**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

1. **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
* Dokumenstasi: *Screenshot*

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

# BAB II DASAR TEORI

1. **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 teroganisasi secara konsisten dan dapat diakses dengan mudah.

1. ***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.

1. ***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.

1. **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.

* 1. 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.

* 1. 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.

* 1. 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.

* 1. 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

1. ***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 diiginkan.
  6. *Customer* memilih *row seat* mana saja yang ingin ditempati.
  7. *Customer* melakukan pembayaran.
  8. C*ustomer* 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 *censor* usia.
  6. Adapun infomasi 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 | *Customer* | #\* Email  \* Nama depan  \* Nama belakang  \* No HP |
| 2 | *Ticket* | #\* *Ticket id*  \* Judul film  \* *Row seat*  \* No studio  \* Tanggal tayang  \* Jam tayang  *\* Payment id*  \* Email |
| 3 | *Movie* | #\* Judul film  \* Genre  \* Tahun rilis  \* Durasi  \* *Censor* |
| 4 | *Payment* | #\* *Payment id*  \* *Payment method*  \* *Payment date time*  \* 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**

* 1. *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**

* 1. *Studio – Ticket*

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

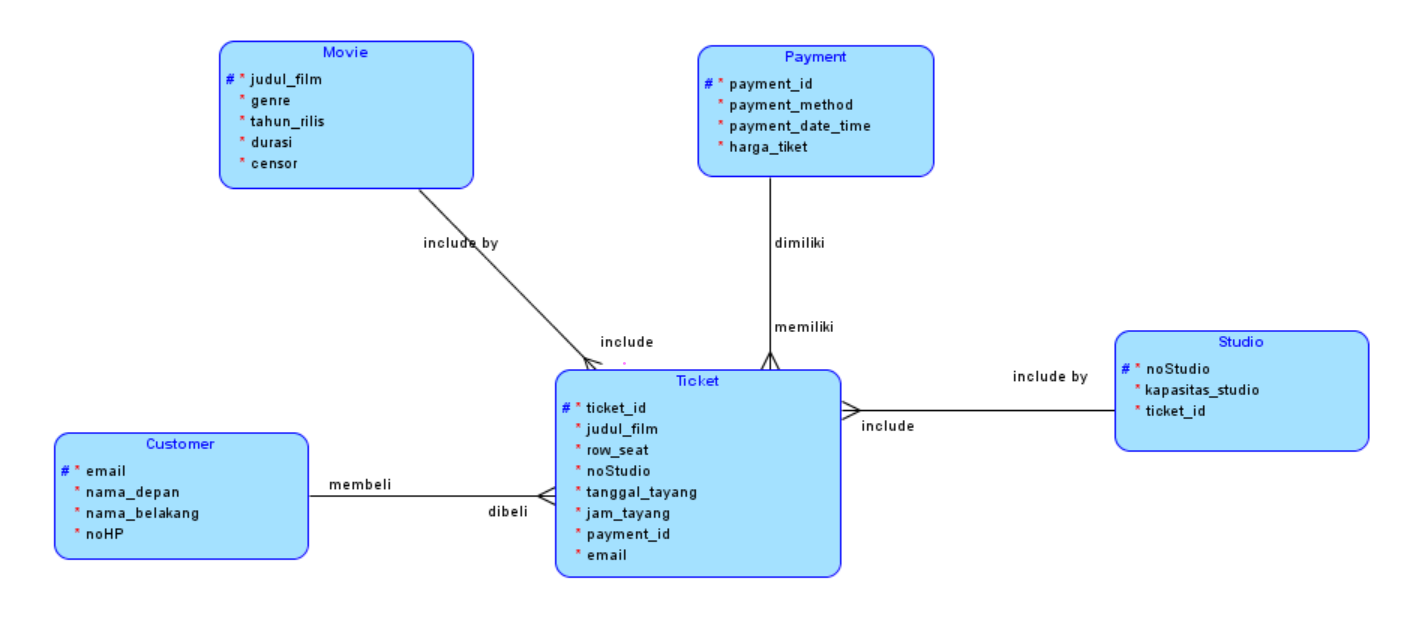
**Kardinalitas: One to Many**

* 1. *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**

1. ***Logical Model***



Penjelasan

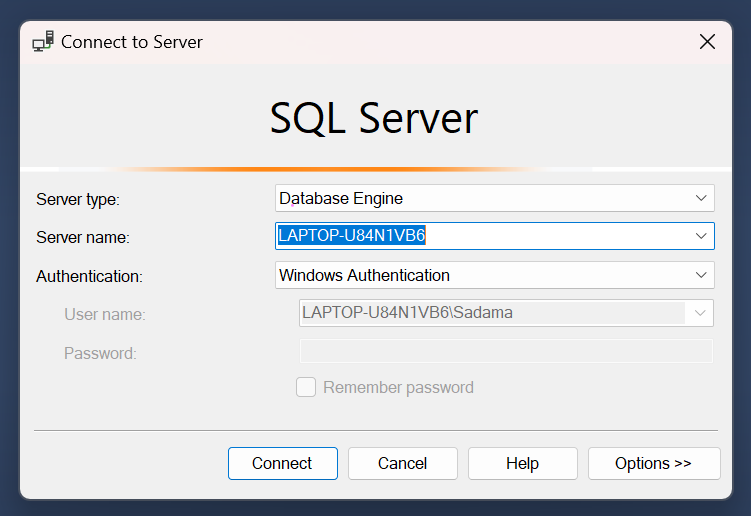
*Logical model* di atas menunjukkan relasi antar tabel dimana entitas *ticket* mengandung semua *foreign key* dari entitas *customer, movie, payment,* dan studio dengan rincian sebagai berikut:

* 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.*

1. ***Database***

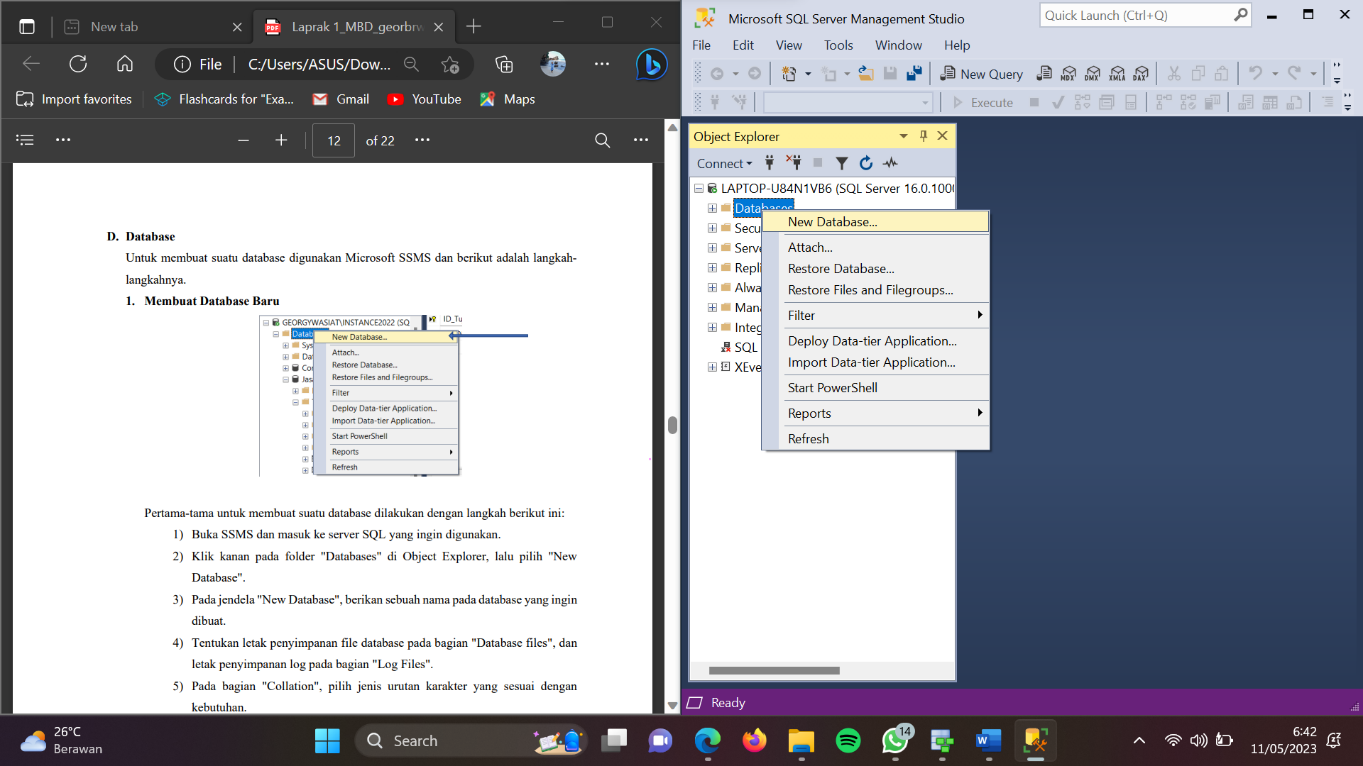
*Database* dibuat menggunakan Microsoft SQL Server dengan langkah-langkah sebagai berikut:

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

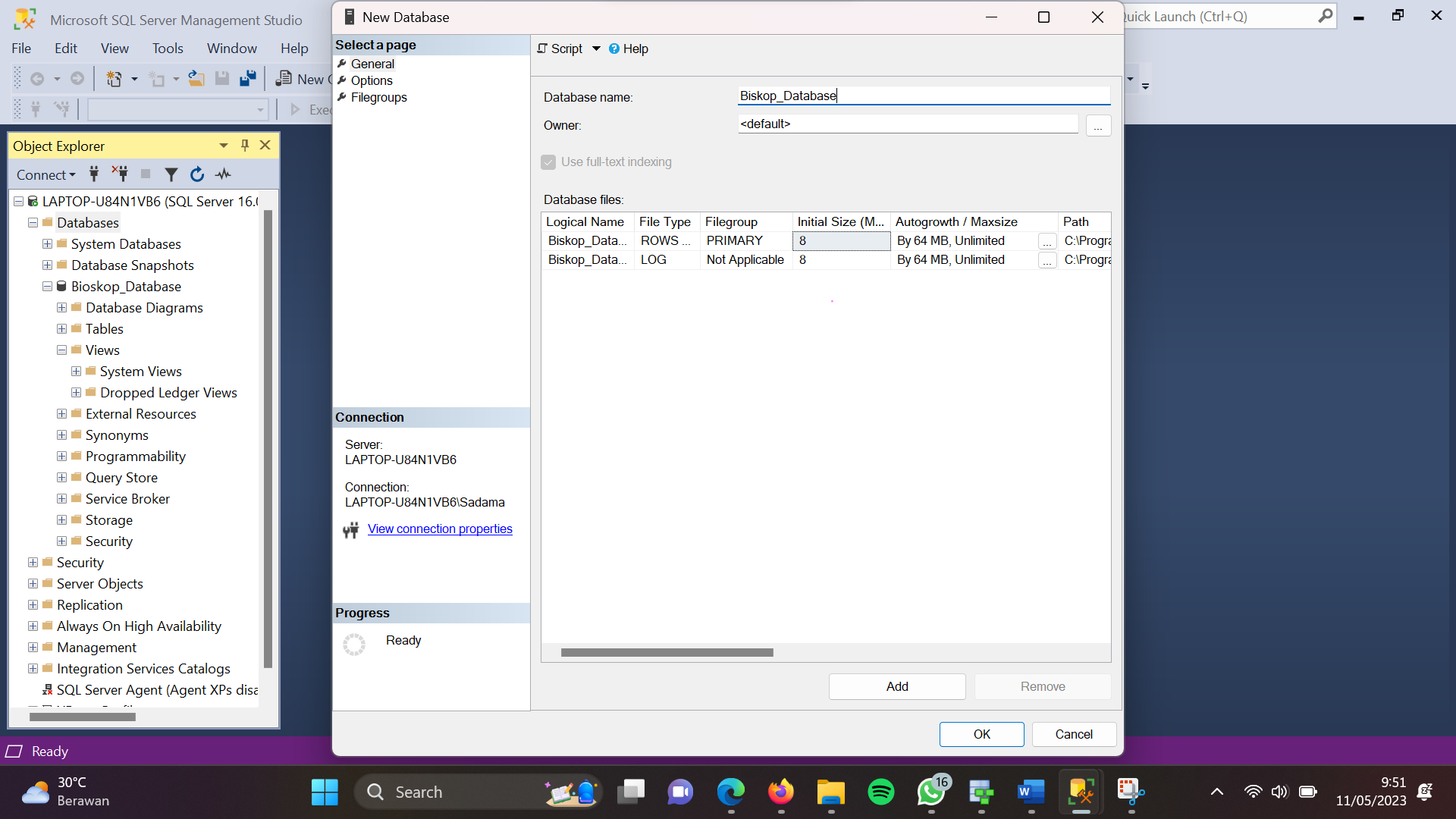


Dalam pembuatan database bioskop ini saya menggunakan *server name default* bawaan laptop. Setelah itu klik *Connect*. Pastikan untuk *connect* dengan *server* yang sama apabila ingin menggunakan *database* yang sebelumnya sudah dibuat dalam *server* tersebut.

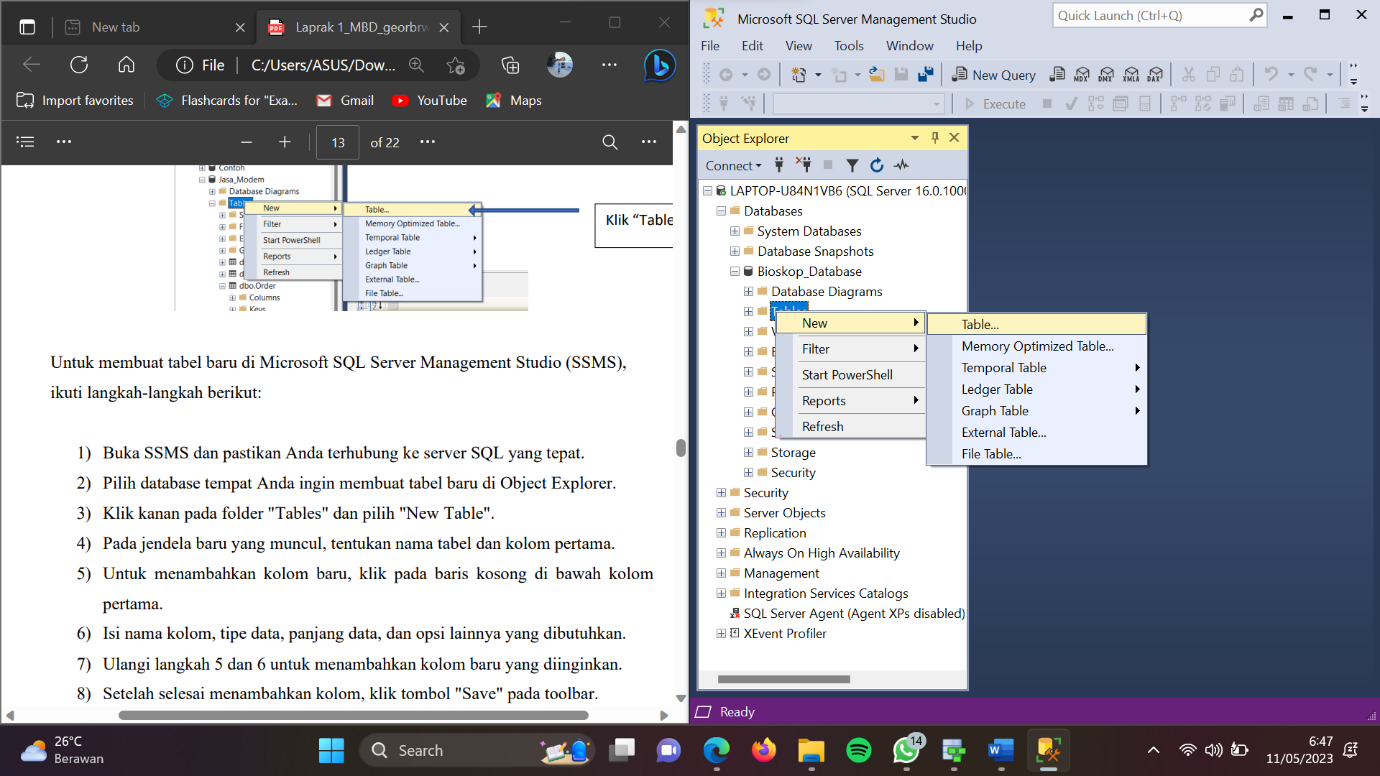
* Untuk membuat *database,* klik kanan pada bagian folder *database* dan klik *new database.*



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



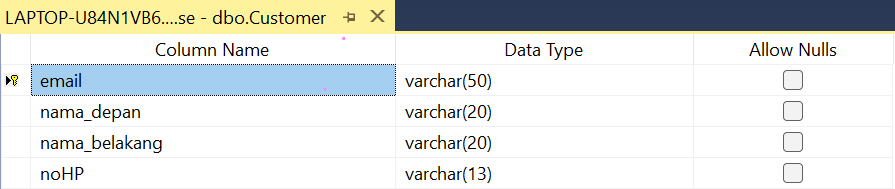
* Muncul *database* *‘Bioskop\_Database’*. Untuk membuat tabel, *expand Bioskop\_Database* dan klik kanan pada file *Tables* 🡪 *New* 🡪 *Table*.



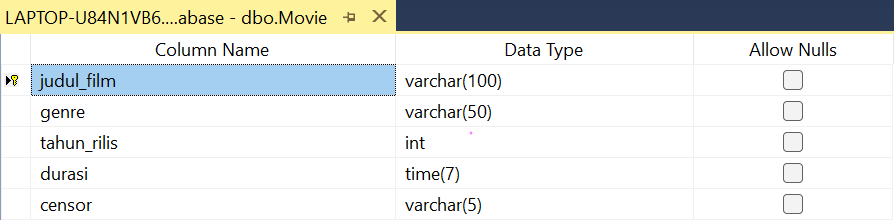
* 1. Menambahkan Atribut dan Tipe Data

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

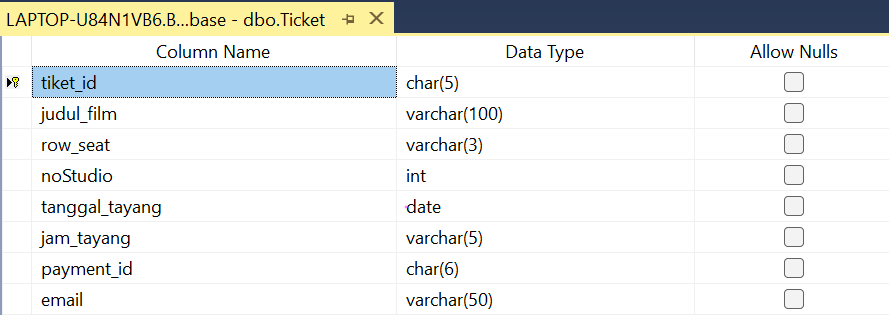
* *Table* *Customer*



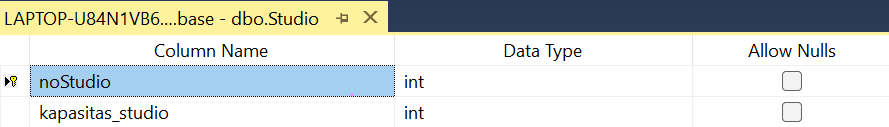
* *Table* *Movie*



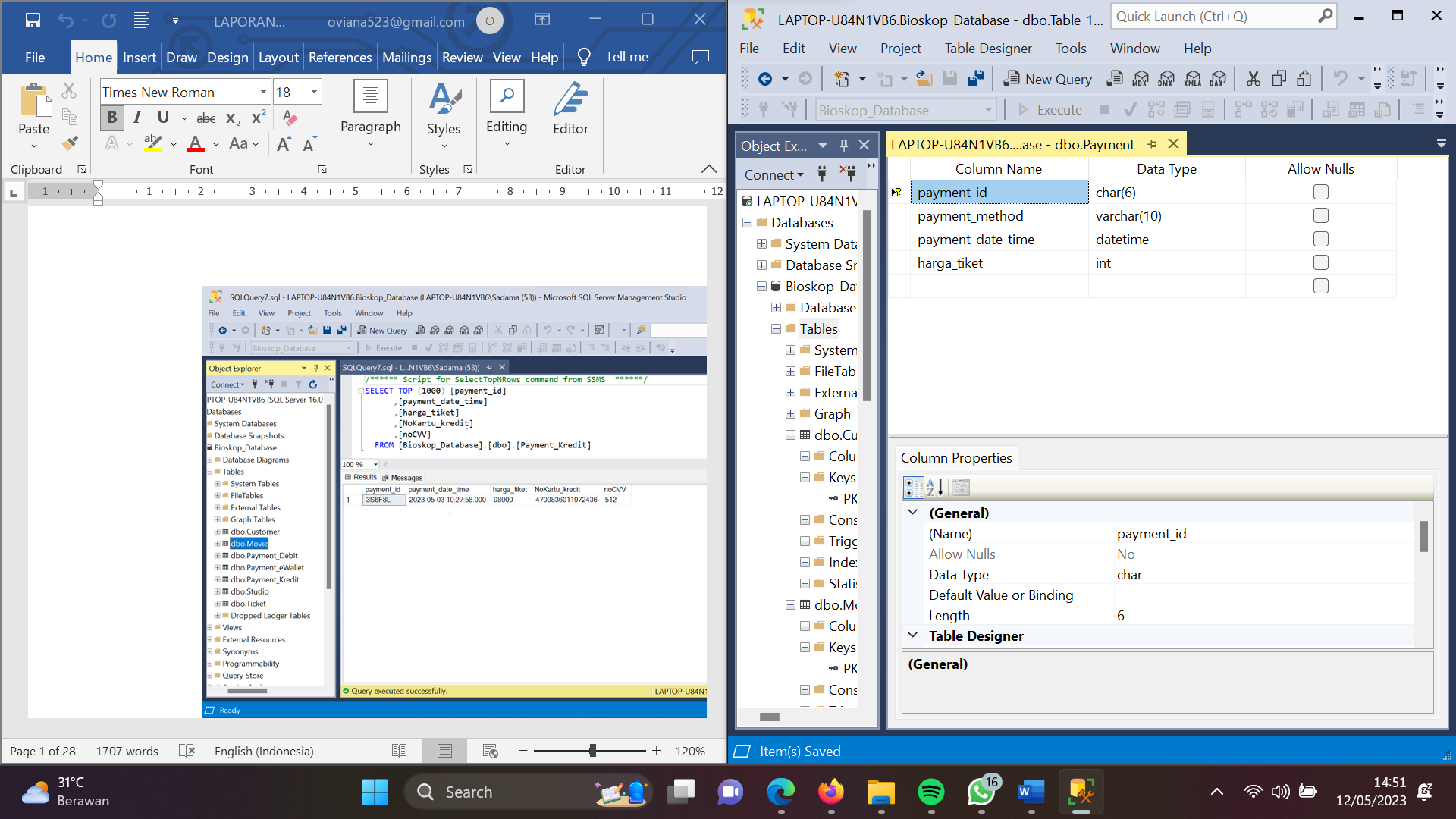
* *Table* *Ticket*



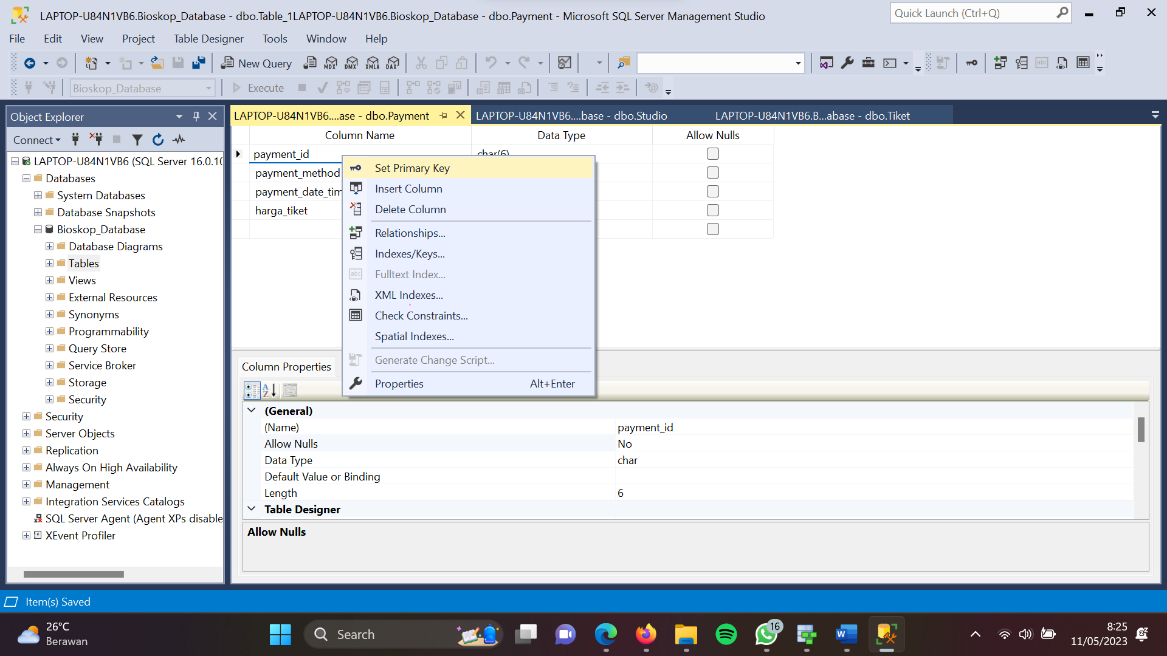
* *Table* *Studio*



* *Table Payment*

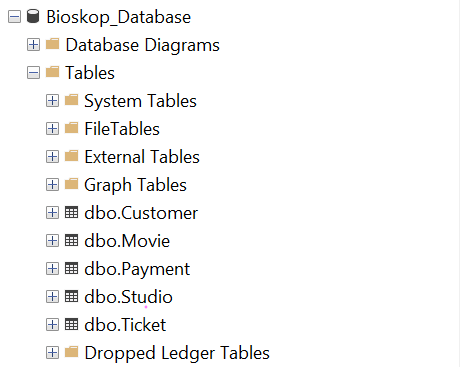


Tambahkan pula *primary key* dari setiap UID tabel dengan klik kanan kolom pada tabel yang ingin dijadikan UID kemudian klik *‘Set Primary Key’.*

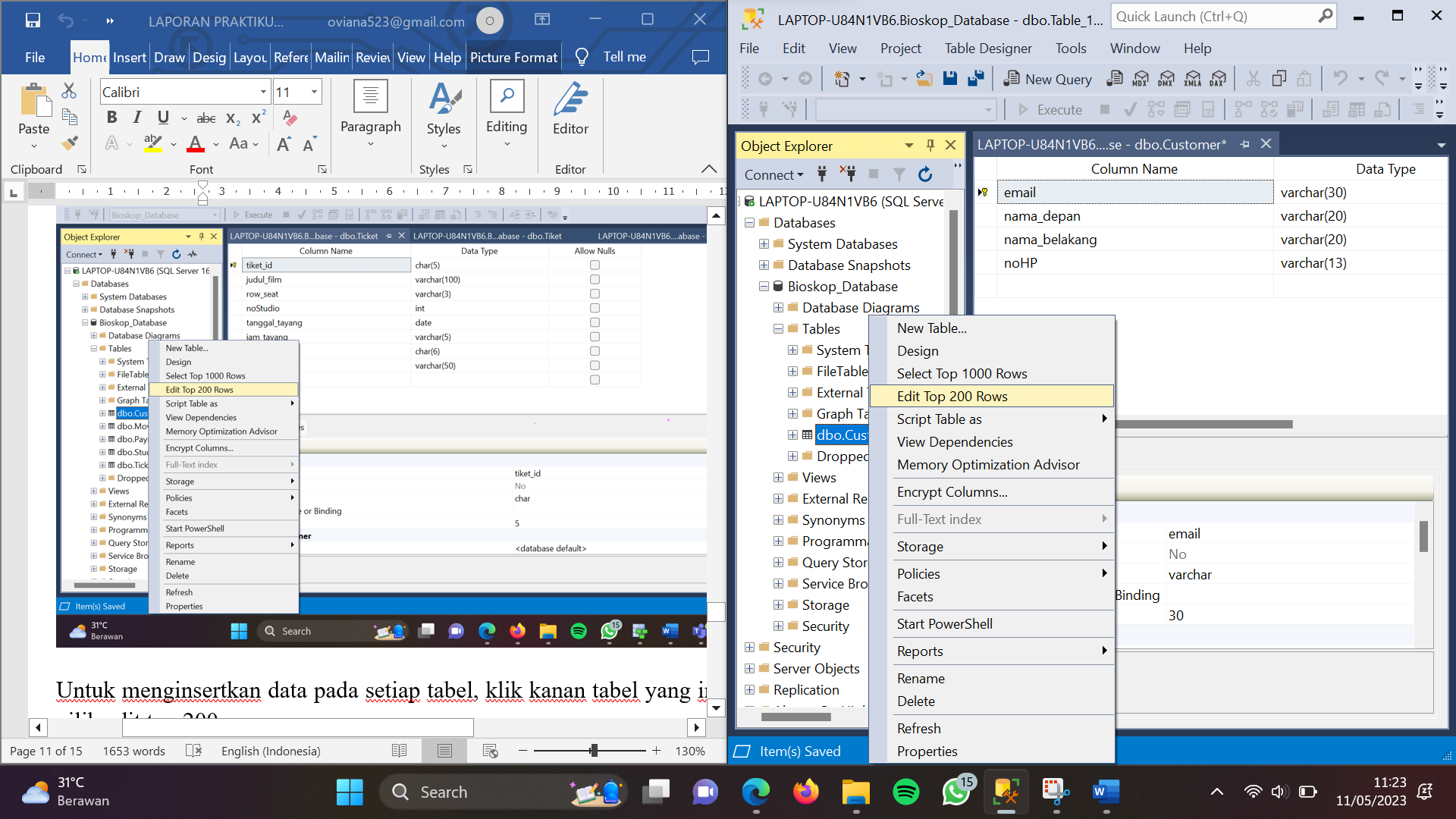


* 1. *Insert* Data Tabel

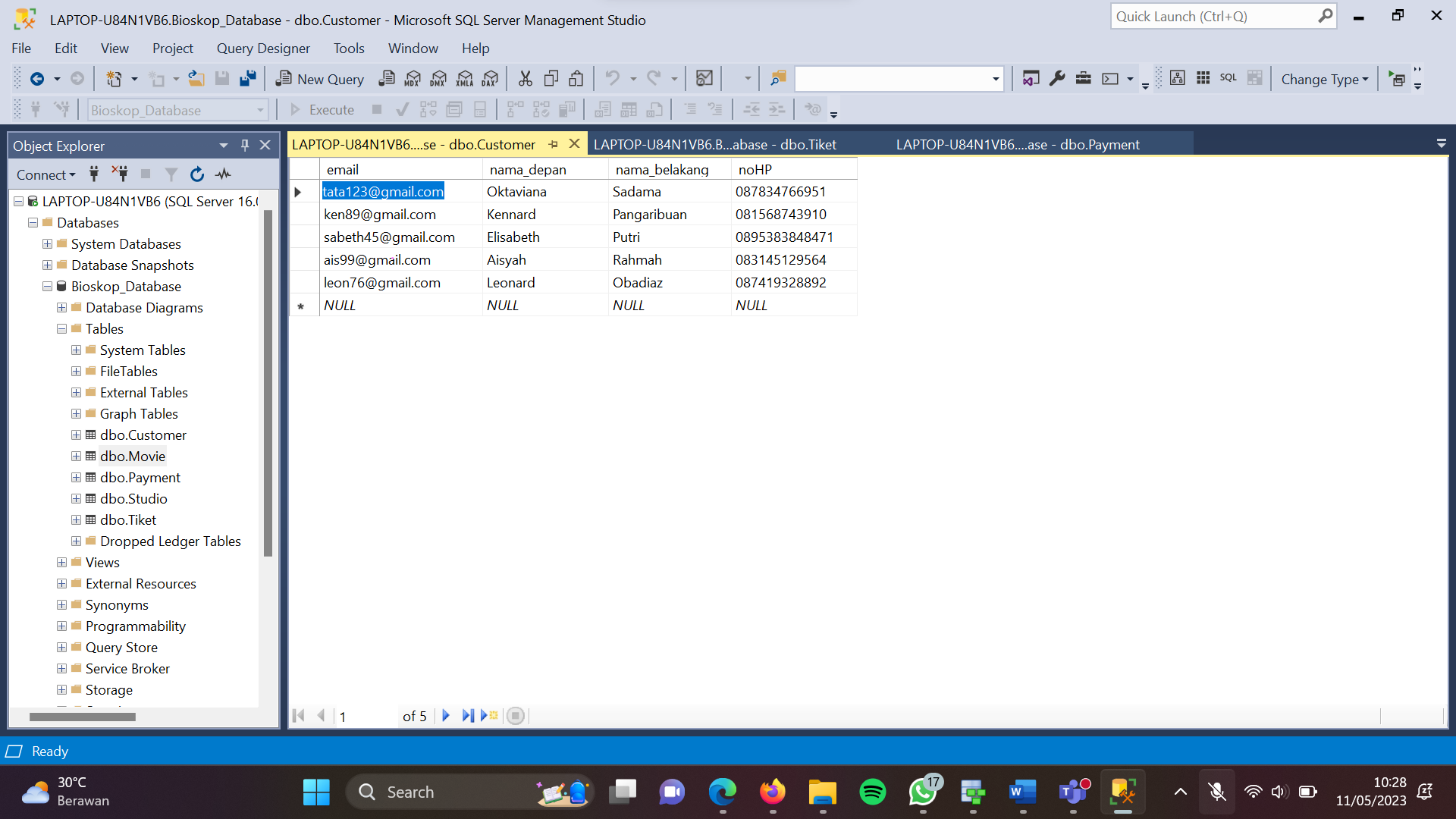
Apabila kolom setiap tabel sudah dibuat dan ditentukan tipe data, serta ketentuan *allow nulls* nya maka tabel-tabel tersebut akan muncul pada folder *table,* sebagai berikut:



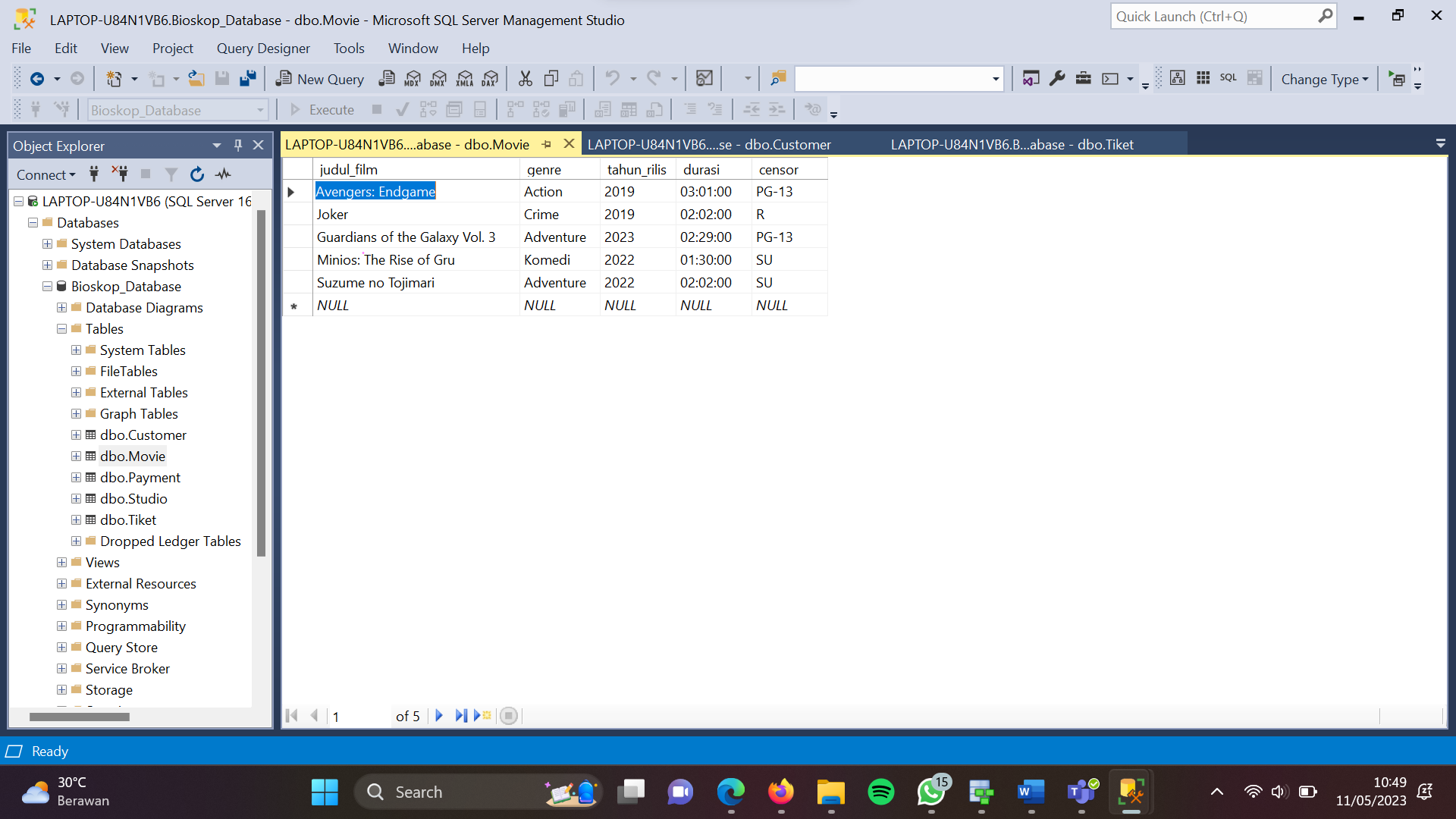
Untuk meng*-insert-*kan data pada setiap tabel, klik kanan tabel yang ingin diinsertkan data dan pilih *‘Edit Top 200 Rows’*.



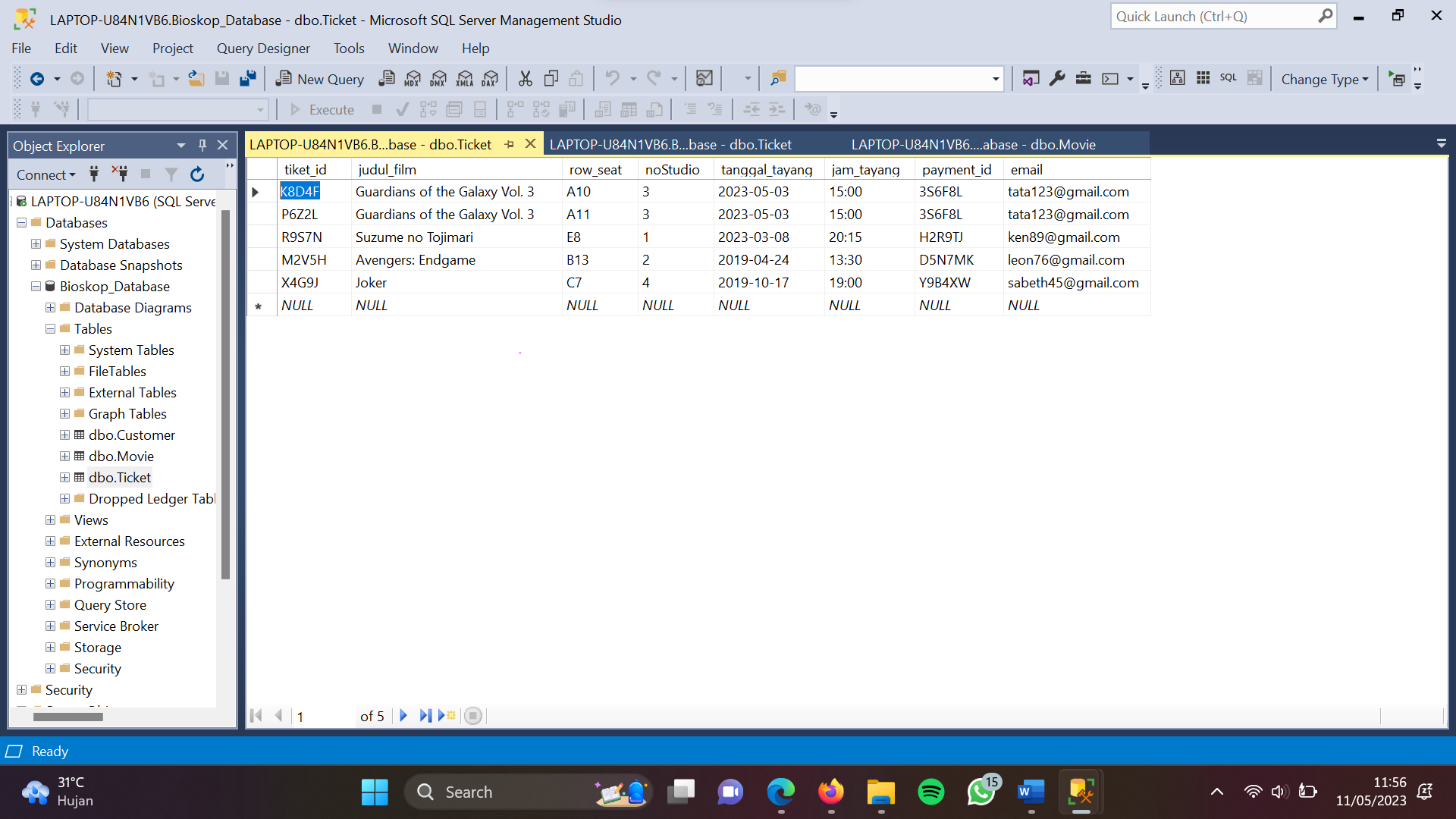
* *Insert Table Customer*



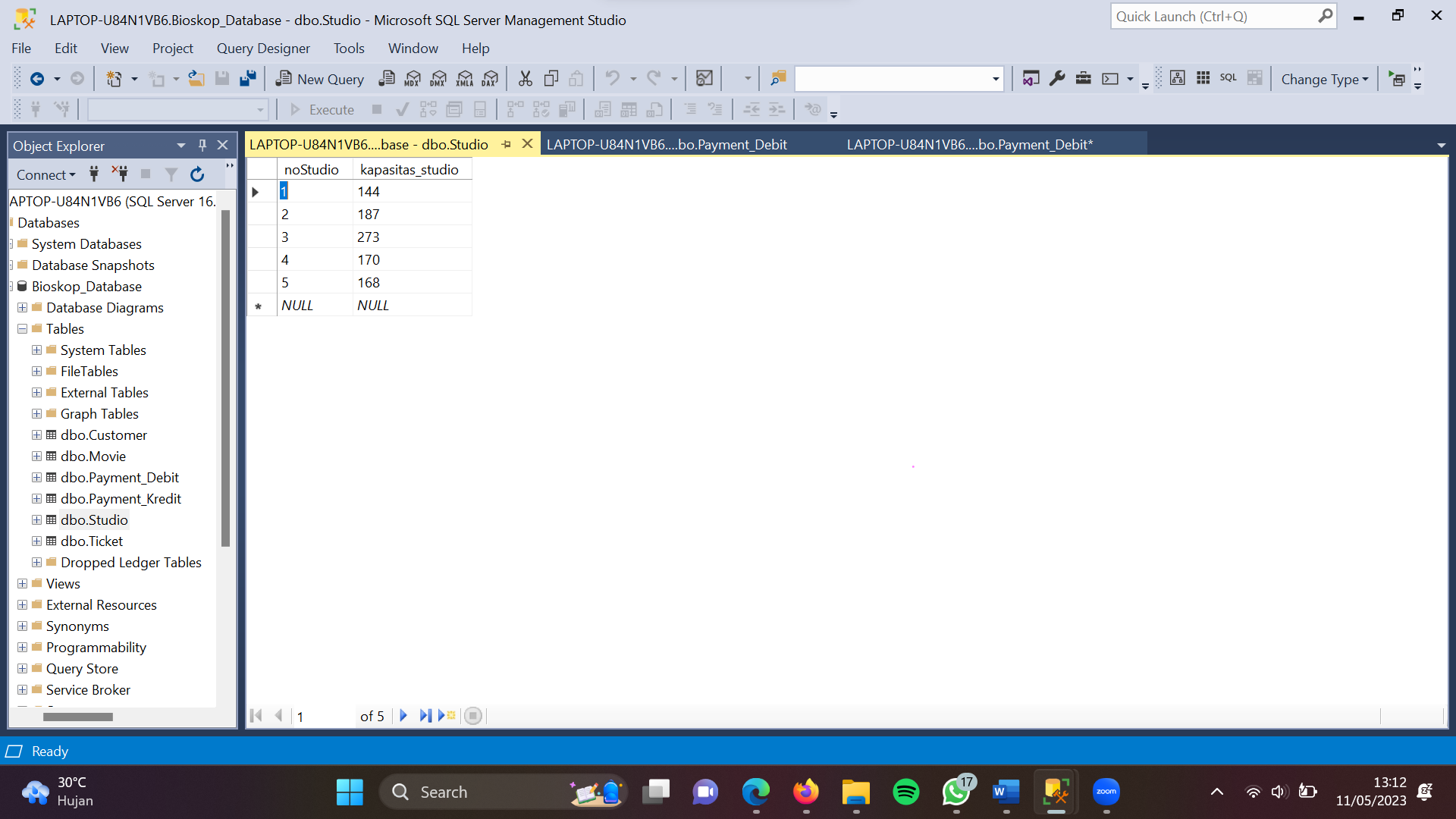
* *Insert Table Movie*



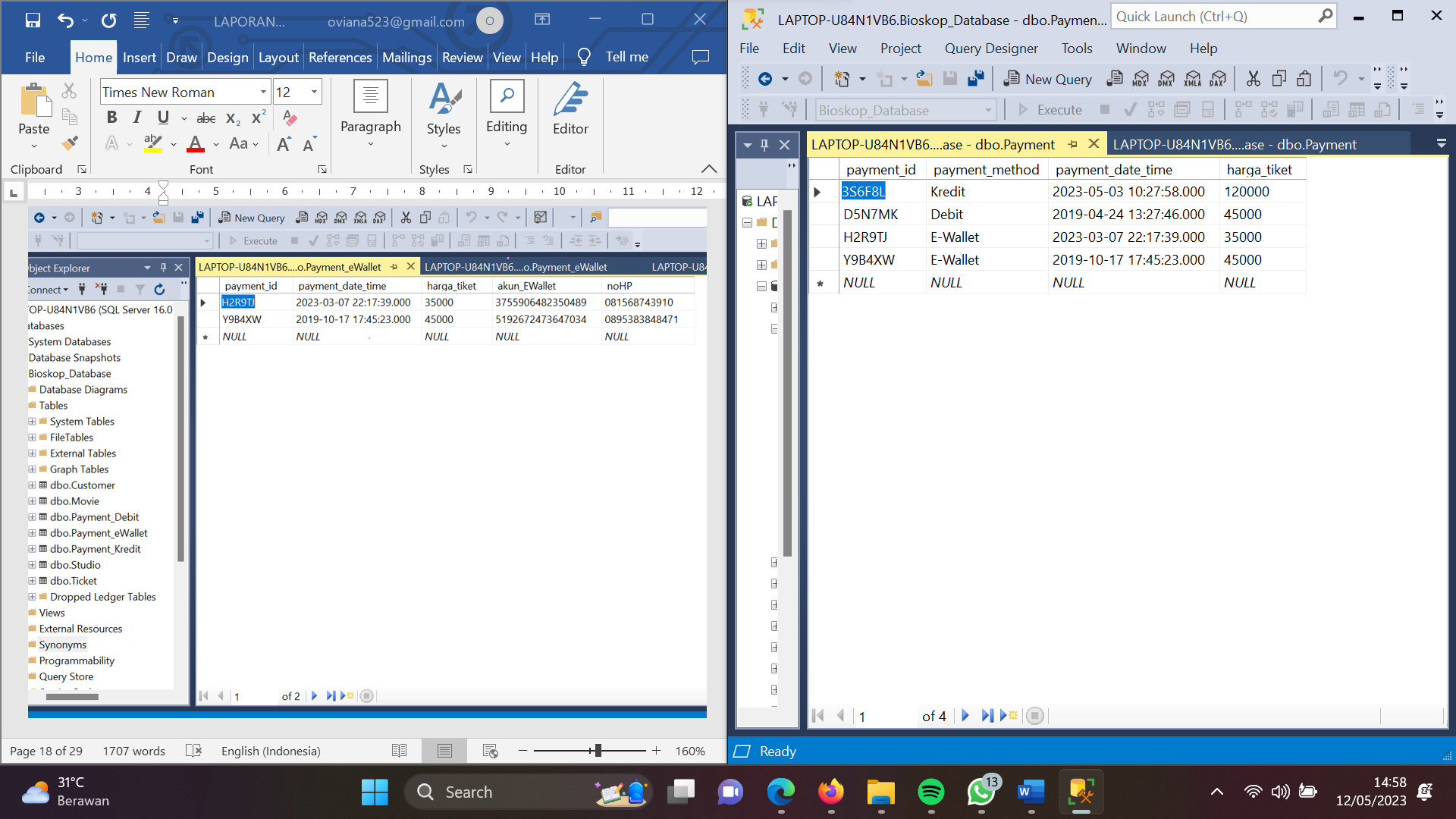
* *Insert Table* *Ticket*



* *Insert Table* *Studio*

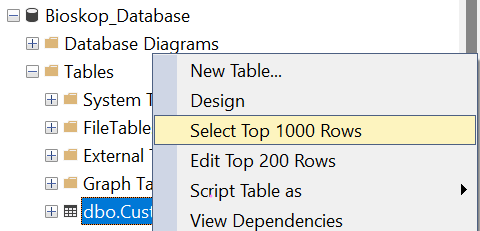


* *Insert Table Payment*

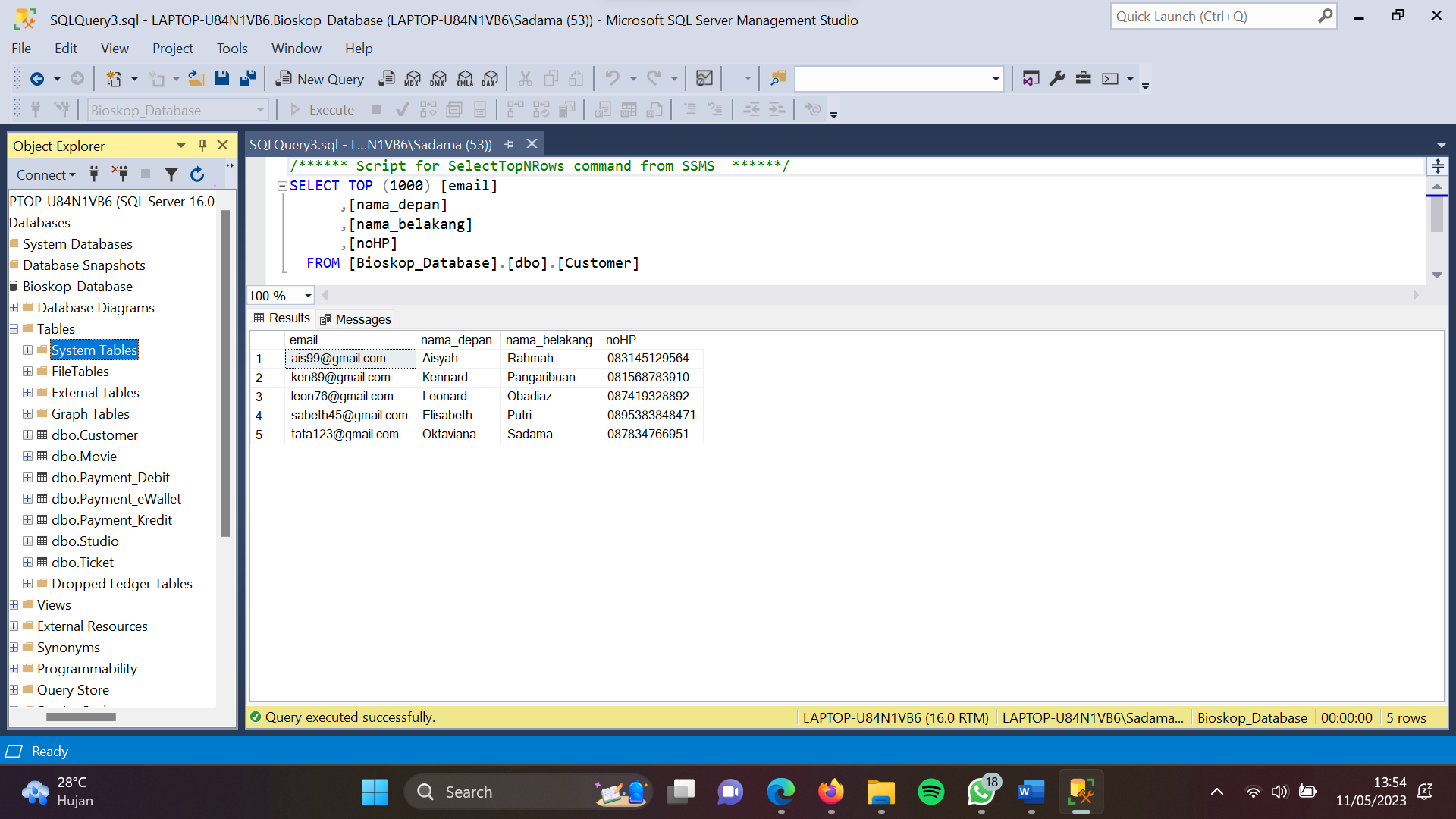


* 1. Menampilkan Data Tabel

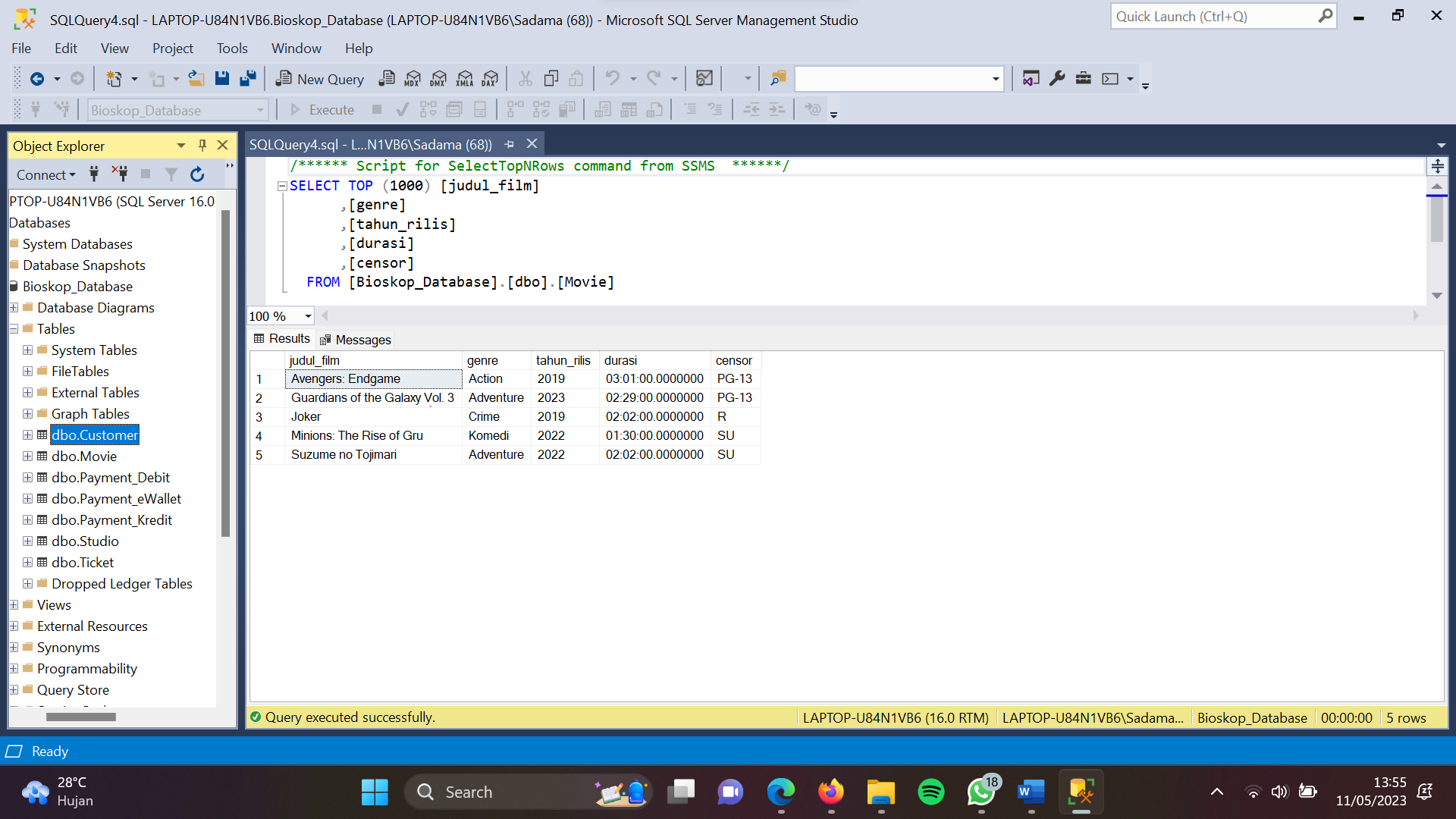
Tampilkan data tabel untuk memastikan bahwa data sudah benar di *input* kan dengan cara klik kanan tabel dan pilih *‘Select Top 1000 Rows’.*



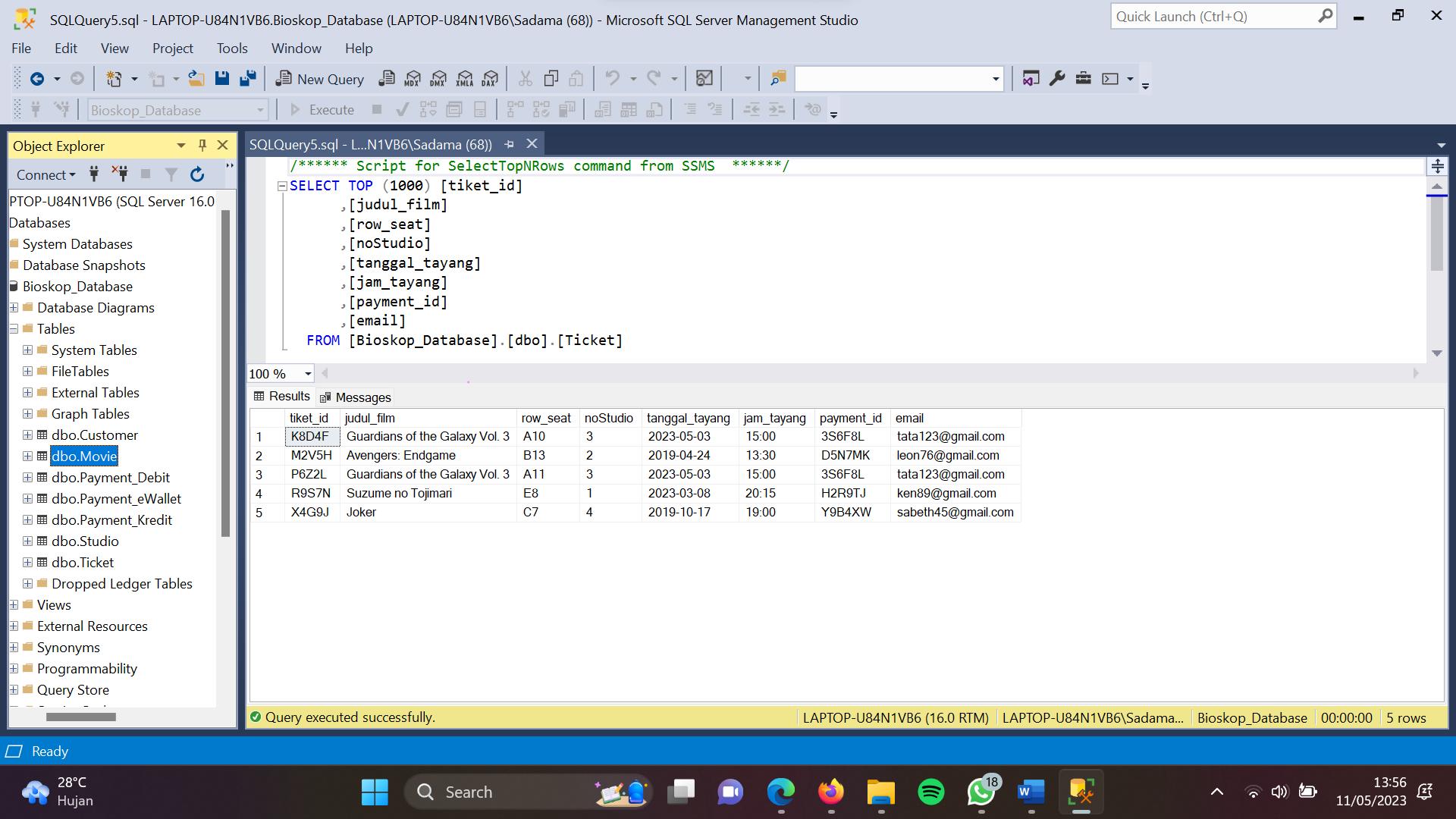
* *Table Customer*



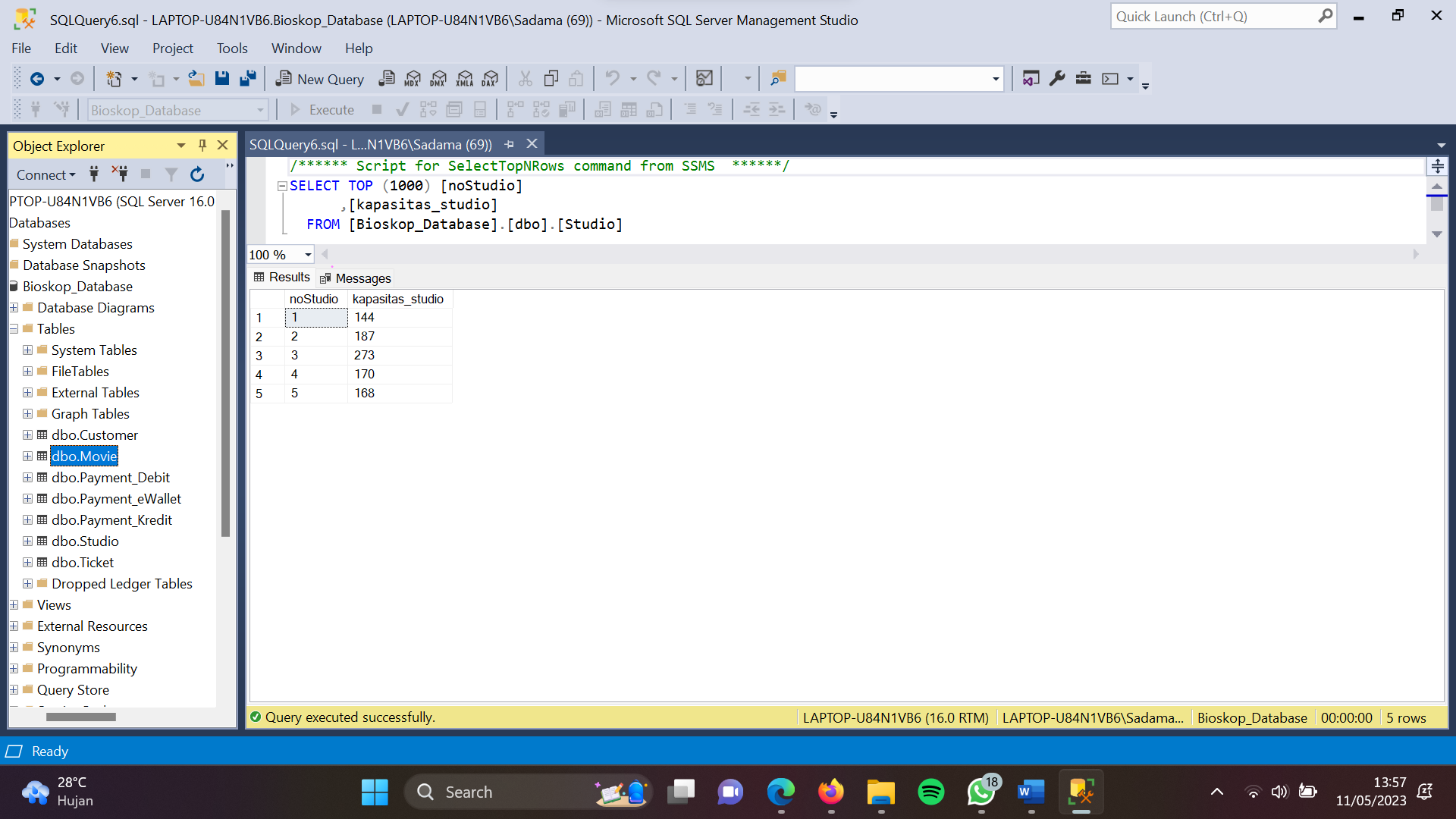
* *Table Movie*



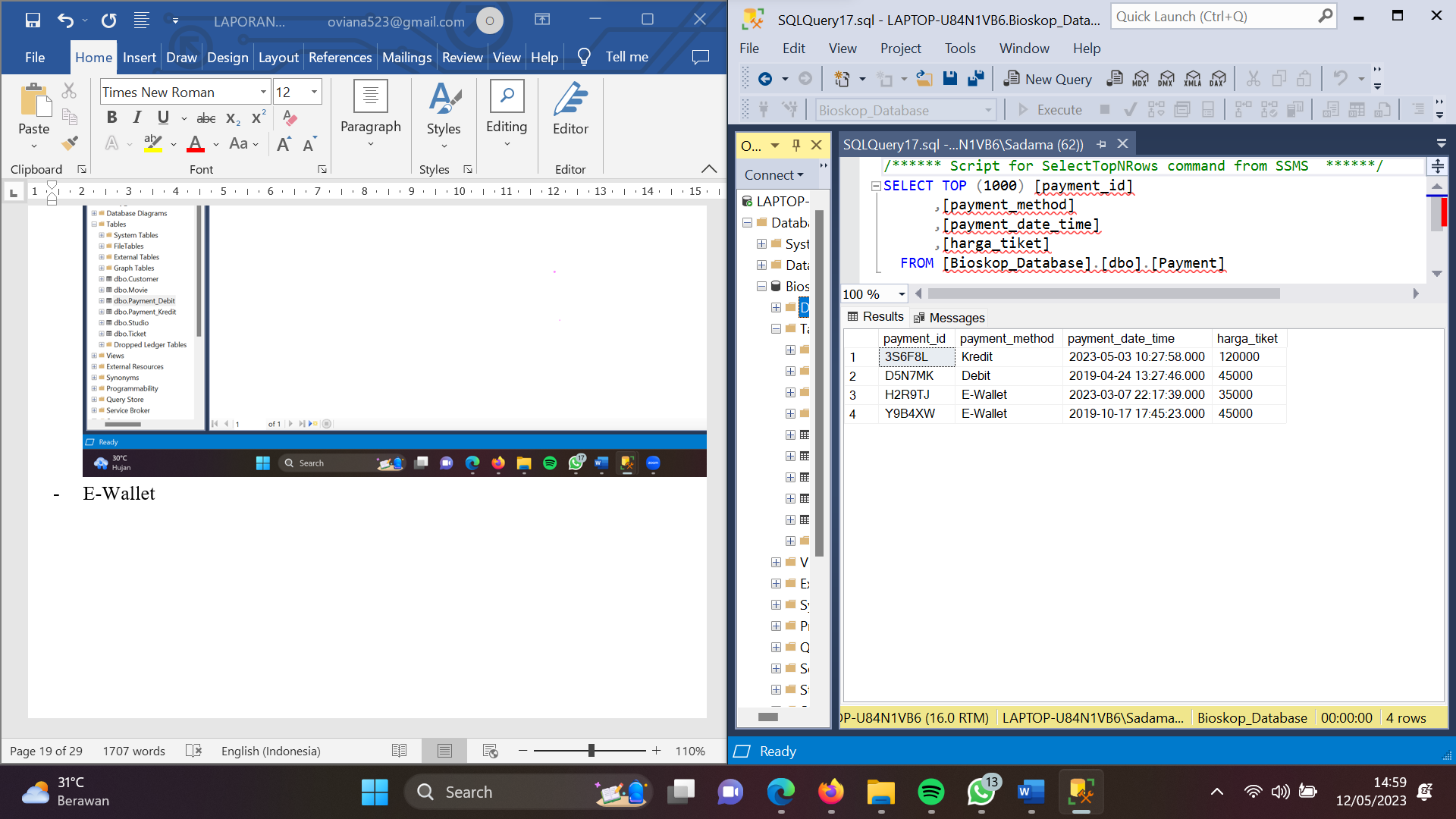
* *Table Ticket*



* *Table Studio*



* *Table Payment*



# 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 *tabel design foreign key.* Dalam studi kasus ini *foreign key* terletak pada *table ticket.*

# 

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

# *Table Customer*

# 

# *Table Movie*

# 

# *Table Ticket*

# 

# *Table Studio*

# 

# *Table Payment*

# 

# 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/6d492170753211f cde587882d77e1e8dcce1bc27