Rakamin Academy X id/x partners

Data Warehouse dan Stored Procedure

ID/X Partners - Data Engineer

Presented by
Irpan Maulana

Irpan Maulana

Hi, I'm a graduate of informatics engineering and I'm a data enthusiast. I have a strong interest in continuing to hone my skills in the data field. I've completed several data science training courses and am currently trying to get into data engineering.

Email:
irpanmaulana038.im@gmail.com

Linkedin :
linkedin.com/in/irpan-maulana-87a1332b5

Github:
github.com/irpanmaulana038

Link

Presentation videos on YouTube: https://youtu.be/0eo0SxYD6Dw

Github Repositorie: github.com/irpanmaulana038/Datawarehouse-StoredProcedure

Drive: drive.google.com/drive/folders/1h5_jQrjMAQPW7D_Drm_1INLkrE-kRQbK?usp=sharing

About ID/X Partners

ID/X Partners is a consulting firm specialising in information technology solutions. specialising in leveraging data analytics and decision making (DAD) solutions combined with risk management and integrated marketing disciplines to help clients optimise portfolio profitability and business processes

Case Study

Salah satu client dari perusahaan ID/X Partners yang bergerak di industri perbankan, memiliki kebutuhan untuk membuat sebuah Data Warehouse dari beberapa sumber data yang berbeda yang tersimpan di dalam sistem mereka.
Permasalahan yang mereka hadapi saat ini adalah mereka kesulitan untuk mengekstrak data dari berbagai sumber (excel, csv, database) secara bersamaan sehingga pelaporan dan analisis data mereka selalu mengalami keterlambatan

01 Membuat Database

Membuat database baru yang akan kita anggap sebagai Data Warehouse dan membuat 3 tabel dimension

02 Job ETL Dimension Table

Membuat job ETL di aplikasi talend untuk memindahkan data dari source ke seluruh tabel Dimension

03 Job ETL FactTransaction

Membuat job ETL untuk menggabungkan data transaksi menjadi satu di tabel FactTransaction

04 Stored Procedure

Membuat dua Stored Procedure (SP) dengan parameter, untuk membantu mereka mendapatkan ringkasan data dengan cepat

Outline

01 Data Warehouse Creation

02 Create ETL Job For Dimension Table

03 Create ETL Job For Fact Table

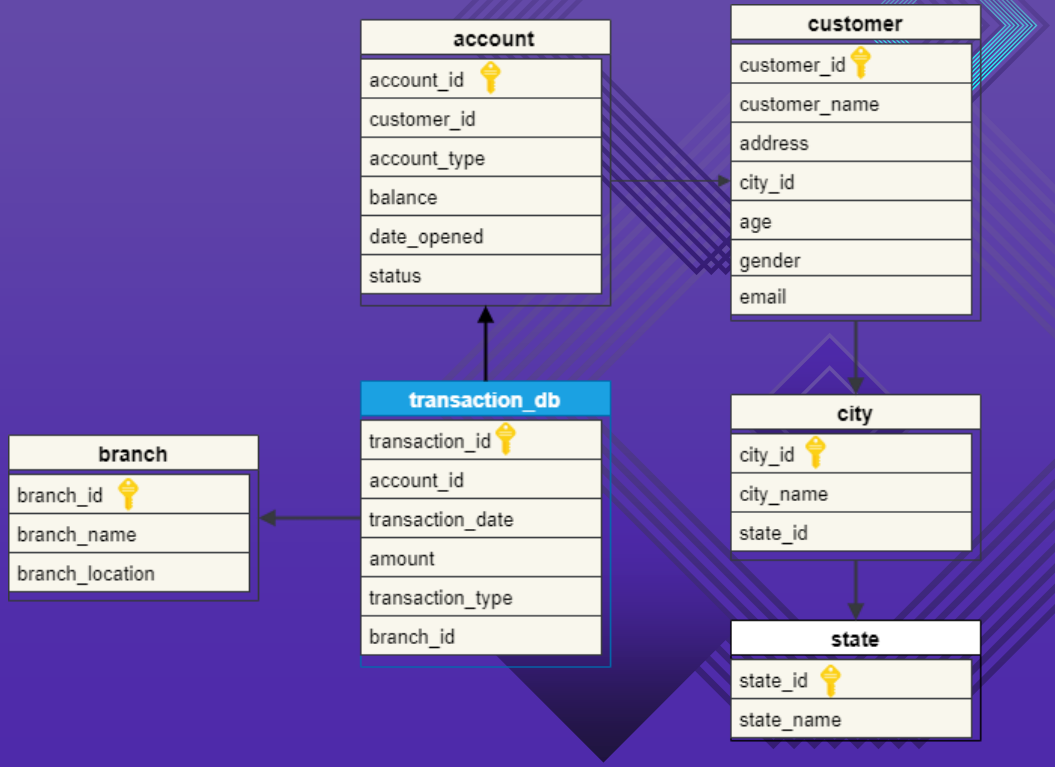
04 Create Stored Procedure

01 Data Warehouse Creation

Restore Database

Merestore Database yang akan digunakan sebagai source dan berisikan 6 tabel

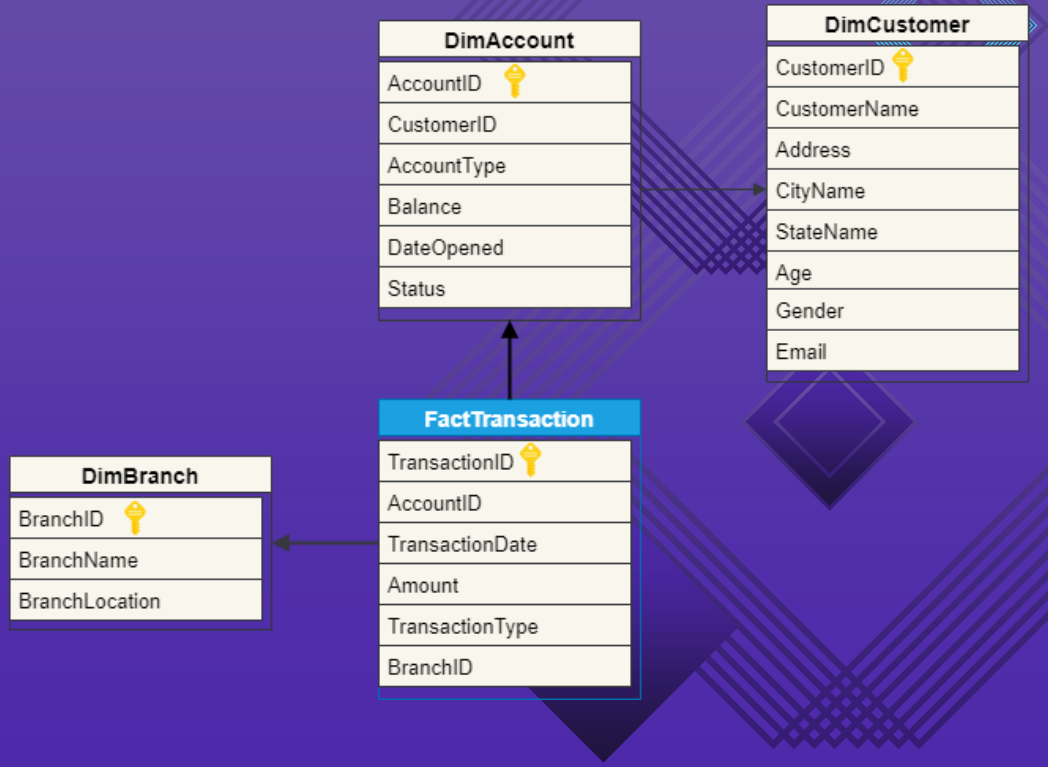
- transaction_db
- account
- branch
- customer
- city
- state



Create Database

Membuat Database bernama DWH sebagai Data Warehouse dan berisi 3 tabel dimension dan 1 tabel

- DimCustomer
- DimAccount
- DimBranch
- FactTransaction



Query Dimension and Fact Table

Dimension Customer Table

```
CREATE TABLE DimCustomer(
CustomerID INT  NOT NULL,
CustomerName VARCHAR(50),
Address VARCHAR(Max) NULL,
CityName VARCHAR(100),
StateName VARCHAR(100),
Age VARCHAR(3) NULL,
Gender VARCHAR(10) NULL,
Email VARCHAR(50) NULL,
PRIMARY KEY(CustomerID)
);
```

Dimension Account Table

```
CREATE TABLE DimAccount(
AccountID INT NOT NULL,
CustomerID INT NULL,
AccountType VARCHAR(10) NULL,
Balance INT NULL,
DataOpened DATETIME2(0) NULL,
Status VARCHAR(10) NULL,
PRIMARY KEY(AccountID),
FOREIGN KEY (CustomerID) REFERENCES
DimCustomer(CustomerID)
);
```

Dimension Branch Table

```
CREATE TABLE DimBranch(
BranchID INT NOT NULL,
BranchName VARCHAR(50) NULL,
BranchLocation VARCHAR(50) NULL,
PRIMARY KEY(BranchID)
);
```

Dimension Fact Transaction Table

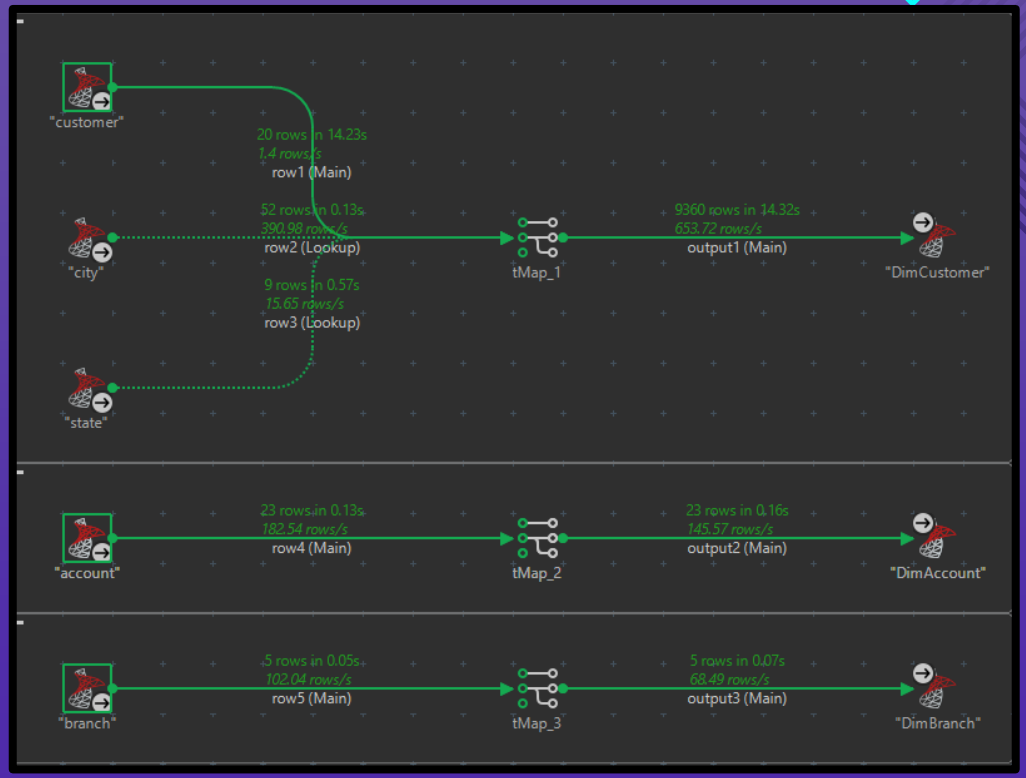
```
CREATE TABLE FactTransaction(
TransactionID INT NOT NULL,
CustomerID INT NULL,
AccountID INT NULL,
Amount INT NULL,
TransactionDate DATETIME2(0),
Transaction_type VARCHAR(50) NULL,
BranchID INT NULL,
PRIMARY KEY(TransactionID),
FOREIGN KEY (AccountID) REFERENCES
DimAccount(AccountID),
FOREIGN KEY (BranchID) REFERENCES
DimBranch(BranchID)
);
```

02

Create
ETL Job
For Dimension
Table

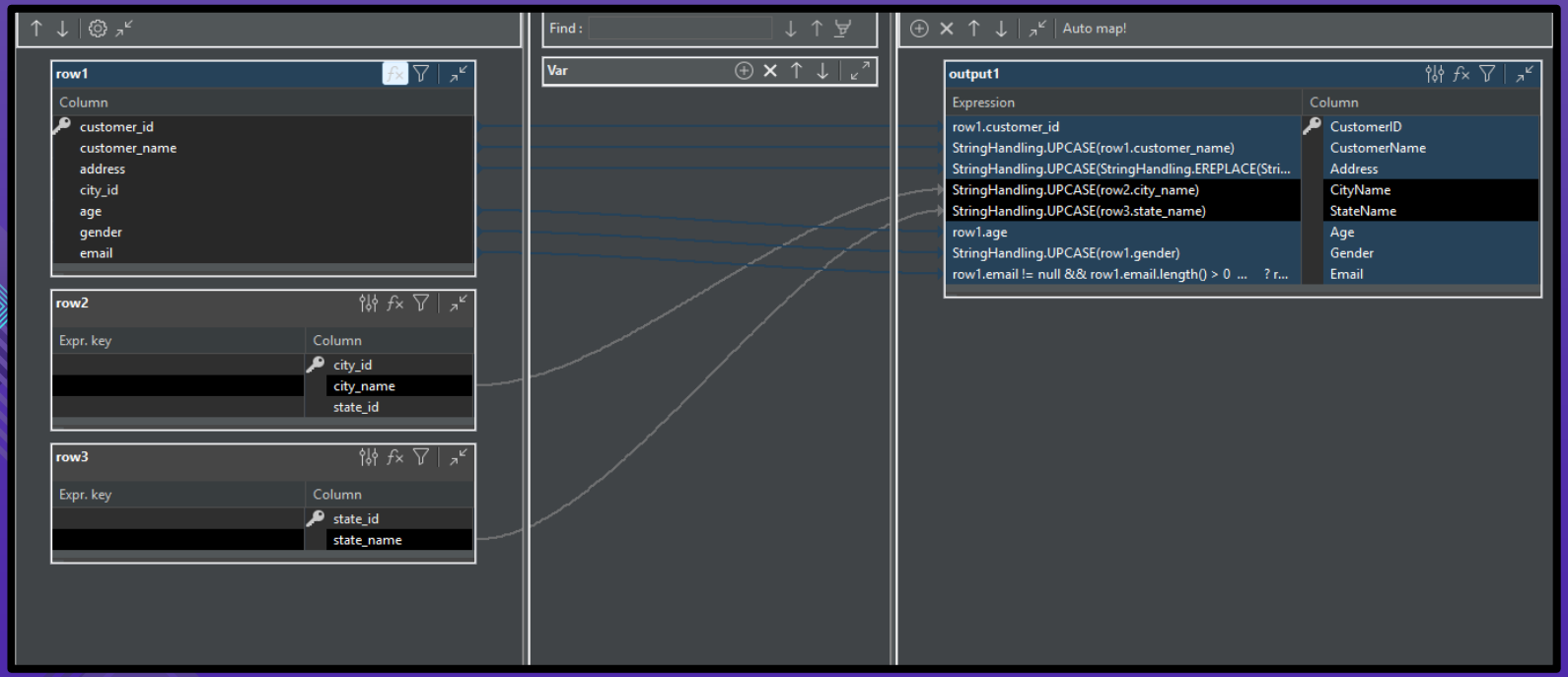
Dimension **Job**

Membuat Job ETL untuk memindahkan Data dari source yang ada di db hasil restore database masuk tMap untuk di proses lalu output dari tMap masuk ke Dimension Tabel



tMap DimCustomer

Khusus untuk DimCustomer kolom CustomerName, Address, CityName, StateName pada tMap data dari source di ubah menjadi kapital dan Kolom CustomerID, Age, Email di ubah mengikuti kaidah PascalCase dengan menambahkan Ekpression pada Tabel Output



Output **Dimension Job**

DimCustomer

	CustomerID	CustomerName	Address	CityName	StateName	Age	Gender	Email
1	1	SHELLY JUWITA	JL. BOULEVARD NO. 31	SUKATANI	BEKASI	25	FEMALE	Shelly@gmail.com
2	2	BOBI RINALDO	JL. MANGGA NO. 1	SUKATANI	BEKASI	31	MALE	Bobi@gmail.com
3	3	ADAM MALIK	JL. KINCIR ANGIN NO. 50	SUKATANI	BEKASI	23	MALE	Adam@gmail.com

DimAccount

	AccountID	CustomerID	AccountType	Balance	DataOpened	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

DimBranch

	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

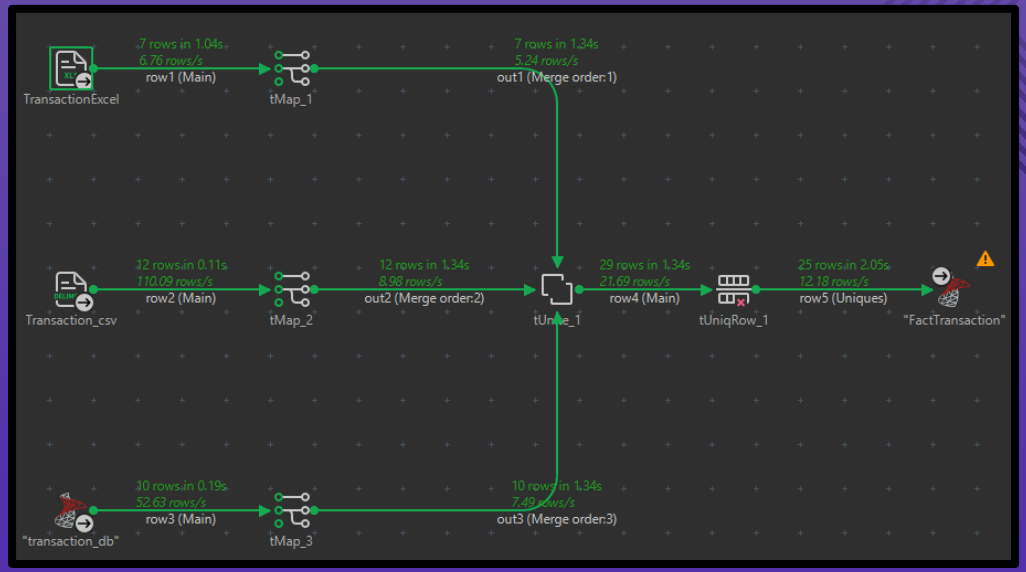
03

Create **ETL Job** for **Fact Table**

Fact Transaction Job

Membuat Job ETL untuk menggabungkan Data dari 3 source berbeda yaitu Excel, Csv dan DB.

Pada Fact Job ini komponen yang digunakan yaitu komponen tMap untuk memproses data dari data input, lalu komponen tUnite untuk menggabungkan data dari ke 3 source berbeda dan komponen tUniqRow untuk data yang duplikat



Output Fact **Job**

FactTransaction

	TransactionID	AccountID	Amount	TransactionDate	Transaction_type	BranchID
1	1	1	100000	2024-01-17 09:10:00.000	Deposit	1
2	2	2	1000000	2024-01-17 10:10:00.000	Deposit	1
3	3	3	10000000	2024-01-18 08:30:00.000	Transfer	1
4	4	3	1000000	2024-01-18 10:45:00.000	Withdrawal	1
5	5	5	200000	2024-01-18 11:10:00.000	Deposit	1

data dari ke 3 source yang berbeda memiliki beberapa data yang sama dan menggunakan komponen tUniq Row sehingga data yang masuk ke tabel FactTransaction tidak ada yang data duplikat

04

Create **Stored Procedure**

Daily Transaction

Menghitung banyaknya transaksi beserta total nominal setiap harinya dengan kolom yang di tampilkan

- Date
- TotalTransaction
- TotalAmount

Menjalankan Store Procedure ini dengan dua parameter yaitu start_date dan end_date untuk menampilkan data berdasarkan rentang tanggal yang di masukan

Query SP Daily Transaction

```
CREATE PROCEDURE DailyTransaction

    @start_date DATE,
    @end_date DATE
AS
BEGIN
    SELECT
        CONVERT(DATE, TransactionDate) AS [Date],
        COUNT(TransactionID) AS TotalTransactions,
        SUM(Amount) AS TotalAmount
    FROM
        FactTransaction
    WHERE
        TransactionDate BETWEEN @start_date
        AND  @end_date
    GROUP BY
        CONVERT(DATE, TransactionDate)
    ORDER BY
        [Date] ASC;
END
```

Output Daily Transaction

Menjalankan Stored Procedure Daily Transaction dengan memasukan 2 parameter start_date dan end_date

Dapat dilihat di bawah ini parameter start_date di isi dengan `'2024-01-17'` dan end_date di isi dengan `'2024-01-19'` serta outputnya pada gambar di samping

	Date	TotalTransactions	TotalAmount
1	2024-01-17	2	1100000
2	2024-01-18	4	11250000
3	2024-01-19	3	5400000
4	2024-01-20	4	4000000

```
EXEC DailyTransaction '2024-01-17', '2024-01-19';
```

Balance PerCustomer

Untuk mengetahui sisa balance per customer dengan kolom yang di tampilkan :

- CustomerName
- AccountType
- Balance
- CurrentBalance

Menjalankan Store Procedure ini dengan parameter customer_name untuk menampilkan data berdasarkan nama customer yang di masukan

Query SP Balance PerCustomer

```
CREATE PROCEDURE BalancePerCustomer
    @customer_name NVARCHAR(255)
AS
BEGIN
    SELECT
        DC.CustomerName,
        DA.AccountType,
        DA.Balance,
        (DA.Balance + ISNULL(SUM(CASE
                WHEN FT.Transaction_type = 'Deposit' THEN
FT.Amount
                ELSE -FT.Amount
            END), 0)) AS CurrentBalance
    FROM
        DimCustomer DC
    INNER JOIN
        DimAccount DA ON DC.CustomerID = DA.CustomerID
    LEFT JOIN
        FactTransaction FT ON DA.AccountID = FT.AccountID
    WHERE
        DC.CustomerName = @customer_name
        AND DA.Status = 'Active'
    GROUP BY
        DC.CustomerName, DA.AccountType, DA.Balance
    ORDER BY
        DC.CustomerName, DA.AccountType;
END
```

Output Balance PerCustomer

Menjalankan Stored Procedure Daily Transaction dengan memasukan parameter customer_name

Dapat dilihat di bawah ini parameter customer_name di isi dengan `'SHELLY JUWITA'` serta outputnya pada gambar di samping

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

```
EXEC DailyTransaction 'SHELLY JUWITA';
```

Rakamin Academy X id/x partners

Thanks!

Do you have any questions?

Irpanmaulana038.im@gmail.com
linkedin.com/in/irpan-maulana-87a1332b5