**MEMBUAT PROJECT SPRING BOOT**

**By : Novelina Erlin Saputri**

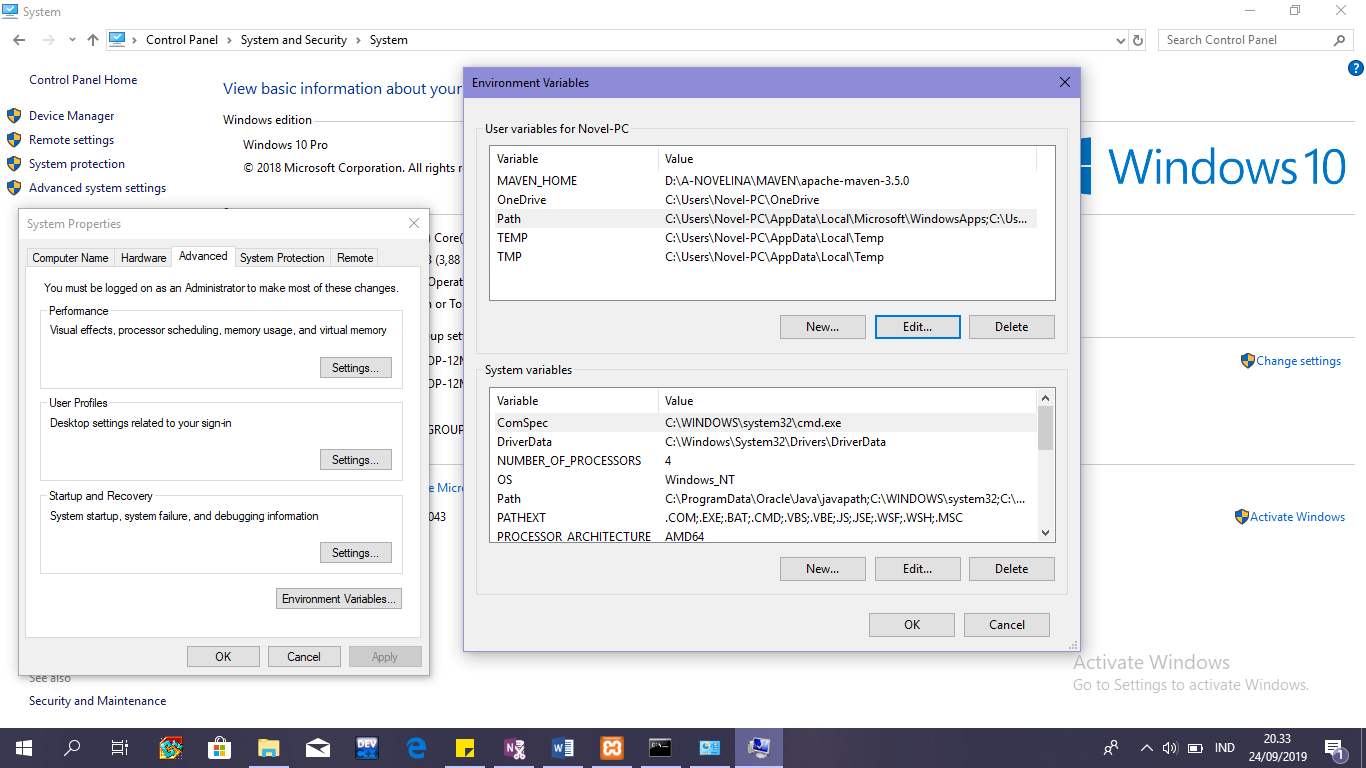
* Aplikasi yang dibutuhkan



1. Eclipse : digunakan untuk pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web.
2. XAMPP : adalah sebagai server yang berdiri sendiri (localhost)
3. SQLyog : yang memudahkan pengguna melakukan administrasi maupun melakukan pengolahan data
4. Postman : fungsinya adalah sebagai REST Client atau istilahnya adalah aplikasi yang digunakan untuk melakukan uji coba REST API yang telah kita buat.

Catatan : Lakukan Start pada XAMPP sebelum memulai projek (MySql)

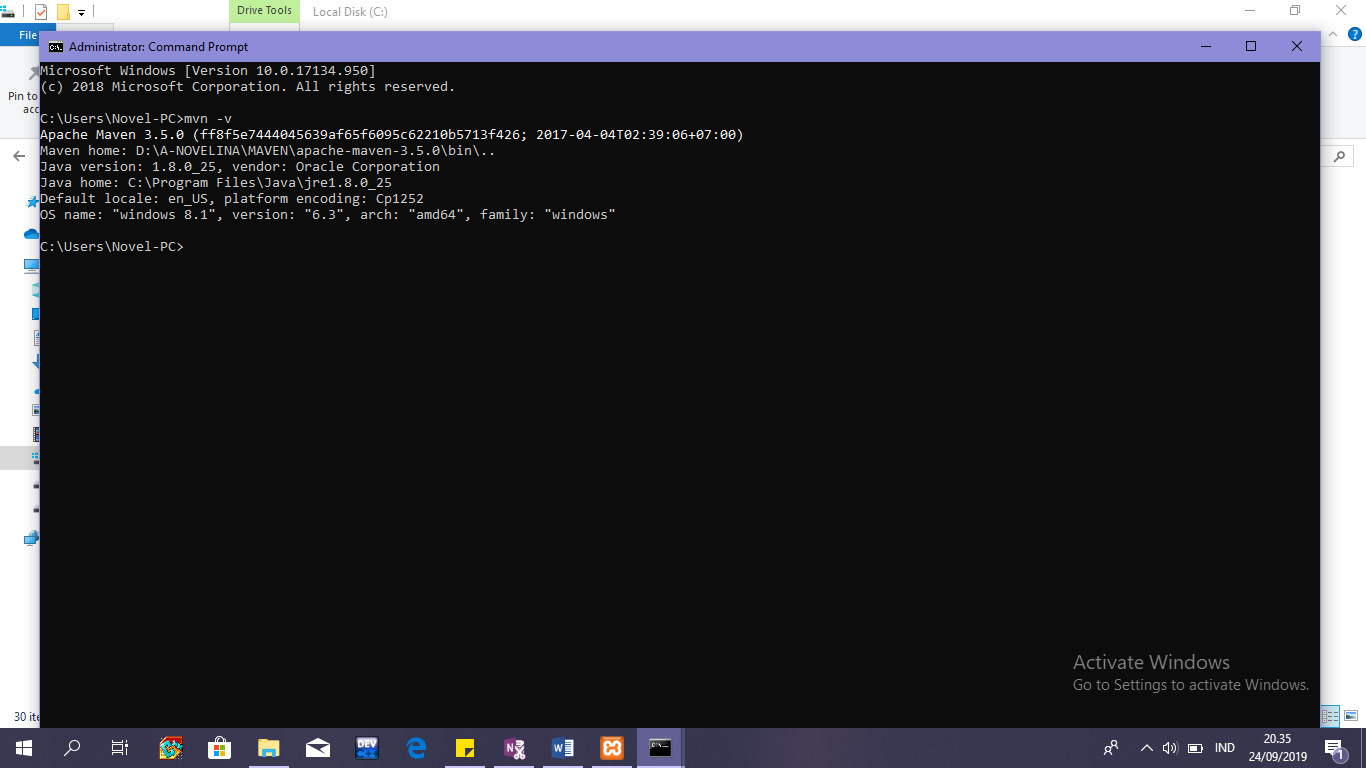
* Install Maven



1. Buka Control Panel
2. Pilik System & Security
3. Pilih System
4. Pilih Advanced System Settings
5. Pilih Environment Variables (maka akan tampil seperti gambar disamping)
6. Kemudian buatlah variabel baru MAVEN\_HOME dengan value adalah lokasi penyimpanan file maven
7. Kemudian edit pada variabel path dengan menambahkan /bin pada akhir value
8. Klik OK

Kemudian untuk mengecek apakah maven sudah terinstall, buka cmd dan ketik **mvn –v**

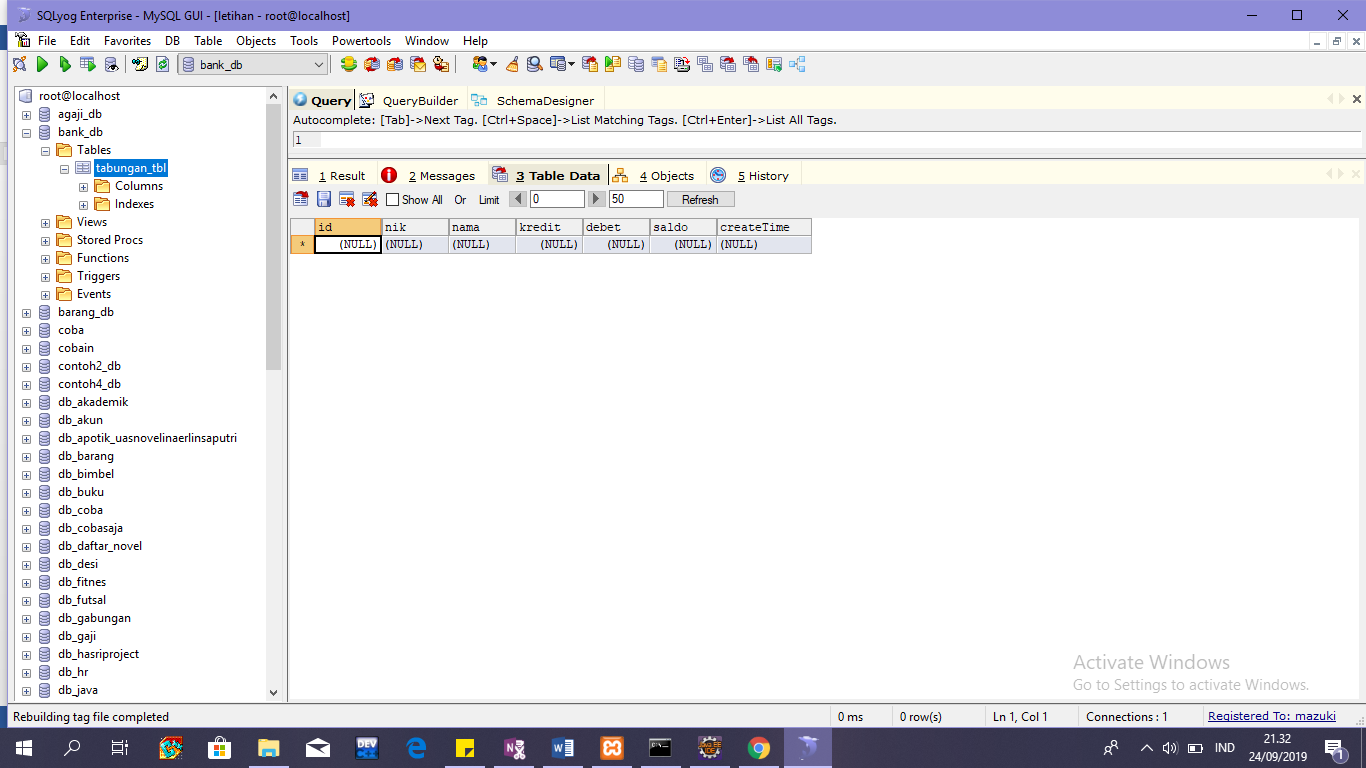
Seperti gambar di bawah ini :



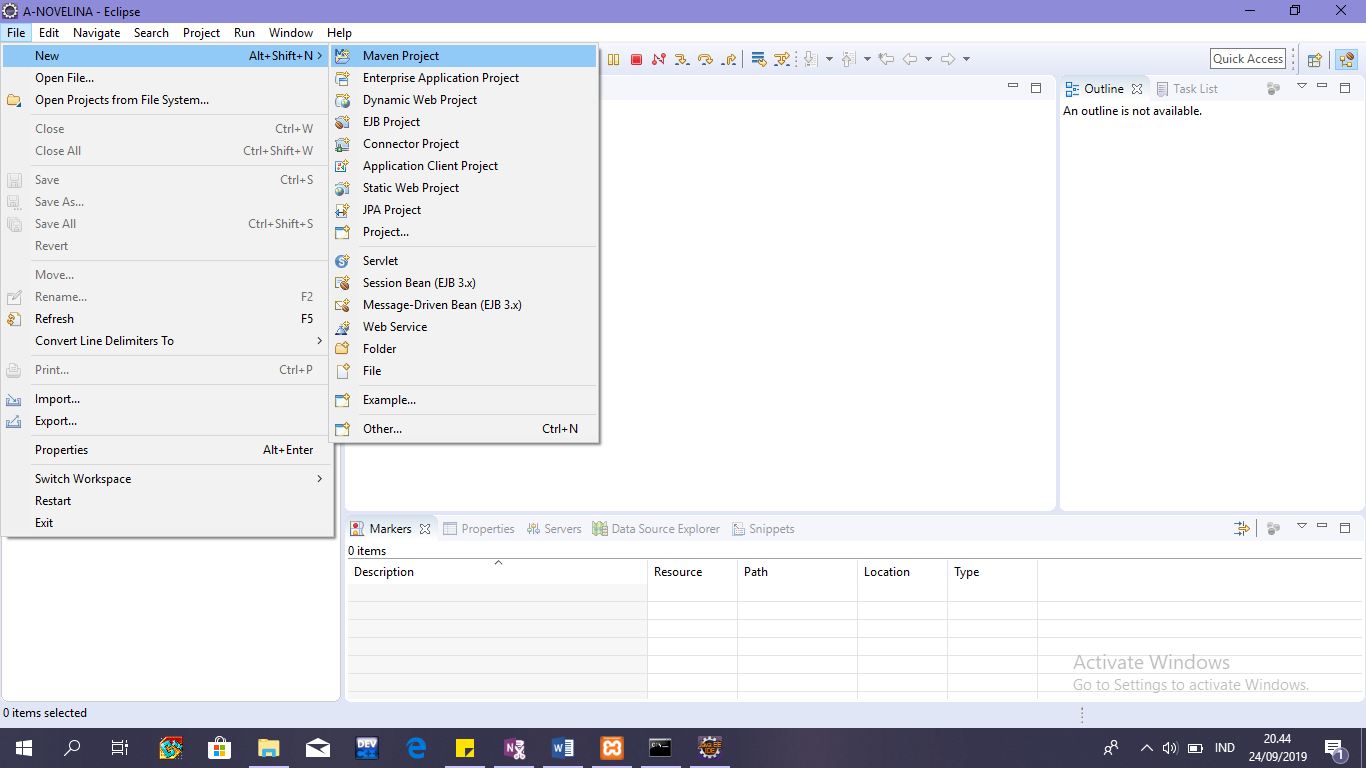
Apabila sudah menampilkan hasil seperti gambar di samping, maka artinya maven sudah ter install

* Create Projek Maven

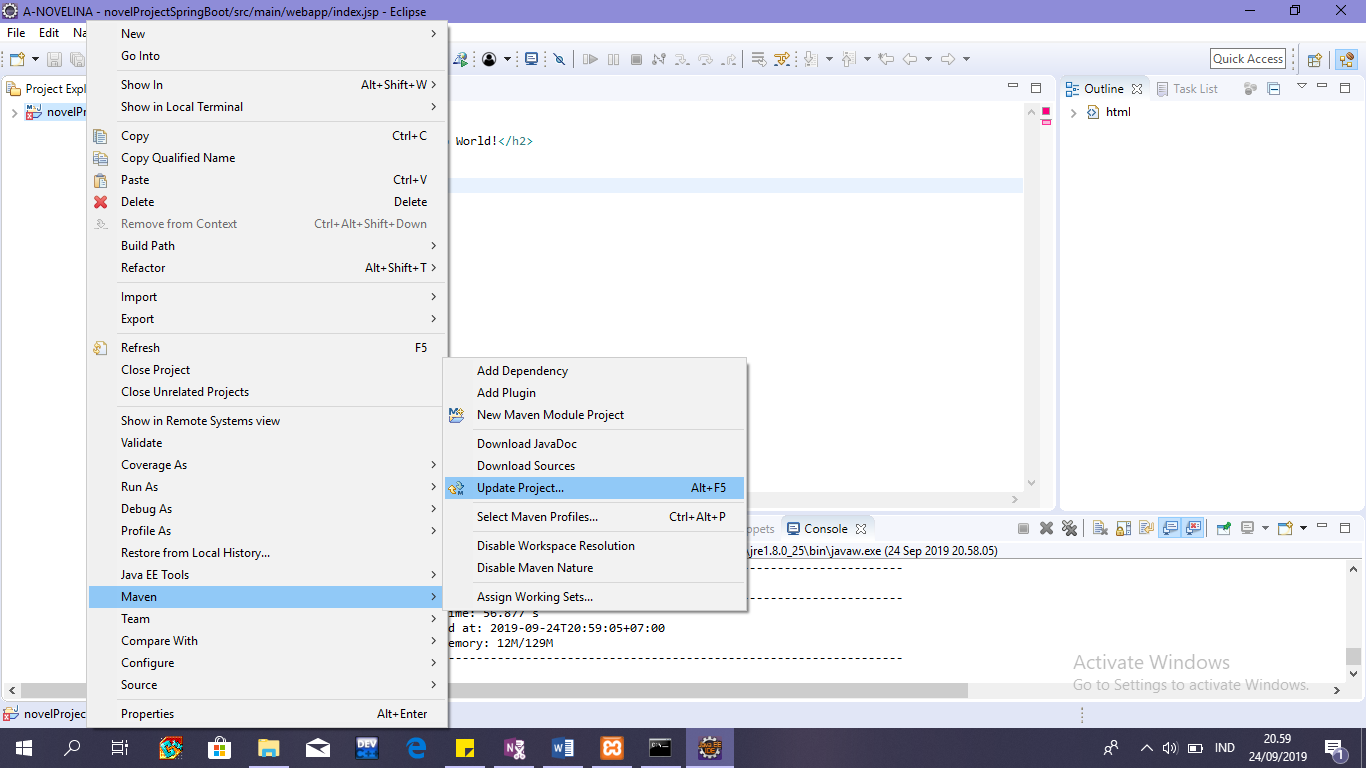
1. Sebelumnya buatlah database seperti di bawah ini pada MySql



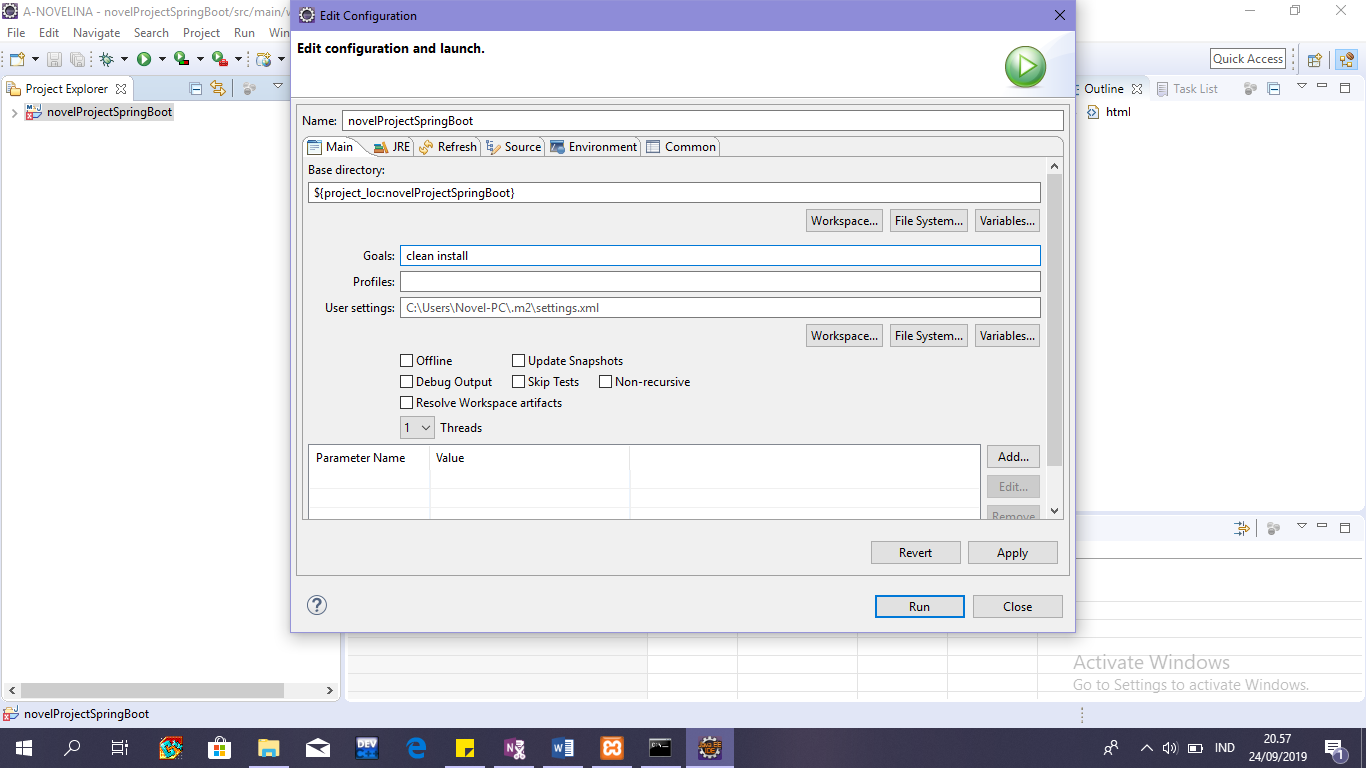
1. Untuk membuat projek maven pada eclipse **File -> New –> Project Maven –> Plih Lokasi Penyimpanan –> Kemudian pada select an Archetype pilih webapp -> Beri nama projek pada Group Id & nama pada Artifact Id -> Finish**



1. Setelah create maka muncul tanda error pada projek, untuk menghilangkannya **Klik kanan projek -> maven -> update project**



1. Run projek dengan cara **Klik kanan projek -> pilih Run As -> Maven Build -> isi Goals : clean install** seperti gambar di bawah ini :



1. Buka pom.xml, kemudian isi dengan ssource seperti di bawah ini :

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>novelProjectSpringBoot</groupId>

<artifactId>novelProjectSpringBoot</artifactId>

<packaging>war</packaging>

<version>0.0.1-SNAPSHOT</version>

<name>novelProjectSpringBoot Maven Webapp</name>

<url>http://maven.apache.org</url>

<parent>

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

<artifactId>spring-boot-starter-parent</artifactId>

<version>1.5.4.RELEASE</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

<dependency>

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

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

</dependency>

<dependency>

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

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

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

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>mysql</groupId>

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

<scope>runtime</scope>

</dependency>

<dependency>

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

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<finalName>novelProjectSpringBoot</finalName>

</build>

</project>

1. Untuk konfigurasi dengan MySql maka **klik kanan pada src/main resource -> new -> file -> application.properties**

spring.datasource.url = jdbc:mysql://localhost:3306/bank\_db?useSSL=false

spring.datasource.username = root

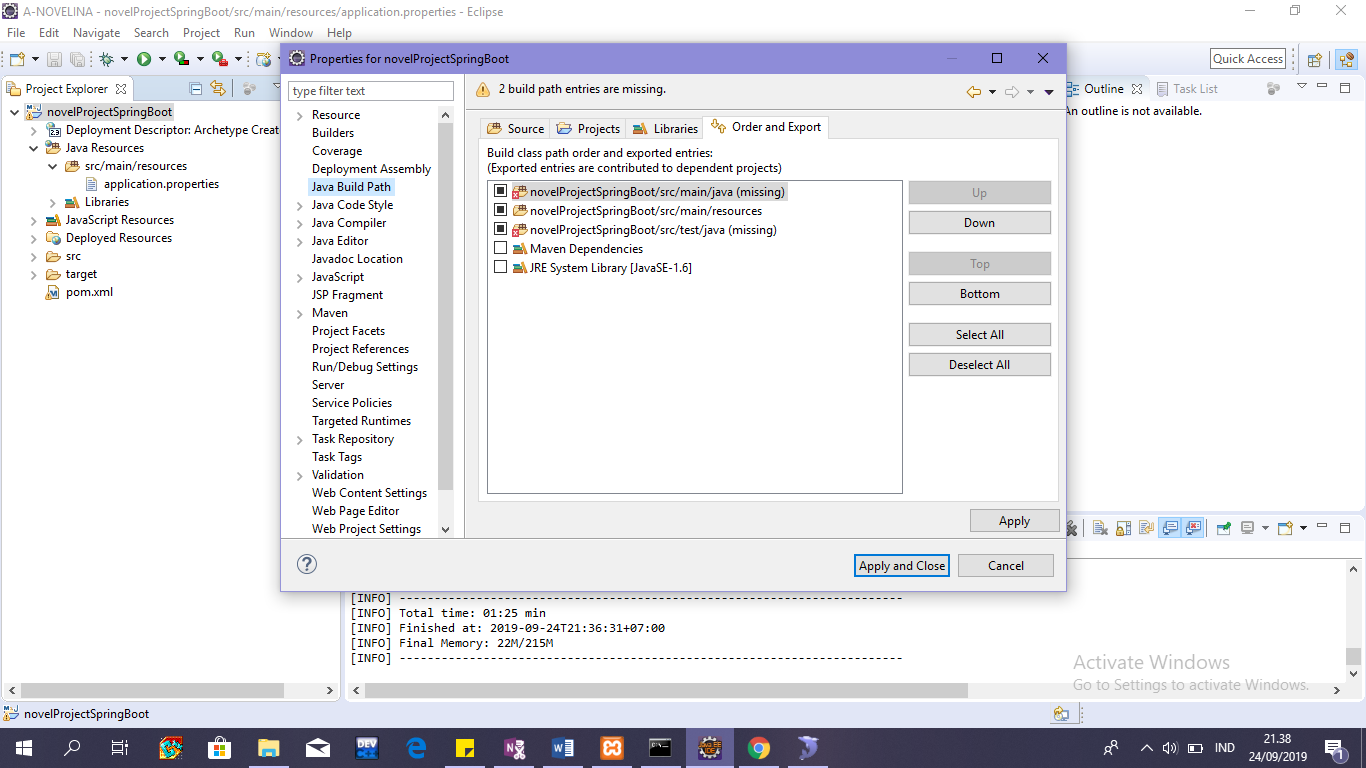
spring.datasource.password =

spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL5Dialect

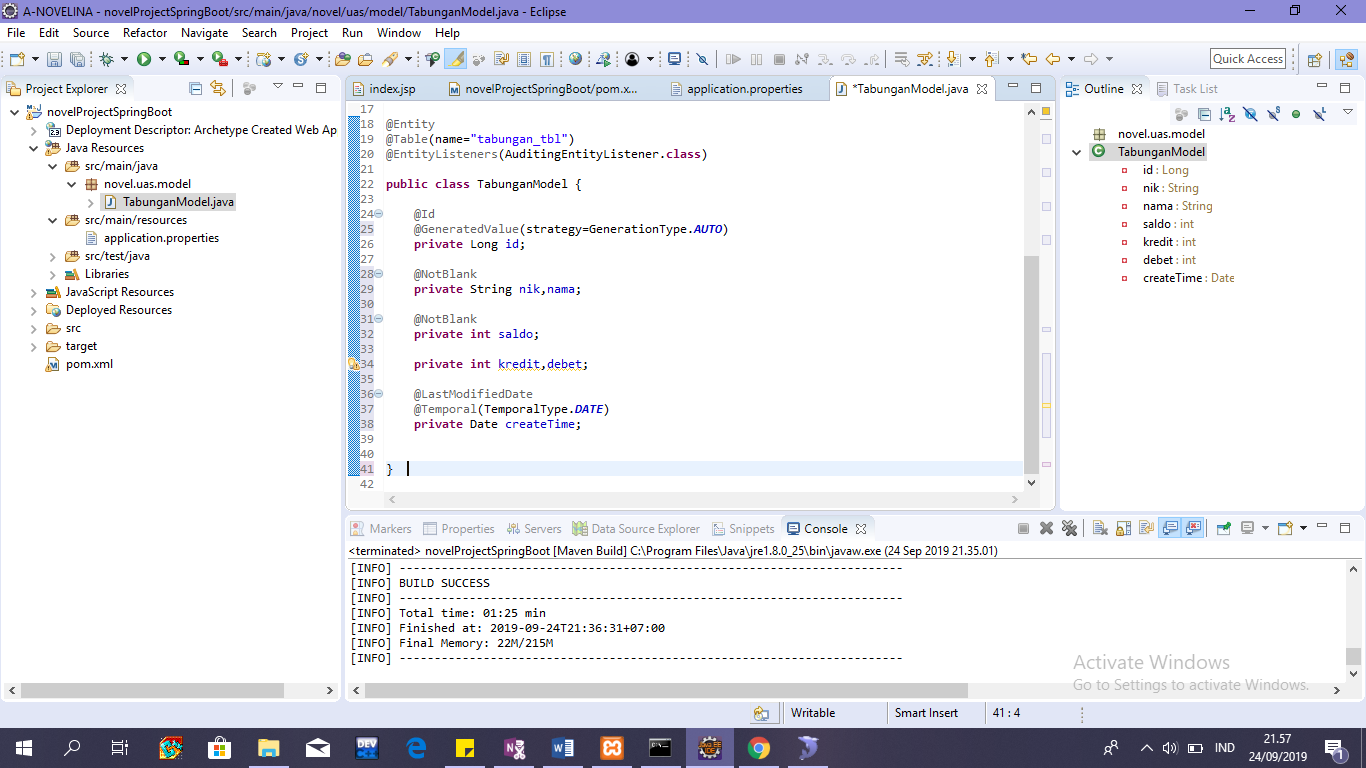
spring.jpa.hibernate.ddl-auto = update

server.port=8085

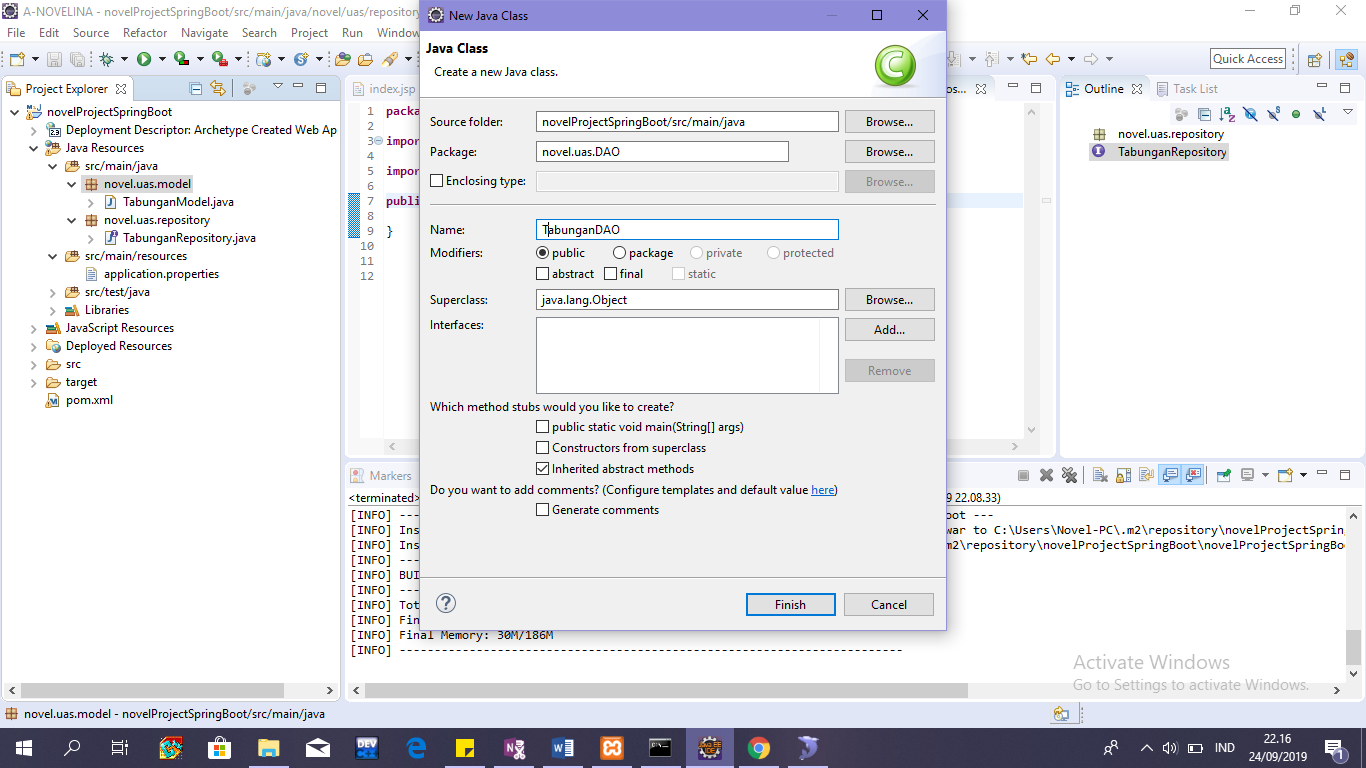
1. Untuk membuat pemodelan dan DAO **Klik kanan pada projek -> properties -> Java Build Path -> ceklist pada (Maven Dependencies & JRE) -> Apply -> Apply and close**



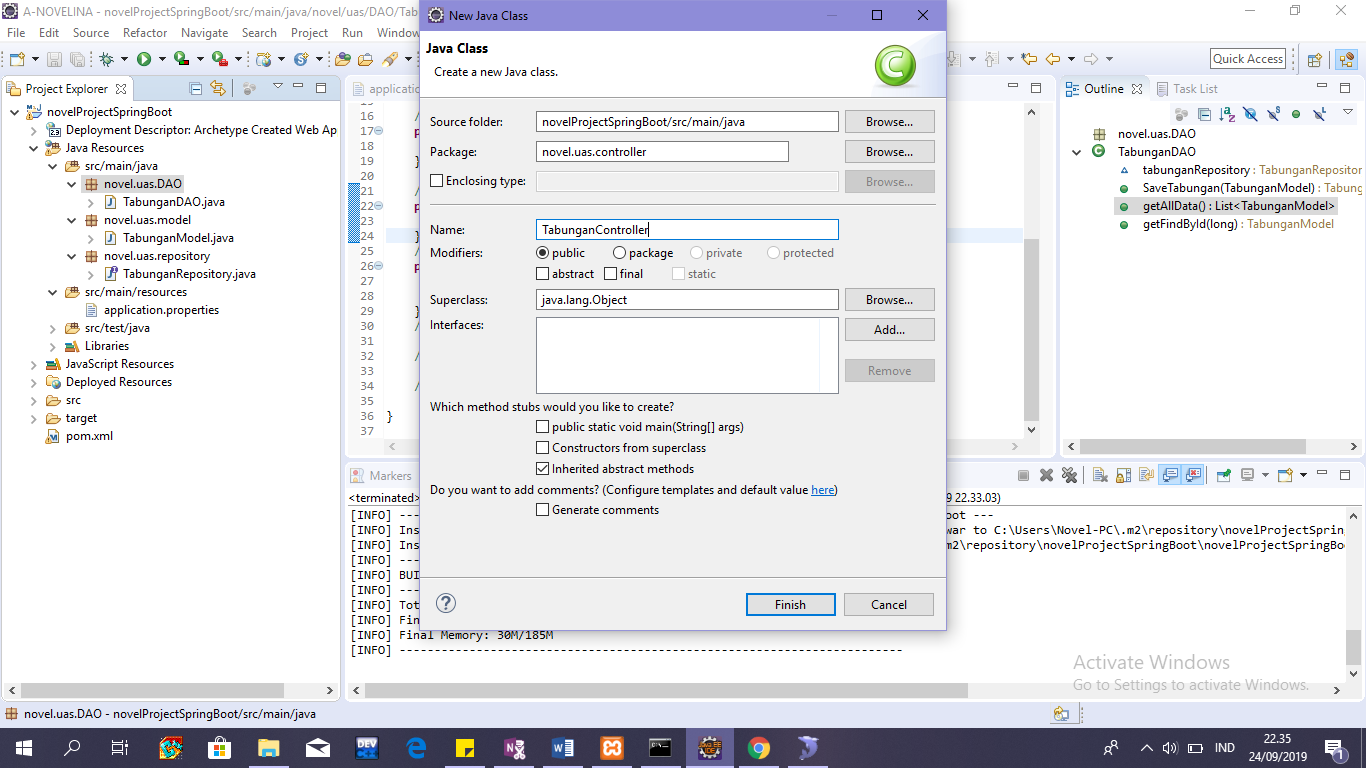
1. Membuat model **klik kanan pada src/main -> new -> class -> beri nama class “TabunganModel” dan package “novel.uas.model” -> Finish**



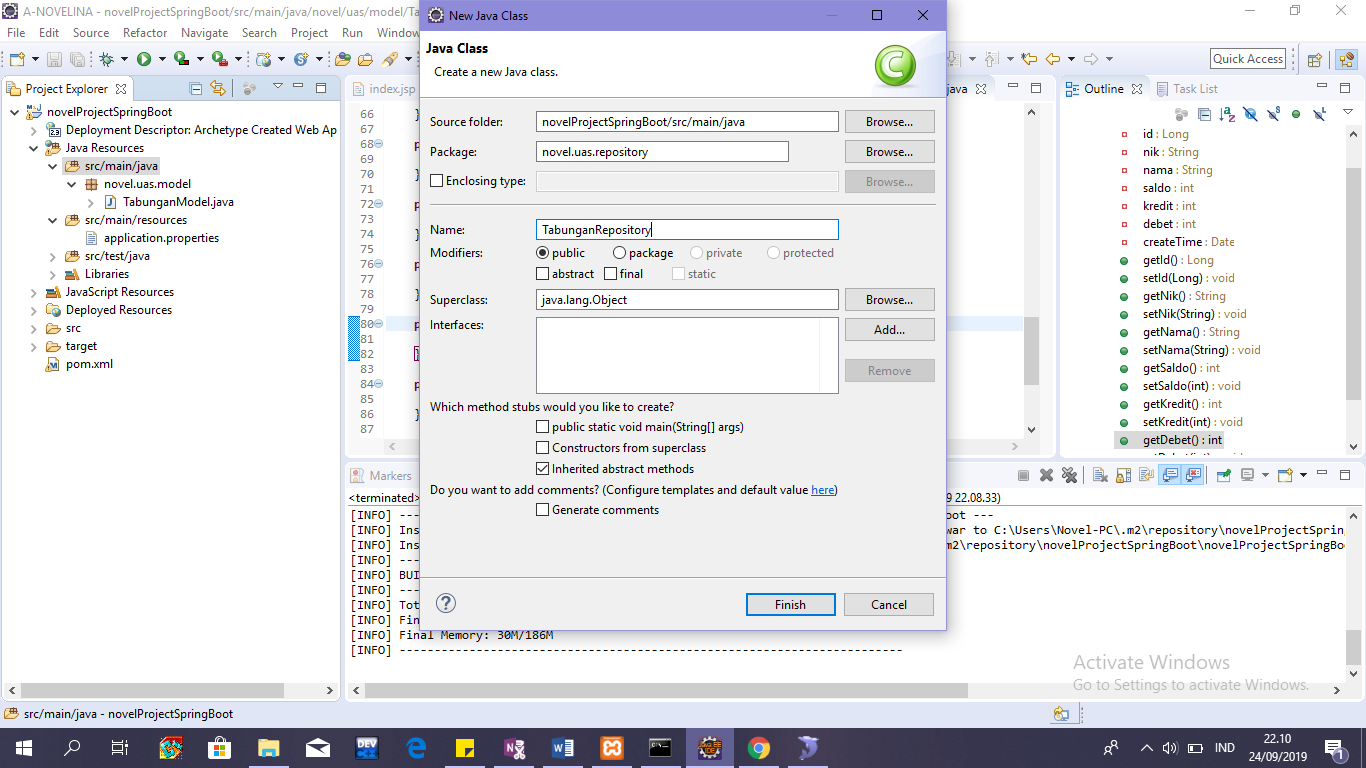
1. Membuat DAO **klik kanan pada src/main -> new -> class -> beri nama class “TabunganDAO” dan package “novel.uas.DAO” -> Finish**



1. Membuat Controller **klik kanan pada src/main -> new -> class -> beri nama class “TabunganController” dan package “novel.uas.controller” -> Finish**



1. Membuat Repository **klik kanan pada src/main -> new -> class -> beri nama class “TabunganRepository” dan package “novel.uas.repository” -> Finish**



**package** novel.uas.repository;

**import** org.springframework.data.jpa.repository.JpaRepository;

**import** novel.uas.model.TabunganModel;

**public** **interface** TabunganRepository **extends** JpaRepository<TabunganModel, Long>{

}

1. Membuat class main **klik kanan pada src/main -> new -> class -> beri nama class “TabunganApplication” dan package “novel.uas” -> Finish**

**package** novel.uas;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

**import** org.springframework.data.jpa.repository.config.EnableJpaAuditing;

@SpringBootApplication

@EnableJpaAuditing

**public** **class** TabunganApplication {

**public** **static** **void** main(String[] arg) {

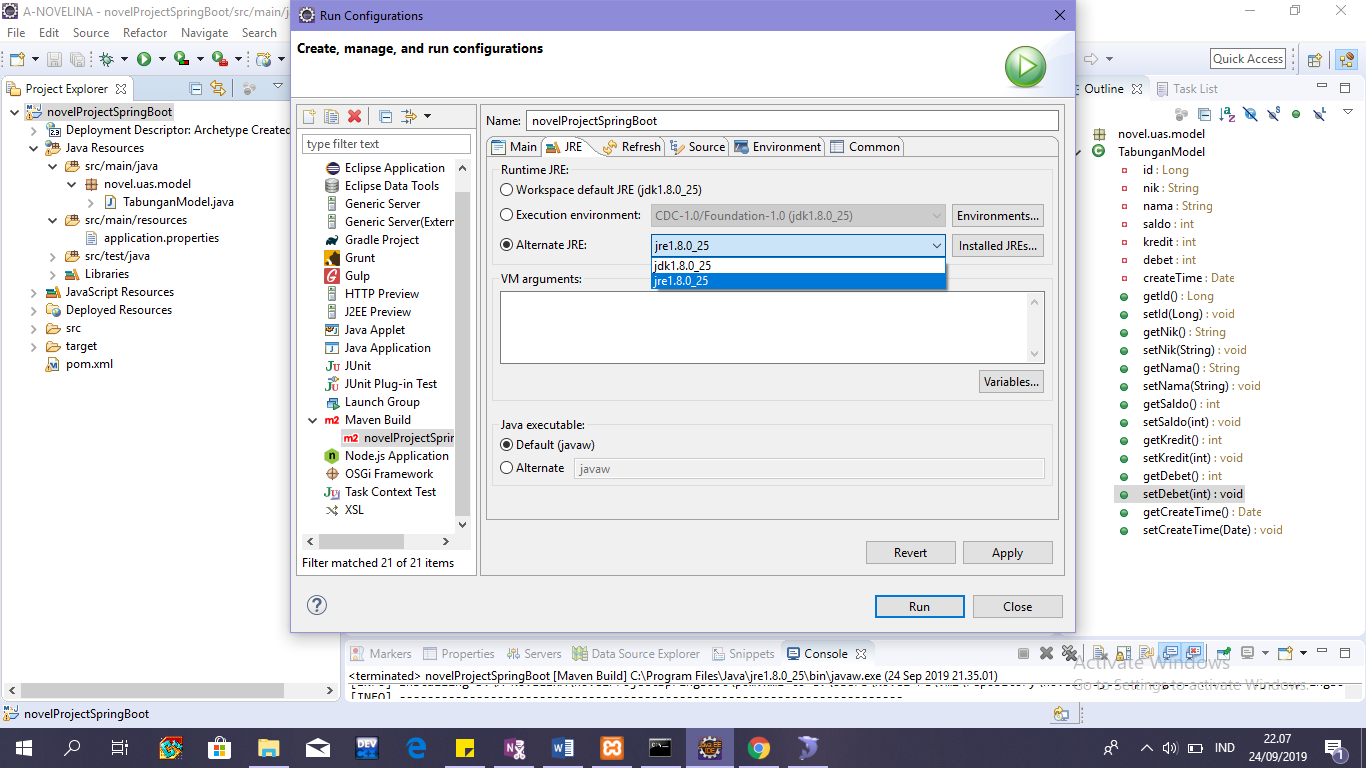
SpringApplication.*run*(TabunganApplication.**class**, arg);

}

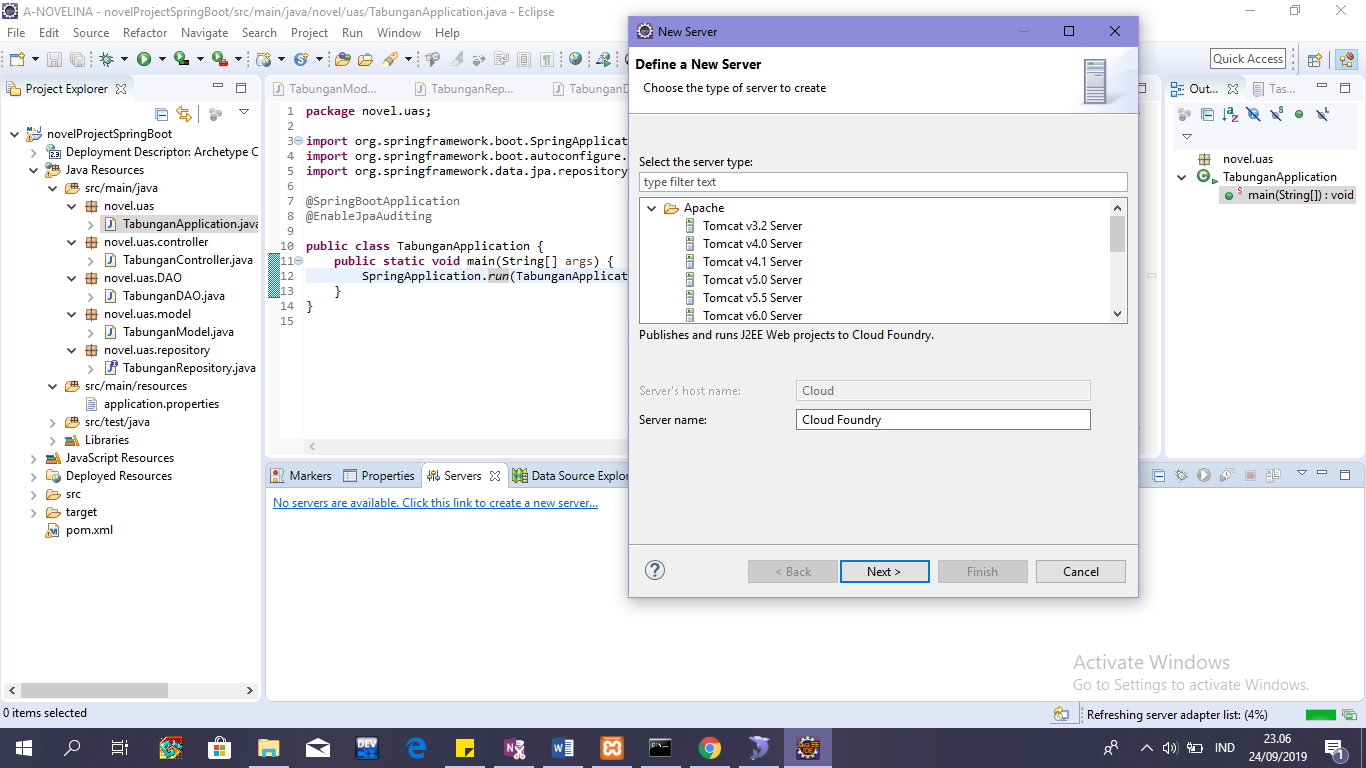
}

1. Mengganti JRE ke JDK

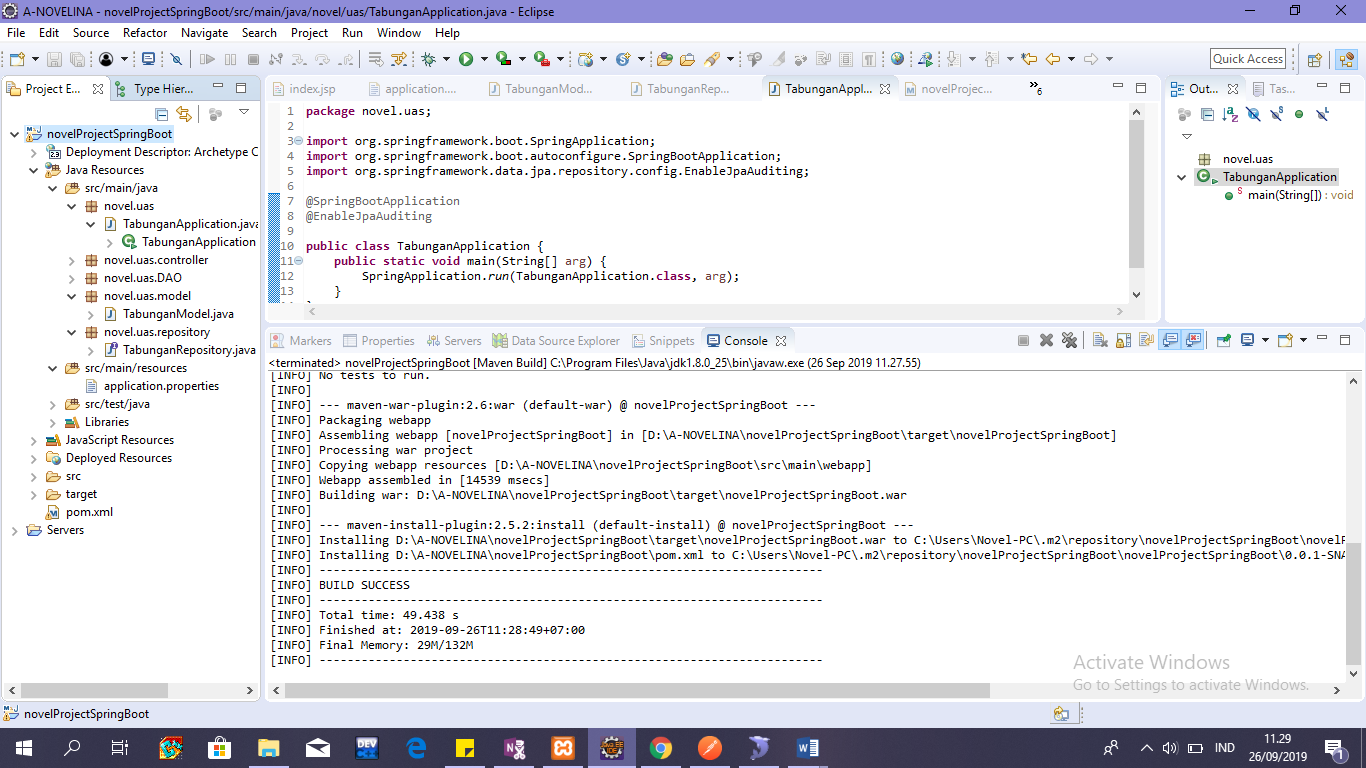
**Klik kanan -> Run As -> Run Configuration -> JRE -> Installed JREs.. -> add -> cari tempat penyimpanan file jdk -> finish**



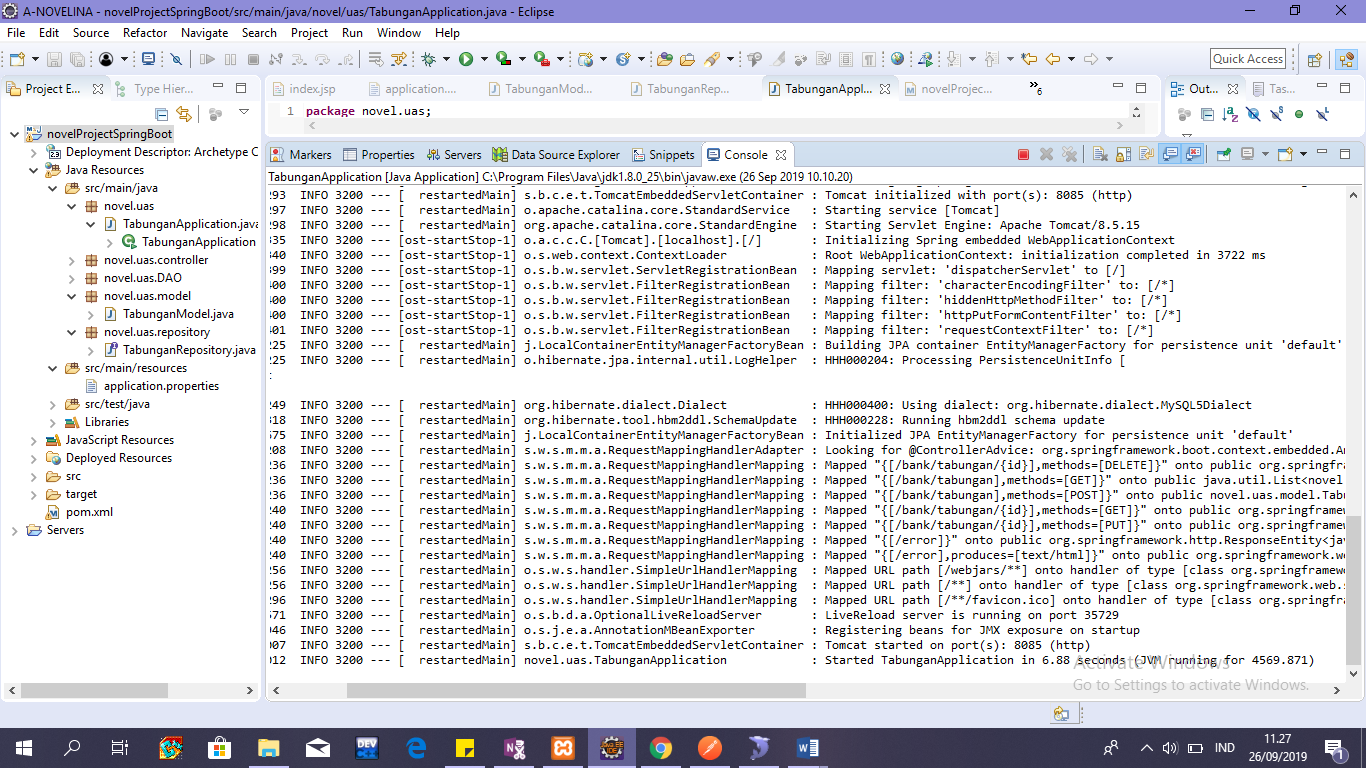
1. Membuat Server Tomcat



1. Untuk mengecek projek kita apakah sukses atau tidak maka **klik kanan pada projek -> Run As -> Maven Build**

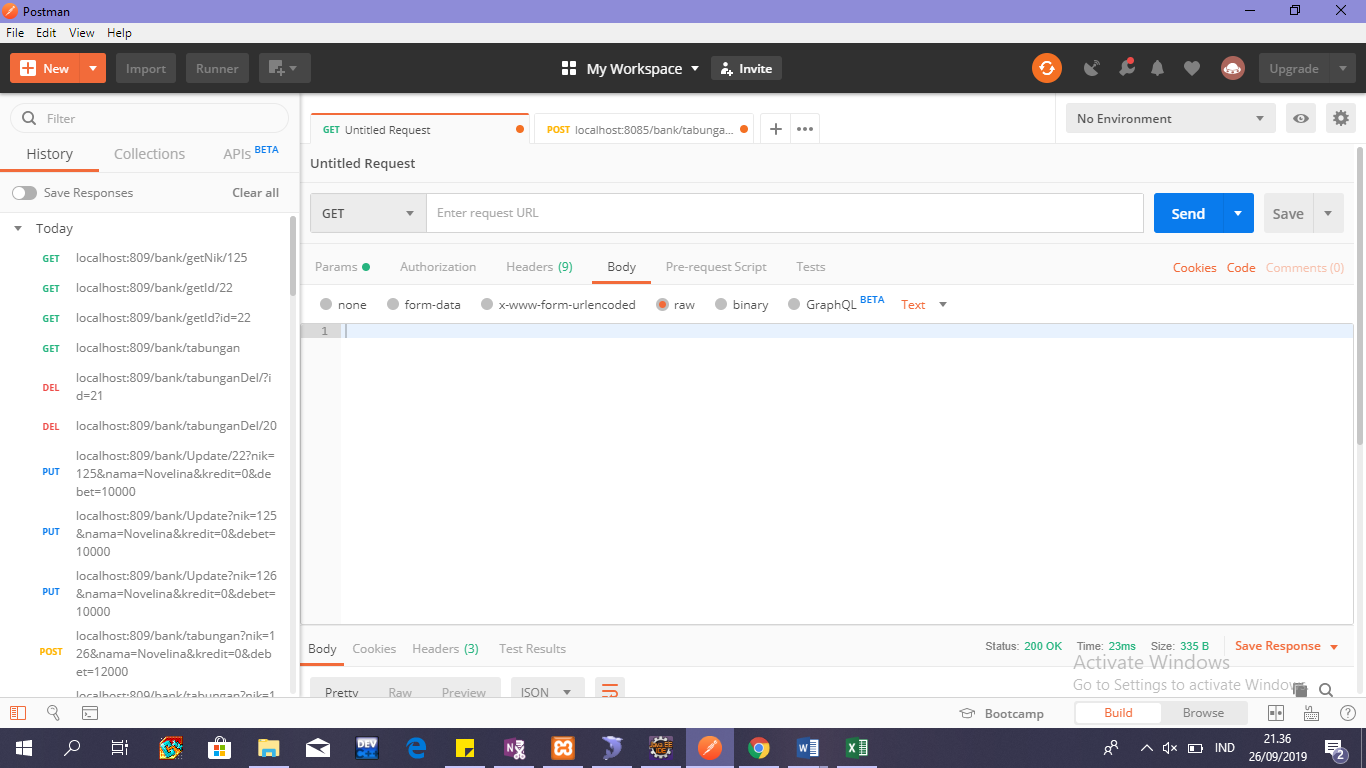


1. Kemudian **Klik kanan pada class TabunganApplication -> Run -> Java Application**, apabila berhasil maka akan menampilkan :



* Menjalankan Project dengan Postman

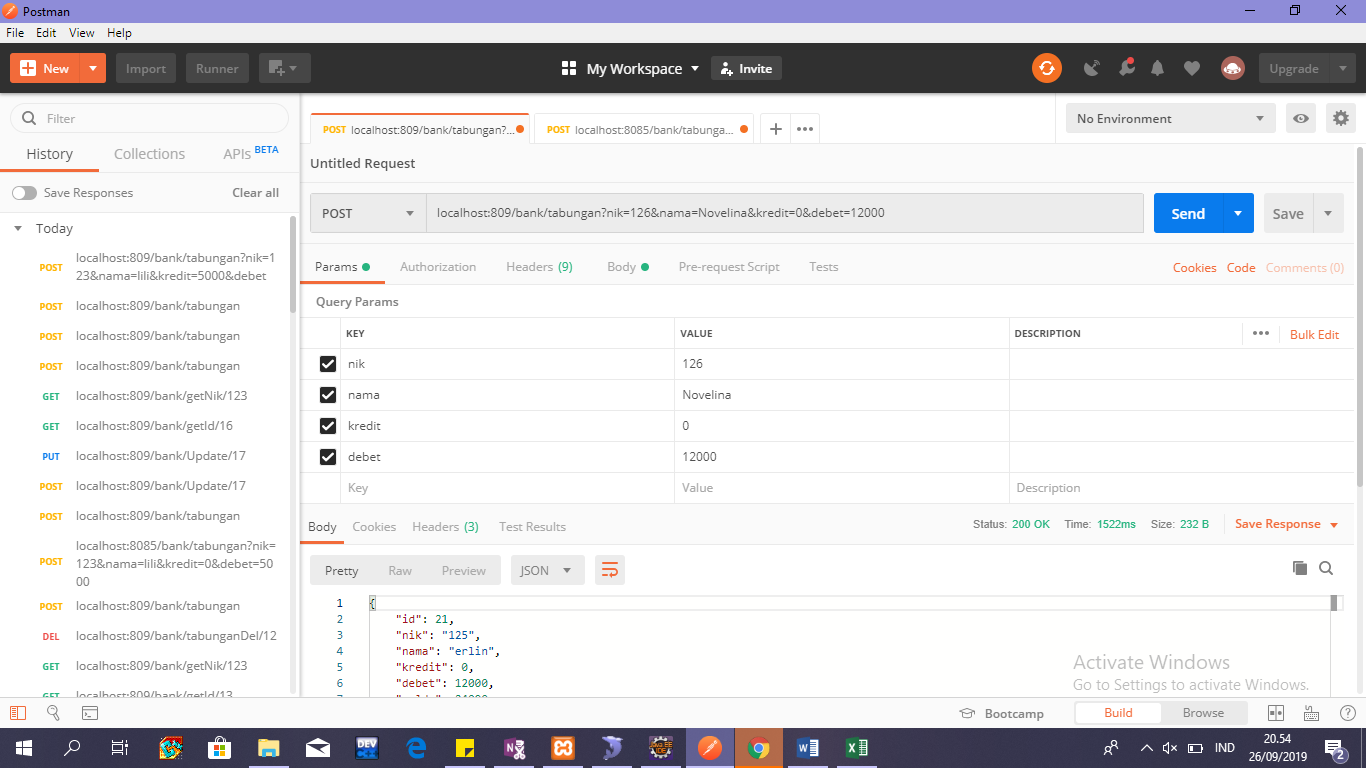
1. Buka Aplikasi Postman
2. Sign in dengan menggunakan akun email
3. Maka akan menampilkan tampilan postman seperti dibawah ini :



1. Untuk mencoba Rest API yang telah dibuat maka tuliskan alamat URL nya kemudian send maka akan tampil hasilnya.
2. Adapun method yang terdapat dalam postman (yang sering digunakan) antara lain :
   * GET digunakan untuk memanggil data yang telah ada
   * POST digunakan untuk menambahkan data ke dalam server
   * PUT digunakan untuk mengupdate data yang telah ada
   * DELETE digunakan untuk menghapus data yang telah ada
3. Contoh cek Rest API POST (untuk menambah data/insert data)

* Masukkan URL (sesuaikan port, kata kunci dan parameter)

localhost:809/bank/tabungan?nik=126&nama=Novelina&kredit=0&debet=12000



* Maka akan menampilkan hasil dari proses POST

