

# **LAPORAN AKHIR**

## **PERTEMUAN 1**



ADMINISTRASI SISTEM APLIKASI DAN DATABASE PADA SQL  
SERVER 2008

Disusun Oleh:

Nama : Elmo Allistair

NPM : 12118220

Kelas : 4KA17

*Kelompok : (Opsional)*

**LEMBAGA PENGEMBANGAN KOMPUTERISASI  
UNIVERSITAS GUNADARMA**

**2021**

## **Tujuan Aktivitas :**

**1. Mengetahui tahap instalasi SQL Server 2008**

**2. Mengetahui objek-objek SQL Server 2008**

**3. Mengetahui Pembuatan Database menggunakan SQL Server 2008**

## **TAHAPAN Pengerjaan**

### **1. Ringkasan Materi**

❖ Terdapat dua perbedaan utama antara SQL Server/Sybase dan Oracle adalah:

- 1) SQL Server tidak memiliki segmen undo atau rollback dan oleh karena itu tidak ada fungsi rollback/commit, atau membaca tampilan data yang konsisten, selain yang sedikit bisa ditampung dari on line redo logs.
- 2) Arsitektur SQL Server adalah: 1 mesin host ke 1 atau lebih, contohnya untuk banyak Database.

❖ SQL Server sebagian besar dikelola dari SQL Server Management Studio, pengecualiannya adalah komponen opsional yang dikelola dari Studio Pengembangan Intelijen Bisnis SQL Server.

❖ SQL Server memiliki komponen opsional, yaitu:

- 1) **Pelaporan SQL Server Services**, Business Intelligence reporting, alternatif Microsoft untuk OBIEE, Discoverer, Crystal Reports. Memiliki integrasi yang sangat baik dengan produk klien Microsoft seperti Excel yang signifikan bagi kegunaan pengguna akhir.
- 2) **Analysis Services**, OLAP (on line analytical processing), cubes, data mining.
- 3) **SQL Server Integration Services**, ETL (extract transform load) dan ekspor / impor tool. Sama dengan ekspor/impor, data pump, sql loader, skrip shell DBA. Ini disebut DTS (Data Transformation Services) di versi lama.

❖ Edisi SQL Server 2008:

- 1) **Express Edition**, versi ringan dari SQL Server 2008. produk ini mencakup program Express Manager (XM) dan mendukung integrasi CLR dan XML
- 2) **Workgroup Edition**, dirancang untuk usaha kecil dan juga digunakan di tingkat departemen. Edisi ini menyediakan dukungan database relasional tanpa intelijen bisnis (BI) dan kemampuan ketersediaan tinggi.
- 3) **Standard Edition**, memberikan versi 32-bit dan dukungan 64-bit untuk kedua x64- dan sistem berbasis itanium, juga mendukung hingga empat prosesor dan termasuk full berbagai fungsi BI.
- 4) **Enterprise Edition**, edisi ini berisi fitur tambahan yang berguna untuk pemasangan yang sangat mahal dengan multiprocessors simetris atau cluster. Fitur tambahan yang paling penting dari Enterprise Edition adalah partisi data, snapshot database, dan pemeliharaan database online.
- 5) **Developer Edition**, memungkinkan pengembang untuk membuat dan menguji semua jenis aplikasi dengan SQL Server pada platform 32 dan 64-bit.

❖ SQL Server Edisi Express memiliki sejumlah batasan teknis yang membuatnya tidak digunakan untuk penerapan data skala besar, yaitu:

- 1) Ukuran database maksimum 4 GB per database (versi 2005) atau 10 GB (versi 2008 dan 2008 R2).
- 2) Memiliki CPU fisik tunggal, beberapa core dan 1 GB RAM (berjalan pada berbagai ukuran sistem RAM, tapi hanya menggunakan 1 GB)
- 3) Tidak adanya layanan SQL Server Agent

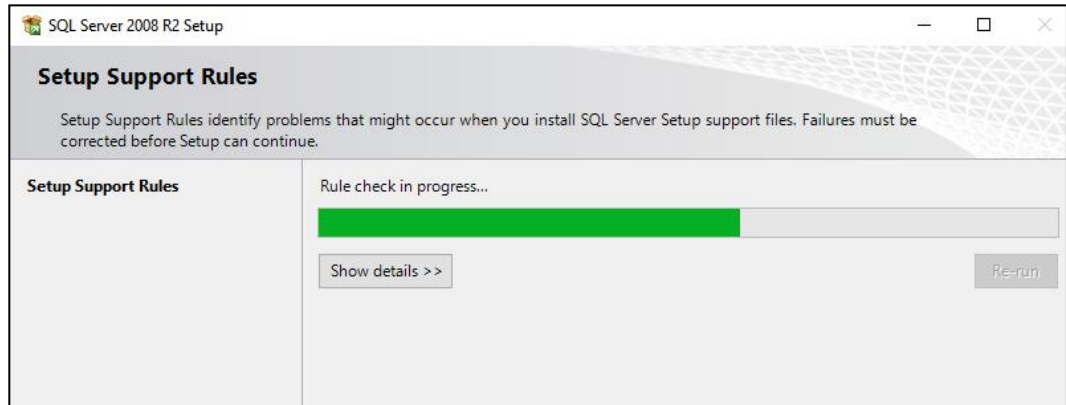
❖ Terdapat beberapa komponen kunci dalam SQL Server, yaitu:

- 1) **Database**, merupakan sekumpulan data yang terorganisir. Terdiri dari satu atau lebih tabel.
- 2) **Tabel**, merupakan kumpulan dari baris/record dan kolom/field.
- 3) **Database Diagram**, adalah representasi grafik dari sebuah database.
- 4) **View**, adalah table virtual yang isinya didefinisikan oleh query database.
- 5) **Indeks**, adalah sebuah file pelengkap yang berhubungan dengan table, yang fungsinya mempercepat waktu akses ke dalam baris-baris table.
- 6) **Stored Procedure**, sebuah program yang ditulis dalam bahasa T-SQL yang disimpan di dalam database SQL Server.

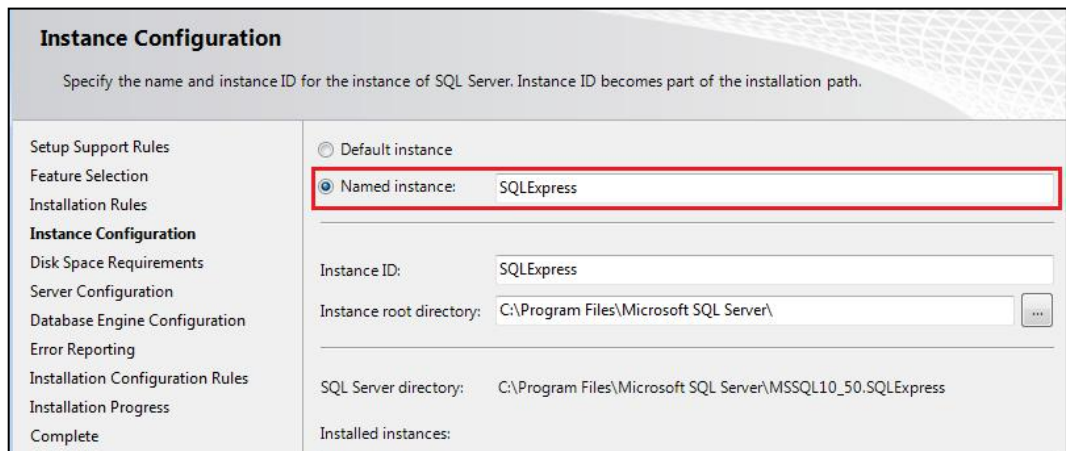
## Langkah-Langkah

*(Jabarkan Langkah-Langkah pengerjaan, pada praktikum hari ini sesuai dengan kasus yang ada)*

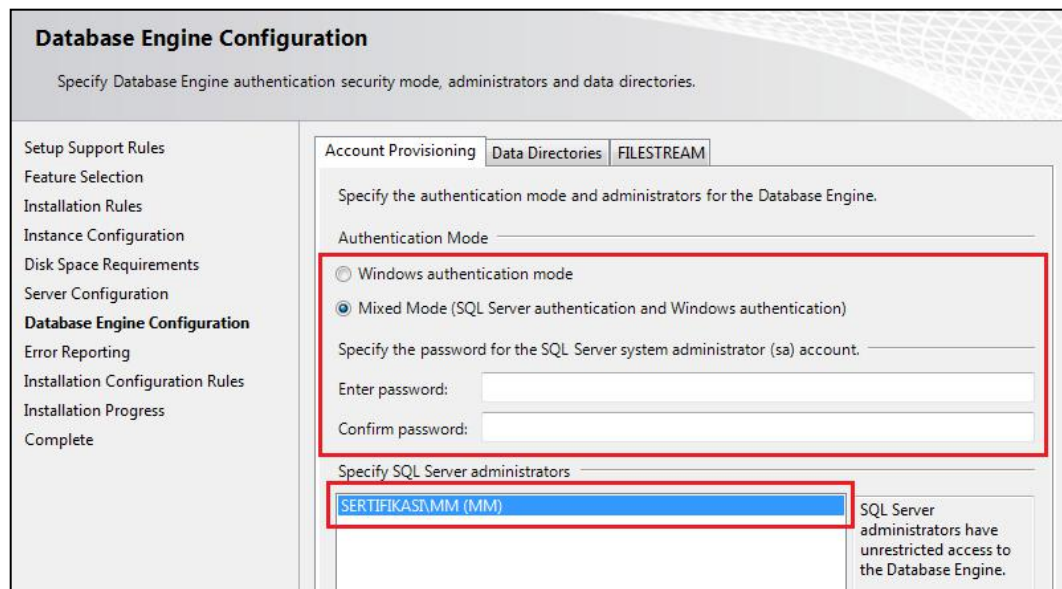
### 1. Install SMSS 2008 hingga selesai



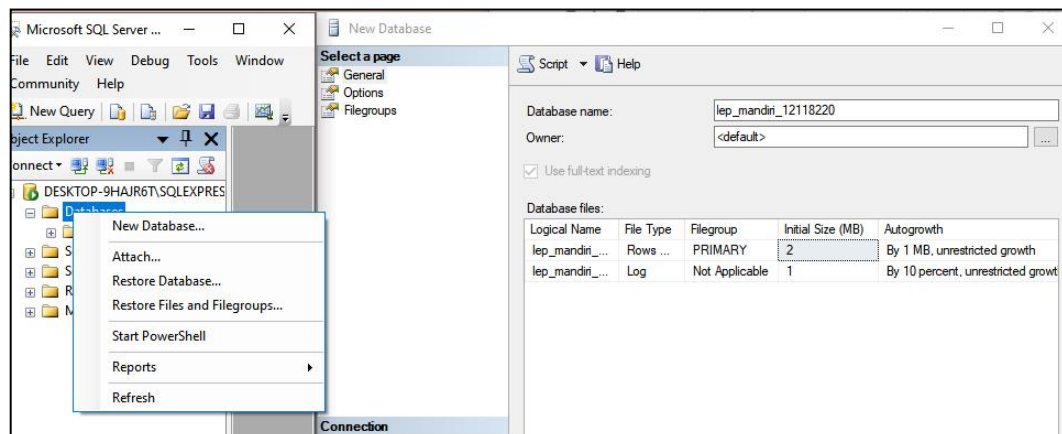
### 2. Pada **Instance Configuration**, atur “Named instance” menjadi “SQLExpress”



3. Pada **Database Engine Configuration**, pilih “Mixed Mode” lalu isikan password “lepkom123”



4. Membuat database



# **LAPORAN AKHIR**

## **PERTEMUAN 1**



### **DML (DATA MANIPULATION LANGUAGE) DAN JOIN TABLE**

Disusun Oleh:

Nama : Elmo Allistair

NPM : 12118220

Kelas : 4KA17

*Kelompok : (Opsional)*

**LEMBAGA PENGEMBANGAN KOMPUTERISASI  
UNIVERSITAS GUNADARMA**

**2021**

## **Tujuan Aktivitas :**

- 1. Mengetahui tahap instalasi SQL Server 2008**
- 2. Mengetahui objek-objek SQL Server 2008**
- 3. Mengetahui Pembuatan Database menggunakan SQL Server 2008**

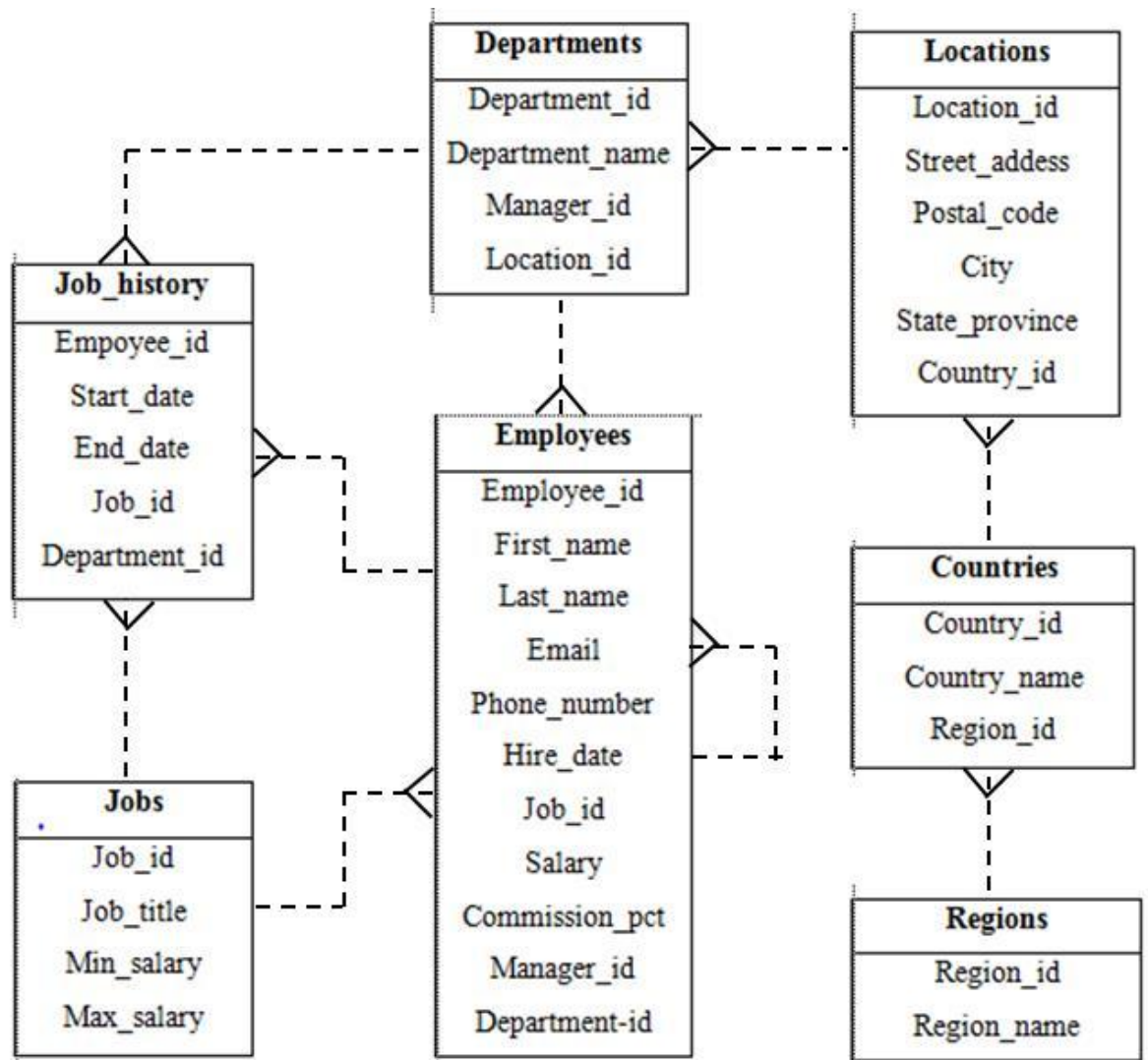
## **TAHAPAN Pengerjaan**

### **1. Ringkasan Materi**

- ❖ DML (Data Manipulation Language) digunakan untuk memanipulasi data yang ada dalam suatu tabel. Perintah yang umum dilakukan adalah:
  - 1) SELECT untuk menampilkan data
  - 2) INSERT untuk menambahkan data baru
  - 3) UPDATE untuk mengubah data yang sudah ada
  - 4) DELETE untuk menghapus data
- ❖ SQL Server menyediakan banyak cara untuk bekerja dengan tabel. Membuat tabel baru dengan menggunakan fitur New Table di SQL Server Management Studio menggunakan perintah CREATE TABLE, memodifikasi tabel yang ada menggunakan fitur Modify Table di SQL Server Management Studio dengan perintah ALTER TABLE.
- ❖ NOT NULL menentukan bahwa kolom tidak dapat menerima nilai null. Namun, nilai null bukan nilai 0, tetapi nilai null adalah yang tidak diisi atau tidak terdefinisi.
- ❖ Primary Key adalah adalah satu atau lebih kolom pada tabel yang memiliki baris yang eksklusif di dalam tabel akan membentuk primary key. Spesifikasi dari primary key memastikan integritas dari tabel. Kolom yang membentuk primary key tidak bisa mengandung nilai null.
- ❖ Foreign Key adalah field di table aktif yang menunjukan ke field kunci pada tabel lain. Penggunaan foreign key mampu mencegah penghapusan baris-baris pada tabel yang aktif, apabila ada referensi field-field kunci dari tabel eksternal.

## 2. Langkah-Langkah

(Jabarkan Langkah-Langkah pengerjaan, pada praktikum hari ini sesuai dengan kasus yang ada)





- 1) Buatlah tabel berdasarkan skema diatas, lengkap dengan primary key dan foregin keynya sehingga terbentuk relasi antar tabel

```
SQLQuery3.sql - ...6TX441UV (53))*
-- No. 1
CREATE TABLE regions (
    region_id    NUMERIC CONSTRAINT region_id_nn NOT NULL,
    region_name  VARCHAR(25)
);

CREATE TABLE countries (
    country_id   CHAR(2)  CONSTRAINT country_id_nn NOT NULL,
    country_name VARCHAR(40),
    region_id    NUMERIC CONSTRAINT country_c_id_pk PRIMARY KEY (country_id)
);

CREATE TABLE locations (
    location_id   NUMERIC(4)  CONSTRAINT location_id_nn NOT NULL,
    street_address VARCHAR(40),
    postal_code   VARCHAR(12),
    city          VARCHAR(30) CONSTRAINT loc_city_nn  NOT NULL,
    state_province VARCHAR(25),
    country_id    CHAR(2)
);

CREATE TABLE departments (
    department_id   NUMERIC(4)  CONSTRAINT department_id_nn NOT NULL,
    department_name VARCHAR(30) CONSTRAINT dept_name_nn  NOT NULL,
    manager_id      NUMERIC(6),
    location_id     NUMERIC(4)
);

CREATE TABLE jobs (
    job_id        VARCHAR(10),
    job_title      VARCHAR(35)  CONSTRAINT job_title_nn  NOT NULL,
    min_salary     NUMERIC(6),
    max_salary     NUMERIC(6)
);
```

```
SQLQuery3.sql - ...6TX441UV (53))*
CREATE TABLE employees (
    employee_id   NUMERIC(6),
    first_name    VARCHAR(20),
    last_name     VARCHAR(25)  CONSTRAINT emp_last_name_nn NOT NULL,
    email         VARCHAR(25)  CONSTRAINT emp_email_nn  NOT NULL,
    phone_NUMBER  VARCHAR(20),
    hire_date     DATE         CONSTRAINT emp_hire_date_nn  NOT NULL,
    job_id        VARCHAR(10)  CONSTRAINT emp_job_nn   NOT NULL,
    salary        NUMERIC(8,2),
    commission_pct NUMERIC(2,2),
    manager_id    NUMERIC(6),
    department_id NUMERIC(4),
    CONSTRAINT emp_salary_min CHECK (salary > 0),
    CONSTRAINT emp_email_uk   UNIQUE (email)
);

CREATE TABLE job_history (
    employee_id   NUMERIC(6)  CONSTRAINT jhist_employee_nn NOT NULL,
    start_date    DATE        CONSTRAINT jhist_start_date_nn NOT NULL,
    end_date      DATE        CONSTRAINT jhist_end_date_nn  NOT NULL,
    job_id        VARCHAR(10)  CONSTRAINT jhist_job_nn   NOT NULL,
    department_id NUMERIC(4),
    CONSTRAINT jhist_date_interval CHECK (end_date > start_date)
);
```

## 2) Input data yang telah disediakan.

SQLQuery3.sql - ...6TX441UV (53))\*

```
SELECT TOP 2 * FROM countries
SELECT TOP 2 * FROM departments
SELECT TOP 2 * FROM employees
SELECT TOP 2 * FROM job_history
SELECT TOP 2 * FROM jobs
SELECT TOP 2 * FROM locations
SELECT TOP 2 * FROM regions
```

Results Messages

| country_id | country_name | region_id |
|------------|--------------|-----------|
| 1 AR       | Argentina    | 2         |
| 2 AU       | Australia    | 3         |

| department_id | department_name | manager_id | location_id |
|---------------|-----------------|------------|-------------|
| 1 10          | Administration  | 200        | 1700        |
| 2 20          | Marketing       | 201        | 1800        |

| employee_id | first_name | last_name | email    | phone_NUMERIC | hire_date  | job_id  | salary   | commission_pct | manager_id | department_id |
|-------------|------------|-----------|----------|---------------|------------|---------|----------|----------------|------------|---------------|
| 1 100       | Steven     | King      | SKING    | 515.123.4567  | 1987-06-17 | AD_PRES | 24000.00 | NULL           | NULL       | 90            |
| 2 101       | Neena      | Kochhar   | NKOCHHAR | 515.123.4568  | 1989-09-21 | AD_VP   | 17000.00 | NULL           | 100        | 90            |

| employee_id | start_date | end_date   | job_id     | department_id |
|-------------|------------|------------|------------|---------------|
| 1 102       | 1993-01-13 | 1998-07-24 | IT_PROG    | 60            |
| 2 101       | 1989-09-21 | 1993-10-27 | AC_ACCOUNT | 110           |

| job_id    | job_title                     | min_salary | max_salary |
|-----------|-------------------------------|------------|------------|
| 1 AD_PRES | President                     | 20000      | 40000      |
| 2 AD_VP   | Administration Vice President | 15000      | 30000      |

| location_id | street_address          | postal_code | city   | state_province | country_id |
|-------------|-------------------------|-------------|--------|----------------|------------|
| 1 1000      | 1297 Via Cola di Rie    | 00989       | Roma   | NULL           | IT         |
| 2 1100      | 93091 Calle della Testa | 10934       | Venice | NULL           | IT         |

| region_id | region_name |
|-----------|-------------|
| 1 1       | Europe      |
| 2 2       | Americas    |

## 3) Tampilkan data karyawan yang memiliki nama berawalan huruf "M".

SQLQuery3.sql - ...6TX441UV (53))\*

```
select * from employees where first_name like 'm%'
```

Results Messages

| employee_id | first_name | last_name | email    | phone_NUMERIC      | hire_date  | job_id   | salary   | commission_pct | manager_id | department_id |
|-------------|------------|-----------|----------|--------------------|------------|----------|----------|----------------|------------|---------------|
| 1 120       | Matthew    | Weiss     | MWEISS   | 650.123.1234       | 1996-07-18 | ST_MAN   | 8000.00  | NULL           | 100        | 50            |
| 2 130       | Mozhe      | Atkinson  | MATKINSO | 650.124.6234       | 1997-10-30 | ST_CLERK | 2800.00  | NULL           | 121        | 50            |
| 3 134       | Michael    | Rogers    | MROGERS  | 650.127.1834       | 1998-08-26 | ST_CLERK | 2900.00  | NULL           | 122        | 50            |
| 4 164       | Mattea     | Marvins   | MMARVINS | 011.44.1346.329268 | 2000-01-24 | SA_REP   | 7200.00  | 0.10           | 147        | 80            |
| 5 182       | Martha     | Sullivan  | MSULLIVA | 650.507.9878       | 1999-06-21 | SH_CLERK | 2500.00  | NULL           | 120        | 50            |
| 6 201       | Michael    | Hartstein | MHARTSTE | 515.123.5555       | 1996-02-17 | MK_MAN   | 13000.00 | NULL           | 100        | 20            |

- 4) Tampilkan data karyawan yang mulai bekerja antara tanggal 20 Mei 1991 dan 17 Juni 1995

SQLQuery3.sql - ...6TX441UV (53))\*

```
select * from employees where hire_date between '1991-05-20' and '1995-06-17'
```

Results Messages

|    | employee_id | first_name | last_name | email    | phone_NUMBER | hire_date  | job_id     | salary   | commission |
|----|-------------|------------|-----------|----------|--------------|------------|------------|----------|------------|
| 2  | 104         | Bruce      | Ernst     | BERNST   | 590.423.4568 | 1991-05-21 | IT_PROG    | 6000.00  | NULL       |
| 3  | 108         | Nancy      | Greenberg | NGREENBE | 515.124.4569 | 1994-08-17 | FI_MGR     | 12000.00 | NULL       |
| 4  | 109         | Daniel     | Faviet    | DFAVIET  | 515.124.4169 | 1994-08-16 | FI_ACCOUNT | 9000.00  | NULL       |
| 5  | 114         | Den        | Raphaely  | DRAPHEAL | 515.127.4561 | 1994-12-07 | PU_MAN     | 11000.00 | NULL       |
| 6  | 115         | Alexander  | Khoo      | AKHOO    | 515.127.4562 | 1995-05-18 | PU_CLERK   | 3100.00  | NULL       |
| 7  | 122         | Payam      | Kaufling  | PKAUFLIN | 650.123.3234 | 1995-05-01 | ST_MAN     | 7900.00  | NULL       |
| 8  | 203         | Susan      | Mavris    | SMAVRIS  | 515.123.7777 | 1994-06-07 | HR_REP     | 6500.00  | NULL       |
| 9  | 204         | Hermann    | Baer      | HBAER    | 515.123.8888 | 1994-06-07 | PR_REP     | 10000.00 | NULL       |
| 10 | 205         | Shelley    | Higgins   | SHIGGINS | 515.123.8080 | 1994-06-07 | AC_MGR     | 12000.00 | NULL       |
| 11 | 206         | William    | Gietz     | WGIEZT   | 515.123.8181 | 1994-06-07 | AC_ACCOUNT | 8300.00  | NULL       |

- 5) Ganti tipe data region\_id menjadi numeric(3) pada tabel countries, kemudian jadikan sebagai foreign key.

SQLQuery3.sql - ...6TX441UV (53))\*

```
alter table regions alter column region_id numeric(3) not null;
alter table regions add constraint reg_id_pk primary key (region_id);
alter table countries alter column region_id numeric(3) not null;
alter table countries add constraint reg_id_fk foreign key (region_id) references
```

Messages

Command(s) completed successfully.

- 6) Gunakan inner join untuk menampilkan region\_name dan country\_name pada tabel regions dan countries.

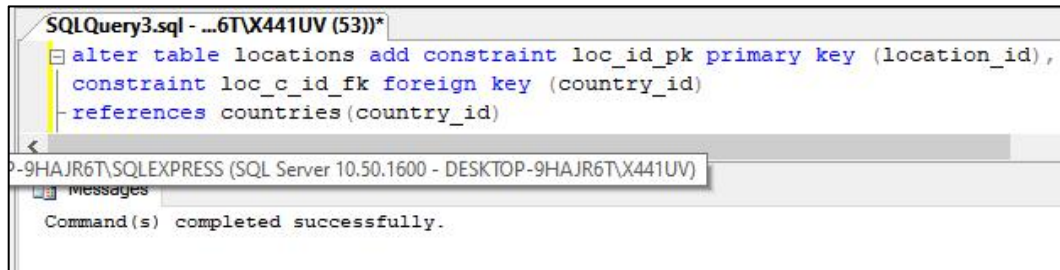
SQLQuery3.sql - ...6TX441UV (53))\*

```
select region name, country name from countries inner join regions
on regions.region_id = countries.region_id
```

Results Messages

|   | region_name | country_name |
|---|-------------|--------------|
| 1 | Americas    | Argentina    |
| 2 | Asia        | Australia    |
| 3 | Europe      | Belgium      |
| 4 | Americas    | Brazil       |
| 5 | Americas    | Canada       |
| 6 | Europe      | Switzerland  |
| 7 | Asia        | China        |
| 8 | Europe      | Germany      |

- 7) Buat primary key pada location\_id di tabel locations serta foreign key di country\_id.

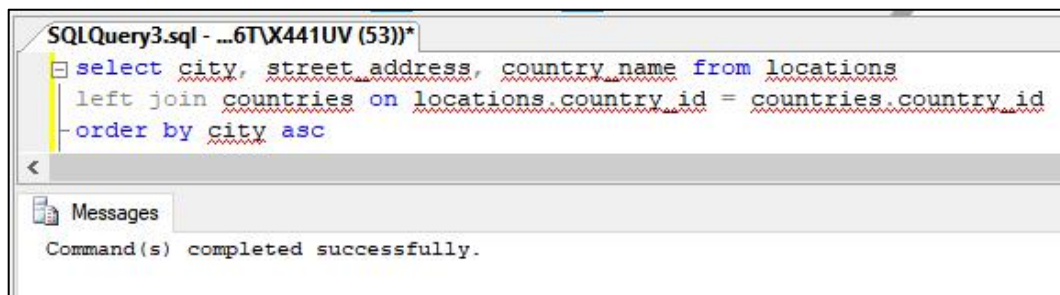


The screenshot shows a SQL query window titled "SQLQuery3.sql - ...6TX441UV (53))". The query is as follows:

```
alter table locations add constraint loc_id_pk primary key (location_id),  
constraint loc_c_id_fk foreign key (country_id)  
references countries(country_id)
```

Below the query, the "Messages" pane shows the message: "Command(s) completed successfully."

- 8) Tampilkan city dan street\_address pada tabel locations, serta country\_name pada tabel countries dengan menggunakan left outer join urutkan secara ascending berdasarkan city pada tabel locations.

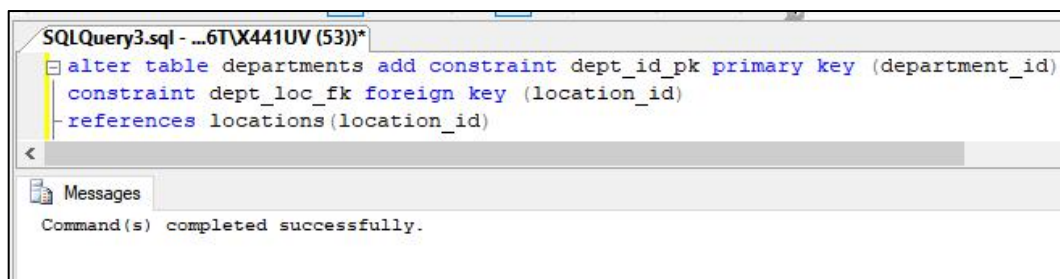


The screenshot shows a SQL query window titled "SQLQuery3.sql - ...6TX441UV (53))". The query is as follows:

```
select city, street_address, country_name from locations  
left join countries on locations.country_id = countries.country_id  
order by city asc
```

Below the query, the "Messages" pane shows the message: "Command(s) completed successfully."

- 9) Pada tabel departments, ganti tipe data locations\_id menjadi NUMERIC(4). Kemudian tambahkan primar key pada department\_id dan foegin key pada loction\_id pada tabel departments.



The screenshot shows a SQL query window titled "SQLQuery3.sql - ...6TX441UV (53))". The query is as follows:

```
alter table departments add constraint dept_id_pk primary key (department_id)  
constraint dept_loc_fk foreign key (location_id)  
references locations(location_id)
```

Below the query, the "Messages" pane shows the message: "Command(s) completed successfully."



- 10) Tampilkan `street_address` dan `postal_code` pada tabel `locations` serta `department_name` pada tabel `departments`, tapi tampilkan `postal_code` yang memiliki no 98199 saja.

SQLQuery3.sql - ...6TX441UV (53))\*

```
select locations.street_address, locations.postal_code,
departments.department_name from locations, departments
where postal_code = '98199'
```

Results Messages

|   | street_address  | postal_code | department_name |
|---|-----------------|-------------|-----------------|
| 1 | 2004 Charade Rd | 98199       | Administration  |
| 2 | 2004 Charade Rd | 98199       | Marketing       |
| 3 | 2004 Charade Rd | 98199       | Purchasing      |
| 4 | 2004 Charade Rd | 98199       | Human Resources |
| 5 | 2004 Charade Rd | 98199       | Shipping        |
| 6 | 2004 Charade Rd | 98199       | IT              |

- 11) Tampilkan `job_title`, `min salary`, `max salary` pada tabel `jobs`, Serta `first name` dan `salary` pada tabel `employee`, dengan `salary` lebih kecil sama dengan 6000

SQLQuery3.sql - ...6TX441UV (53))\*

```
select jobs.job_title, jobs.min_salary, jobs.max_salary,
employees.first_name, employees.salary
from jobs, employees
where salary <= 6000
```

Results Messages

|   | job_title | min_salary | max_salary | first_name | salary   |
|---|-----------|------------|------------|------------|----------|
| 1 | President | 20000      | 40000      | Steven     | 24000.00 |
| 2 | President | 20000      | 40000      | Neena      | 17000.00 |
| 3 | President | 20000      | 40000      | Lex        | 17000.00 |
| 4 | President | 20000      | 40000      | Alexander  | 9000.00  |
| 5 | President | 20000      | 40000      | Bruce      | 6000.00  |
| 6 | President | 20000      | 40000      | David      | 4800.00  |

- 12) Tampilkan nama belakang, gaji bulanan pegawai dan kode departemen tempat pegawai tersebut bekerja khusus untuk pegawai yang bekerja pada departemen dengan kode 100 atau 101, dan memiliki gaji lebih besar atau sama dengan 1000.

SQLQuery3.sql - ...6TX441UV (53))\*

```
select last_name, salary, department_id from employees
where department_id between '100' and '101'
and salary >= 1000
```

Results Messages

|   | last_name | salary   | department_id |
|---|-----------|----------|---------------|
| 1 | Greenberg | 12000.00 | 100           |
| 2 | Faviet    | 9000.00  | 100           |
| 3 | Chen      | 8200.00  | 100           |
| 4 | Sciarra   | 7700.00  | 100           |
| 5 | Urman     | 7800.00  | 100           |
| 6 | Popp      | 6900.00  | 100           |

- 13) Tampilkan job title, min salary pada tabel jobs dan first name, salary pada tabel employees, serta department name pada tabel departments yang memiliki first name berawalan huruf “S”.

SQLQuery3.sql - ...6TX441UV (53))\*

```
select jobs.job_id, jobs.min_salary,
employees.first_name, employees.salary,
departments.department_name from jobs, employees, departments
where employees.first_name like 'S%'
```

Results Messages

|   | job_id  | min_salary | first_name | salary   | department_name  |
|---|---------|------------|------------|----------|------------------|
| 1 | AD_PRES | 20000      | Steven     | 24000.00 | Administration   |
| 2 | AD_PRES | 20000      | Steven     | 24000.00 | Marketing        |
| 3 | AD_PRES | 20000      | Steven     | 24000.00 | Purchasing       |
| 4 | AD_PRES | 20000      | Steven     | 24000.00 | Human Resources  |
| 5 | AD_PRES | 20000      | Steven     | 24000.00 | Shipping         |
| 6 | AD_PRES | 20000      | Steven     | 24000.00 | IT               |
| 7 | AD_PRES | 20000      | Steven     | 24000.00 | Public Relations |

- 14) Tampilkan last name, hire date, commission pct pada tabel employee, department name pada tabel department, serta city, state province dan country name pada tabel countries. Tampilkan commission pct NOT NULL.

SQLQuery3.sql - DE...(53)) Executing...\*

```
select employees.last_name, employees.hire_date, employees.commission_pct,  
departments.department_name, locations.city, locations.state_province,  
countries.country_name from employees, departments, locations, countries  
where employees.commission_pct is not null
```

Results Messages

|   | last_name | hire_date  | commission_pct | department_name | city | state_province | country_name |
|---|-----------|------------|----------------|-----------------|------|----------------|--------------|
| 1 | Russell   | 1996-10-01 | 0.40           | Administration  | Roma | NULL           | Argentina    |
| 2 | Partners  | 1997-01-05 | 0.30           | Administration  | Roma | NULL           | Argentina    |
| 3 | Errazuriz | 1997-03-10 | 0.30           | Administration  | Roma | NULL           | Argentina    |
| 4 | Cambrault | 1999-10-15 | 0.30           | Administration  | Roma | NULL           | Argentina    |
| 5 | Zlotkey   | 2000-01-29 | 0.20           | Administration  | Roma | NULL           | Argentina    |
| 6 | Tucker    | 1997-01-30 | 0.30           | Administration  | Roma | NULL           | Argentina    |
| 7 | Bernstein | 1997-03-24 | 0.25           | Administration  | Roma | NULL           | Argentina    |

# **LAPORAN AKHIR**

## **PERTEMUAN 3**



(STORED PROCEDURE DAN FUNCTION)

Disusun Oleh:

Nama : Elmo Allistair

NPM : 12118220

Kelas : 4KA17

*Kelompok : (Opsional)*

**LEMBAGA PENGEMBANGAN KOMPUTERISASI**

**UNIVERSITAS GUNADARMA**

**2021**



**PERTEMUAN : 3(Tiga)**

**Tujuan Aktivitas :**

- 1. Memahami penggunaan serta perbedaan stored procedure dan function**
- 2. Memahami jenis-jenis stored procedure dan function**
- 3. Dapat membuat, mengubah, serta menghapus stored procedure dan function**

## **TAHAPAN Pengerjaan**

### **1. Ringkasan Materi**

- ❖ Stored procedure dalam SQL Server adalah kumpulan dari satu atau lebih pernyataan Transact-SQL atau suatu referensi ke metode Common Runtime Language (CLR) Framework Microsoft .NET.
- ❖ Procedure dapat dikatakan seperti construct pada bahasa pemrograman lainnya karena procedure dapat melakukan beberapa hal seperti berikut.
  - 1) Menerima parameter input dan mengembalikan banyak nilai dalam bentuk parameter output ke program yang dipanggil.
  - 2) Mengandung pernyataan pemrograman yang dapat melakukan berbagai operasi dalam database, termasuk memanggil procedure lainnya.
  - 3) Mengembalikan nilai status ke program yang dipanggil untuk mengindikasikan kesuksesan atau kegagalannya, beserta penyebab kegagalan tersebut,
- ❖ Jenis- Jenis Stored Procedure adalah:
  - 1) User-Defined, dapat dibuat dalam database yang dibuat oleh user atau dalam semua database system kecuali database Resource.
  - 2) Temporary, suatu bentuk dari procedure user-defined. Procedure ini mirip seperti procedure permanen, namun temporary procedure disimpan dalam database sementara.

- 3) System, Procedure System adalah procedure bawaan dari SQL Server. Procedure ini disimpan secara fisik dalam database internal Recourse yang tersembunyi, dan muncul secara logikal pada schema sys dari semua database, baik database system maupun database user.
- 4) Extended User-Defined, Procedure extended memungkinkan untuk membuat rutin eksternal dalam suatu bahasa pemrograman seperti bahasa C. Procedure ini adalah DLL yang dapat dijalankan dan dimuat secara dinamis oleh SQL Server.

❖ Manfaat Menggunakan Stored Procedure antara lain:

- 1) Mengurangi Traffic Jaringan Server/Client, perintah pada suatu procedure dieksekusi sebagai satu batch kode. Hal ini dapat mengurangi traffic jaringan secara signifikan antara server dan klien karena hanya panggilan untuk mengeksekusi procedure yang dikirim melalui jaringan.
- 2) Memperkuat Keamanan, banyak user dan program klien dapat melakukan operasi pada objek-objek database pokok melalui suatu procedure, meskipun user dan program tersebut tidak memiliki hak akses langsung pada objek-objek pokok tersebut Procedure tersebut mengontrol proses dan aktifitas apa yang dapat dilakukan serta tetap melindungi objek-objek database pokok tersebut.
- 3) Penggunaan Kode Secara Berulang, kode untuk setiap operasi database yang berulang adalah kandidat sempurna untuk enkapsulasi pada procedure. Hal ini dapat mengurangi kebutuhan untuk menulis ulang kode yang sama, mengurangi ketidakkonsistenan kode yang dibuat.
- 4) Mempermudah Pemeliharaan, ketika aplikasi klien memanggil procedure dan menyimpan operasi database pada tingkat data, hanya procedure yang harus di-update untuk setiap perubahan pada database pokok.
- 5) Meningkatkan Performance, secara default, suatu procedure telah ter-compile pada saat pertama kali dieksekusi dan membuat suatu rencana eksekusi yang digunakan kembali untuk eksekusi selanjutnya.

## 2. Langkah-Langkah

### Membuat multi-statement table-valued

```
SQLQuery1.sql - ...6TX441UV (52))*
use lep_mandiri_12118220;

create PROC WHAT_DB_is_that @ID int
as select DB_NAME (@ID) as THATDB
```

Messages  
Command(s) completed successfully.

### Membuat function yang akan mengembalikan nilai berupa table.

```
SQLQuery1.sql - ...6TX441UV (52))*
CREATE FUNCTION TotalTable ()
RETURNS int AS
BEGIN
    DECLARE @TableCount AS int;
    SELECT @TableCount = COUNT (TABLE_NAME)
    FROM INFORMATION_SCHEMA.TABLES WHERE
    TABLE_TYPE='BASE TABLE';
    RETURN @TableCount;
END
```

Messages  
Command(s) completed successfully.

### Melihat jumlah tabel dalam dbo.TotalTable

```
SQLQuery1.sql - ...6TX441UV (52))*
SELECT dbo.TotalTable ();
```

Results Messages

|   | (No column name) |
|---|------------------|
| 1 | 7                |

## Memasukkan data untuk function tunjangan

```
SQLQuery1.sql - ...6TX441UV (52))*
CREATE FUNCTION TunjanganHariRaya ( @Tgl_Hari_Raya date)
RETURNS @TunjanganHariRaya TABLE (
EmployeeID numeric (6),
First_Name varchar (20),
Last_Name varchar (25),
Gaji Numeric (8,2),
Masa_Kerja Numeric (5,2),
THR Numeric (8,2)
)
AS
BEGIN
    DECLARE @MasaKerja TABLE (masakerja numeric(5,2), emp_
    DECLARE @LoopCounter INT , @MaxEmployeeId INT, @MasaKe
    SELECT @LoopCounter = min(employee_id) , @MaxEmployeeId
    INSERT INTO @MasaKerja
    SELECT DATEDIFF(year,hire_date,@Tgl_Hari_Raya), employ
    WHILE (@LoopCounter IS NOT NULL AND @LoopCounter <= @Ma
    BEGIN
```

Messages  
Command(s) completed successfully.

## Melihat data tunjangan pada tahun 1980,2000, dan 2021.

SQLQuery1.sql - ...6TX441UV (52))\*

```
SELECT TOP (5) * from employees;
SELECT TOP (5) * from tunjanganhariraya ('1980');
SELECT TOP (5) * from tunjanganhariraya ('2000');
SELECT TOP (5) * from tunjanganhariraya ('2021');
```

|   | employee_id | first_name | last_name | email    | phone_NUMERIC | hire_date  | job_id  | salary   | commission_pct | manager_id | department_id |
|---|-------------|------------|-----------|----------|---------------|------------|---------|----------|----------------|------------|---------------|
| 1 | 100         | Steven     | King      | SKING    | 515.123.4567  | 1987-06-17 | AD_PRES | 24000.00 | NULL           | NULL       | 90            |
| 2 | 101         | Neena      | Kochhar   | NKOCHHAR | 515.123.4568  | 1989-09-21 | AD_VP   | 17000.00 | NULL           | 100        | 90            |
| 3 | 102         | Lex        | De Haan   | LDEHAAN  | 515.123.4569  | 1993-01-13 | AD_VP   | 17000.00 | NULL           | 100        | 90            |
| 4 | 103         | Alexander  | Hunold    | AHUNOLD  | 590.423.4567  | 1990-01-03 | IT_PROG | 9000.00  | NULL           | 102        | 60            |
| 5 | 104         | Bruce      | Ernst     | BERNST   | 590.423.4568  | 1991-05-21 | IT_PROG | 6000.00  | NULL           | 103        | 60            |

|   | EmployeeID | First_Name | Last_Name | Gaji     | Masa_Kerja | THR      |
|---|------------|------------|-----------|----------|------------|----------|
| 1 | 100        | Steven     | King      | 24000.00 | -7.00      | 18000.00 |
| 2 | 101        | Neena      | Kochhar   | 17000.00 | -9.00      | 12750.00 |
| 3 | 102        | Lex        | De Haan   | 17000.00 | -13.00     | 12750.00 |
| 4 | 103        | Alexander  | Hunold    | 9000.00  | -10.00     | 6750.00  |
| 5 | 104        | Bruce      | Ernst     | 6000.00  | -11.00     | 4500.00  |

|   | EmployeeID | First_Name | Last_Name | Gaji     | Masa_Kerja | THR      |
|---|------------|------------|-----------|----------|------------|----------|
| 1 | 100        | Steven     | King      | 24000.00 | 13.00      | 36000.00 |
| 2 | 101        | Neena      | Kochhar   | 17000.00 | 11.00      | 25500.00 |
| 3 | 102        | Lex        | De Haan   | 17000.00 | 7.00       | 22100.00 |
| 4 | 103        | Alexander  | Hunold    | 9000.00  | 10.00      | 12600.00 |
| 5 | 104        | Bruce      | Ernst     | 6000.00  | 9.00       | 8400.00  |

|   | EmployeeID | First_Name | Last_Name | Gaji     | Masa_Kerja | THR      |
|---|------------|------------|-----------|----------|------------|----------|
| 1 | 100        | Steven     | King      | 24000.00 | 34.00      | 36000.00 |
| 2 | 101        | Neena      | Kochhar   | 17000.00 | 32.00      | 25500.00 |
| 3 | 102        | Lex        | De Haan   | 17000.00 | 28.00      | 25500.00 |
| 4 | 103        | Alexander  | Hunold    | 9000.00  | 31.00      | 13500.00 |
| 5 | 104        | Bruce      | Ernst     | 6000.00  | 30.00      | 9000.00  |

# **LAPORAN AKHIR**

## **PERTEMUAN 4**



(LOGIN – USER – ROLE - PERMISSION)

Disusun Oleh:

Nama : Elmo Allistair

NPM : 12118220

Kelas : 4KA17

*Kelompok : (Opsional)*

**LEMBAGA PENGEMBANGAN KOMPUTERISASI**

**UNIVERSITAS GUNADARMA**

**2021**

**PERTEMUAN : 4(Empat)**

**Tujuan Aktivitas :**

- 1. Dapat membuat login, user, role dan permission**
- 2. Dapat memodifikasi login, user, role dan permission**
- 3. Mampu mempraktekan dan menerapkan pada basis data**

## **TAHAPAN Pengerjaan**

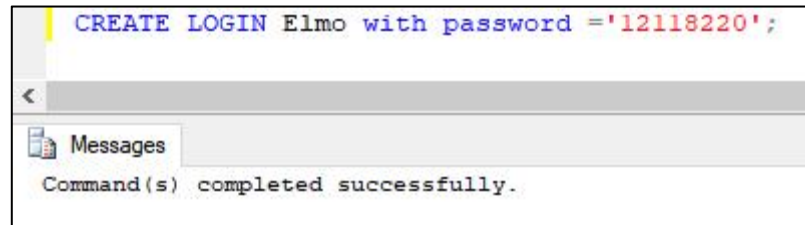
### **1. Ringkasan Materi**

- ❖ Login merupakan kepala dari suatu sistem keamanan atau sebuah entitas yang dapat di konfirmasi oleh sistem keamanan. Setiap pengguna harus melakukan login untuk bisa terkoneksi ke dalam SQL server.
- ❖ Ada tiga statement dalam Transact-SQL (T-SQL) yang digunakan pada proses pembuatan login, yaitu :
  - 1) CREATE LOGIN : membuat login
  - 2) ALTER LOGIN : memodifikasi login
  - 3) DROP LOGIN : menghapus login
- ❖ User adalah bagian dari database level security. Login harus terhubung dengan database user untuk saling terhubung. Login dapat terhubung dengan database yang berbeda sebagai pengguna yang berbeda tapi bisa juga hanya terhubung dengan satu pengguna pada masing- masing database.
- ❖ Role digunakan untuk mengelola permission secara efisien. Memberikan permission untuk role, kemudian menambah dan menghapus user serta login untuk role. Dengan menggunakan role, permission tidak harus di atur secara individual untuk setiap pengguna.
- ❖ SQL Server mendukung 4 tipe role, yaitu :
  - 1) Fixed server roles
  - 2) User-defined server roles
  - 3) Fixed database roles
  - 4) User- defined database roles

## 2. Langkah-Langkah

### #LATIHAN LOGIN

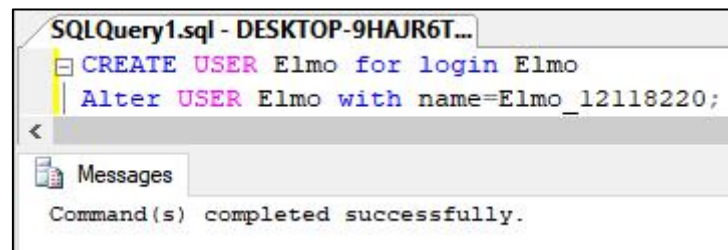
Buatlah login dengan login name adalah NAMAanda dengan passwordnya adalah NPM anda



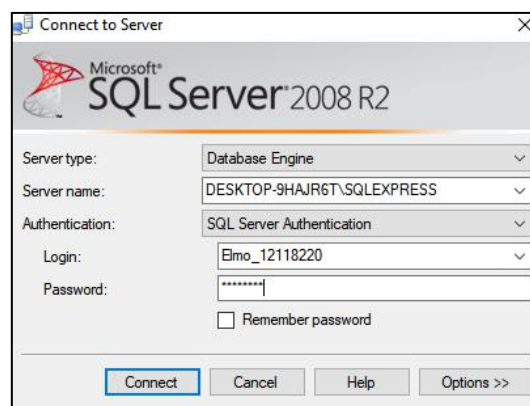
### # LATIHAN USER

Setelah selesai membuat login, buatlah user untuk login yang telah anda buat sebelumnya dengan menggunakan SQL Server Management Studio dengan format username adalah NPManda-NAMA anda.

Kemudian buatlah satu user lagi dengan menggunakan T-SQL dengan format username adalah NAMAanda-NPManda



Login dengan menggunakan user NPManda-NAMA anda



Terjadi error karena user Elmo\_12118220 tidak diberikan izin untuk membuat/memodifikasi database.

```
CREATE TABLE BIODATA (
  NPM numeric(8) Primary Key,
  Nama Varchar(50) not null,
  Kelas Varchar(5) not null,
  Jurusan Varchar(30) not null);
```

Messages  
Msg 262, Level 14, State 1, Line 1  
CREATE TABLE permission denied in database 'master'.

## #LATIHAN ROLE

Sebutkan empat macam jenis-jenis role yang biasa digunakan dalam basis data di SQL Server

| Fixed server-level role | Description   |
|-------------------------|---|
| sysadmin                | Members of the sysadmin fixed server role can perform any activity in the server.   |
| serveradmin             | Members of the serveradmin fixed server role can change server-wide configuration options and shut down the server.   |
| securityadmin           | Members of the securityadmin fixed server role manage logins and their properties. They can <b>GRANT</b> , <b>DENY</b> , and <b>REVOKE</b> server-level permissions. They can also <b>GRANT</b> , <b>DENY</b> , and <b>REVOKE</b> database-level permissions if they have access to a database. Additionally, they can reset passwords for SQL Server logins.<br><br><b>IMPORTANT:</b> The ability to grant access to the Database Engine and to configure user permissions allows the security admin to assign most server permissions. The securityadmin role should be treated as equivalent to the sysadmin role. |
| processadmin            | Members of the processadmin fixed server role can end processes that are running in an instance of SQL Server.  |

Buatlah role dengan format nama untuk role adalah NPManda dengan authorisasi sebagai security admin

Server roles:

- ☐ bulkadmin
- ☐ dbcreator
- ☐ diskadmin
- ☐ processadmin
- ☒ public
- ☒ securityadmin
- ☐ serveradmin
- ☐ setupadmin
- ☐ sysadmin



## #LATIHAN PERMISSION

Role yang telah anda buat berikan hak akses yang hanya bisa mengakses select, insert, update dan delete pada database dan tabel yang telah anda buat sebelumnya untuk user yang telah anda buat (NPManda-NAMAanda dan NAMAanda-NPManda)

```
GRANT CREATE TABLE TO Elmo_12118220
GRANT INSERT,UPDATE,DELETE,SELECT To Elmo_12118220
```

Messages  
Command(s) completed successfully.

2. Login sebagai NAMAanda-NPManda kemudian masukkan data ke tabel BIODATA

```
Insert into BIODATA values
(12345678,'Ani Budi Citra','2KA05','Sistem Informasi'),
(56789012,'Dea Efani Fita Gita','1IA01','Teknik Informatika'),
(34567890,'Hari Irfandi Jonas','3DB04','Manajemen Informatika'),
(23456789,'Kamia Lana Musa','4KB02','Sistem Komputer'),
(45678901,'Nando Oprah Prawira','2DC03','Teknik Komputer')
```

Messages  
Command(s) completed successfully.

```
Insert into BIODATA values
(12345678,'Ani Budi Citra','2KA05','Sistem Informasi'),
(56789012,'Dea Efani Fita Gita','1IA01','Teknik Informatika'),
(34567890,'Hari Irfandi Jonas','3DB04','Manajemen Informatika'),
(23456789,'Kamia Lana Musa','4KB02','Sistem Komputer'),
(45678901,'Nando Oprah Prawira','2DC03','Teknik Komputer');

select * from BIODATA
```

|   | NPM      | Nama                | Kelas | Jurusan               |
|---|----------|---------------------|-------|-----------------------|
| 1 | 12345678 | Ani Budi Citra      | 2KA05 | Sistem Informasi      |
| 2 | 23456789 | Kamia Lana Musa     | 4KB02 | Sistem Komputer       |
| 3 | 34567890 | Hari Irfandi Jonas  | 3DB04 | Manajemen Informatika |
| 4 | 45678901 | Nando Oprah Prawira | 2DC03 | Teknik Komputer       |
| 5 | 56789012 | Dea Efani Fita Gita | 1IA01 | Teknik Informatika    |

# **LAPORAN AKHIR**

## **PERTEMUAN 5**



(SQL SERVER SECURITY)

Disusun Oleh:

Nama : Elmo Allistair

NPM : 12118220

Kelas : 4KA17

*Kelompok : (Opsional)*

**LEMBAGA PENGEMBANGAN KOMPUTERISASI**  
**UNIVERSITAS GUNADARMA**

**2021**

**PERTEMUAN : 5**

**Tujuan Aktivitas :**

- 1. Memahami security principals dan securables di SQL Server**
- 2. Memahami mode otentikasi di SQL Server**
- 3. Memahami users dan logins istimewa di SQL Server**
- 4. Dapat menerapkan password policy yang aman di SQL Server**
- 5. Memahami dan mencegah SQL Injection di SQL Server**

## **TAHAPAN Pengerjaan**

### **1. Ringkasan Materi**

- ❖ Terdapat tiga level keamanan yang dapat diolah pada security principals SQL Server, yaitu Windows, SQL Server, dan database. Pada masing-masing level terdapat lagi security principals yang dapat diolah. Level-level tersebut adalah: Windows level, SQL server level dan Database level
- ❖ Terdapat dua macam mode otentikasi pada SQL Server yaitu:
  - 1) Mode otentikasi Windows: Penggunaan mode ini paling cocok ketika database hanya diakses dalam satu lingkup organisasi.
  - 2) Mode otentikasi SQL Server dan Windows (mixed mode): Penggunaan mode ini paling cocok ketika database juga diakses oleh pengguna yang berada di luar lingkup suatu organisasi (diluar domain Windows) atau pun pengguna yang tidak menggunakan perangkat Windows.
- ❖ Logins dan Users Istimewa di SQL Server:
  - 1) Administrators group, sebuah grup lokal dalam server database. Anggota dari grup ini biasanya akun local Administrator user dan user lainnya yang telah diatur sebagai administrator dalam lokal sistem.
  - 2) Administrator, sebuah user account lokal di server. Akun ini memberikan hak akses sebagai administrator pada sistem. Jika SQL Server terpasang di Windows domain, maka administrator account biasanya memiliki hak akses secara domain juga.
  - 3) Sa login, adalah akun sistem administrator dengan model keamanan baru yang telah terintegrasi dan diperluas, sa tidak lagi dibutuhkan. Sa disediakan untuk kompatibilitas terhadap SQL Server versi sebelumnya.
  - 4) Guest User adalah pengguna khusus yang dapat Anda tambahkan ke database untuk memungkinkan seseorang dengan login SQL Server yang valid untuk mengakses database.

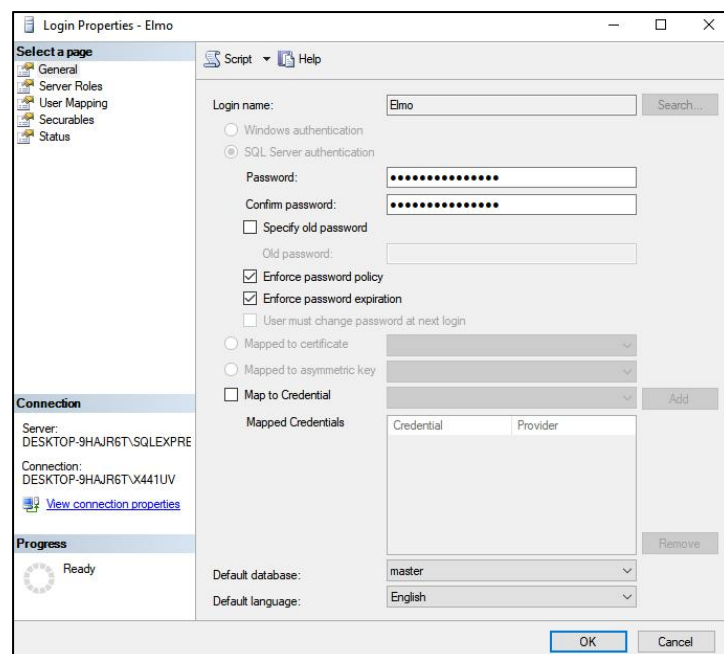
## 2. Langkah-Langkah

- a. Manakah mode otentikasi yang lebih baik antara mode Windows dan mode campuran? Sertakan alasannya!

Mode otentikasi Windows, manfaat utama mode ini adalah menggunakan enkripsi untuk mengotentikasi pengguna SQL Server, selain itu penggunaan mode ini memungkinkan pengguna untuk memusatkan administrasi akun untuk seluruh perusahaan di satu direktori.

- b. Buatlah password policy untuk perusahaan x pada Local/Domain Security Policy, berikan alasan yang konkrit untuk masing-masing kebijakan/policy yang telah Anda tentukan.

Domain policy digunakan untuk memastikan bahwa password login telah memenuhi kebijakan yang ada memiliki beberapa keuntungan, terutama dalam proses audit, Administrator tidak perlu memeriksa secara manual apakah password login telah memenuhi standar yang telah ditetapkan atau belum.



- c. Buatlah store procedure yang dapat diinjeksi dan aman dari sql injection. Store procedure harus mengembalikan tabel yang berisi nama pekerja dan nama departemen berdasarkan manager pekerja tersebut. Injeksi yang dilakukan yaitu injeksi yang mengembalikan daftar semua akun login yang ada dalam SQL Server instance yang sedang diserang.

Membuat Stored Procedure:

```
CREATE PROCEDURE [sp_demo_injection01]
    @first_name NVARCHAR(MAX)
AS
BEGIN
    DECLARE @sqlcmd NVARCHAR(MAX);

    SET @sqlcmd = N'SELECT employee_id,first_name,last_name,email
                    FROM dbo.employees
                    WHERE first_name= '''+@first_name+'''';

    EXECUTE (@sqlcmd)
END
```

Messages  
Command(s) completed successfully.

Mengeksekusi Procedure

```
Declare @var sysname
set @var = 'Alexander'
EXEC sp_demo_injection01 @var
```

Results

|   | employee_id | first_name | last_name | email   |
|---|-------------|------------|-----------|---------|
| 1 | 103         | Alexander  | Hunold    | AHUNOLD |
| 2 | 115         | Alexander  | Khoo      | AKHOO   |

Menguji keamanan Store Procedure dari sql injection.

```
CREATE PROCEDURE [sp_demo_injection02]
    @first_name NVARCHAR(MAX)
AS
BEGIN
    DECLARE @SQLCMD NVARCHAR(MAX);
    DECLARE @PARAMS NVARCHAR(MAX);
    SET @SQLCMD = N'SELECT employee_id,first_name,last_name,email
    FROM dbo.employees
    WHERE first_name= @first_name';
    SET @PARAMS = N'@first_name NVARCHAR(MAX)';
    EXECUTE sp_executesql @sqlcmd, @params, @first_name;
END
```

Messages

Command(s) completed successfully.

```
DECLARE @var sysname

SET @var = 'name employee'; delete from dbo.employees;
PRINT 'table employees telah dikosongkan!';
-- data on table employees has been deleted!!'
EXEC sp_demo_injection02 @var
```

Results Messages

| employee_id | first_name | last_name | email |
|-------------|------------|-----------|-------|
|-------------|------------|-----------|-------|

```
DECLARE @var sysname

SET @var = 'name employee'; delete from dbo.employees;
PRINT 'table employees telah dikosongkan!';
-- data on table employees has been deleted!!'
EXEC sp_demo_injection01 @var
```

Results Messages

(0 row(s) affected)

(107 row(s) affected)

table employees telah dikosongkan!

## Menampilkan data security login

```
DECLARE @var sysname
SET @var = 'nama employee'; select * from master.sys.sql_logins;
PRINT '' TABLE USER Login'';
-- Table Selected!!'
EXEC sp_demo_injection01 @var
```

Results Messages

| employee_id | first_name | last_name | email |
|-------------|------------|-----------|-------|
|-------------|------------|-----------|-------|

|   | name                              | principal_id | sid                                | type | type_desc |
|---|-----------------------------------|--------------|------------------------------------|------|-----------|
| 1 | sa                                | 1            | 0x01                               | S    | SQL_LOGIN |
| 2 | ##MS_PolicyEventProcessingLogin## | 256          | 0x0A6983CDF023464B9E86E4EEAB92C5DA | S    | SQL_LOGIN |
| 3 | ##MS_PolicyTsqlExecutionLogin##   | 257          | 0x8F651FE8547A4644A0C06CA83723A876 | S    | SQL_LOGIN |

## Menampilkan data login user

```
DECLARE @var sysname
SET @var = 'nama employee'; select * from master.sys.sql_logins;
PRINT '' TABLE USER Login'';
-- Table Selected!!'
EXEC sp_demo_injection02 @var
```

Results Messages

| employee_id | first_name | last_name | email |
|-------------|------------|-----------|-------|
|-------------|------------|-----------|-------|

# **LAPORAN AKHIR**

## **PERTEMUAN 6**



*(PERFORMING DATABASE BACKUP)*

Disusun Oleh:

Nama : Elmo Allistair

NPM : 12118220

Kelas : 4KA17

*Kelompok : (Opsional)*

**LEMBAGA PENGEMBANGAN KOMPUTERISASI**

**UNIVERSITAS GUNADARMA**

**2021**



**PERTEMUAN : 6**

**Tujuan Aktivitas :**

- 1. Membuat Rencana Backup and Recovery**
- 2. Memilih Perangkat dan Media Backup**
- 3. Menggunakan Strategi Backup**
- 4. Membuat Perangkat Backup**
- 5. Melakukan Backup**

## **TAHAPAN Pengerjaan**

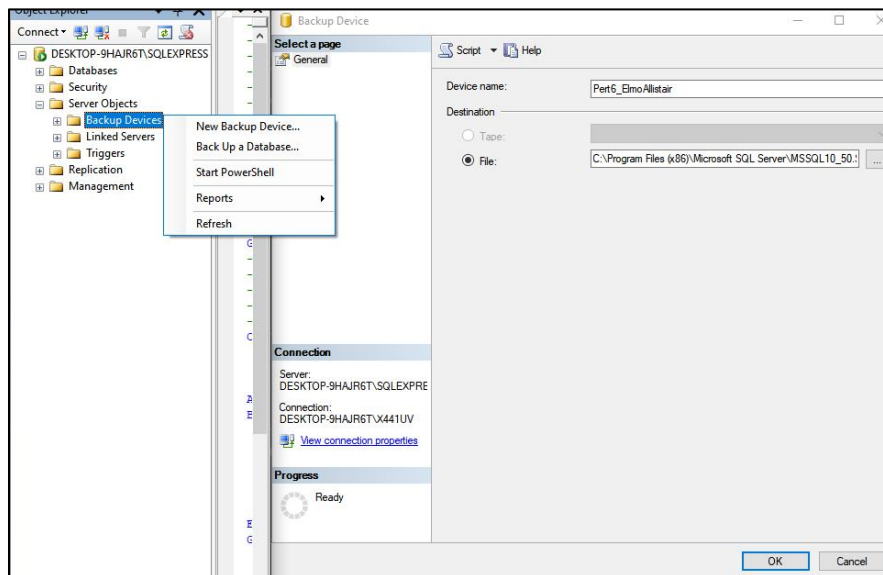
### **1. Ringkasan Materi**

- ❖ Membuat backup database berbeda dengan membackup server atau workstation, terutama karena sering perlu menggabungkan semua (atau hampir semua) teknik yang ada untuk memastikan bahwa dapat memulihkan database sepenuhnya.
- ❖ Jenis backup antara lain:
  - 1) Full database backup, mencakup semua objek, tabel sistem, dan data. Saat backup dimulai, SQL Server menyalin semua yang ada di database dan juga menyertakan bagian dari log transaksi yang dibutuhkan saat backup sedang berlangsung.
  - 2) Diferensial backup, dirancang untuk membuat backup data yang telah berubah sejak full backup terakhir. Karena hanya perubahan yang tersimpan, jenis cadangan ini memerlukan sedikit waktu, dan dapat dilakukan lebih sering.
  - 3) Transaction Log Backup, adalah pencatatan serial dari semua modifikasi basis data dan digunakan selama operasi pemulihan untuk commit transaksi yang telah selesai dan untuk roll back transaksi yang belum selesai.
  - 4) File dan filegroup backup, memungkinkan membuat backup file database dan filegroup daripada keseluruhan database. Jenis backup ini berguna jika berurusan dengan database besar dan ingin menghemat waktu dengan membackup file individual daripada keseluruhan database. Banyak faktor yang mempengaruhi backup file dan filegroup.

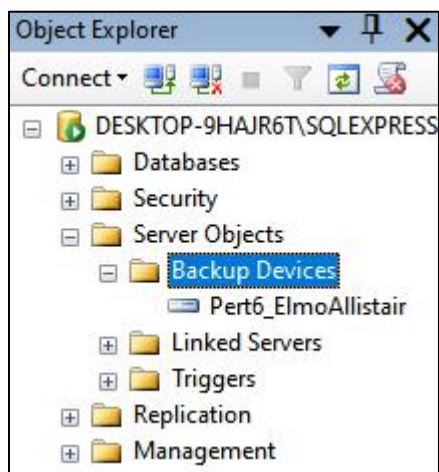
## 2. Langkah-Langkah

Lakukan Backup terhadap Database LePkom, untuk semua jenis Backup. Capture dan beri penjelasan langkah-langkah backup tersebut pada File Laporan.

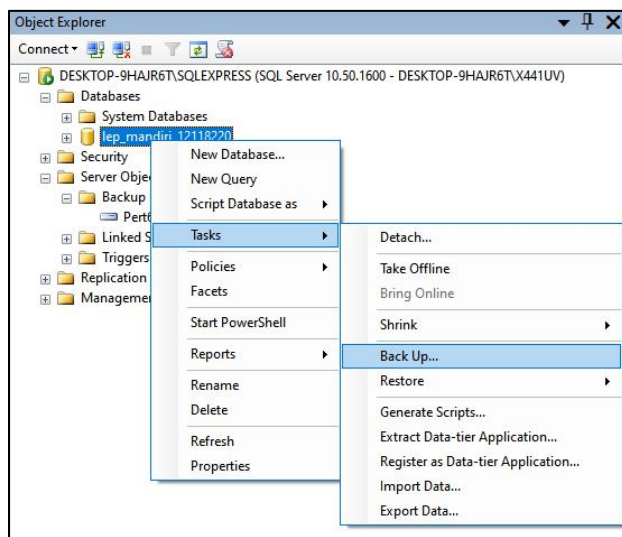
Klik kanan Server Object > Backup Devices. Pilih “New Backup Device”. Berikan nama device lalu klik “OK”



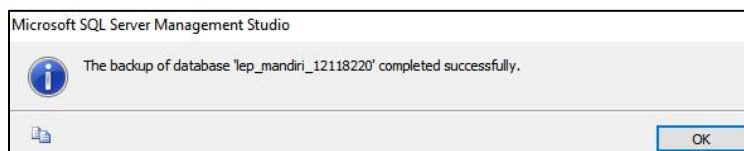
Periksa Backup Devices yang telah dibuat di Object Explorer



Klik kanan database lep\_mandiri\_12118220, klik “Task” lalu pilih “Back Up”



Konfirmasi Backup berhasil dilakukan



# **LAPORAN AKHIR**

## **PERTEMUAN 7**



### **PERFORMING DATABASE RECOVERY**

Disusun Oleh:

Nama : Elmo Allistair

NPM : 12118220

Kelas : 4KA17

*Kelompok : (Opsional)*

**LEMBAGA PENGEMBANGAN KOMPUTERISASI**

**UNIVERSITAS GUNADARMA**

**2021**

**PERTEMUAN : 7**

**Tujuan Aktivitas :**

**1. Praktikan diharapkan dapat melakukan database recovery**

## **TAHAPAN Pengerjaan**

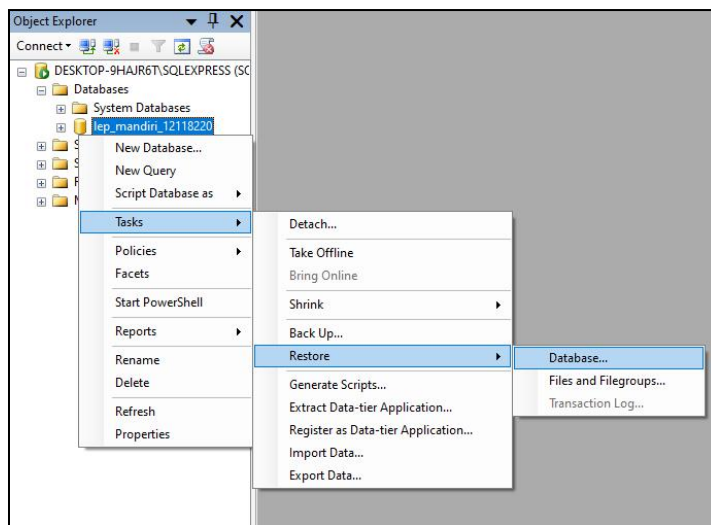
### **1. Ringkasan Materi**

❖ SQL Server 2008 menggunakan model pemulihan untuk membantu merencanakan backup. Jenis database yang dibackup dan jenis backup yang dilakukan mendorong pilihan untuk model pemulihan. Tiga model pemulihan tersedia:

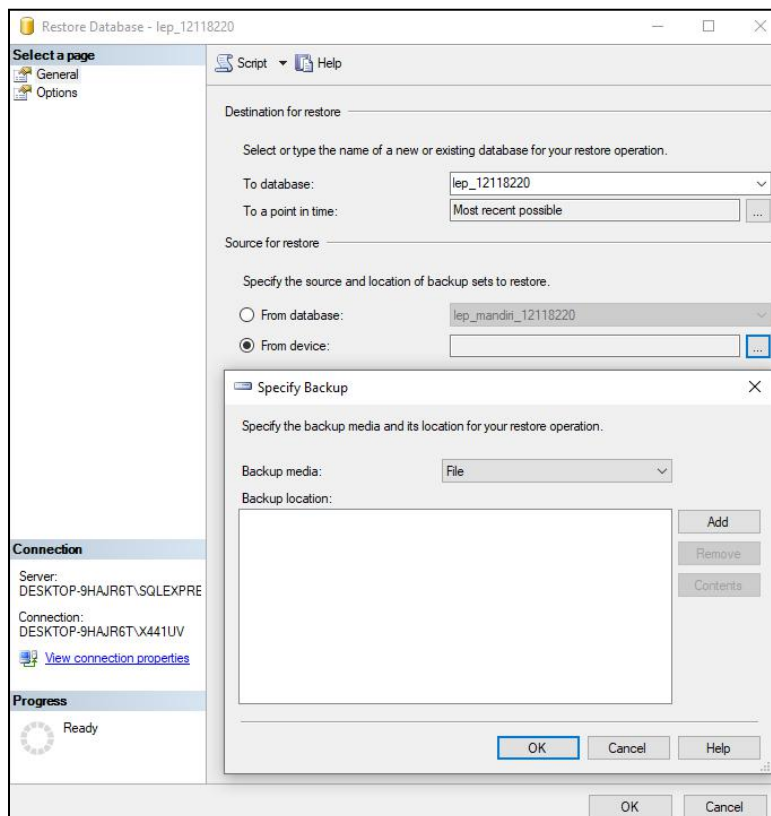
- 1) Simple, model pemulihan sederhana dirancang untuk database yang perlu dipulihkan sampai pada titik backup terakhir. Strategi backup dengan model ini harus terdiri dari backup penuh dan diferensial. Backup log transaksi tidak dapat dilakukan saat model pemulihan sederhana diaktifkan. SQL Server 2008 mengaktifkan pilihan Truncate Log On Checkpoint, yang menghapus entri tidak aktif dalam log transaksi di Checkpoint. Model ini membersihkan log transaksi, maka sangat ideal untuk kebanyakan database sistem.
- 2) Full, model pemulihan penuh dirancang untuk database yang perlu dipulihkan sampai titik kegagalan atau pada titik waktu tertentu. Bila menggunakan model ini, semua operasi dicatat, termasuk operasi massal dan pemuatan massal data.
- 3) Bulk-logged. Model pemulihan massal-log (bulk-logged) mengurangi penggunaan ruang log namun tetap mempertahankan sebagian besar fleksibilitas model pemulihan penuh. Dengan model ini, operasi massal dan muatan massal minimal login dan tidak dapat dikendalikan secara per operasi. Anda perlu mengulang secara manual operasi massal dan beban massal jika database gagal sebelum Anda melakukan backup penuh atau diferensial.

## 2. Langkah-Langkah

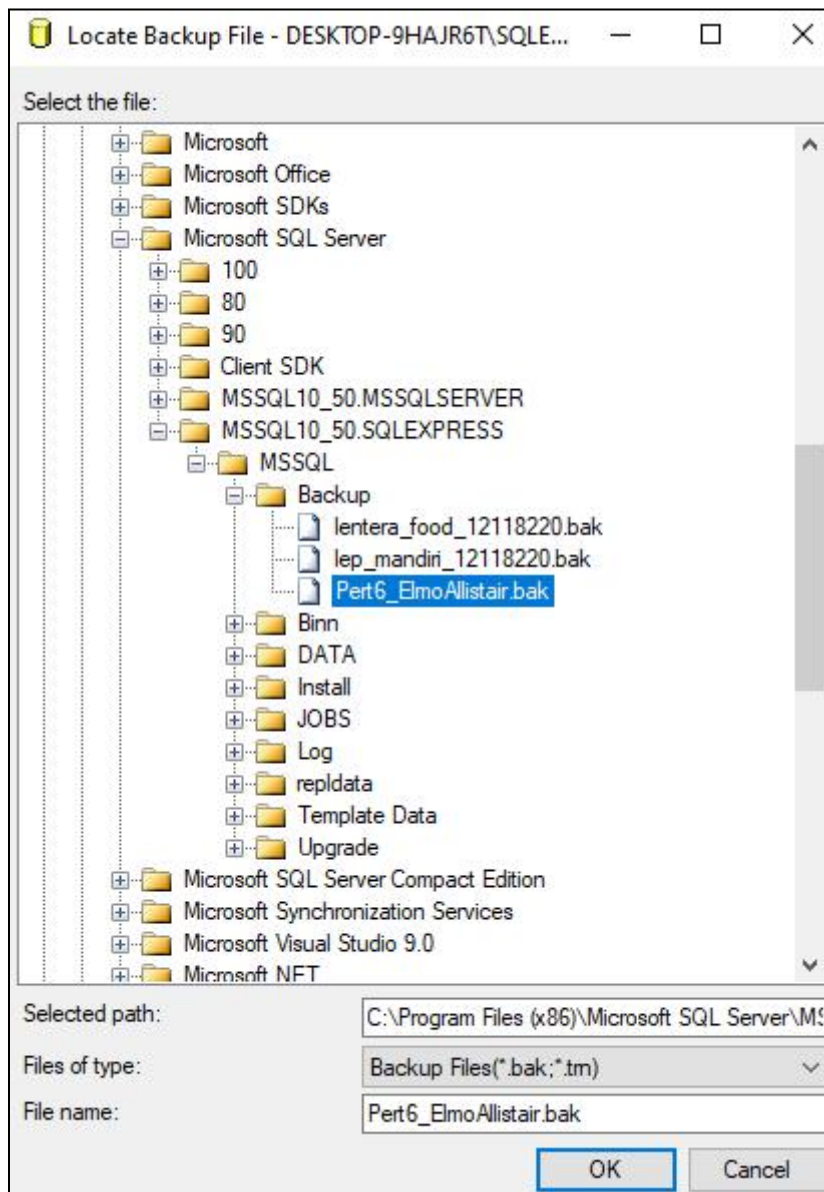
Klik kanan database **lep\_mandiri\_12118220**, pilih **Task > Restore > Database**.



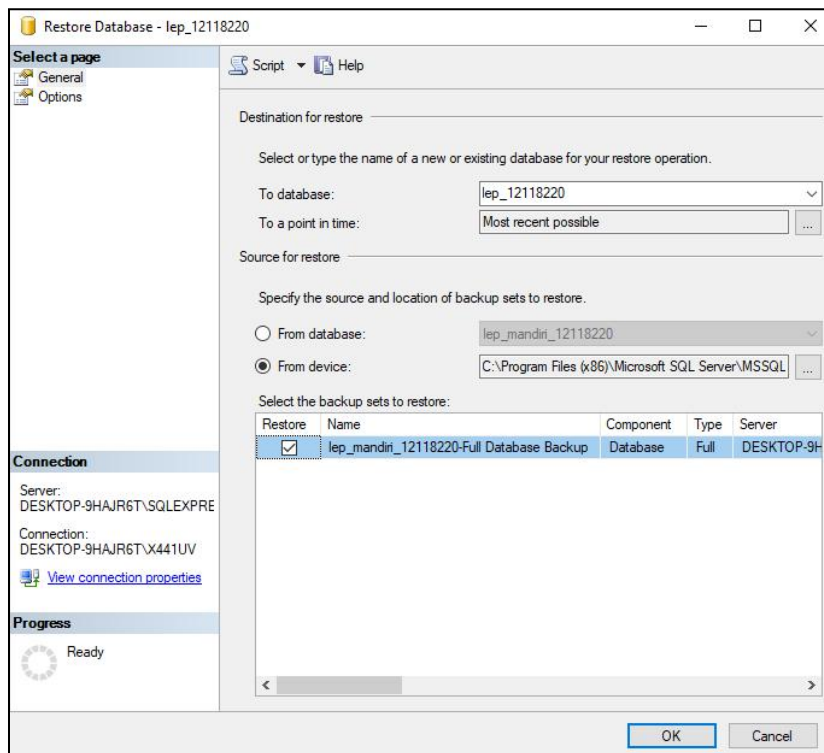
Atur destinasi database ke database **lep\_12118220**, centang opsi “From Device” lalu klik “...”. Pilih “Add”



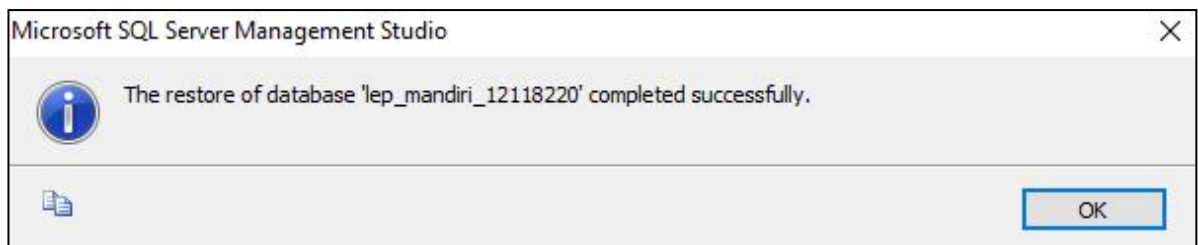
Pilih backup “**Pert6\_ElmoAlistair.bak**” lalu klik “OK”



Centang set backup “lep\_mandiri\_12118220-Full Database Backup” lalu klik “OK”



Tunggu proses restore selesai, lalu akan muncul notifikasi restore database telah sukses.





# **LAPORAN AKHIR**

## **PERTEMUAN 8**



### **TRANSFERING DATA**

Disusun Oleh:

Nama : Elmo Allistair

NPM : 12118220

Kelas : 4KA17

*Kelompok : (Opsional)*

**LEMBAGA PENGEMBANGAN KOMPUTERISASI**

**UNIVERSITAS GUNADARMA**

**2021**

**PERTEMUAN : 8**

**Tujuan Aktivitas :**

**1. Praktikan dapat melakukan ekspor data**

**2. Praktikan dapat melakukan import data**

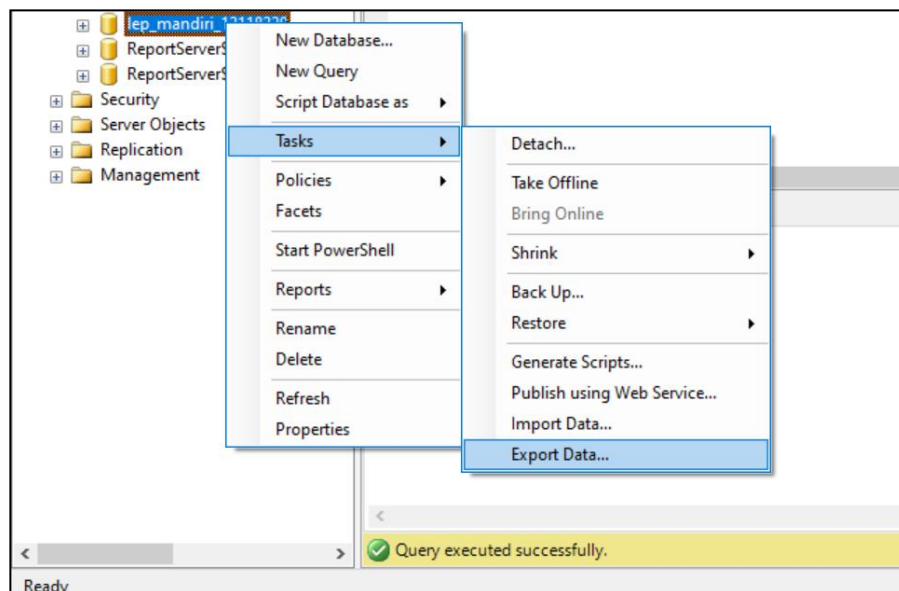
## **TAHAPAN Pengerjaan**

### **1. Ringkasan Materi**

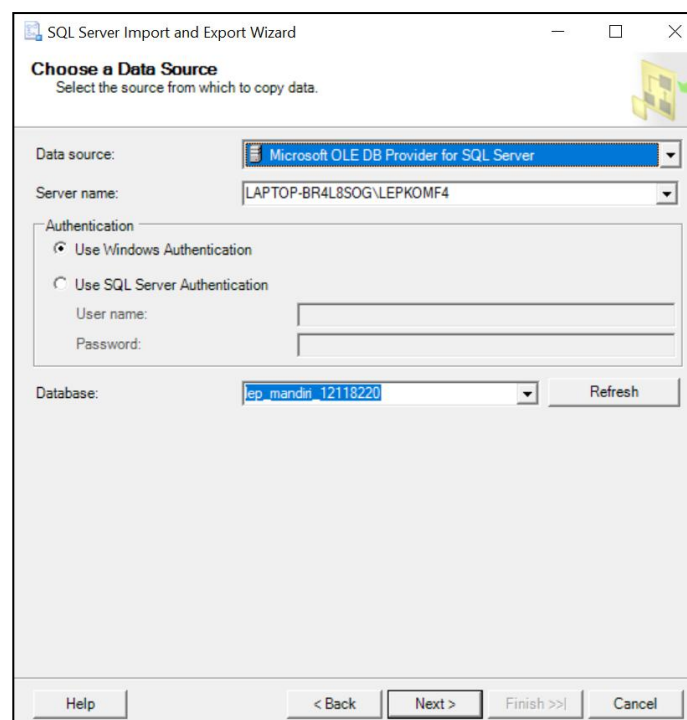
- ❖ SQL Server Integration Services (SSIS) adalah alat yang digunakan untuk melakukan operasi ETL; yaitu mengekstrak, mengubah dan memuat data.
- ❖ Pemrosesan ETL biasa dilakukan dalam aplikasi data warehousing (DW), namun SSIS sama sekali tidak terbatas hanya pada DW; misalnya ketika membuat Rencana Pemeliharaan menggunakan SQL Server Management Studio (SSMS), sebuah paket SSIS dibuat.
- ❖ Pada tingkat yang lebih tinggi, SSIS menyediakan kemampuan untuk:
  - 1) Mengambil data dari hampir semua sumber.
  - 2) Melakukan berbagai transformasi pada data; misalnya mengkonversi dari satu jenis ke yang lain, dikonversi ke huruf besar atau huruf kecil, melakukan perhitungan, dan lain-lain.
  - 3) Memuat (load) data ke hampir semua sumber.
  - 4) Mendefinisikan alur kerja (workflow).
- ❖ Versi pertama SSIS dirilis dengan SQL Server 2005. SSIS adalah pengganti Data Transformation Services (DTS) yang tersedia dengan SQL Server 7.0 dan SQL Server 2000. SSIS menggunakan kemampuan yang diperkenalkan dengan DTS. Pada bab ini akan dibahas mengenai bagaimana mengambil dan memuat data dari dan ke berbagai sumber (MS. Access dan MS. Excel). Mengambil data dari suatu sumber untuk dimasukkan kedalam database disebut dengan impor. sedangkan memuat data dari database ke suatu sumber disebut dengan ekspor.

## 2. Langkah-Langkah

Klik kanan database lep\_mandiri\_12118220, pilih Task > Export Data

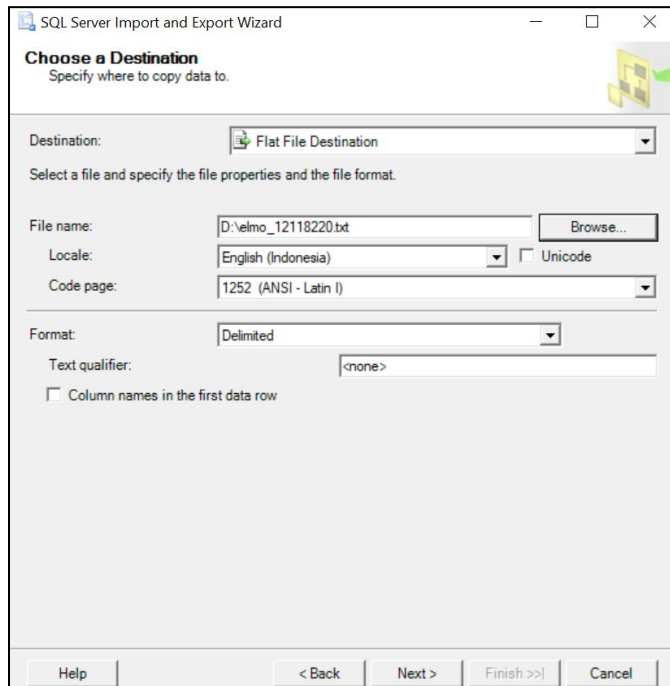


Pada pemilihan Data Source, pilih **Microsoft OLE DB Provider for SQL Server** kemudian untuk database pilih **lep\_mandiri\_12118220**



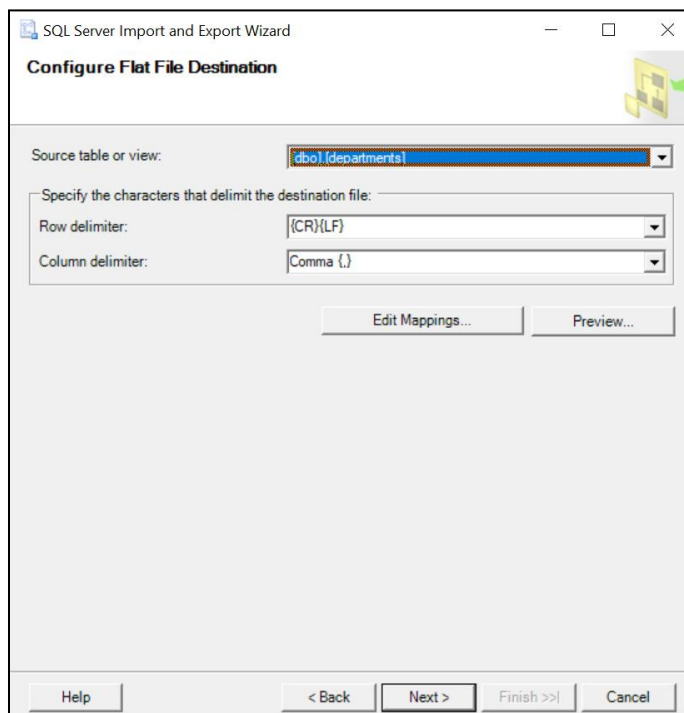
## A. Export txt

Pada pemilihan Destination, pilih **Flat File Destination**, buat file dengan nama **elmo\_12118220.txt**



The screenshot shows the 'Choose a Destination' step of the SQL Server Import and Export Wizard. The 'Destination' dropdown is set to 'Flat File Destination'. Below it, the 'File name' is 'D:\elmo\_12118220.txt', 'Locale' is 'English (Indonesia)', and 'Code page' is '1252 (ANSI - Latin I)'. The 'Format' is 'Delimited' and 'Text qualifier' is '<none>'. There is an unchecked checkbox for 'Column names in the first data row'. Navigation buttons at the bottom include 'Help', '< Back', 'Next >', 'Finish >>', and 'Cancel'.

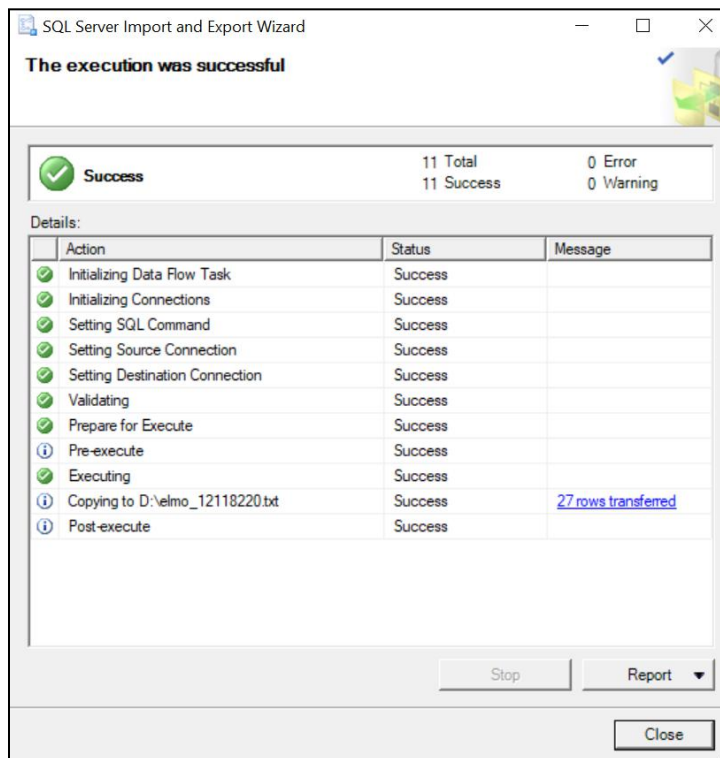
Pilih **[dbo].[departments]** sebagai source table or view



The screenshot shows the 'Configure Flat File Destination' step of the SQL Server Import and Export Wizard. The 'Source table or view' dropdown is set to '[dbo].[departments]'. Below it, 'Row delimiter' is '{CR}{LF}' and 'Column delimiter' is 'Comma (,)'.

Buttons for 'Edit Mappings...' and 'Preview...' are visible. Navigation buttons at the bottom include 'Help', '< Back', 'Next >', 'Finish >>', and 'Cancel'.

Tunggu proses eksekusi selesai

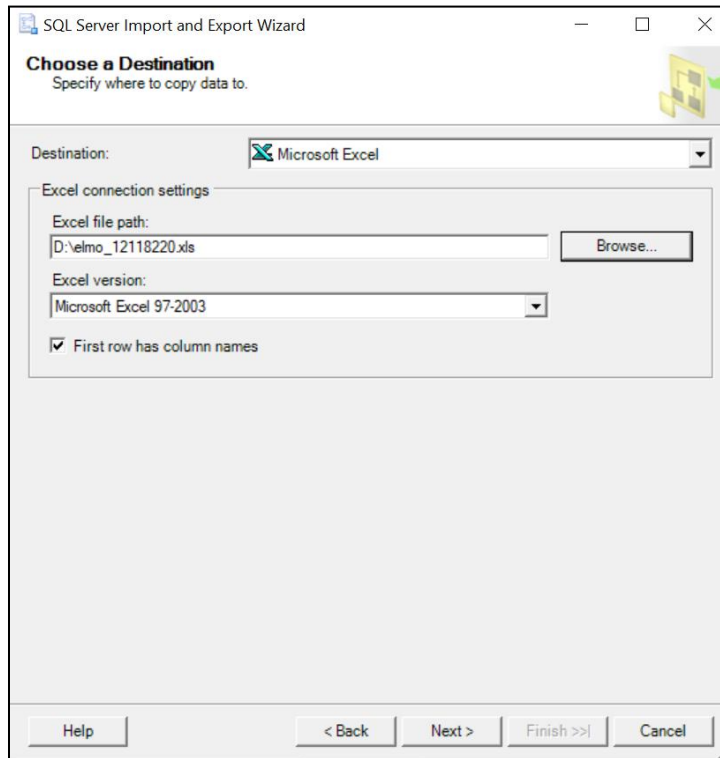


Konfirmasi file **elmo\_12118220.txt** sudah berisi data yang diekspor

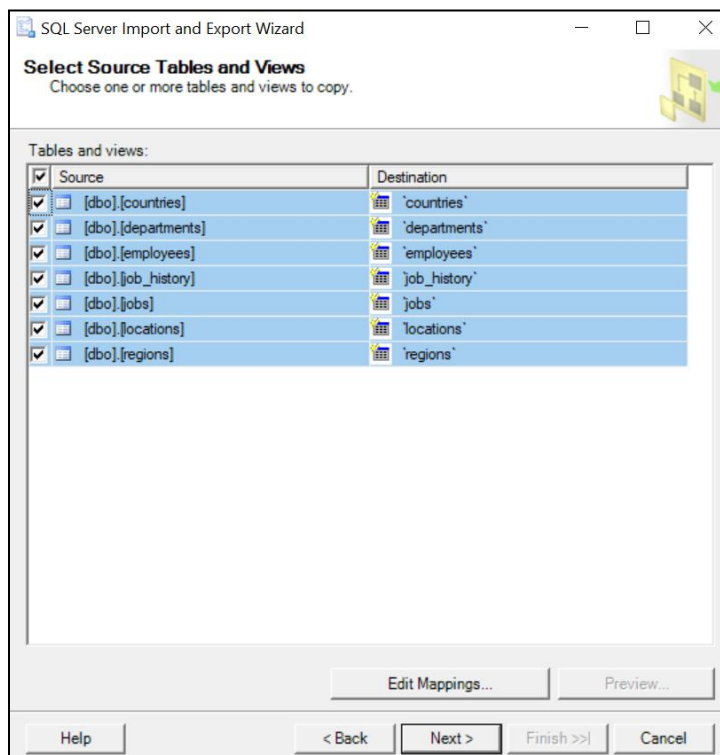


## B. Export xls

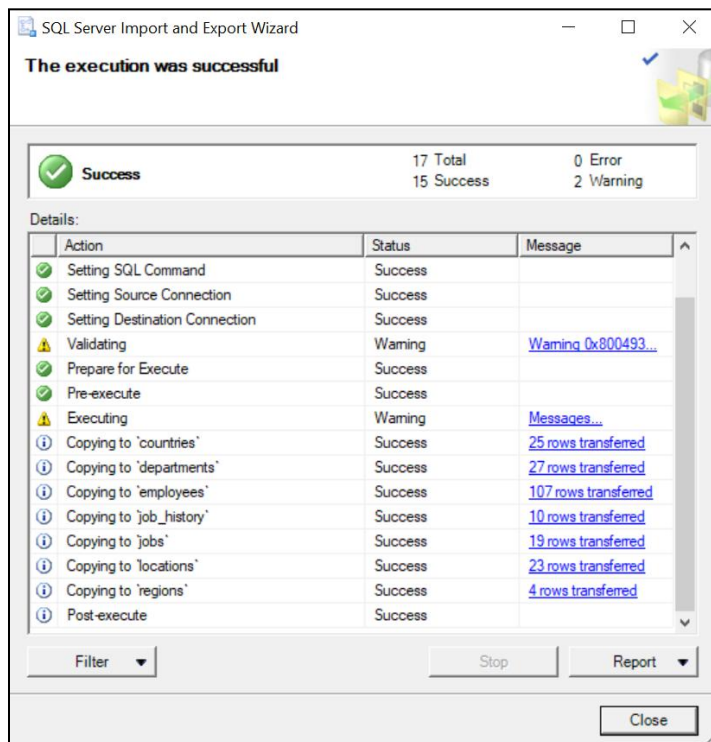
Pada pemilihan Destination, pilih **Microsoft Excel**, buat file dengan nama **elmo\_12118220.txt**



Centang semua tabel ingin akan diekspor



Tunggu proses eksekusi selesai

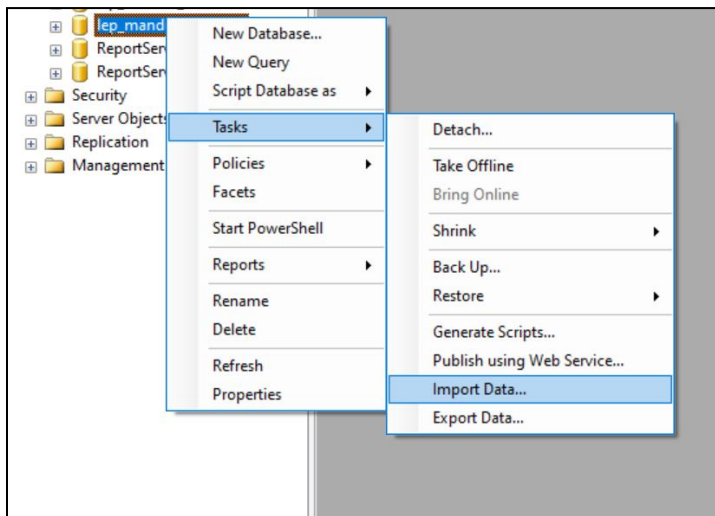


Konfirmasi file **elmo\_12118220.xls** sudah berisi data yang diekspor

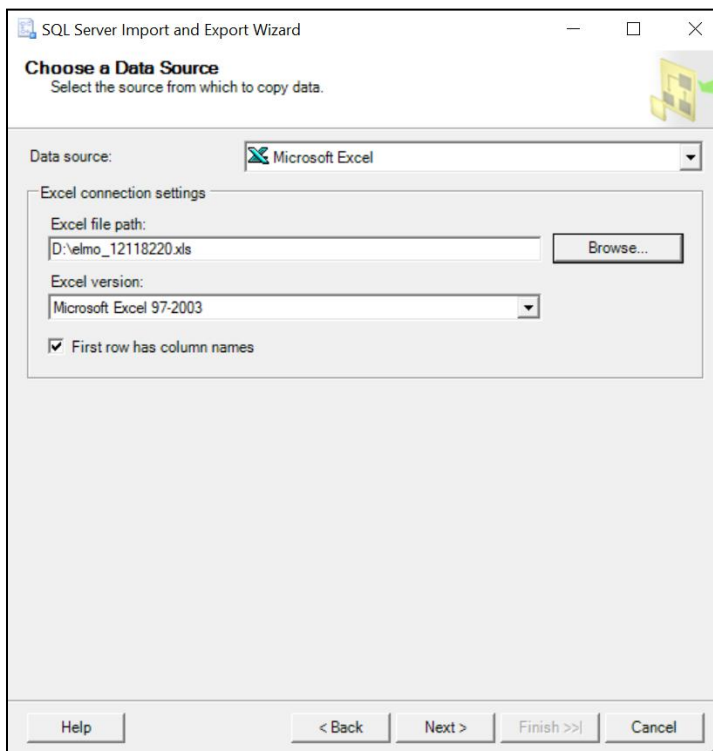
|    | A        | B          | C         | D         | E             | F   |
|----|----------|------------|-----------|-----------|---------------|-----|
| 1  | employee | start_date | end_date  | job_id    | department_id |     |
| 2  |          | 102        | 1993-01-1 | 1998-07-2 | IT_PROG       | 60  |
| 3  |          | 101        | 1989-09-2 | 1993-10-2 | AC_ACCOUNT    | 110 |
| 4  |          | 101        | 1993-10-2 | 1997-03-1 | AC_MGR        | 110 |
| 5  |          | 201        | 1996-02-1 | 1999-12-1 | MK_REP        | 20  |
| 6  |          | 114        | 1998-03-2 | 1999-12-3 | ST_CLERK      | 50  |
| 7  |          | 122        | 1999-01-0 | 1999-12-3 | ST_CLERK      | 50  |
| 8  |          | 200        | 1987-09-1 | 1993-06-1 | AD_ASSTANT    | 90  |
| 9  |          | 176        | 1998-03-2 | 1998-12-3 | SA_REP        | 80  |
| 10 |          | 176        | 1999-01-0 | 1999-12-3 | SA_MAN        | 80  |
| 11 |          | 200        | 1994-07-0 | 1998-12-3 | AC_ACCOUNT    | 90  |
| 12 |          |            |           |           |               |     |

### C. Import file excel ke database baru

Klik kanan database **lep\_mandiri\_12118220**, Pilih Task > Import Data



Pada pemilihan Data Source, pilih **Microsoft Excel**, buat file dengan nama **elmo\_12118220.xls**





Pada pemilihan Destination, pilih **Microsoft OLE DB Provider for SQL Server**, pilih database **elmo\_12118220**

SQL Server Import and Export Wizard

**Choose a Destination**  
Specify where to copy data to.

Destination: Microsoft OLE DB Provider for SQL Server

Server name: LAPTOP-BR4L8SOG\LEPKOMF4

Authentication:  
☒ Use Windows Authentication  
☐ Use SQL Server Authentication  
User name:   
Password:

Database: elmo\_12118220 Refresh New...

Help < Back Next > Finish >> Cancel

Centang semua tabel ingin akan diimport

SQL Server Import and Export Wizard

**Select Source Tables and Views**  
Choose one or more tables and views to copy.

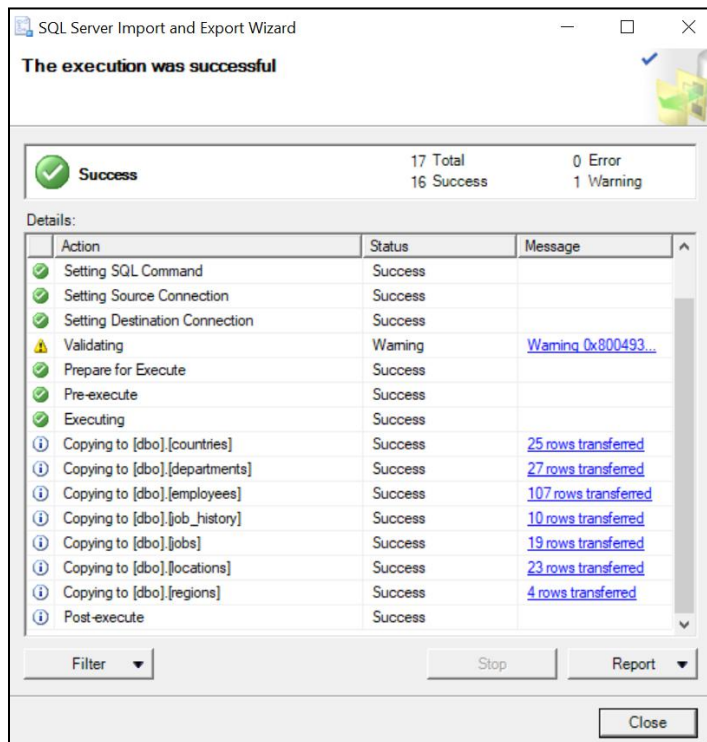
Tables and views:

| Source  | Destination         |
|---|---------------------|
| <input checked="" type="checkbox"/> 'countries'   | [dbo].[countries]   |
| <input type="checkbox"/> 'countries\$'            |                     |
| <input checked="" type="checkbox"/> 'departments' | [dbo].[departments] |
| <input type="checkbox"/> 'departments\$'          |                     |
| <input checked="" type="checkbox"/> 'employees'   | [dbo].[employees]   |
| <input type="checkbox"/> 'employees\$'            |                     |
| <input checked="" type="checkbox"/> 'job_history' | [dbo].[job_history] |
| <input type="checkbox"/> 'job_history\$'          |                     |
| <input checked="" type="checkbox"/> 'jobs'        | [dbo].[jobs]        |
| <input type="checkbox"/> 'jobs\$'                 |                     |
| <input checked="" type="checkbox"/> 'locations'   | [dbo].[locations]   |
| <input type="checkbox"/> 'locations\$'            |                     |
| <input checked="" type="checkbox"/> 'regions'     | [dbo].[regions]     |
| <input type="checkbox"/> 'regions\$'              |                     |

Edit Mappings... Preview...

Help < Back Next > Finish >> Cancel

Tunggu proses eksekusi selesai



Konfirmasi tabel berhasil diimpor

