

Project-Based Virtual Internship Final Task

Data Warehouse & Stored Procedure

ID/X Partners - Data Engineer

Presented by

Deni Yuniawan



Deni Yuniawan

About Me

I am a data analyst and data enthusiast who is currently exploring the data field so that I can have a career in the data field, especially data analysts.

Experience & Certificate

-  **Home Credit Indonesia X Rakamin**
Virtual Internship
Desember 2024
-  **Data Science Bootcamp Batch 45**
Rakamin Academy
Mei 2024-Oktober 2024

About Company



id/x partners was established in 2002 by ex-bankers and management consultants who have vast experiences in credit cycle and process management, scoring development, and performance management. Our combined experience has served corporations across Asia and Australia regions and in multiple industries, specifically financial services, telecommunications, manufacturing and retail.

id/x partners provides consulting services that specializes in utilizing data analytic and decisioning (DAD) solutions combined with an integrated risk management and marketing discipline to help clients optimize the portfolio profitability and business process.

Comprehensive consulting service and technology solutions offered by id/x partners makes it as a one-stop service provider.

Project Portfolio

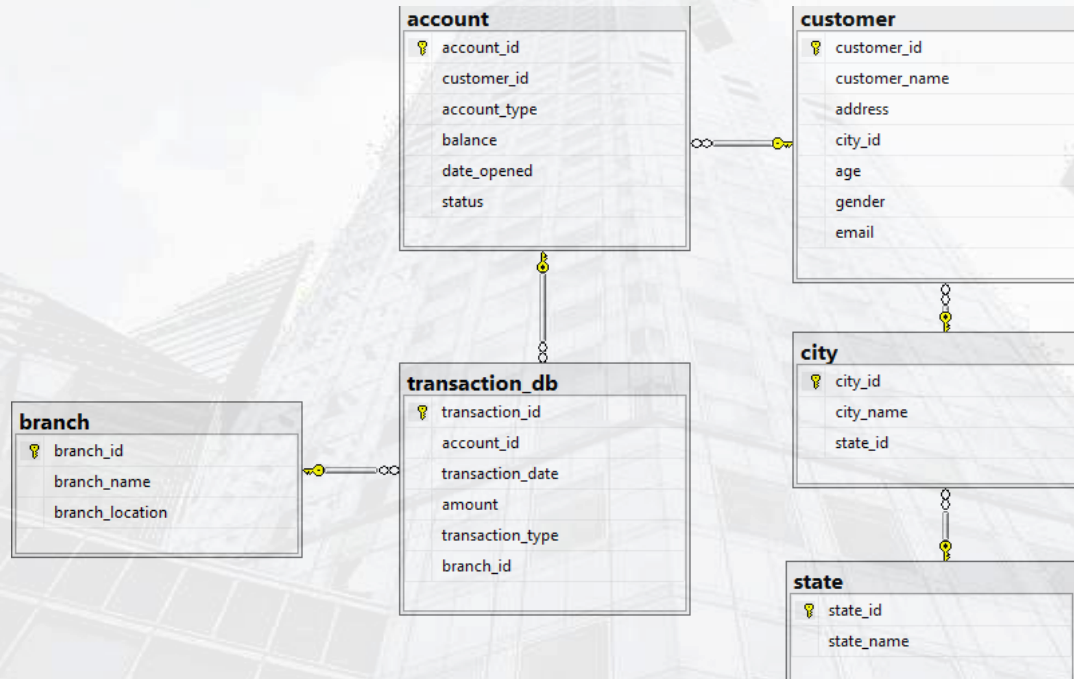
Latar Belakang

- Klien dari ID/X Partners yang bergerak di industri perbankan memerlukan solusi untuk mengintegrasikan data dari berbagai sumber seperti file excel, csv, dan database SQL Server.
- Tantangan utama adalah mengatasi keterlambatan pelaporan dan analisis data akibat kesulitan dalam penggabungan data dari sumber yang berbeda.

Project Portfolio

Database Diagram

Relasi antar tabel yang disediakan dapat dilihat pada gambar berikut

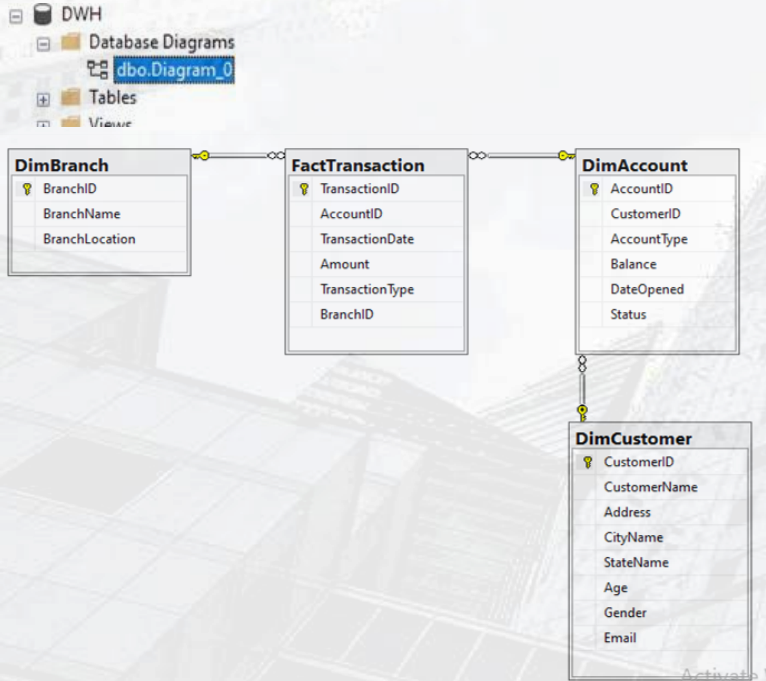


Project Portfolio

Challenge

1. Membuat sebuah database baru yang akan kita anggap sebagai Data Warehouse baru bernama DWH. Setelah itu buat tiga tabel dimension yaitu DimAccount, DimCustomer, DimBranch dan satu tabel fact yaitu FactTransaction.
2. Membuat job ETL di aplikasi talend untuk memindahkan data dari source ke seluruh tabel Dimension. Khusus untuk tabel DimCustomer, format kolom yang disimpan adalah CustomerID, CustomerName, Address, CityName, StateName, Age, Gender, Email. Semua data dari kolom tersebut diubah menjadi huruf kapital kecuali untuk kolom CustomerID, Age dan Email
3. Membuat job ETL untuk menggabungkan data transaksi (transaksi_excel, transaksi_csv, transaksi_db) menjadi satu di tabel FactTransaction.
4. Membuat dua Stored Procedure (SP) dengan parameter, untuk membantu mereka mendapatkan ringkasan data dengan cepat. Stored Procedure yang diminta yaitu a.) DailyTransaction, untuk menghitung banyaknya transaksi beserta total nominalnya setiap harinya. b.) BalancePerCustomer, untuk mengetahui sisa balance per customer.

1. Data Warehouse Creation

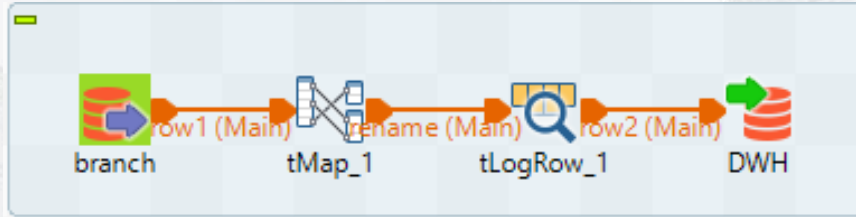


- Membuat Database '**DWH**' Dengan menggunakan Microsoft SQL Server Management Studio bisa melalui query sql atau melalui opsi database pada panel sebelah kiri klik kanan pilih New Database.
- Database 'DWH' digunakan sebagai data warehouse. Struktur tabel yang dibangun dalam data warehouse ini meliputi:
 1. **FactTransaction:** Berisi kolom-kolom seperti TransactionID, AccountID, TransactionDate, Amount, TransactionType, dan BranchID.
 2. **DimBranch:** Memuat informasi seperti BranchID, BranchName, dan BranchLocation.
 3. **DimAccount:** Mencakup kolom AccountID, CustomerID, AccountType, Balance, DateOpened, dan Status.
 4. **DimCustomer:** Menyimpan data CustomerID, CustomerName, Address, CityName, StateName, Age, Gender, dan Email.

2. Create ETL Job for DimBranch

Proses ETL untuk tabel DimBranch mencakup ekstraksi data dari database sampel, pengubahan nama kolom, dan pemuatan data ke dalam data warehouse.

Seluruh proses ini dilakukan menggunakan komponen tMap.



Talend Job

	branch_id	branch_name	branch_location
1	1	KC Jakarta	Jl. Gatot Subroto No 13
2	2	KC Bogor	Jl. Padjajaran No 43
3	3	KC Depok	Jl. Raya Sawangan No 34
4	4	KC Tangerang	Jl. Cisauk No 50

Input

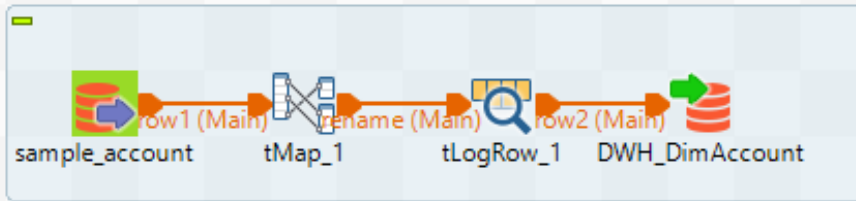
	BranchID	BranchName	BranchLocation
1	1	KC Jakarta	Jl. Gatot Subroto No 13
2	2	KC Bogor	Jl. Padjajaran No 43
3	3	KC Depok	Jl. Raya Sawangan No 34
4	4	KC Tangerang	Jl. Cisauk No 50

Output

2. Create ETL Job for DimAccount

Proses ETL untuk tabel DimAccount mengikuti langkah yang serupa, yaitu mengekstraksi data dari database, mengubah nama kolom, dan memuatnya ke dalam data warehouse.

Seluruh proses ini dijalankan menggunakan komponen tMap.



Talend Job

	account_id	customer_id	account_type	balance	date_opened	status
1	1	1	saving	1500000	2020-05-01 09:00:00	active
2	2	2	saving	500000	2020-06-01 10:00:00	active
3	3	1	checking	25000000	2020-06-21 09:00:00	active
4	4	3	checking	4500000	2021-06-24 11:00:00	terminated

Input

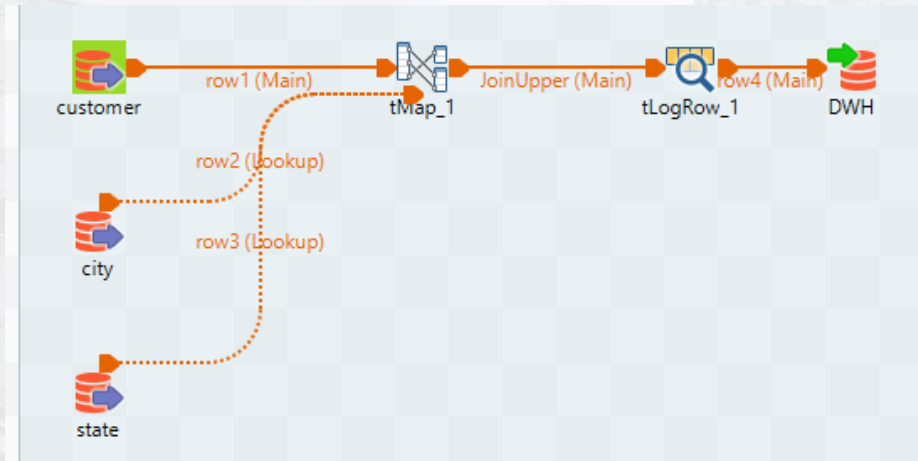
	AccountID	CustomerID	AccountType	Balance	DateOpened	Status
1	1	1	saving	1500000	2020-05-01 09:00:00	active
2	2	2	saving	500000	2020-06-01 10:00:00	active
3	3	1	checking	25000000	2020-06-21 09:00:00	active
4	4	3	checking	4500000	2021-06-24 11:00:00	terminated

Output

2. Create ETL Job for DimCustomer

Pada data warehouse yang telah dirancang, tabel DimCustomer mencakup kolom CityName dan StateName. Oleh karena itu, dalam proses ETL, data dari tabel Customer, City, dan State di database sumber perlu digabungkan terlebih dahulu.

Selain itu, ada ketentuan untuk mengubah semua data menjadi huruf kapital, kecuali kolom CustomerID, Age, dan Email. Kedua proses ini dilakukan secara bersamaan menggunakan komponen tMap.



Talend Job

	customer_id	customer_name	address	city_id	age	gender	email
1	1	Shelly Juwita	Jl. Boulevard No. 31	2	25	female	shelly@gmail.com
2	2	Bobi Rinaldo	Jl. Mangga No. 1	3	31	male	Bobi@gmail.com
3	3	Adam Malik	Jl. Kincir Angin No. 50	5	23	male	Adam@gmail.com
4	4	Susi Rahmawati	Jl. Kenanga No. 11	7	45	female	Susi@gmail.com

Input

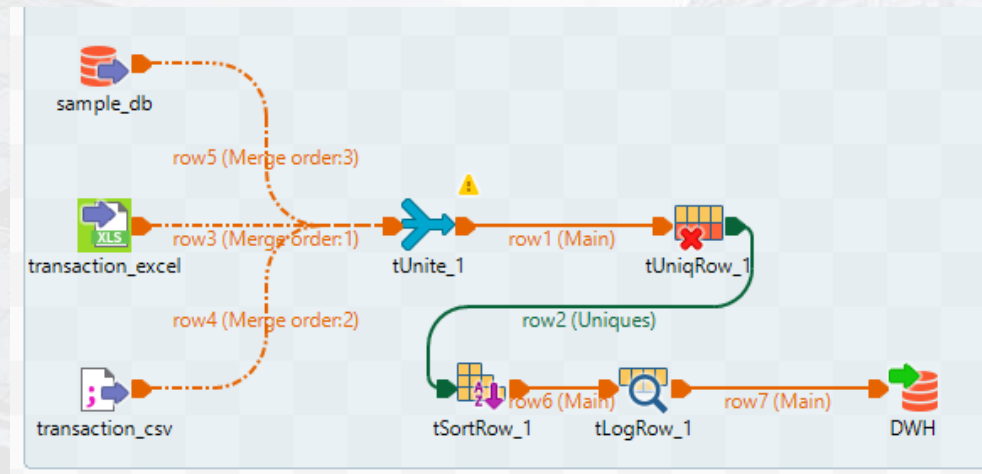
	CustomerID	CustomerName	Address	CityName	StateName	Age	Gender	Email
1	1	SHELLY JUWITA	JL. BOULEVARD NO. 31	KELAPA GADING	JAKARTA UTARA	25	female	shelly@gmail.com
2	2	BOBI RINALDO	JL. MANGGA NO. 1	TANJUNG PRIOK	JAKARTA UTARA	31	male	Bobi@gmail.com
3	3	ADAM MALIK	JL. KINCIR ANGIN NO. 50	PADEMANGAN	JAKARTA UTARA	23	male	Adam@gmail.com
4	4	SUSI RAHMAWATI	JL. KENANGA NO. 11	CILANDAK	JAKARTA SELATAN	45	female	Susi@gmail.com

Output

3. Create ETL Job for FactTransaction

Dalam job ETL ini, diperlukan metode untuk menggabungkan data dari berbagai sumber, seperti file CSV, Excel, dan database lain.

Untuk mencapai tujuan ini, digunakan komponen tUnite dan tUniqRow agar data dapat digabungkan tanpa duplikasi. Selain itu, komponen tSortRow digunakan untuk mengurutkan data berdasarkan ID transaksi.



Talend Job

transaction_csv	14,13,21-01-2024 14:00:00,1500000,Deposit,4
transaction_excel	6 6 18/01/2024 13:10 50000 Withdrawal 1

	transaction_id	account_id	transaction_date	amount	transaction_type	branch_id
1	1	1	2024-01-17 09:10:00	100000	Deposit	1
2	2	2	2024-01-17 10:10:00	1000000	Deposit	1
3	3	3	2024-01-18 08:30:00	10000000	Transfer	1
4	4	3	2024-01-18 10:45:00	1000000	Withdrawal	1

Input

	TransactionID	AccountID	TransactionDate	Amount	TransactionType	BranchID
1	1	1	2024-01-17 09:10:00	100000	Deposit	1
2	2	2	2024-01-17 10:10:00	1000000	Deposit	1
3	3	3	2024-01-18 08:30:00	10000000	Transfer	1
4	4	3	2024-01-18 10:45:00	1000000	Withdrawal	1

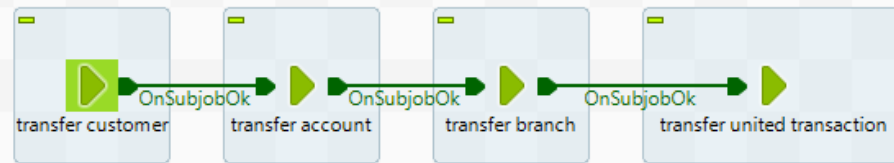
Output

4. ETL Orchestration

tRunJob

Untuk mempermudah proses ETL, satu job ETL dibuat untuk mengoordinasikan keseluruhan proses. Komponen tRunJob digunakan untuk menjalankan sub-job, sementara trigger OnSubjobOK memastikan sub-job berikutnya dimulai setelah yang sebelumnya selesai.

Saat job ini dijalankan, seluruh proses ETL—mulai dari pemindahan data dari berbagai sumber hingga transformasi data—akan berjalan secara otomatis.



Talend Job

5. Create Stored Procedure BalancePerCustomer

Stored procedure berikut dirancang untuk memperoleh saldo terkini per pelanggan. Prosedur ini menerima parameter berupa nama pelanggan yang ingin dicari saldonya.

```
CREATE PROCEDURE BalancePerCustomer
    @name VARCHAR(100)
AS
BEGIN
    SELECT C.CustomerName, A.AccountType, A.Balance,
        (A.Balance + SUM(
            CASE
                WHEN TransactionType != 'Deposit' THEN -T.Amount
            ELSE T.Amount
            END)) AS CurrentBalance
    FROM FactTransaction T
    JOIN DimAccount A ON T.AccountID = A.AccountID
    JOIN DimCustomer C ON A.CustomerID = C.CustomerID
    WHERE A.Status = 'active'
    GROUP BY C.CustomerName, A.AccountType, A.Balance
    HAVING CustomerName LIKE '%' + @name + '%'
END;
```

	TransactionID	CustomerName	TransactionDate	AccountType	Balance	Amount	TransactionType	Status
1	1	SHELLY JUWITA	2024-01-17 09:10:00	saving	1500000	100000	Deposit	active
2	3	SHELLY JUWITA	2024-01-18 08:30:00	checking	25000000	10000000	Transfer	active
3	4	SHELLY JUWITA	2024-01-18 10:45:00	checking	25000000	1000000	Withdrawal	active

EXEC BalancePerCustomer @name='Shelly'

	CustomerName	AccountType	Balance	CurrentBalance
1	SHELLY JUWITA	checking	25000000	14000000
2	SHELLY JUWITA	saving	1500000	1600000

	TransactionID	CustomerName	TransactionDate	AccountType	Balance	Amount	TransactionType	Status
1	17	ANI NURAINI	2024-01-22 13:10:00	checking	25000000	100000	Withdrawal	active
2	18	ANI NURAINI	2024-01-22 10:20:00	saving	10000000	700000	Deposit	active

EXEC BalancePerCustomer @name='ani'

	CustomerName	AccountType	Balance	CurrentBalance
1	ANI NURAINI	checking	25000000	24900000
2	ANI NURAINI	saving	10000000	10700000

5. Create Stored Procedure Daily Transaction

Stored procedure ini dibuat untuk memperoleh jumlah transaksi beserta total nominalnya dalam rentang waktu tertentu. Prosedur ini menerima parameter berupa tanggal awal dan akhir yang ingin dianalisis.

```
CREATE PROCEDURE DailyTransaction
    @start_date date,
    @end_date date
AS
BEGIN
    N
    SELECT CONVERT(date, TransactionDate) AS Date,
    COUNT(*) AS TotalTransactions,
    SUM(Amount) AS TotalAmount
    FROM FactTransaction
    WHERE TransactionDate >= @start_date AND TransactionDate <
    DATEADD(day, 1, @end_date)
    GROUP BY CONVERT(date, TransactionDate);
END;
```

Query Stored Procedure

```
EXEC DailyTransaction @start_date='2024-01-18' , @end_date='2024-01-20';
```

	Date	TotalTransactions	TotalAmount
1	2024-01-18	4	11250000
2	2024-01-19	3	5400000
3	2024-01-20	4	4000000

Contoh Penggunaan

Thank You

Github: https://github.com/deniywn/rakamin-idx-DE_pbi



Rakamin
Academy



id/x

partners