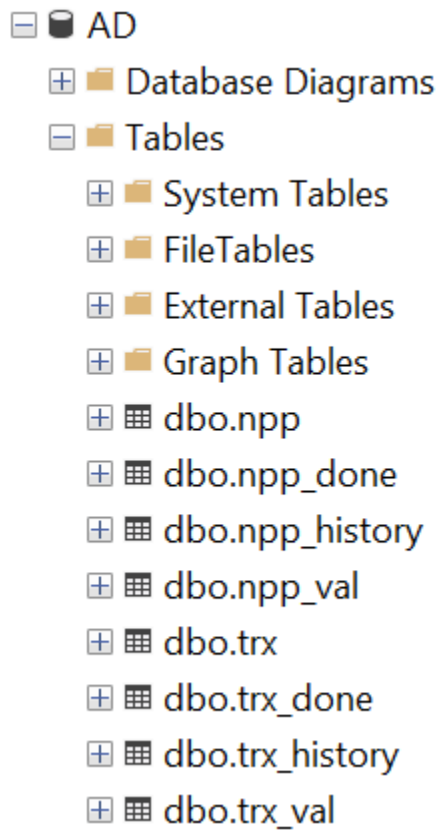


**Mochammad Aditya Putra Suhendar**

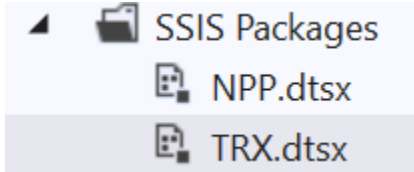
**Application as Data Engineer to Asosiasi Fintech Pendanaan Bersama Indonesia (AFPI)**

Soal dan data dapat dilihat pada [link ini](#)

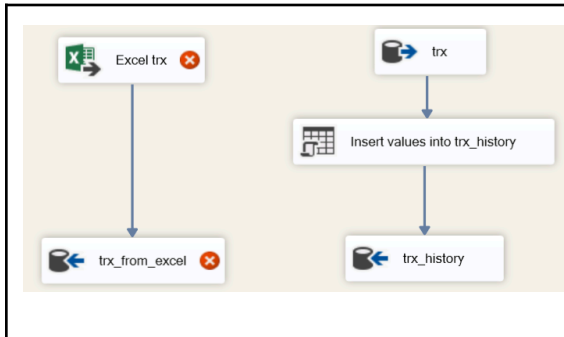
### **STEP 1: Pembuatan database dan tables TRX & NPP**

	<p>Nama database saya buat simple as 'AD', kemudian disini ada 2 jenis kepentingan tabel yaitu untuk TRX dan NPP, masing-masing memiliki 4 tabel yang nantinya akan digunakan secara berurutan sebagai berikut:</p> <p>Contoh TRX: trx &gt; trx_history &gt; trx_val &gt; trx_done</p> <ol style="list-style-type: none"><li>1. Keperluan dan perbedaanya sebagai berikut: trx : Menyimpan data yang sudah di extract dari excel/laporan</li><li>2. trx_history: Menyimpan log history kapan saja data tersebut masuk (tidak di truncate)</li><li>3. trx_val : Menyimpan data yang sudah dilakukan validasi (saat ini menggunakan stored procedure untuk melakukan validasi)</li><li>4. trx_done : menyimpan data yang sudah di validasi untuk digunakan dalam view</li></ol> <p>Begitu pula sama halnya dengan NPP.</p>
--	--

### **STEP 2: Pembuatan ETL (dengan contoh TRX, NPP mengikuti, akan dibandingkan jika terdapat perbedaan)**

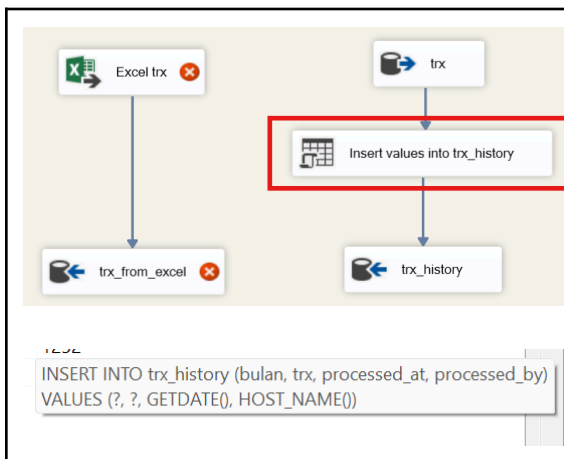
	<p>Terdapat 2 packages yang dibuat, masing-masing keperluannya berbeda, namun stepnya sama.</p>
---	---

## 1. Extract



Step yang seharusnya digunakan adalah yang kiri, namun karena excelnya perlu penyesuaian dan dapat mengacak format,

Saya menggunakan yang kanan dengan input jumlah manual kemudian diproses dan data yang berisi nama desktop dan waktu saat diproses masuk ke dalam tabel `trx_history` menjadikannya log.



Di dalam box 'Insert values into `trx_history`' adalah query tersebut untuk memasukan informasi waktu saat diproses beserta nama desktop atau pelaku.

99 %

select \* from trx  
select \* from trx\_history

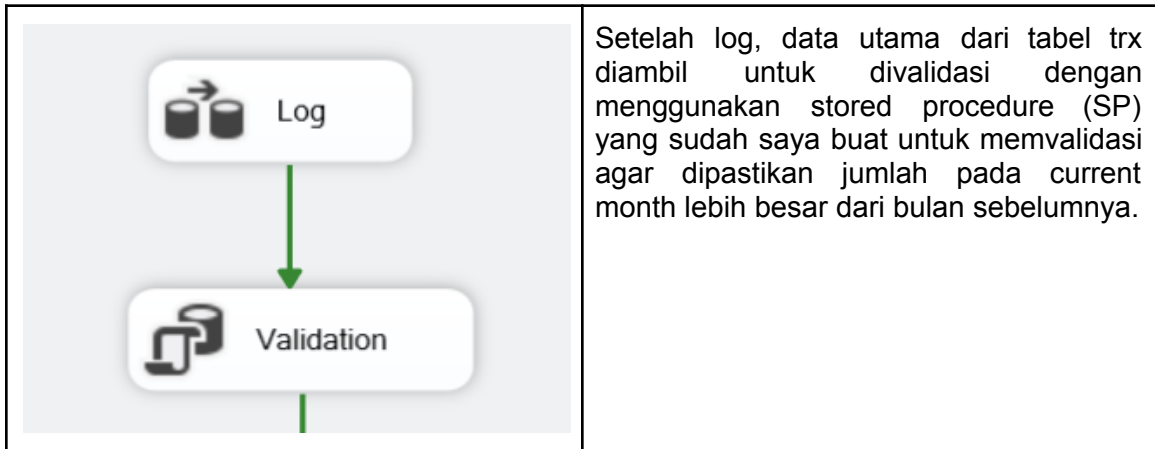
Results Messages

	id	bulan	trx
1	1	2023-12-01	872328649
2	2	2024-01-01	882544241
3	3	2024-02-01	891766171
4	4	2024-03-01	901558065
5	5	2024-04-01	913802708
6	6	2024-05-01	922495285
7	7	2024-06-01	933812537
8	8	2024-07-01	948166037

	id	bulan	trx	processed_at	processed_by
1	1	2023-12-01	872328649	2025-04-17 06:57:03.067	SUHENDAR-AKUNTP
2	2	2024-01-01	882544241	2025-04-17 06:57:03.090	SUHENDAR-AKUNTP
3	3	2024-02-01	891766171	2025-04-17 06:57:03.113	SUHENDAR-AKUNTP
4	4	2024-03-01	901558065	2025-04-17 06:57:03.137	SUHENDAR-AKUNTP
5	5	2024-04-01	913802708	2025-04-17 06:57:03.160	SUHENDAR-AKUNTP
6	6	2024-05-01	922495285	2025-04-17 06:57:03.183	SUHENDAR-AKUNTP
7	7	2024-06-01	933812537	2025-04-17 06:57:03.207	SUHENDAR-AKUNTP
8	8	2024-07-01	948166037	2025-04-17 06:57:03.230	SUHENDAR-AKUNTP
9	9	2024-08-01	958001161	2025-04-17 06:57:03.250	SUHENDAR-AKUNTP
10	1...	2024-09-01	971075900	2025-04-17 06:57:03.277	SUHENDAR-AKUNTP
11	1...	2024-10-01	985326426	2025-04-17 06:57:03.297	SUHENDAR-AKUNTP
12	1...	2024-11-01	998895879	2025-04-17 06:57:03.320	SUHENDAR-AKUNTP
13	1...	2024-12-01	1013430...	2025-04-17 06:57:03.340	SUHENDAR-AKUNTP

Berikut adalah isi data tabel `trx` dan hasil dari `trx_history`.

## 2. Transform



<div><div><div></div><div>Stored Procedures</div></div><div><div></div><div>System Stored Procedures</div></div><div><div></div><div>dbo.sp_npp_val</div></div><div><div></div><div>dbo.sp_trx_val</div></div></div>	<p>Berikut adalah masing-masing SP untuk memenuhi keperluan:</p> <p>TRX bulan H &gt; TRX bulan H-1 NPP bulan H &gt;= NPP bulan H-1</p>
--	--

<pre>ALTER PROCEDURE [dbo].[sp_trx_val] AS BEGIN     -- Kosongkan tabel dulu     TRUNCATE TABLE trx_val;      -- Insert data dari trx (biarkan IDENTITY bekerja)     INSERT INTO trx_val (bulan, trx, is_valid trx)     SELECT bulan, trx, 0 FROM trx;      -- Validasi kenaikan trx     ;WITH trx_cte AS (         SELECT             id,             bulan,             trx,             LAG(trx) OVER (ORDER BY bulan) AS prev_trx         FROM trx_val     )     UPDATE v     SET is_valid trx = CASE         WHEN t.prev_trx IS NULL THEN 1 -- baris pertama, set is_valid_trx = 1         WHEN v.trx &gt; t.prev_trx THEN 1         ELSE 0     END     FROM trx_val v     INNER JOIN trx_cte t ON v.id = t.id; END</pre>	<p>Berikut adalah isi SP untuk validasi trx.</p> <p>Di mulai truncate table validasi terlebih dahulu agar dipastikan kosong, lalu divalidasi dengan tambahan kolom baru 'is_valid_'</p>
--	---

```

ALTER PROCEDURE [dbo].[sp_npp_val]
AS
BEGIN
    -- Kosongkan tabel dulu
    TRUNCATE TABLE npp_val;

    -- Insert data dari npp (biarkan IDENTITY bekerja)
    INSERT INTO npp_val (bulan, npp, is_valid_npp)
    SELECT bulan, npp, 0 FROM npp;

    -- Validasi kenaikan npp
    ;WITH npp_cte AS (
        SELECT
            id,
            bulan,
            npp,
            LAG(npp) OVER (ORDER BY bulan) AS prev_npp
        FROM npp_val
    )
    UPDATE v
    SET is_valid_npp = CASE
        WHEN t.prev_npp IS NULL THEN 1 -- Untuk baris pertama, validasi di-set 1
        WHEN v.npp >= t.prev_npp THEN 1
        ELSE 0
    END
    FROM npp_val v
    INNER JOIN npp_cte t ON v.id = t.id;
END

```

Berikut adalah isi SP untuk validasi npp.

Terdapat perbedaan dengan trx, kalau npp persyaratannya lebih kecil atau sama dengan bulan lalu.

select \* from trx\_val

99 %

Results		Messages		
	id	bulan	trx	is_valid_trx
1	1	2023-12-01	872328649	1
2	2	2024-01-01	882544241	1
3	3	2024-02-01	891766171	1
4	4	2024-03-01	901558065	1
5	5	2024-04-01	913802708	1
6	6	2024-05-01	922495285	1
7	7	2024-06-01	933812537	1
8	8	2024-07-01	948166037	1
9	9	2024-08-01	958001161	1
10	10	2024-09-01	971075900	1
11	11	2024-10-01	985326426	1
12	12	2024-11-01	998895879	1
13	13	2024-12-01	1013430034	1

Berikut adalah hasil data trx yang sudah divalidasi dengan arti angka '1' sebagai True yang artinya jumlah current month lebih besar dari masing-masing bulan sebelumnya.

Direct input

Linked Azure Resources

```
INSERT INTO trx_done (bulan, trx, is_valid_trx, processed_at, processed_by)
SELECT
    bulan,
    trx,
    CASE
        WHEN is_valid_trx = 1 THEN 'valid'
        WHEN is_valid_trx = 0 THEN 'invalid'
    END AS is_valid_trx,
    GETDATE() AS processed_at, -- Waktu pemrosesan sekarang
    HOST_NAME() AS processed_by -- Nama komputer yang memproses
FROM trx_val;
```

Value dengan angka pada kolom 'is\_valid\_' diubah menjadi varchar jika 1 adalah 'valid' dan 0 adalah 'invalid' dengan memasukkan value tambahan seperti waktu saat proses dan nama desktop user/pelaku

3. Load

```
graph TD
    A[Truncate trx_done] --> B[Insert into trx_done]
```

Data yang sudah di validasi dan di transform kemudian di load ke dalam tabel akhir yaitu trx\_done.

Tidak lupa di truncate dahulu agar ada pengkinian data.

select \* from trx\_done

99 %

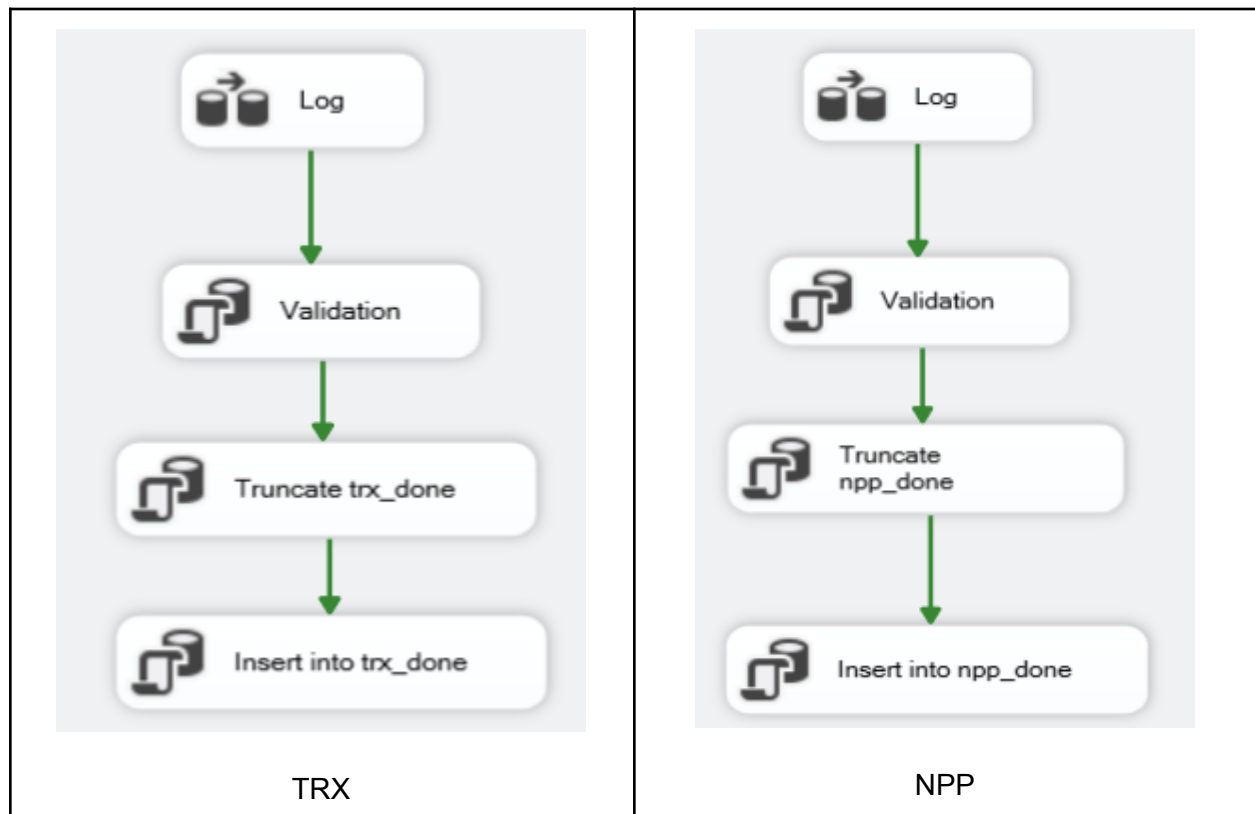
Results

Messages

	id	bulan	trx	is_valid_trx	processed_at	processed_by
1	1	2023-12-01	872328649	valid	2025-04-17 07:13:35.133	SUHENDAR-AKUNTP
2	2	2024-01-01	882544241	valid	2025-04-17 07:13:35.133	SUHENDAR-AKUNTP
3	3	2024-02-01	891766171	valid	2025-04-17 07:13:35.133	SUHENDAR-AKUNTP
4	4	2024-03-01	901558065	valid	2025-04-17 07:13:35.133	SUHENDAR-AKUNTP
5	5	2024-04-01	913802708	valid	2025-04-17 07:13:35.133	SUHENDAR-AKUNTP
6	6	2024-05-01	922495285	valid	2025-04-17 07:13:35.133	SUHENDAR-AKUNTP
7	7	2024-06-01	933812537	valid	2025-04-17 07:13:35.133	SUHENDAR-AKUNTP
8	8	2024-07-01	948166037	valid	2025-04-17 07:13:35.133	SUHENDAR-AKUNTP
9	9	2024-08-01	958001161	valid	2025-04-17 07:13:35.133	SUHENDAR-AKUNTP
10	10	2024-09-01	971075900	valid	2025-04-17 07:13:35.133	SUHENDAR-AKUNTP
11	11	2024-10-01	985326426	valid	2025-04-17 07:13:35.133	SUHENDAR-AKUNTP
12	12	2024-11-01	998895879	valid	2025-04-17 07:13:35.133	SUHENDAR-AKUNTP
13	13	2024-12-01	1013430034	valid	2025-04-17 07:13:35.133	SUHENDAR-AKUNTP

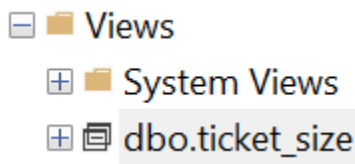
Berikut adalah hasil di tabel akhir trx\_done.

Sehingga ETL Flownya akan terbentuk seperti ini:



### Pembuatan View Ticket Size

Dengan rumus  $\text{Ticket Size} = \text{NPP} / \text{TRX}$  saya membuat view untuk melihat data hasil akhir yang diimport dari tabel `trx_done` dan `npp_done`, berikut hasilnya:

	Dapat dilihat pada query disamping, dengan menggunakan join kolom bulan di masing-masing table (TRX dan NPP), dapat ditemukan hasil pembagian as <code>ticket_size</code> .
---	---

```

CREATE VIEW [dbo].[ticket_size] AS
SELECT
    t.id,
    t.bulan,
    (n.npp / t.trx) AS ticket_size
FROM
    dbo.trx_done t
JOIN
    dbo.npp_done n ON t.bulan = n.bulan;

```

select \* from ticket\_size

99 %

Results Messages

	id	bulan	ticket_size
1	1	2023-12-01	0.0008748363485193755226
2	2	2024-01-01	0.0008903200581873152804
3	3	2024-02-01	0.0009043829270801078638
4	4	2024-03-01	0.0009197240889858824567
5	5	2024-04-01	0.0009308687888020572598
6	6	2024-05-01	0.0009480109700506490935
7	7	2024-06-01	0.0009628840847314518331
8	8	2024-07-01	0.0009763706185143604758
9	9	2024-08-01	0.0009924857491900262947
10	10	2024-09-01	0.0010075352297384787327
11	11	2024-10-01	0.0010091386303689778436
12	12	2024-11-01	0.0010218887788604041282
13	13	2024-12-01	0.0010347399867961679138

Berikut adalah hasil dari pemanggilan view ticket\_size yang dapat dilihat karena validasinya sudah benar bahwasannya ada peningkatan trx dan npp di setiap bulannya, nilai ticket size juga bertambah seiring penambahan/pergantian bulan.