# **MODUL IV : MENGHUBUNGKAN KE DATABASE MYSQL**



**Kompetensi Pengetahuan**

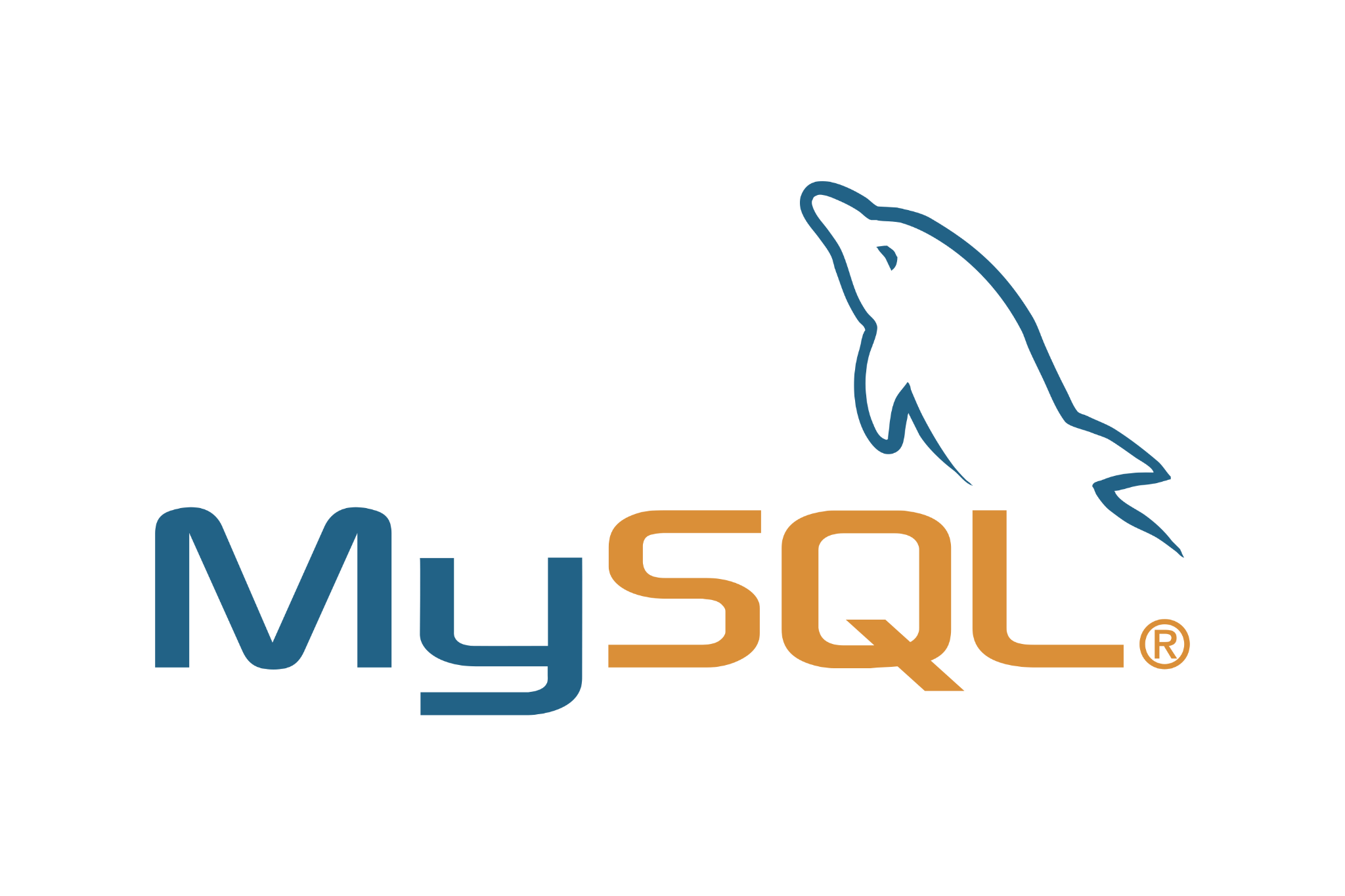
1. Peserta dapat memahami database SQL.
2. Peserta dapat memahami cara integrasi Spring Boot dan database MySQL

**Kompetensi Keterampilan**

1. Peserta dapat menginstall MySQL Server
2. Peserta dapat membuat database di dalam MySQL Server
3. Peserta dapat mengintegrasikan Spring Boot dengan MySQL Server dan database yang telah dikembangkan

Pada modul ini akan dibahas mengenai database MySQL dan integrasi dari project yang sebelumnya telah dikembangkan dengan Database MySQL.

## **Database dan MySQL**

****

Database merupakan kumpulan data yang terorganisir, yang umumnya disimpan dan diakses secara elektronik dari suatu sistem komputer. Adapun berbicara mengenai database maka perlu diketahui juga RDBMS dan SQL.

* **Relational database management system (RDBMS)** adalah sebuah program komputer (atau secara lebih tipikal adalah seperangkat program komputer) yang dirancang untuk mengatur/memanajemen sebuah basis data sebagai sekumpulan data yang disimpan secara terstruktur, dan melakukan operasi-operasi atas data atas permintaan penggunanya.
* **MySQL** adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Setiap pengguna dapat secara bebas menggunakan, mendistribusikan, dan membuat karya turunan dari MySQL. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya; SQL (Structured Query Language). Contoh Aplikasi RDBMS lainnya adalah PostgresQL, SQLite, Oracle DB, dll.
* **SQL** adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis yang digunakan untuk RDBMS.

Langkah-langkah dalam instalasi MySQL Server serta membuat database adalah sebagai berikut

1. Untuk mempermudah dalam menggunakan MySQL download MySQL Workbench sebagai Database Client untuk MySQL Server [https://dev.mysql.com/downloads/workbench](https://dev.mysql.com/downloads/workbench/).

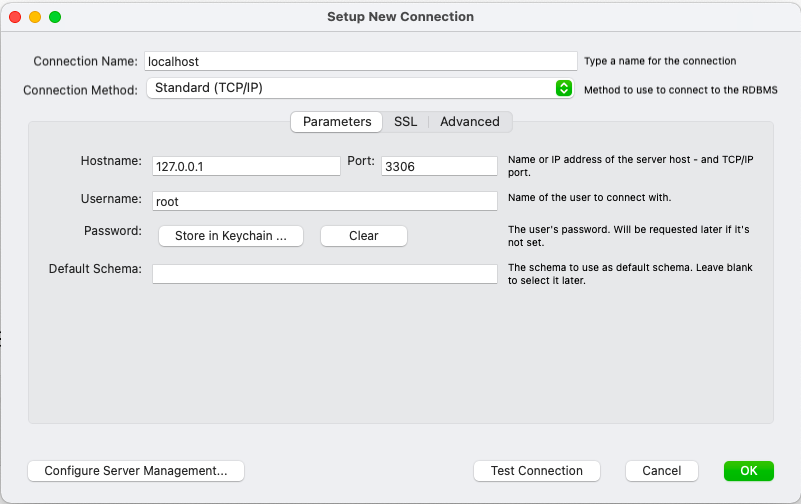


1. Lakukan instalasi MySQL Workbench dan MySQL Server, pada proses instalasi akan dilakukan set password dari MySQL Server dengan username default **root.** Setelah proses instalasi selesai buka aplikasi MySQL Workbench.

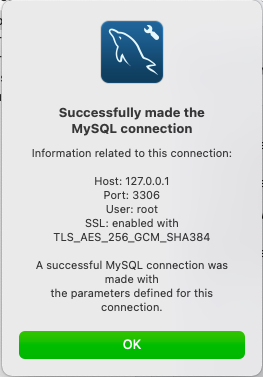


1. Masukkan credential untuk connect ke dalam MySQL Server di dalam local

* Connection Name : Localhost
* Hostname : 127.0.0.1
* Port : 3306
* Username : root (default)
* Password : Masukkan password sesuai yang di setting ketika proses instalasi MySQL Server & Workbench



1. Pilih tombol Test Connection, maka akan muncul dialog **Successfully Made the MySQL Connection**



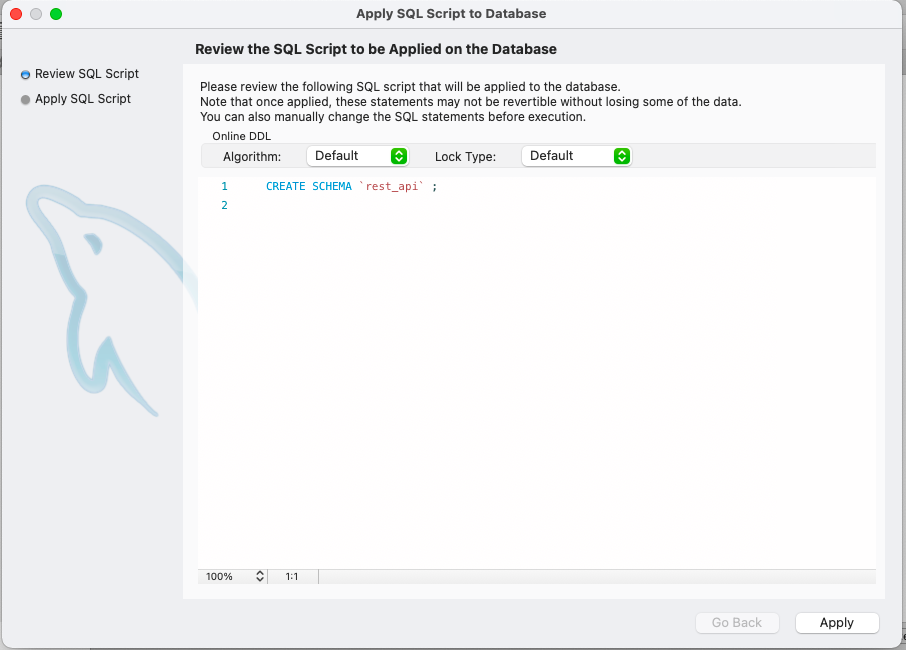
1. Pilih tombol OK, lalu akan muncul IDE Client Database MySQL klik tab Administration, lalu pilih menu Server Status, pastikan **Server Status Running**



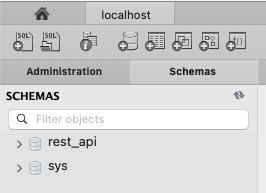
1. Buat sebuah database baru dengan pilih tab **schema**, lalu klik **kanan**, lalu pilih **create schema**



1. Isikan nama schema dengan rest\_api (nama bisa apapun), setelah itu klik tombol **apply**
2. Lalu klik tombol apply lagi



1. Pastikan database baru dengan nama rest\_api sudah muncul di window schema untuk menunjukan bahwa telah berhasil membuat database baru



## **JPA (ORM) dan Hibernate**

ORM (Object Relational Mapping) adalah mekanisme interaksi dengan database dengan menggunakan sebuah Oriented Object Class seperti membuat sebuah tabel dengan sebuah model class pojo.



Pada sebuah aplikasi java spring boot dalam berinteraksi dengan sebuah database seperti MySQL maka melalui sebuah connector jdbc. 

* **JDBC (Java Database Connectivity)** adalah sebuah API Java yang digunakan untuk menghubungkan dan manipulasi ke dalam database
* **JPA**  **(Java Persistence API)** adalah framework yang didedikasikan untuk pengelolaan data relasional (menggunakan lapisan persistensi) dalam aplikasi Java.
* **Hibernate** adalah adalah implementasi khusus dari JPA tersebut.

1. Pada modul **spring-api** pada pom.xml tambahkan dependency JPA dan MySQL Connector sehingga seperti berikut
   * Dependency **spring-boot-starter-data-jpa** : Digunakan untuk menggunakan JPA dependency seperti mendapatkan data, mengubah data, menambah data di dalam database.
   * Dependency **mysql-connector-java :** Digunakan untuk mengkoneksikan project Spring ke MySQL Server



**Pom.xml (modul spring-api)**

. . .

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<scope>runtime</scope>

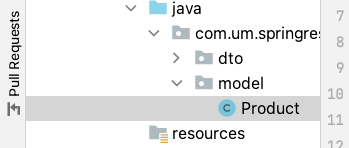
</dependency>

</dependencies>

. . .

1. Pada file product dalam package model di dalam modul product-api jadikan sebagai ORM untuk model dalam membuat sebuah table ke dalam sebuah database adapun annotation yang ditambahkan adalah
   * **@Entity :** Mendeklarasi bahwa class tersebut akan digunakan untuk *mapping* sebuah tabel dalam database
   * **@Table** : Memberikan sebuah nama table yang akan digunakan dalam database
   * **@Id** : digunakan untuk mendefinisikan primary key
   * **@GeneratedValue :** digunakan untuk generate value id
   * **@NotBlank** : digunakan agar isi pada field nya NOT NULL
   * **@Column** : digunakan untuk memberikan pengaturan tambahan pada kolom tersebut
   * **@Temporal** : digunakan untuk merubah data berupa tanggal dan waktu dari database dan ke database
   * **@CreatedDate** : digunakan untuk mendapatkan waktu dibuatnya data
   * **@LastModifiedDate :** digunakan untuk mendapatkan waktu terakhir data diedit

Selain annotation diatas tambahkan implementasi interface Serializible agar pengiriman model object lebih mudah



**Product.java**

import lombok.\*;

import org.hibernate.annotations.CreationTimestamp;

import org.hibernate.annotations.UpdateTimestamp;

import javax.persistence.\*;

import java.io.Serializable;

import java.util.Date;

@Entity

@Data

@Table(name = "products")

@Builder

@NoArgsConstructor

@AllArgsConstructor(access = AccessLevel.*PRIVATE*)

public class Product implements Serializable {

@Id

@GeneratedValue(strategy = GenerationType.*IDENTITY*)

private Long id;

@Column(name = "name", unique = true, nullable = false, length = 255)

private String name;

@Column(name = "description")

private String description;

@Column(name = "price")

private Double price;

@Column(name = "created\_at")

@CreationTimestamp

private Date createdAt;

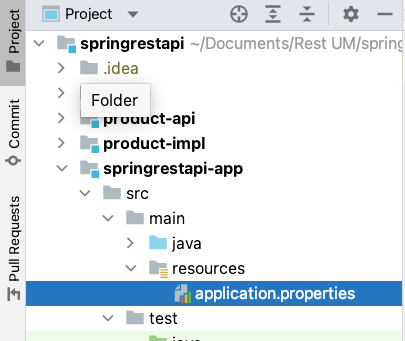
@Column(name = "updated\_at")

@UpdateTimestamp

private Date updatedAt;

}

1. Tambahkan setting pada properties di module level app dengan MySQL Database. Adapun tambahan setting yang diperlukan ada di file application.properties di dalam folder main pada module springrestapi-app



**application.properties**

server.port = 9000

*## MySQL - rest\_api database*

spring.datasource.url=jdbc:mysql://localhost:3306/rest\_api

spring.datasource.username=root

spring.datasource.password=123ABC4d.

*# drop n create table, good for testing, comment this in production*

spring.jpa.hibernate.ddl-auto=update

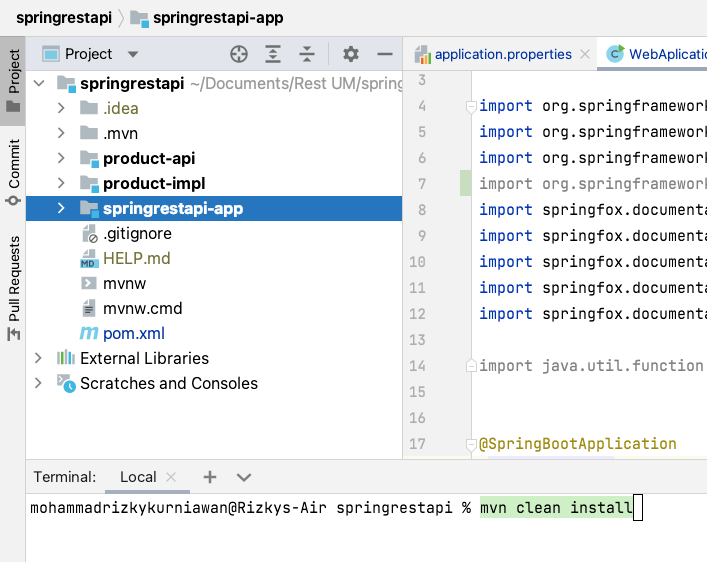
spring.jpa.show-sql=true

Keterangan :



* **spring.datasource.url** diisi alamat database yang tadi telah dibuat, yaitu spring\_book
* **spring.datasource.username** dan **spring.datasource.password** diisi username dan password pada database tersebut
* **spring.jpa.hibernate.ddl-auto** = update digunakan agar JPA secara otomatis memetakan setiap perubahan model. Sebagai contoh, table akan secara otomatis dibuat apabila belum ada dan akan mengupdate setiap perubahan nama.

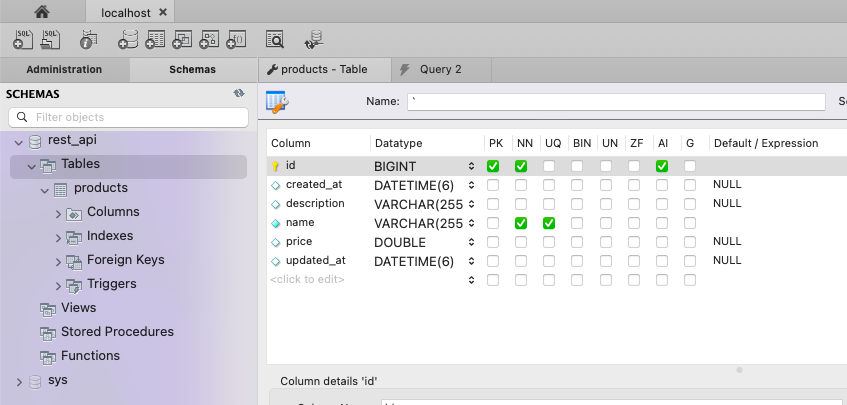
1. Lakukan perintah pada terminal mvn clean install pada folder root project



1. Running aplikasi pada class WebAplication didalam modul springrestapi-app



1. Setelah berhasil running maka otomatis di dalam database rest-api akan dibuatkan sebuah tabel baru dengan nama product dan atribut sesuai dengan mapping di dalam class Product sebelumnya. Hal ini dapat di lihat di MySQL Workbench



# 

## **Rangkuman Modul IV**

1. **Database** merupakan kumpulan data yang terorganisir, yang umumnya disimpan dan diakses secara elektronik dari suatu sistem komputer.
2. **MySQL** adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License).
3. Salah satu tools untuk database client adalah **MySQL Workbench**
4. **ORM (Object Relational Mapping)** adalah mekanisme interaksi dengan database dengan menggunakan sebuah Oriented Object Class seperti membuat sebuah tabel dengan sebuah model class pojo.



## **Tugas Praktikum (Ketrampilan) Modul IV**

1. Install MySQL Server dan Database Client MySQL Workbench
2. Buat database baru ke dalam MySQL Server dengan MySQL Workbench dengan **rest\_api**
3. Lakukan integration project Spring Boot Rest API pada modul III dengan database MySQL Server sesuai dengan langkah-langkah pada Modul IV

# 

# 

## **Tugas Pilihan Ganda (Pengetahuan) Modul IV**

1. Buka Link Berikut (Token : 444)

[**https://forms.gle/MSLgcoWpbSDzaEA98**](https://forms.gle/MSLgcoWpbSDzaEA98)

1. Isikan identitas dan Jawab Soal Pilihan Ganda dalam form tersebut dengan waktu 10 menit
2. Jumlah Soal terdapat 10 Soal