
**LAPORAN PRAKTIKUM MAHASISWA
PEMEROGRAMAN SQL II**

“SELECT, SUB QUERY DAN JOIN”

Minggu 5, 6, dan 7



Oleh:

NAMA : Haldian

NPM : 20753050

KELAS : Manajemen Informatika B

Dosen : Eko Win Kenali, S.Kom, M.Cs

**MANAJEMEN INFORMATIKA
JURUSAN EKONOMI DAN BISNIS
POLITEKNIK NEGERI LAMPUNG
2022**

KATA PENGANTAR

Alhamdulillah, puji syukur ke hadirat Allah SWT yang telah memberikan rahmat dan inayah-Nya sehingga kami dapat menyelesaikan Laporan Praktikum Pemrograman SQL Lanjut tentang *Select, Sub Query dan Join*.

Terima kasih saya ucapan kepada Bapak Eko Win Kenali yang telah membantu kami baik secara moral maupun materi. Terima kasih juga saya ucapan kepada teman-teman seperjuangan yang telah mendukung kami sehingga kami bisa menyelesaikan tugas ini tepat waktu.

Kami menyadari, bahwa laporan praktikum yang kami buat ini masih jauh dari kata sempurna baik segi penyusunan, bahasa, maupun penulisannya. Oleh karena itu, kami sangat mengharapkan kritik dan saran yang membangun dari semua pembaca guna menjadi acuan agar penulis bisa menjadi lebih baik lagi di masa mendatang. Semoga laporan praktikum ini bisa menambah wawasan para pembaca dan bisa bermanfaat untuk perkembangan dan peningkatan ilmu pengetahuan.

Kalainda, 5 April 2022

HALDIAN

DAFTAR ISI

DAFTAR ISI.....	3
BAB 1. PENDAHULUAN	4
A. LATAR BELAKANG.....	4
B. MASALAH	4
C. TUJUAN	4
BAB 2. LANDASAN TEORI	5
A.<i>TRANSACT-SQL SELECT</i>	5
B. <i>TRANSACT-SQL SELECT DENGAN JOIN</i>.....	8
C. <i>TRANSACT-SQL SELECT DENGAN FUNGSI BUILT-IN</i>	11
BAB 3. HASIL DAN PEMBAHASAN.....	15
A. TAHAPAN PRAKTIKUM.....	15
1. Minggu #5	15
2. Minggu #6	29
3. Minggu #7	34
B. TUGAS DAN PERTANYAAN	40
1. Minggu #5	40
2. Minggu #6	45
3. Minggu #7	53
BAB 4. PENUTUP	58
A. KESIMPULAN.....	58
B. SARAN	58
PUSTAKA	59

BAB 1. PENDAHULUAN

A. LATAR BELAKANG

SQL (Structured Query Language) merupakan suatu sintaks atau perintah yang digunakan untuk mengakses data dalam sebuah *database*, SQL sendiri juga bisa digunakan untuk memanipulasi sebuah data dalam database seperti untuk menambah data (*insert*), mengubah data (*update*), menampilkan data (*select*), dan juga menghapus data (*delete*). Biasanya sintaks ini disebut dengan *Data Manipulation Language* (DML) yang merupakan bagian dari SQL (Structured Query Language).

DML(Data Manipulation Language) merupakan kumpulan suatu perintah yang juga berfungsi untuk memanipulasi sebuah data dalam *database*, mulai dari menambah data, mengubah data, menampilkan data, dan menghapus data. Yang merupakan dari kumpulan perintah DML yaitu *INSERT*, *UPDATE*, *SELECT*, dan *DELETE*. Namun pada laporan untuk kali ini kita hanya akan memperdalam perintah SQL mengenai *SELECT* dengan *Sub-query*, dan *JOIN*.

B. MASALAH

1. Mahasiswa belum memahami apa itu *sub-query* dan *JOIN* pada perintah *SELECT*.
2. Mahasiswa belum mengerti bagaimana penerapan *sub-query* dan *JOIN*.
3. Mahasiswa masih belum bisa memaksimalkan penerapan *sub-query* dan *JOIN*.

C. TUJUAN

1. Mahasiswa mampu memahami apa itu *sub-query* dan *JOIN* pada perintah *SELECT*.
2. Mahasiswa mampu memahami bagaimana penerapan *sub-query* dan *JOIN*.
3. Mahasiswa mampu memaksimalkan penerapan *sub-query* dan *JOIN*.

BAB 2. LANDASAN TEORI

A. TRANSACT-SQL SELECT

➤ **SELECT**

Pernyataan **SELECT** digunakan untuk memilih data dari database. Data yang dikembalikan disimpan dalam tabel hasil, yang disebut kumpulan hasil.

➤ **SELECT Syntax**

```
SELECT column1, column2, ...
FROM table_name;
```

Di sini, kolom1, kolom2, ... adalah nama bidang tabel yang ingin Anda pilih datanya. Jika Anda ingin memilih semua bidang yang tersedia dalam tabel, gunakan sintaks berikut:

```
SELECT * FROM table_name;
```

1. **DISTINCT**

Pernyataan **SELECT DISTINCT** digunakan untuk mengembalikan nilai yang *distinct* (berbeda). Kata kunci **DISTINCT** menghilangkan *record* yang duplikat dari hasil.

➤ **SELECT DISTINCT Syntax**

```
SELECT DISTINCT column1, column2, ...
FROM table_name;
```

2. **WHERE**

WHERE digunakan untuk memfilter record. Ini digunakan untuk mengekstrak hanya catatan yang memenuhi kondisi tertentu. **WHERE** tidak hanya digunakan dalam pernyataan **SELECT**, tetapi juga digunakan dalam **UPDATE**, **DELETE**, dll.

➤ **WHERE Syntax**

```
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

3. AND, OR, NOT

WHERE dapat digabungkan dengan operator AND, OR, dan NOT. Operator AND dan OR digunakan untuk memfilter record berdasarkan lebih dari satu kondisi:

- ❖ Operator AND menampilkan record jika semua kondisi yang dipisahkan oleh AND adalah TRUE.
- ❖ Operator OR menampilkan record jika salah satu kondisi yang dipisahkan oleh OR adalah TRUE.
- ❖ Operator NOT menampilkan record jika kondisi NOT TRUE.

➤ AND Syntax

```
SELECT column1, column2, ...
FROM table_name
WHERE condition1 AND condition2 AND condition3 ...;
```

➤ OR Syntax

```
SELECT column1, column2, ...
FROM table_name
WHERE condition1 OR condition2 OR condition3 ...;
```

➤ NOT Syntax

```
SELECT column1, column2, ...
FROM table_name
WHERE NOT condition;
```

4. ORDER BY

Kata kunci *ORDER BY* digunakan untuk mengurutkan hasil dalam urutan naik atau turun. Kata kunci *ORDER BY* mengurutkan rekaman dalam urutan naik secara *default*. Untuk mengatur urutan hasil dapat menggunakan kata kunci *ASC/DESC*.

➤ ORDER BY Syntax

```
SELECT column1, column2, ...
FROM table_name
ORDER BY column1, column2, ... ASC|DESC;
```

5. GROUP BY

GROUP BY mengelompokkan baris yang memiliki nilai yang sama ke dalam baris ringkasan. Pernyataan *GROUP BY* sering digunakan dengan fungsi agregat (*COUNT()*, *MAX()*, *MIN()*, *SUM()*, *AVG()*) untuk mengelompokkan hasil yang ditetapkan oleh satu atau lebih kolom.

➤ GROUP BY Syntax

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
ORDER BY column_name(s);
```

6. HAVING

HAVING ditambahkan ke SQL karena kata kunci *WHERE* tidak dapat digunakan dengan fungsi agregat.

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
HAVING condition
ORDER BY column_name(s);
```

7. LIKE

Operator *LIKE* digunakan dalam klausa *WHERE* untuk mencari pola tertentu dalam kolom. Ada dua karakter pengganti yang sering digunakan terkait dengan operator *LIKE*: Tanda persen (%) mewakili nol, satu, atau beberapa karakter dan Tanda garis bawah (_) mewakili satu karakter tunggal. *LIKE* juga dapat digabungkan dengan beberapa kondisi menggunakan operator *AND* atau *OR*.

➤ Operator Like

```
SELECT column1, column2, ...
FROM table_name
WHERE columnN LIKE pattern;
```

8. WILDCARD

wildcard digunakan untuk menggantikan satu atau lebih karakter dalam sebuah string. Karakter wildcard digunakan dengan operator *LIKE*. Operator *LIKE* digunakan dalam klausa *WHERE* untuk mencari pola tertentu dalam kolom.

Slb.	Deskripsi	Contoh Penggunaan
%	Mewakili nol atau lebih karakter.	bl% menemukan bl, black, blue, dan blob.
_	Merupakan satu karakter.	h_t menemukan hot, hat, dan hit.
[]	Mewakili karakter tunggal apa pun dalam tanda kurung.	h[oa]t menemukan hot dan hat, tetapi tidak hit.
^	Mewakili karakter apa pun yang tidak ada di dalam tanda kurung.	h[^oa]t menemukan hit, tetapi tidak hot dan hat.
-	Mewakili berbagai karakter.	c[a-b]t menemukan cat dan cbt.

```
SELECT * FROM Customers
WHERE City LIKE '_ondon';
```

B. TRANSACT-SQL SELECT DENGAN JOIN

1. JOIN

JOIN digunakan untuk menggabungkan baris dari dua atau lebih tabel, berdasarkan kolom terkait di antara mereka. *INNER JOIN* mengembalikan rekaman yang memiliki nilai yang cocok di kedua tabel

➤ INNER JOIN Syntax

```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;
```

2. ***RIGHT JOIN***

Kata kunci *RIGHT JOIN* mengembalikan semua *record* dari tabel kanan (table2), dan *record* yang cocok dari tabel kiri (table1). Hasilnya adalah 0 *record* dari sisi kiri, jika tidak ada yang cocok.

➤ **RIGHT JOIN Syntax**

```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name = table2.column_name;
```

3. ***LEFT JOIN***

Kata kunci *LEFT JOIN* mengembalikan semua *record* dari tabel kiri (table1), dan *record* yang cocok dari tabel kanan (table2). Hasilnya adalah 0 *record* dari sisi kanan, jika tidak ada yang cocok.

➤ **LEFT JOIN Syntax**

```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name = table2.column_name;
```

4. ***FULL JOIN***

Kata kunci *FULL OUTER JOIN* mengembalikan semua *record* ketika ada kecocokan dalam *record* tabel kiri (table1) atau kanan (table2).

➤ **FULL JOIN Syntax**

```
SELECT column_name(s)
FROM table1
FULL OUTER JOIN table2
ON table1.column_name = table2.column_name
WHERE condition;
```

5. UNION

Operator *UNION* digunakan untuk menggabungkan kumpulan hasil dari dua atau lebih pernyataan *SELECT*. Setiap pernyataan *SELECT* dalam *UNION* harus memiliki jumlah kolom yang sama, Kolom juga harus memiliki tipe data yang serupa, Kolom di setiap pernyataan *SELECT* juga harus dalam urutan yang sama.

➤ UNION Syntax

```
SELECT column_name(s) FROM table1
UNION
SELECT column_name(s) FROM table2;
```

```
SELECT column_name(s) FROM table1
UNION ALL
SELECT column_name(s) FROM table2;
```

6. SELECT TOP

SELECT TOP digunakan untuk menentukan jumlah *record* yang akan dikembalikan. *SELECT TOP* berguna pada tabel besar dengan ribuan *record*. Mengembalikan sejumlah besar rekaman dapat memengaruhi kinerja.

➤ SELECT TOP Syntax

```
SELECT TOP number|percent column_name(s)
FROM table_name
WHERE condition;
```

7. IN

Operator *IN* memungkinkan Anda untuk menentukan beberapa nilai dalam klausa *WHERE*. Operator *IN* adalah singkatan dari beberapa kondisi *OR*.

➤ IN Syntax

```
SELECT column_name(s)
FROM table_name
WHERE column_name IN (value1, value2, ...);
```

OR Syntax

```
SELECT column_name(s)
FROM table_name
WHERE column_name IN (SELECT STATEMENT);
```

8. BETWEEN

Operator *BETWEEN* memilih nilai dalam rentang tertentu. Nilainya bisa berupa angka, teks, atau tanggal. Operator *BETWEEN* bersifat inklusif: nilai awal dan akhir disertakan.

➤ BETWEEN Syntax

```
SELECT column_name(s)
FROM table_name
WHERE column_name BETWEEN value1 AND value2;
```

9. ISNULL

Fungsi *ISNULL()* mengembalikan nilai yang ditentukan jika ekspresinya adalah *NULL*. Jika ekspresi *NOT NULL*, fungsi ini mengembalikan ekspresi tersebut.

➤ IS NULL Syntax

```
SELECT column_names
FROM table_name
WHERE column_name IS NULL;
```

➤ IS NOT NULL Syntax

```
SELECT column_names
FROM table_name
WHERE column_name IS NOT NULL;
```

C. TRANSACT-SQL SELECT DENGAN FUNGSI BUILT-IN

1. AVG()

Fungsi *AVG()* mengembalikan nilai rata-rata kolom numerik.

➤ AVG() Syntax

```
SELECT AVG(column_name)
FROM table_name
WHERE condition;
```

2. *COUNT()*

Fungsi *COUNT()* mengembalikan jumlah rekaman yang dikembalikan oleh *SELECT*.

➤ *COUNT() Syntax*

```
SELECT COUNT(column_name)
FROM table_name
WHERE condition;
```

3. *MAX()*

Fungsi *MAX()* mengembalikan nilai maksimum dalam satu set nilai.

➤ *MAX() Syntax*

```
SELECT MAX(column_name)
FROM table_name
WHERE condition;
```

4. *MIN()*

Fungsi *MIN()* mengembalikan nilai minimum dalam satu set nilai.

➤ *MIN() Syntax*

```
SELECT MIN(column_name)
FROM table_name
WHERE condition;
```

5. *SUM()*

Fungsi *SUM()* menghitung jumlah sekumpulan nilai.

➤ *SUM() Syntax*

```
SELECT SUM(column_name)
FROM table_name
WHERE condition;
```

6. *MID()*

Fungsi *MID()* mengekstrak substring dari string (dimulai dari posisi apa pun).
Catatan: Fungsi *MID()* dan *SUBSTR()* sama dengan fungsi *SUBSTRING()*.

```
SELECT MID("SQL Tutorial", 5, 3) AS ExtractString;
```

7. *LEN()*

Fungsi *LEN()* mengembalikan panjang *string*.

```
SELECT LEN(' Haldian.com ');
```

8. *ROUND()*

Fungsi *ROUND()* membulatkan angka ke sejumlah tempat desimal tertentu.

```
SELECT ROUND(235.415, 2) AS RoundValue;
```

9. *NOW()*

NOW() mengembalikan tanggal dan waktu saat ini. Catatan: Tanggal dan waktu ditampilkan sebagai "YYYY-MM-DD HH-MM-SS" (string) atau sebagai YYYYMMDDHHMMSS.ududu (numerik).

```
SELECT NOW();
```

10. *GETDATE()*

Fungsi *GETDATE()* mengembalikan tanggal dan waktu sistem *database* saat ini, dalam format 'YYYY-MM-DD hh:mm:ss.mmm'.

```
SELECT GETDATE();
```

11. *UPPER()*

Fungsi *UPPER()* mengubah *string* menjadi huruf besar.

```
SELECT UPPER('SQL Tutorial is FUN!');
```

12. *LOWER()*

Fungsi *LOWER()* mengubah *string* menjadi huruf kecil.

```
SELECT LOWER('SQL Tutorial is FUN!');
```

BAB 3. HASIL DAN PEMBAHASAN

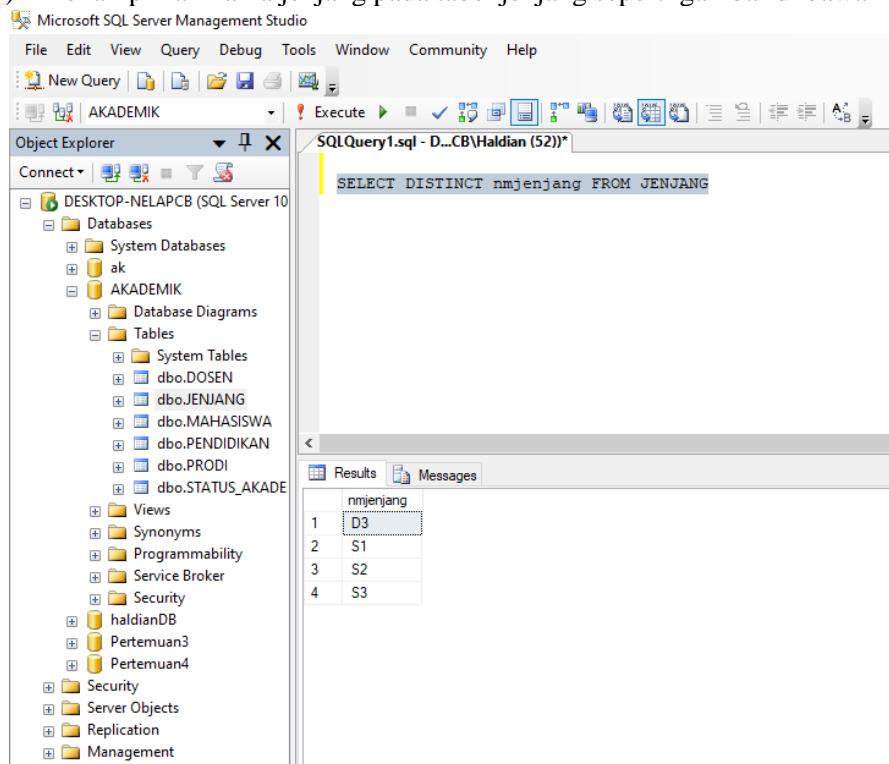
A. TAHAPAN PRAKTIKUM

1. Minggu #5

1. Persiapkan alat dan bahan
2. Menjalankan aplikasi: *Microsoft SQL Server Management Studio* (MSQLS-MS)
3. Gunakanlah perintah *transact-SQL Query (SELECT)* untuk menyajikan data dari tabel-tabel yang ada pada *database* “AKADEMIK”, dengan memanfaatkan klausa pendukung berikut: *DISTINCT, WHERE, AND, OR, ORDER BY, GROUP BY, HAVING, LIKE, dan WILDCARD*. (catatan: masing-masing klausa pendukung 3 contoh perintah, dengan data yang berbeda)

A. DISTINCT

- 1) Menampilkan nama jenjang pada tabel jenjang seperti gambar di bawah ini



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'AKADEMIK' containing tables like DOSEN, JENJANG, MAHASISWA, PENDIDIKAN, PRODI, and STATUS_AKADEMIK. The central pane displays a query window with the following SQL code:

```
SELECT DISTINCT nmjenjang FROM JENJANG
```

The Results pane shows the output of the query:

nmjenjang
1 D3
2 S1
3 S2
4 S3

- 2) Menampilkan namaprodi pada table prodi seperti gambar di bawah ini.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'AKADEMIK' containing tables like DOSEN, JENJANG, MAHASISWA, PENDIDIKAN, PRODI, and STATUS_AKADEMIK. The SQL Query window on the right contains the following query:

```
SELECT DISTINCT nmprodi FROM PRODI
```

The Results pane displays the output of the query:

nmprodi
1 Agribisnis
2 Agribisnis Pangan
3 Akuntansi
4 Akuntansi Perpajakan
5 Manajemen Informatika

- 3) Menampilkan nama mahasiswa pada table mahasiswa seperti gambar di bawah ini

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'AKADEMIK' containing tables like DOSEN, JENJANG, MAHASISWA, PENDIDIKAN, PRODI, and STATUS_AKADEMIK. The SQL Query window on the right contains the following query:

```
SELECT DISTINCT nama FROM MAHASISWA
```

The Results pane displays the output of the query:

nama
1 Agus
2 Budi
3 Edi
4 Hardi
5 Maya
6 Rendi
7 Ria
8 Riana
9 Rudi
10 Santi

B. WHERE

- 1) Menampilkan mahasiswa dengan idmhs=1 seperti gambar di bawah ini

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'AKADEMIK' containing tables like DOSEN, JENJANG, MAHASISWA, PENDIDIKAN, PRODI, and STATUS_AKADEMIK. The SQL Query window on the right contains the following code:

```
SELECT * FROM MAHASISWA where idmhs=1
```

The Results pane displays the following data:

	idmhs	npm	nama	tempat_lhr	tanggal_lhr	sex	thn_masuk	idprodi	idstatusaka
1	1	17750001	Ria	Metro	2000-10-10	P	2017	6	1

- 2) Menampilkan dosen dengan iddosen=1 seperti gambar di bawah ini

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'AKADEMIK' containing tables like DOSEN, JENJANG, MAHASISWA, PENDIDIKAN, PRODI, and STATUS_AKADEMIK. The SQL Query window on the right contains the following code:

```
SELECT * FROM dosen where iddosen=1
```

The Results pane displays the following data:

	iddosen	nmdosen	gelar	tempat_lhr	tanggal_lhr	sex	idprodi	idpendidikan
1	1	Subarjo	S.Kom	Metro	1978-10-10	L	6	1

- 3) Menampilkan pendidikan dengan idpendidikan=1 seperti gambar di bawah ini

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure, including the AKADEMIK database with tables like DOSEN, JENJANG, MAHASISWA, PENDIDIKAN, PRODI, and STATUS_AKADEMIK. The SQL Query window on the right contains the following query:

```
SELECT * FROM PENDIDIKAN where idpendidikan=1
```

The results pane shows one row of data:

	idpendidikan	nmpendidikan
1	1	S1

C. AND

- 1) Menampilkan mahasiswa dari idstatusaka=1 dan menampilkan laki-laki seperti gambar di bawah ini

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure, including the AKADEMIK database with tables like DOSEN, JENJANG, MAHASISWA, PENDIDIKAN, PRODI, and STATUS_AKADEMIK. The SQL Query window on the right contains the following query:

```
SELECT * FROM MAHASISWA where idstatusaka=1 and sex='L'
```

The results pane shows six rows of data:

	idmhs	npm	nama	tempat_lhr	tanggal_lhr	sex	thn_masuk	idprodi	idstatusaka
1	2	17760001	Agus	Kalianda	2000-10-16	L	2017	7	1
2	5	17790001	Budi	Bengkulu	2001-02-17	L	2017	10	1
3	6	17750002	Edi	Kotabumi	2001-03-11	L	2017	6	1
4	8	17770002	Hardi	Natar	2001-03-16	L	2017	8	1
5	9	17780002	Rendi	Metro	2000-01-03	L	2017	9	1
6	10	17790002	Rudi	Mesuji	2000-05-04	L	2017	10	1

- 2) Menampilkan dosen dari idpendidikan = 2 dan lahir di bandung seperti gambar di bawah ini

Microsoft SQL Server Management Studio

File Edit View Query Debug Tools Window Community Help

New Query | Execute | Object Explorer | SQLQuery1.sql - D...CB\Haldian (52)*

Object Explorer

DESKTOP-NELAPCB (SQL Server 10)

- Databases
 - System Databases
 - ak
 - AKADEMIK
 - Database Diagrams
 - Tables
 - System Tables
 - dbo.DOSEN
 - dbo.JENJANG
 - dbo.MAHASISWA
 - dbo.PENDIDIKAN
 - dbo.PRODI
 - dbo.STATUS_AKADEMIK
 - Views
 - Synonyms
 - Programmability
 - Service Broker
 - Security
 - haldianDB
 - Pertemuan3
 - Pertemuan4
 - Security
 - Server Objects
 - Replication
 - Management

SQLQuery1.sql - D...CB\Haldian (52)*

```
select * from DOSEN where idpendidikan=2 and tempat_lhr='Bandung'
```

Results

	iddosen	nmndosen	gelar	tempat_lhr	tanggal_lhr	sex	idprodi	idpendidikan
1	3	Heni	S.E., M.M.	Bandung	1980-02-19	P	8	2

- 3) Menampilkan data prodi dari idjenjang=1 dan akreditasi B seperti gambar di bawah ini

Microsoft SQL Server Management Studio

File Edit View Query Debug Tools Window Community Help

New Query | Execute | Object Explorer | SQLQuery16.sql - ...PCB\Haldian (53) | SQLQuery1.sql - D...CB\Haldian (52)*

Object Explorer

DESKTOP-NELAPCB (SQL Server 10)

- Databases
 - System Databases
 - ak
 - AKADEMIK
 - Database Diagrams
 - Tables
 - System Tables
 - dbo.DOSEN
 - dbo.JENJANG
 - dbo.MAHASISWA
 - dbo.PENDIDIKAN
 - dbo.PRODI
 - dbo.STATUS_AKADEMIK
 - Views
 - Synonyms
 - Programmability
 - Service Broker
 - Security
 - haldianDB
 - Pertemuan3
 - Pertemuan4
 - Security
 - Server Objects
 - Replication
 - Management

SQLQuery16.sql - ...PCB\Haldian (53) | SQLQuery1.sql - D...CB\Haldian (52)*

```
select * from PRODI where idjenjang=1 and akreditasi='B'
```

Results

	idprod	nmprodi	akreditasi	Thn_berdiri	idjenjang
1	6	Manajemen Informatika	B	2006	1
2	9	Agnbisnis	B	2002	1

D. OR

- 1) Menampilkan data mahasiswa idprodi=6 or tempat_lhr='Bandung' seperti gambar di bawah ini

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure for 'DESKTOP-NELAPCB'. The 'Tables' node under 'AKADEMIK' contains 'MAHASISWA'. The 'Results' tab on the right displays the output of the following query:

```
select * from MAHASISWA where idprodi=6 or tempat_lhr='Bandung'
```

The results table shows three rows of student data:

idmhs	npm	nama	tempat_lhr	tanggal_lhr	sex	thn_masuk	idprodi	idstatusaka
1	17750001	Ria	Metro	2000-10-10	P	2017	6	1
2	17780001	Riana	Bandung	2002-09-22	P	2017	9	1
3	17750002	Edi	Kotabumi	2001-03-11	L	2017	6	1

- 2) Menampilkan prodi idprod=6 or idjenjang=1 seperti gambar di bawah ini

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure for 'DESKTOP-NELAPCB'. The 'Tables' node under 'AKADEMIK' contains 'PRODI'. The 'Results' tab on the right displays the output of the following query:

```
select * from PRODI where idprod=6 or idjenjang=1
```

The results table shows three rows of program data:

idprod	nmprodi	akreditasi	Thn_berdiri	idjenjang
6	Manajemen Informatika	B	2006	1
7	Akuntansi	A	2004	1
9	Agribisnis	B	2002	1

- 3) Menampilkan data dosen dari idpendidikan=1 or tempat_lhr='jakarta' seperti gambar di bawah ini

```
Microsoft SQL Server Management Studio
File Edit View Query Debug Tools Window Community Help
New Query Execute
Object Explorer
Connect DESKTOP-NELAPCB (SQL Server 10
Databases System Databases ak
AKADEMIK Database Diagrams
Tables System Tables dbo.DOSEN
dbo.JENJANG dbo.MAHASISWA
dbo.PENDIDIKAN dbo.PRODI
dbo.STATUS_AKADE Views
Synonyms Programmability Service Broker
Security haldianDB
Pertemuan3 Pertemuan4
Security Server Objects Replication Management
SQLQuery19.sql - ...PCB\Haldian (52) SQLQuery1.sql - D...CB\Haldian (53)*
select * from DOSEN where idpendidikan=1 or tempat_lhr='Jakarta'
Results Messages
iddosen nmdosen gelar tempat_lhr tanggal_lhr sex idprodi idpendidikan
1 Subarjo S.Kom Metro 1978-10-10 L 6 1
2 Wendi S.E., M.M. Jakarta 1977-10-11 L 7 2
3 Suwondo S.P. Jakarta 1980-09-22 L 9 1
4 Hendro S.P., M.P. Jakarta 2981-01-17 L 10 2
```

E. ORDER BY

- 1) Menampilkan data dosen mengurutkan nama dosen hasil dalam urutan asc dari atas ke bawah seperti gambar di bawah ini

```
Microsoft SQL Server Management Studio
File Edit View Query Debug Tools Window Community Help
New Query Execute
Object Explorer
Connect DESKTOP-NELAPCB (SQL Server 10
Databases System Databases ak
AKADEMIK Database Diagrams
Tables System Tables dbo.DOSEN
dbo.JENJANG dbo.MAHASISWA
dbo.PENDIDIKAN dbo.PRODI
dbo.STATUS_AKADE Views
Synonyms Programmability Service Broker
Security haldianDB
Pertemuan3 Pertemuan4
Security Server Objects Replication Management
SQLQuery1.sql - D...CB\Haldian (53)* SQLQuery19.sql - ...PCB\Haldian (52)
select * from DOSEN ORDER BY nmdosen asc
Results Messages
iddosen nmdosen gelar tempat_lhr tanggal_lhr sex idprodi idpendidikan
9 Hendro S.P., M.P. Jakarta 2981-01-17 L 10 2
3 Heni S.E., M.M. Bandung 1980-02-19 P 8 2
1 Subarjo S.Kom Metro 1978-10-10 L 6 1
8 Suwondo S.P. Jakarta 1980-09-22 L 9 1
2 Wendi S.E., M.M. Jakarta 1977-10-11 L 7 2
```

- 2) Menampilkan nmjenjang dari bawah ke atas dengan desc seperti gambar di bawah ini

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'AKADEMIK' containing tables like DOSEN, JENJANG, MAHASISWA, PENDIDIKAN, PRODI, and STATUS_AKADEMIK. The SQL Query window on the right contains the following query:

```
select * from JENJANG ORDER BY nmjenjang desc
```

The Results pane displays the following data:

	idjenjang	nmjenjang
1	4	S3
2	3	S2
3	2	S1
4	1	D3

- 3) Menampilkan prodi dari bawah ke atas dengan asc seperti gambar di bawah ini

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'AKADEMIK' containing tables like DOSEN, JENJANG, MAHASISWA, PENDIDIKAN, PRODI, and STATUS_AKADEMIK. The SQL Query window on the right contains the following query:

```
select * from PRODI ORDER BY nmprodi asc
```

The Results pane displays the following data:

	idprod	nmprodi	akreditasi	Thn_berdiri	idjenjang
1	9	Agribisnis	B	2002	1
2	10	Agribisnis Pangan	-	2018	2
3	7	Akuntansi	A	2004	1
4	8	Akuntansi Perpajakan	-	2018	2
5	6	Manajemen Informatika	B	2006	1

F. GROUP BY

- 1) Menampilkan nmprodi dengan group by seperti pada gambar di bawah ini

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure for 'DESKTOP-NELAPCB (SQL Server 10)'. The 'Tables' node under 'AKADEMIK' contains several tables: PRODI, MAHASISWA, PENDIDIKAN, DOSEN, JENJANG, and STATUS_AKADEMIK. The 'Results' tab in the bottom pane displays the output of the following query:

```
select Distinct nmprodi,COUNT(nmprodi) as Jumlah  
from PRODI  
group by nmprodi
```

The results are:

nmprodi	Jumlah
Agribisnis	1
Agribisnis Pangan	1
Akuntansi	1
Akuntansi Perpajakan	1
Manajemen Informatika	1

- 2) Menampilkan nmjenjang pada table jenjang dengan group by seperti gambar di bawah ini

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure for 'DESKTOP-NELAPCB (SQL Server 10)'. The 'Tables' node under 'AKADEMIK' contains several tables: PRODI, MAHASISWA, PENDIDIKAN, DOSEN, JENJANG, and STATUS_AKADEMIK. The 'Results' tab in the bottom pane displays the output of the following query:

```
select Distinct nmjenjang,COUNT(nmjenjang) as Jumlah  
from JENJANG  
group by nmjenjang
```

The results are:

nmjenjang	Jumlah
D3	1
S1	1
S2	1
S3	1

- 3) Menampilkan nama dosen dan gelar dosen pada table dosen dengan group by seperti gambar di bawah ini

```

Microsoft SQL Server Management Studio
File Edit View Query Debug Tools Window Community Help
New Query | Execute
AKADEMIK SQLQuery25.sql - ...PCB\Haldian (52)) SQLQuery1.sql - D...CB\Haldian (53))*
Object Explorer
Connect ▾
DESKTOP-NELAPCB (SQL Server 10
Databases
System Databases
ak
AKADEMIK
Database Diagrams
Tables
System Tables
dbo.DOSEN
dbo.JENJANG
dbo.MAHASISWA
dbo.PENDIDIKAN
dbo.PRODI
dbo.STATUS_AKADE
Views
Synonyms
Programmability
Service Broker
Security
haldianDB
Pertemuan3
Pertemuan4
Security
Server Objects
Replication
Management
SQLQuery25.sql - ...PCB\Haldian (52))
select Distinct nmdosen, gelar,COUNT(gelar) as Jumlah
from DOSEN
group by nmdosen, gelar
Results Messages
nmdosen gelar Jumlah
1 Heni S.E., M.M., 1
2 Wendi S.E., M.M., 1
3 Subarjo S.Kom 1
4 Suwondo S.P. 1
5 Hendro S.P., M.P 1

```

G. HAVING

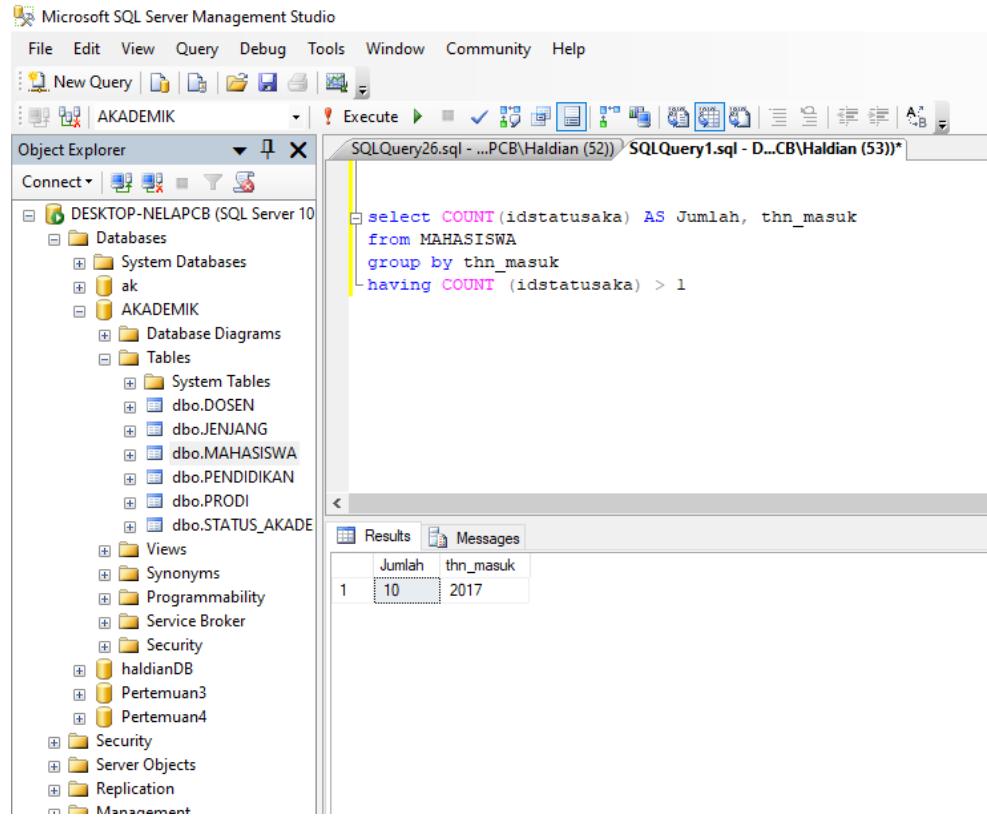
- 1) Menampilkan jumlah dosen menggunakan having seperti gambar di bawah ini

```

Microsoft SQL Server Management Studio
File Edit View Query Debug Tools Window Community Help
New Query | Execute
AKADEMIK SQLQuery1.sql - D...CB\Haldian (53))*
Object Explorer
Connect ▾
DESKTOP-NELAPCB (SQL Server 10
Databases
System Databases
ak
AKADEMIK
Database Diagrams
Tables
System Tables
dbo.DOSEN
dbo.JENJANG
dbo.MAHASISWA
dbo.PENDIDIKAN
dbo.PRODI
dbo.STATUS_AKADE
Views
Synonyms
Programmability
Service Broker
Security
haldianDB
Pertemuan3
Pertemuan4
Security
Server Objects
Replication
Management
SQLQuery1.sql - D...CB\Haldian (53))*
select COUNT(idpendidikan) AS Jumlah, gelar
from DOSEN
group by gelar
having COUNT (idpendidikan) > 1
Results Messages
Jumlah gelar
1 2 S.E., M.M.,

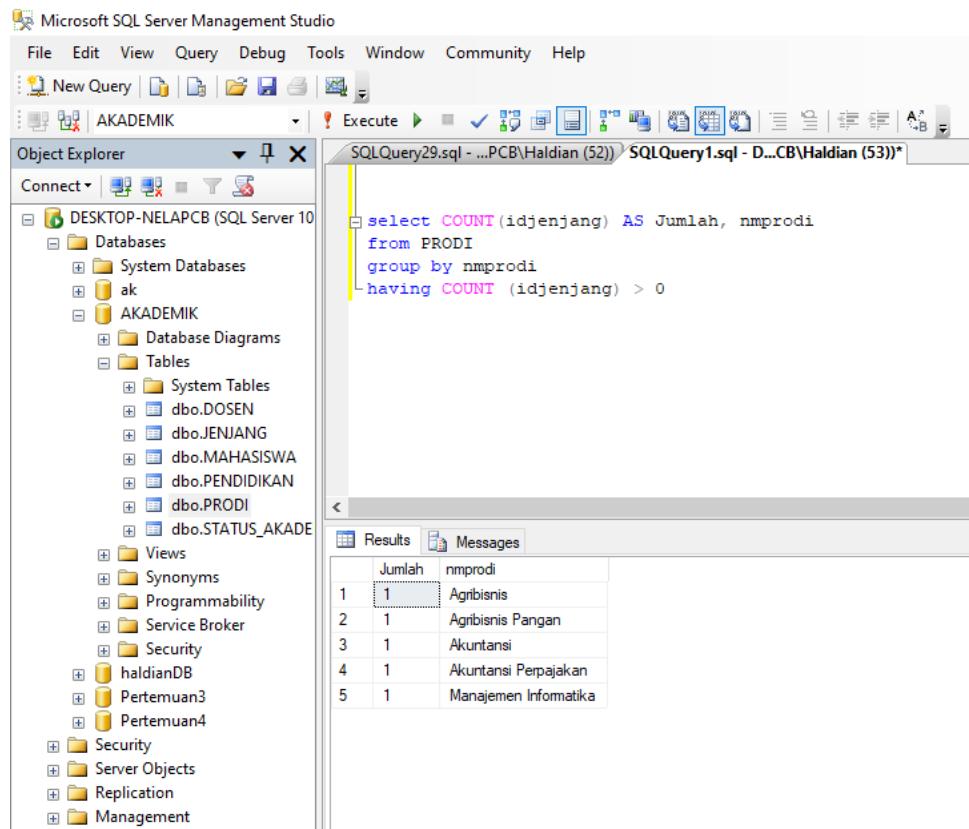
```

- 2) Menampilkan table mahasiswa dengan having count idstatusaka > 1 seperti gambar di bawah ini



```
Microsoft SQL Server Management Studio
File Edit View Query Debug Tools Window Community Help
New Query | Object Explorer | SQLQuery26.sql - ...PCB\Haldian (52) | SQLQuery1.sql - D...CB\Haldian (53)*
Object Explorer
Connect ▾
DESKTOP-NELAPCB (SQL Server 10)
Databases
System Databases
ak
AKADEMIK
Database Diagrams
Tables
System Tables
dbo.DOSEN
dbo.JENJANG
dbo.MAHASISWA
dbo.PENDIDIKAN
dbo.PRODI
dbo.STATUS_AKADEMIK
Views
Synonyms
Programmability
Service Broker
Security
haldianDB
Pertemuan3
Pertemuan4
Security
Server Objects
Replication
Management
SQLQuery26.sql - ...PCB\Haldian (52) | SQLQuery1.sql - D...CB\Haldian (53)*
select COUNT(idstatusaka) AS Jumlah, thn_masuk
from MAHASISWA
group by thn_masuk
having COUNT (idstatusaka) > 1
Results Messages
Jumlah thn_masuk
1 10 2017
```

- 3) Menampilkan nmprodi yang lebih dari 0 dengan having count idjenjang seperti gambar di bawah ini



```
Microsoft SQL Server Management Studio
File Edit View Query Debug Tools Window Community Help
New Query | Object Explorer | SQLQuery29.sql - ...PCB\Haldian (52) | SQLQuery1.sql - D...CB\Haldian (53)*
Object Explorer
Connect ▾
DESKTOP-NELAPCB (SQL Server 10)
Databases
System Databases
ak
AKADEMIK
Database Diagrams
Tables
System Tables
dbo.DOSEN
dbo.JENJANG
dbo.MAHASISWA
dbo.PENDIDIKAN
dbo.PRODI
dbo.STATUS_AKADEMIK
Views
Synonyms
Programmability
Service Broker
Security
haldianDB
Pertemuan3
Pertemuan4
Security
Server Objects
Replication
Management
SQLQuery29.sql - ...PCB\Haldian (52) | SQLQuery1.sql - D...CB\Haldian (53)*
select COUNT(idjenjang) AS Jumlah, nmprodi
from PRODI
group by nmprodi
having COUNT (idjenjang) > 0
Results Messages
Jumlah nmprodi
1 Agribisnis
2 Agribisnis Pangan
3 Akuntansi
4 Akuntansi Perpajakan
5 Manajemen Informatika
```

H. LIKE

- 1) Menampilkan data mahasiswa yang dimulai dengan Like Ka% seperti gambar di bawah ini

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'AKADEMIK' containing several tables like DOSEN, JENJANG, MAHASISWA, PENDIDIKAN, PRODI, and STATUS_AKADEMIK. The SQL Query window on the right contains the following query:

```
select * from MAHASISWA where tempat_lhr like 'Ka%'
```

The Results pane below shows one row of data:

idmhs	npm	nama	tempat_lhr	tanggal_lhr	sex	thn_masuk	idprodi	idstatusaka	
1	2	17760001	Aagus	Kalianda	2000-10-16	L	2017	7	1

- 2) Menampilkan data dosen yang dimulai dengan like gelar s% seperti gambar di bawah ini

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'AKADEMIK' containing several tables like DOSEN, JENJANG, MAHASISWA, PENDIDIKAN, PRODI, and STATUS_AKADEMIK. The SQL Query window on the right contains the following query:

```
select * from Dosen where gelar like 'S%'
```

The Results pane below shows five rows of data:

iddosen	nmdosen	gelar	tempat_lhr	tanggal_lhr	sex	idprodi	idpendidikan
1	Subarjo	S.Kom	Metro	1978-10-10	L	6	1
2	Wendi	S.E., M.M..	Jakarta	1977-10-11	L	7	2
3	Heni	S.E., M.M..	Bandung	1980-02-19	P	8	2
4	Suwondo	S.P.	Jakarta	1980-09-22	L	9	1
5	Hendro	S.P., M.P	Jakarta	2981-01-17	L	10	2

- 3) Menampilkan data prodi yang di mulai dengan like nmprodi m% seperti gambar di bawah ini

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure, including the AKADEMIK database which contains tables like PRODI, DOSEN, JENJANG, and MAHASISWA. The Results pane on the right displays the output of the query:

```
select * from PRODI where nmprodi like 'M%'
```

	idprod	nmprodi	akreditasi	Thn_berdiri	idjenjang
1	6	Manajemen Informatika	B	2006	1

I. WILRCARD

- 1) Menampilkan data prodi yang di akhiri huruf n seperti gambar di bawah ini

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure, including the AKADEMIK database which contains tables like PRODI, DOSEN, JENJANG, and MAHASISWA. The Results pane on the right displays the output of the query:

```
select * from PRODI where nmprodi like '%n';
```

	idprod	nmprodi	akreditasi	Thn_berdiri	idjenjang
1	8	Akuntansi Perpajakan	-	2018	2
2	10	Agribisnis Pangan	-	2018	2

- 2) Menampilkan data mahasiswa yang di awali huruf a seperti gambar di bawah ini

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the database 'AKADEMIK' is selected. In the Results pane, a query is run:

```
select * from MAHASISWA where nama like 'a_%';
```

The results show one row of data:

	idmhs	npm	nama	tempat_lhr	tanggal_lhr	sex	thn_masuk	idprodi	idstatusaka
1	2	17760001	Agu	Kalianda	2000-10-16	L	2017	7	1

- 3) Menampilkan data yang memilih semua data 's','w','h' seperti gambar di bawah ini

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the database 'AKADEMIK' is selected. In the Results pane, a query is run:

```
select * from DOSEN where nm dosen like '[swh]%';
```

The results show five rows of data:

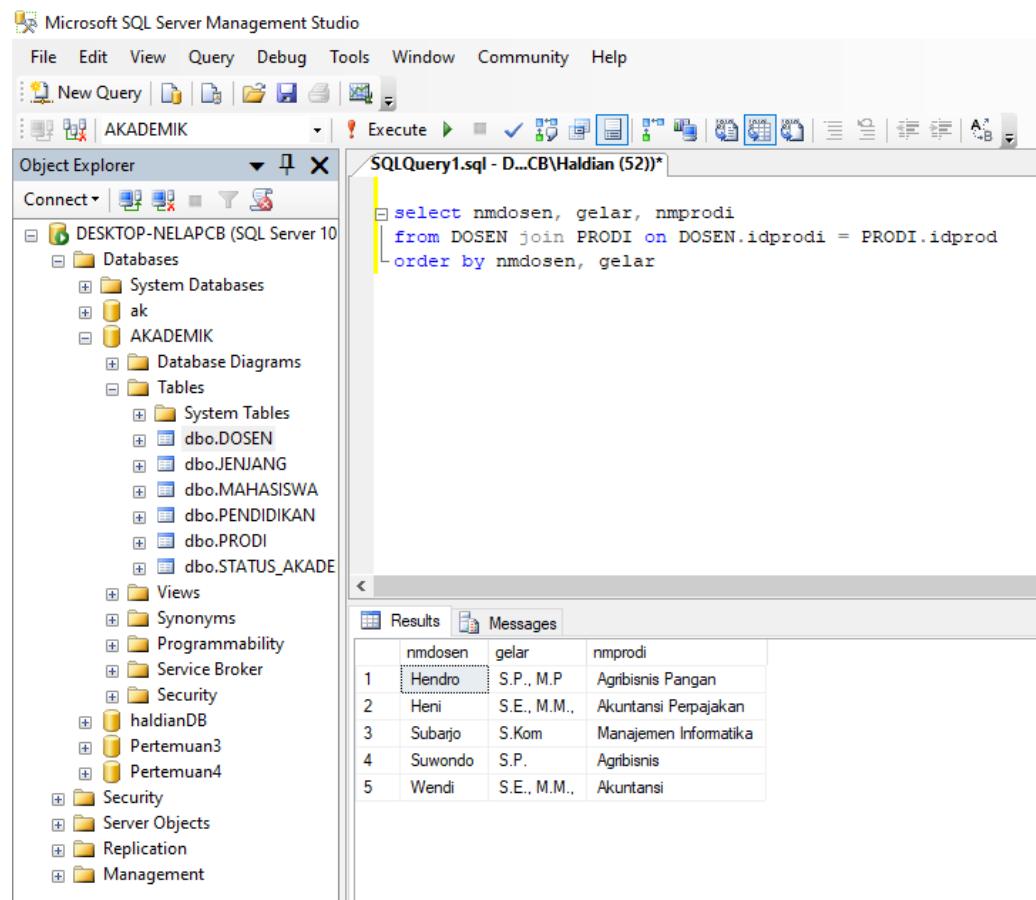
	iddosen	nm dosen	gelar	tempat_lhr	tanggal_lhr	sex	idprodi	idpendidikan
1	1	Subarjo	S.Kom	Metro	1978-10-10	L	6	1
2	2	Wendi	S.E., M.M..	Jakarta	1977-10-11	L	7	2
3	3	Heni	S.E., M.M..	Bandung	1980-02-19	P	8	2
4	8	Suwondo	S.P.	Jakarta	1980-09-22	L	9	1
5	9	Hendro	S.P., M.P	Jakarta	2981-01-17	L	10	2

2. Minggu #6

1. Persiapkan alat dan bahan
2. Menjalankan aplikasi: *Microsoft SQL Server Management Studio (MSQLS-MS)*
3. Gunakanlah perintah *transact-SQL Query (SELECT)* untuk menyajikan data dari tabel-tabel yang ada pada database “AKADEMIK”, dengan memanfaatkan klausa pendukung berikut: *JOIN, INNER JOIN, LEFT JOIN, FULL JOIN, UNION, TOP, IN, BETWEN, dan ISNULL*. (catatan: masing-masing klausa, 1 contoh perintah SQL)

A. *JOIN*

Menampilkan daftar dosen, dan gelarnya beserta nama prodi seperti gambar di bawah ini



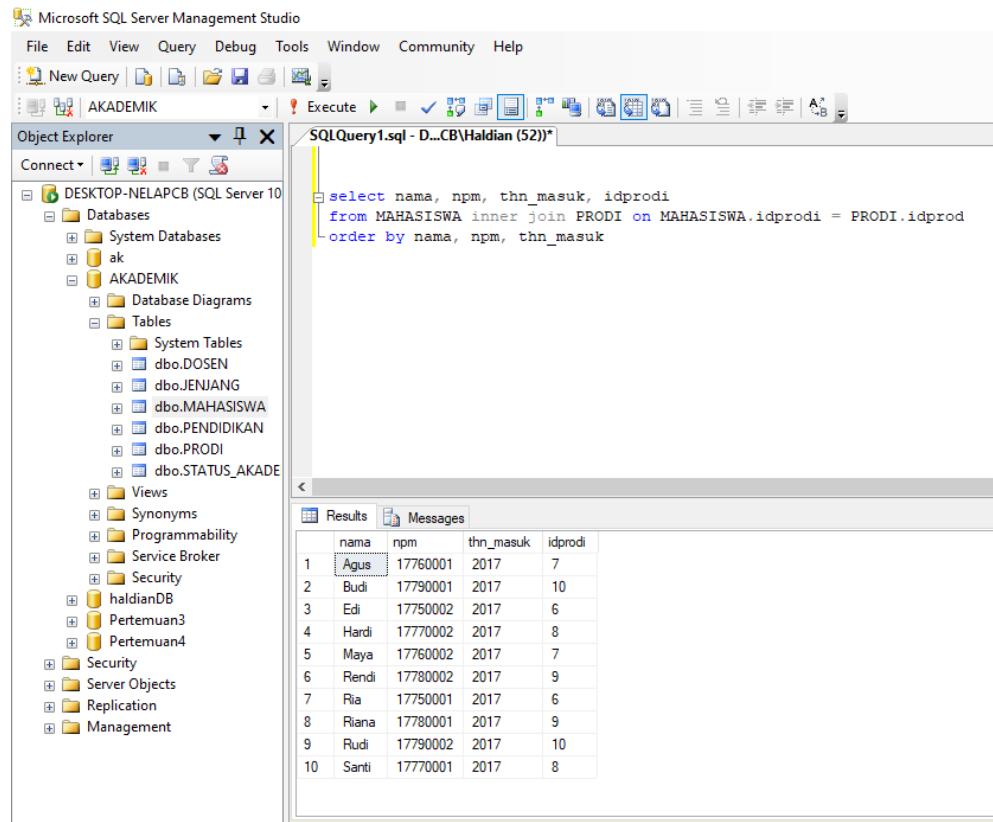
The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer pane on the left shows a tree view of databases, including 'DESKTOP-NELAPCB (SQL Server 10)' which contains 'AKADEMIK' and other system databases like 'ak' and 'haldianDB'. The 'Tables' node under 'AKADEMIK' lists several tables: DOSEN, JENJANG, MAHASISWA, PENDIDIKAN, PRODI, and STATUS_AKADEMIK. The 'Results' pane on the right displays the output of a SQL query:

```
select nmdosen, gelar, nmprodi
from DOSEN join PRODI on DOSEN.idprodi = PRODI.idprod
order by nmdosen, gelar
```

	nmdosen	gelar	nmprodi
1	Hendro	S.P., M.P	Agribisnis Pangan
2	Heni	S.E., M.M.,	Akuntansi Perpajakan
3	Subarjo	S.Kom	Manajemen Informatika
4	Suwondo	S.P.	Agribisnis
5	Wendi	S.E., M.M.,	Akuntansi

B. INNER JOIN

Menampilkan daftar mahasiswa nama, npm dan tahun masuk serta idprodinya itu sendiri seperti gambar di bawah ini



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'DESKTOP-NELAPCB' with various schemas like 'System Databases', 'ak', 'AKADEMIK', and 'haldianDB'. The 'Tables' node under 'AKADEMIK' lists tables such as 'DOSEN', 'JENJANG', 'MAHASISWA', 'PENDIDIKAN', 'PRODI', and 'STATUS_AKADEMIK'. The 'Results' tab on the right displays the output of the following query:

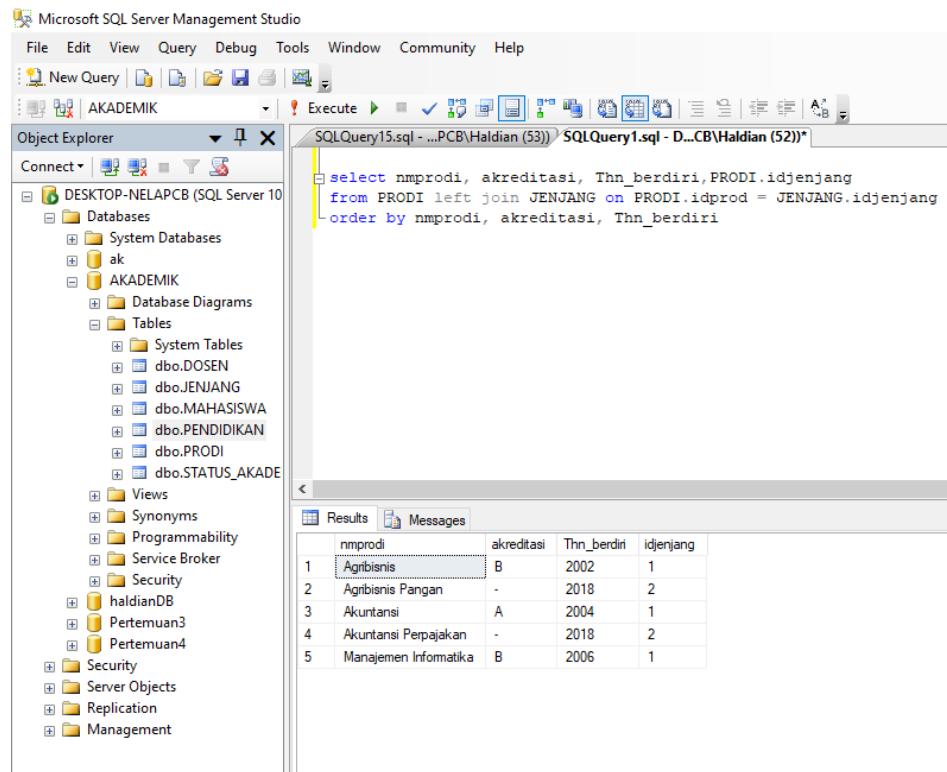
```
select nama, npm, thn_masuk, idprodi
from MAHASISWA inner join PRODI on MAHASISWA.idprodi = PRODI.idprod
order by nama, npm, thn_masuk
```

The results show 10 rows of data:

	nama	npm	thn_masuk	idprodi
1	Agus	17760001	2017	7
2	Budi	17790001	2017	10
3	Edi	17750002	2017	6
4	Hardi	17770002	2017	8
5	Maya	17760002	2017	7
6	Rendi	17780002	2017	9
7	Ria	17750001	2017	6
8	Riana	17780001	2017	9
9	Rudi	17790002	2017	10
10	Santi	17770001	2017	8

C. LEFT JOIN

Menampilkan nmprodi, akreditasi, thn_bediri dan idejenjang seperti gambar di bawah ini



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'DESKTOP-NELAPCB' with various schemas like 'System Databases', 'ak', 'AKADEMIK', and 'haldianDB'. The 'Tables' node under 'AKADEMIK' lists tables such as 'DOSEN', 'JENJANG', 'MAHASISWA', 'PENDIDIKAN', 'PRODI', and 'STATUS_AKADEMIK'. The 'Results' tab on the right displays the output of the following query:

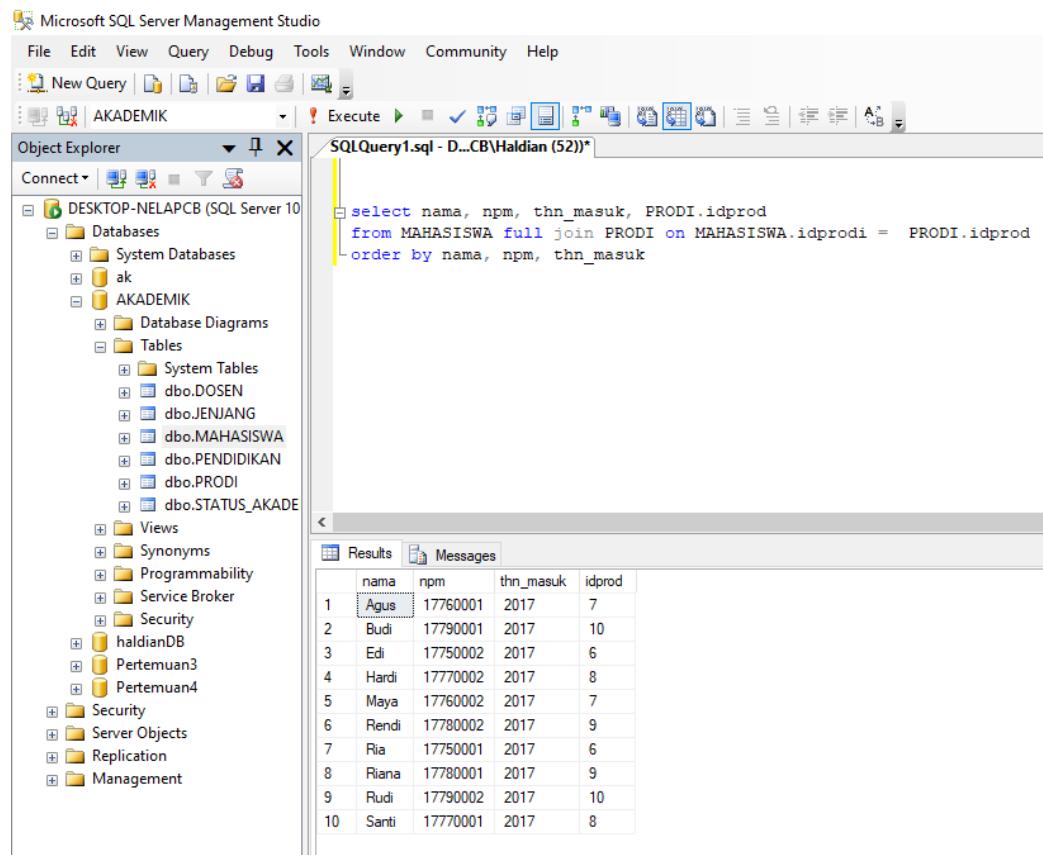
```
select nmprodi, akreditasi, Thn_berdiri, PRODI.idjenjang
from PRODI left join JENJANG on PRODI.idprod = JENJANG.idjenjang
order by nmprodi, akreditasi, Thn_berdiri
```

The results show 5 rows of data:

	nmprodi	akreditasi	Thn_berdiri	idjenjang
1	Agrabisnis	B	2002	1
2	Agrabisnis Pangan	-	2018	2
3	Akuntansi	A	2004	1
4	Akuntansi Perpajakan	-	2018	2
5	Manajemen Informatika	B	2006	1

D. FULL JOIN

Menampilkan daftar mahasiswa nama, npm tahun masuk dan idprodi seperti gambar di bawah ini



The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the database 'AKADEMIK' is selected. In the center pane, a query window titled 'SQLQuery1.sql - D...CB\Haldian (52)*' contains the following T-SQL code:

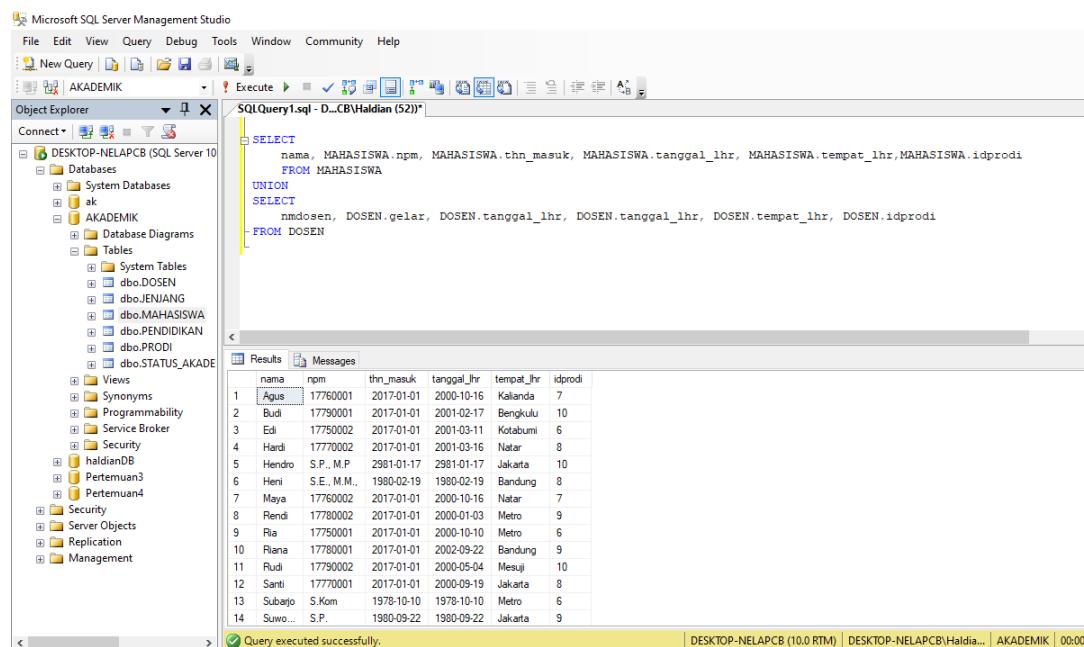
```
select nama, npm, thn_masuk, PRODI.idprod
from MAHASISWA full join PRODI on MAHASISWA.idprodi = PRODI.idprod
order by nama, npm, thn_masuk
```

The results pane displays a table with 10 rows of data:

	nama	npm	thn_masuk	idprod
1	Agus	17760001	2017	7
2	Budi	17790001	2017	10
3	Edi	17750002	2017	6
4	Hardi	17770002	2017	8
5	Maya	17760002	2017	7
6	Rendi	17780002	2017	9
7	Ria	17750001	2017	6
8	Riana	17780001	2017	9
9	Rudi	17790002	2017	10
10	Santi	17770001	2017	8

E. UNION

Menampilkan aktivitas akademik dosen dan mahasiswa seperti gambar di bawah ini



The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the database 'AKADEMIK' is selected. In the center pane, a query window titled 'SQLQuery1.sql - D...CB\Haldian (52)*' contains the following T-SQL code:

```
SELECT
    nama, MAHASISWA.npm, MAHASISWA.thn_masuk, MAHASISWA.tanggal_lhr, MAHASISWA.tempat_lhr, MAHASISWA.idprodi
    FROM MAHASISWA
UNION
SELECT
    nmdoSEN, DOSEN.gelar, DOSEN.tanggal_lhr, DOSEN.tanggal_lhr, DOSEN.tempat_lhr, DOSEN.idprodi
    FROM DOSEN
```

The results pane displays a table with 14 rows of data:

	nama	npm	thn_masuk	tanggal_lhr	tempat_lhr	idprodi
1	Agus	17760001	2017-01-01	2000-10-16	Kallanda	7
2	Budi	17790001	2017-01-01	2001-02-17	Bengkulu	10
3	Edi	17750002	2017-01-01	2001-03-11	Kotabumi	6
4	Hardi	17770002	2017-01-01	2001-03-16	Natar	8
5	Hendro	S.P. M.P.	2981-01-17	Jakarta	10	
6	Heni	S.E. M.M.	1980-02-19	1980-02-19	Bandung	8
7	Maya	17760002	2017-01-01	2000-10-16	Natar	7
8	Rendi	17780002	2017-01-01	2000-01-03	Metro	9
9	Ria	17750001	2017-01-01	2000-10-10	Metro	6
10	Riana	17780001	2017-01-01	2002-09-22	Bandung	9
11	Rudi	17790002	2017-01-01	2000-05-04	Mesuji	10
12	Santi	17770001	2017-01-01	2000-09-19	Jakarta	8
13	Subajo	S.Kom	1978-10-10	1978-10-10	Metro	6
14	Suwo...	S.P.	1980-09-22	1980-09-22	Jakarta	9

At the bottom of the results pane, a message indicates: 'Query executed successfully.'

F. TOP

Menampilkan data mahasiswa dari 3 teratas dengan top seperti gambar di bawah ini

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure for 'DESKTOP-NELAPCB'. The 'Tables' node under 'AKADEMIK' contains 'MAHASISWA'. The 'SQLQuery1.sql' window on the right contains the following SQL code:

```
select top 3 * from MAHASISWA
```

The results grid shows three rows of data:

	idmhs	npm	nama	tempat_lhr	tanggal_lhr	sex	thn_masuk	idprodi	idstatusaka
1	1	17750001	Ria	Metro	2000-10-10	P	2017	6	1
2	2	17760001	Agus	Kalianda	2000-10-16	L	2017	7	1
3	3	17770001	Santi	Jakarta	2000-09-19	P	2017	8	1

G. IN

Menampilkan data dosen dari idprodi 6.7.8 saja dengan In seperti gambar di bawah ini

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure for 'DESKTOP-NELAPCB'. The 'Tables' node under 'AKADEMIK' contains 'DOSEN'. The 'SQLQuery17.sql' window on the right contains the following SQL code:

```
select * from DOSEN where idprodi in (6,7,8)
```

The results grid shows three rows of data:

	iddosen	nmdosen	gelar	tempat_lhr	tanggal_lhr	sex	idprodi	idpendidikan
1	1	Subarjo	S.Kom	Metro	1978-10-10	L	6	1
2	2	Wendi	S.E., M.M.	Jakarta	1977-10-11	L	7	2
3	3	Heni	S.E., M.M.	Bandung	1980-02-19	P	8	2

H. BETWEEN

Menampilkan data mahasiswa yang berumur 20 – 21 tahun seperti gambar di bawah ini

```
Microsoft SQL Server Management Studio
File Edit View Query Debug Tools Window Community Help
New Query Object Explorer SQLQuery1.sql - D...CBVHaldian (52)* SQLQuery17.sql - ...PCB(Haldian (53))
Object Explorer
Connect > DESKTOP-NELAPCB (SQL Server 10
  Databases
    System Databases
    ak
    AKADEMIK
      Database Diagrams
      Tables
        System Tables
          dbo.DOSEN
          dbo.JENJANG
          dbo.MAHASISWA
          dbo.PENDIDIKAN
          dbo.PRODI
          dbo.STATUS_AKADEMIK
        Views
        Synonyms
        Programmability
        Service Broker
        Security
        haldianDB
        Pertemuan3
        Pertemuan4
      Security
      Server Objects
      Replication
      Management
Results Messages
select
  nama, npm, tanggal_lhr, tempat_lhr, DATEDIFF(year, tanggal_lhr, getdate()) as umur
from MAHASISWA
where DATEDIFF(YEAR, tanggal_lhr, GETDATE())
  between 20 and 21
```

	nama	npm	tanggal_lhr	tempat_lhr	umur
1	Riana	17780001	2002-09-22	Bandung	20
2	Budi	17790001	2001-02-17	Bengkulu	21
3	Edi	17750002	2001-03-11	Kotabumi	21
4	Hardi	17770002	2001-03-16	Natar	21

I. ISNULL

Menampilkan data keterangan status akademik mahasiswa dengan distinct case dan full join menghasilkan ‘Ada’ atau ‘Tidak ada’ seperti gambar di bawah ini

```
Microsoft SQL Server Management Studio
File Edit View Query Debug Tools Window Community Help
New Query Object Explorer SQLQuery1.sql - D...CBVHaldian (52)* SQLQuery17.sql - ...PCB(Haldian (53))
Object Explorer
Connect > DESKTOP-NELAPCB (SQL Server 10
  Databases
    System Databases
    ak
    AKADEMIK
      Database Diagrams
      Tables
        System Tables
          dbo.DOSEN
          dbo.JENJANG
          dbo.MAHASISWA
          dbo.PENDIDIKAN
          dbo.PRODI
          dbo.STATUS_AKADEMIK
        Views
        Synonyms
        Programmability
        Service Broker
        Security
        haldianDB
        Pertemuan3
        Pertemuan4
      Security
      Server Objects
      Replication
      Management
Results Messages
select distinct nmstatusaka,
  case
    when isnull(MAHASISWA.idstatusaka, 0) = 1 then 'Ada'
    when isnull(MAHASISWA.idstatusaka, 0) = 0 then 'Tidak Ada'
  end as [MAHASISWA]
from STATUS_AKADEMIK full join MAHASISWA on STATUS_AKADEMIK.idstatusaka = MAHASISWA.idstatusaka
```

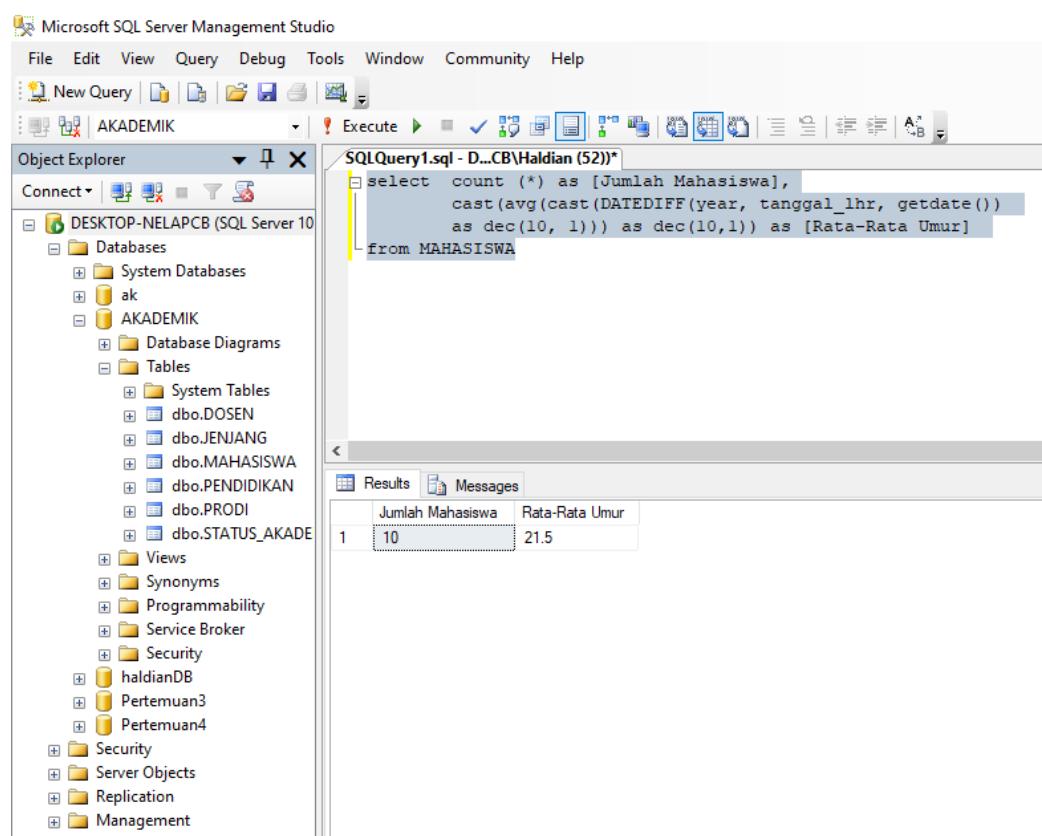
nmstatusaka	MAHASISWA
Aktif	Ada
Aktif	Tidak Ada
Aktif-Cuti	Tidak Ada
Dikeluarkan	Tidak Ada
Mengundurkan Diri-berhenti	Tidak Ada
Mengundurkan Diri-pindah	Tidak Ada
Transfer Studi	Tidak Ada

3. Minggu #7

1. Persiapkan alat dan bahan
2. Menjalankan aplikasi: *Microsoft SQL Server Management Studio (MSQLS-MS)*
3. Gunakanlah perintah *transact-SQL Query (SELECT)* untuk menyajikan data dari tabel-tabel yang ada pada database “AKADEMIK”, dengan memanfaatkan klausa pendukung berikut: *AVG, COUNT, MAX, MIN, SUM, MID, LEN, ROUND, NOW, UCASE, LCASE*. (catatan: masing-masing klausa, 1 contoh perintah SQL)

A. *AVG()*

Menampilkan jumlah mahasiswa dan rata-rata umr dari mahasiswa seperti gambar di bawah ini



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure, including the AKADEMIK database which contains several tables like DOSEN, JENJANG, MAHASISWA, PENDIDIKAN, PRODI, and STATUS_AKADEMIK. The SQL Query window on the right displays a query:

```
select count (*) as [Jumlah Mahasiswa],  
       cast(avg(cast(DATEDIFF(year, tanggal_lhr, getdate())  
                     as dec(10, 1))) as dec(10,1)) as [Rata-Rata Umur]  
  from MAHASISWA
```

The Results pane at the bottom shows the output of the query:

	Jumlah Mahasiswa	Rata-Rata Umur
1	10	21.5

B. COUNT()

Menampilkan jumlah dosen dari masing-masing prodi seperti gambar di bawah ini

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'AKADEMIK' containing various tables like DOSEN, JENJANG, MAHASISWA, PENDIDIKAN, PRODI, and STATUS_AKADEMIK. The central pane displays a T-SQL query:

```
select nmprodi as [Nama Prodi],  
       count(DOSEN.gelar) as Jumlah  
  from PRODI inner join DOSEN on PRODI.idprod = DOSEN.idprodi  
 group by nmprodi
```

The results pane shows the output of the query:

Nama Prodi	Jumlah
Agribisnis	1
Agribisnis Pangan	1
Akuntansi	1
Akuntansi Perpajakan	1
Manajemen Informatika	1

C. MAX()

Menampilkan umur maximal dari dosen seperti gambar di bawah ini

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'AKADEMIK' containing various tables like DOSEN, JENJANG, MAHASISWA, PENDIDIKAN, PRODI, and STATUS_AKADEMIK. The central pane displays a T-SQL query:

```
select COUNT(*) as [Jumlah Dosen],  
       MAX(datediff(year, tanggal_lhr, GETDATE())) as [Umur Max]  
  from DOSEN
```

The results pane shows the output of the query:

Jumlah Dosen	Umur Max
5	45

D. MIN()

Menampilkan minimal umur mahasiswa seperti gambar di bawah ini

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'AKADEMIK' containing various tables like DOSEN, JENJANG, MAHASISWA, PENDIDIKAN, PRODI, and STATUS_AKADEMIK. The SQL Query window on the right contains the following query:

```
select COUNT(*) as [Jumlah Mahasiswa],  
       MIN(datediff(year, tanggal_lhr, GETDATE())) as [Umur Min]  
from MAHASISWA
```

The Results pane at the bottom shows the output:

	Jumlah Mahasiswa	Umur Min
1	10	20

E. SUM()

Menampilkan jumlah prodi dan jumlah jenjang dengan sum seperti gambar di bawah ini

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'AKADEMIK' containing various tables like DOSEN, JENJANG, MAHASISWA, PENDIDIKAN, PRODI, and STATUS_AKADEMIK. The SQL Query window on the right contains the following query:

```
select COUNT(*) as [Jumlah Prodi],  
       SUM(idjenjang) as [Jumlah Jenjang]  
from PRODI
```

The Results pane at the bottom shows the output:

	Jumlah Prodi	Jumlah Jenjang
1	5	7

F. MID/SUBSTRING

Menampilkan data mahasiswa yang bertempat lahir di Jakarta dan tahun lahir hanya di ambil tahunnya saja seperti gambar di bawah ini

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'AKADEMIK' under 'DESKTOP-NELAPCB'. The 'Tables' node contains several tables including 'MAHASISWA'. The 'SQLQuery1.sql' window on the right contains the following SQL code:

```
select npm, nama, tempat_lhr,
       SUBSTRING(CAST(tempat_lhr as nvarchar), 1, 4) as [Tahun Lahir]
  from MAHASISWA where tempat_lhr='Jakarta'
```

The results pane shows one row of data:

npm	nama	tempat_lhr	Tahun Lahir
17770001	Santi	Jakarta	2000

G. LEN

Mengambil data dosen yang lahir di bandung dan mengambil nilai panjang nama dengan LEN seperti gambar di bawah ini

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'AKADEMIK' under 'DESKTOP-NELAPCB'. The 'Tables' node contains several tables including 'DOSEN'. The 'SQLQuery1.sql' window on the right contains the following SQL code:

```
select nm_dosen, gelar,
       LEN(nm_dosen) as [Panjang Nama]
  from DOSEN where tempat_lhr = 'Bandung'
```

The results pane shows one row of data:

nm_dosen	gelar	Panjang Nama
Heni	S.E., M.M.	4

H. ROUND()

Menampilkan pembulatan dari jumlah DOSEN dengan oprator pembagi Seperti gambar di bawah ini

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'AKADEMIK' containing tables like DOSEN, JENJANG, MAHASISWA, PENDIDIKAN, PRODI, and STATUS_AKADEMIK. The SQL Query window on the right contains the following code:

```
select COUNT(*) as DOSEN, 5 as Pembagi,
       round(cast(count(*) as dec (5,1))/5,1) as hasil
  from DOSEN
```

The Results pane at the bottom shows the output:

	DOSEN	Pembagi	hasil
1	5	5	1.00000

I. GETDATE()

Menampilkan nama dosen, gelar, tanggal_lhr, sekarang dan umur seperti gambar di bawah ini

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'AKADEMIK' containing tables like DOSEN, JENJANG, MAHASISWA, PENDIDIKAN, PRODI, and STATUS_AKADEMIK. The SQL Query window on the right contains the following code:

```
select nmdosen, gelar, tanggal_lhr,
       CAST(getdate() as date) as Sekarang,
       DATEDIFF(year, tanggal_lhr, GETDATE()) as Umur
  from DOSEN
```

The Results pane at the bottom shows the output:

	nmdosen	gelar	tanggal_lhr	Sekarang	Umur
1	Subajo	S.Kom	1978-10-10	2022-04-10	44
2	Wendi	S.E., M.M.	1977-10-11	2022-04-10	45
3	Heni	S.E., M.M.	1980-02-19	2022-04-10	42
4	Suwondo	S.P.	1980-09-22	2022-04-10	42
5	Hendro	S.P., M.P	2981-01-17	2022-04-10	-959

J. **UPPER()**

Menampilkan nama prodi menjadi besar semua dengan seperti gambar di bawah ini

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'AKADEMIK' containing various tables like DOSEN, JENJANG, MAHASISWA, PENDIDIKAN, PRODI, and STATUS_AKADEMIK. The SQL Query window on the right contains the following SQL code:

```
select UPPER(nmprodi) as prodi, akreditasi, Thn_berdiri
from PRODI
```

The Results pane displays the output of the query:

prodi	akreditasi	Thn_berdiri
MANAJEMEN INFORMATIKA	B	2006
AKUNTANSI	A	2004
AKUNTANSI PERPAJAKAN	-	2018
AGRIBISNIS	B	2002
AGRIBISNIS PANGAN	-	2018

K. **LOWER()**

Menampilkan nama dosen menjadi kecil semua seperti gambar di bawah ini

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'AKADEMIK' containing various tables like DOSEN, JENJANG, MAHASISWA, PENDIDIKAN, PRODI, and STATUS_AKADEMIK. The SQL Query window on the right contains the following SQL code:

```
select LOWER(nmdosen) as NamaDosen, idpendidikan
from DOSEN where idpendidikan=2
```

The Results pane displays the output of the query:

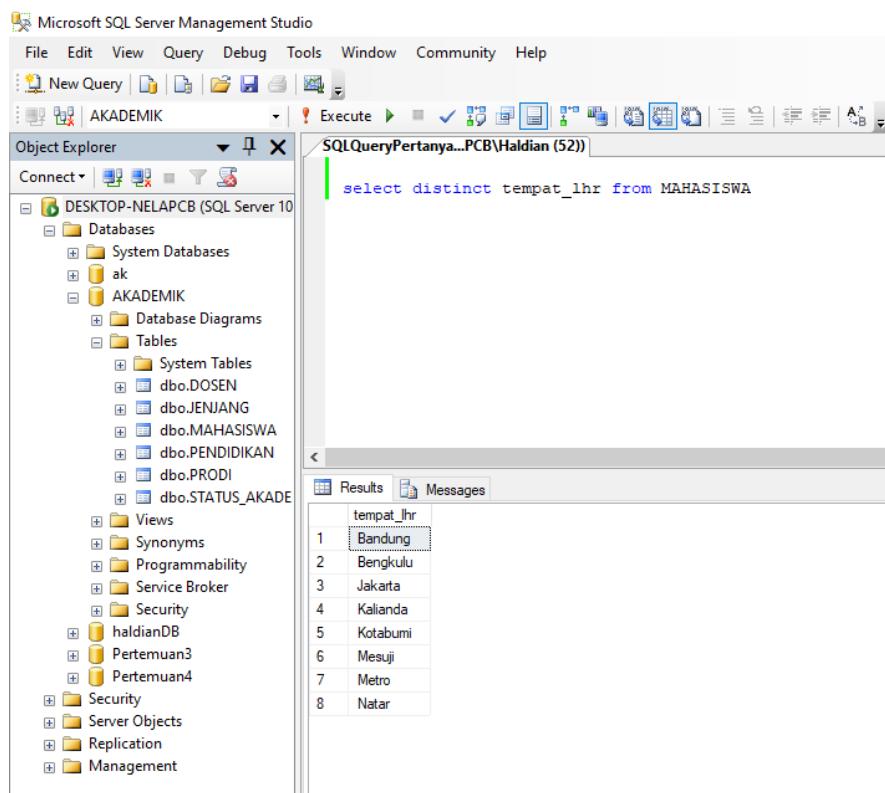
NamaDosen	idpendidikan
wendi	2
heni	2
hendro	2

B. TUGAS DAN PERTANYAAN

1. Minggu #5

- A. Buatlah dokumen T-SQL dari praktikum minggu 5 prosedur kerja 3 (Sudah terlampir di prosedur kerja).
- B. Menggunakan perintah T-SQL (SELECT), sajikan informasi berikut:

- Data kota tempat lahir mahasiswa



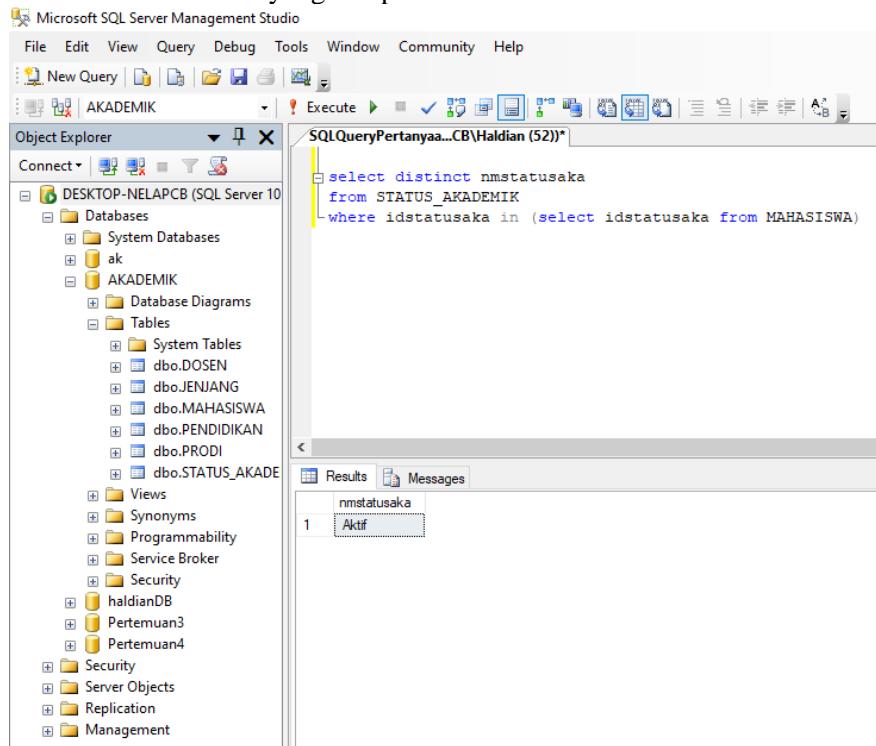
The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the database 'AKADEMIK' is selected, and the 'Tables' node is expanded, showing tables like DOSEN, JENJANG, MAHASISWA, PENDIDIKAN, PRODI, and STATUS_AKADEMIK. In the center pane, a query window titled 'SQLQueryPertanya...PCB\Haldian (52)>' contains the following SQL code:

```
select distinct tempat_lhr from MAHASISWA
```

The results pane shows a table with one column 'tempat_lhr' containing the following data:

tempat_lhr
1 Bandung
2 Bengkulu
3 Jakarta
4 Kalianda
5 Kotabumi
6 Mesuji
7 Metro
8 Natar

- Data status akademik yang ada pada mahasiswa terdaftar.



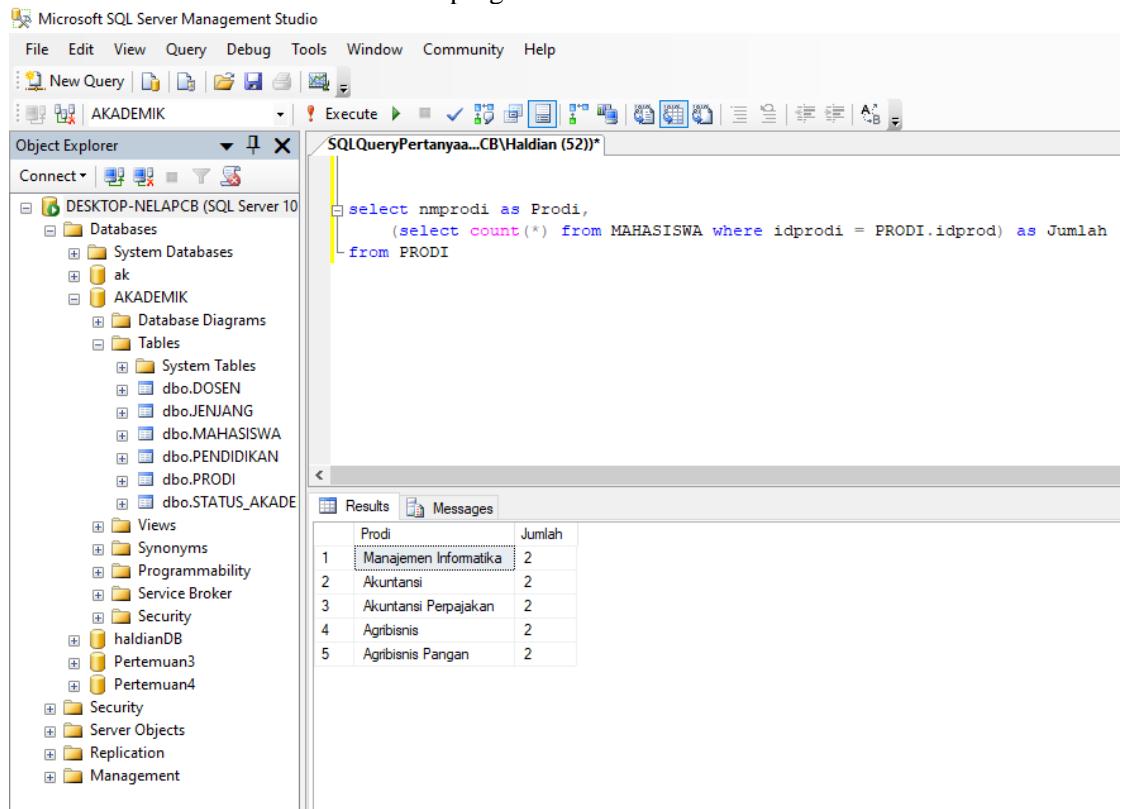
The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the database 'AKADEMIK' is selected, and the 'Tables' node is expanded, showing tables like DOSEN, JENJANG, MAHASISWA, PENDIDIKAN, PRODI, and STATUS_AKADEMIK. In the center pane, a query window titled 'SQLQueryPertanya...CB\Haldian (52)*' contains the following SQL code:

```
select distinct nmstatusaka  
from STATUS_AKADEMIK  
where idstatusaka in (select idstatusaka from MAHASISWA)
```

The results pane shows a table with one column 'nmstatusaka' containing the following data:

nmstatusaka
1 Aktif

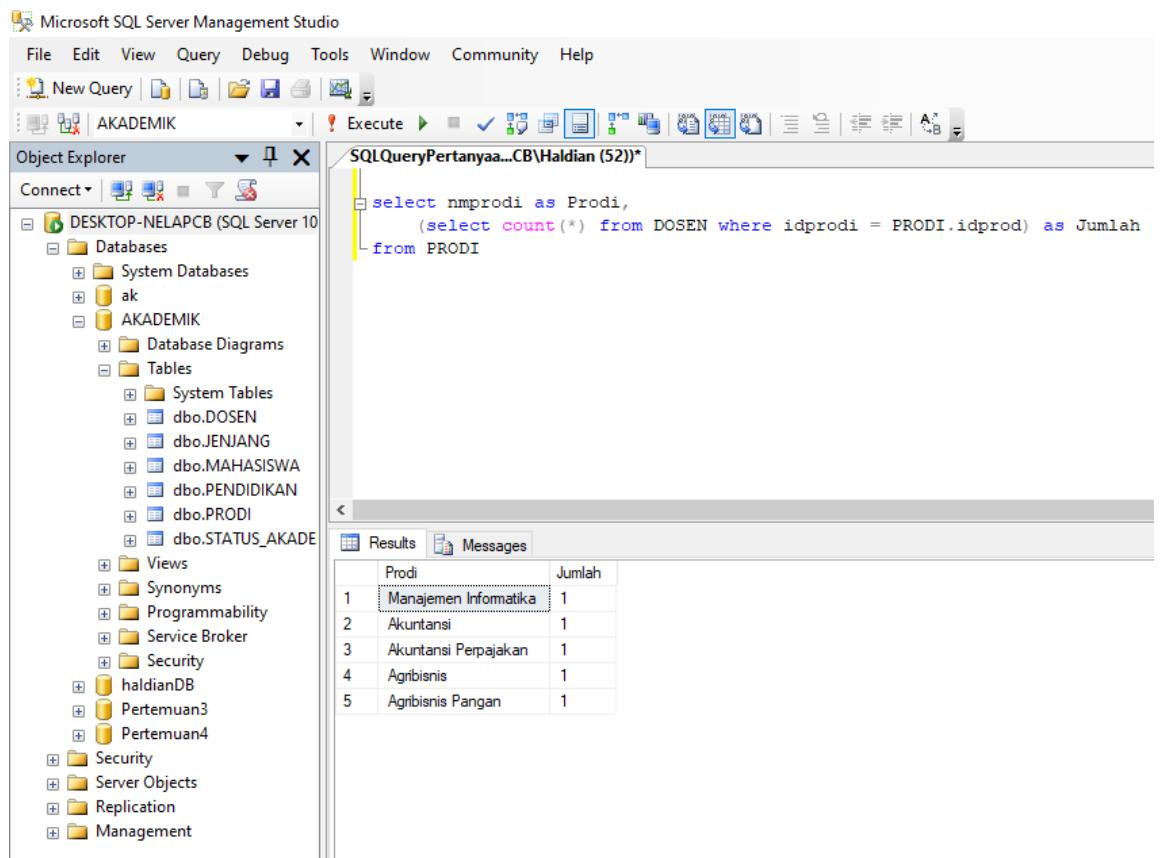
- Data Jumlah mahasiswa berdasarkan program studi



```
Microsoft SQL Server Management Studio
File Edit View Query Debug Tools Window Community Help
New Query Execute
Object Explorer Results Messages
SQLQueryPertanyaan...CB\Haldian (52)*
select nmprodi as Prodi,
       (select count(*) from MAHASISWA where idprodi = PRODI.idprod) as Jumlah
from PRODI
```

	Prodi	Jumlah
1	Manajemen Informatika	2
2	Akuntansi	2
3	Akuntansi Perpajakan	2
4	Agribisnis	2
5	Agribisnis Pangan	2

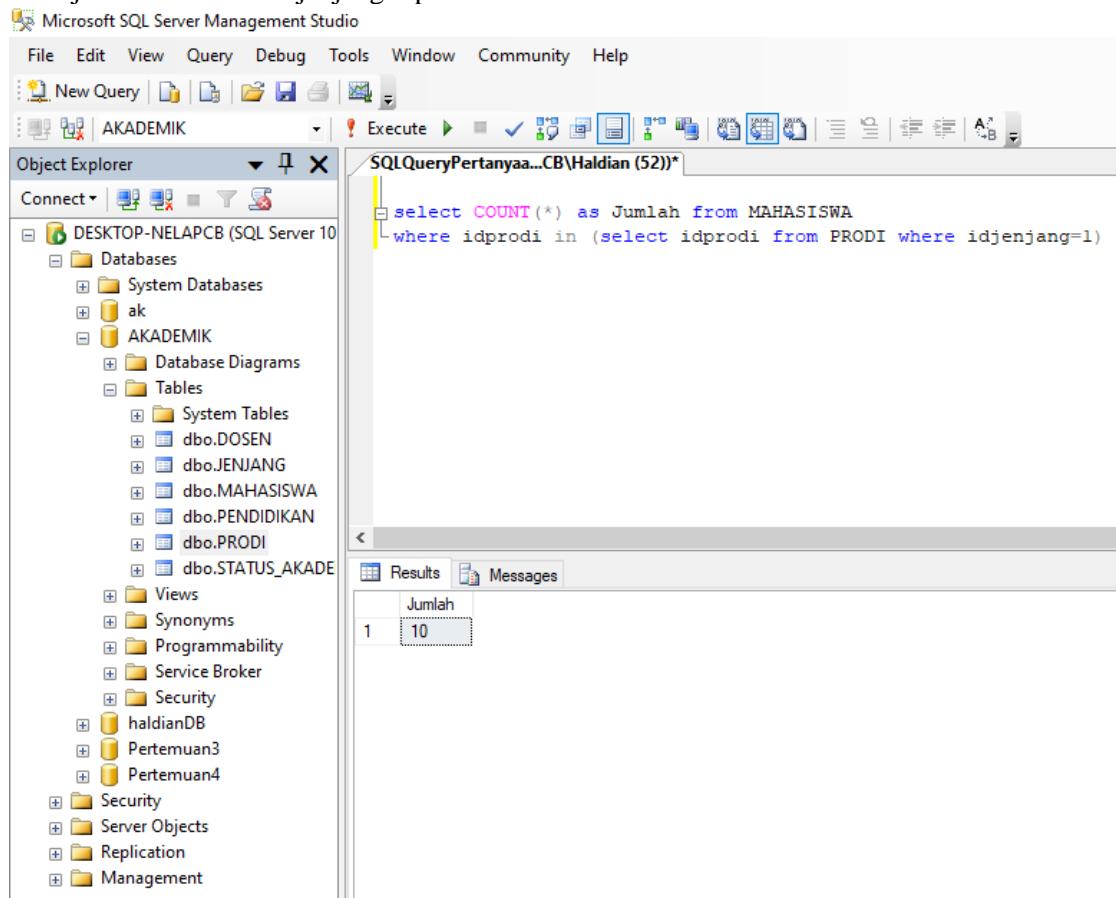
- Data Jumlah dosen berdasarkan program studi



```
Microsoft SQL Server Management Studio
File Edit View Query Debug Tools Window Community Help
New Query Execute
Object Explorer Results Messages
SQLQueryPertanyaan...CB\Haldian (52)*
select nmprodi as Prodi,
       (select count(*) from DOSEN where idprodi = PRODI.idprod) as Jumlah
from PRODI
```

	Prodi	Jumlah
1	Manajemen Informatika	1
2	Akuntansi	1
3	Akuntansi Perpajakan	1
4	Agribisnis	1
5	Agribisnis Pangan	1

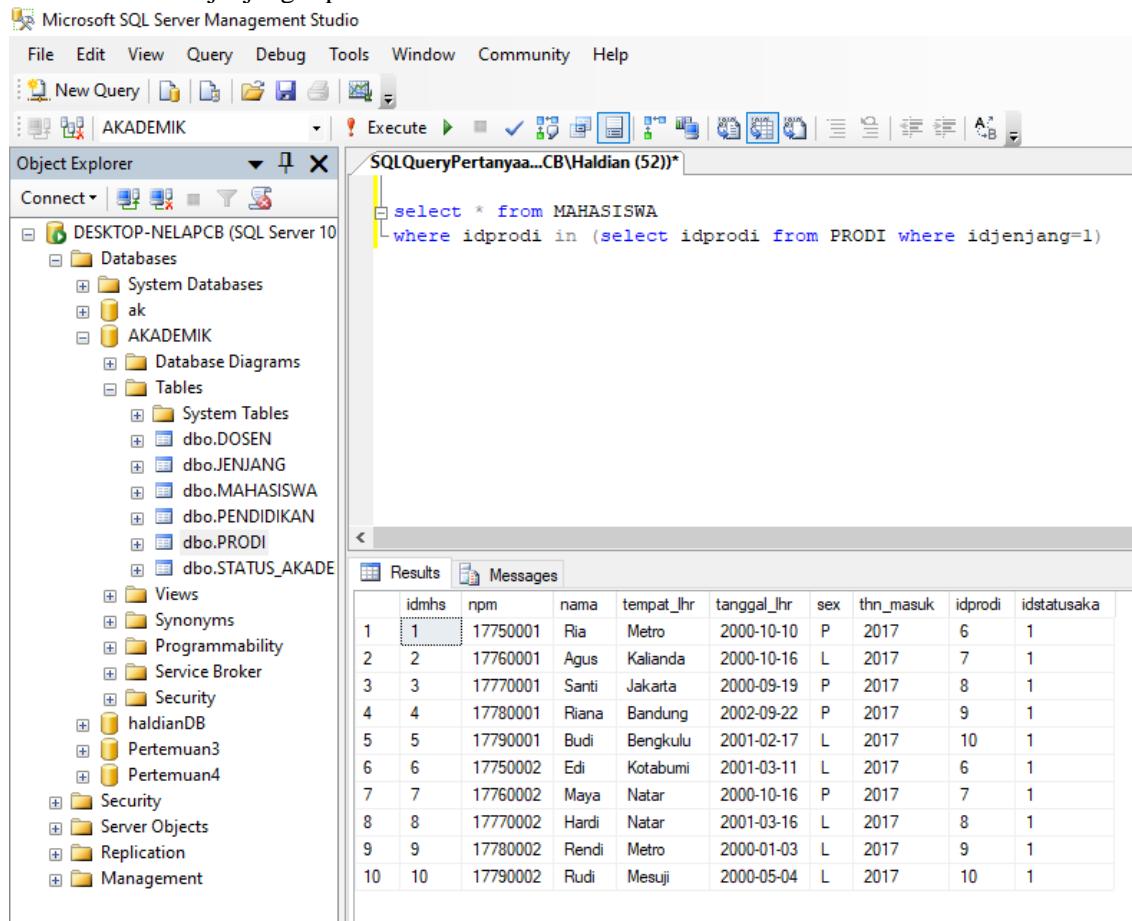
➤ Data jumlah mahasiswa jenjang diploma



```
select COUNT(*) as Jumlah from MAHASISWA
where idprodi in (select idprodi from PRODI where idjenjang=1)
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure, including the AKADEMIK database which contains tables like DOSEN, JENJANG, MAHASISWA, PENDIDIKAN, PRODI, and STATUS_AKADEMIK. The Results pane on the right displays the output of the executed query, showing a single row with the value 10 under the column 'Jumlah'.

➤ Data mahasiswa jenjang diploma



```
select * from MAHASISWA
where idprodi in (select idprodi from PRODI where idjenjang=1)
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure, including the AKADEMIK database which contains tables like DOSEN, JENJANG, MAHASISWA, PENDIDIKAN, PRODI, and STATUS_AKADEMIK. The Results pane on the right displays the output of the executed query, showing 10 rows of student data. The columns are idmhs, npm, nama, tempat_lhr, tanggal_lhr, sex, thn_masuk, idprodi, and idstatusaka.

	idmhs	npm	nama	tempat_lhr	tanggal_lhr	sex	thn_masuk	idprodi	idstatusaka
1	1	17750001	Ria	Metro	2000-10-10	P	2017	6	1
2	2	17760001	Agus	Kallanda	2000-10-16	L	2017	7	1
3	3	17770001	Santi	Jakarta	2000-09-19	P	2017	8	1
4	4	17780001	Riana	Bandung	2002-09-22	P	2017	9	1
5	5	17790001	Budi	Bengkulu	2001-02-17	L	2017	10	1
6	6	17750002	Edi	Kotabumi	2001-03-11	L	2017	6	1
7	7	17760002	Maya	Natar	2000-10-16	P	2017	7	1
8	8	17770002	Hardi	Natar	2001-03-16	L	2017	8	1
9	9	17780002	Rendi	Metro	2000-01-03	L	2017	9	1
10	10	17790002	Rudi	Mesuji	2000-05-04	L	2017	10	1

- Data jumlah mahasiswa jenjang sarjana terapan

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a connection to 'DESKTOP-NELAPCB (SQL Server 10)' with the 'AKADEMIK' database selected. The 'Tables' node under 'AKADEMIK' contains 'MAHASISWA'. The 'Results' tab on the right displays the output of the following query:

```
select count(*) as Jumlah from MAHASISWA
where idprodi in (select idprodi from PRODI where idjenjang=2)
```

The results show a single row with 'Jumlah' = 10.

- Data dosen jenjang diploma

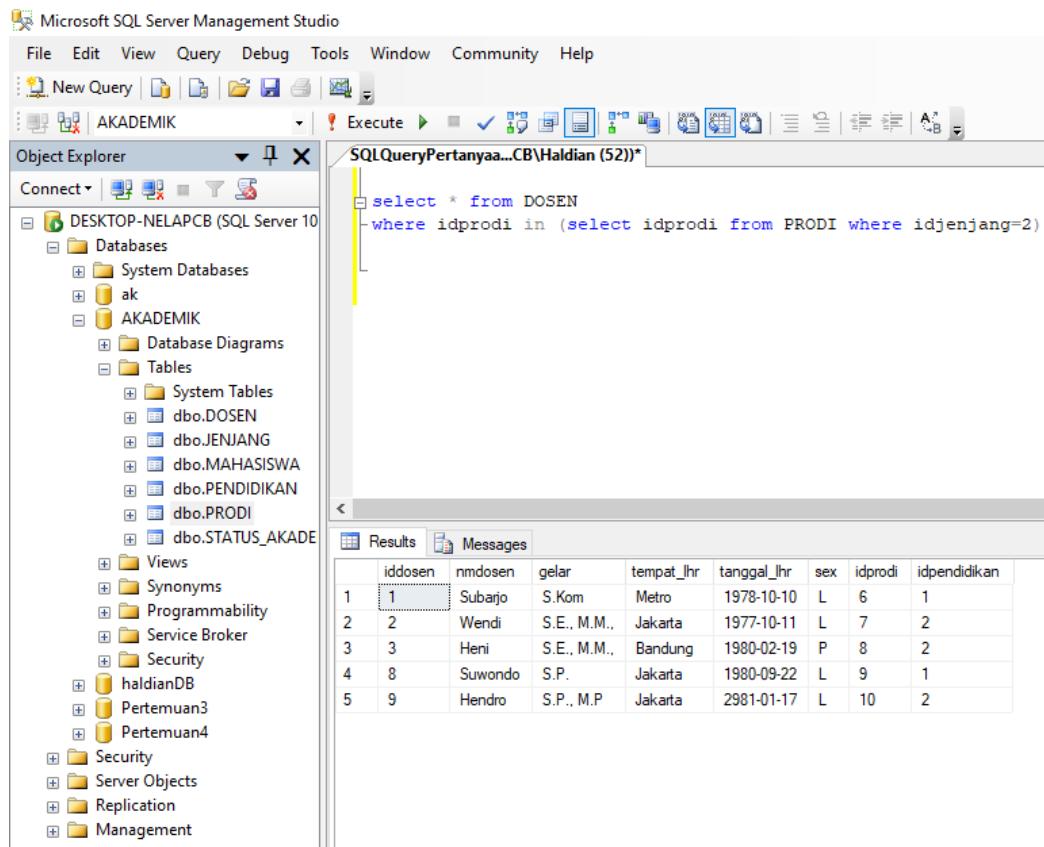
The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a connection to 'DESKTOP-NELAPCB (SQL Server 10)' with the 'AKADEMIK' database selected. The 'Tables' node under 'AKADEMIK' contains 'DOSEN'. The 'Results' tab on the right displays the output of the following query:

```
select * from DOSEN
where idprodi in (select idprodi from PRODI where idjenjang=1)
```

The results show five rows of data:

	iddosen	nmdosen	gelar	tempat_lhr	tanggal_lhr	sex	idprodi	idpendidikan
1	1	Subarjo	S.Kom	Metro	1978-10-10	L	6	1
2	2	Wendi	S.E., M.M.	Jakarta	1977-10-11	L	7	2
3	3	Heni	S.E., M.M..	Bandung	1980-02-19	P	8	2
4	8	Suwondo	S.P.	Jakarta	1980-09-22	L	9	1
5	9	Hendro	S.P., M.P	Jakarta	2981-01-17	L	10	2

- Data dosen jenjang sarjana terapan



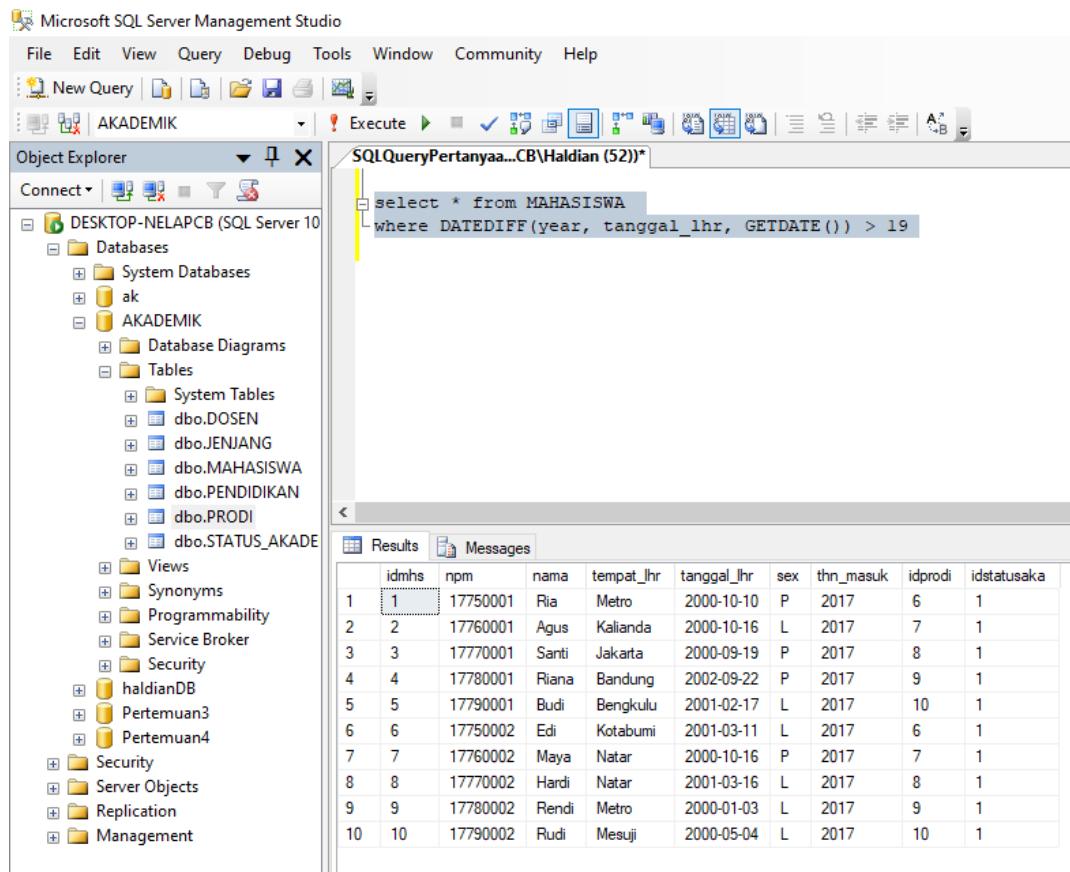
The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure under 'DESKTOP-NELAPCB (SQL Server 10)'. The 'Tables' node under 'AKADEMIK' contains 'dbo.DOSEN', 'dbo.JENJANG', 'dbo.MAHASISWA', 'dbo.PENDIDIKAN', 'dbo.PRODI', and 'dbo.STATUS_AKADEMIK'. The 'Results' tab on the right displays the output of the following query:

```
select * from DOSEN
where idjenjang=2
```

The results show five rows of data:

	iddosen	nmdosen	gelar	tempat_lhr	tanggal_lhr	sex	idprodi	idpendidikan
1	1	Subajyo	S.Kom	Metro	1978-10-10	L	6	1
2	2	Wendi	S.E., M.M.	Jakarta	1977-10-11	L	7	2
3	3	Heni	S.E., M.M.	Bandung	1980-02-19	P	8	2
4	8	Suwondo	S.P.	Jakarta	1980-09-22	L	9	1
5	9	Hendro	S.P., M.P	Jakarta	2981-01-17	L	10	2

- Data mahasiswa yang umurnya saat ini lebih dari 19 tahun



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure under 'DESKTOP-NELAPCB (SQL Server 10)'. The 'Tables' node under 'AKADEMIK' contains 'dbo.DOSEN', 'dbo.JENJANG', 'dbo.MAHASISWA', 'dbo.PENDIDIKAN', 'dbo.PRODI', and 'dbo.STATUS_AKADEMIK'. The 'Results' tab on the right displays the output of the following query:

```
select * from MAHASISWA
where DATEDIFF(year, tanggal_lhr, GETDATE()) > 19
```

The results show ten rows of data:

	idmhs	npm	nama	tempat_lhr	tanggal_lhr	sex	thn_masuk	idprodi	idstatusaka
1	1	17750001	Ria	Metro	2000-10-10	P	2017	6	1
2	2	17760001	Agus	Kalianda	2000-10-16	L	2017	7	1
3	3	17770001	Santi	Jakarta	2000-09-19	P	2017	8	1
4	4	17780001	Riana	Bandung	2002-09-22	P	2017	9	1
5	5	17790001	Budi	Bengkulu	2001-02-17	L	2017	10	1
6	6	17750002	Edi	Kotabumi	2001-03-11	L	2017	6	1
7	7	17760002	Maya	Natar	2000-10-16	P	2017	7	1
8	8	17770002	Hardi	Natar	2001-03-16	L	2017	8	1
9	9	17780002	Rendi	Metro	2000-01-03	L	2017	9	1
10	10	17790002	Rudi	Mesuji	2000-05-04	L	2017	10	1

2. Minggu #6

- A. Buatlah dokumen T-SQL dari praktikum minggu 6 prosedur kerja 3. (Sudah terlampir di prosedur kerja).
- B. Menggunakan perintah T-SQL (SELECT), kerjakan tugas berikut:
- Jika klausa atau kondisi INNER JOIN diimplementasikan pada tabel-tabel yang ada pada database “AKADEMIK”. Berapa kombinasi relasi 2 tabel yang dapat dibuat, Data apa saja yang dapat disajikan, dan bagaimana perintah SQL yang ditulis untuk penyajian datadata tersebut.
 - ✓ Kombinasi relasi 2 tabel yang dapat dibuat yaitu: PRODI dengan JENJANG, PRODI dengan MAHASISWA, PRODI dengan DOSEN, MAHASISWA dengan STATUS_AKADEMIK, dan DOSEN dengan PENDIDIKAN.
 - ✓ Data yang dapat di sajikan yaitu menggabungkan baris dari dua atau lebih tabel, berdasarkan kolom terkait di antara mereka. Serta mengembalikan rekaman yang memiliki nilai yang cocok di kedua tabel
 - ✓ Contoh penulisan perintah SQL:

Menampilkan daftar DOSEN beserta nama PRODI-nya

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'AKADEMIK' containing several tables: DOSEN, JENJANG, MAHASISWA, PENDIDIKAN, PRODI, and STATUS_AKADEMIK. The 'SQLQuery1.sql' window in the center contains the following T-SQL code:

```
select nmdosen, gelar, nmprodi, PRODI.idprod
from PRODI inner join DOSEN on PRODI.idprod = DOSEN.idprod
```

The 'Results' tab at the bottom displays the output of the query as a table:

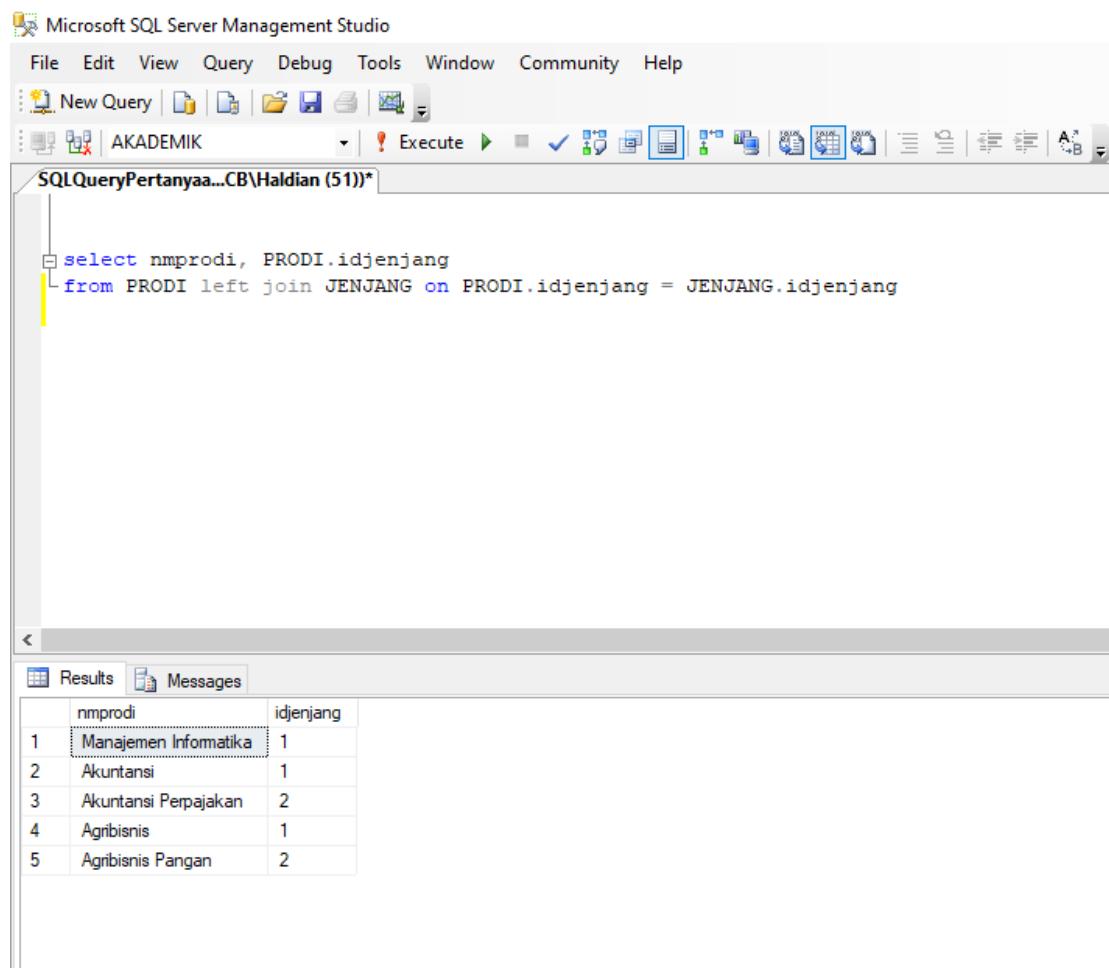
	nmdosen	gelar	nmprodi	idprod
1	Subarjo	S.Kom	Manajemen Informatika	6
2	Wendi	S.E., M.M.	Akuntansi	7
3	Heni	S.E., M.M.	Akuntansi Perpajakan	8
4	Suwondo	S.P.	Agribisnis	9
5	Hendro	S.P., M.P	Agribisnis Pangan	10

- Implementasikan klausula atau kondisi LEFT JOIN pada 2 tabel yang ada pada database “AKADEMIK”. Tuliskan data apa yang disajikan, dan bagaimana perintah SQL yang ditulis untuk penyajian data tersebut.
- ✓ Mengembalikan semua record dari tabel kiri (tabel1), dan record yang cocok dari tabel kanan (tabel2). Hasilnya adalah 0 record dari sisi kanan, jika tidak ada yang cocok.

Menampilkan data. Pada contoh disini saya mengambil data Prodi dan Jenjang

- ✓ Contoh penulisan perintah SQL:

Menampilkan daftar PRODI dan JENJANG



The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "Microsoft SQL Server Management Studio". The toolbar includes standard options like File, Edit, View, Query, Debug, Tools, Window, Community, and Help. Below the toolbar is a ribbon bar with tabs for New Query, Execute, and various database management icons. The main window has a tab labeled "SQLQueryPertanyaan...CB\Haldian (51)*". In the query editor, the following SQL code is written:

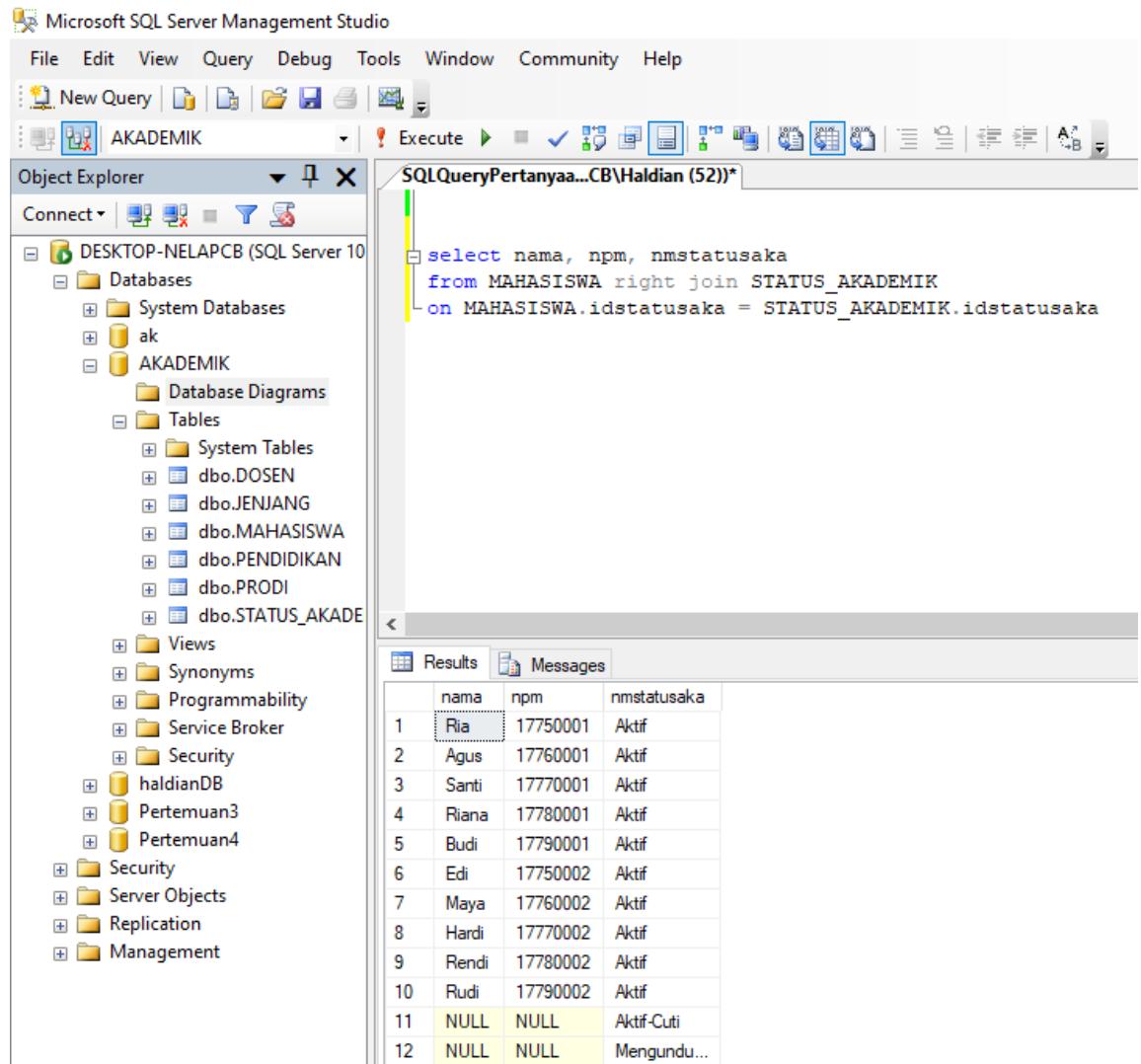
```
select nmprodi, PRODI.idjenjang
from PRODI left join JENJANG on PRODI.idjenjang = JENJANG.idjenjang
```

Below the query editor is a results grid titled "Results". The grid displays the following data:

	nmprodi	idjenjang
1	Manajemen Informatika	1
2	Akuntansi	1
3	Akuntansi Perpajakan	2
4	Agribisnis	1
5	Agribisnis Pangan	2

- Implementasikan klausula atau kondisi RIGHT JOIN pada 2 tabel yang ada pada database “AKADEMIK”. Tuliskan data apa yang disajikan, dan bagaimana perintah SQL yang ditulis untuk penyajian data tersebut.
- ✓ mengembalikan semua record dari tabel kanan (table2), dan record yang cocok dari tabel kiri (table1). Hasilnya adalah 0 record dari sisi kiri, jika tidak ada yang cocok.
- ✓ Contoh penulisan perintah SQL:

Menampilkan MAHASISWA dengan STATUS_AKADEMIK



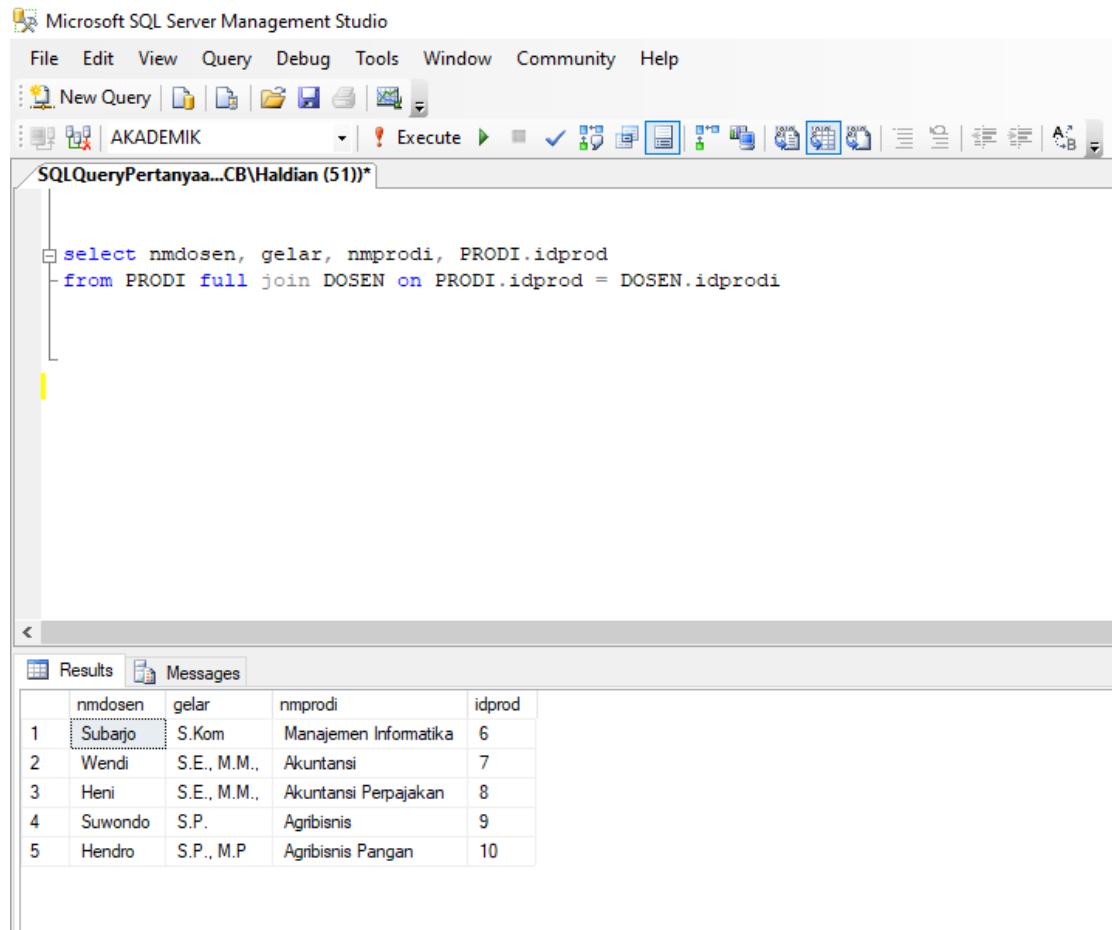
```
Microsoft SQL Server Management Studio
File Edit View Query Debug Tools Window Community Help
New Query Execute
AKADEMIK Object Explorer SQLQueryPertanya...CB\Haldian (52)*
DESKTOP-NELAPCB (SQL Server 10)
Databases System Databases ak AKADEMIK Database Diagrams
Tables System Tables dbo.DΟSEN
dbo.JENJANG dbo.MAHASISWA
dbo.PENDIDIKAN dbo.PRODI
dbo.STATUS_AKADE Views Synonyms Programmability Service Broker Security
haldianDB Pertemuan3 Pertemuan4 Security Server Objects Replication Management
Results Messages
```

```
select nama, npm, nmstatusaka
from MAHASISWA right join STATUS_AKADEMIK
on MAHASISWA.idstatusaka = STATUS_AKADEMIK.idstatusaka
```

	nama	npm	nmstatusaka
1	Ria	17750001	Aktif
2	Agus	17760001	Aktif
3	Santi	17770001	Aktif
4	Riana	17780001	Aktif
5	Budi	17790001	Aktif
6	Edi	17750002	Aktif
7	Maya	17760002	Aktif
8	Hardi	17770002	Aktif
9	Rendi	17780002	Aktif
10	Rudi	17790002	Aktif
11	NULL	NULL	Aktif-Cuti
12	NULL	NULL	Mengundu...

- Implementasikan klausula atau kondisi FULL JOIN pada 2 tabel yang ada pada database “AKADEMIK”. Tuliskan data apa yang disajikan, dan bagaimana perintah SQL yang ditulis untuk penyajian data tersebut.
- ✓ Mengembalikan seluruh baris dari kedua tabel yang dikenai ON termasuk data-data yang bernilai NULL.
- ✓ Contoh penulisan perintah SQL:

Menampilkan Dosen dan Prodi



The screenshot shows the Microsoft SQL Server Management Studio interface. The top menu bar includes File, Edit, View, Query, Debug, Tools, Window, Community, and Help. Below the menu is a toolbar with various icons. The main window has a title bar "SQLQueryPertanyaan...CB\Haldian (51)*". The query editor contains the following SQL code:

```
select nm dosen, gelar, nm prodi, PRODI.idprod
from PRODI full join DOSEN on PRODI.idprod = DOSEN.idprod
```

The results pane at the bottom displays the following data:

	nm dosen	gelar	nm prodi	idprod
1	Subajo	S.Kom	Manajemen Informatika	6
2	Wendi	S.E., M.M..	Akuntansi	7
3	Heni	S.E., M.M..	Akuntansi Perpajakan	8
4	Suwondo	S.P.	Agribisnis	9
5	Hendro	S.P., M.P	Agribisnis Pangan	10

- Implementasikan klausula atau kondisi UNION pada 2 tabel yang ada pada database “AKADEMIK”. tuliskan data apa saja yang disajikan, dan bagaimana perintah SQL yang ditulis untuk penyajian data tersebut
- ✓ Menggabungkan kumpulan hasil dari dua atau lebih pernyataan SELECT. Setiap pernyataan SELECT dan harus memiliki jumlah kolom yang sama, Kolom juga harus memiliki tipe data yang serupa, Kolom di setiap pernyataan SELECT juga harus dalam urutan yang sama.
- ✓ Contoh penulisan perintah SQL:

Menampilkan seluruh civitas akademika

```

Microsoft SQL Server Management Studio
File Edit View Query Debug Tools Window Community Help
New Query | Execute | 
Object Explorer | SQLQuery1.sql - D...CB\Haldian (52)* |
SELECT
    nama, MAHASISWA.npm, MAHASISWA.thn_masuk, MAHASISWA.tanggal_lhr, MAHASISWA.tempat_lhr, MAHASISWA.idprodi
    FROM MAHASISWA
UNION
SELECT
    nimdoSEN, DOSEN.gelar, DOSEN.tanggal_lhr, DOSEN.tanggal_lhr, DOSEN.tempat_lhr, DOSEN.idprodi
    FROM DOSEN

```

	nama	npm	thn_masuk	tanggal_lhr	tempat_lhr	idprodi
1	Agus	17760001	2017-01-01	2000-10-16	Kalianda	7
2	Budi	17790001	2017-01-01	2001-02-17	Bengkulu	10
3	Edi	17750002	2017-01-01	2001-03-11	Kotabumi	6
4	Hardi	17770002	2017-01-01	2001-03-16	Natar	8
5	Hendro	S.P., M.P	2981-01-17	2981-01-17	Jakarta	10
6	Heri	S.E., M.M.	1980-02-19	1980-02-19	Bandung	8
7	Maya	17760002	2017-01-01	2000-10-16	Natar	7
8	Rendi	17780002	2017-01-01	2000-01-03	Metro	9
9	Ria	17750001	2017-01-01	2000-10-10	Metro	6
10	Riana	17780001	2017-01-01	2002-09-22	Bandung	9
11	Rudi	17790002	2017-01-01	2000-05-04	Mesuji	10
12	Santi	17770001	2017-01-01	2000-09-19	Jakarta	8
13	Subago	S.Kom	1978-10-10	1978-10-10	Metro	6
14	Suwo...	S.P.	1980-09-22	1980-09-22	Jakarta	9

Query executed successfully. | DESKTOP-NELAPCB (10.0 RTM) | DESKTOP-NELAPCB\Haldia... | AKADEMIK | 00:00

C. Menggunakan perintah T-SQL (SELECT), Sajikan informasi berikut:

- 5 Data mahasiswa dari seluruh program studi urut berdasarkan nama

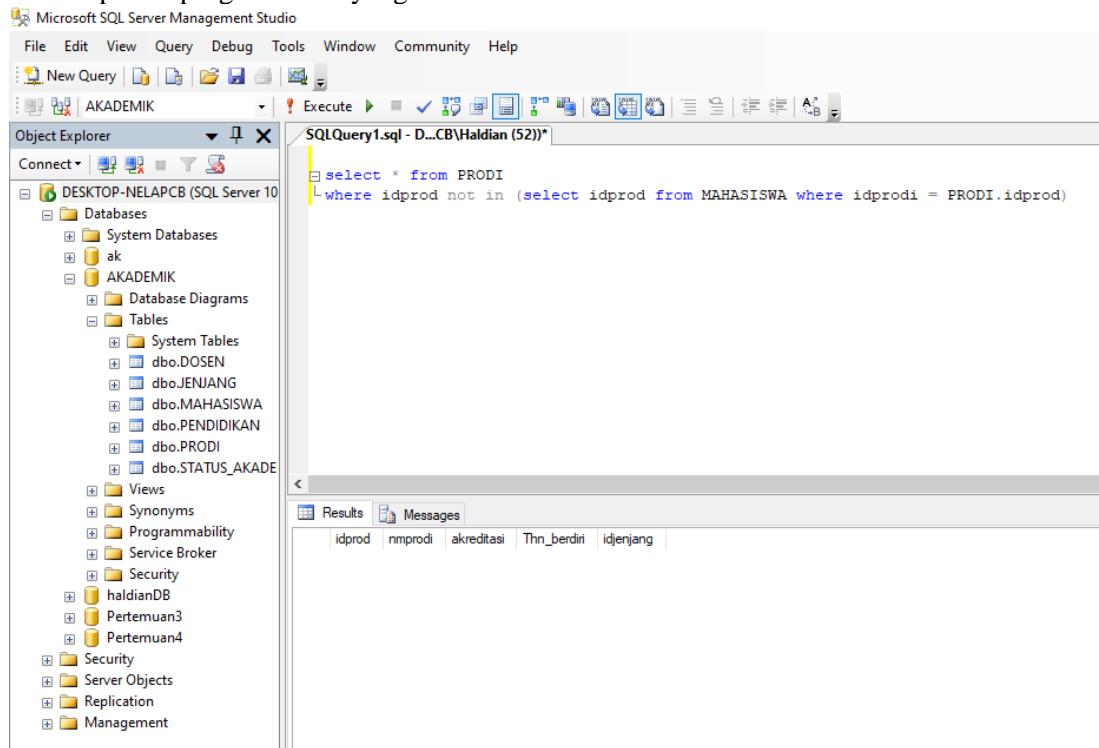
```

Microsoft SQL Server Management Studio
File Edit View Query Debug Tools Window Community Help
New Query | Execute | 
Object Explorer | SQLQueryPertanyaan...CB\Haldian (52)* |
SELECT top 5 * from MAHASISWA

```

	idmhs	npm	nama	tempat_lhr	tanggal_lhr	sex	thn_masuk	idprodi	idstatusaka
1	1	17750001	Ria	Metro	2000-10-10	P	2017	6	1
2	2	17760001	Agus	Kalianda	2000-10-16	L	2017	7	1
3	3	17770001	Santi	Jakarta	2000-09-19	P	2017	8	1
4	4	17780001	Riana	Bandung	2002-09-22	P	2017	9	1
5	5	17790001	Budi	Bengkulu	2001-02-17	L	2017	10	1

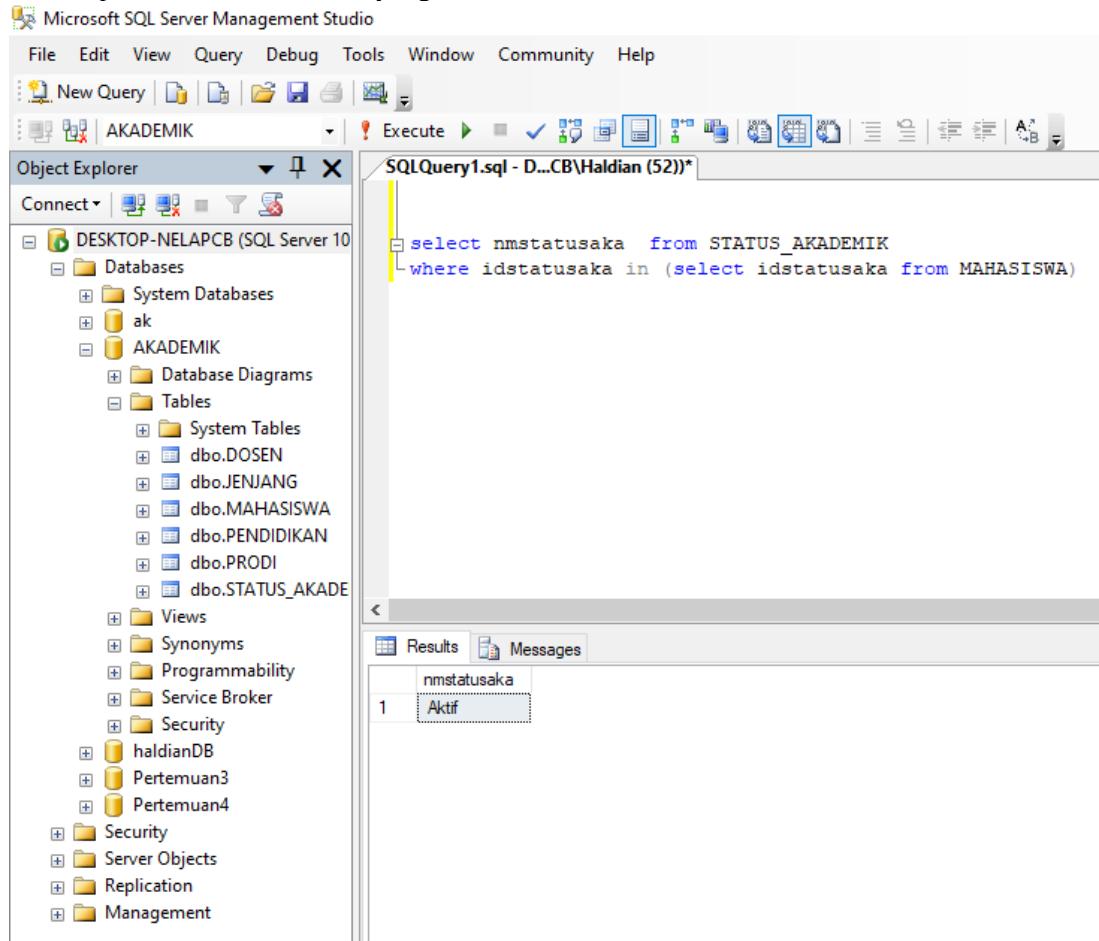
- Menampilkan program studi yang tidak memiliki mahasiswa



```
Microsoft SQL Server Management Studio
File Edit View Query Debug Tools Window Community Help
New Query Execute Object Explorer SQLQuery1.sql - D...CB\Haldian (52)*
DESKTOP-NELAPCB (SQL Server 10)
AKADEMIK
Databases
System Databases
ak
AKADEMIK
Database Diagrams
Tables
System Tables
dbo.DΟSEN
dbo.JENJANG
dbo.MAHASISWA
dbo.PENDIDIKAN
dbo.PRODI
dbo.STATUS_AKADE
Views
Synonyms
Programmability
Service Broker
Security
haldianDB
Pertemuan3
Pertemuan4
Security
Server Objects
Replication
Management
Object Explorer
Connect
SQLQuery1.sql - D...CB\Haldian (52)*
select * from PRODI
where idprod not in (select idprod from MAHASISWA where idprodi = PRODI.idprod)

Results Messages
idprod nmprod akreditasi Thn_berdiri idjenjang
```

- Menampilkan status akademik yang dimiliki mahasiswa



```
Microsoft SQL Server Management Studio
File Edit View Query Debug Tools Window Community Help
New Query Execute Object Explorer SQLQuery1.sql - D...CB\Haldian (52)*
DESKTOP-NELAPCB (SQL Server 10)
AKADEMIK
Databases
System Databases
ak
AKADEMIK
Database Diagrams
Tables
System Tables
dbo.DΟSEN
dbo.JENJANG
dbo.MAHASISWA
dbo.PENDIDIKAN
dbo.PRODI
dbo.STATUS_AKADE
Views
Synonyms
Programmability
Service Broker
Security
haldianDB
Pertemuan3
Pertemuan4
Security
Server Objects
Replication
Management
Object Explorer
Connect
SQLQuery1.sql - D...CB\Haldian (52)*
select nmstatusaka from STATUS_AKADEMIK
where idstatusaka in (select idstatusaka from MAHASISWA)

Results Messages
nmstatusaka
1 Aktif
```

- Menampilkan mahasiswa yang terdaftar pada tahun 2015 s.d. 2017

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'AKADEMIK' containing tables like DOSEN, JENJANG, MAHASISWA, PENDIDIKAN, PRODI, and STATUS_AKADEMIK. The SQL Query window on the right contains the following code:

```
select * from MAHASISWA
where thn_masuk between 2015 and 2017
```

The Results pane displays the output of the query, which is a table of student records. The columns are idmhs, npm, nama, tempat_lhr, tanggal_lhr, sex, thn_masuk, idprodi, and idstatusaka. The data is as follows:

	idmhs	npm	nama	tempat_lhr	tanggal_lhr	sex	thn_masuk	idprodi	idstatusaka
1	1	17750001	Ria	Metro	2000-10-10	P	2017	6	1
2	2	17760001	Agus	Kallanda	2000-10-16	L	2017	7	1
3	3	17770001	Santi	Jakarta	2000-09-19	P	2017	8	1
4	4	17780001	Riana	Bandung	2002-09-22	P	2017	9	1
5	5	17790001	Budi	Bengkulu	2001-02-17	L	2017	10	1
6	6	17750002	Edi	Kotabumi	2001-03-11	L	2017	6	1
7	7	17760002	Maya	Natar	2000-10-16	P	2017	7	1
8	8	17770002	Hardi	Natar	2001-03-16	L	2017	8	1
9	9	17780002	Rendi	Metro	2000-01-03	L	2017	9	1
10	10	17790002	Rudi	Mesuji	2000-05-04	L	2017	10	1

- Menampilkan Dosen yang lahir dari tanggal 01-01-1976 s.d. 31-12-1995

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'AKADEMIK' containing tables like DOSEN, JENJANG, MAHASISWA, PENDIDIKAN, PRODI, and STATUS_AKADEMIK. The SQL Query window on the right contains the following code:

```
select * from DOSEN
where tanggal_lhr between '1976-01-01' and '1995-12-31'
```

The Results pane displays the output of the query, which is a table of professor records. The columns are iddosen, nmdosen, gelar, tempat_lhr, tanggal_lhr, sex, idprodi, and idpendidikan. The data is as follows:

	iddosen	nmdosen	gelar	tempat_lhr	tanggal_lhr	sex	idprodi	idpendidikan
1	1	Subarjo	S.Kom	Metro	1978-10-10	L	6	1
2	2	Wendi	S.E., M.M.	Jakarta	1977-10-11	L	7	2
3	3	Heni	S.E., M.M.	Bandung	1980-02-19	P	8	2
4	8	Suwondo	S.P.	Jakarta	1980-09-22	L	9	1

- Menampilkan dosen yang data tempat lahir belum terisi

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure under 'DESKTOP-NELAPCB (SQL Server 10)'. The 'Tables' node under 'AKADEMIK' contains several tables: DOSEN, JENJANG, MAHASISWA, PENDIDIKAN, PRODI, and STATUS_AKADEMIK. The 'SQLQuery1.sql' window on the right contains the following SQL query:

```
select * from DOSEN where tempat_lhr is null
```

The results pane shows a table with columns: iddosen, nm dosen, gelar, tempat_lhr, tanggal_lhr, sex, idprodi, and idpendidikan. There are currently no rows displayed.

- Menampilkan 3 mahasiswa yang program studinya berjenjang D3 dan dosennya berjenjang S2

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure under 'DESKTOP-NELAPCB (SQL Server 10)'. The 'Tables' node under 'AKADEMIK' contains several tables: DOSEN, JENJANG, MAHASISWA, PENDIDIKAN, PRODI, and STATUS_AKADEMIK. The 'SQLQuery1.sql' window on the right contains the following SQL query:

```
select top 3 * from MAHASISWA
inner join PRODI on MAHASISWA.idprodi = PRODI.idprod
inner join DOSEN on PRODI.idprod = DOSEN.idprodi
where idjenjang = 1 and idpendidikan=2
```

The results pane shows a table with columns: idmhs, npm, nama, tempat_lhr, tanggal_lhr, sex, thn_masuk, idprodi, idstatusaka, idprod, nmprodi, akreditasi, Thn_berdiri, idjenjang, iddosen, nm dosen, gelar, tempat_lhr, and tanggal_lhr. The data is as follows:

idmhs	npm	nama	tempat_lhr	tanggal_lhr	sex	thn_masuk	idprodi	idstatusaka	idprod	nmprodi	akreditasi	Thn_berdiri	idjenjang	iddosen	nm dosen	gelar	tempat_lhr	tanggal_lhr
1	2	1776001	Agus	Kalanda	L	2000-10-16	7	1	7	Akuntansi	A	2004	1	2	Wendi	S.E., M.M.,	Jakarta	1977-10
2	7	1776002	Maya	Natar	P	2000-10-16	7	1	7	Akuntansi	A	2004	1	2	Wendi	S.E., M.M.,	Jakarta	1977-10

3. Minggu #7

- A. Buatlah dokumen T-SQL dari praktikum minggu 7 prosedur kerja 3. (Sudah terlampir di prosedur kerja).
- B. Menggunakan perintah T-SQL (SELECT), Sajikan informasi berikut:
 - Rata-rata umur mahasiswa program studi D3 manajemen informatika

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'AKADEMIK' containing various tables like DOSEN, JENJANG, MAHASISWA, PENDIDIKAN, PRODI, and STATUS_AKADEMIK. The central pane displays a query window with the following SQL code:

```
SQLQuery1.sql - D...CB\Haldian (52)*
select count (*) as [Jumlah Mahasiswa],
       cast(avg(cast(DATEDIFF(year, tanggal_lhr, getdate())))
            as dec(10, 1)) as [Rata-Rata Umur]
from MAHASISWA
```

The results pane shows a single row of data:

	Jumlah Mahasiswa	Rata-Rata Umur
1	10	21.5

- Jumlah mahasiswa masing-masing program studi

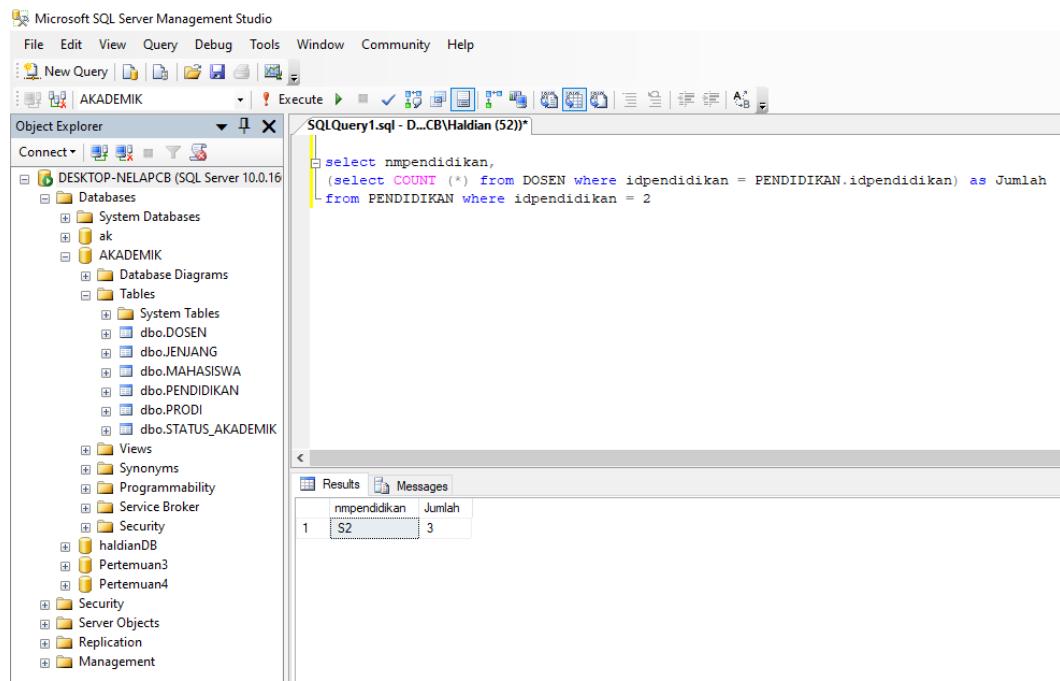
The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'AKADEMIK' containing various tables like DOSEN, JENJANG, MAHASISWA, PENDIDIKAN, PRODI, and STATUS_AKADEMIK. The central pane displays a query window with the following SQL code:

```
SQLQuery1.sql - D...CB\Haldian (52)*
select nmprodi,
       (select COUNT (*) from MAHASISWA where idprodi = PRODI.idprod) as Jumlah
from PRODI
```

The results pane shows a list of program studies and their counts:

nmprodi	Jumlah
Manajemen Informatika	2
Akuntansi	2
Akuntansi Perpajakan	2
Agribisnis	2
Agribisnis Pangan	2

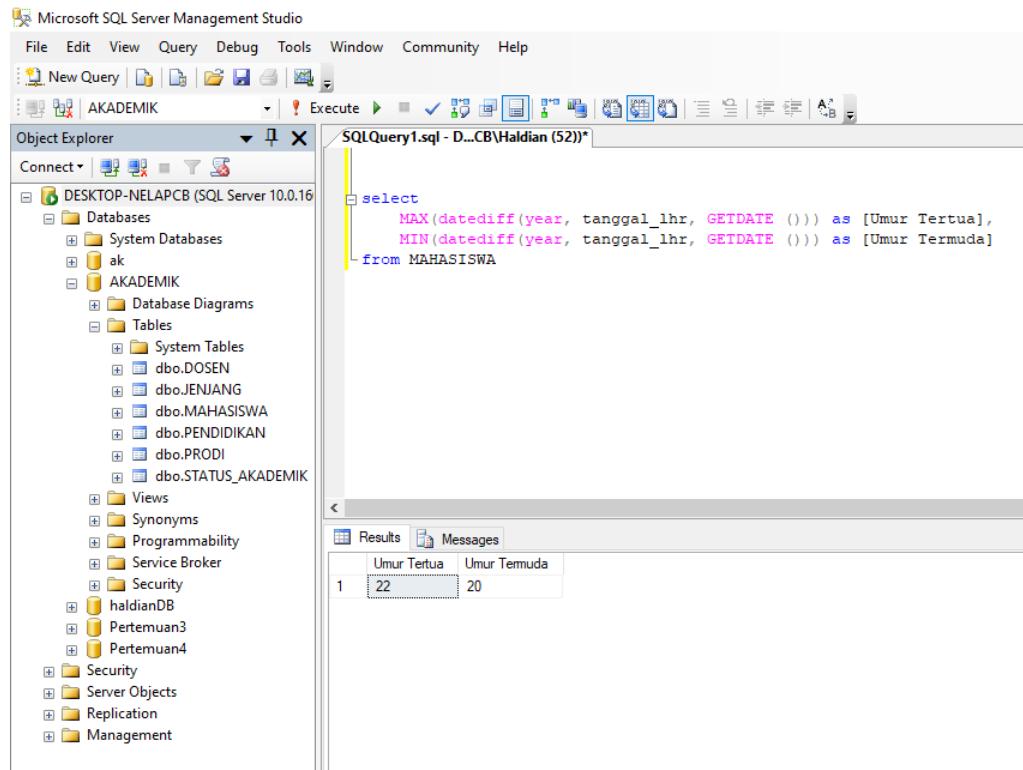
➤ Jumlah dosen dengan jenjang pendidikan S2



```
Microsoft SQL Server Management Studio
File Edit View Query Debug Tools Window Community Help
New Query | Execute | Results | Messages
Object Explorer | AKADEMIK | SQLQuery1.sql - D...CB\Haldian (52)*
select nmpendidikan,
(select COUNT (*) from DOSEN where idpendidikan = PENDIDIKAN.idpendidikan) as Jumlah
from PENDIDIKAN where idpendidikan = 2
```

nmpendidikan	Jumlah
S2	3

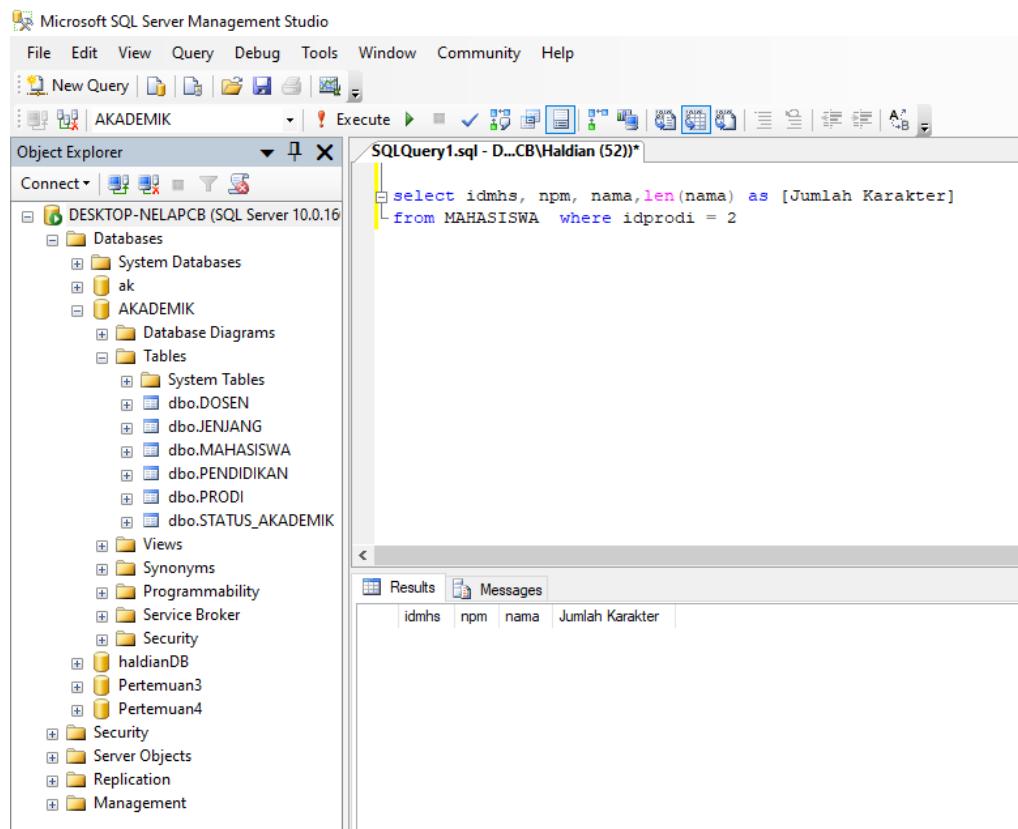
➤ Umur Mahasiswa Tertua dan Termuda dari seluruh Program Studi



```
Microsoft SQL Server Management Studio
File Edit View Query Debug Tools Window Community Help
New Query | Execute | Results | Messages
Object Explorer | AKADEMIK | SQLQuery1.sql - D...CB\Haldian (52)*
select
MAX(datediff(year, tanggal_lhr, GETDATE ()) ) as [Umur Tertua],
MIN(datediff(year, tanggal_lhr, GETDATE ()) ) as [Umur Termuda]
from MAHASISWA
```

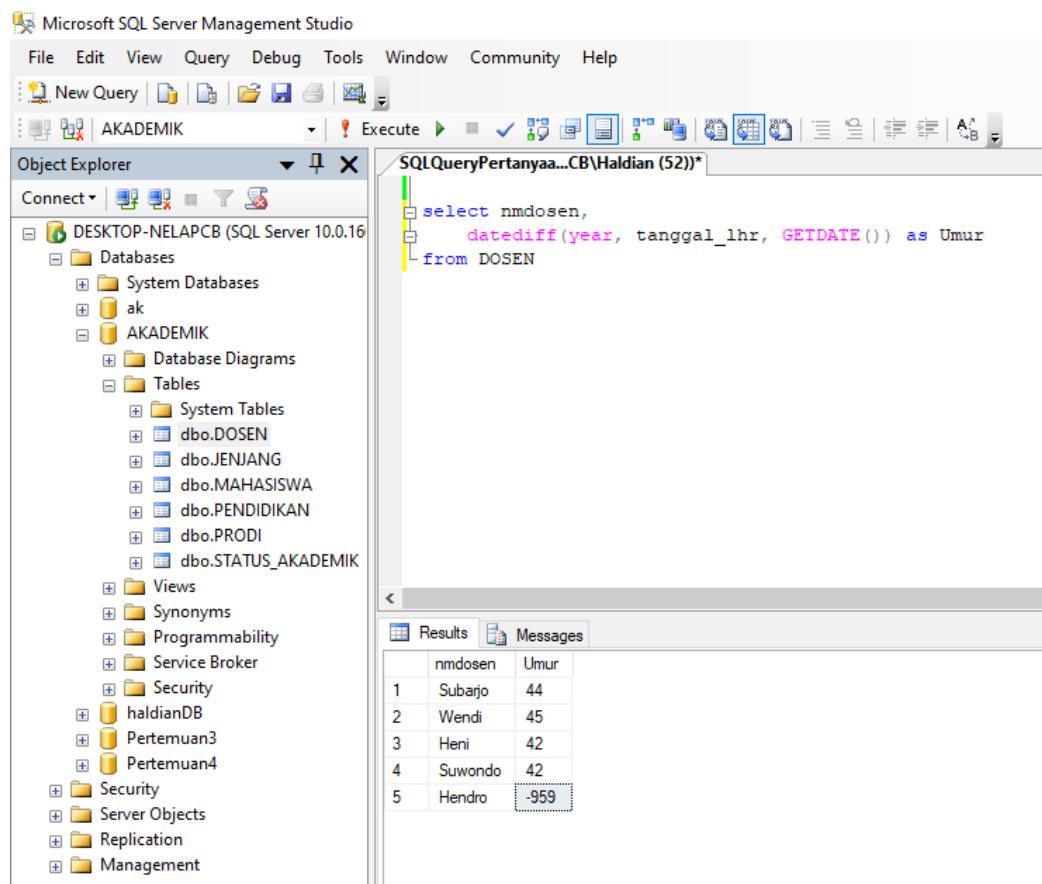
Umur Tertua	Umur Termuda
22	20

- Jumlah karakter masing-masing nama mahasiswa Program Studi D3 Akuntansi



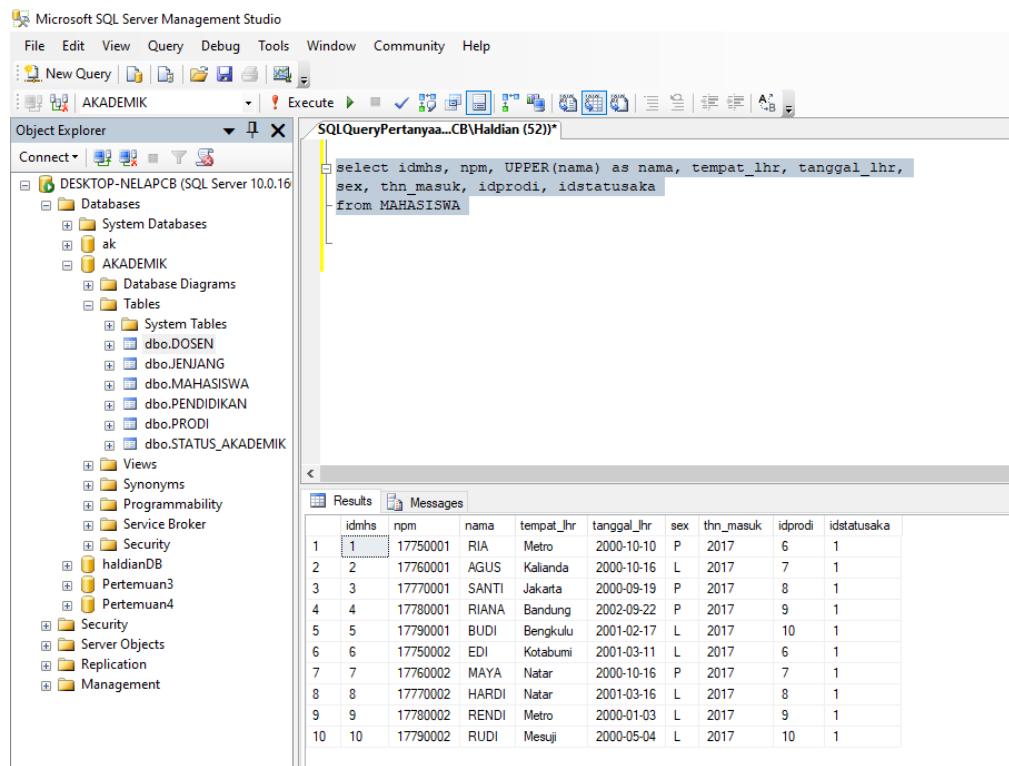
```
Microsoft SQL Server Management Studio
File Edit View Query Debug Tools Window Community Help
New Query | Execute | Object Explorer | SQLQuery1.sql - D...CB\Haldian (52)*
Object Explorer
Connect | DESKTOP-NELAPCB (SQL Server 10.0.16)
Databases
System Databases
ak
AKADEMIK
Database Diagrams
Tables
System Tables
dbo.DOSEN
dbo.JENJANG
dbo.MAHASISWA
dbo.PENDIDIKAN
dbo.PRODI
dbo.STATUS_AKADEMIK
Views
Synonyms
Programmability
Service Broker
Security
haldianDB
Pertemuan3
Pertemuan4
Security
Server Objects
Replication
Management
SQLQuery1.sql - D...CB\Haldian (52)*
select idmhs, npm, nama, len(nama) as [Jumlah Karakter]
from MAHASISWA where idprodi = 2
Results Messages
idmhs npm nama Jumlah Karakter
```

- Daftar Dosen dengan format kolom: ‘nmdosen’ dan ‘umur’



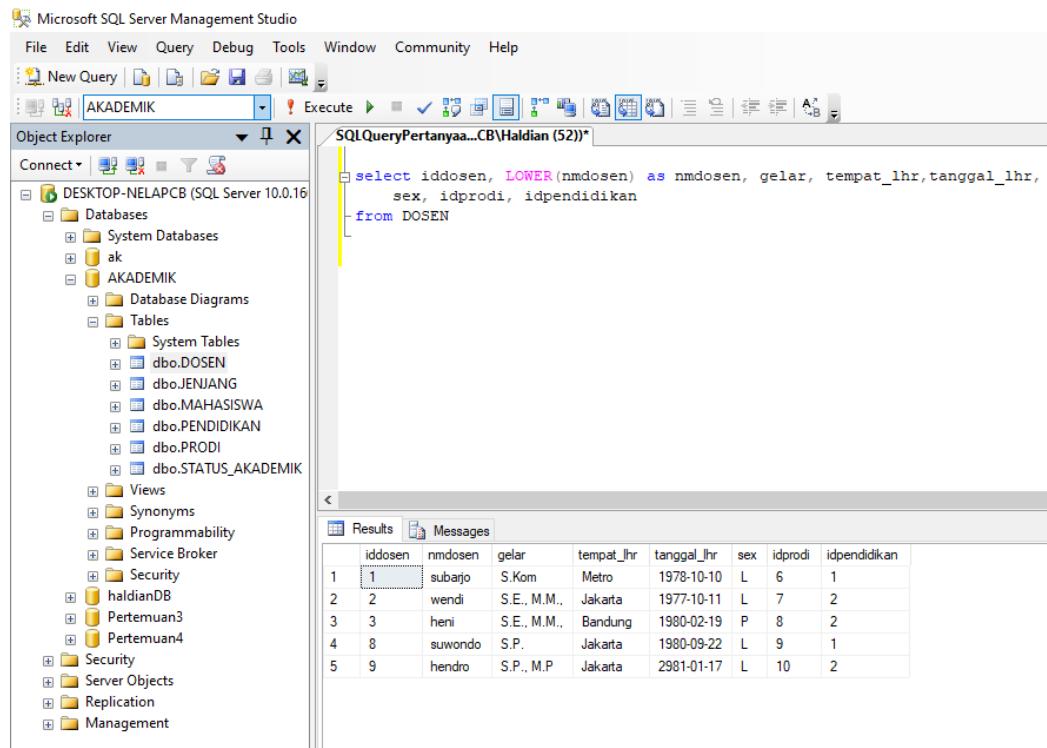
```
Microsoft SQL Server Management Studio
File Edit View Query Debug Tools Window Community Help
New Query | Execute | Object Explorer | SQLQueryPertanya...CB\Haldian (52)*
Object Explorer
Connect | DESKTOP-NELAPCB (SQL Server 10.0.16)
Databases
System Databases
ak
AKADEMIK
Database Diagrams
Tables
System Tables
dbo.DOSEN
dbo.JENJANG
dbo.MAHASISWA
dbo.PENDIDIKAN
dbo.PRODI
dbo.STATUS_AKADEMIK
Views
Synonyms
Programmability
Service Broker
Security
haldianDB
Pertemuan3
Pertemuan4
Security
Server Objects
Replication
Management
SQLQueryPertanya...CB\Haldian (52)*
select nmdosen,
       datediff(year, tanggal_lhr, GETDATE()) as Umur
from DOSEN
Results Messages
nmdosen Umur
1 Subajo 44
2 Wendi 45
3 Heni 42
4 Suwondo 42
5 Hendro -959
```

- Daftar Mahasiswa; dimana nama mahasiswa disajikan dengan Huruf Kapital



```
Microsoft SQL Server Management Studio
File Edit View Query Debug Tools Window Community Help
New Query Execute
Object Explorer
Connect AKADEMIK
DESKTOP-NELAPCB (SQL Server 10.0.16)
Databases
System Databases
ak
AKADEMIK
Database Diagrams
Tables
System Tables
dbo.DOSEN
dbo.JENJANG
dbo.MAHASISWA
dbo.PENDIDIKAN
dbo.PRODI
dbo.STATUS_AKADEMIK
Views
Synonyms
Programmability
Service Broker
Security
haldianDB
Pertemuan3
Pertemuan4
Security
Server Objects
Replication
Management
SQLQueryPertanyaan...CB\Haldian (52)*
select idmhs, npm, UPPER(nome) as nome, tempat_lhr, tanggal_lhr,
sex, thn_masuk, idprodi, idstatusaka
from MAHASISWA
Results Messages
idmhs npm nome tempat_lhr tanggal_lhr sex thn_masuk idprodi idstatusaka
1 17750001 RIA Metro 2000-10-10 P 2017 6 1
2 17760001 AGUS Kalianda 2000-10-16 L 2017 7 1
3 17770001 SANTI Jakarta 2000-09-19 P 2017 8 1
4 17780001 RIANA Bandung 2002-09-22 P 2017 9 1
5 17790001 BUDI Bengkulu 2001-02-17 L 2017 10 1
6 17750002 EDI Kotabumi 2001-03-11 L 2017 6 1
7 17760002 MAYA Natar 2000-10-16 P 2017 7 1
8 17770002 HARDI Natar 2001-03-16 L 2017 8 1
9 17780002 RENDI Metro 2000-01-03 L 2017 9 1
10 17790002 RUDI Mesuji 2000-05-04 L 2017 10 1
```

- Daftar Dosen; dimana nama dosen disajikan dengan huruf kecil



```
Microsoft SQL Server Management Studio
File Edit View Query Debug Tools Window Community Help
New Query Execute
Object Explorer
Connect AKADEMIK
DESKTOP-NELAPCB (SQL Server 10.0.16)
Databases
System Databases
ak
AKADEMIK
Database Diagrams
Tables
System Tables
dbo.DOSSEN
dbo.JENJANG
dbo.MAHASISWA
dbo.PENDIDIKAN
dbo.PRODI
dbo.STATUS_AKADEMIK
Views
Synonyms
Programmability
Service Broker
Security
haldianDB
Pertemuan3
Pertemuan4
Security
Server Objects
Replication
Management
SQLQueryPertanyaan...CB\Haldian (52)*
select iddosen, LOWER(nmdosen) as nmdosen, gelar, tempat_lhr, tanggal_lhr,
sex, idprodi, idpendidikan
from DOSEN
Results Messages
iddosen nmdosen gelar tempat_lhr tanggal_lhr sex idprodi idpendidikan
1 subarjo S.Kom Metro 1978-10-10 L 6 1
2 wendi S.E., M.M. Jakarta 1977-10-11 L 7 2
3 henri S.E., M.M. Bandung 1980-02-19 P 8 2
4 suwondo S.P. Jakarta 1980-09-22 L 9 1
5 hendro S.P., M.P Jakarta 2981-01-17 L 10 2
```

- Contoh penggunaan sintak perintah SUM dan ROUND pada studi kasus Database “AKADEMIK”
 - ❖ Menampilkan jumlah prodi dan jumlah jenjang dengan sum seperti gambar di bawah ini

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure for 'DESKTOP-NELAPCB (SQL Server 10)'. The 'AKADEMIK' database is expanded, showing tables like DOSEN, JENJANG, MAHASISWA, PENDIDIKAN, PRODI, and STATUS_AKADEMIK. The 'Tables' node under AKADEMIK is also expanded. The 'Results' tab in the bottom pane displays the output of a query:

```
select COUNT(*) as [Jumlah Prodi],
       SUM(idjenjang) as [Jumlah Jenjang]
  from PRODI
```

	Jumlah Prodi	Jumlah Jenjang
1	5	7

- ❖ Menampilkan pembulatan dari jumlah DOSEN dengan operator pembagi Seperti gambar di bawah ini

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure for 'DESKTOP-NELAPCB (SQL Server 10)'. The 'AKADEMIK' database is expanded, showing tables like DOSEN, JENJANG, MAHASISWA, PENDIDIKAN, PRODI, and STATUS_AKADEMIK. The 'Tables' node under AKADEMIK is also expanded. The 'Results' tab in the bottom pane displays the output of a query:

```
select COUNT(*) as DOSEN, 5 as Pembagi,
       round(cast(count(*) as dec (5,1))/5,1) as hasil
  from DOSEN
```

DOSEN	Pembagi	hasil
1	5	1.000000

BAB 4. PENUTUP

A. KESIMPULAN

SQL (Structured Query Language) merupakan suatu sintaks atau perintah yang di gunakan untuk mengakses data di dalam sebuah *database*, SQL yang digunakan untuk memanipulasi sebuah data dalam database seperti untuk menambah data (*insert*), mengubah data (*update*), menampilkan data (*select*), dan juga menghapus data (*delete*). Secara umum, SQL terdiri dari dua bahasa, yaitu *Data Definition Language* (DDL) dan *Data Manipulation Language* (DML). Implementasi DDL dan DML berbeda untuk tiap sistem manajemen basis data (SMBD), namun secara umum implementasi tiap bahasa ini memiliki bentuk standar yang ditetapkan masing-masing bahasa .

B. SARAN

Setelah melakukan praktikum *Transact-SQL* tentang *SELECT*, *SUBQUERY*, dan *JOIN* di harapkan mahasiswa dapat mengimplementasikannya dalam project-project SQL dan mahasiswa dapat melanjutkan materi selanjutnya mengenai SQL Lanjut.

PUSTAKA

Tutorials Point, Microsoft SQL Server, Tutorials Point Ltd, USA, 2003

Ken Henderson, The Guru's Guide to Transact-SQL, USA, AddisonWesley, 2000

Jan L. Harrington, SQL Clearly Explained, Third Edition, USA, Morgan Kaufmann Publishers, 2010

Eko Win Kenali, Pemrograman SQL menggunakan DBMS Microsoft SQL Server, UP Politeknik Negeri Lampung, 201

SQL Tersedia di: <https://www.w3schools.com/sql/> [Diakses 5 April 2022].

Setiadi, M.F. (2017). Data Manipulation Language Pada MySQL. [online] mfikri. Tersedia di: <https://mfikri.com/artikel/Data-Manipulation-Language.html> [Diakses 5 April 2022].