



**Project Based Internship**

# **Data Warehouse Foundation**

## **Principle of Data Warehouse**



## Daftar Isi

Exploration Source	3
A. Introduction	4
B. Concept	4
C. Fact Table	6
D. Dimension Table	7
E. Perbedaan Fact Table dan Dimension Table	7
F. Normalization	8
Studi Kasus : Perancangan ERD untuk Data Warehouse di Bank Digital	10
REFERENCE	13

## Exploration Source

### [Article]

[Memahami ERD, Model Data, dan Komponennya - Dicoding Blog](#)

[What is Entity Relationship Diagram \(ERD\)?](#)

[Mengenal Entity Relationship Diagram \(ERD\) | by M Luthfi Ridhwan | Medium](#)

### [Video]

[Database Normalization 1NF 2NF 3NF](#)

[Learn Database Normalization - 1NF, 2NF, 3NF, 4NF, 5NF](#)



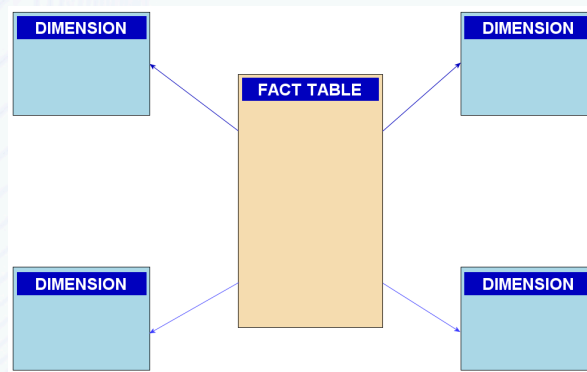
## A. Introduction

*Data Warehouse* merupakan sebuah tempat penyimpanan data yang terpusat. Konsep tersebut diambil dari sebuah konsep gudang penyimpanan (*warehouse*) dimana sebuah gudang dapat menampung apapun didalamnya. Dalam istilah data, *data warehouse* dapat menampung semua jenis data yang digunakan untuk sumber pembentukan keputusan berbasis data bagi sebuah perusahaan.

## B. Concept

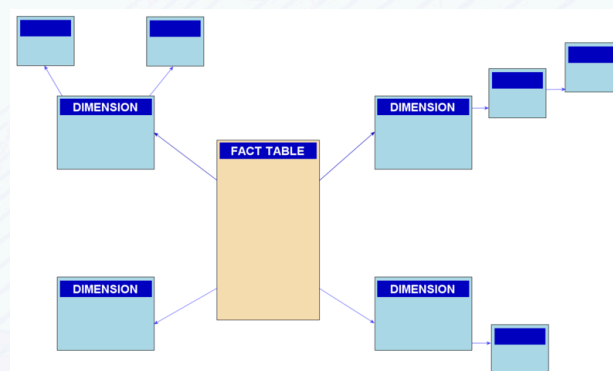
Dalam sebuah *database* tradisional, database digunakan untuk tempat penyimpanan data transaksi, dimana data tersebut akan terus bertambah seiring dengan bertambahnya jumlah transaksi dalam perusahaan. *Database* seperti ini umumnya ditangani secara terpisah dengan database operasional lainnya, dimana database ini sering dikenal dengan istilah *online transaction processing* (OLTP). Sistem OLTP ini digunakan untuk mendukung proses operasional bisnis yang bersifat harian karena hanya menampilkan kondisi perusahaan apa adanya. Sebuah data warehouse dapat menampung sistem OLTP ini ataupun sistem *online analytical processing* (OLAP) dimana data tersebut dapat digunakan untuk level analisis yang lebih tinggi dan pengambilan keputusan.

Untuk mempermudah pengambilan keputusan ini, model desain dari data warehouse ini dibuat sedemikian rupa agar mempermudah akses terhadap data dan juga untuk efisiensi biaya. Model ini umum disebut dengan *dimensional model* atau yang umum disebut dengan *star schema*.



Gambar 1.1 *Star Schema*

*Star Schema* terdiri atas dua entitas, yaitu *fact table* dan banyak *dimension table*. Karena bentuknya yang apabila disusun menyerupai bintang dimana terpusat pada sebuah *fact table* itu sendiri, maka sering disebut juga sebagai *star schema* (berbentuk seperti sebuah bintang). Namun tidak seluruh *data warehouse* berbentuk seperti *star schema*. Ada beberapa *data warehouse* yang memiliki bentuk lainnya, seperti *snowflake schema* dengan sebuah *fact table* dan *dimension table* yang bercabang.



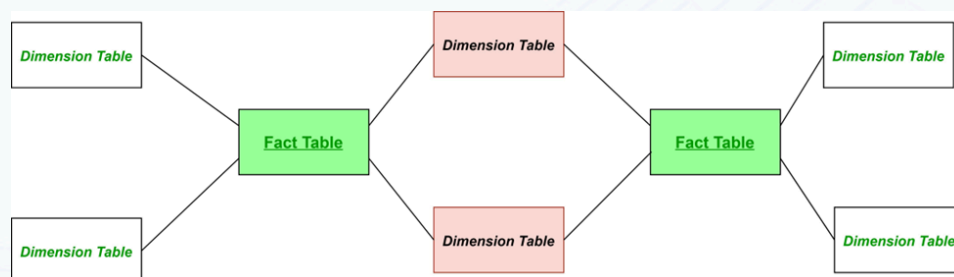
Gambar 1.2 *Snowflake Schema*

Mari coba lihat konsep lain dari dalam dunia *data warehouse*, yakni salah satunya adalah *Fact Constellation schema*.

Apa sih *Fact Constellation schema* itu?

Jadi *Fact Constellation* adalah suatu skema untuk merepresentasikan model multidimensional. Ini adalah kumpulan dari beberapa *fact table* yang memiliki beberapa tabel dimensi yang sama. Anda dapat melihatnya sebagai kumpulan dari beberapa skema bintang, dan oleh karena itu juga dikenal sebagai skema *Galaxy*. Ini adalah salah satu skema yang banyak digunakan untuk desain *Data Warehouse* dan jauh lebih kompleks dibandingkan dengan skema bintang dan skema *snowflake*. Untuk sistem yang kompleks, kita memerlukan *fact constellations*.

Skema *fact constellation* memungkinkan para analis untuk menganalisis berbagai aspek operasi kesehatan dengan efisien, sambil menggunakan kembali tabel dimensi bersama. Berikut adalah Struktur Umum dari *Fact Constellation Schema*.



Gambar 1.3 *Fact Constellation Schema*

### C. Fact Table

Sebelumnya kita sudah membahas mengenai bentuk dari *data warehouse* dimana penyusunannya dapat kita susun seperti sebuah *Star Schema*, *Snowflake Schema*, ataupun bentuk *schema* lainnya. *Fact table* dapat dikatakan merupakan inti dari skema yang dibentuk tersebut. *Fact table* sendiri



merupakan sebuah tabel *database* berisikan data, layaknya sebuah *dimension table*, namun isi data pada *fact table* ini merupakan data yang umumnya memiliki unsur waktu didalamnya dan disusun berdasarkan waktu tersebut. Jenis *fact table* ini biasanya ditemukan dalam sebuah data transaksi dimana peran dari kolom waktu tersebut menentukan analisa bisnis kedepannya.

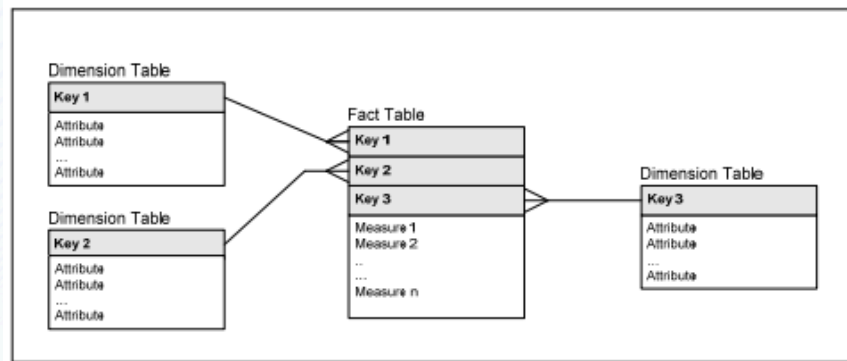
#### **D. Dimension Table**

*Dimension Table* sendiri merupakan sebuah tabel pada *database* yang berisikan data, sama seperti *fact table*. Bentuk tabel ini dikatakan sebuah *dimension table* karena pada umumnya tabel ini berisikan data yang mendefinisikan beberapa atau semua kolom yang ada pada *fact table*.

#### **E. Perbedaan Fact Table dan Dimension Table**

Perbedaan mendasar pada *fact table* dan juga *dimension table* ini adalah dimana pada *dimension table* umumnya tidak terdapat komponen waktu yang digunakan untuk mendefinisikan adanya data transaksi. Sebagai contoh, pada *fact table* kita memiliki informasi transaksi dimana transaksi tersebut dilakukan oleh *customer* dengan nomor *customer* 123. Pada umumnya, kita tidak menuliskan nama *customer* pada *fact table* dan hanya menuliskan sebuah kolom referensi ID nya saja (hal ini terkait dengan normalisasi tabel) dimana tujuannya untuk efisiensi penggunaan *data warehouse*. Untuk mendapatkan data siapakah *customer* 123, maka akan dilakukan proses penggabungan *fact table* dengan *dimension table customer* dimana pada *dimension table customer* ini memuat informasi nomor *customer* dan juga informasi nama dan informasi lainnya terkait *customer* tersebut. Informasi nomor *customer*, nama, dan

informasi lainnya ini umum kita sebut sebagai atribut. Kita baru akan mendapatkan informasi *customer* 123 apabila kita menggabungkan dimension table *customer* dengan *fact table* tersebut. Perhatikan gambar berikut ini untuk lebih jelasnya.



Gambar 1.4 Perbedaan *Fact Table* dan *Dimension Table*

## F. Normalization

Dalam pembentukan *dimension table* dan *fact table* pada sebuah data warehouse, umumnya kita akan mengenal istilah normalisasi. Normalisasi ini merupakan sebuah proses untuk mengorganisir data dalam sebuah database agar lebih mudah untuk melakukan analisa dan menuliskan query SQL. Teknik yang digunakan dalam melakukan normalisasi adalah dengan mengurangi data yang berulang dan juga data yang tidak terstruktur.

Dalam proses normalisasi, terdapat tiga bentuk normalisasi utama yang sering digunakan: 1 Normal Form (1NF), 2 Normal Form (2NF), dan 3 Normal Form (3NF). Bentuk-bentuk ini memberikan panduan tentang cara mengorganisir data agar menghindari redundansi dan anomali dalam database. 1NF mengharuskan setiap sel dalam tabel memiliki nilai atomik (tidak dapat dibagi lagi), 2NF memastikan bahwa setiap atribut non-kunci sepenuhnya bergantung



pada kunci utama, dan 3NF memastikan bahwa tidak ada ketergantungan transitif antar atribut non-kunci. Dengan mengikuti prinsip-prinsip ini, normalisasi membantu meningkatkan efisiensi penyimpanan data dan mengurangi peluang kesalahan dalam pemrosesan data.

Setelah proses normalisasi selesai, database dapat dibangun menggunakan model skema bintang atau snowflake, yang mencakup fact table sebagai inti yang terhubung dengan beberapa dimension table. Fact table berisi data numerik atau kuantitatif yang dapat dianalisis, sementara dimension table menyimpan atribut atau konteks tambahan. Dengan memahami dan menerapkan prinsip normalisasi, kita dapat menciptakan struktur data warehouse yang kokoh dan efisien untuk mendukung analisis data yang lebih baik.

Konsep yang digunakan adalah kita memecah sebuah data OLAP menjadi *fact table* dan banyak sekali *dimension table*, dan *dimension table* tersebut dipecah sampai menjadi bentuk yang paling sederhana. Berdasarkan tahapan normalisasi yang dilakukan, biasanya dapat dibagi kedalam beberapa bentuk seperti 1NF, ataupun 2NF, dan 3NF. Bentuk ini merupakan bentuk umum normalisasi data yang dilakukan dalam sebuah tabel dalam database. Untuk penjelasan lebih lengkap mengenai normalisasi sebuah tabel akan dibahas pada pembahasan artikel 5 mengenai topik normalisasi.

## Studi Kasus : Perancangan ERD untuk *Data Warehouse* di Bank Digital

Sebagai seorang konsultan Data Engineer di IDX Partners yang memiliki *client* sebuah Bank Digital, Anda diminta untuk merancang dan mengimplementasikan *data warehouse* yang dapat menyatukan dan menganalisis data dari berbagai sumber. Sedikit informasi Bank Digital tersebut merupakan sebuah lembaga keuangan modern, menghadapi kebutuhan untuk meningkatkan manajemen data mereka untuk mendukung keputusan yang lebih cepat dan akurat.

### SOLUSI :

#### 1. Tahap Perencanaan

IDX Partners bekerja bersama tim manajemen *Bank Digital* untuk memahami visi bisnis, tujuan, dan kebutuhan data kritis. Setelah pemahaman yang mendalam tercapai, mereka menyusun rencana proyek yang mencakup perincian mengenai struktur data yang diinginkan dan penggunaan ERD.

#### 2. Tahap Analisis

Anda dan Tim IDX Partners bekerja sama dengan departemen-departemen utama di Bank Digital untuk mengidentifikasi entitas dan hubungan kritis dalam operasi perbankan. Mereka menggali ke dalam struktur data yang ada dan mendefinisikan kebutuhan analisis, termasuk pelaporan keuangan, manajemen risiko, dan analisis kinerja produk.

#### 3. Desain ERD

Menggunakan hasil analisis, IDX Partners merancang *Entity Relationship Diagram* (ERD) yang mencerminkan struktur data yang diinginkan. ERD ini mencakup entitas seperti Nasabah, Rekening, Transaksi, dan Produk Keuangan, serta hubungan antar entitas yang merepresentasikan bagaimana data saling terkait.



#### 4. Pengembangan dan Implementasi

Berdasarkan ERD yang dirancang, IDX Partners membangun *data warehouse* dan mengintegrasikan data dari sumber yang berbeda seperti sistem perbankan inti, Product\_Keuangan (*platform e-channel*), dan data transaksi. Proses ETL dirancang untuk memastikan integritas dan konsistensi data.

#### 5. Pengujian

*Data warehouse* dan proses ETL diuji secara menyeluruh untuk memastikan akurasi data, kepatuhan terhadap regulasi, dan kinerja yang baik. Tim internal Bank Digital turut serta dalam pengujian untuk memastikan bahwa solusi memenuhi harapan dan kebutuhan bisnis.

#### 6. Implementasi dan Pemeliharaan



Setelah pengujian sukses, *data warehouse* diimplementasikan ke dalam lingkungan produksi Bank Digital. IDX Partners membantu dalam konfigurasi lingkungan produksi dan memberikan pelatihan kepada staf. IDX Partners juga memberikan dukungan pemeliharaan, memonitor kinerja data warehouse, dan menyediakan pembaruan sesuai kebutuhan Bank Digital.

Melalui kerjasama yang erat antara Bank Digital dan IDX Partners, implementasi *data warehouse* ini memberikan infrastruktur data yang kokoh, mendukung keputusan berbasis data, dan meningkatkan efisiensi operasional Bank Digital dalam era keuangan digital.



## REFERENCE

[Data Warehousing Systems: Foundations and Architectures](#)

[Snowflake schema - Wikipedia](#)

[How to Design a Data Warehouse- Best Practices and Examples](#)

[Bagaimana Cara Membuat ERD dan Contohnya - Dicoding Blog](#)

[Fact Constellation in Data Warehouse modeling - GeeksforGeeks](#)