

**TUGAS KELOMPOK PRAKTIKUM MANAJEMEN BASIS
DATA PERTEMUAN-4 TENTANG INDEX & POINTERS**



Asisten Praktikum :

Alya Safina & Izazava Putri Asari

Disusun oleh Kelompok 6 Lab A2 :

Aditya Arif Setiawan (24060121140144)

Alfonso Michael N H (24060123140156)

Anggita Mahardika Syarifa (24060119130089)

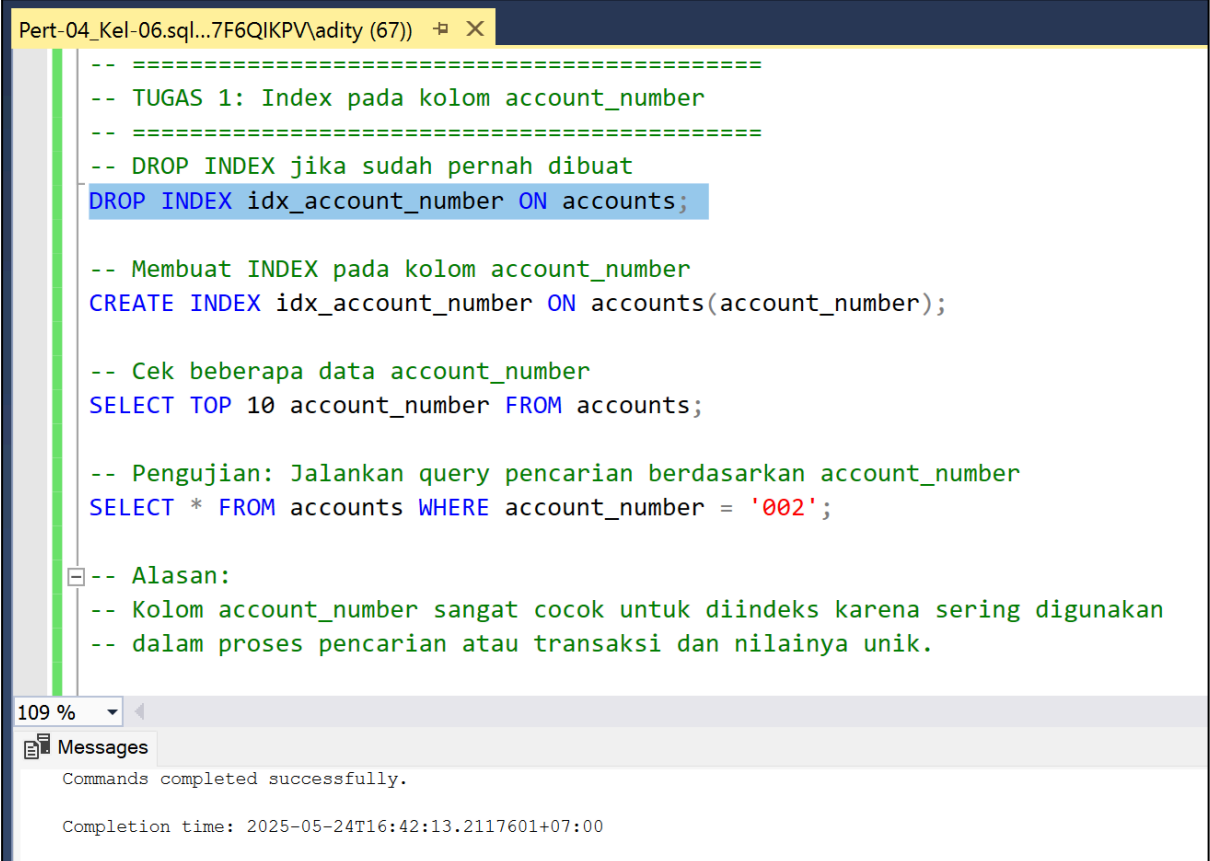
**DEPARTEMEN ILMU KOMPUTER/INFORMATIKA
FAKULTAS SAINS DAN MATEMATIKA
UNIVERSITAS DIPONEGORO**

2025

TOPIK 5 INDEX & POINTERS

1. Buatlah index pada tabel accounts untuk mempercepat pencarian. Pilih salah satu kolom pada table tersebut dan berikan alasan pemilihan kolom tersebut sebagai index!

Screenshot:



```
-- =====
-- TUGAS 1: Index pada kolom account_number
-- =====
-- DROP INDEX jika sudah pernah dibuat
DROP INDEX idx_account_number ON accounts;

-- Membuat INDEX pada kolom account_number
CREATE INDEX idx_account_number ON accounts(account_number);

-- Cek beberapa data account_number
SELECT TOP 10 account_number FROM accounts;

-- Pengujian: Jalankan query pencarian berdasarkan account_number
SELECT * FROM accounts WHERE account_number = '002';

-- Alasan:
-- Kolom account_number sangat cocok untuk diindeks karena sering digunakan
-- dalam proses pencarian atau transaksi dan nilainya unik.
```

109 %

Messages

Commands completed successfully.

Completion time: 2025-05-24T16:42:13.2117601+07:00

```
Pert-04_Kel-06.sql...7F6QIKPV\adity (67) X
-- =====
-- TUGAS 1: Index pada kolom account_number
-- =====
-- DROP INDEX jika sudah pernah dibuat
DROP INDEX idx_account_number ON accounts;

-- Membuat INDEX pada kolom account_number
CREATE INDEX idx_account_number ON accounts(account_number);

-- Cek beberapa data account_number
SELECT TOP 10 account_number FROM accounts;

-- Pengujian: Jalankan query pencarian berdasarkan account_number
SELECT * FROM accounts WHERE account_number = '002';

-- Alasan:
-- Kolom account_number sangat cocok untuk diindeks karena sering digunakan
-- dalam proses pencarian atau transaksi dan nilainya unik.
```

109 %

Messages

Commands completed successfully.

Completion time: 2025-05-24T16:43:00.4471979+07:00

```
Pert-04_Kel-06.sql...7F6QIKPV\adity (67) X
-- =====
-- TUGAS 1: Index pada kolom account_number
-- =====
-- DROP INDEX jika sudah pernah dibuat
DROP INDEX idx_account_number ON accounts;

-- Membuat INDEX pada kolom account_number
CREATE INDEX idx_account_number ON accounts(account_number);

-- Cek beberapa data account_number
SELECT TOP 10 account_number FROM accounts;

-- Pengujian: Jalankan query pencarian berdasarkan account_number
SELECT * FROM accounts WHERE account_number = '002';

-- Alasan:
-- Kolom account_number sangat cocok untuk diindeks karena sering digunakan
-- dalam proses pencarian atau transaksi dan nilainya unik.
```

109 %

Results Messages

	account_number
1	001
2	002

```
Pert-04_Kel-06.sql...7F6QIKPV\adity (67))
-- =====
-- TUGAS 1: Index pada kolom account_number
-- =====
-- DROP INDEX jika sudah pernah dibuat
DROP INDEX idx_account_number ON accounts;

-- Membuat INDEX pada kolom account_number
CREATE INDEX idx_account_number ON accounts(account_number);

-- Cek beberapa data account_number
SELECT TOP 10 account_number FROM accounts;

-- Pengujian: Jalankan query pencarian berdasarkan account_number
SELECT * FROM accounts WHERE account_number = '002';

-- Alasan:
-- Kolom account_number sangat cocok untuk diindeks karena sering digunakan
-- dalam proses pencarian atau transaksi dan nilainya unik.
```

109 %

Results Messages

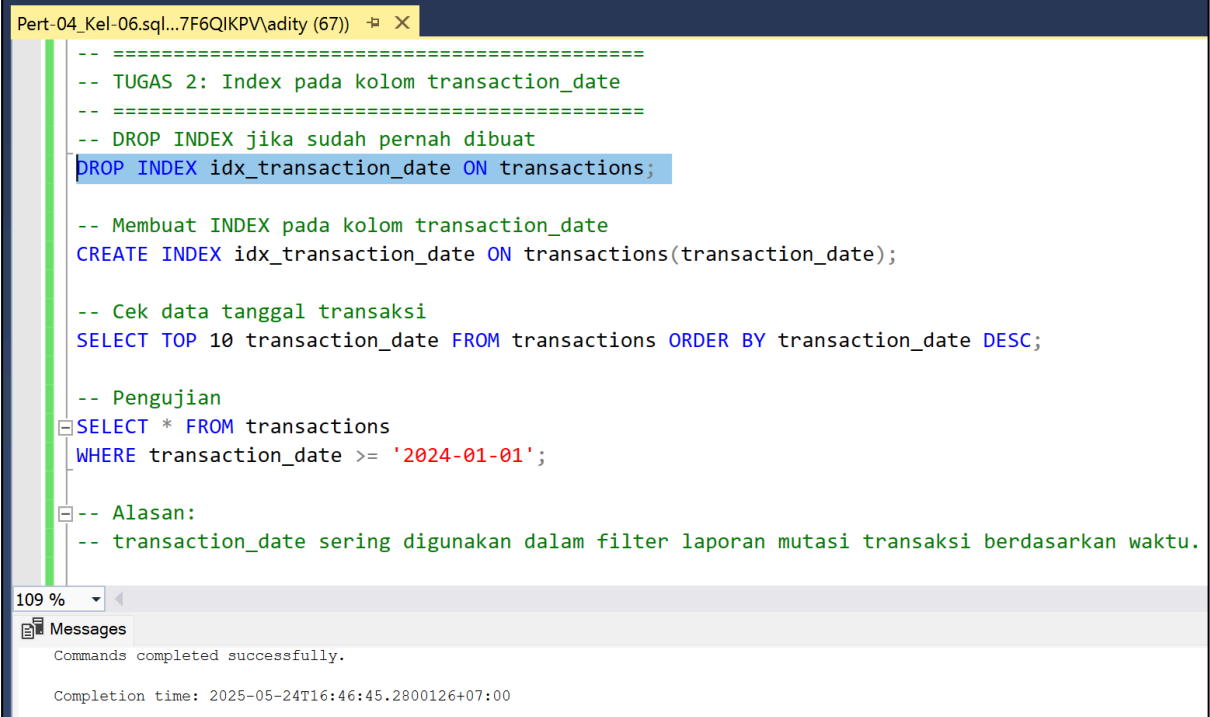
	account_id	customer_id	account_number	account_type	balance	created_at
1	ACC002	CUST002	002	savings	150000.00	2025-05-21 14:19:44.893

Alasan:

Kolom **account_number** sangat cocok untuk diindeks karena sering digunakan dalam proses pencarian atau transaksi dan nilainya unik.

2. Buatlah index pada tabel transactions untuk mempercepat pencarian. Pilih salah satu kolom pada table tersebut dan berikan alasan pemilihan kolom tersebut sebagai index!

Screenshot:



```
-- =====
-- TUGAS 2: Index pada kolom transaction_date
-- =====
-- DROP INDEX jika sudah pernah dibuat
DROP INDEX idx_transaction_date ON transactions;

-- Membuat INDEX pada kolom transaction_date
CREATE INDEX idx_transaction_date ON transactions(transaction_date);

-- Cek data tanggal transaksi
SELECT TOP 10 transaction_date FROM transactions ORDER BY transaction_date DESC;

-- Pengujian
SELECT * FROM transactions
WHERE transaction_date >= '2024-01-01';

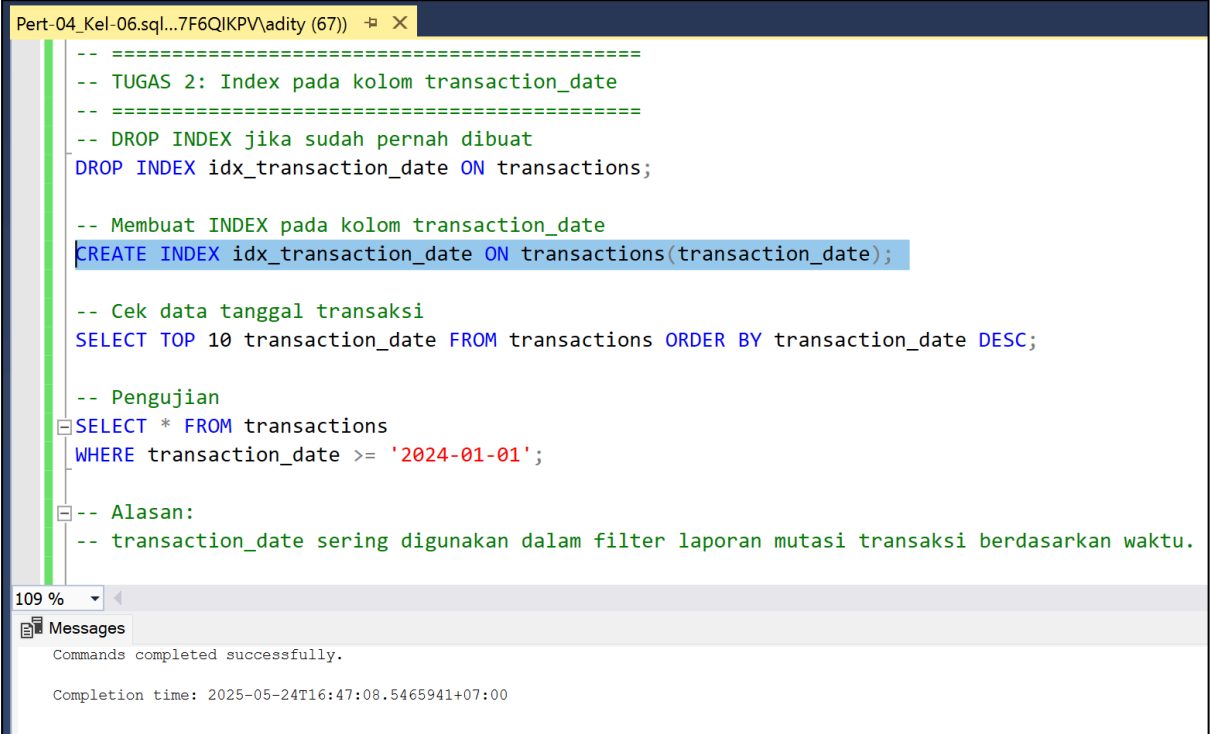
-- Alasan:
-- transaction_date sering digunakan dalam filter laporan mutasi transaksi berdasarkan waktu.
```

109 %

Messages

Commands completed successfully.

Completion time: 2025-05-24T16:46:45.2800126+07:00



```
-- =====
-- TUGAS 2: Index pada kolom transaction_date
-- =====
-- DROP INDEX jika sudah pernah dibuat
DROP INDEX idx_transaction_date ON transactions;

-- Membuat INDEX pada kolom transaction_date
CREATE INDEX idx_transaction_date ON transactions(transaction_date);

-- Cek data tanggal transaksi
SELECT TOP 10 transaction_date FROM transactions ORDER BY transaction_date DESC;

-- Pengujian
SELECT * FROM transactions
WHERE transaction_date >= '2024-01-01';

-- Alasan:
-- transaction_date sering digunakan dalam filter laporan mutasi transaksi berdasarkan waktu.
```

109 %

Messages

Commands completed successfully.

Completion time: 2025-05-24T16:47:08.5465941+07:00

Pert-04_Kel-06.sql...7F6QIKPV\adity (67))

```
-- =====
-- TUGAS 2: Index pada kolom transaction_date
-- =====
-- DROP INDEX jika sudah pernah dibuat
DROP INDEX idx_transaction_date ON transactions;

-- Membuat INDEX pada kolom transaction_date
CREATE INDEX idx_transaction_date ON transactions(transaction_date);

-- Cek data tanggal transaksi
SELECT TOP 10 transaction_date FROM transactions ORDER BY transaction_date DESC;

-- Pengujian
SELECT * FROM transactions
WHERE transaction_date >= '2024-01-01';

-- Alasan:
-- transaction_date sering digunakan dalam filter laporan mutasi transaksi berdasarkan waktu.
```

109 %

Results Messages

	transaction_date
1	2025-05-21 14:21:20.357
2	2025-05-21 14:20:54.777

Pert-04_Kel-06.sql...7F6QIKPV\adity (67))

```
-- =====
-- TUGAS 2: Index pada kolom transaction_date
-- =====
-- DROP INDEX jika sudah pernah dibuat
DROP INDEX idx_transaction_date ON transactions;

-- Membuat INDEX pada kolom transaction_date
CREATE INDEX idx_transaction_date ON transactions(transaction_date);

-- Cek data tanggal transaksi
SELECT TOP 10 transaction_date FROM transactions ORDER BY transaction_date DESC;

-- Pengujian
SELECT * FROM transactions
WHERE transaction_date >= '2024-01-01';

-- Alasan:
-- transaction_date sering digunakan dalam filter laporan mutasi transaksi berdasarkan waktu.
```

109 %

Results Messages

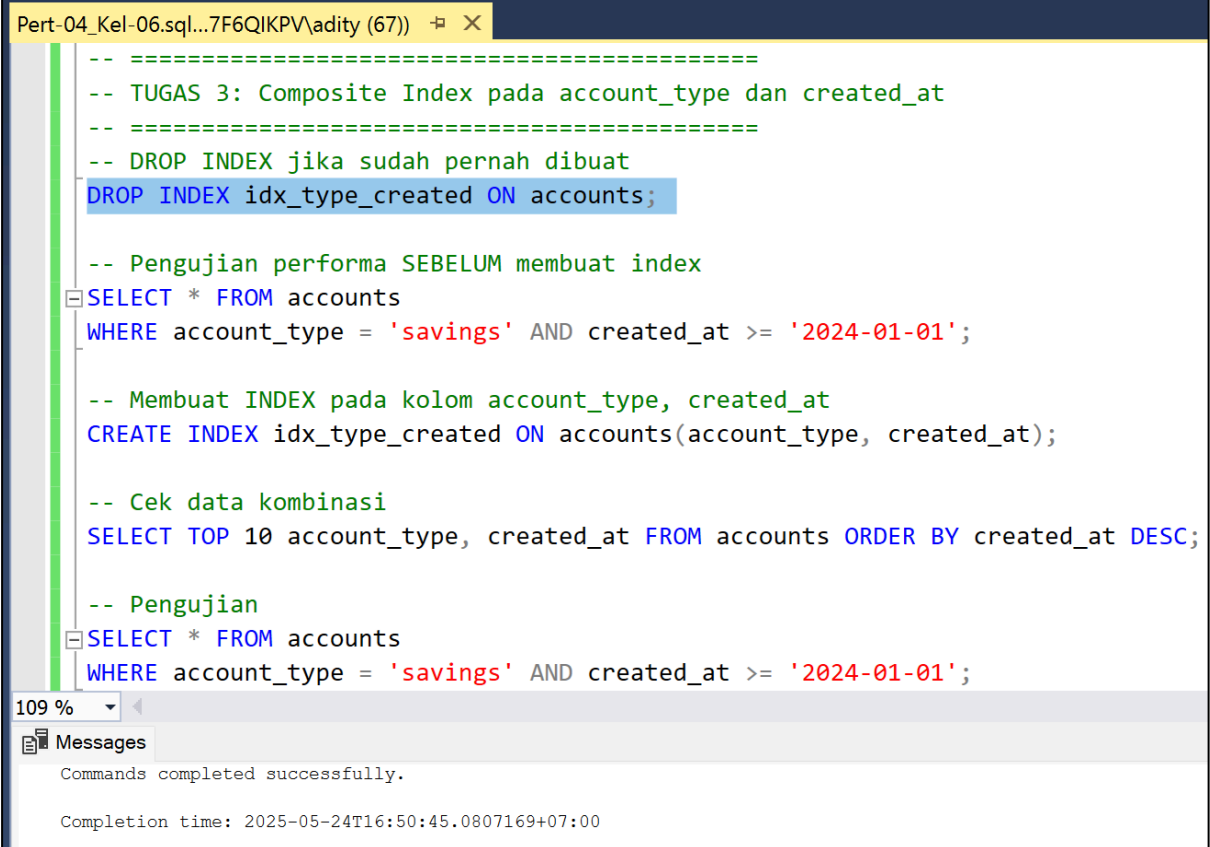
	transaction_id	account_id	transaction_type_id	amount	transaction_date	description	reference_account
1	382D1116-3045-4493-B11B-40E93F828FFE	ACC001	3	50000.00	2025-05-21 14:21:20.357	Test berhasil	NULL
2	9BA28F66-94B5-4017-8404-31AFDE661B9B	ACC001	3	150000.00	2025-05-21 14:20:54.777	Test gagal	NULL

Alasan:

transaction_date sering digunakan dalam filter laporan mutasi transaksi berdasarkan waktu.

3. Buatlah composite index pada tabel accounts. Pilih lebih dari satu kolom pada tabel tersebut dan uji performa query sebelum dan setelah menggunakan index! (query pengujian berdasarkan kolom yang dipilih sebagai index)

Screenshot:



```
Pert-04_Kel-06.sql...7F6QIKPV\adity (67) X
-- =====
-- TUGAS 3: Composite Index pada account_type dan created_at
-- =====
-- DROP INDEX jika sudah pernah dibuat
DROP INDEX idx_type_created ON accounts;

-- Pengujian performa SEBELUM membuat index
SELECT * FROM accounts
WHERE account_type = 'savings' AND created_at >= '2024-01-01';

-- Membuat INDEX pada kolom account_type, created_at
CREATE INDEX idx_type_created ON accounts(account_type, created_at);

-- Cek data kombinasi
SELECT TOP 10 account_type, created_at FROM accounts ORDER BY created_at DESC;

-- Pengujian
SELECT * FROM accounts
WHERE account_type = 'savings' AND created_at >= '2024-01-01';

109 %
Messages
Commands completed successfully.

Completion time: 2025-05-24T16:50:45.0807169+07:00
```

```
Pert-04_Kel-06.sql...7F6QIKPV\adity (67))  X
-- =====
-- TUGAS 3: Composite Index pada account_type dan created_at
-- =====
-- DROP INDEX jika sudah pernah dibuat
DROP INDEX idx_type_created ON accounts;

-- Pengujian performa SEBELUM membuat index
SELECT * FROM accounts
WHERE account_type = 'savings' AND created_at >= '2024-01-01';

-- Membuat INDEX pada kolom account_type, created_at
CREATE INDEX idx_type_created ON accounts(account_type, created_at);

-- Cek data kombinasi
SELECT TOP 10 account_type, created_at FROM accounts ORDER BY created_at DESC;

-- Pengujian
SELECT * FROM accounts
WHERE account_type = 'savings' AND created_at >= '2024-01-01';
```

109 %

Results Messages

	account_id	customer_id	account_number	account_type	balance	created_at
1	ACC001	CUST001	001	savings	500000.00	2025-05-21 14:19:44.893
2	ACC002	CUST002	002	savings	150000.00	2025-05-21 14:19:44.893

```
Pert-04_Kel-06.sql...7F6QIKPV\adity (67))  X
-- =====
-- TUGAS 3: Composite Index pada account_type dan created_at
-- =====
-- DROP INDEX jika sudah pernah dibuat
DROP INDEX idx_type_created ON accounts;

-- Pengujian performa SEBELUM membuat index
SELECT * FROM accounts
WHERE account_type = 'savings' AND created_at >= '2024-01-01';

-- Membuat INDEX pada kolom account_type, created_at
CREATE INDEX idx_type_created ON accounts(account_type, created_at);

-- Cek data kombinasi
SELECT TOP 10 account_type, created_at FROM accounts ORDER BY created_at DESC;

-- Pengujian
SELECT * FROM accounts
WHERE account_type = 'savings' AND created_at >= '2024-01-01';
```

109 %

Messages

Commands completed successfully.

Completion time: 2025-05-24T16:51:38.3210162+07:00


```
Pert-04_Kel-06.sql...7F6QIKPV\adity (67) X
-- DROP INDEX jika sudah pernah dibuat
DROP INDEX idx_type_created ON accounts;

-- Pengujian performa SEBELUM membuat index
SELECT * FROM accounts
WHERE account_type = 'savings' AND created_at >= '2024-01-01';

-- Membuat INDEX pada kolom account_type, created_at
CREATE INDEX idx_type_created ON accounts(account_type, created_at);

-- Cek data kombinasi
SELECT TOP 10 account_type, created_at FROM accounts ORDER BY created_at DESC;

-- Pengujian
SELECT * FROM accounts
WHERE account_type = 'savings' AND created_at >= '2024-01-01';

-- Alasan:
-- Kombinasi kolom ini sangat umum digunakan dalam laporan pembukaan akun berdasarkan jenis dan tanggal.
```

109 %

Results Messages

	account_type	created_at
1	savings	2025-05-21 14:19:44.893
2	savings	2025-05-21 14:19:44.893

```
Pert-04_Kel-06.sql...7F6QIKPV\adity (67) X
-- DROP INDEX jika sudah pernah dibuat
DROP INDEX idx_type_created ON accounts;

-- Pengujian performa SEBELUM membuat index
SELECT * FROM accounts
WHERE account_type = 'savings' AND created_at >= '2024-01-01';

-- Membuat INDEX pada kolom account_type, created_at
CREATE INDEX idx_type_created ON accounts(account_type, created_at);

-- Cek data kombinasi
SELECT TOP 10 account_type, created_at FROM accounts ORDER BY created_at DESC;

-- Pengujian
SELECT * FROM accounts
WHERE account_type = 'savings' AND created_at >= '2024-01-01';

-- Alasan:
-- Kombinasi kolom ini sangat umum digunakan dalam laporan pembukaan akun berdasarkan jenis dan tanggal.
```

109 %

Results Messages

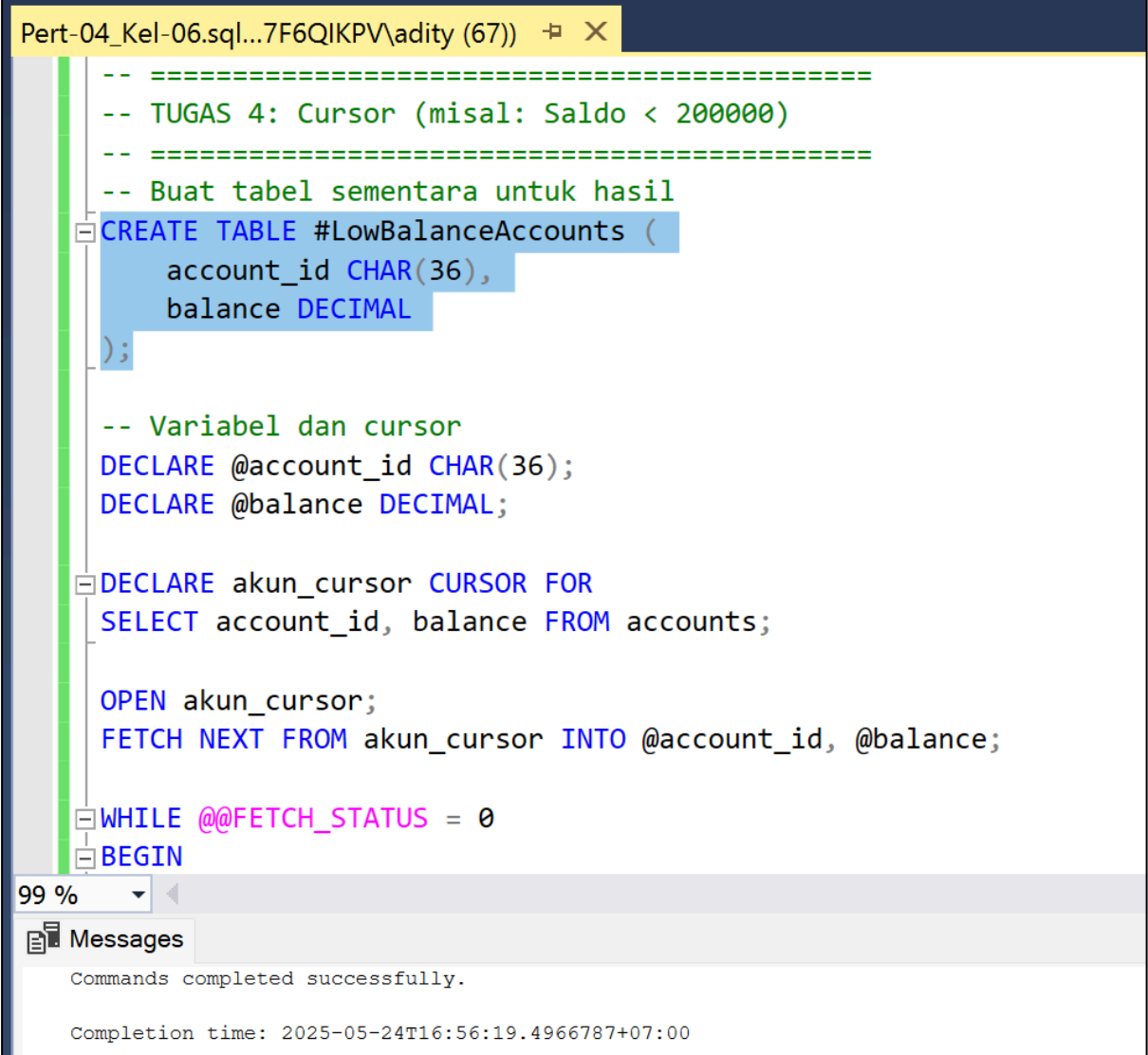
	account_id	customer_id	account_number	account_type	balance	created_at
1	ACC001	CUST001	001	savings	500000.00	2025-05-21 14:19:44.893
2	ACC002	CUST002	002	savings	150000.00	2025-05-21 14:19:44.893

Alasan:

Kombinasi kolom ini sangat umum digunakan dalam laporan pembukaan akun berdasarkan jenis dan tanggal.

4. Buat cursor untuk membaca semua akun dari tabel accounts, lalu periksa balance masing-masing. Jika balance < 9.000 (catatan: yang digunakan **balance < 200.000**, agar terdeteksi), tampilkan pesan: "Saldo rendah untuk account_id: [account_id]".

Screenshot:



```
Pert-04_Kel-06.sql...7F6QIKPV\adity (67))
-- =====
-- TUGAS 4: Cursor (misal: Saldo < 200000)
-- =====
-- Buat tabel sementara untuk hasil
CREATE TABLE #LowBalanceAccounts (
    account_id CHAR(36),
    balance DECIMAL
);

-- Variabel dan cursor
DECLARE @account_id CHAR(36);
DECLARE @balance DECIMAL;

DECLARE akun_cursor CURSOR FOR
SELECT account_id, balance FROM accounts;

OPEN akun_cursor;
FETCH NEXT FROM akun_cursor INTO @account_id, @balance;

WHILE @@FETCH_STATUS = 0
BEGIN
```

99 %

Messages

Commands completed successfully.

Completion time: 2025-05-24T16:56:19.4966787+07:00

Pert-04_Kel-06.sql...7F6QIKPV\adity (67))

```
-- Variabel dan cursor
```

```
DECLARE @account_id CHAR(36);
```

```
DECLARE @balance DECIMAL;
```

```
DECLARE akun_cursor CURSOR FOR
```

```
SELECT account_id, balance FROM accounts;
```

```
OPEN akun_cursor;
```

```
FETCH NEXT FROM akun_cursor INTO @account_id, @balance;
```

```
WHILE @@FETCH_STATUS = 0
```

```
BEGIN
```

```
    IF @balance < 200000
```

```
    BEGIN
```

```
        INSERT INTO #LowBalanceAccounts(account_id, balance)
```

```
        VALUES (@account_id, @balance);
```

```
    END
```

```
    FETCH NEXT FROM akun_cursor INTO @account_id, @balance;
```

```
END;
```

99 %

Messages

(1 row affected)

Completion time: 2025-05-24T16:57:29.6964195+07:00

Pert-04_Kel-06.sql...7F6QIKPV\adity (67))

```
BEGIN
  IF @balance < 200000
  BEGIN
    INSERT INTO #LowBalanceAccounts(account_id, balance)
    VALUES (@account_id, @balance);
  END
  FETCH NEXT FROM akun_cursor INTO @account_id, @balance;
END;

-- Tutup cursor
CLOSE akun_cursor;
DEALLOCATE akun_cursor;

-- Tampilkan hasil
SELECT 'Saldo rendah untuk account_id: ' + account_id AS Warning, balance
FROM #LowBalanceAccounts;

-- Hapus tabel sementara
DROP TABLE #LowBalanceAccounts;
```

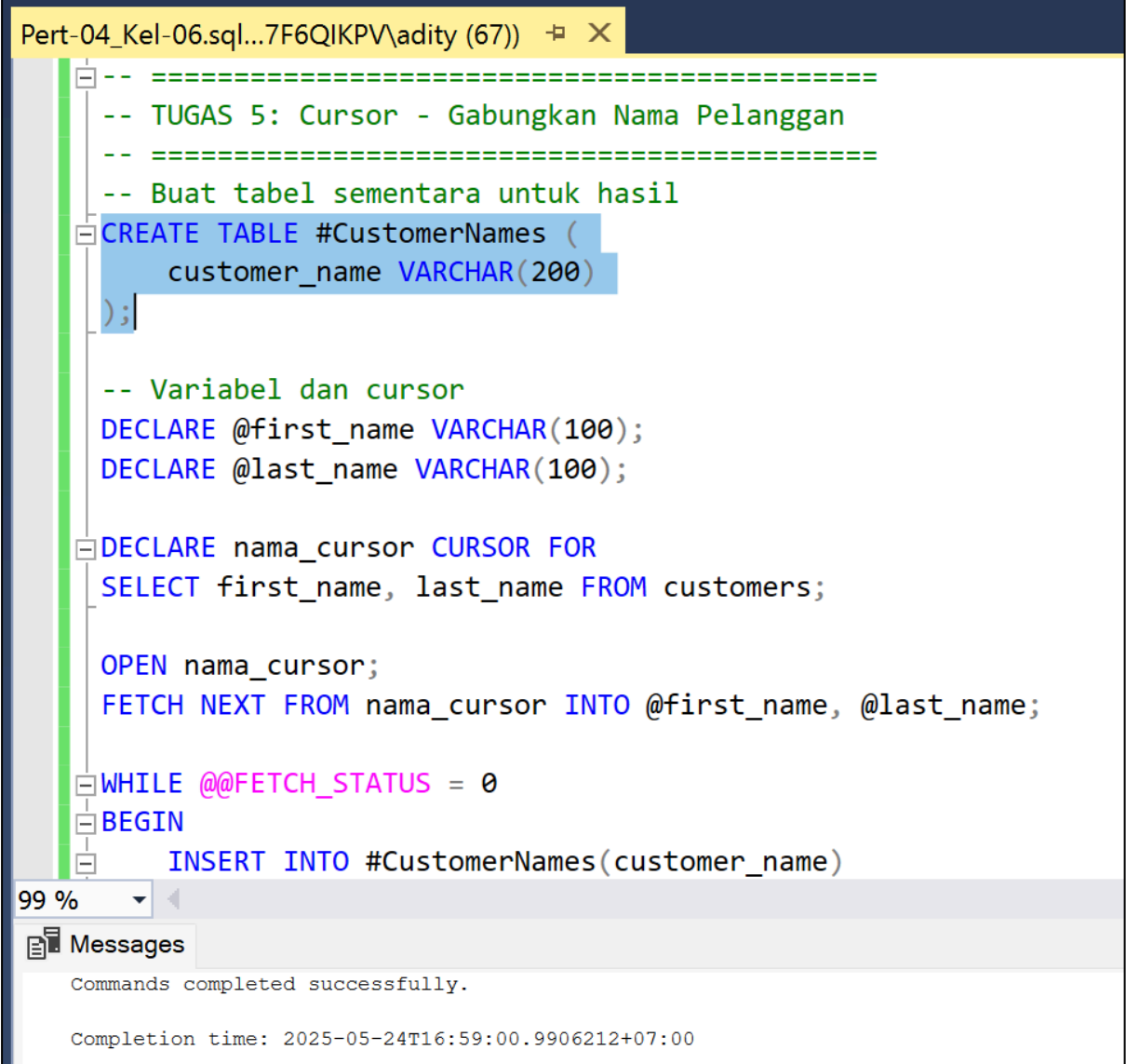
99 %

Results Messages

	Warning	balance
1	Saldo rendah untuk account_id: ACC002	150000

5. Buat cursor untuk membaca semua pelanggan (customers) dan gabungkan **first_name** dan **last_name** untuk ditampilkan dengan format: "Customer: [Nama Lengkap]".

Screenshot:



```
Pert-04_Kel-06.sql...7F6QIKPV\adity (67)) X
-- =====
-- TUGAS 5: Cursor - Gabungkan Nama Pelanggan
-- =====
-- Buat tabel sementara untuk hasil
CREATE TABLE #CustomerNames (
    customer_name VARCHAR(200)
);

-- Variabel dan cursor
DECLARE @first_name VARCHAR(100);
DECLARE @last_name VARCHAR(100);

DECLARE nama_cursor CURSOR FOR
SELECT first_name, last_name FROM customers;

OPEN nama_cursor;
FETCH NEXT FROM nama_cursor INTO @first_name, @last_name;

WHILE @@FETCH_STATUS = 0
BEGIN
    INSERT INTO #CustomerNames(customer_name)
```

99 %

Messages

Commands completed successfully.

Completion time: 2025-05-24T16:59:00.9906212+07:00

```
);

-- Variabel dan cursor
DECLARE @first_name VARCHAR(100);
DECLARE @last_name VARCHAR(100);

DECLARE nama_cursor CURSOR FOR
SELECT first_name, last_name FROM customers;

OPEN nama_cursor;
FETCH NEXT FROM nama_cursor INTO @first_name, @last_name;

WHILE @@FETCH_STATUS = 0
BEGIN
    INSERT INTO #CustomerNames(customer_name)
    VALUES ('Customer: ' + @first_name + ' ' + @last_name);
    FETCH NEXT FROM nama_cursor INTO @first_name, @last_name;
END;

-- Tutup cursor
CLOSE nama_cursor;
```

99 %

Messages

(1 row affected)
 (1 row affected)
 (1 row affected)
 (1 row affected)
 (1 row affected)
 (1 row affected)
 (1 row affected)
 (1 row affected)
 (1 row affected)
 (1 row affected)
 (1 row affected)
 (1 row affected)
 (1 row affected)

99 %

✓ Query executed successfully.

Pert-04_Kel-06.sql...7F6QIKPV\adity (67))

```
WHILE @@FETCH_STATUS = 0
BEGIN
    INSERT INTO #CustomerNames(customer_name)
    VALUES ('Customer: ' + @first_name + ' ' + @last_name);
    FETCH NEXT FROM nama_cursor INTO @first_name, @last_name;
END;

-- Tutup cursor
CLOSE nama_cursor;
DEALLOCATE nama_cursor;

-- Tampilkan hasil
SELECT * FROM #CustomerNames;

-- Hapus tabel sementara
DROP TABLE #CustomerNames;
```

99 %

Results Messages

	customer_name
1	Customer: Sergeant Maith
2	Customer: Shaine Matic
3	Customer: Buddie Sandison
4	Customer: Stephanus Frear
5	Customer: Corette Vicent
6	Customer: Devon McQuode
7	Customer: Jacquenette Hilldrup
8	Customer: Bobette Shambrooke
9	Customer: Donnajean Amerighi
10	Customer: Barbabra Swires
11	Customer: Sandro Alten
12	Customer: Nester Niccolls
13	Customer: Rebekah Conkie
14	Customer: Sidonnie Marks
15	Customer: Niel Yuryatin
16	Customer: Spike Sill
17	Customer: Billye Lyfe
18	Customer: Meggi MacDermott
19	Customer: Horst Morecomb
20	Customer: Jyoti Treanor
21	Customer: Boy Rowsby
22	Customer: Galen Biddy

✓ Query executed successfully.