



NAMA : Narendra Daniswara Alamsyah
NIM : 244107060095
KELAS : SIB-2G
MATKUL : Basis Data Lanjut

Praktikum 00:

Pertanyaan 1:

a.



b.

1. Relasi nasabah - rekening (One-to-Many)

Foreign Key: rekening.nasabah_id -- nasabah.nasabah_id

Penjelasan: Satu nasabah dapat memiliki banyak rekening (tabungan, giro, deposito, dll), tetapi satu rekening hanya dimiliki oleh satu nasabah.

2. Relasi rekening - mutasi_rekening (One-to-Many)

Foreign Key: mutasi_rekening.rekening_id -- rekening.rekening_id

Penjelasan: Satu rekening dapat memiliki banyak mutasi, tetapi satu mutasi hanya terkait dengan satu rekening tertentu.

3. Relasi transaksi - mutasi_rekening (One-to-Many)

Foreign Key: mutasi_rekening.transaksi_id -- transaksi.transaksi_id

Penjelasan: Satu transaksi dapat menghasilkan beberapa mutasi (misalnya transfer menghasilkan 2 mutasi: debit di rekening asal dan kredit di rekening tujuan).

4. Relasi merchant - transaksi (One-to-Many)

Foreign Key: transaksi.merchant_id -- merchant.merchant_id

Penjelasan: Satu merchant dapat menerima banyak transaksi payment, tetapi satu transaksi payment hanya terkait dengan satu merchant. Relasi ini optional (dapat NULL) karena tidak semua transaksi melibatkan merchant.

5. Relasi Implisit transaksi - rekening

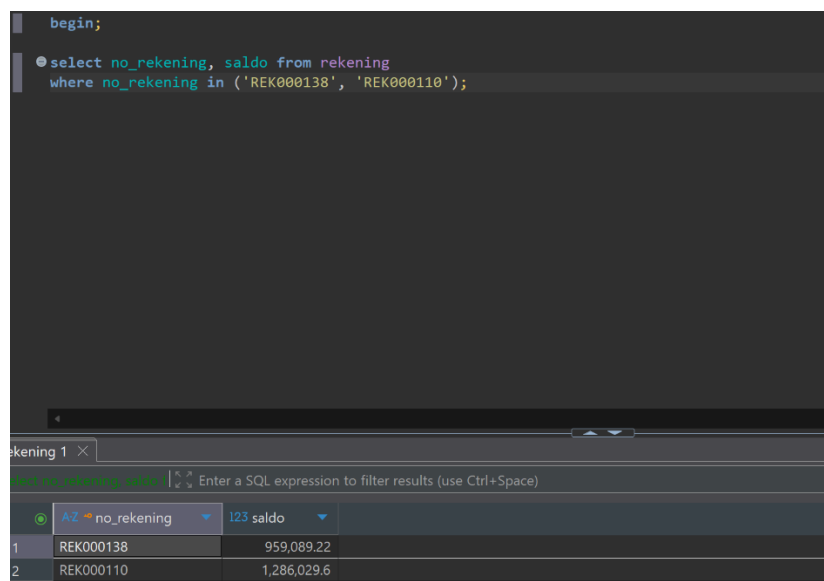
Melalui: transaksi.rekening_asal dan transaksi.rekening_tujuan

Penjelasan: Tabel transaksi memiliki referensi ke rekening melalui field rekening_asal dan rekening_tujuan, namun tidak digambarkan sebagai FK formal dalam diagram.

Praktikum 01:

1 – 3.

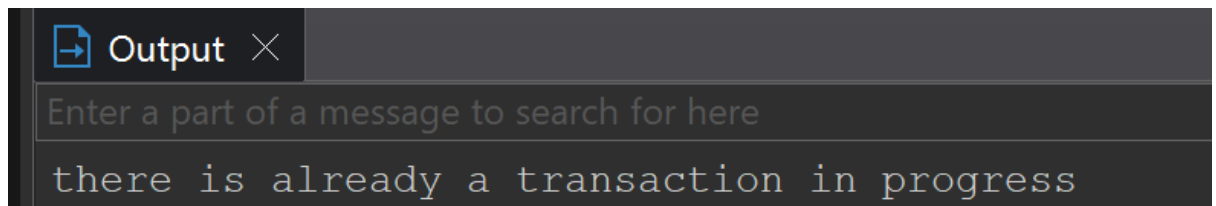
```
begin;
select no_rekening, saldo from rekening
where no_rekening in ('REK000138', 'REK000110');
```



The screenshot shows a database query result in a dark-themed interface. The query executed is: `select no_rekening, saldo from rekening where no_rekening in ('REK000138', 'REK000110');`. The result is displayed in a table with two columns: `no_rekening` and `saldo`. The first row shows `REK000138` with a balance of `959,089.22`. The second row shows `REK000110` with a balance of `1,286,029.6`.

	no_rekening	saldo
1	REK000138	959,089.22
2	REK000110	1,286,029.6

4.



5.

Statistics 1	
Name	Value
Updated Rows	1
Execute time	0.004s
Start time	Thu Oct 09 12:29:48 WIB 2025
Finish time	Thu Oct 09 12:29:48 WIB 2025
Query	update rekening set saldo = saldo - 100000 where no_rekening = 'REK000138'

6.

Statistics 1	
Name	Value
Updated Rows	1
Execute time	0.002s
Start time	Thu Oct 09 12:31:33 WIB 2025
Finish time	Thu Oct 09 12:31:33 WIB 2025
Query	update rekening set saldo = saldo + 100000 where no_rekening = 'REK000110'

7.

	A-Z no_rekening	123 saldo
1	REK000138	659,089.22
2	REK000110	1,486,029.6

8.

Name	Value
Updated Rows	1
Execute time	0.007s
Start time	Thu Oct 09 12:34:42 WIB 2025
Finish time	Thu Oct 09 12:34:42 WIB 2025
Query	<pre>insert into transaksi (rekening_asal, rekening_tujuan, jenis_transaksi, jumlah, berita, tanggal_transaksi, status) values ('REK000138', 'REK000110', 'TRANSFER', 100000, 'Transfer manual tanpa trigger', '2025-10-07', 'SUCCESS')</pre>

9.

```

SQL Error [42P01]: ERROR: relation "rek_asal" does not exist
Position: 146

insert into mutasi_rekening (rekening_id, transaksi_id, jenis_mutasi)
select
  (select rekening_id from rek_asal),
  currval('transaksi_transaksi_id_seq'),
  'DEBIT',
  100000,
  (select saldo from rekening where no_rekening = 'REK000138'),
  cast('15-01-2024' as date)
union all
select
  (select rekening_id from rek_tujuan),
  currval('transaksi_transaksi_id_seq'),
  'CREDIT',
  100000,
  (select saldo from rekening where no_rekening = 'REK000110'),
  cast('15-01-2024' as date)

```

10.

```

select
  r.no_rekening, r.saldo, m.jenis_mutasi,
  m.jumlah as mutasi_jumlah, m.saldo_akhir
from rekening r
left join mutasi_rekening m on r.rekening_id = m.rekening_id
where r.no_rekening in ('REK000138', 'REK000110')
and m.transaksi_id = currval('transaksi_transaksi_id_seq');

```

Statistics 1 X

```

SQL Error [25P02]: ERROR: current transaction is aborted,
commands ignored until end of transaction block

Error position: line: 5 pos: 145

select
  r.no_rekening, r.saldo, m.jenis_mutasi,
  m.jumlah as mutasi_jumlah, m.saldo_akhir
from rekening r
left join mutasi_rekening m on r.rekening_id = m.rekening_id
where r.no_rekening in ('REK000138', 'REK000110')
and m.transaksi_id = currval('transaksi_transaksi_id_seq')

```

11 – 12.

	AZ no_rekening	123 saldo	
1	REK000138	959,089.22	
2	REK000110	1,286,029.6	

13. Hasil Pengamatan

Langkah 3:

- REK00138: 959,089.22
- REK000110 1,286,029.6

Langkah 7:

- REK00138: 659,089.22
- REK000110 1,486,029.6

Langkah 10:

- REK00138: ERROR??
- REK000110 ERROR??

Langkah 12:



- REK00138: 959,089.22
- REK000110 1,286,029.6

14. Pertanyaan Analisis



- Jika transaksi berhenti di langkah 7 tanpa COMMIT, Data akan kembali ke kondisi sebelum BEGIN jika tidak di-COMMIT. Ini sesuai dengan prinsip Atomicity dalam ACID.
- Jika terjadi error di langkah 4 atau langkah 5 (UPDATE debit) Transaksi akan berhenti atau pada langkah tersebut. PostgreSQL akan otomatis membatalkan seluruh transaksi (implicit ROLLBACK).
- Jika menjalankan ROLLBACK setelah COMMIT, ROLLBACK TIDAK akan berpengaruh apa-apa. Data yang sudah di-COMMIT tetap tersimpan secara permanen dan tidak bisa dibatalkan

Praktikum 02:

1 – 3.

	A-Z  no_rekening ▼	123 saldo ▼	
1	REK000190	8,242,284.23	
2	REK000098	4,048,211.55	
3	REK000149	14,202,793.49	
4	REK000095	2,817,020.19	
5	REK000160	10,284,349.61	



4.

	A-Z  no_rekening ▼	123 saldo ▼	
1	REK000190	8,242,284.23	
2	REK000149	14,202,793.49	
3	REK000095	2,817,020.19	
4	REK000160	10,284,349.61	
5	REK000098	3,898,211.55	



5.

	REK000190	8,242,284.23
	REK000149	14,202,793.49
	REK000095	2,817,020.19
	REK000098	3,898,211.55
	REK000160	12,784,349.61

6.

	A-Z  no_rekening ▼	123 saldo ▼
	REK000190	8,242,284.23
	REK000098	4,048,211.55
	REK000149	14,202,793.49
	REK000095	2,817,020.19
	REK000160	10,284,349.61

7.

	A-Z  no_rekening ▼	123 saldo ▼
1	REK000190	8,242,284.23
2	REK000098	4,048,211.55
3	REK000149	14,202,793.49
4	REK000095	2,817,020.19
5	REK000160	10,284,349.61

8. Hasil Pengamatan:

Langkah 3:

- REK000190: 8,242,284.23
- REK000098: 4,048,211.55
- REK000149: 14,202,793.49
- REK000160: 2,817,020.19
- REK000095: 10,284,349.61

Langkah 4:

- REK000190: 8,242,284.23
- REK000098: 3,898,211.55
- REK000149: 14,202,793.49
- REK000160: 10,284,349.61
- REK000095: 2,817,020.19

Langkah 5:

- REK000190: 8,242,284.23
- REK000098: 3,898,211.55
- REK000149: 14,202,793.49
- REK000160: 12,784,349.61
- REK000095: 2,817,020.19

Langkah 6:

- REK000190: 8,242,284.23
- REK000098: 4,048,211.55
- REK000149: 14,202,793.49
- REK000160: 10,284,349.61
- REK000095: 2,817,020.19

Langkah 7:

- REK000190: 8,242,284.23
- REK000098: 4,048,211.55
- REK000149: 14,202,793.49
- REK000160: 10,284,349.61
- REK000095: 2,817,020.19

9. Pertanyaan Analisis

- Jika terjadi error di langkah 5, maka transaksi akan berhenti pada saat error terjadi. Lalu, PostgreSQL akan menandai transaksi dalam status "aborted".
- Jika menggunakan ROLLBACK tanpa SAVEPOINT, seluruh transaksi dibatalkan sejak BEGIN.

Praktikum 03:

1 – 4.

	A-Z no_rekening	123 saldo	
1	REK000101	6,810,934.13	
2	REK000173	4,844,481.38	

5.

	A-Z no_rekening ▼	123 saldo ▼
1	REK000101	6,810,934.13
2	REK000173	4,844,481.38

6.

	A-Z no_rekening ▼	123 saldo ▼
1	REK000101	6,810,934.13
2	REK000173	5,844,481.38

7.

	A-Z no_rekening ▼	123 saldo ▼	
1	REK000101	6,810,934.13	
2	REK000173	5,844,481.38	

8. Hasil Pengamatan

Langkah 4:

- REK000101: 6,810,934.13
- REK000173: 4,844,481.38

Langkah 5:

- REK000101: 6,810,934.13
- REK000173: 4,844,481.38

Langkah 6:

- REK000101: 6,810,934.13
- REK000173: 5,844,481.38

Langkah 7:

- REK000101: 6,810,934.13
- REK000173: 5,844,481.38

9. Pertanyaan Analisis





- a. Ketika Window 1 melakukan SELECT pertama kali setelah BEGIN, Window 1 akan melihat saldo sebelum ada perubahan dari Window 2.
- b. READ COMMITTED adalah trade-off antara konsistensi dan performa:
 - Cukup aman untuk mencegah dirty read.
 - Tidak menjamin konsistensi penuh dalam satu transaksi.
 - Performa tinggi dan cocok untuk aplikasi real-time seperti mobile banking.

Praktikum 04:

6.

	A-Z trigger_name ▼	A-Z table_name ▼	A-Z timing ▼	A-Z event ▼
1	trigger_validate_saldo_sebelum_transaksi	transaksi	BEFORE	INSERT
2	trigger_update_rekening_saldo	transaksi	AFTER	INSERT

7.

	A-Z  no_rekening 	123 saldo 	
1	REK000180	1,000,000	
2	REK000114	1,000,000	

8.

```
memproses update saldo untuk: topup
```

[illegible]

9.

	123 transaksi_id	AZ rekening_asal	AZ rekening_tujuan	AZ jenis_transaksi	123 jumlah	AZ berita
1	5,006	REK000114	[NULL]	payment	150,000	pembayaran an
2	156	REK000114	REK000114	TOPUP	248,259.55	Topup saldo
3	1,969	REK000114	REK000101	TRANSFER	447,921.38	Transfer ke REK
4	1,467	REK000114	REK000170	TRANSFER	142,484.1	Transfer ke REK
5	4,172	REK000114	[NULL]	WITHDRAWAL	303,071.07	Tarik tunai
6	2,524	REK000114	[NULL]	PAYMENT	280,587.64	Pembayaran To
7	2,756	REK000114	[NULL]	PAYMENT	166,722.26	Pembayaran Pu
8	2,939	REK000114	REK000114	TOPUP	501,101.66	Topup saldo
9	2,973	REK000114	REK000114	TOPUP	238,233.39	Topup saldo
10	3,608	REK000114	REK000163	TRANSFER	161,072.6	Transfer ke REK
11	1,723	REK000114	[NULL]	PAYMENT	332,304.75	Pembayaran Ne

10.

Name	Value
Updated Rows	1
Execute time	0.019s
Start time	Thu Oct 09 14:39:45 WIB 2025
Finish time	Thu Oct 09 14:39:45 WIB 2025
Query	insert into transaksi (rekening_asal, rekening_tujuan, jenis_transaksi, jumlah, berita, merchant_id, status) values ('REK000114', NULL, 'PAYMENT', 950000, 'Payment berbahaya', 3, 'SUCCESS')

11.

	A-Z no_rekening	123 saldo
1	REK000180	1,200,000
2	REK000114	850,000

	A-Z rekening_asal	A-Z jenis_transaksi	123 jumlah	A-Z status	🕒 tanggal_transaksi
1	REK000114	PAYMENT	950,000	SUCCESS	2025-10-09
2	REK000114	payment	150,000	success	2025-10-09
3	REK000180	topup	200,000	success	2025-10-09
4	REK000114	PAYMENT	950,000	SUCCESS	2025-10-09
5	REK000180	WITHDRAWAL	167,125.75	SUCCESS	2025-10-05
6	REK000114	TOPUP	248,259.55	SUCCESS	2025-10-04
7	REK000114	PAYMENT	166,722.26	SUCCESS	2025-10-03
8	REK000114	TRANSFER	447,921.38	SUCCESS	2025-10-03
9	REK000114	TRANSFER	142,484.1	SUCCESS	2025-10-03
10	REK000114	WITHDRAWAL	303,071.07	SUCCESS	2025-10-03
11	REK000114	PAYMENT	280,587.64	SUCCESS	2025-10-03
12	REK000114	TOPUP	501,101.66	SUCCESS	2025-10-02

12. Hasil Pengamatan

Langkah 6: Kedua trigger aktif.

Langkah 7:

- REK000114: 1,000,000
- REK000180: 1,000,000

Langkah 8:

- REK000180: 1,200,00
- Topup berhasil.

Langkah 9:

- Saldo payment berhasil.
- REK000114: 850,000

Langkah 10:

- Payment gagal.

Langkah 11:

- REK000114: 1,200,000
- REK000180: 850,000

13. Pertanyaan Analisis

a.

- Validasi saldo menggunakan BEFORE INSERT karena trigger tersebut mencegah transaksi invalid masuk ke database. Selain itu, jika validasi gagal, transaksi dibatalkan sebelum data tersimpan.
- Update saldo menggunakan AFTER INSERT karena trigger tersebut memastikan transaksi sudah berhasil tersimpan sebelum update saldo. Jika update saldo gagal, transaksi sudah tercatat dan bisa di-handle.

b.

