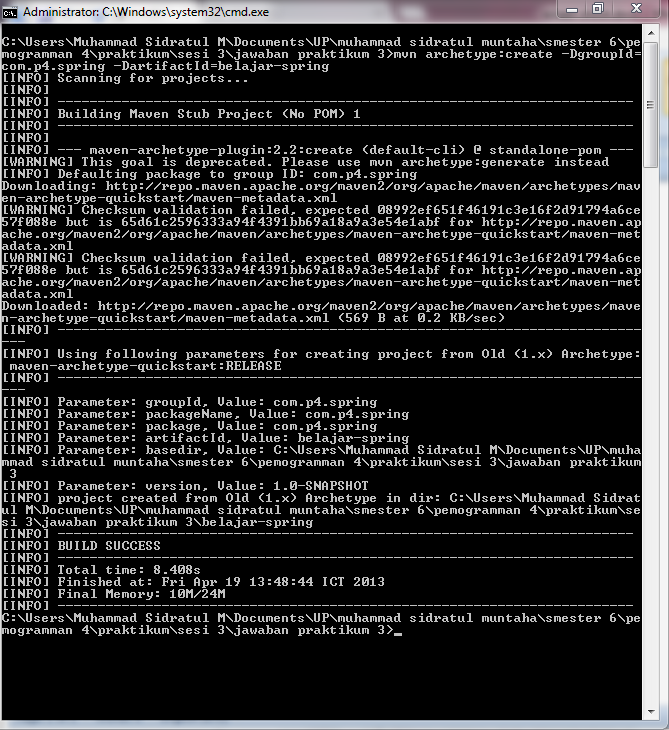
**PEMROGRAMMAN 4**

**PRAKTIKUM 3**

**Nama : Muhammad Sidratul Muntaha**

**Npm : 4510210013**



membuat project maven

**pom.xml**

|  |
| --- |
| <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/xsd/maven-4.0.0.xsd">  <modelVersion>4.0.0</modelVersion>  <groupId>com.p4.spring</groupId>  <artifactId>belajar-spring</artifactId>  <version>1.0-SNAPSHOT</version>  <packaging>jar</packaging>  <name>belajar-spring</name>  <url>http://maven.apache.org</url>  <properties>  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  </properties>  <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>3.2.1.RELEASE</version>  </dependency>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-context</artifactId>  <version>3.2.2.RELEASE</version>  </dependency>  <dependency>  <groupId>mysql</groupId>  <artifactId>mysql-connector-java</artifactId>  <version>5.1.23</version>  </dependency>  <dependency>  <groupId>commons-dbcp</groupId>  <artifactId>commons-dbcp</artifactId>  <version>1.4</version>  </dependency>  </dependencies>  </project> |

config-spring.xml

|  |
| --- |
| <?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="driveClassName" value="com.mysql.jdbc.Driver"> </property>  <property name="url" value="jdbc:mysql://localhost/p4"> </property>  <property name="username" value="root"> </property>  <property name="password" value=""> </property>  </bean>    <bean id="pd" class="com.p4.spring.ProdukDao">  <property name="password" value=""> </property>  </bean>    <bean id="p" class="com.p4.spring.Produk">  <property name="kode" value="ab002"> </property>  <property name="nama" value="teh botol"> </property>  <property name="harga" ref="10000"> </property>  <property name="lastUpdate" value="2013-04-12"> </property>  </bean>  <!-- more bean definitions go here -->  <bean id="dateEditor"  class="org.springframework.beans.propertyeditors.CustomDateEditor">    <constructor-arg>  <bean class="java.text.SimpleDateFormat">  <constructor-arg value="yyyy-MM-dd" />  </bean>  </constructor-arg>  <constructor-arg value="true" />  </bean>    <bean class="org.springframework.beans.factory.config.CustomEditorConfigurer">  <property name="customEditors">  <map>  <entry key="java.util.Date">  <ref local="dateEditor" />  </entry>  </map>  </property>  </bean>  </beans> |

Produk.java

|  |
| --- |
| package com.p4.spring;  import java.math.BigDecimal;  import java.util.Date;  public class Produk {  private Integer id;  private String kode;  private String nama;  private BigDecimal harga;  private Date terakhirUpdate;    public void setId(Integer id){  this.id=id;  }    public Integer getId(){  return id;  }    public void setKode(String kode){  this.id=id;  }    public String getKode(){  return kode;  }    public void setNama(String nama){  this.nama=nama;  }    public String getNama(){  return nama;  }    public void setHarga(BigDecimal harga){  this.harga=harga;  }    public BigDecimal getHarga(){  return harga;  }    public void setLastUpdate(Date terakhirUpdate){  this.terakhirUpdate=terakhirUpdate;  }    public Date getLastUpdate(){  return terakhirUpdate;  }  } |

ProdukDao.java

|  |
| --- |
| package com.p4.spring;  import com.p4.spring.Produk;  import com.p4.spring.ProdukMapper;  import java.util.List;  import javax.sql.DataSource;  import org.springframework.jdbc.core.JdbcTemplate;  public class ProdukDao {  private DataSource ds;  private JdbcTemplate jt;    public void simpan(Produk p){  String SQL = "insert into Student (id,kode,nama,harga,terakhir\_update) values (?,?,?,?)";  jt.update( SQL,p.getKode(),p.getNama(),p.getHarga(),p.getLastUpdate());  System.out.println("Created Record");  }  public List<Produk> cariSemuaProduk(){  String SQL = "select \* from produk";  List <Produk> produk = jt.query(SQL,new ProdukMapper());  return produk;  }  } |

ProdukMapper.java

|  |
| --- |
| /\*  \* To change this template, choose Tools | Templates  \* and open the template in the editor.  \*/  package com.p4.spring;  import java.sql.SQLException;  import org.springframework.jdbc.core.RowMapper;  import java.sql.ResultSet;  import com.p4.spring.Produk;  import java.math.BigDecimal;  public class ProdukMapper implements RowMapper<Produk> {  public Produk mapRow(ResultSet rs,int rowNum) throws SQLException{  Produk p = new Produk();  p.setId(rs.getInt("id"));  p.setKode(rs.getString("kode"));  p.setNama(rs.getString("Nama"));  p.setHarga(rs.getBigDecimal("harga"));  p.setLastUpdate(rs.getDate("terakhir\_update"));  return p;    }  } |

Main.java

|  |
| --- |
| package com.p4.spring;  import com.p4.spring.Produk;  import com.p4.spring.ProdukDao;  import java.math.BigDecimal;  import java.util.List;  import org.springframework.context.support.ClassPathXmlApplicationContext;  import org.springframework.context.ApplicationContext;  public class Main {  public static void main(String[] args){  ApplicationContext context = new ClassPathXmlApplicationContext(new String[] {"config-spring.xml"});  Produk p=context.getBean("p",com.p4.spring.Produk.class);  ProdukDao pd=context.getBean("pd",com.p4.spring.ProdukDao.class);    pd.simpan(p);  List<Produk> lp = pd.cariSemuaProduk();    }  } |