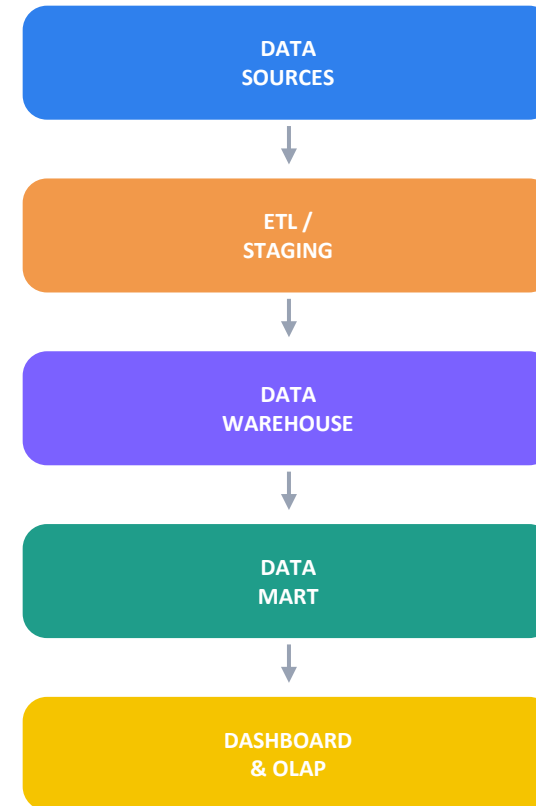


BUSINESS INTELLIGENCE

Pertemuan 3

Data Warehouse: Fondasi Data untuk Business Intelligence

Memahami bagaimana data operasional diintegrasikan, disimpan, dan dimodelkan agar siap untuk analitik dan pengambilan keputusan.



FROM TRANSACTIONS TO TRUSTED ANALYTICS

Capaian Pembelajaran Pertemuan 3

Fokus OBE: mahasiswa memahami konsep sekaligus mampu memodelkan data warehouse sederhana dari kebutuhan bisnis.

3

Sub-CPMK 3

Mampu menjelaskan konsep Data Warehouse, karakteristiknya, proses ETL, perbedaan OLTP dan OLAP, serta merancang model dimensional sederhana untuk kebutuhan Business Intelligence.



Indikator Keberhasilan

1. Menjelaskan karakteristik Data Warehouse.
2. Membedakan OLTP dengan OLAP.
3. Menentukan grain, fact, dimension, dan measure.
4. Menjelaskan peran ETL dan kualitas data.



Bukti Belajar

- Diskusi pemantik dan mini-case
- Latihan identifikasi fact–dimension
- Case method perancangan star schema
- Cek pemahaman formatif
- Exit ticket: “satu row pada fact table berarti apa?”

Dari Arsitektur BI ke Data Warehouse

Pertemuan 2 melihat aliran end-to-end. Pertemuan 3 memperdalam “jantung penyimpanan analitik” pada arsitektur tersebut.

2 Pertemuan 2 • Arsitektur

Data source → ingestion/ETL → warehouse → OLAP/semantic layer → dashboard/analytics.

Pertanyaan utama: bagaimana data bergerak dan siapa yang memakainya?



3 Pertemuan 3 • Data Warehouse

Hari ini kita fokus pada:

- Mengapa perlu warehouse?
- Apa bedanya dengan database transaksi?
- Bagaimana struktur fact & dimension?
- Bagaimana ETL menjaga data tetap konsisten?
- Bagaimana model ini menjadi fondasi OLAP dan dashboard?

Agenda Hari Ini

Alur belajar: masalah bisnis → konsep → pemodelan → praktik → evaluasi.

1

Pemantik: query analitik vs sistem transaksi

2

Konsep & karakteristik Data Warehouse

3

OLTP vs OLAP

4

Dimensional modeling: grain, fact, dimension

5

ETL, staging, data quality & data mart

6

Studi kasus ritel & star schema

7

Case method, quiz & refleksi

PERTANYAAN KUNCI

“Bagaimana mengubah jutaan transaksi harian menjadi data historis yang konsisten, cepat dianalisis, dan dipercaya oleh seluruh unit bisnis?”

Mengapa Query Laporan Bisa Membuat Sistem Kasir Lambat?

Bayangkan sebuah jaringan ritel dengan 500 toko dan jutaan transaksi per hari.

A

Sistem Kasir (OLTP)

Harus cepat mencatat transaksi:

- scan barcode
- hitung harga
- simpan pembayaran
- update stok

Prioritas: transaksi singkat, konsisten, dan selalu tersedia.

B

Tim Manajemen

Ingin menjawab pertanyaan:

- tren penjualan 3 tahun
- produk paling profitable
- performa toko per wilayah
- pola musiman & promosi

Query-nya berat dan memindai data besar.

!

Konflik

Jika analitik berat dijalankan langsung pada database kasir:

1. resource berebut
2. query transaksi melambat
3. histori sulit dikelola
4. definisi data bisa berbeda

Solusi: pisahkan workload analitik.

PERTANYAAN: di mana sebaiknya analitik historis dijalankan?

Apa Itu Data Warehouse?

Repositori terintegrasi untuk menyimpan data historis yang dirancang terutama untuk analitik dan pengambilan keputusan.

INTI DEFINISI

DATA WAREHOUSE

“Satu tempat terkontrol untuk menyatukan data dari banyak sumber, mempertahankan histori, dan menyajikannya dalam struktur yang mudah dianalisis.”

1

Subject-oriented

Disusun menurut area bisnis: sales, customer, finance.

2

Integrated

Format, kode, dan definisi diseragamkan lintas sumber.

3

Time-variant

Histori disimpan agar perubahan dari waktu ke waktu dapat dianalisis.

4

Non-volatile

Data analitik relatif stabil; umumnya ditambah, bukan diubah seperti transaksi.

Mengapa Database Operasional Tidak Cukup?

Database transaksi sangat baik untuk operasi harian, tetapi tidak selalu cocok untuk analitik lintas waktu dan lintas sistem.

1

Workload berbeda

Transaksi butuh response milidetik; analitik butuh scan, agregasi, dan join besar.

2

Struktur berbeda

OLTP cenderung ternormalisasi; analitik lebih mudah dengan model dimensional.

3

Historisasi

Sistem operasional sering hanya menyimpan status terkini.

4

Integrasi

Nama pelanggan, kode produk, mata uang, dan kalender dapat berbeda antar sistem.

5

Governance

Definisi KPI harus konsisten agar semua unit melihat “angka yang sama”.

6

Kinerja

Analitik berat sebaiknya tidak mengganggu aplikasi operasional.

4 Karakteristik Klasik Data Warehouse

Karakteristik ini membantu membedakan warehouse dari database transaksi biasa.

01

Subject-Oriented

Fokus pada subjek bisnis penting: pelanggan, produk, penjualan, supplier.

02

Integrated

Data dari banyak sistem diseragamkan—format tanggal, kode, unit, definisi, dan key.

03

Time-Variant

Setiap fakta memiliki konteks waktu sehingga tren dan perubahan dapat dilihat.

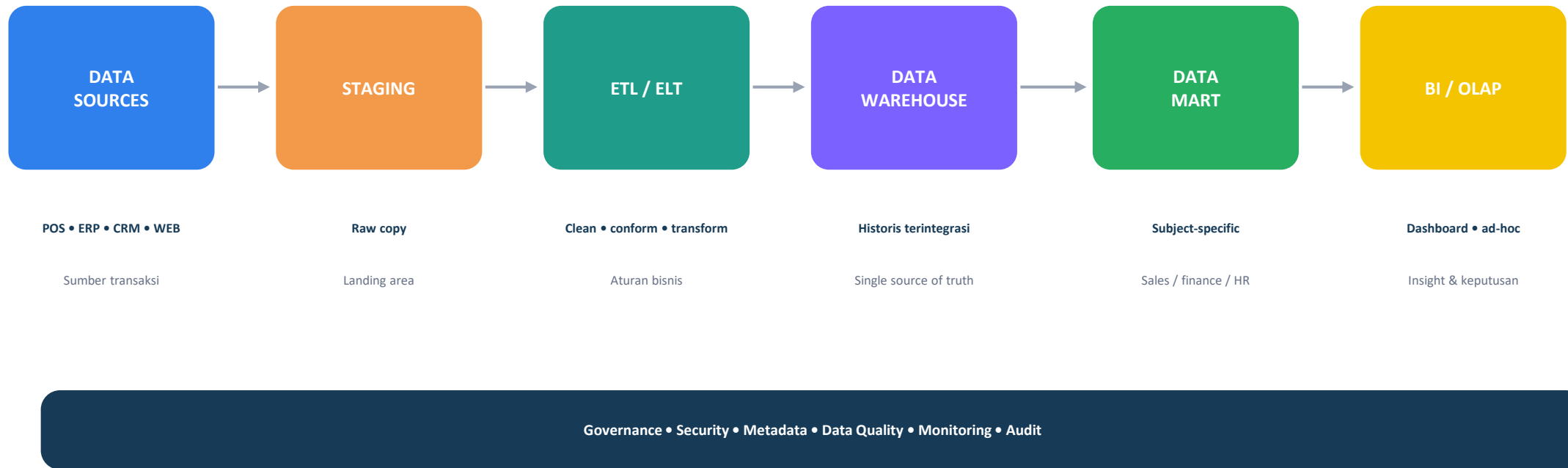
04

Non-Volatile

Data historis biasanya dimuat secara terkendali dan tidak terus-menerus diubah oleh user operasional.

Aliran Data: Dari Sistem Operasional ke Data Warehouse

Warehouse tidak berdiri sendiri; ia berada di tengah pipeline data analitik.



OLTP vs Data Warehouse / OLAP

Sistem transaksi dan sistem analitik dioptimalkan untuk tujuan yang berbeda.

Aspek	OLTP	DW / OLAP
Tujuan	Mencatat transaksi operasional	Analisis, tren, perbandingan
Query	Pendek, sering, predictable	Kompleks, agregasi, ad-hoc
Data	Status terkini / detail transaksi	Historis, terintegrasi, lintas sumber
Model	Normalized (3NF)	Dimensional: star / snowflake
Update	Insert / update / delete kontinu	Batch / incremental load terkontrol
Pengguna	Kasir, aplikasi, staf operasi	Analisis, manajer, eksekutif
Contoh	Simpan order baru	Penjualan per wilayah 36 bulan

Mulai dari Grain: “Satu Baris Berarti Apa?”

Grain adalah tingkat detail terendah dari sebuah fact table. Menentukan grain harus dilakukan sebelum memilih dimensions dan measures.

PERTANYAAN PENTING

“Satu baris pada fact table ini merepresentasikan apa?”

Contoh grain yang baik:

- satu item pada satu transaksi kasir
- satu saldo akun per hari
- satu klik iklan per user per campaign

1

Terlalu kasar

1 baris = total penjualan bulanan toko
Sulit drill-down ke produk / transaksi.

2

Terlalu campur

Baris kadang per invoice, kadang per item
Measure menjadi tidak konsisten.

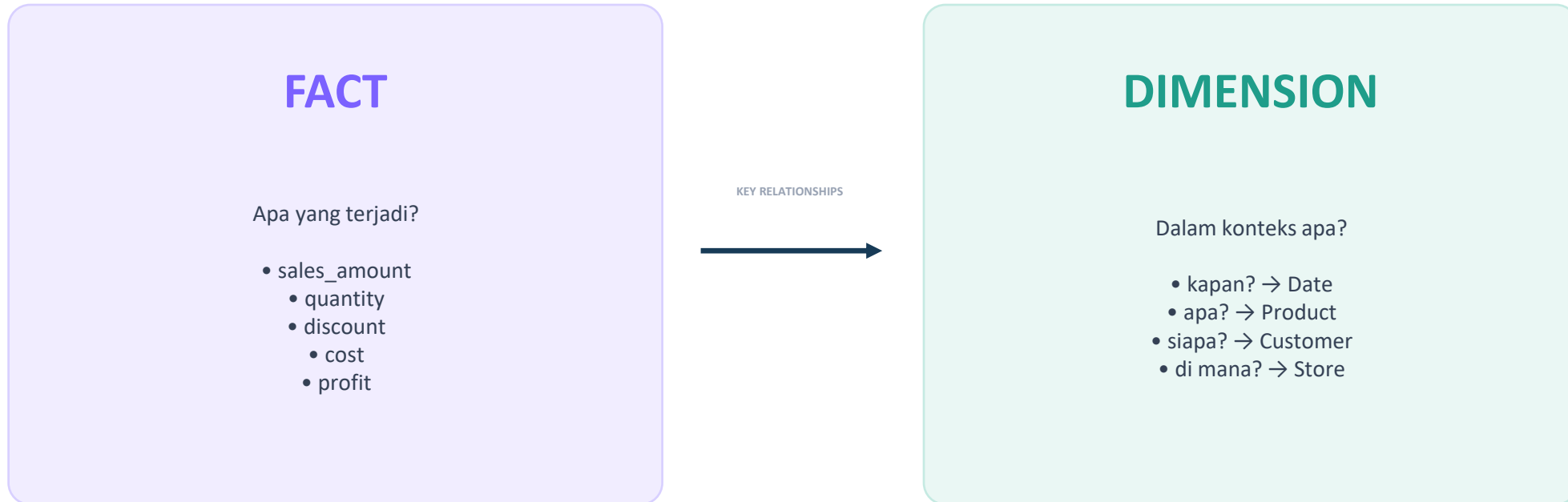
3

Jelas & atomik

1 baris = 1 produk dalam 1 transaksi
Mendukung agregasi fleksibel.

Fact & Dimension: Struktur Inti Model Analitik

Model dimensional memisahkan “angka yang diukur” dari “konteks yang menjelaskan angka tersebut”.



Fact Table: Menyimpan Kejadian dan Measure

Fact table biasanya memiliki banyak baris, foreign key ke dimensions, dan angka yang dapat diagregasi.

FACT_SALES

date_key	product_key	store_key	customer_key	qty	sales_amount	cost
20260919	P001	S010	C1934	2	48.000	31.000
20260919	P083	S010	C1934	1	25.000	16.000
20260919	P114	S002	C0042	3	90.000	60.000



Foreign Keys

Measures

1

Ciri umum

- Grain jelas
- Volume tinggi
- Numeric measures
- Foreign keys ke dimension

?

Pertanyaan yang didukung

“Berapa penjualan produk X, di toko Y, pada bulan Z, untuk segmen pelanggan tertentu?”

Dimension: Memberi Konteks untuk Filter, Grouping, dan Drill-Down

Dimension berisi atribut deskriptif yang membuat angka pada fact dapat dipahami oleh bisnis.

DIM_DATE

date_key • date • day_name • month • quarter • year

DIM_PRODUCT

product_key • sku • product_name • brand • category • subcategory

DIM_STORE

store_key • store_name • city • region • format

DIM_CUSTOMER

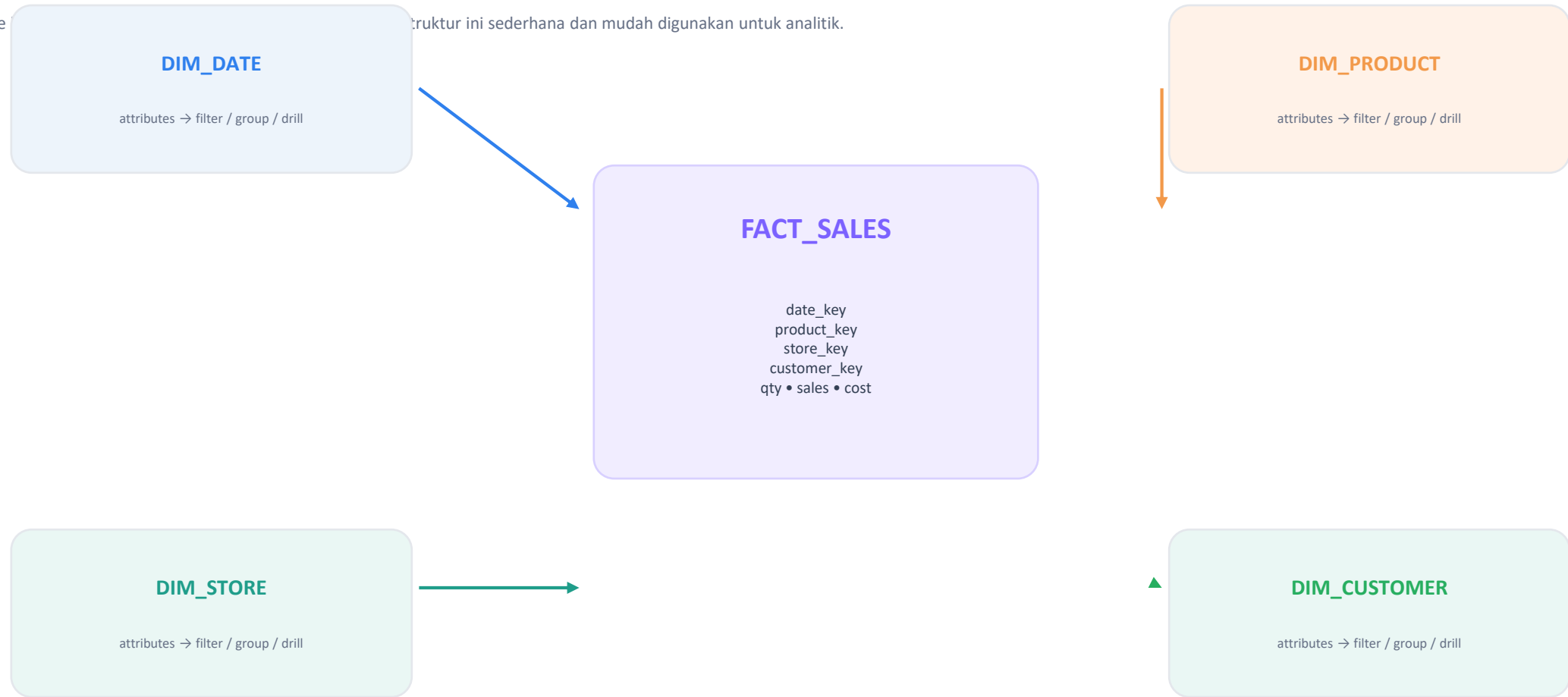
customer_key • segment • gender • age_band • loyalty_tier

Dimension = bahasa bisnis untuk menjelaskan angka

Contoh Star Schema: Penjualan Ritel

Fact table

struktur ini sederhana dan mudah digunakan untuk analitik.



Keuntungan: query lebih mudah • definisi bisnis jelas • performa analitik baik • user-friendly

Star Schema vs Snowflake Schema

Keduanya valid; pilihan tergantung kebutuhan kesederhanaan, reuse, dan kompleksitas model.

★ STAR SCHEMA

Dimension relatif denormalisasi.

- ✓ Lebih mudah dipahami user bisnis
- ✓ Lebih sedikit join
- ✓ Cocok untuk semantic model / BI

Trade-off: beberapa atribut dapat berulang di dimension.

❄ SNOWFLAKE SCHEMA

Dimension dinormalisasi ke tabel turunan.

- ✓ Mengurangi redundancy
- ✓ Hierarki tertentu lebih terstruktur

Trade-off: join lebih banyak dan model lebih kompleks untuk pengguna.

Prinsip praktis: utamakan model yang paling mudah dipahami dan konsisten untuk kebutuhan analitik.

Tidak Semua Measure Bisa Dijumlahkan dengan Cara yang Sama

Pemahaman tipe measure mencegah agregasi yang menyesatkan.

1

Additive

Dapat dijumlahkan di semua dimension.

Contoh:

Sales amount, quantity, cost

2

Semi-additive

Dapat dijumlahkan pada sebagian dimension, tetapi tidak pada waktu.

Contoh:

Account balance, inventory level

3

Non-additive

Tidak boleh langsung dijumlahkan; perlu formula/agregasi khusus.

Contoh:

Ratio, margin %, average price

Contoh jebakan: menjumlahkan “margin %” antar toko $\neq$ margin perusahaan. Gunakan $SUM(profit) / SUM(sales)$.

Business Key, Surrogate Key & Slowly Changing Dimension

Warehouse perlu menjaga identitas bisnis sekaligus histori perubahan atribut.

BK

Business Key

Key dari sistem sumber.

Contoh: customer_id = C00123.

Masalah: bisa berubah, bentrok antar sistem, atau dipakai ulang.

SK

Surrogate Key

Key buatan warehouse.

Contoh: customer_key = 109845.

Keuntungan: stabil, independen dari source, dan mendukung historisasi.

SCD

SCD

Menentukan cara menyimpan perubahan atribut dimension.

Type 1: overwrite

Type 2: tambah baris historis

Type 3: simpan current & previous secara terbatas.

Contoh: pelanggan pindah kota. Apakah laporan tahun lalu harus memakai kota lama atau kota baru?

ETL untuk Data Warehouse: Lebih dari Sekadar Memindahkan Data

ETL menerapkan aturan kualitas, harmonisasi, historisasi, dan audit sebelum data siap dianalisis.



Contoh transformasi: kode toko “JKT-01”, “Jakarta01”, dan “001-JKT” → satu store_key yang konsisten; timestamp UTC → kalender bisnis lokal; data customer duplikat → satu master record.

Staging Area & Quality Gates

Staging menyimpan data sebelum masuk ke warehouse; quality gates mencegah data bermasalah merusak laporan.

S

Staging Area

Tempat landing data source secara temporer / raw.

Fungsi:

- isolasi source dari transformasi
- restart / reprocess
- audit input
- debugging pipeline

1

Completeness

Apakah field wajib terisi?

2

Validity

Apakah format / range valid?

3

Uniqueness

Apakah duplicate dikendalikan?

4

Consistency

Apakah kode dan definisi lintas sumber sama?

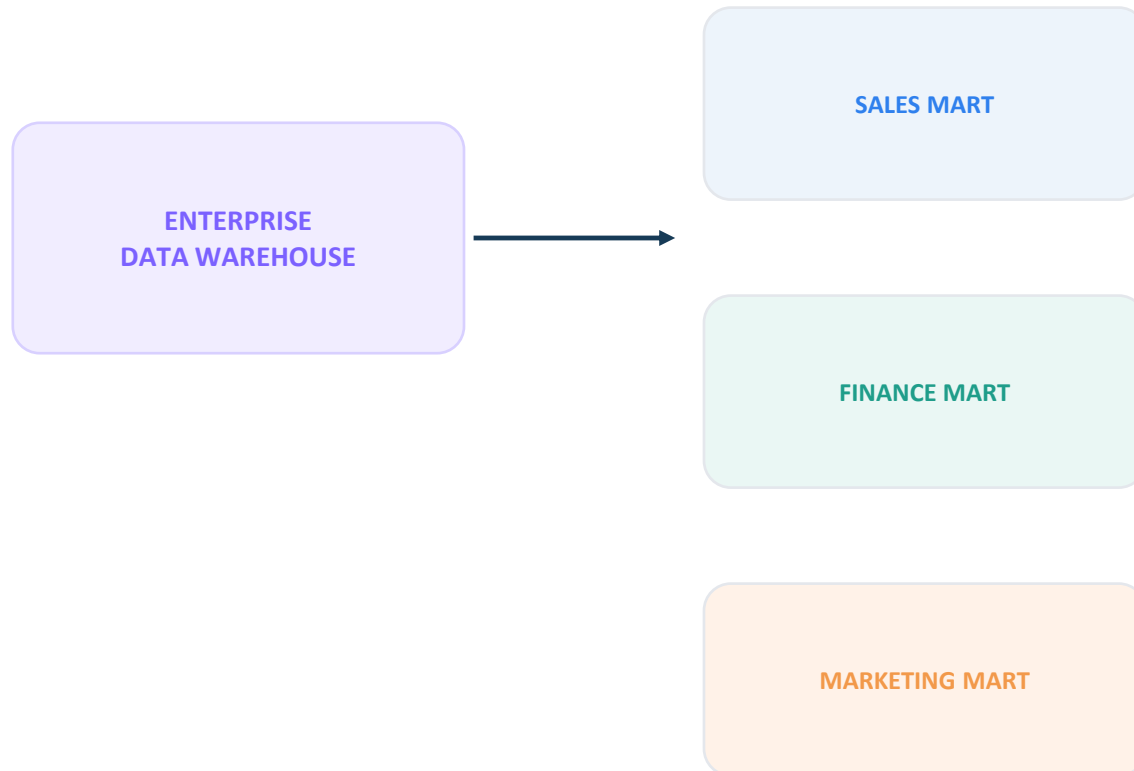
5

Reconciliation

Apakah total source = total hasil load?

Data Mart & Conformed Dimensions

Data mart menyajikan subset data untuk domain tertentu; conformed dimensions menjaga bahasa bisnis tetap sama.



CONFORMED DIMENSIONS

DIM_DATE

dipakai dengan definisi kalender yang sama di Sales, Finance, dan Marketing.

DIM_PRODUCT

kategori dan brand harus konsisten di semua mart.

DIM_CUSTOMER

segmentasi pelanggan harus memakai master dan aturan yang sama.

Tiga Pendekatan Populer untuk Platform Analitik

Tidak ada satu arsitektur yang cocok untuk semua organisasi. Pilihan dipengaruhi skala, kecepatan, skill, dan ekosistem teknologi.

K

Kimball / Bus

Dimensional data marts terintegrasi lewat conformed dimensions.

Kuat untuk delivery BI bertahap dan user bisnis.

Fokus: proses bisnis + star schema.

I

Inmon / EDW

Enterprise warehouse terintegrasi terlebih dahulu, biasanya lebih normalized; data marts diturunkan dari EDW.

Fokus: integrasi enterprise.

L

Modern Lakehouse

Data lake + kemampuan warehouse/SQL; mendukung BI, data science, dan semi-structured data pada satu platform.

Fokus: fleksibilitas & skala cloud.

Prinsip tetap sama: data harus dapat dipercaya, terdefinisi, terlacak, dan sesuai kebutuhan bisnis.

Kasus “Nusantara Mart”: Kebutuhan Bisnis

Manajemen ingin satu model data yang mampu menjawab pertanyaan lintas toko, produk, pelanggan, dan waktu.

1 Q1

Berapa omzet, quantity, cost, dan profit per hari / bulan?

2 Q2

Produk dan kategori mana yang tumbuh atau turun?

3 Q3

Toko / wilayah mana yang paling profitable?

4 Q4

Apakah promo meningkatkan volume atau hanya menurunkan margin?

5 Q5

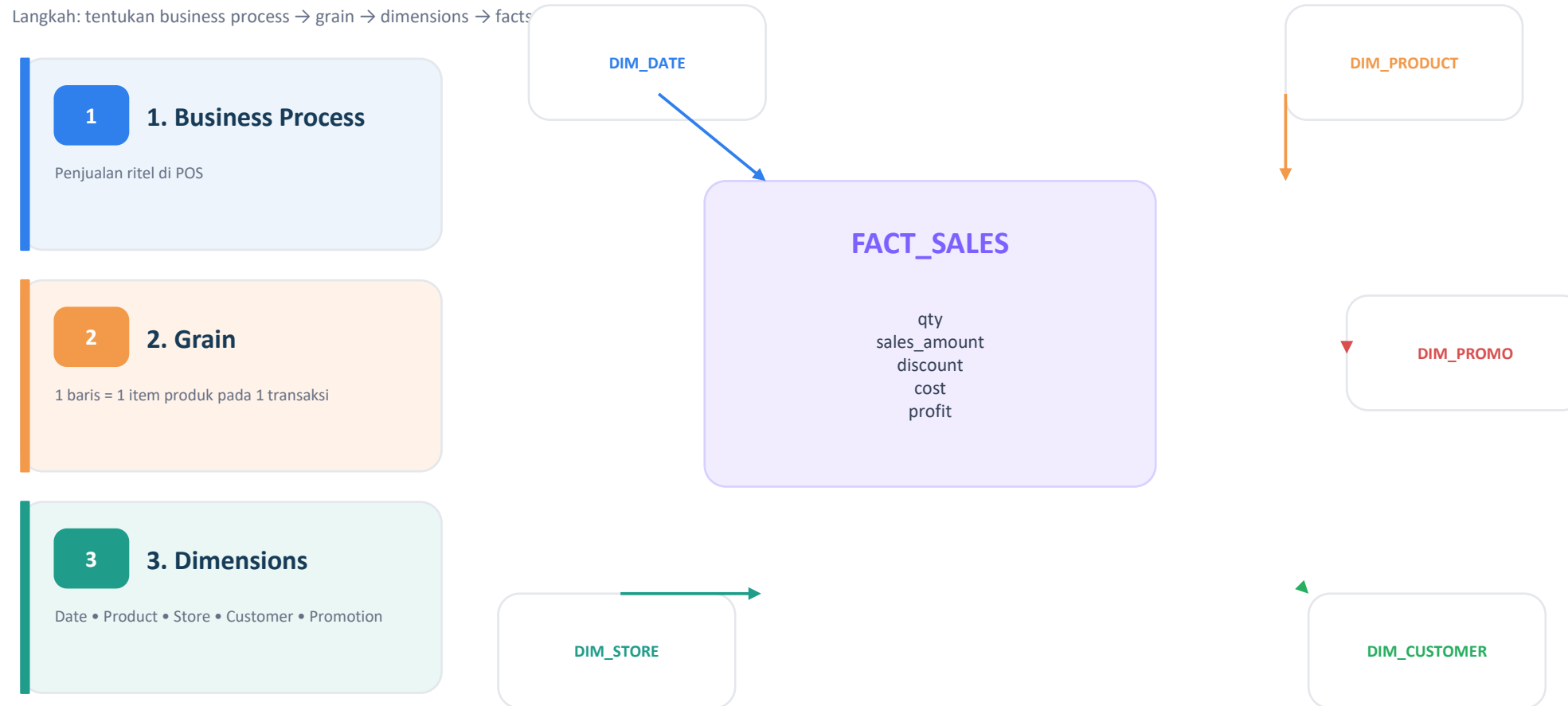
Bagaimana perilaku segmen loyalty vs non-loyalty?

6 Q6

Bagaimana seasonality dibanding periode tahun lalu?

Dari Kebutuhan Bisnis ke Star Schema

Langkah: tentukan business process → grain → dimensions → facts



Bagaimana Model Dimensional Memudahkan Analitik?

User bisnis berpikir dalam pertanyaan; model star menerjemahkannya menjadi filter, grouping, dan aggregation.

?

Pertanyaan Bisnis

“Berapa profit per kategori produk dan wilayah selama Q3 2026 dibanding Q3 2025?”

Filter: quarter / year

Group: category / region

Measure: SUM(profit)

Compare: year-over-year

SELECT

p.category, s.region, d.year,
SUM(f.profit) AS profit

FROM

fact_sales f
JOIN dim_date d ...
JOIN dim_product p ...
JOIN dim_store s ...

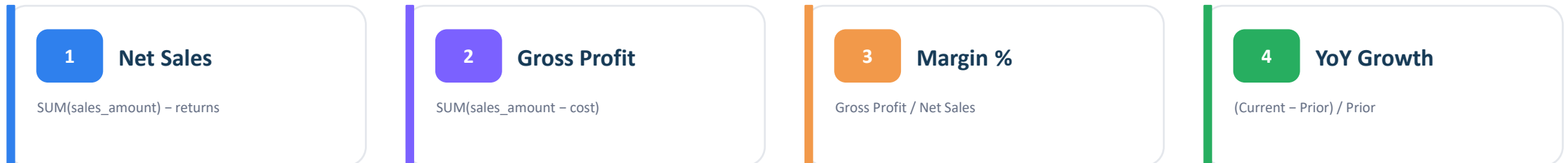
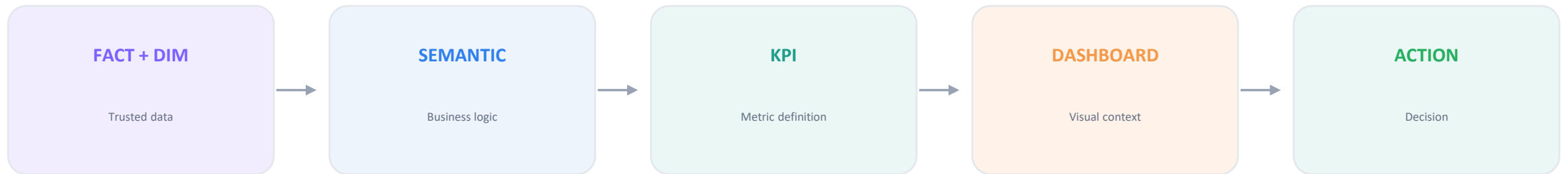
WHERE

d.quarter = 3 AND d.year IN (2025, 2026)
GROUP BY p.category, s.region, d.year;

Inti konsep, bukan syntax: fact menyediakan measure; dimension menyediakan konteks.

Dari Data Warehouse ke Dashboard

Warehouse menghasilkan data yang konsisten; semantic layer dan dashboard menerjemahkannya menjadi indikator bisnis.



Kesalahan Umum dalam Mendesain Data Warehouse

Sebagian besar masalah bukan karena tool, tetapi karena grain, definisi, dan governance yang tidak jelas.



Grain ambigu

Satu fact mencampur transaksi, invoice, dan summary.



Fact terlalu “wide”

Semua atribut ditempel ke fact sehingga sulit dipelihara.



Dimension tanpa histori

Perubahan customer/store membuat laporan masa lalu berubah.



KPI tidak disepakati

Finance dan Sales memakai rumus revenue berbeda.



No reconciliation

Tidak ada pembuktian total source = total warehouse.



Load tanpa lineage

Sulit menjawab angka berasal dari source mana dan transformasi apa.

Workshop 20 Menit: Rancang Star Schema Sederhana

Kelompok 3–4 mahasiswa. Pilih satu domain, lalu tetapkan grain, dimensions, dan measures.

PILIH KASUS

A. Food Delivery

B. E-Commerce

C. Rumah Sakit

D. Kampus / Akademik

1 Business process

Apa proses yang dianalisis?

2 Grain

Satu baris fact berarti apa?

3 Dimensions

Konteks: siapa, apa, kapan, di mana?

4 Measures

Angka apa yang dihitung?

5 KPI

Pertanyaan bisnis apa yang dijawab?

OUTPUT: gambar star schema + 3 pertanyaan analitik

Tugas Kasus: Marketplace “BelanjaKita”

Presentasikan rancangan model data warehouse yang mampu mendukung analitik penjualan, customer, seller, dan promosi.

A

Konteks

BelanjaKita memiliki data order, payment, seller, product, customer, promotion, dan shipment. Data berasal dari beberapa microservice dengan format yang berbeda.

B

Tugas

1. Tentukan grain fact utama.
2. Tentukan minimal 5 dimensions.
3. Tentukan minimal 5 measures.
4. Pilih strategi historisasi customer/seller.
5. Buat 4 KPI yang ingin dipantau.

C

Presentasi

Durasi 5 menit per kelompok.

Nilai dari:

- ketepatan grain
- konsistensi model
- relevansi KPI
- argumentasi
- keterbacaan diagram

Cek Pemahaman • Pertemuan 3

Jawab cepat tanpa membuka catatan. Diskusikan setelah semua selesai.

1

1. Apa perbedaan utama tujuan OLTP dan Data Warehouse?

2

2. Mengapa grain harus ditentukan sebelum dimensions dan measures?

3

3. Mana yang termasuk dimension: sales_amount, product_category, quantity, cost?

4

4. Apa fungsi surrogate key pada dimension?

5

5. Sebutkan satu contoh measure semi-additive dan jelaskan mengapa tidak boleh dijumlahkan terhadap waktu.

Kunci Cek Pemahaman

Gunakan jawaban ini sebagai titik diskusi; mahasiswa tetap perlu menjelaskan alasan dan contoh.

1 Jawaban 1

OLTP untuk transaksi operasional cepat; DW untuk analitik historis, integrasi, dan agregasi.

2 Jawaban 2

Grain menentukan tingkat detail. Tanpa grain yang jelas, dimension dan measure dapat memiliki makna yang tidak konsisten.

3 Jawaban 3

product_category adalah atribut dimension; sales_amount, quantity, dan cost adalah measures.

4 Jawaban 4

Surrogate key memberi identitas warehouse yang stabil, independen dari key source, dan mendukung historisasi/SCD.

5 Jawaban 5

Contoh: saldo rekening / inventory level. Nilai antar waktu adalah snapshot, sehingga tidak boleh dijumlahkan dari hari ke hari.

Ringkasan Pertemuan 3

Enam gagasan inti yang harus melekat sebelum masuk ke OLAP dan multidimensional analysis.

1

Warehouse memisahkan workload analitik dari transaksi dan menyimpan histori terintegrasi.

1

2

Karakteristik klasik: subject-oriented, integrated, time-variant, non-volatile.

2

3

OLTP dan OLAP berbeda tujuan, struktur, workload, dan pola akses.

3

4

Pemodelan dimensional dimulai dari grain, lalu dimensions dan facts.

4

5

ETL + data quality + historisasi membuat data dapat dipercaya dan konsisten.

5

6

Star schema menjadi fondasi yang mudah dipakai untuk semantic layer, OLAP, dan dashboard.

6

Persiapan Pertemuan 4

Topik berikutnya: OLAP & Multidimensional Data.

1 Data Cube

Dimension, measure, cell

2 Hierarchy

Year → Quarter → Month → Day

3 Slice & Dice

Memilih subset cube

4 Drill-Down / Roll-Up

Mengubah level detail

5 Pivot

Mengubah perspektif analisis

Bacaan awal: pahami kembali star schema, hierarki pada dimension, dan perbedaan fact vs dimension.

Referensi Utama

Bahan ajar disusun dari konsep data warehousing dan dimensional modeling yang mapan.

- 1 Kimball, R. & Ross, M. — **The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, 3rd Ed.**
- 2 Inmon, W. H. — Building the Data Warehouse.
- 3 Golfarelli, M. & Rizzi, S. — Data Warehouse Design: Modern Principles and Methodologies.
- 4 Sharda, R., Delen, D., & Turban, E. — Business Intelligence, Analytics, and Data Science: A Managerial Perspective.
- 5 Laudon, K. C. & Laudon, J. P. — Management Information Systems: Managing the Digital Firm.
- 6 Materi dan RPS Business Intelligence S1 Sistem Informasi — Pertemuan 3: Data Warehouse, ETL, OLTP vs OLAP.

Pertemuan 3 selesai • Lanjutkan dengan latihan pemodelan star schema sebelum masuk ke OLAP.