**Mochammad Aditya Putra Suhendar**

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

Soal dan data dapat dilihat pada [link ini](https://docs.google.com/document/u/0/d/1HWyOStEG__afyAupTHft1RGDZ4fiY6lyZN5obVRWLQc/edit)

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

|  | 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:  Contoh TRX:  trx > trx\_history > trx\_val > trx\_done   1. Keperluan dan perbedaanya sebagai berikut: trx : Menyimpan data yang sudah di extract dari excel/laporan 2. trx\_history: Menyimpan log history kapan saja data tersebut masuk (tidak di truncate) 3. trx\_val : Menyimpan data yang sudah dilakukan validasi (saat ini menggunakan stored procedure untuk melakukan validasi) 4. trx\_done : menyimpan data yang sudah di validasi untuk digunakan dalam view   Begitu pula sama halnya dengan NPP. |
| --- | --- |

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

|  | Terdapat 2 packages yang dibuat, masing-masing keperluannya berbeda, namun stepnya sama. |
| --- | --- |

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. |
| --- | --- |

|  | Berikut adalah isi data tabel trx dan hasil dari trx\_history. |
| --- | --- |

1. **Transform**

|  | Setelah log, data utama dari tabel trx diambil untuk divalidasi dengan menggunakan stored procedure (SP) yang sudah saya buat untuk memvalidasi agar dipastikan jumlah pada current month lebih besar dari bulan sebelumnya. |
| --- | --- |

|  | Berikut adalah masing-masing SP untuk memenuhi keperluan:  TRX bulan H > TRX bulan H-1  NPP bulan H >= NPP bulan H-1 |
| --- | --- |

|  | Berikut adalah isi SP untuk validasi trx.  Di mulai truncate table validasi terlebih dahulu agar dipastikan kosong, lalu divalidasi dengan tambahan kolom baru ‘is\_valid\_’ |
| --- | --- |

|  | Berikut adalah isi SP untuk validasi npp.  Terdapat perbedaan dengan trx, kalau npp persyaratannya lebih kecil atau sama dengan bulan lalu. |
| --- | --- |

|  | 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. |
| --- | --- |

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

1. **Load**

|  | 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. |
| --- | --- |

|  | Berikut adalah hasil di tabel akhir trx\_done. |
| --- | --- |

Sehingga ETL Flownya akan terbentuk seperti ini:

| TRX | NPP |
| --- | --- |

**Pembuatan View Ticket Size**

Dengan rumus Ticket Size = NPP / 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 ticket\_size. |
| --- | --- |

|  | 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. |
| --- | --- |