

PRAKTIKUM LABORATORIUM TEKNIK INFORMATIKA

ACTIVITY PERTEMUAN KE - 5

Nama : Dimas Katon Rahmat A.
NPM : 50421392
Kelas : 4IA25
Mata Praktikum : Rekayasa Perangkat Lunak 2

LISTING

1. Koneksi Hibernate ke Database

Mengkonfigurasi Hibernate untuk menghubungkan aplikasi dengan database.

```
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
  <session-factory>
    <!-- Database connection settings -->
    <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
    <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/db_toko</property>
    <property name="hibernate.connection.username">root</property>
    <property name="hibernate.connection.password"></property>
    <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>

    <!-- Echo all executed SQL to stdout -->
    <property name="hibernate.show_sql">true</property>

    <!-- Drop and re-create the database schema on startup -->
    <property name="hibernate.hbm2ddl.auto">update</property>
  </session-factory>
</hibernate-configuration>
```

2. Model Produk

```

import javax.persistence.*;

@Entity
@Table(name = "produk")
public class ProdukAksesoris {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;

    @Column(name = "kode_produk")
    private String kodeProduk;

    @Column(name = "nama_produk")
    private String namaProduk;

    @Column(name = "harga")
    private int harga;

    @Column(name = "kategori")
    private String kategori;

    // Getter and Setter methods
}

```

3. Class HibernateUtil

Class ini akan menyediakan metode untuk mendapatkan session Hibernate. Class ini penting untuk mengelola koneksi database melalui Hibernate.

```

import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

public class HibernateUtil {
    private static final SessionFactory sessionFactory = buildSessionFactory();

    private static SessionFactory buildSessionFactory() {
        try {
            return new Configuration().configure().buildSessionFactory();
        } catch (Throwable ex) {
            throw new ExceptionInInitializerError(ex);
        }
    }

    public static SessionFactory getSessionFactory() {
        return sessionFactory;
    }
}

```

4. Class ControllerProduk

Class ini berfungsi sebagai controller untuk mengatur interaksi antara pengguna dan database. Method insertProduk digunakan untuk menambahkan produk baru ke database, sementara getProdukList digunakan untuk mengambil data produk dari database.

```

import org.hibernate.Session;
import org.hibernate.Transaction;
import java.util.List;

public class ControllerProduk {

    public void insertProduk(ProdukAksesoris produk) {
        Transaction transaction = null;
        try (Session session = HibernateUtil.getSessionFactory().openSession()) {
            transaction = session.beginTransaction();
            session.save(produk);
            transaction.commit();
        } catch (Exception e) {
            if (transaction != null) {
                transaction.rollback();
            }
            e.printStackTrace();
        }
    }

    public List<ProdukAksesoris> getProdukList() {
        try (Session session = HibernateUtil.getSessionFactory().openSession()) {
            return session.createQuery("from ProdukAksesoris", ProdukAksesoris.class).list();
        }
    }
}

```

5. Tampilan Form NetBeans

Di NetBeans, buat form GUI menggunakan Swing. Tambahkan JTextField untuk kodeProduk, namaProduk, harga, dan kategori, serta tombol untuk menyimpan dan menampilkan data. Berikut adalah contoh method untuk menangani event klik tombol di form.

```

private void btnSimpanActionPerformed(java.awt.event.ActionEvent evt) {
    ControllerProduk controller = new ControllerProduk();
    ProdukAksesoris produk = new ProdukAksesoris();
    produk.setKodeProduk(txtKodeProduk.getText());
    produk.setNamaProduk(txtNamaProduk.getText());
    produk.setHarga(Integer.parseInt(txtHarga.getText()));
    produk.setKategori(txtKategori.getText());

    controller.insertProduk(produk);
    JOptionPane.showMessageDialog(this, "Produk berhasil disimpan!");
}

private void btnTampilkanActionPerformed(java.awt.event.ActionEvent evt) {
    ControllerProduk controller = new ControllerProduk();
    List<ProdukAksesoris> listProduk = controller.getProdukList();

    DefaultTableModel model = (DefaultTableModel) jTableProduk.getModel();
    model.setRowCount(0);
    for (ProdukAksesoris p : listProduk) {
        model.addRow(new Object[]{p.getKodeProduk(), p.getNamaProduk(), p.getHarga(), p.getKategori()});
    }
}

```

OUTPUT

Ayam Bakar 77

Kode Produk	<input type="text"/>
Nama Produk	<input type="text"/>
Harga	<input type="text"/>
Kategori	<input type="text"/>

Simpan

Tampilkan