

# Pembuatan Data Warehouse secara Berjenjang dari Data Transaksi dengan ETL Script PHP (Studi Kasus Perusahaan Payment Gateway)

*Creating a Data Warehouse in Stages from Transaction Data with PHP ETL Script  
(Case Study of Payment Gateway Company)*

Yuliana Aisyah<sup>1</sup>, Syaiful Anwar<sup>2</sup>, Samidi<sup>3</sup>

<sup>1,2,3</sup>Magister Ilmu Komputer, Fakultas Teknologi Informasi, Universitas Budi Luhur  
E-mail: <sup>1</sup>2111601726@student.budiluhur.ac.id, <sup>2</sup>2111601718@student.budiluhur.ac.id,  
<sup>3</sup>samidi@student.budiluhur.ac.id

## **Abstrak**

Setiap perusahaan payment gateway yang mendapat izin operasional Bank Indonesia (BI), harus melaporkan data transaksinya setiap bulan ke BI. Permasalahan terjadi karena database transaksi produksi tersebar di banyak database, dimana database dipisahkan berdasarkan channel (merchant) dan payment type (Credit card, VA, retail store, e-wallet), hal ini menyulitkan saat pembuatan laporan. Penelitian ini ditujukan untuk membuat data warehouse yang menggabungkan seluruh database dengan menggunakan script PHP yang melakukan proses ETL (Extract, Transform, dan Load), dengan cara berjenjang 2 tahap, yaitu tahap 1 melakukan backup seluruh database dari banyak server ke satu server dengan 3 tujuan sekaligus yaitu backup transaksi, meningkatkan performance database production dan membantu proses rekonsiliasi dan tahap 2 melakukan pembuatan data warehouse dari semua database backup dari tahap 1. Penelitian yang digunakan adalah kuantitatif verifikatif terhadap table-table yang diperlukan pada system OLTP (Online Transaction Processing). Pengujian data warehouse dilakukan secara kuantitatif dengan membandingkan hasil summary dari data warehouse dengan gabungan summary semua database aslinya dan secara verifikatif oleh petugas report melalui User Acceptance Test. Hasil dari penelitian adalah sebuah script untuk backup data transaksi dari semua database dari banyak server ke satu server, sebuah script PHP yang berfungsi sebagai pelaksana ETL dan sebuah data warehouse.

Kata kunci : payment gateway, regulator BI, data warehouse, ETL, Script PHP, Start Schema, user acceptance test .

## **Abstract**

*Every payment gateway company that has an operational permit from Bank Indonesia (BI), must report its transaction data every month to BI. The problem occurs because the production transaction database is spread over many databases, where the databases are separated by channel (merchant) and payment type (Credit card, VA, retail store, e-wallet), this makes it difficult to generate reports. This research is aimed at creating a data warehouse that combines all databases using PHP scripts that carry out ETL (Extract, Transform, and Load) processes, in a tiered manner in 2 stages, namely stage 1 backing up the entire database from multiple servers to one server with 3 objectives at the same time, namely transaction backups, improving database production performance and assisting the reconciliation process and stage 2 creating a data warehouse from all backup databases from stage 1. The research used is quantitative verification of the tables needed in the OLTP (Online Transaction Processing) system. Data warehouse testing is carried out quantitatively by comparing the summary results from the data warehouse with a combined summary of all the original databases and verified by a report officer through a User Acceptance Test. The results of the research are a script for backing up transaction data from all databases from multiple servers to one server, a PHP script that functions as an ETL executor and a data warehouse.*

*Keywords: payment gateway, BI regulator, data warehouse, ETL, PHP script, Start Schema, user acceptance test.*

## 1. PENDAHULUAN

Perusahaan Payment gateway yang menjadi objek penelitian ini telah bergerak di bidang payment gateway sejak beberapa tahun yang lalu, setahun terakhir perusahaan mengajukan ijin resmi penyelenggaraan payment gateway ke BI, dikarenakan akan menyelenggarakan gateway pembayaran kartu kredit dan kartu debit yang terkoneksi langsung dengan bank, tidak menggunakan jasa payment gateway pihak ke tiga lagi. Ternyata seiring dengan proses perijinan tersebut, bank Indonesia meminta semua *payment type* selain kartu kredit dan kartu debit yang akan dilakukan koneksi langsung ke bank, juga menjadi bagian dari *payment type* yang harus dilaporkan transaksinya ke BI. Penambahan *payment channel* lain dan *payment type* lain setelah proses perijinan juga termasuk yang harus dilaporkan ke BI. Perusahaan sudah memiliki aplikasi yang berfungsi sebagai gateway dari masing-masing payment channel dan payment type itu sendiri-sendiri, sehingga databasenya pun terpisah-pisah juga sesuai aplikasinya. Proses pelaporan ke Bank Indonesia bersifat summary dari transaksi yang harus mempunyai format yang sama untuk seluruh data payment channel, oleh karenanya setiap mau dilakukan pelaporan, data yang diperlukan harus diambil dari masing-masing database, dimana tidak semua database mempunyai struktur table yang sama, sehingga perlu dilakukan penyesuaian-penyesuaian, untuk diseragamkan. Hal ini tentunya tidak efisien dan memakan waktu, oleh karenanya perlu dibuat data warehouse yang secara otomatis menggabungkan data-data dari multi database tersebut untuk memudahkan saat pelaporan.

Ada 55 database dari aplikasi lama yang struktur tablenya hampir sama yang dibuat lebih dari 5 tahun, 1 database yang dibuat sekitar 4 tahun yang lalu dan 1 database yang dibuat 1 tahun belakangan, yang struktur tablenya ada perubahan karena ada upaya untuk menggabungkannya menjadi satu database *multichannel* (merchant/biller) dan *multitype* pembayaran, hal ini dibuat karena database ini diakses oleh aplikasi terbaru yang dibuat untuk memudahkan kalau ada penambahan channel dan payment type baru. Dengan adanya penambahan aplikasi baru ini tidak membuat aplikasi dan database lama digantikan, tapi tetap masih berjalan, karena mitra-mitra lama masih tetap dilayani dan tidak dilakukan proses migrasi, mengingat transaksinya sudah cukup besar dan berjalan lama, perubahan aplikasi menyebabkan perubahan protocol, diperkirakan akan membuat mitra lama kerepotan untuk melakukan perubahan. Aplikasi baru ditujukan untuk mitra-mitra baru yang akan berintegrasi dengan perusahaan. Penelitian ini ditujukan untuk membuat data warehouse yang menggabungkan 57 database yang terdiri dari 55 database dari aplikasi lama dan 2 database dari aplikasi yang baru sehingga proses pembuatan laporan menjadi efisien, efektif dan otomatis. Untuk itu dilakukan penelitian tentang bagaimana menggabungkan data dari banyak database tersebut menjadi sebuah data warehouse. Karena salah satu peneliti sering menggunakan script PHP didalam pembuatan program, setelah dipelajari tentang proses yang ada di dalam proses ETL dan membaca literatur jurnal yang melakukan pembuatan ETL sendiri dengan Bahasa tertentu, maka peneliti mencoba membuat proses ETL ini menggunakan script PHP.

Studi literature yang dilakukan untuk membantu di dalam pembuatan data warehouse ini adalah mempelajari Sistem Extract-Transform-Load (ETL) yang merupakan dasar (pondasi) dari data warehouse. Sistem ETL yang dirancang dengan baik mengekstrak data dari sumbernya sistem, menjaga kualitas data dan standar konsistensi, menyesuaikan data sehingga sumber yang terpisah-pisah dapat digunakan bersama, dan akhirnya mengirimkan data dalam format siap presentasi sehingga pengembang aplikasi dapat membangun Aplikasi dan pengguna akhir dapat membuat keputusan[1], kemudian mempelajari metodologi yang ditulis oleh Kimball di dalam pembuatan desain model data warehouse yang terdiri atas 4 langkah serta mempelajari jurnal-jurnal yang berhubungan dengan pembuatan data warehouse yang didalamnya ada proses ETL, sebagai bahan acuan untuk penulis dalam pembuatan *data warehouse* yang dibutuhkan supaya sesuai dengan harusnya.

Pada jurnal terdahulu tentang proses ETL dalam pembuatan *data warehouse* salah satunya yang dilakukan oleh Adi Supriatna dengan judul perancangan data warehouse pada perpustakaan bina sarana informatika, di jurnal tersebut disebutkan bagaimana proses ETL

dilakukan dengan bantuan software *Pentaho Kettle Data Integration*[2]. Jurnal lain yang membahas tentang ETL yang merupakan salah satu proses inti dari pembuatan data warehouse adalah jurnal yang berjudul *Penerapan Extraction-Transformation-Loading (ETL) Dalam Data Warehouse (Studi Kasus : Departemen Pertanian)* yang dibuat oleh Rahmadi Wijaya tahun 2016. Di dalam jurnal ini proses pembuatan ETL menggunakan fitur dari DBMS SQL server yaitu Data Transformation Service (DTS) menggunakan ActiveX Script Task dalam bahasa VBScript[3]. Jurnal lain yang diteliti adalah jurnal yang ditulis oleh I Gusti Ngurah Agung Trisna Putra (2020) dengan judul *IMPLEMENTASI ETL DATA WAREHOUSE DENGAN KONSEP FITUR METADATA DAN CLEANSING DATA*, melakukan proses ETL menggunakan program yang ditulis dalam bahasa *python*[4]. Penelitian lain yang berjudul *Rancang Bangun Engine ETL Data Warehouse dengan Menggunakan Bahasa Python* yang dilakukan oleh Putra, I Made Suwija, dkk, juga melakukan proses pembuatan ETL script nya menggunakan Bahasa *python*. [5]. Penelitian dengan ETL lain yang berjudul *PERANCANGAN DATA WAREHOUSE dan DASHBOARD PT. JAYA TEKNIK* melakukan proses ETL dengan software *Pentaho Kettle*[6]. Penelitian yang berjudul, *Modeling data mart using ETL (extract, transform, load) webservice concept on feeder with a dashboard*, membuat ETL menggunakan script PHP yang sama dengan yang dilakukan penelitian ini.[7]. Penelitian lain yang berjudul, *Pengembangan Data Warehouse menggunakan Metode Kimball ( Studi Kasus : Ekspor & Impor Fauna dan Flora Hias Air Laut )*, membahas bagaimana membuat data warehouse dengan Langkah-langkah yang ditulis oleh Kimpball seperti yang dilakukan dalam penelitian ini, disamping itu penulisnya melakukan uji kecepatan dari proses ETL yang dibuat menggunakan Talend Studio for Data Integration dan Windows Task[8]. Penelitian yang berjudul, *Pemodelan Datawarehouse Koperasi Dengan Metode Four-Step Kimball*, membahas cara membuat data warehouse dengan menggunakan metode Kimball yang disingkat menjadi hanya 4 langkah seperti yang dilakukan dalam penelitian ini.[9]

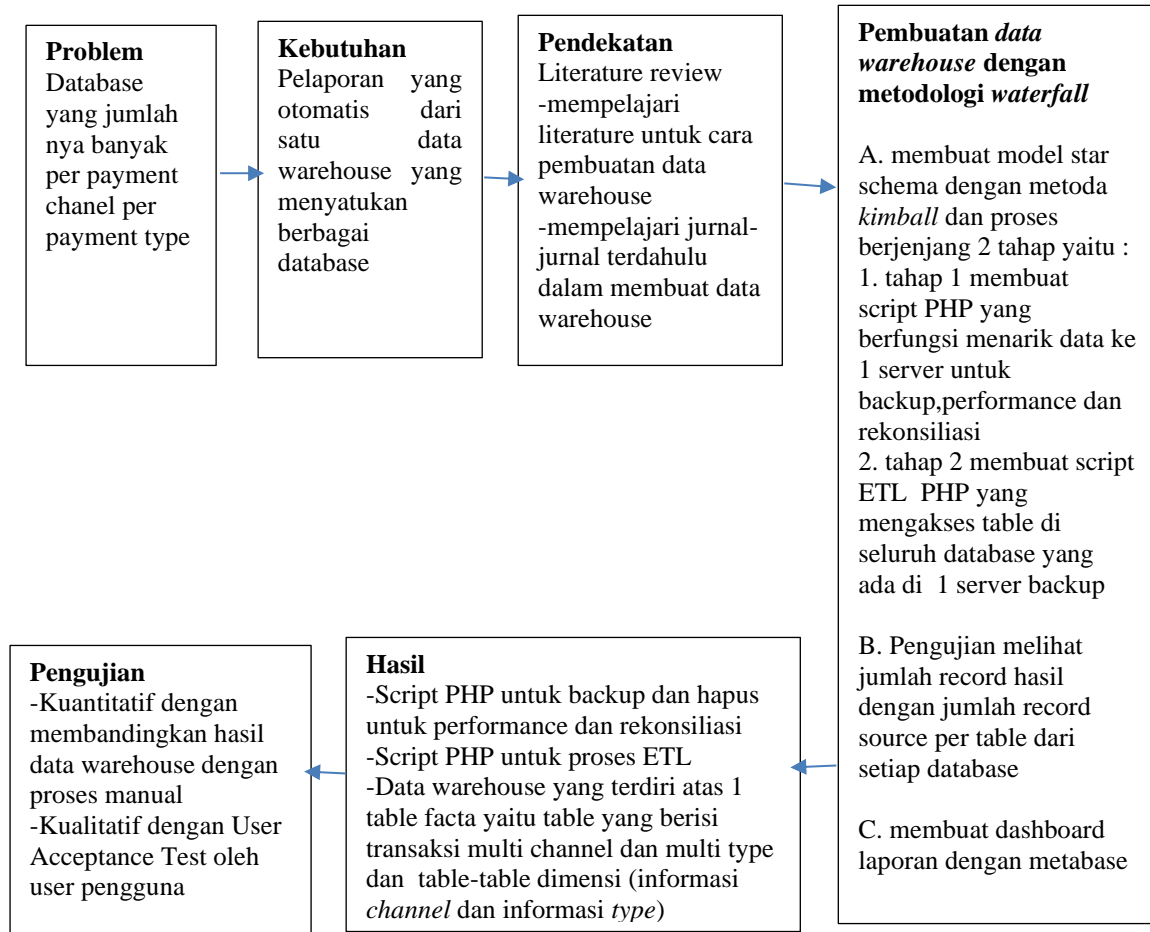
Perbedaan jurnal ini dengan jurnal-jurnal sejenis sebelumnya adalah terletak pada pembuatan proses ETL yang dilakukan menggunakan program script PHP yang dibuat oleh sendiri, tidak menggunakan *tools* umum yang ada di database atau software yang sudah ada lainnya, disamping itu digunakan proses scheduler untuk melakukan proses ETL periodic menggunakan tools yang ada di sistem operasi linux yaitu crontab. Hal lain yang menjadi perbedaan dari penelitian ini adalah data warehouse dibuat berjenjang 2 tahap karena pertimbangan peneliti adalah untuk dapat melaksanakan 3 *task* sekaligus, *task* pertama adalah melakukan proses backup seluruh transaksi di semua database ke satu server dalam bentuk file sql dan database dan melakukan penghapusan data transaksi di database production yang berusia di atas 6 bulan, supaya data transaksi yang ada di database produksi berkurang sehingga akan meningkatkan performansi database saat proses searching dan update, dan task 3 menyediakan data untuk keperluan rekonsiliasi transaksi H+1, kemudian dilanjutkan dengan tahap ke dua yaitu pembuatan data warehouse yang berasal dari semua database yang ada di satu server backup untuk pelaporan ke BI dan pelaporan lainnya misalnya untuk keperluan analisa bisnis. Setelah terbentuk data warehouse, proses pelaporannya dilakukan menggunakan software metabase untuk kemudian disimpan menjadi file akhir yang berbentuk excel yang diberikan setiap bulan ke BI. Pengambilan datanya menggunakan populasi (data sesungguhnya), dan dilakukan dengan query langsung ke semua database yang akan dilakukan oleh proses ETL menjadi data warehouse.

## 2. METODE PENELITIAN

Metodologi yang digunakan oleh penulis dalam pembuatan data warehouse adalah metodologi *waterfall* yang merupakan salah satu metodologi *SDLC (software development live cycle)*. Penelitian jurnal ini merupakan penelitian *experimental* karena database yang diproses adalah database yang ada di perusahaan, kemudian dicoba berbagai cara untuk pembuatan data warehouse yang cocok dengan kebutuhan perusahaan, kuantitatif karena hasil data warehouse menyajikan angka-angka yang berupa jumlah transaksi dan verifikasi karena hasil data warehouse akan diverifikasi apakah angka-angka yang dihasilkannya sudah sesuai dengan angka-angka dari proses manual.

### 2.1. Kerangka Berpikir

Berikut gambaran urutan alur proses yang dilakukan dalam penelitian ini, sebelum dilakukan pembuatan data warehouse dimulai dari identifikasi masalah atau problem:



Gambar 1 Alur Proses Penelitian

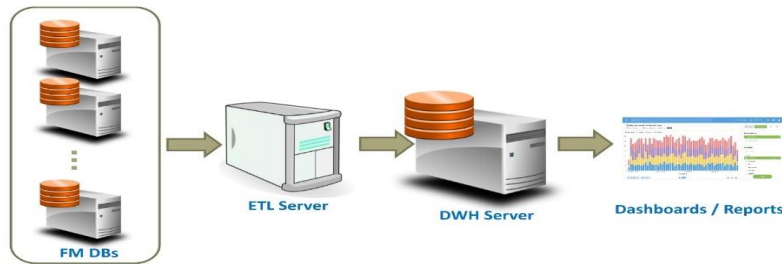
### 2.2. Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah database transaksi dari sebuah perusahaan payment gateway, data berasal dari 57 database perusahaan yang dikumpulkan secara berjenjang 2 tahap menggunakan proses query sebelum membentuk sebuah data warehouse. Query pertama dilakukan terhadap tabel-tabel transaksi harian yang diambil dari semua database dengan melakukan proses backup menggunakan script PHP ke dalam satu server, setelah itu query kedua menggunakan script PHP yang melakukan proses ETL terhadap data-data yang ada di satu server untuk dibuat menjadi data warehouse. Script yang dibuat dijalankan di dua server, server pertama untuk menjalankan query pertama, dan server kedua untuk menjalankan query ke dua.

### 2.3. Model Data Warehouse

Model data warehouse yang dibangun dengan metodologi waterfall dapat digambarkan sebagai berikut:

### DWH Schema



Gambar 2 Model dataware house

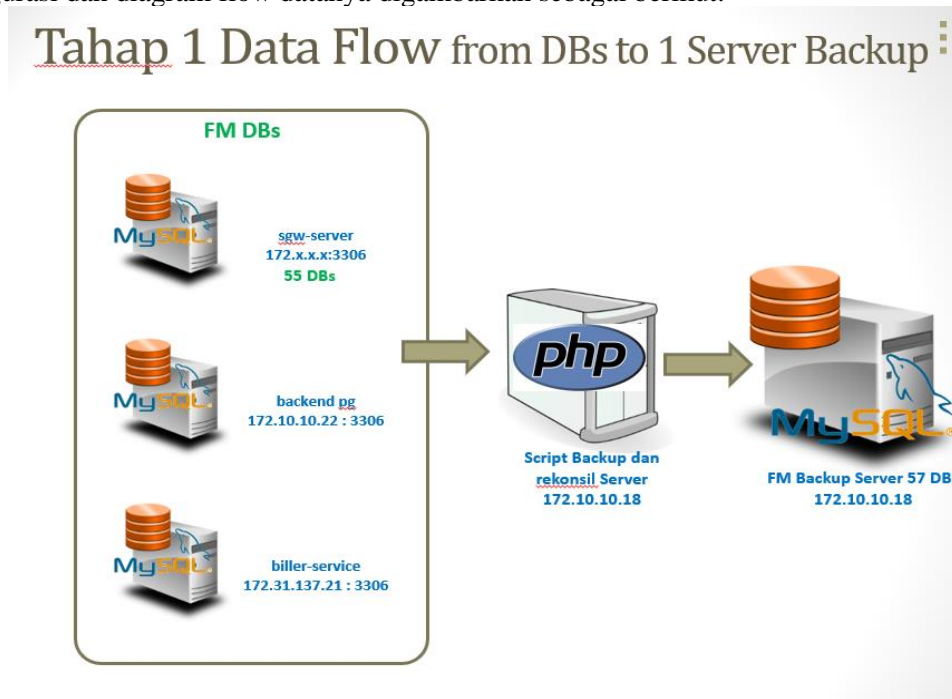
Pembuatan model data warehouse di atas dilakukan secara berjenjang 2 tahap sebagai berikut:

#### 2.3.1. Tahap 1

Pada tahap ini dilakukan proses membackup data transaksi dari database-database di semua server dalam periode tertentu ke satu server dalam bentuk file sql dan database :

1. menyiapkan 1 server dengan hdd yang besar untuk backup seluruh transaksi
2. membuat script PHP untuk melakukan proses backup, untuk masing-masing database di buatkan script dalam folder masing-masing, untuk memudahkan kalau ada error di satu proses terhadap satu database di dalam mentracenya, jadi tidak mempengaruhi proses terhadap database lain.

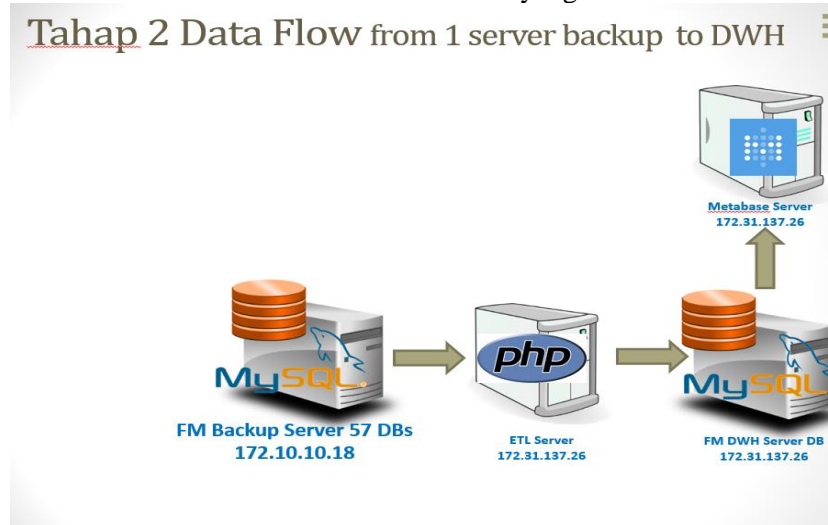
Konfigurasi dan diagram flow datanya digambarkan sebagai berikut:



Gambar 3 Tahap 1 proses backup semua database server ke database-database di 1 server

### 2.3.2. Tahap 2

Pembuatan *data warehouse* dari database-database yang ada di 1 server backup.



Gambar 4 Tahap 2 pembentukan DWH dari database-database di 1 server dari tahap 1

### 2.4. Langkah-Langkah Pembuatan Data Warehouse

Untuk membuat model Data Warehouse (DWH) di atas dilakukan dengan langkah-langkah yang sesuai dengan metodologi *waterfall* dari *SDLC* yang bersifat serial, dimulai dari proses perencanaan untuk mengidentifikasi apa saja yang perlu dipersiapkan mulai dari mempelajari kebutuhan dan kondisi sistem yang ada, proses analisa untuk mempelajari lebih detail kondisi sistem yang ada meliputi database-database untuk dibuat desain data warehousesnya, proses desain untuk melakukan desain dari data warehouse meliputi desain table-table yang ada di data warehouse, ETL dsb., dan implementasi pada sistem untuk menerapkan apa yang ada di dalam desain menjadi terealisasi dan terinstall di server sehingga siap digunakan.

## 3. HASIL DAN PEMBAHASAN

Pada bagian ini akan dijelaskan pembahasan dari langkah-langkah yang dilakukan yang dilakukan dalam penelitian yaitu :

### A. Proses perencanaan

Dimulai dengan identifikasi masalah dimana terdapat kebutuhan untuk membuat laporan yang lebih efisien, efektif dan otomatis, dikarenakan proses pelaporan merupakan proses rutin, sehingga kalau selalu dibuat dengan cara manual, membuat pembuat laporan melakukan pekerjaan yang seharusnya bisa disederhanakan. Metode pengumpulan data dilakukan dengan mengambil data transaksi dari seluruh database transaksi secara langsung milik dari perusahaan payment gateway tempat salah satu peneliti bekerja. Dengan adanya 55 database lama dan 2 database baru yang harus dijadikan satu kesatuan dalam pembuatan laporan, hal ini tentu membuat pengerjaan laporan menjadi pekerjaan yang cukup merepotkan. Untuk mengatasi masalah di atas dibuat perencanaan untuk membuat sebuah *data warehouse*, sehingga software pelaporan yang bersifat OLAP (online analytical processing) hanya mengacu kepada satu database saja yaitu *data warehouse*.

### B. Proses Analisa

Proses selanjutnya yang dilakukan adalah melakukan analisa menggunakan metode *analisis deskriptif verifikatif* terhadap table-table yang ada di dalam 57 database (55 database lama dan 2 database baru), dipisahkan antara table yang jenisnya master atau setting, dan table transaction.

Dari proses analisa ini dihasilkan daftar table-table dari database-database yang akan dilakukan proses ETL untuk digabungkan menjadi *data warehouse* sebagai berikut:

no	database name	table name	datetime field name	backup file folder					
					26	sgw_mkm_fp	logtransaction	servitimereqinfo	trans_mkm_fp
1	sgw_idv_aj	logtransaction	servitimereqinfo	trans_indovision_aj	27	sgw_mkm_gkm	logtransaction	servitimereqinfo	trans_mkm_gkm
2	sgw_idv_alfamart	logtransaction	servitimereqinfo	trans_indovision_alfamart	28	sgw_mkm_idm	logtransaction	servitimereqinfo	trans_mkm_idm-a
3	sgw_idv_bkp	logtransaction	servitimereqinfo	trans_indovision_bkp	29	sgw_mkm_mkm	logtransaction	servitimereqinfo	trans_mkm_mkm
4	fmpostranslog_idv_bms	logtransaction	servitimereqinfo	trans_indovision_bms	30	sgw_mkm_mpay	logtransaction	servitimereqinfo	trans_mkm_mpay
5	sgw_idv_chp	logtransaction	servitimereqinfo	trans_indovision_chp	31	sgw_mkm_np	logtransaction	servitimereqinfo	trans_mkm_np
6	sgw_idv_fp	logtransaction	servitimereqinfo	trans_indovision_fp	32	sgw_mkm_pg	logtransaction	servitimereqinfo	trans_mkm_pg
7	sgw_idv_gkm	logtransaction	servitimereqinfo	trans_indovision_gkm	33	sgw_mkm_pgd	logtransaction	servitimereqinfo	trans_mkm_pgd
8	fmpostranslog_idv_gsp	logtransaction	servitimereqinfo	trans_indovision_gsp	34	sgw_mkm_pos	logtransaction	servitimereqinfo	trans_mkm_pos
9	sgw_idv_idm	logtransaction	servitimereqinfo	trans_idv_idm	35	sgw_mkm_web	logtransaction	servitimereqinfo	trans_mkm_web
10	fmpostranslog_idv_kopnus	logtransaction	servitimereqinfo	trans_indovision_knu	36	sgw_mkm_xik	logtransaction	servitimereqinfo	trans_mkm_xik
11	sgw_idv_kpo	logtransaction	servitimereqinfo	trans_indovision_kpo	37	sgw_mncf_aj	logtransaction	servitimereqinfo	trans_mncfinance_aj
12	sgw_idv_mkm	logtransaction	servitimereqinfo	trans_indovision_mkm	38	sgw_mncf_alf	log_transaction_inout	ServerTime_inc	trans_mncfinance_alf
13	sgw_idv_mpay	logtransaction	servitimereqinfo	trans_indovision_mpay	39	sgw_mncf_fp	logtransaction	servitimereqinfo	trans_mncfinance_fp
14	fmpostranslog_idv_pac	logtransaction	servitimereqinfo	trans_indovision_pac	40	sgw_mncf_idm	logtransaction	servitimereqinfo	trans_mncfinance_idm
15	sgw_idv_pgd	logtransaction	servitimereqinfo	trans_indovision_pgd	41	sgw_mncf_mpay	logtransaction	servitimereqinfo	trans_mncfinance_mpay
16	fmpostranslog_idv_pos	logtransaction	servitimereqinfo	trans_indovision_pos	42	sgw_mncf_pgd	logtransaction	servitimereqinfo	trans_mncfinance_pgd
17	sgw_idv_sjm	logtransaction	servitimereqinfo	trans_indovision_sjm	43	sgw_mncf_pos	logtransaction	servitimereqinfo	trans_mncfinance_pos
18	fmpostranslog_idv_smart	logtransaction	servitimereqinfo	trans_indovision_smt	44	sgw_mncf_xik	logtransaction	servitimereqinfo	trans_mncfinance_xik
19	sgw_idv_fmuser	logtransaction	servitimereqinfo	trans_indovision_web	45	sgw_mncva_fp	logtransaction	servitimereqinfo	trans_mncnow_fp
20	fmpostranslog_idv_xlk	logtransaction	servitimereqinfo	trans_indovision_xlk	46	sgw_mncva_np	logtransaction	servitimereqinfo	trans_mncnow_np
21	sgw_mkm_aj	logtransaction	servitimereqinfo	trans_mkm_aj	47	sgw_mncshop_aj	logtransaction	servitimereqinfo	trans_mncshop_aj
22	sgw_mkm_alf	logtransaction	servitimereqinfo	trans_mkm_alf	48	sgw_mncshop_alf	logtransaction	servitimereqinfo	trans_mncshop_alf
23	sgw_mkm_bca	logtransaction	servitimereqinfo	trans_mkm_bca	49	sgw_mncshop_fp	logtransaction	servitimereqinfo	trans_mncshop_fp
24	sgw_mkm_bkp	logtransaction	servitimereqinfo	trans_mkm_bkp	50	sgw_mncshop_idm	logtransaction	servitimereqinfo	trans_mncshop_idm
25	sgw_mkm_fin	logtransaction	servitimereqinfo	trans_mkm_fin	51	sgw_mncshop_mpay	logtransaction	servitimereqinfo	trans_mncshop_mpay
					52	sgw_mncshop_np	logtransaction	servitimereqinfo	trans_mncshop_np
					53	sgw_mncshop_pos	logtransaction	servitimereqinfo	trans_mncshop_pos
					54	sgw_mncf_alf	tb_logtransaction_reversal	servitimereversal	trans_mncfinance_alf
					55	sgw_finnet_mncbank	Finnet42	tPayment	trans_finnet_mncbank

Gambar 5 Daftar table

**Database : backend\_pg**

- backend\_pg\_\_logdataorderid
- backend\_pg\_\_logtransaction

**Database : biller\_service**

- biller\_service\_\_logtransaction
- biller\_service\_\_logtransaction\_cekstatus
- biller\_service\_\_logtransaction\_update
- biller\_service\_\_master\_biller\_product\_src

**Faderated table :**

- final\_finnet\_\_final\_finnet\_202104
- final\_mkm\_\_final\_mkm\_202104
- final\_mncfinance\_\_final\_mncf\_202104
- final\_mncshop\_\_final\_mncshop\_202104

**C. Proses Desain**

Proses selanjutnya setelah proses analisa adalah mendesain model table yang ada di *data warehouse* yang berbentuk star schema. Untuk membentuk model table star schema digunakan metode *kimball* yang lebih sederhana yang terdiri atas 4 langkah. Pendekatan metode Kimball dalam pembangunan model *data warehouse*, dimulai dengan membangun tabel yang terdiri dari tabel fakta dan dimensi. Tabel fakta berisikan metrik yang biasa data recordnya berasal dari tabel transaksi di relational database dan tabel dimensi berisikan atribut yang data recordnya berasal dari tabel master di relational database. Pendekatan ini tidak memperhatikan aturan normalisasi. Untuk menentukan table fakta , terlebih dahulu dibuat pemilihan proses dan pemilihan grain.

**C.1 Pemilihan Proses**

Proses bisnis yang dikaji dalam penelitian ini adalah bisnis inti perusahaan payment gateway yaitu menyediakan layanan pembayaran untuk *multi channel (merchant atau biller)* kepada pelanggannya, layanan pembayaran ini harus mencakup semua type pembayaran (*multi type*) misalnya kartu kredit, kartu debit, direct debit, online payment, wallet, virtual account dan retail store. Karena telah mendapatkan ijin usaha *payment gateway* dari regulator **Bank Indonesia**, perusahaan wajib melaporkan semua transaksi penjualan yang berasal dari multi channel dan multi type pembayarannya ke bank Indonesia. Karena kondisi yang ada untuk setiap channel dan

setiap type pembayaran mempunyai database sendiri-sendiri, proses pelaporan menjadi tidak efisien dan efektif karena proses dilakukan secara manual dengan akses ke setiap database. Untuk itu ada kebutuhan untuk membuat sebuah *data warehouse* yang menyatukan semua database tersebut supaya pelaporan menjadi efisien, efektif dan otomatis.

Tabel 1 Pemilihan Proses

Proses Bisnis	Deskripsi	Fungsi yang terkait
Pelaporan data transaksi multi channel dan multi type pembayaran	Memberikan laporan bulanan data transaksi pembayaran multi channel dan multi type ke BI	Semua data transaksi pembayaran dari seluruh channel dan seluruh type pembayaran yang berasal dari semua aplikasi yang ada

### C.2 Pemilihan Grain

Untuk membuat laporan ke Bank Indonesia, harus memenuhi format laporan standar yang diberikan bank Indonesia yaitu laporan transaksi yang terdiri atas: Payment type, Payment channel, jumlah transaksi, nilai dan jumlah user nya per bulan. Jadi grain yang digunakan adalah payment type, payment channel, jumlah transaksi, nilai transaksi dan jumlah user.

Tabel 2 Pemilihan Grain

Grain	Deskripsi	Proses bisnis yang terkait
Informasi transaksi pembayaran	Informasi transaksi pembayaran yang disajikan per payment channel meliputi: payment type, payment channel, jumlah transaksi, nilai transaksi dan jumlah user.	Data transaksi pembayaran multi channel multi type

### C.3 Penentuan Table Dimensi

Menentukan table dimensi yang berhubungan dengan fakta yang akan mengatur isi untuk bertanya mengenai fakta

Tabel 3 Table Dimensi

Table Dimensi	Deskripsi	Grain
Dim_PaymentType	Detail dari semua type pembayaran tempat terjadinya transaksi mempunyai atribut: payment_type dan payment_name	Informasi transaksi pembayaran
Dim_PaymentChannel	Detail dari semua merchant /biller yang melakukan pembayaran dengan jasa dari perusahaan payment gateway mempunyai atribut: channel code dan channel_name	Informasi transaksi pembayaran
Dim_Transaksi 1-57	Detail dari log transaksi per payment chanel untuk setiap payment type	Informasi transaksi pembayaran



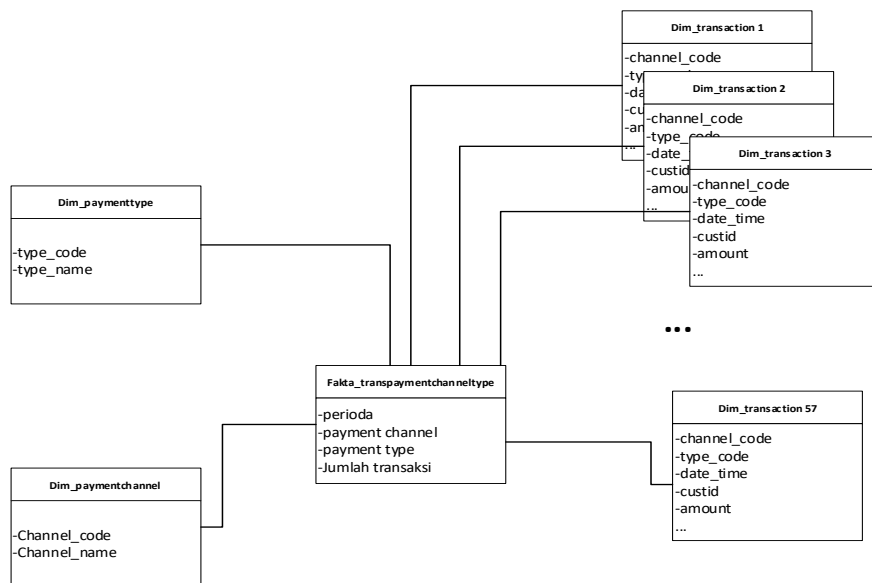
C.4. Penentuan Table Fakta

Fakta yang ada di dalam data warehouse mengimplementasikan grain yang sudah ditentukan sebelumnya.

Tabel 4 Table Fakta

Table Fakta	Deskripsi	Dimensi
Fakta_TransPaymentChanelType (dm_summary_use)	Berisikan informasi pembayaran multi channel yang meliputi payment channel, payment type, jumlah transaksi, nilai transaksi dan jumlah user.	Dim PaymentChannel dan Dim PaymentType

Berikut gambaran diagram ER (Entity Relationship) dari data warehouse yang terbentuk:



Gambar 5 Diagram ER dari dataware house

D. Proses Implementasi

Hasil yang didapat dari Implementasi pembentukan data warehouse adalah script PHP untuk proses backup semua transaksi dari semua database, script PHP yang berfungsi sebagai ETL dan Data warehouse yang berbentuk star schema yang terdiri dari 1 table fakta yang berisi summary transaksi per channel dan per type, dan table-table dimensi yang berisi informasi channel dan informasi type pembayaran. Untuk pembentukan table fakta di bantu dengan scheduler yang menjalankan script untuk mengambil semua data yang dimasukan ke dalam table-table kategori dimensi dan kemudian dibuatkan summarynya per hari dan disimpan di dalam table fakta (dm\_summary\_use). Berikut adalah screen capture-screen capture hasil yang dibuat dalam 2 tahap pembuatan data warehouse.

```
#!/bin/bash
# Script untuk backup data ke folder backup
# Author: [Nama Anda]
# Date: [Tanggal]

# Variabel
SOURCE="/mnt/c:/Users/username/Desktop/backup"
TARGET="/mnt/d:/Users/username/Desktop/backup"

# Fungsi untuk backup file
backup_file() {
    local file="$1"
    local dest="$2"
    cp -r "$file" "$dest"
}

# Fungsi untuk backup folder
backup_folder() {
    local folder="$1"
    local dest="$2"
    cp -r "$folder" "$dest"
}

# Backup file
backup_file "$SOURCE" "$TARGET"

# Backup folder
backup_folder "$SOURCE" "$TARGET"

# Hapus backup lama
rm -rf "$TARGET"

# Selesai
echo "Backup selesai."
exit 0
```

Gambar 6 screen shoot script backup

```
#!/bin/bash
# Script untuk membuat table dm_summary_use sebagai table facta dari table-table dimensi
# Author: [Nama Anda]
# Date: [Tanggal]

# Variabel
SOURCE="/mnt/c:/Users/username/Desktop/backup"
TARGET="/mnt/d:/Users/username/Desktop/backup"

# Fungsi untuk backup file
backup_file() {
    local file="$1"
    local dest="$2"
    cp -r "$file" "$dest"
}

# Fungsi untuk backup folder
backup_folder() {
    local folder="$1"
    local dest="$2"
    cp -r "$folder" "$dest"
}

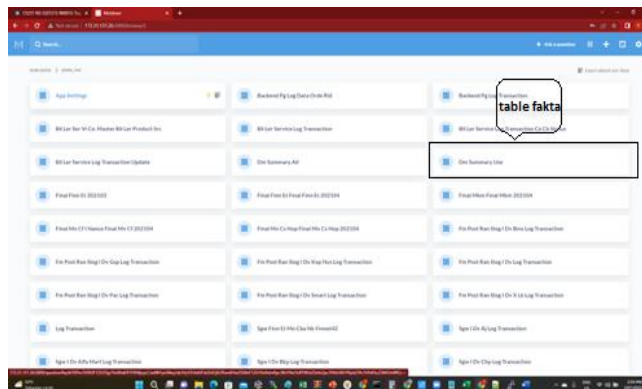
# Backup file
backup_file "$SOURCE" "$TARGET"

# Backup folder
backup_folder "$SOURCE" "$TARGET"

# Hapus backup lama
rm -rf "$TARGET"

# Selesai
echo "Backup selesai."
exit 0
```

Gambar 7. Screenshot script untuk membuat table *dm\_summary\_use* sebagai table facta dari table-table dimensi.



Gambar 8. Table-table dimensi dan table fact yang dihasilkan dari script ETL

### E. Pengujian

Pengujian terhadap hasil data warehouse dilakukan secara kuantitatif dengan membandingkan angka-angka yang dihasilkan dengan angka-angka yang dibuat selama ini dalam bentuk manual. Berikut contoh pengujian terhadap beberapa sample data yang dihasilkan: Transaksi indovision untuk spin, credit card dan doku wallet

BUNCPN				FACTINDO				Indomaret				Affamart				SPINPAY				Credit card				Doku/Walet			
Transaksi	Tagihan	Transaksi	Tagihan	Transaksi	Tagihan	Transaksi	Tagihan	Transaksi	Tagihan	Transaksi	Tagihan	Transaksi	Tagihan	Transaksi	Tagihan	Transaksi	Tagihan	Transaksi	Tagihan	Transaksi	Tagihan	Transaksi	Tagihan	Transaksi	Tagihan		
0	0	0	0	968	202,235,591	985	117,222,166	1	35,000	5	4,932,949	3	115,134	0	0	0	0	0	0	0	0	0	0	0	0		
0	0	0	0	1,199	241,988,781	717	116,556,871	2	314,800	5	4,932,949	2	52,000	0	0	0	0	0	0	0	0	0	0	0	0		
0	0	0	0	1,257	245,699,004	688	129,789,251	2	365,000	2	948,545	1	54,000	0	0	0	0	0	0	0	0	0	0	0	0		
0	0	0	0	1,046	204,652,347	729	120,731,188	4	700,500	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0	0	0	0	922	164,788,834	446	78,816,264	3	345,000	3	1,596,124	2	109,000	0	0	0	0	0	0	0	0	0	0	0	0		
0	0	0	0	1,248	254,052,096	625	129,836,045	5	765,000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0	0	0	0	1,229	254,685,750	662	129,104,640	1	714,800	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0	0	0	0	1,034	212,495,302	694	138,224,059	4	255,000	2	128,000	0	143,000	0	0	0	0	0	0	0	0	0	0	0	0		
0	0	0	0	940	197,921,299	683	115,543,756	6	279,000	3	1,588,149	5	305,000	0	0	0	0	0	0	0	0	0	0	0	0		
0	0	0	0	982	193,370,483	598	119,298,426	4	598,646	5	3,305,850	1	74,000	0	0	0	0	0	0	0	0	0	0	0	0		
0	0	0	0	831	185,588,739	447	87,606,584	2	70,000	3	1,179,156	5	305,000	0	0	0	0	0	0	0	0	0	0	0	0		
0	0	0	0	649	120,063,244	446	76,296,135	3	130,000	5	1,072,700	2	209,000	0	0	0	0	0	0	0	0	0	0	0	0		
0	0	0	0	901	188,076,075	522	98,546,671	0	0	10	2,462,900	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0	0	0	0	867	208,571,160	551	121,884,931	0	0	7	4,386,750	3	123,000	0	0	0	0	0	0	0	0	0	0	0	0		
0	0	0	0	831	179,824,084	487	106,035,578	2	305,000	4	3,369,322	1	54,000	0	0	0	0	0	0	0	0	0	0	0	0		
0	0	0	0	831	186,420,954	471	93,514,046	3	1,703,950	1	304,900	0	25,000	0	0	0	0	0	0	0	0	0	0	0	0		
0	0	0	0	783	176,668,628	437	99,655,036	0	0	3	277,349	1	74,000	0	0	0	0	0	0	0	0	0	0	0	0		
0	0	0	0	711	149,735,511	423	80,669,145	1	60,000	0	0	1	113,000	0	0	0	0	0	0	0	0	0	0	0	0		
0	0	0	0	625	122,643,640	293	53,572,526	1	281,000	4	352,417	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0	0	0	0	946	191,347,824	566	109,866,439	1	2,000	2	3,144,600	1	54,000	0	0	0	0	0	0	0	0	0	0	0	0		
0	0	0	0	707	160,914,684	464	100,797,806	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0	0	0	0	636	143,631,341	393	89,819,763	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0	0	0	0	636	139,896,543	388	86,250,879	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0	0	0	0	687	141,699,061	432	95,621,107	0	0	5	3,761,417	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0	0	0	0	656	139,468,572	396	73,179,161	0	0	2	141,751	1	277,000	0	0	0	0	0	0	0	0	0	0	0	0		
0	0	0	0	541	100,579,126	319	70,621,692	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0	0	0	0	821	186,970,330	495	95,498,610	6	977,325	3	419,000	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0	0	0	0	757	146,848,517	474	99,695,962	1	163,000	4	1,799,700	1	54,000	0	0	0	0	0	0	0	0	0	0	0	0		
0	0	0	0	667	146,348,667	466	83,742,035	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0	0	0	0	828	185,535,049	488	95,079,878	1	60,000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0	0	0	0	23,744	5,492,735,235	15,067	3,016,073,300	53	8,144,371	93	4,177,454	31	0	0	0	0	0	0	0	0	0	0	0	0	0		

Gambar 9. Data jumlah transaksi yang diambil manual

Gambar 10. Data transaksi yang diambil melalui dashboard metabase dari data warehouse

Dimana salah satu contoh jumlah transaksi di laporan manual dan laporan yang dihasilkan metabase dari data warehouse ini sama yaitu :

Tabel 5 Contoh jumlah transaksi

Payment type	Laporan manual	Laporan data warehouse
Spin (spinpay)	53	53
Credit card	93	93
Doku walet	31	31

Pengujian berikutnya adalah pengujian kualitatif yaitu oleh user yang bekerja pada bagian pelaporan data, karena hasil yang di dapat dengan manual dan *data warehouse* sama, maka untuk selanjutnya pelaporan ke BI dilakukan dengan mengakses dashboard metabase yang mengakses *data warehouse*.

Berikut report dashboard yang dibuat pakai tools metabase untuk menampilkan table fakta.

Gambar 9 Summary per bulan

#### *F. Analisa Hasil*

Dari pengujian terhadap data warehouse yang dihasilkan dapat dianalisa bahwa nilai jumlah transaksi yang didapat dari proses manual dengan nilai yang didapat dari data warehouse menghasilkan nilai yang sama, sehingga laporan yang sebelumnya dibuat manual dapat digantikan dengan mengakses dashboard dari data warehouse yang otomatis menghasilkan laporan, dengan demikian pembuatan model data warehouse berjenjang dapat menghasilkan proses efisiensi dalam pelaporan perusahaan.

### 4. KESIMPULAN DAN SARAN

Kesimpulan dan saran yang dihasilkan dari penelitian ini adalah sebagai berikut:

#### *4.1. Kesimpulan*

1. Dari data warehouse yang dihasilkan dapat dianalisa bahwa untuk menghasilkan sebuah data warehouse dari berbagai database yang digabungkan dapat dilakukan dengan membuat proses ETL menggunakan script PHP buatan sendiri.
2. Dengan adanya data warehouse proses pembuatan laporan menjadi lebih efisien karena dapat dilakukan otomatis, tidak harus mengumpulkan satu persatu dari setiap database sehingga waktu menjadi lebih cepat di dalam proses pembuatan laporannya.
3. Pembuatan data warehouse secara berjenjang 2 tahap dapat membantu proses lain yang harus dilakukan oleh perusahaan payment gateway tersebut dilakukan dalam waktu yang sama, karena tidak hanya membuat data warehouse tapi juga sekaligus dapat dimanfaatkan untuk proses backup database dan proses rekonsiliasi dengan pihak *partner* dari perusahaan.

#### *4.2. Saran*

Usulan untuk penelitian selanjutnya adalah pembuatan data warehouse yang tidak hanya ditujukan untuk pembuatan laporan ke Bank Indonesia tapi untuk keperluan bisnis analisis internal perusahaan, dengan menambahkan table fakta dan table dimensional dari data transaksi yang ada, disamping itu perlu diadakannya penelitian tentang masalah tuning dan maintenance dari data warehouse ke depannya, karena data akan terus membengkak, kalau tidak dilakukan tuning dan maintenance akan menyebabkan performance data warehouse menurun dan berpotensi untuk hilang, disebabkan hardisk atau media penyimpanan yang rusak, karena tidak ada backup

### DAFTAR PUSTAKA

- [1] R. Kimball, *The Data Warehouse ETL Toolkit*. 2004.
- [2] A. Supriyatna and M. Wahyudi, "Perancangan Data Warehouse Pada Perpustakaan," *Semin. Nas. Apl. Teknol. Inf. 2012 (SNATI 2012)*, vol. 2012, no. Snati, pp. 15–16, 2007.
- [3] R. Wijaya and B. Pudjoatmodjo, "Penerapan Extraction-Transformation-Loading (ETL) Dalam Data Warehouse (Studi Kasus : Departemen Pertanian)," *J. Nas. Pendidik. Tek. Inform.*, vol. 5, no. 2, p. 61, 2016, doi: 10.23887/janapati.v5i2.9855.
- [4] I. G. N. A. Trisna Putra, I. N. A. Mahendra, and I. M. S. Putra, "IMPLEMENTASI ETL DATA WAREHOUSE DENGAN KONSEP FITUR METADATA DAN CLEANSING DATA PADA TOKO KUE," *SISTEMASI*, vol. 9, no. 2, p. 274, May 2020, doi: 10.32520/stmsi.v9i2.732.
- [5] I. M. S. Putra and D. K. T. Adhitya Putra, "Rancang Bangun Engine ETL Data Warehouse dengan Menggunakan Bahasa Python," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 3, no. 2, pp. 113–123, Aug. 2019, doi: 10.29207/resti.v3i2.872.
- [6] D. Pt and J. Teknik, "PERANCANGAN DATA WAREHOUSE dan," vol. 10, no. 1, pp. 17–24, 2022, doi: 10.21063/jtif.2022.V10.1.17-24.
- [7] D. K. Aditiya Hermawan, Yusuf Kurnia, Niki Destiandi, "Modeling data mart using ETL (extract, transform, load) webservice concept on feeder with a dashboard," *Int. J. Eng. Technol.*, 2018.

- [8] T. R. Priono, W. Purnomo, and N. Y. Setiawan, “Pengembangan Data Warehouse menggunakan Metode Kimball ( Studi Kasus : Ekspor & Impor Fauna dan Flora Hias Air Laut ),” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 8, p. 1, 2021.
- [9] T. Informatika and S. A. Bangsa, “Pemodelan Datawarehouse Koperasi Dengan Metode Four-Step Kimball,” vol. I, no. 2, pp. 220–227, 2015.
- [10] Kimball, R., Ross, M., & Anisimov, A. A. (2003). *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling (2nd Edition)*. In *SIGMOD Record* (Vol. 32, Issue 3). <https://doi.org/10.1145/945721.945741>
- [11] Marbun, I. R., & Somya, R. (2021). Perancangan Data Warehouse untuk Data Transaksi Penjualan Menggunakan Schema Snowflake Studi Kasus : Online Market Dataset. *Universitas Kristen Satya Wacana*, 5(1), 87–91.
- [12] Doro, E., & Stevalin, B. (2012). Analisis Data dengan Menggunakan ERD dan Model Konseptual Data Warehouse. *Jurnal Informatika*, 5(1), 71–85.
- [13] Widianty, W. (2015). Data Warehouse Design with Kimball Method: Case Study of Fahrenheit Manufacturing Systems. *ComTech: Computer, Mathematics and Engineering Applications*, 6(4), 604. <https://doi.org/10.21512/comtech.v6i4.2200>
- [14] Iswardani, P. R., Pramana, I. W. S., & Saputra, K. O. (n.d.). *Design of Data Warehouse for Monitoring Hotel ' s Food and Beverage Cost*. 1–4.
- [15] Foster, E. C., & Godbole, S. (2016). Database systems: A pragmatic approach. In *Database Systems: A Pragmatic Approach*. <https://doi.org/10.1007/978-1-4842-1191-5>