

LAPORAN PRAKTIKUM
PRAKTIKUM 1:
“MEMBUAT DATABASE BIOSKOP”



Disusun Oleh :

Oktaviana Sadama Nur Azizah
24060121130060

PRAKTIKUM MANAJEMEN BASIS DATA
LAB A2

DEPARTEMEN ILMU KOMPUTER / INFORMATIKA
FAKULTAS SAINS DAN MATEMATIKA
UNIVERSITAS DIPONEGORO
SEMARANG
2023

BAB I

PENDAHULUAN

A. Rumusan Masalah

Membuat *database* sesuai dengan *business scenario* yang terdiri dari:

1. Deskripsi mengenai *business scenario*
2. *Logical Model*
3. Penjelasan *database* yang dibuat, yaitu:
 - Cara membuat tabel
 - Menampilkan isi tabel
 - Dokumentasi: *Screenshot*

B. Tujuan

1. Mampu menjelaskan struktur organisasi atau perusahaan berdasarkan *business scenario* yang ada.
2. Mampu membuat *logical model* berdasarkan *business scenario* guna memvisualisasi hubungan antara tindakan dan hasil yang diinginkan dari suatu program.
3. Mampu mengoperasikan Microsoft SQL Server sebagai sarana pembuatan *database*.

BAB II

DASAR TEORI

A. Basis Data dan Sistem Manajemen Basis Data (DBMS)

Basis data adalah kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi. Basis data merupakan aspek penting dalam sistem informasi karena berfungsi sebagai gudang penyimpanan data yang akan diolah lebih lanjut. Basis data dapat mengorganisasi data, menghindari duplikasi data, menghindari hubungan antar data yang tidak jelas dan juga *update* yang rumit.

Proses memasukkan dan mengambil data ke dan dari media penyimpanan data memerlukan perangkat lunak yang disebut Sistem Manajemen Basis Data (DBMS). DBMS merupakan sistem perangkat lunak yang memungkinkan pengguna basis data untuk memelihara, mengontrol dan mengakses data secara praktis dan efisien.

Dengan kata lain, semua akses ke basis data akan ditangani oleh DBMS. DBMS menjadi lapisan yang menghubungkan basis data dengan program aplikasi untuk memastikan bahwa basis data tetap terorganisasi secara konsisten dan dapat diakses dengan mudah.

B. *Business Scenario*

Business scenario adalah deskripsi singkat tentang suatu situasi atau kejadian dalam konteks bisnis atau organisasi. *Business scenario* dapat digunakan untuk memodelkan atau mensimulasikan kejadian-kejadian yang mungkin terjadi dalam sebuah perusahaan atau organisasi.

Dalam konteks teknologi informasi, *business scenario* sering digunakan untuk memahami kebutuhan bisnis dan sebagai sarana untuk merancang solusi teknologi informasi yang sesuai dengan kebutuhan tersebut.

C. *Logical Model*

Logical model merupakan tahapan kedua dalam mendesain suatu *database*. Tahapan ini menganalisis atribut, entitas, dan relasi yang sudah ditentukan di tahap *conceptual* agar. Jika pada tahap *conceptual* dihasilkan sebuah ERD, maka pada tahapan ini kita mengubah ERD tersebut menjadi suatu *relational database* dan *Enhance* ERD.

Tahapan dan proses pembuatan *logical model* adalah sebagai berikut:

1. Menentukan relasi-relasi yang ada dalam *relational database*.

2. Melakukan normalisasi untuk melihat apakah ada model data yang sudah dibuat dapat mendukung transaksi yang ada.
3. Mengecek apakah relasi-relasi yang sudah dibuat dapat mendukung transaksi *user*. Apabila tahapan ini tidak bisa dilakukan maka ada yang salah dengan data model yang dibuat.
4. Mengidentifikasi *constraint* yang ada pada *relational database* yang sudah dibuat sekaligus membuat *general constraint* apabila belum dibuat.

D. Microsoft SQL Server

Microsoft SQL Server adalah sebuah sistem manajemen basis data relasional yang dikembangkan oleh Microsoft. SQL Server dirancang untuk mengelola, menyimpan, dan mengambil data dari berbagai aplikasi bisnis yang berbeda. Dengan SQL Server, pengguna dapat membuat dan mengelola basis data relasional yang besar dan kompleks, serta melakukan analisis data untuk mendapatkan wawasan bisnis yang lebih baik.

SQL Server memiliki berbagai fitur dan fungsionalitas, antara lain:

1. Manajemen Basis Data
SQL Server menyediakan alat dan fitur yang kuat untuk membuat, mengelola, dan memelihara basis data relasional. Termasuk pembuatan tabel, indeks, dan *constraint*, serta kemampuan untuk melakukan *backup* dan *restore* basis data.
2. Pengambilan Data
SQL Server memiliki kemampuan untuk mengambil data dari berbagai sumber, termasuk sumber data yang berbeda-beda. Hal ini memudahkan pengguna untuk mengintegrasikan data dari berbagai aplikasi bisnis dan sumber data lainnya.
3. Analisis Data
SQL Server menyediakan kemampuan untuk melakukan analisis data yang canggih, termasuk pembuatan laporan dan kueri kompleks. Hal ini memungkinkan pengguna untuk mendapatkan wawasan bisnis yang lebih baik.
4. Keamanan Data
SQL Server menyediakan fitur keamanan yang kuat untuk melindungi data dari ancaman seperti peretasan dan serangan malware. Termasuk kemampuan untuk mengenkripsi data dan mengontrol akses ke data yang sensitif.
5. Skalabilitas
SQL Server dapat digunakan untuk mengelola basis data yang sangat besar dan kompleks, serta dapat diintegrasikan dengan berbagai sistem dan aplikasi bisnis.

BAB III

PEMBAHASAN

A. *Business Scenario*

Business Scenario: Efisiensi pembelian tiket di suatu Bioskop secara *online*.

Suatu bioskop ingin memperbaiki sistem penjualan tiket agar lebih efisien. Bioskop tersebut memutuskan untuk mengimplementasikan sistem pembelian tiket secara *online*. Adapun prosedur pembelian tiket secara *online* yang dapat dilakukan oleh *customer* sebagai berikut:

1. *Customer* ingin menonton film di bioskop.
2. *Customer* sudah mengunduh aplikasi pembelian tiket bioskop secara *online* dan membuat akun dengan memasukkan data nama depan, nama belakang, email, dan nomor HP.
3. *Customer* membeli tiket secara *online* dengan akun yang sudah dibuat.
4. *Customer* memilih film yang ingin ditonton.
5. *Customer* memilih jadwal penayangan film yang diinginkan.
6. *Customer* memilih *row seat* mana saja yang ingin ditempati.
7. *Customer* melakukan pembayaran.
8. *Customer* mendapatkan *e-ticket*.
9. *Customer* mencetak tiket fisik di bioskop.
10. *Customer* memasuki studio seperti yang tertera pada tiket.
11. *Customer* duduk di *row seat* yang sudah di-*booking*.

Business Rules

1. *Customer* dapat membeli satu atau lebih tiket dengan akun yang dimiliki dimana terdapat beberapa atribut dari akun *customer* yang meliputi informasi nama depan, nama belakang, email, dan nomor HP.
2. Setelah melakukan pembayaran, *customer* akan dialihkan ke *payment page* yang berisi rincian seperti *payment id*, *payment method*, *payment date time*, dan total harga tiket.
3. Dari *e-ticket* yang sudah dibeli dapat diketahui pula rincian mengenai *payment*, *customer*, *movie* yang ditonton dan informasi mengenai studio.
4. Setiap *e-ticket* yang dibeli oleh *customer* berisi data berupa *ticket id*, judul film, tanggal penayangan, jam tayang, *row seat*, nomor studio, *payment id*, dan email.
5. Informasi mengenai *movie* yang dapat dilihat oleh *customer* adalah judul film, genre, tahun rilis, durasi, dan juga *sensor* usia.

6. Adapun informasi mengenai studio yang dapat diketahui *customer* adalah mengenai lokasi studio yang ditampilkan melalui nomor studio dan juga kapasitas studio.

Adapun entitas dan atribut dari studi kasus di atas adalah sebagai berikut:

No	Entitas	Atribut
1	<i>Customer</i>	#* Email * Nama depan * Nama belakang * No HP
2	<i>Ticket</i>	#* <i>Ticket id</i> * Judul film * <i>Row seat</i> * No studio * Tanggal tayang * Jam tayang * <i>Payment id</i> * Email
3	<i>Movie</i>	#* Judul film * Genre * Tahun rilis * Durasi * <i>Censor</i>
4	<i>Payment</i>	#* <i>Payment id</i> * <i>Payment method</i> * <i>Payment date time</i> * Harga tiket
5	Studio	#* No studio * Kapasitas studio

Berikut juga dijelaskan kardinalitas dan optionalitas dari studi kasus di atas:

1. *Customer – Ticket*

Optionalitas:

Seorang *customer* yang akan menonton film **pasti** akan membeli satu atau lebih tiket. Satu atau lebih tiket **pasti** akan dibeli oleh *customer* yang akan menonton film.

Kardinalitas: One to Many

2. *Movie – Ticket*

Optionalitas:

Infomasi film yang akan ditonton **pasti** ada dalam setiap *ticket*. Satu atau lebih tiket **pasti** akan memiliki informasi mengenai film yang akan ditonton.

Kardinalitas: One to Many

3. *Studio – Ticket*

Optionalitas:

Infomasi studio **pasti** ada dalam setiap *ticket*. Satu atau lebih tiket **pasti** memiliki informasi studio.

Kardinalitas: One to Many

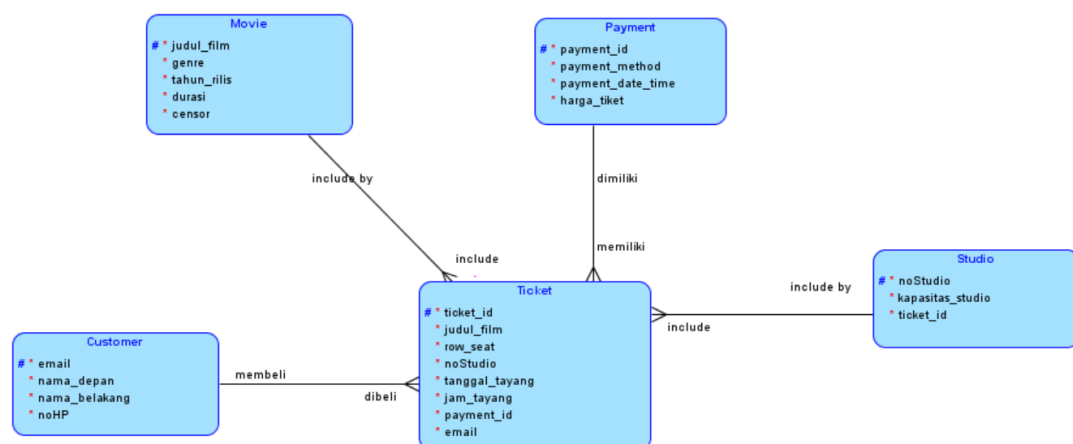
4. *Payment – Ticket*

Optionalitas:

Infomasi mengenai *payment* **pasti** ada dalam setiap *ticket*. Satu atau lebih tiket **pasti** memiliki informasi *payment* yang telah dilakukan.

Kardinalitas: One to Many

B. Logical Model



Penjelasan

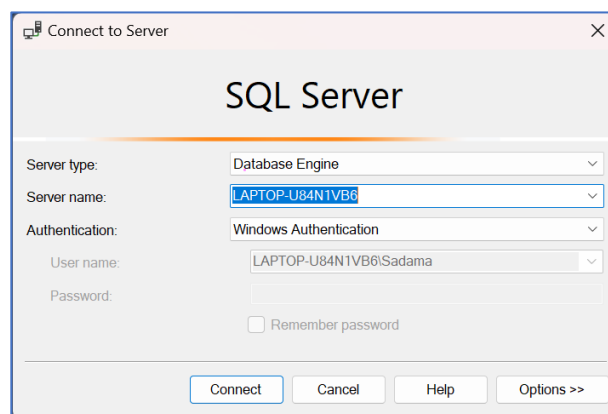
Logical model di atas menunjukkan relasi antar tabel dimana entitas *ticket* mengandung semua *foreign key* dari entitas *customer*, *movie*, *payment*, dan studio dengan rincian sebagai berikut:

1. Atribut email merupakan *foreign key* yang menghubungkan tabel *ticket* dengan tabel *customer*.
2. Atribut judul film merupakan *foreign key* yang menghubungkan tabel *ticket* dengan tabel *movie*.
3. Atribut nomor studio merupakan *foreign key* yang menghubungkan tabel *ticket* dengan tabel studio.
4. Atribut *payment id* merupakan *foreign key* yang menghubungkan tabel *ticket* dengan tabel *payment*.

C. Database

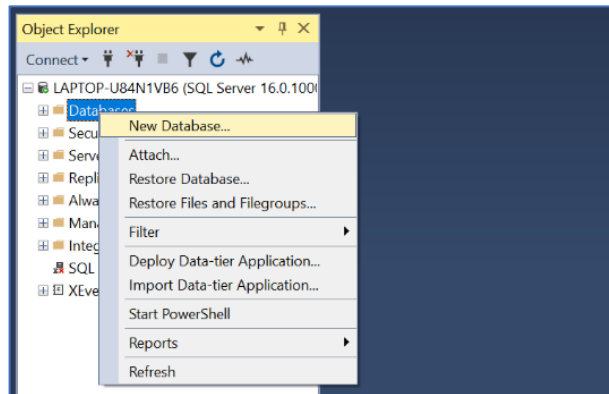
Database dibuat menggunakan Microsoft SQL Server dengan langkah-langkah sebagai berikut:

1. Membuat *Database* Baru
 - Buka aplikasi Microsoft SQL Server Management Studio, maka akan muncul *pop-up* 'Connect to Server'.

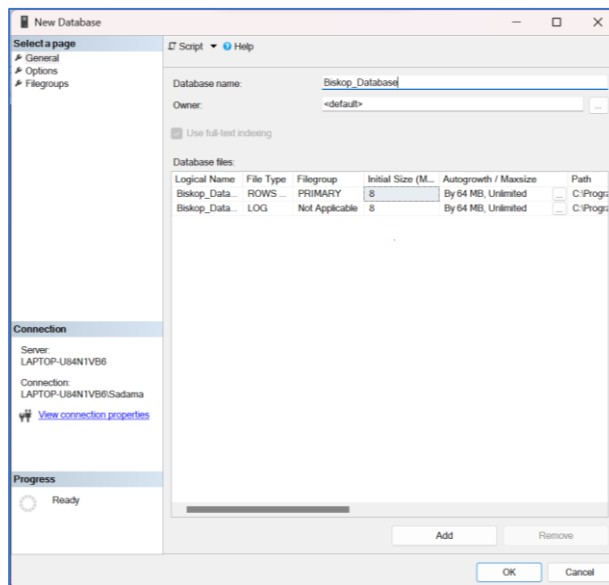


Dalam pembuatan database bioskop ini saya menggunakan *server name default* bawaan laptop. Setelah itu klik *Connect*. Pastikan untuk *connect* dengan *server* yang sama apabila ingin menggunakan *database* yang sebelumnya sudah dibuat dalam *server* tersebut.

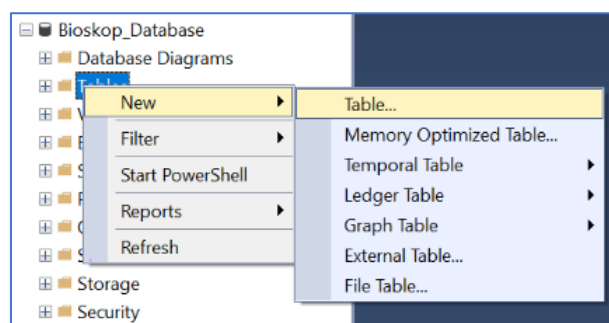
- Untuk membuat *database*, klik kanan pada bagian folder *database* dan klik *new database*.



- Beri nama *database*. Pada bagian LOG dapat ditentukan pula lokasi untuk *storage database*. Klik OK.



- Muncul *database* '*Bioskop_Database*'. Untuk membuat tabel, *expand Bioskop_Database* dan klik kanan pada file *Tables* → *New* → *Table*.



2. Menambahkan Atribut dan Tipe Data

Tambahkan atribut dan tipe data pada tabel yang ingin dibuat. *Set juga allow nulls dan primary key nya. Simpan dan beri nama tabel.*

– Table Customer

LAPTOP-U84N1VB6....se - dbo.Customer ✕			
	Column Name	Data Type	Allow Nulls
✕	email	varchar(50)	<input type="checkbox"/>
	nama_depan	varchar(20)	<input type="checkbox"/>
	nama_belakang	varchar(20)	<input type="checkbox"/>
	noHP	varchar(13)	<input type="checkbox"/>

– Table Movie

LAPTOP-U84N1VB6....abase - dbo.Movie ✕			
	Column Name	Data Type	Allow Nulls
✕	judul_film	varchar(100)	<input type="checkbox"/>
	genre	varchar(50)	<input type="checkbox"/>
	tahun_rilis	int	<input type="checkbox"/>
	durasi	time(7)	<input type="checkbox"/>
	sensor	varchar(5)	<input type="checkbox"/>

– Table Ticket

LAPTOP-U84N1VB6.B...base - dbo.Ticket ✕			
	Column Name	Data Type	Allow Nulls
✕	tiket_id	char(5)	<input type="checkbox"/>
	judul_film	varchar(100)	<input type="checkbox"/>
	row_seat	varchar(3)	<input type="checkbox"/>
	noStudio	int	<input type="checkbox"/>
	tanggal_tayang	date	<input type="checkbox"/>
	jam_tayang	varchar(5)	<input type="checkbox"/>
	payment_id	char(6)	<input type="checkbox"/>
	email	varchar(50)	<input type="checkbox"/>

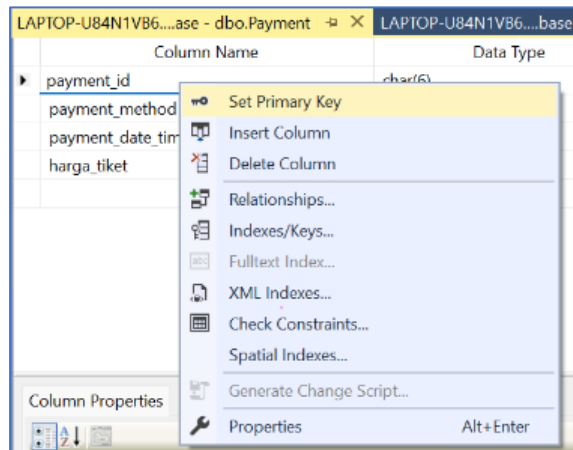
– Table Studio

LAPTOP-U84N1VB6....base - dbo.Studio ✕			
	Column Name	Data Type	Allow Nulls
✕	noStudio	int	<input type="checkbox"/>
	kapasitas_studio	int	<input type="checkbox"/>

– Table Payment

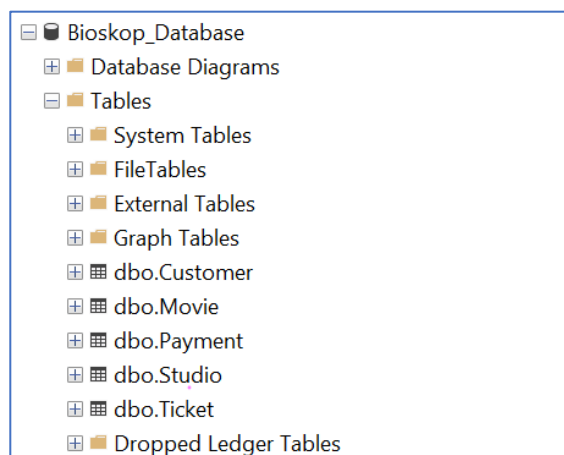
LAPTOP-U84N1VB6....ase - dbo.Payment ✕			
	Column Name	Data Type	Allow Nulls
✕	payment_id	char(6)	<input type="checkbox"/>
	payment_method	varchar(10)	<input type="checkbox"/>
	payment_date_time	datetime	<input type="checkbox"/>
	harga_tiket	int	<input type="checkbox"/>

Tambahkan pula *primary key* dari setiap UID tabel dengan klik kanan kolom pada tabel yang ingin dijadikan UID kemudian klik 'Set Primary Key'.

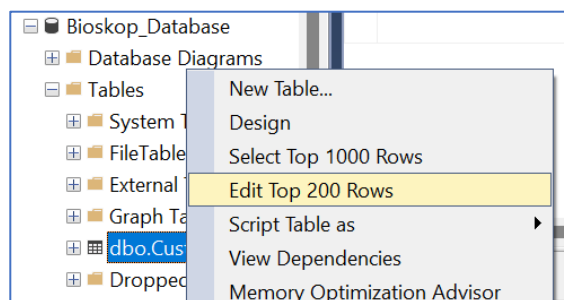


3. Insert Data Tabel

Apabila kolom setiap tabel sudah dibuat dan ditentukan tipe data, serta ketentuan *allow nulls* nya maka tabel-tabel tersebut akan muncul pada folder *table*, sebagai berikut:



Untuk meng-*insert*-kan data pada setiap tabel, klik kanan tabel yang ingin diinsertkan data dan pilih '*Edit Top 200 Rows*'.



– *Insert Table Customer*

LAPTOP-U84N1VB6....se - dbo.Customer		LAPTOP-U84N1VB6.B...abase - dbo.Tiket		
	email	nama_depan	nama_belakang	noHP
▶	tata123@gmail.com	Oktaviana	Sadama	087834766951
	ken89@gmail.com	Kennard	Pangaribuan	081568743910
	sabeth45@gmail.com	Elisabeth	Putri	0895383848471
	ais99@gmail.com	Aisyah	Rahmah	083145129564
	leon76@gmail.com	Leonard	Obadiaz	087419328892
*	NULL	NULL	NULL	NULL

– *Insert Table Movie*

LAPTOP-U84N1VB6....abase - dbo.Movie		LAPTOP-U84N1VB6....se - dbo.Customer			
	judul_film	genre	tahun_rilis	durasi	sensor
▶	Avengers: Endgame	Action	2019	03:01:00	PG-13
	Joker	Crime	2019	02:02:00	R
	Guardians of the Galaxy Vol. 3	Adventure	2023	02:29:00	PG-13
	Minios: The Rise of Gru	Komedi	2022	01:30:00	SU
	Suzume no Tojimari	Adventure	2022	02:02:00	SU
*	NULL	NULL	NULL	NULL	NULL

– *Insert Table Ticket*

LAPTOP-U84N1VB6.B...base - dbo.Ticket		LAPTOP-U84N1VB6.B...base - dbo.Ticket		LAPTOP-U84N1VB6....abase - dbo.Movie				
	tiket_id	judul_film	row_seat	noStudio	tanggal_tayang	jam_tayang	payment_id	email
▶	K8D4F	Guardians of the Galaxy Vol. 3	A10	3	2023-05-03	15:00	3S6F8L	tata123@gmail.com
	P6Z2L	Guardians of the Galaxy Vol. 3	A11	3	2023-05-03	15:00	3S6F8L	tata123@gmail.com
	R9S7N	Suzume no Tojimari	E8	1	2023-03-08	20:15	H2R9TJ	ken89@gmail.com
	M2V5H	Avengers: Endgame	B13	2	2019-04-24	13:30	D5N7MK	leon76@gmail.com
	X4G9J	Joker	C7	4	2019-10-17	19:00	Y9B4XW	sabeth45@gmail.com
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

– *Insert Table Studio*

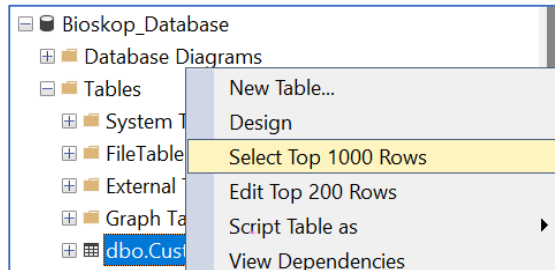
LAPTOP-U84N1VB6....base - dbo.Studio		
	noStudio	kapasitas_studio
▶	1	144
	2	187
	3	273
	4	170
	5	168
*	NULL	NULL

– *Insert Table Payment*

LAPTOP-U84N1VB6....ase - dbo.Payment		LAPTOP-U84N1VB6....ase - dbo.Payment		
	payment_id	payment_method	payment_date_time	harga_tiket
▶	3S6F8L	Kredit	2023-05-03 10:27:58.000	120000
	D5N7MK	Debit	2019-04-24 13:27:46.000	45000
	H2R9TJ	E-Wallet	2023-03-07 22:17:39.000	35000
	Y9B4XW	E-Wallet	2019-10-17 17:45:23.000	45000
*	NULL	NULL	NULL	NULL

4. Menampilkan Data Tabel

Tampilkan data tabel untuk memastikan bahwa data sudah benar di *input* kan dengan cara klik kanan tabel dan pilih 'Select Top 1000 Rows'.



– Table Customer

SQLQuery3.sql - L...N1VB6\Sadama (53) X

```
/****** Script for SelectTopNRows command from SSMS *****/
SELECT TOP (1000) [email]
, [nama_depan]
, [nama_belakang]
, [noHP]
FROM [Bioskop_Database].[dbo].[Customer]
```

100 %

Results Messages

	email	nama_depan	nama_belakang	noHP
1	ais99@gmail.com	Aisyah	Rahmah	083145129564
2	ken89@gmail.com	Kennard	Pangaribuan	081568783910
3	leon76@gmail.com	Leonard	Obadiaz	087419328892
4	sabeth45@gmail.com	Elisabeth	Putri	0895383848471
5	tata123@gmail.com	Oktaviana	Sadama	087834766951

– Table Movie

SQLQuery4.sql - L...N1VB6\Sadama (68) X

```
/****** Script for SelectTopNRows command from SSMS *****/
SELECT TOP (1000) [judul_film]
, [genre]
, [tahun_rilis]
, [durasi]
, [censor]
FROM [Bioskop_Database].[dbo].[Movie]
```

100 %

Results Messages

	judul_film	genre	tahun_rilis	durasi	censor
1	Avengers: Endgame	Action	2019	03:01:00.0000000	PG-13
2	Guardians of the Galaxy Vol. 3	Adventure	2023	02:29:00.0000000	PG-13
3	Joker	Crime	2019	02:02:00.0000000	R
4	Minions: The Rise of Gru	Komedi	2022	01:30:00.0000000	SU
5	Suzume no Tojimari	Adventure	2022	02:02:00.0000000	SU

– Table Ticket

SQLQuery5.sql - L...N1VB6\Sadama (68) X

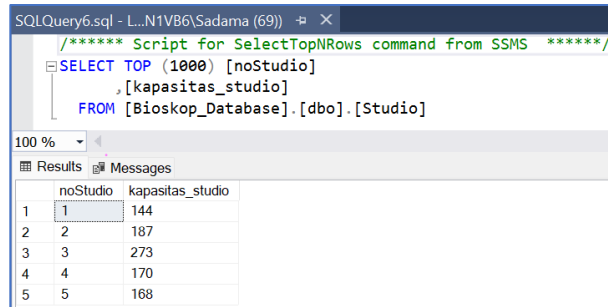
```
/****** Script for SelectTopNRows command from SSMS *****/
SELECT TOP (1000) [tiket_id]
, [judul_film]
, [row_seat]
, [noStudio]
, [tanggal_tayang]
, [jam_tayang]
, [payment_id]
, [email]
FROM [Bioskop_Database].[dbo].[Ticket]
```

100 %

Results Messages

	tiket_id	judul_film	row_seat	noStudio	tanggal_tayang	jam_tayang	payment_id	email
1	K8D4F	Guardians of the Galaxy Vol. 3	A10	3	2023-05-03	15:00	356F8L	tata123@gmail.com
2	MZV5H	Avengers: Endgame	B13	2	2019-04-24	13:30	DSN7MK	leon76@gmail.com
3	P6Z2L	Guardians of the Galaxy Vol. 3	A11	3	2023-05-03	15:00	356F8L	tata123@gmail.com
4	R9S7N	Suzume no Tojimari	E8	1	2023-03-08	20:15	H2R9TJ	ken89@gmail.com
5	X4G9J	Joker	C7	4	2019-10-17	19:00	Y9B4XW	sabeth45@gmail.com

– *Table Studio*



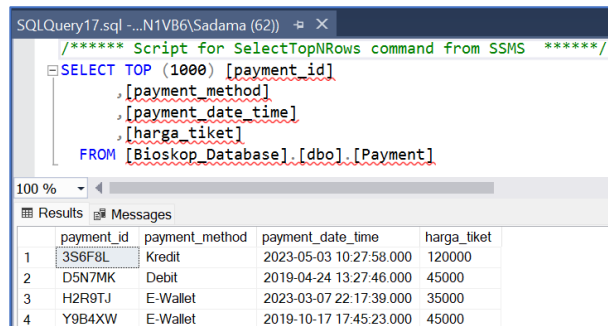
```

/***** Script for SelectTopNRows command from SSMS *****/
SELECT TOP (1000) [noStudio]
,[kapasitas_studio]
FROM [Bioskop_Database].[dbo].[Studio]

```

	noStudio	kapasitas_studio
1	1	144
2	2	187
3	3	273
4	4	170
5	5	168

– *Table Payment*



```

/***** Script for SelectTopNRows command from SSMS *****/
SELECT TOP (1000) [payment_id]
,[payment_method]
,[payment_date_time]
,[harga_tiket]
FROM [Bioskop_Database].[dbo].[Payment]

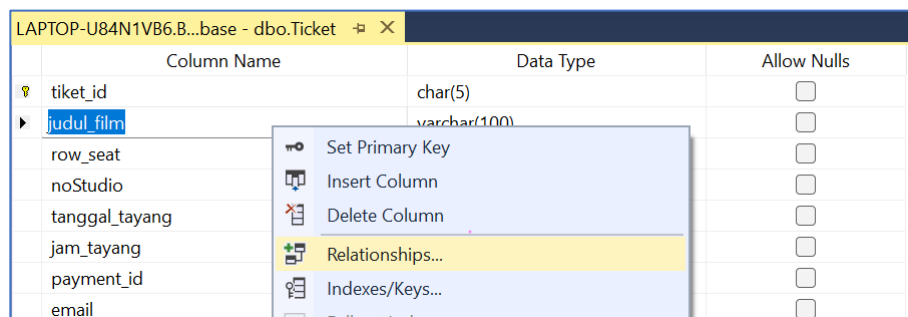
```

	payment_id	payment_method	payment_date_time	harga_tiket
1	3S6F8L	Kredit	2023-05-03 10:27:58.000	120000
2	D5N7MK	Debit	2019-04-24 13:27:46.000	45000
3	H2R9TJ	E-Wallet	2023-03-07 22:17:39.000	35000
4	Y9B4XW	E-Wallet	2019-10-17 17:45:23.000	45000

5. Membuat Relasi Antar Tabel

Relasi antar tabel dibuat dengan menghubungkan *foreign key* suatu tabel dengan *primary key* pada tabel lain. Berikut langkah-langkah yang dapat diikuti.

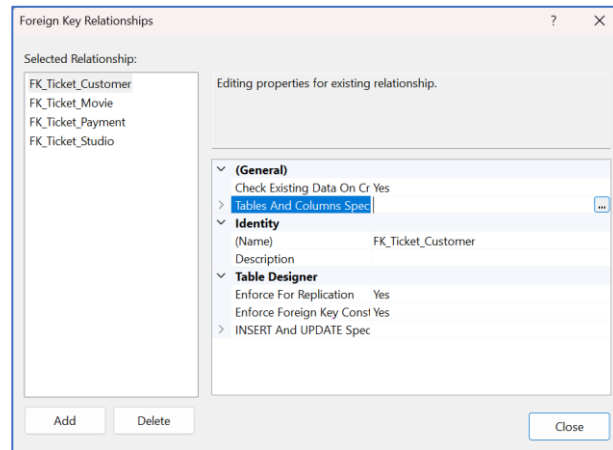
- Buka *table design foreign key*. Dalam studi kasus ini *foreign key* terletak pada *table ticket*.



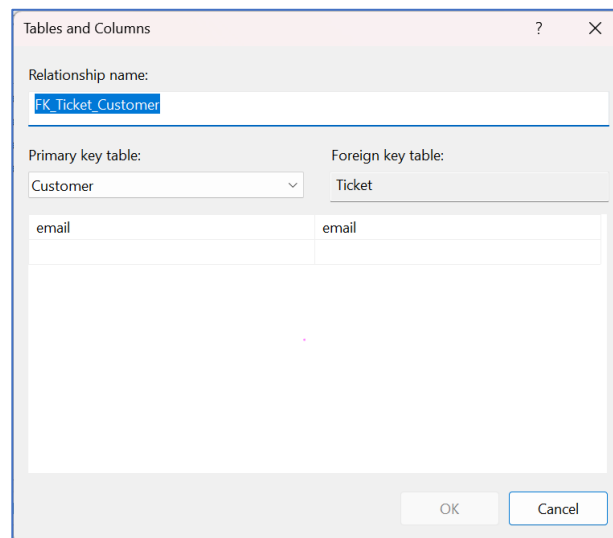
Column Name	Data Type	Allow Nulls
tiket_id	char(5)	<input type="checkbox"/>
judul_film	varchar(100)	<input type="checkbox"/>
row_seat		<input type="checkbox"/>
noStudio		<input type="checkbox"/>
tanggal_tayang		<input type="checkbox"/>
jam_tayang		<input type="checkbox"/>
payment_id		<input type="checkbox"/>
email		<input type="checkbox"/>

Klik kanan tabel dan pilih '*Relationship*'.

- Klik '*Add*' dan klik titik tiga pada *Tables and Columns Specification*.

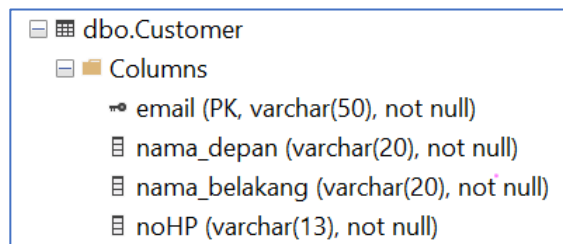


- Isikan *foreign key* dari suatu tabel dan *primary key* tabel yang akan dihubungkan atau dibuat relasi. Nama relasi akan secara *default* terisi dan dapat pula diubah. Klik OK.



- *Expand* tabel dan *expand* folder untuk memastikan bahwa relasi antar tabel sudah benar dibuat.

a) *Table Customer*



b) *Table Movie*

dbo.Movie
Columns
judul_film (PK, varchar(100), not null)
genre (varchar(50), not null)
tahun_rilis (int, not null)
durasi (time(7), not null)
sensor (varchar(5), not null)

c) *Table Ticket*

dbo.Ticket
Columns
tiket_id (PK, char(5), not null)
judul_film (FK, varchar(100), not null)
row_seat (varchar(3), not null)
noStudio (FK, int, not null)
tanggal_tayang (date, not null)
jam_tayang (varchar(5), not null)
payment_id (FK, char(6), not null)
email (FK, varchar(50), not null)

d) *Table Studio*

dbo.Studio
Columns
noStudio (PK, int, not null)
kapasitas_studio (int, not null)

e) *Table Payment*

dbo.Payment
Columns
payment_id (PK, char(6), not null)
payment_method (varchar(10), not null)
payment_date_time (datetime, not null)
harga_tiket (int, not null)

DAFTAR PUSTAKA

- Connolly, T.M., & Begg, C.E. 2002. *Database systems: A practical approach to design, implementation, and management*. Harlow, England: Addison-Wesley.
- Mustofa, Zaenal. 2021. *Basis Data Dan DBMS*. Diakses pada 9 Mei 2023, dari <https://teknik-informatika-s1.stekom.ac.id/informasi/baca/Basis-Data-dan-DBMS/6d492170753211fcde587882d77e1e8dcce1bc27>