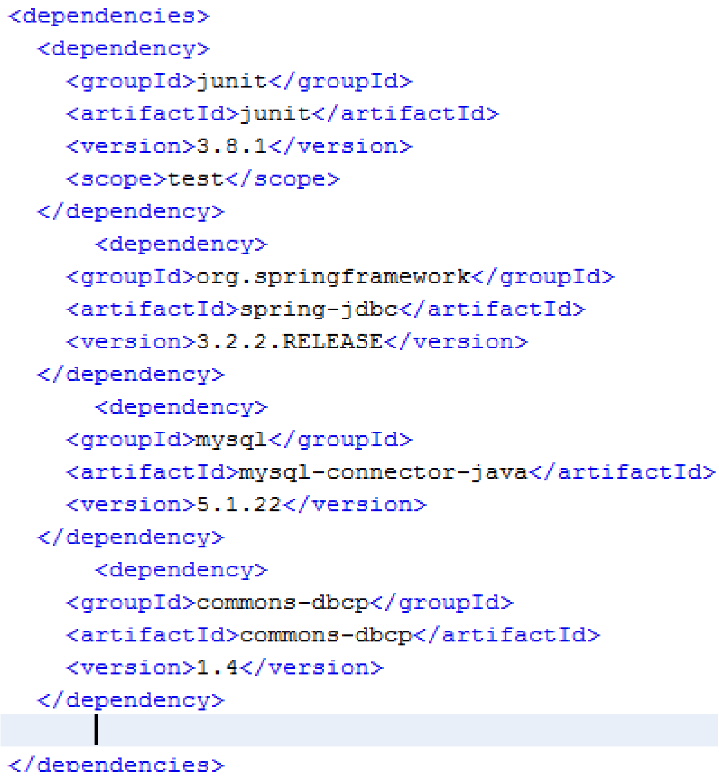
Spring Framework

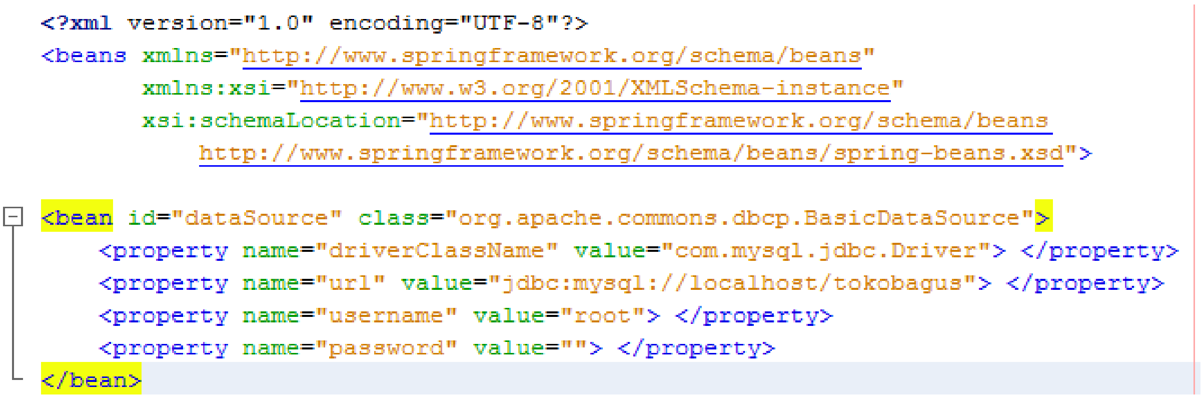
1. Menambahkan dependensi :
   - Dependensi Spring Framework
   - Dependensi MySQL
   - Dependensi commons-dbcp



```
<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>3.8.1</version>
    <scope>test</scope>
  </dependency>
      <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-jdbc</artifactId>
    <version>3.2.2.RELEASE</version>
  </dependency>
      <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>5.1.22</version>
  </dependency>
      <dependency>
    <groupId>commons-dbcp</groupId>
    <artifactId>commons-dbcp</artifactId>
    <version>1.4</version>
  </dependency>
      
</dependencies>
```

Gambar 1. POM.xml

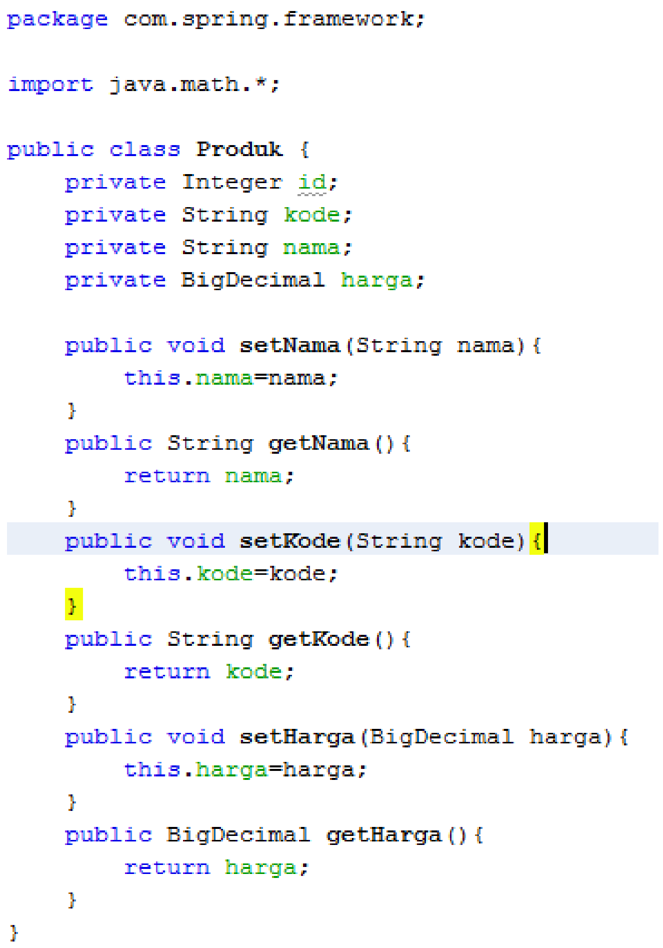
2. Mendeklarasikan object spring framework pada



```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
           http://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource">
    <property name="driverClassName" value="com.mysql.jdbc.Driver"> </property>
    <property name="url" value="jdbc:mysql://localhost/tokobagus"> </property>
    <property name="username" value="root"> </property>
    <property name="password" value=""> </property>
</bean>
```

Gambar 2. config-spring.xml

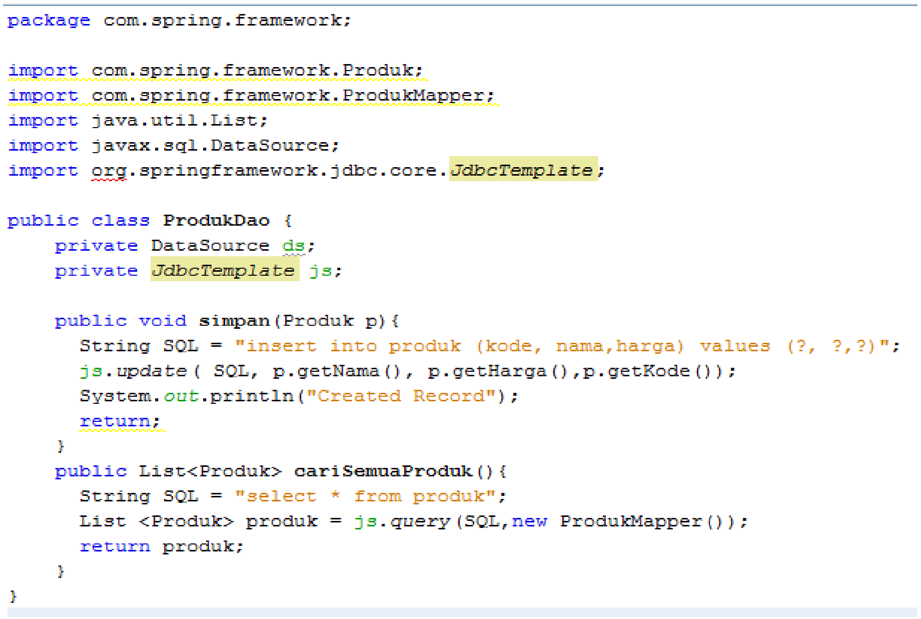
3. Membuat class Produk



```
package com.spring.framework;

import java.math.*;

public class Produk {
    private Integer id;
    private String kode;
    private String nama;
    private BigDecimal harga;

    public void setNama(String nama){
        this.nama=nama;
    }
    public String getNama(){
        return nama;
    }
    public void setKode(String kode){
        this.kode=kode;
    }
    public String getKode(){
        return kode;
    }
    public void setHarga(BigDecimal harga){
        this.harga=harga;
    }
    public BigDecimal getHarga(){
        return harga;
    }
}
```

Gambar 3. Produk.java

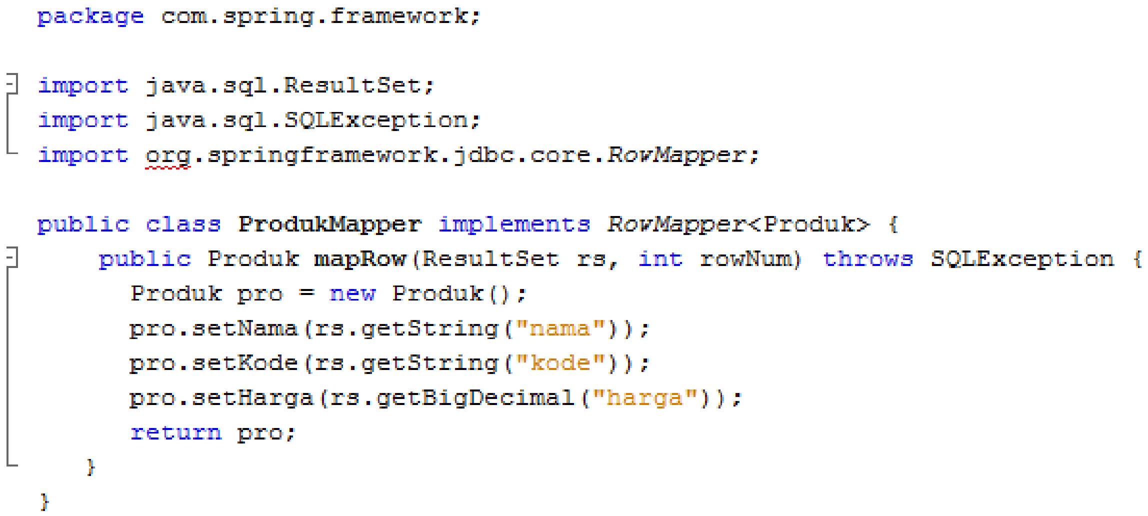
4. Membuat class ProdukDao



```
package com.spring.framework;

import com.spring.framework.Produk;
import com.spring.framework.ProdukMapper;
import java.util.List;
import javax.sql.DataSource;
import org.springframework.jdbc.core.JdbcTemplate;

public class ProdukDao {
    private DataSource ds;
    private JdbcTemplate js;

    public void simpan(Produk p){
      String SQL = "insert into produk (kode, nama,harga) values (?, ?,?)";
      js.update( SQL, p.getNama(), p.getHarga(),p.getKode());
      System.out.println("Created Record");
      return;
    }
    public List<Produk> cariSemuaProduk(){
      String SQL = "select * from produk";
      List <Produk> produk = js.query(SQL,new ProdukMapper());
      return produk;
    }
}
```

Gambar 4. ProdukDao.java

5. Membuat ProdukMapper



```
package com.spring.framework;

import java.sql.ResultSet;
import java.sql.SQLException;
import org.springframework.jdbc.core.RowMapper;

public class ProdukMapper implements RowMapper<Produk> {
    public Produk mapRow(ResultSet rs, int rowNum) throws SQLException {
      Produk pro = new Produk();
      pro.setNama(rs.getString("nama"));
      pro.setKode(rs.getString("kode"));
      pro.setHarga(rs.getBigDecimal("harga"));
      return pro;
   }
}
```

Gambar 5. ProdukMapper.java

6. Hasil Program.

   Masih belum bisa bekerja dengan baik. a.k.a “ERROR”.