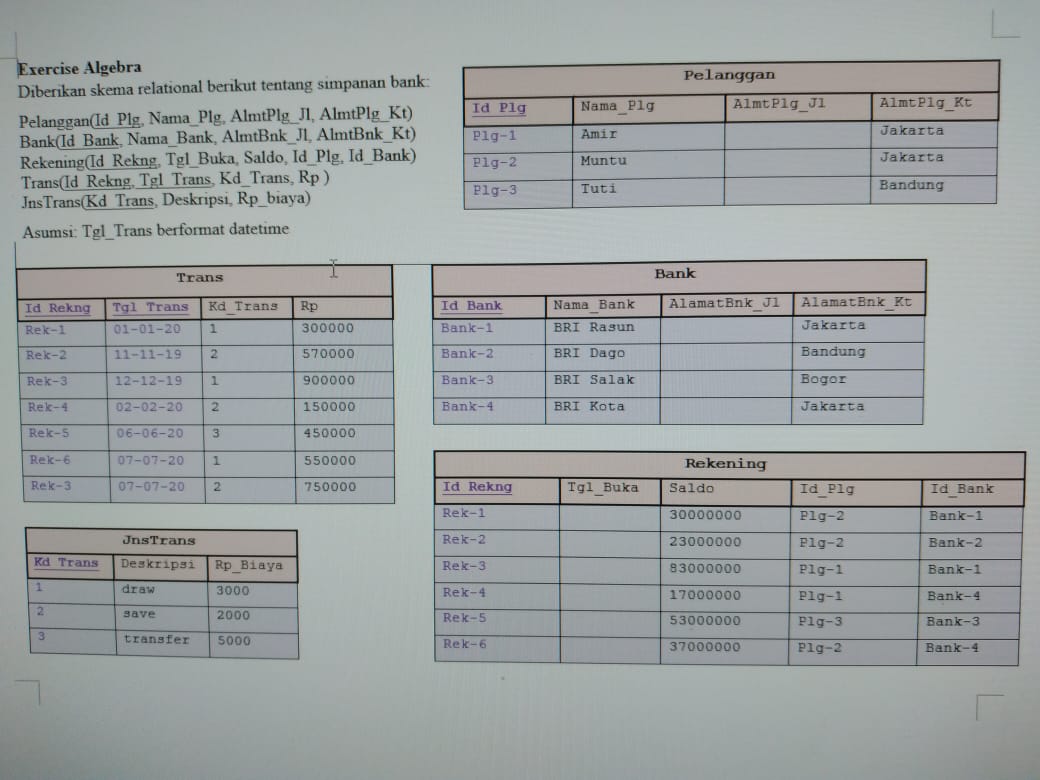
**Instruksi:**

1. Buatlah skema relasional menggunakan RelaX. Skema dan data didalamnya harus sesuai dengan gambar yang sudah diberikan. Lalu kerjakan Soal Latihan Algebra di bawah!
2. Latihan dikerjakan secara individu dan dikumpulkan di **Teams** pada **Assignment Latihan Algebra (Minggu ke-4)**.
3. Waktu pengumpulan jawaban sampai dengan hari ini pukul 23.59 WIB.

**Soal Latihan Algebra:**

1. Dapatkan pelanggan yang mempunyai nomor rekening di cabang “Jakarta” dan “Bogor” !  
   2. Dapatkan pelanggan yang mempunyai rekening di cabang kota “Jakarta” tetapi tidak di “Bogor”  
   3. Dapatkan pelanggan yang mempunyai nomor rekening di cabang “Jakarta” atau “Bogor” !  
   4. Dapatkan pelanggan yang mempunyai nomor rekening di cabang "Jakarta" dan "Bandung"!  
   5. Dapatkan semua pelanggan yang mempunyai rekening disemua cabang yang terletak di kota "Jakarta"!  
   6. Dapatkan pelanggan yang mempunyai simpanan dengan saldo terbesar!



Jawaban :

**Data Set**

group: Exercise

Bank = {

Id\_Bank:string, Nama\_Bank:string, AlamatBank\_Kt:string

'Bank-1', 'BRI Rasun', 'Jakarta'

'Bank-2', 'BRI Dago', 'Bandung'

'Bank-3', 'BRI Salak', 'Bogor'

'Bank-4', 'BRI Kota', 'Jakarta'

}

Rekening = {

Id\_Rekng:string, Saldo:number, Id\_Plg:string, Id\_Bank:string

'Rek-1', 30000000, 'Plg-2', 'Bank-1'

'Rek-2', 23000000, 'Plg-2', 'Bank-2'

'Rek-3', 83000000, 'Plg-1', 'Bank-1'

'Rek-4', 17000000, 'Plg-1', 'Bank-4'

'Rek-5', 53000000, 'Plg-3', 'Bank-3'

'Rek-6', 37000000, 'Plg-2', 'Bank-4'

}

Pelanggan = {

Id\_Plg:string, Nama\_Plg:string, AlmtPlg\_Jl:string, AlmtPlg\_Kt:string

Plg-1, Amir, Cilandak, Jakarta

Plg-2, Muntu, Cilandak, Jakarta

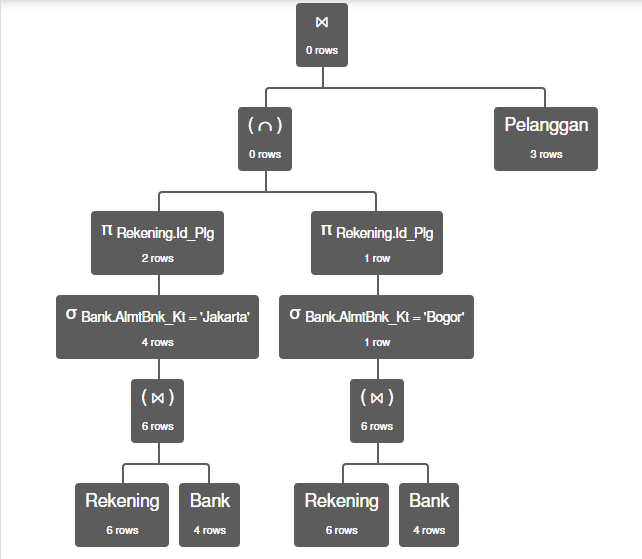
Plg-3, Tuti, 'Buah Batu', Bandung

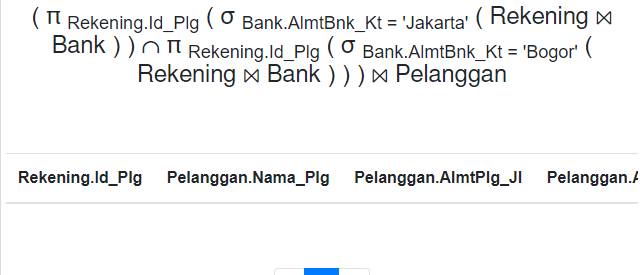
}

**Algebra Relational**

-- 1

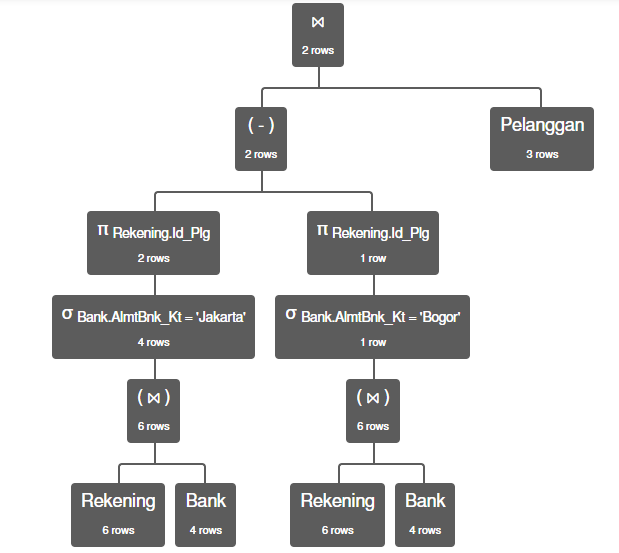
(π Rekening.Id\_Plg (σ Bank.AlmtBnk\_Kt='Jakarta' (Rekening ⨝ Bank)) ∩ π Rekening.Id\_Plg (σ Bank.AlmtBnk\_Kt='Bogor' (Rekening ⨝ Bank))) ⨝ Pelanggan

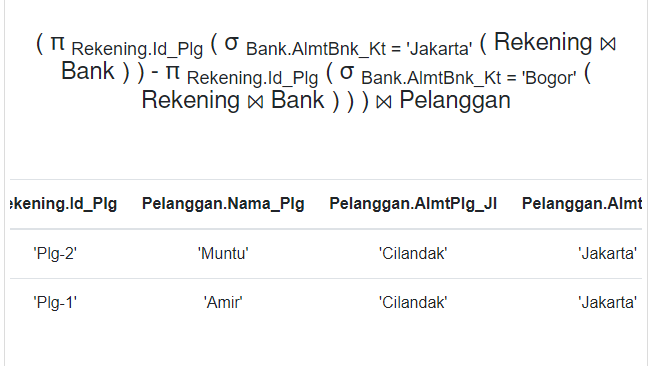




-- 2

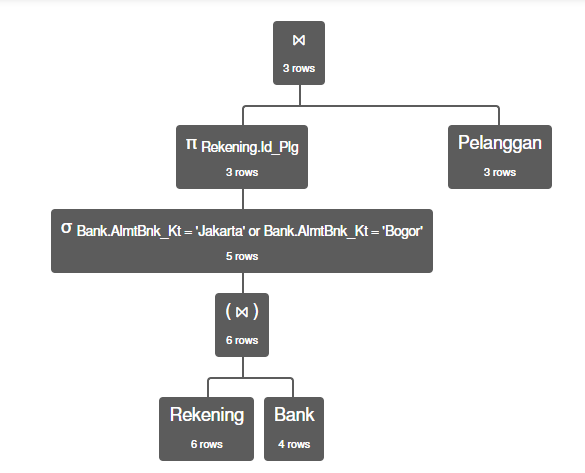
(π Rekening.Id\_Plg (σ Bank.AlmtBnk\_Kt='Jakarta' (Rekening ⨝ Bank)) - π Rekening.Id\_Plg (σ Bank.AlmtBnk\_Kt='Bogor' (Rekening ⨝ Bank))) ⨝ Pelanggan





-- 3

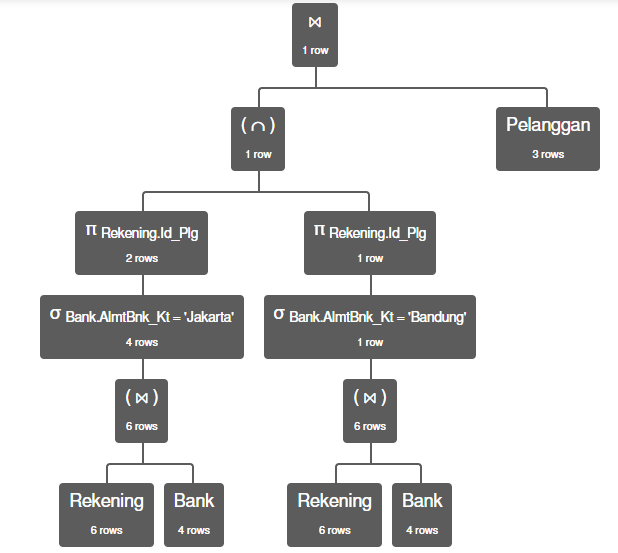
(π Rekening.Id\_Plg (σ Bank.AlmtBnk\_Kt='Jakarta' ∨ Bank.AlmtBnk\_Kt='Bogor' (Rekening ⨝ Bank))) ⨝ Pelanggan

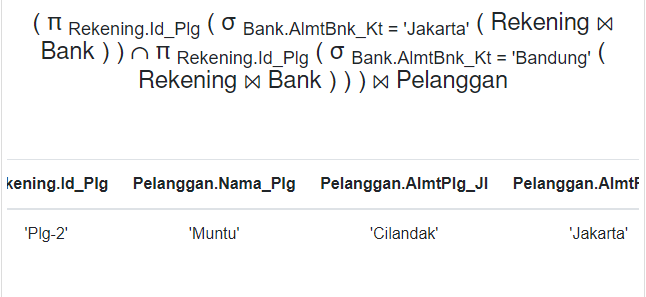




-- 4

(π Rekening.Id\_Plg (σ Bank.AlmtBnk\_Kt='Jakarta' (Rekening ⨝ Bank)) ∩ π Rekening.Id\_Plg (σ Bank.AlmtBnk\_Kt='Bandung' (Rekening ⨝ Bank))) ⨝ Pelanggan



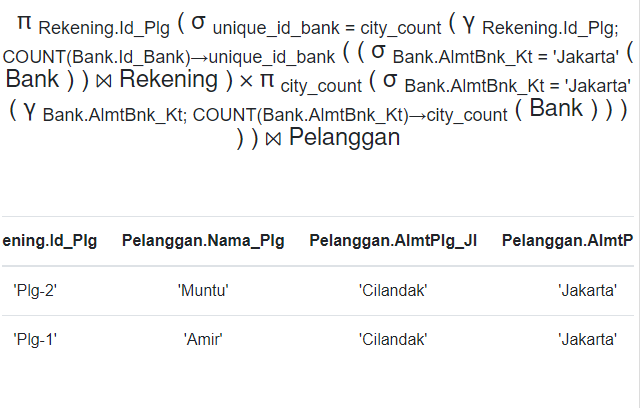


-- 5

jakarta\_count = π city\_count (σ Bank.AlmtBnk\_Kt='Jakarta' (γ Bank.AlmtBnk\_Kt; count(Bank.AlmtBnk\_Kt)→city\_count (Bank)))

bank\_count = γ Rekening.Id\_Plg; count(Bank.Id\_Bank)→unique\_id\_bank ((σ Bank.AlmtBnk\_Kt='Jakarta' (Bank)) ⨝ Rekening)

π Rekening.Id\_Plg (σ unique\_id\_bank=city\_count (bank\_count ⨯ jakarta\_count)) ⨝ Pelanggan



-- 6

RekSaldo1 = ρ Saldo→Saldo1 π Rekening.Saldo (Rekening)

RekSaldo2 = ρ Saldo→Saldo2 π Rekening.Saldo (Rekening)

π Rekening.Id\_Plg ((π Rekening.Saldo (Rekening) - π Rekening.Saldo1 (σ Rekening.Saldo1<Rekening.Saldo2 (RekSaldo1 x RekSaldo2))) ⨝ Rekening) ⨝ Pelanggan

