses atau pengukuran event*
 - Contoh: sales table, order table



Dimension Table

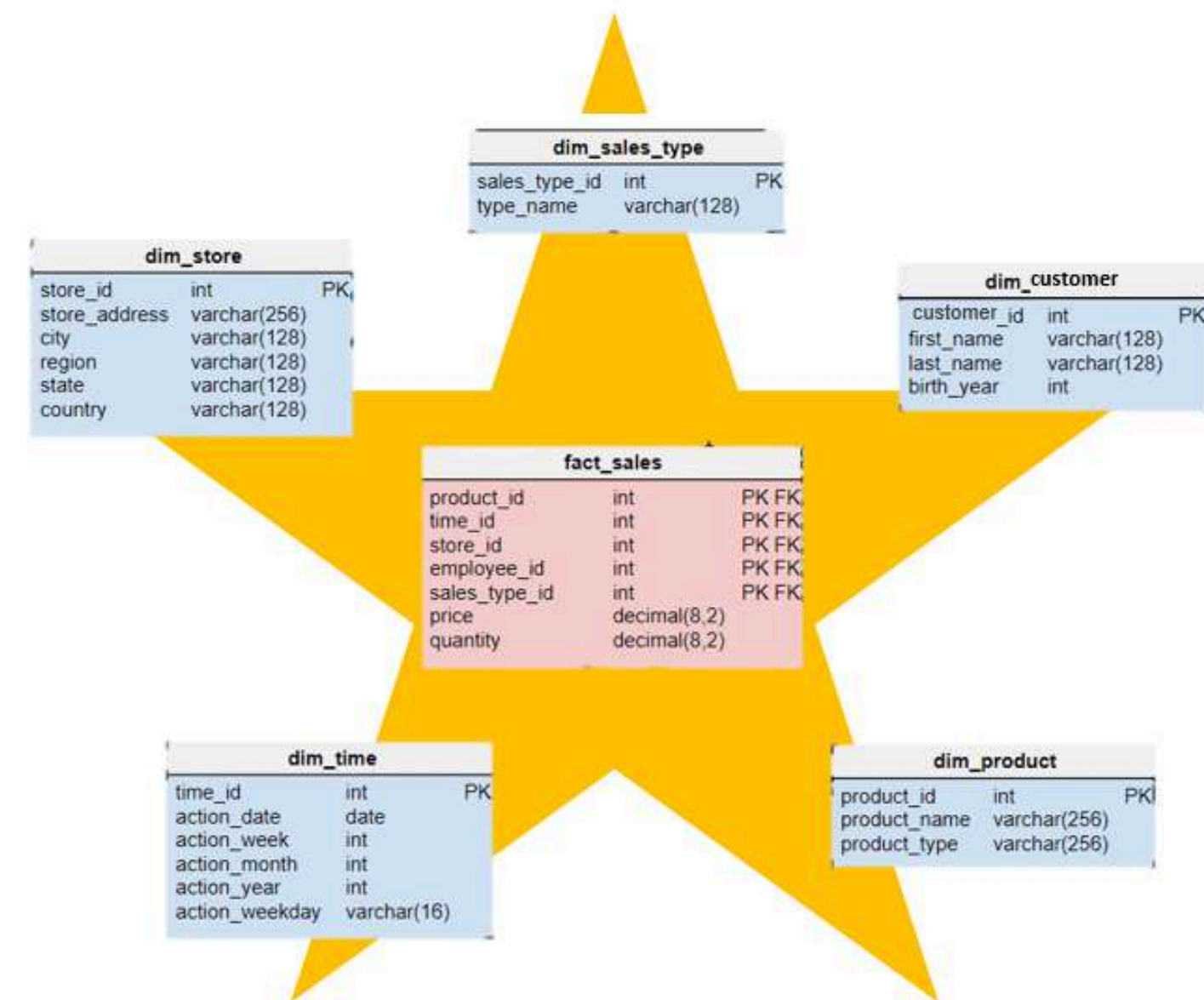
- **Dimension Table** berisi **atribut dari fact table** yang melingkupinya.
 - Contoh: Table Customer, Table Product
- Biasanya akan dicocokkan dengan key pada Fact Table untuk mendapatkan label keterangan dan atribut.



Star Schema

Keuntungan dari star schema adalah:

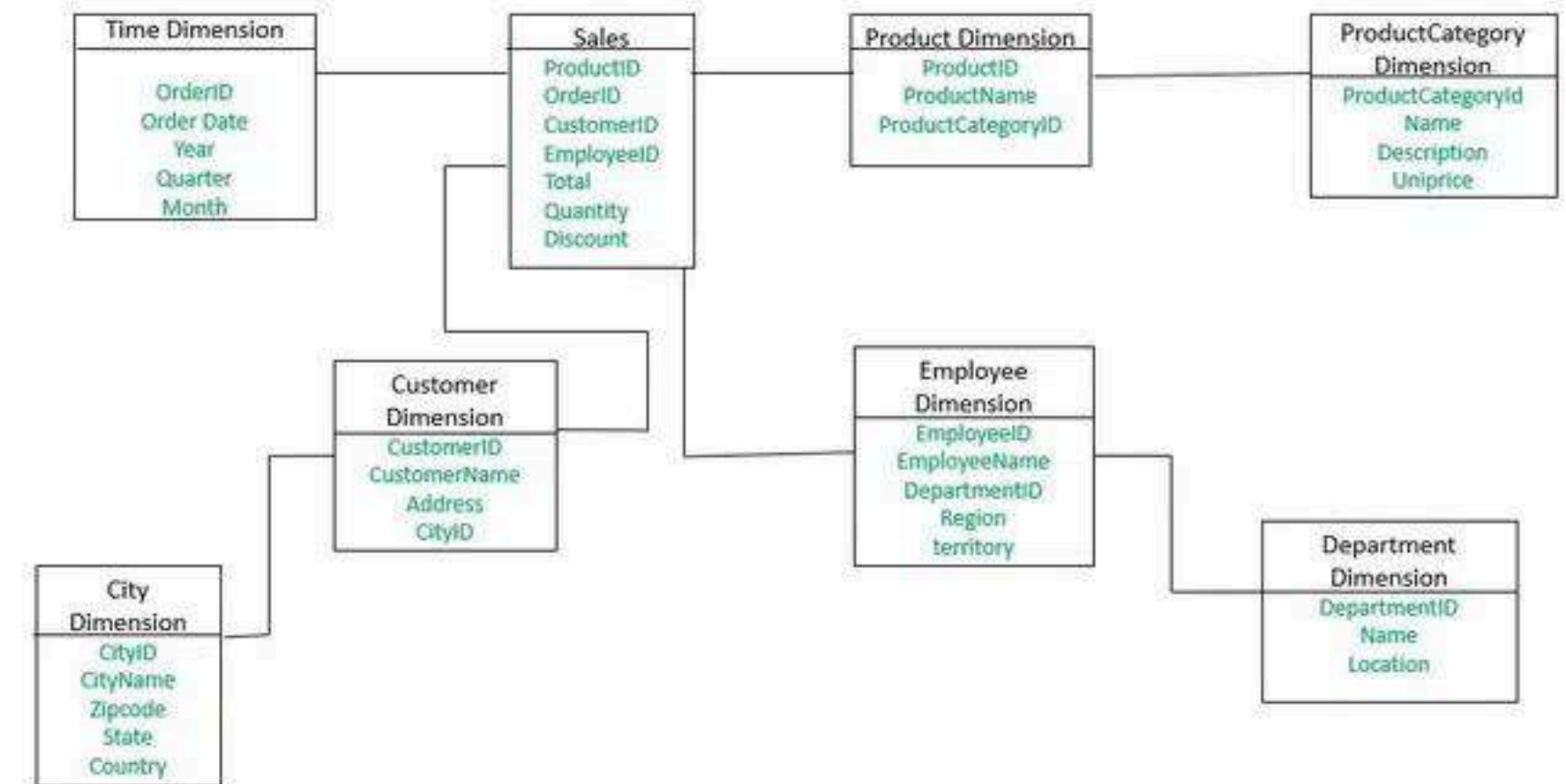
- Menyajikan pemetaan langsung dan jelas antara desain skema dan entitas bisnis yang akan dianalisis. Selain itu, format star schema mudah dimengerti
- Star Schema mudah diubah dan diperluas, hal ini menyederhanakan proses development
- Sebagian besar BI tools di market mendukung star schema



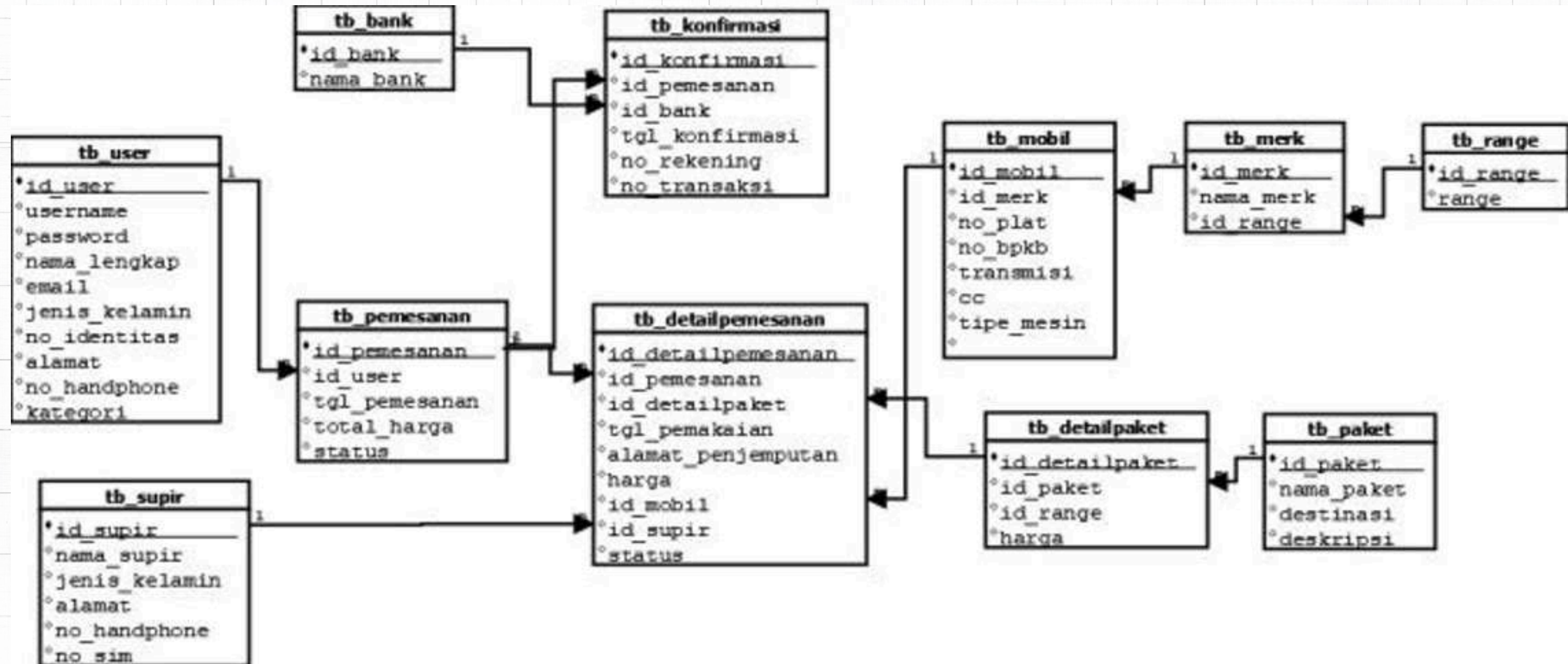
Snowflake Schema

Keuntungan dan kerugian menggunakan snowflake:

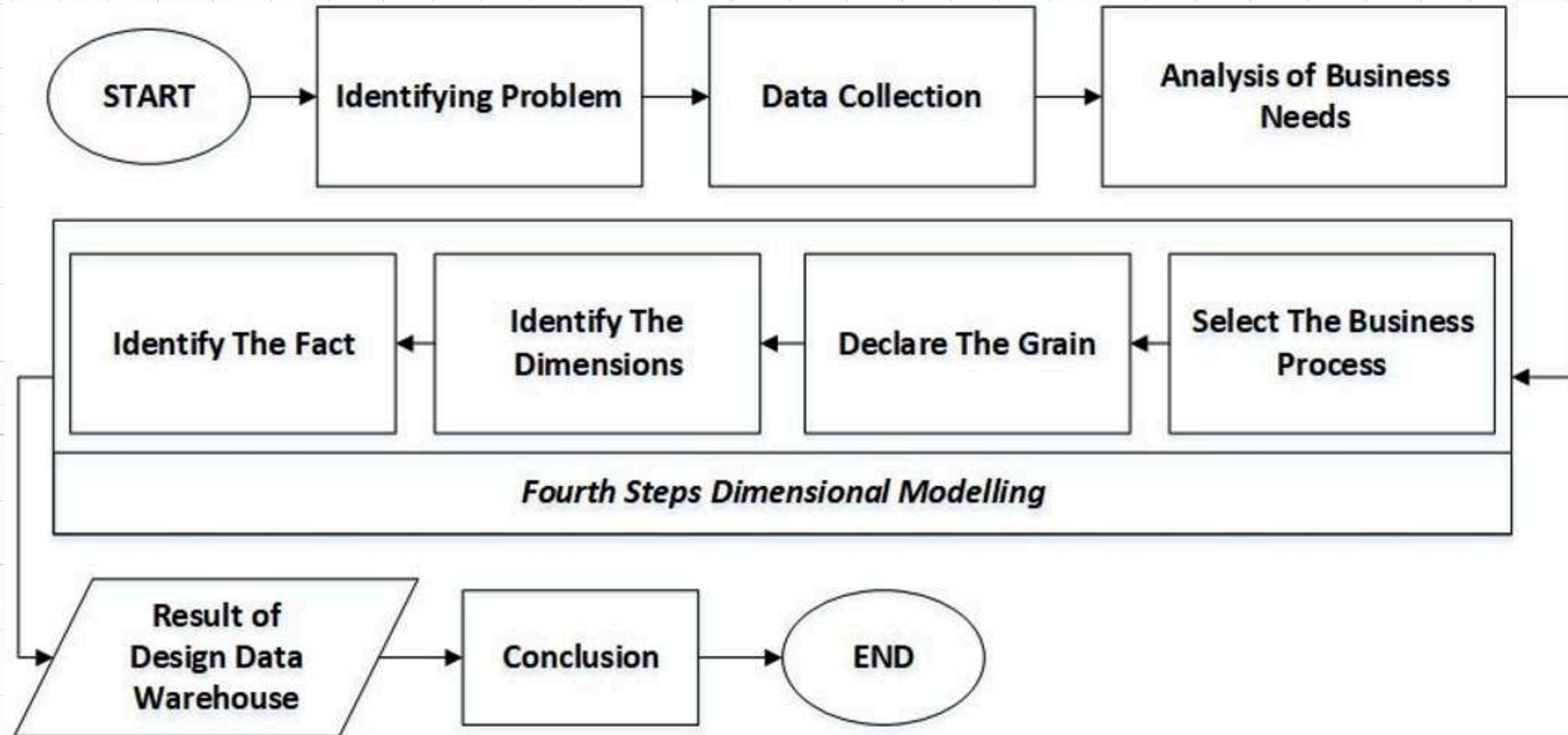
- Menghilangkan redundansi dengan menormalkan struktur dimensi
- Mengurangi beberapa biaya penyimpanan, tetapi dengan mengorbankan performa dalam pencarian data
- Membutuhkan lebih banyak foreign key
- Meningkatkan kompleksitas query dan efektivitas query



Case Study (E-Travel)



Step to Design Data Warehouse Model



Analysis of Business Needs

Dalam analisis optimalisasi bisnis, beberapa data penting diperlukan sesuai dengan target bisnis, antara lain:

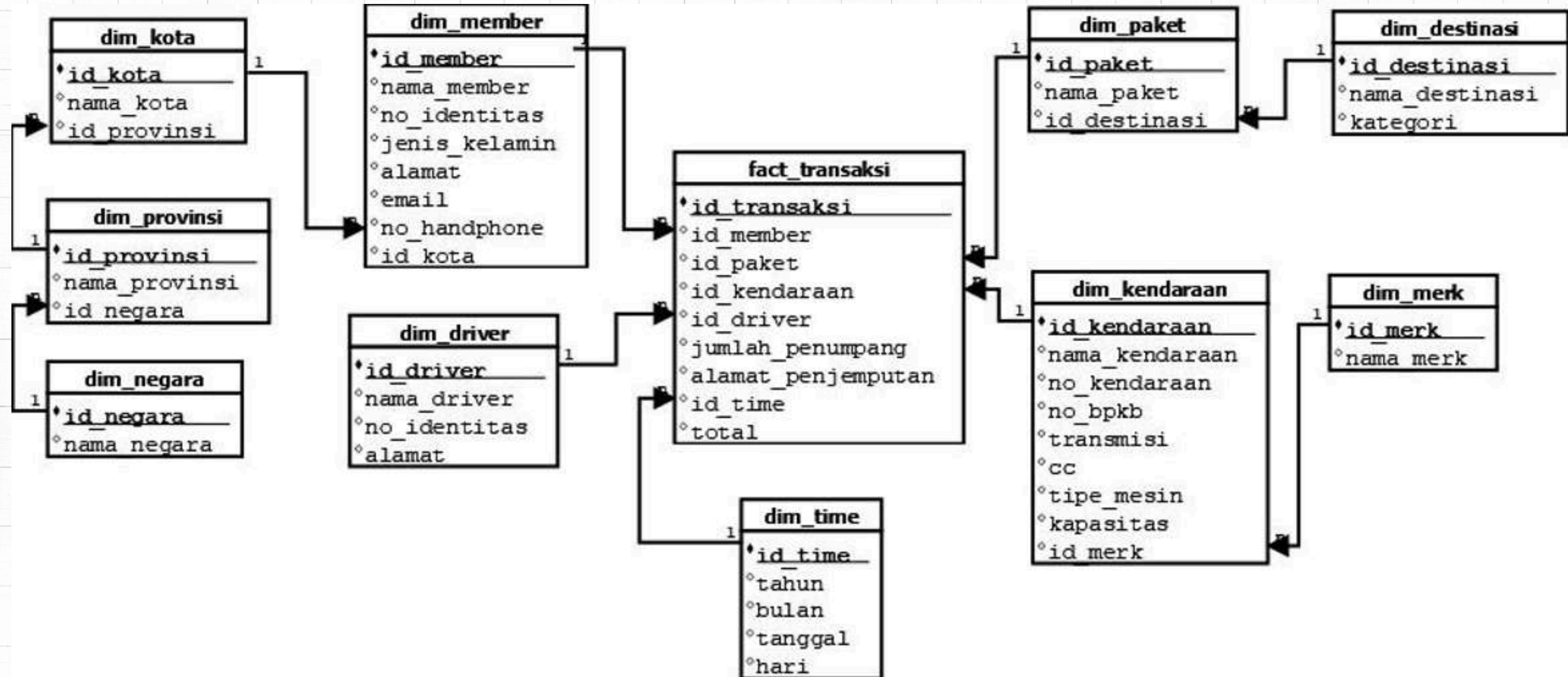
1. Data Kendaraan: Informasi lengkap mengenai sejarah penggunaan kendaraan diperlukan untuk menentukan intensitas penggunaan kendaraan berdasarkan data pemesanan.
2. Data Pengemudi: Penting untuk memahami rekam jejak pelayanan setiap pengemudi dalam melayani perjalanan, yang direkam dalam periode waktu tertentu yang terkait dengan data pesanan.
3. Data Pelanggan: Berkaitan erat dengan data pesanan, data pelanggan memungkinkan penghitungan jumlah pelanggan yang menggunakan layanan e-travel dalam periode waktu tertentu.
4. Data Paket Wisata: Diperlukan untuk mengetahui preferensi paket wisata yang paling diminati oleh pelanggan, dengan catatan data terkait dengan pesanan.
5. Data Pesanan: Merupakan data yang krusial karena merupakan tempat penyimpanan semua data lainnya dan menjadi dasar evaluasi dalam optimalisasi bisnis e-travel.

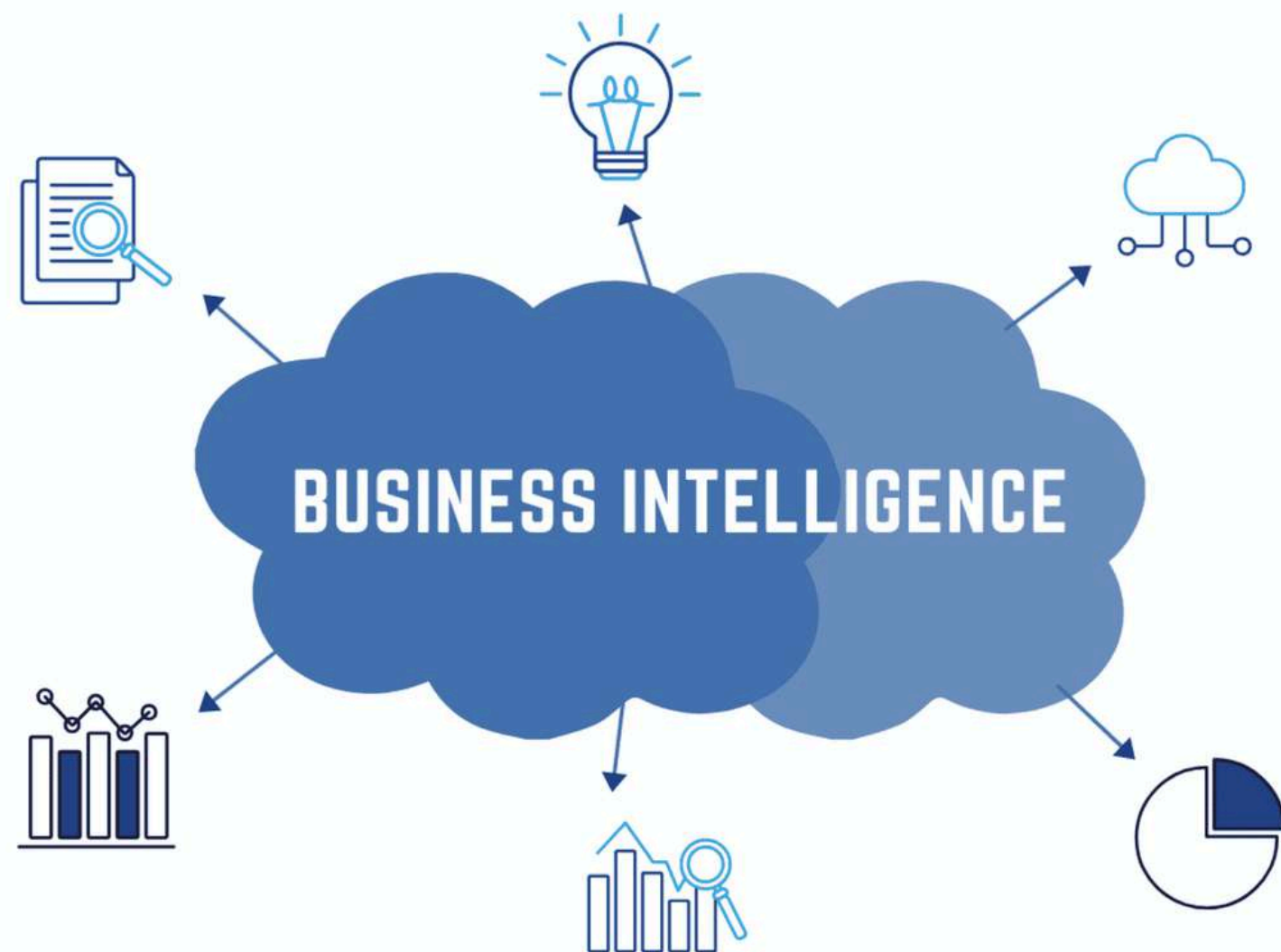
Declare The Grain

Grain dalam konteks e-travel adalah representasi setiap data yang terdokumentasi dalam tabel fakta yang akan dibuat. Grain dalam desain data warehouse digunakan untuk mencatat proses transaksi data dari layanan bisnis yang disediakan oleh e-travel kepada pelanggan.

Grain Dim	Amount of Order	Vehicle Use	Tour Package	Driver Service
Tour Package			✓	
Vehicle		✓		
Customer	✓			
Driver				✓
Time	✓	✓	✓	✓

Identify The Dimensions





BUSINESS INTELLIGENCE

- Apa itu Business Intelligence?
- Data Visualization

Apa itu Business Intelligence ?

Business Intelligence (BI) adalah teknologi yang membantu bisnis mengumpulkan, menyimpan, dan menganalisis data untuk membuat keputusan yang lebih baik. BI melibatkan alat dan metode untuk:

- Mengumpulkan data dari sumber internal dan eksternal.
- Menyimpan data untuk diolah lebih lanjut.
- Menganalisis data dengan membuat kueri dan laporan.
- Menyajikan hasil analisis melalui laporan, dasbor, dan visualisasi data.

Dengan BI, organisasi dapat memahami tren, mengidentifikasi peluang, dan meningkatkan efisiensi operasional berdasarkan data yang akurat.



Mengapa Perusahaan harus memiliki BI ?



Perusahaan membutuhkan Business Intelligence (BI) untuk meningkatkan pengambilan keputusan dan kinerja bisnis:

1. **Pengambilan Keputusan Lebih Baik:** BI menyediakan data akurat dan terkini, membantu manajemen membuat keputusan berdasarkan fakta.
2. **Efisiensi Operasional:** BI menganalisis proses bisnis untuk mengidentifikasi dan memperbaiki area yang tidak efisien, mengurangi biaya, dan meningkatkan produktivitas.
3. **Pemahaman Pelanggan:** BI menganalisis perilaku dan preferensi pelanggan, memungkinkan perusahaan merancang produk dan layanan yang lebih sesuai.
4. **Identifikasi Peluang dan Manajemen Risiko:** BI menganalisis tren pasar dan data kompetitif untuk menemukan peluang baru dan mengidentifikasi serta mengelola potensi risiko secara efektif.
5. **Prediksi dan Pengukuran Kinerja:** BI menggunakan data historis untuk membuat prediksi masa depan dan memantau kinerja perusahaan terhadap target dan KPI, membantu dalam perencanaan strategis dan evaluasi kinerja.

Data Visualization

Data Visualization adalah apa pun yang mengubah sumber data menjadi representasi visual bagan, grafik, peta, bahkan hanya tabel dan tentunya membantu untuk mempermudah dan memahami suatu pola yang kompleks, tren dan juga insight dari sebuah data.

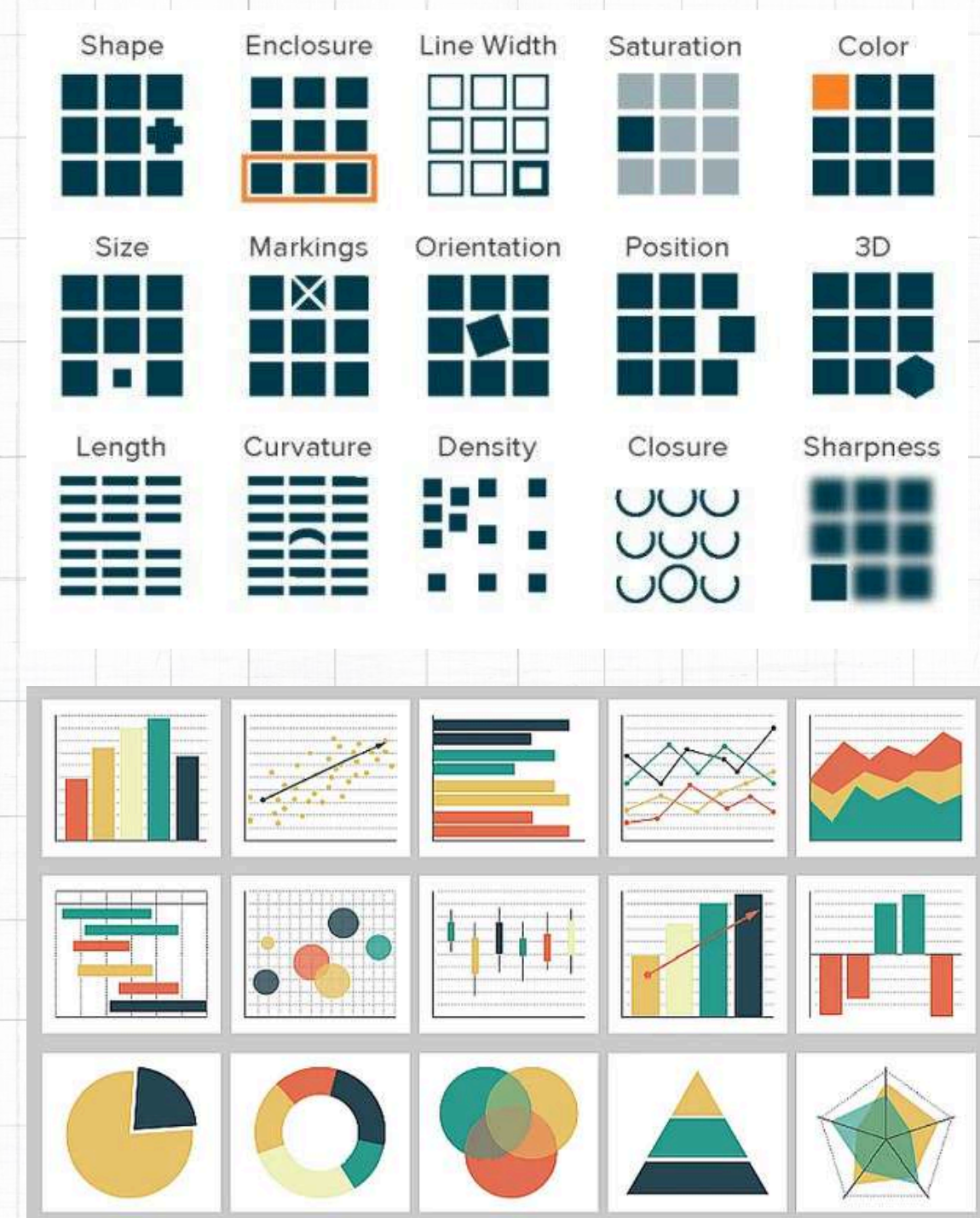
Tujuan dari Data Visualization:

- Menyederhanakan data kompleks: Mengubah data yang rumit menjadi bentuk yang lebih mudah dipahami.
- Mengidentifikasi pola dan tren: Membantu mengidentifikasi pola, tren, dan anomali dalam data.
- Komunikasi efektif: Memudahkan penyampaian informasi kepada audiens yang mungkin tidak familiar dengan detail teknis data.

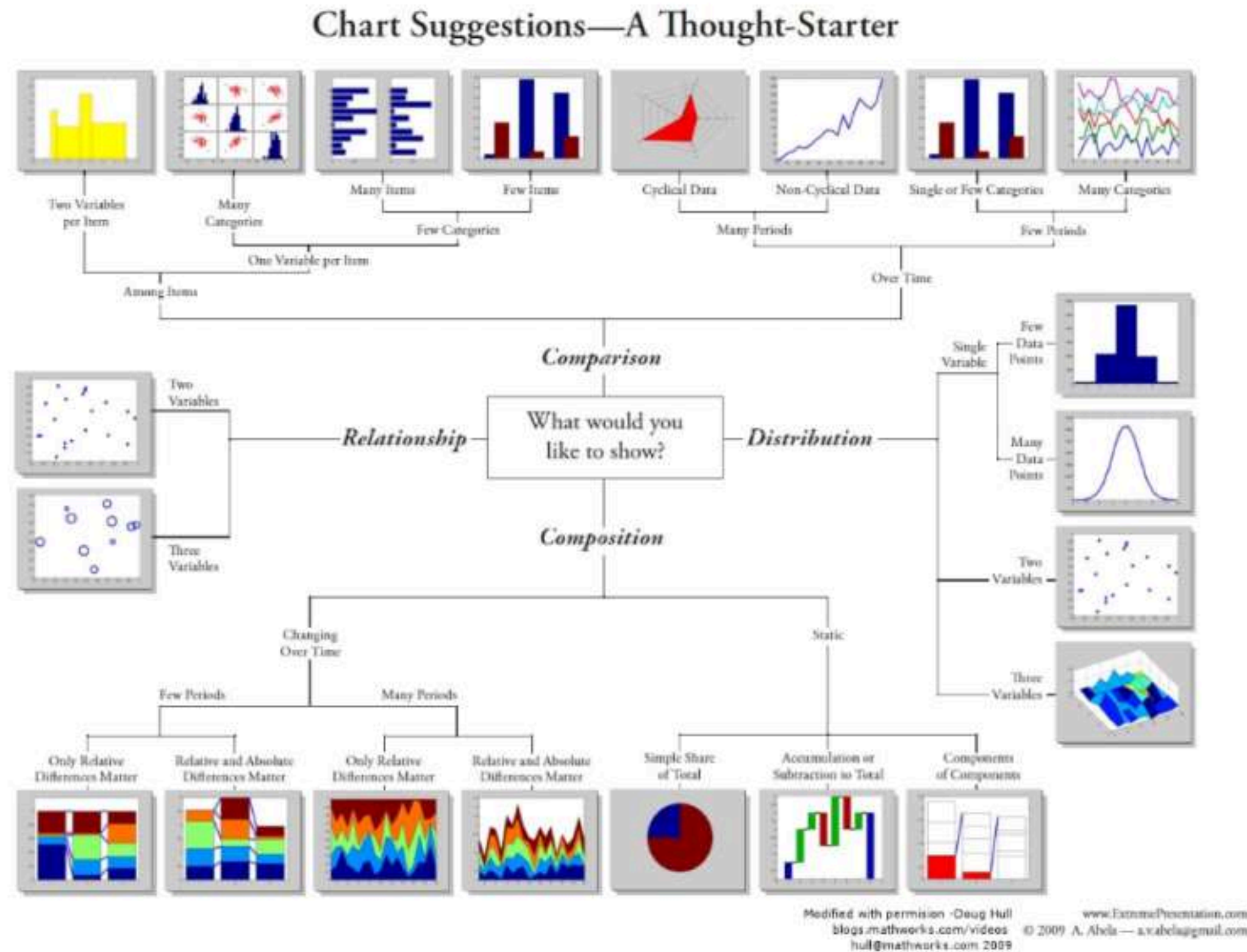


Data Visualization Principles

- **Tell the truth:** Jangan manipulasi data untuk menciptakan kesan yang salah. Pastikan visualisasi mencerminkan realitas data.
- **Know your audience:** Pahami siapa yang akan melihat visualisasi kita. Sesuaikan tingkat detail dan kompleksitas dengan pemahaman dan kebutuhan mereka.
- **Choose the right chart:** Pilih jenis grafik yang paling efektif untuk menyampaikan informasi. Misalnya, gunakan grafik batang untuk membandingkan kuantitas, atau diagram lingkaran untuk menunjukkan proporsi.
- **Emphasize the important:** Soroti elemen penting dalam visualisasi kita. Gunakan warna, ukuran, atau posisi untuk menarik perhatian ke informasi kunci.
- **Form should follow function:** Desain visualisasi harus mendukung tujuan komunikasi. Jangan biarkan keinginan untuk membuat sesuatu yang “cantik” mengganggu penyampaian informasi.

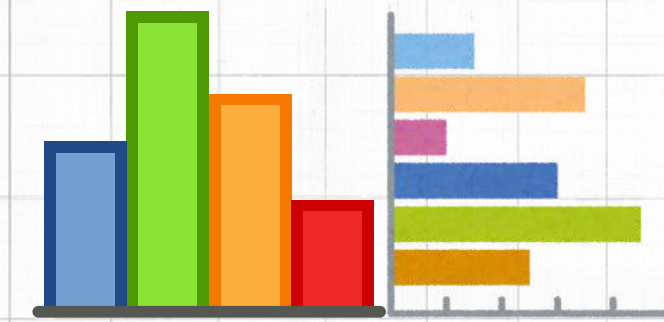


The Common Visualization

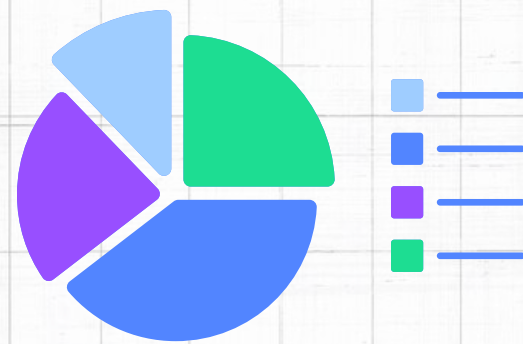


Bar Chart

Bar Chart digunakan ketika kita ingin menunjukkan distribusi titik data atau melakukan perbandingan nilai di berbagai subkelompok data. Variabel utama diagram batang adalah variabel kategorinya. Sebaliknya, variabel sekunder akan bersifat numerik.



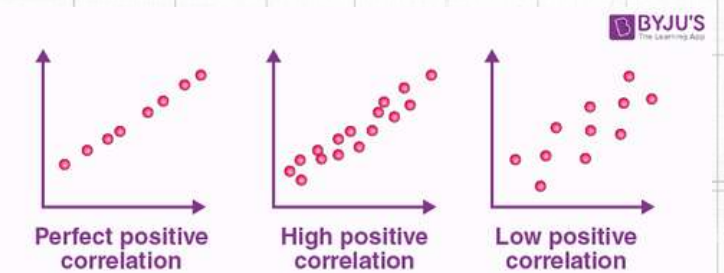
Pie Chart



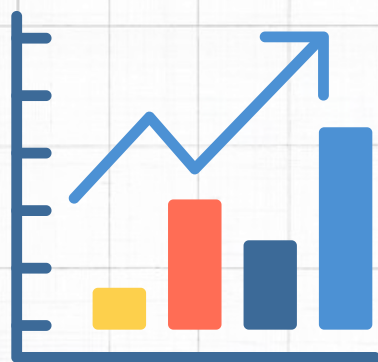
Pie chart digunakan ketika saat ingin membandingkan nilai atau jumlah antara beberapa kategori yang berbeda secara visual. Contohnya, jika kita ingin membandingkan penjualan produk A, B, dan C dalam satu bulan, bar chart akan menampilkan ketinggian batang yang mewakili jumlah penjualan masing-masing produk secara jelas dan mudah dimengerti.

Scatter Plot

Kita menggunakan scatter plot saat ingin melihat hubungan antara dua variabel, misalnya apakah ada korelasi antara tinggi badan dan berat badan seseorang. Dengan scatter plot, kita dapat melihat pola atau hubungan antara kedua variabel tersebut dengan melihat sebaran titik-titik data pada grafik.



Line Chart



Kita menggunakan line chart saat ingin menunjukkan perubahan atau tren dari satu variabel seiring waktu. Misalnya, jika kita ingin melihat bagaimana penjualan suatu produk berubah dari bulan ke bulan selama setahun, line chart akan membantu kita melihat tren naik atau turunnya penjualan secara visual.

Google Data Studio

Google Data Studio adalah alat visualisasi data yang memungkinkan pengguna untuk membuat laporan dan dasbor yang interaktif dan informatif.

Data Studio dapat terhubung ke set data seperti:

Google Marketing Platform :

Google Ads, Analytics, Display & Video 360, Search Ads 360

Google Database Platform :

BigQuery, MySQL, dan PostgreSQL

Produk konsumen Google :

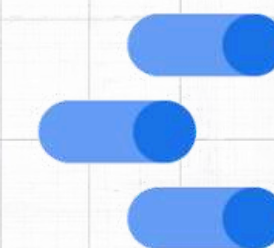
Spreadsheet, YouTube, and Search Console

Platform media sosial :

Facebook, Reddit, dan Twitter

File flat :

file CSV dan Google Cloud Storage



Data Studio

Thank You
