

**LAPORAN TUGAS KULIAH
PEMROGRAMAN BERORIENTASI OBJEK**



Dosen Pengajar :

Bayu Adhi Nugroho, Ph.D

Disusun oleh :

ARRIO EKA FIRMANSYAH (09020622021)

UNIVERSITAS ISLAM NEGERI SUNAN AMPEL SURABAYA

FAKULTAS SAINS DAN TEKNOLOGI

SISTEM INFORMASI

SURABAYA

2023

TUGAS : Membuat koneksi database menggunakan persistence

Pilihan menu :

1. Membuat CLASS buku dan berisi bagian bagian yang dibutuhkan (isbn, judul, tahun terbit, penerbit)

```
public class BukuPOJO {  
  
    private String isbn;  
    private String judulBuku;  
    private String tahunTerbit;  
    private String penerbit;  
  
    /**  
     * @return the isbn  
     */  
    public String getIsbn() {  
        return isbn;  
    }  
  
    /**  
     * @param isbn the isbn to set  
     */  
    public void setIsbn(String isbn) {  
        this.isbn = isbn;  
    }  
  
    /**  
     * @return the judulBuku  
     */  
    public String getJudulBuku() {  
        return judulBuku;  
    }  
  
    /**  
     * @param judulBuku the judulBuku to set  
     */  
    public void setJudulBuku(String judulBuku) {  
        this.judulBuku = judulBuku;  
    }  
  
    /**  
     * @return the tahunTerbit  
     */  
    public String getTahunTerbit() {  
        return tahunTerbit;  
    }  
}
```

2. Kemudian membuat entity CLASS, yaitu dengan klik kanan pada packages - new - entity class. Masukkan database connection kemudian finish. Lalu akan muncul seperti :

```
@Entity
@Table(name = "buku")
@XmlRootElement
@NamedQueries({
    @NamedQuery(name = "Buku.findAll", query = "SELECT b FROM Buku b"),
    @NamedQuery(name = "Buku.findByIsbn", query = "SELECT b FROM Buku b WHERE b.isbn = :isbn"),
    @NamedQuery(name = "Buku.findByJudul", query = "SELECT b FROM Buku b WHERE b.judul = :judul"),
    @NamedQuery(name = "Buku.findByTahun", query = "SELECT b FROM Buku b WHERE b.tahun = :tahun"),
    @NamedQuery(name = "Buku.findByPenerbit", query = "SELECT b FROM Buku b WHERE b.penerbit = :penerbit"))
public class Buku implements Serializable {

    private static final long serialVersionUID = 1L;
    @Id
    @Basic(optional = false)
    @Column(name = "isbn")
    private String isbn;
    @Column(name = "judul")
    private String judul;
    @Column(name = "tahun")
    private String tahun;
    @Column(name = "penerbit")
    private String penerbit;

    public Buku() {
    }

    public Buku(String isbn) {
        this.isbn = isbn;
    }

    public String getIsbn() {
        return isbn;
    }

    public void setIsbn(String isbn) {
        this.isbn = isbn;
    }

    public String getJudul() {
        return judul;
    }
}
```

3. Buat java gui sesuai kebutuhan.
4. Pada masing masing tombol tidak lagi menggunakan connection, tetapi menggunakan entity manager atau disebut dengan persistence.

UNTUK TAMPILAN TABEL :

```
private void tampil() {

    DefaultTableModel model = (DefaultTableModel) jTableBuku.getModel();
    model.setRowCount(0);

    EntityManagerFactory emf = Persistence.createEntityManagerFactory("Pertemuan9PU");
    EntityManager em = emf.createEntityManager();
    em.getTransaction().begin();

    CriteriaBuilder cb = em.getCriteriaBuilder();
    CriteriaQuery<Buku> cq = cb.createQuery(Buku.class);
    Root<Buku> stud = cq.from(Buku.class);
    cq.select(stud.get("isbn"));

    CriteriaQuery<Buku> select = cq.select(stud);
    TypedQuery<Buku> q = em.createQuery(select);
    List<Buku> list = q.getResultList();

    for (Buku data : list) {
        Object[] baris = new Object[4];
        baris[0] = data.getIsbn();
        baris[1] = data.getJudul();
        baris[2] = data.getTahun();
        baris[3] = data.getPenerbit();

        model.addRow(baris);
    }

    em.getTransaction().commit();
    em.close();
    emf.close();
}
```

UNTUK TOMBOL SIMPAN :

```
private void jButtonSimpanActionPerformed(java.awt.event.ActionEvent evt) {

    try {
        String isbn = jTextFieldISBN.getText().trim();
        String judul = jTextFieldJudul.getText();
        String tahun = jTextFieldTahun.getText();
        String penerbit = jTextFieldPenerbit.getText();

        EntityManager entityManager = Persistence.createEntityManagerFactory("Pertemuan9PU");
        entityManager.getTransaction().begin();
        CriteriaBuilder cb = entityManager.getCriteriaBuilder();

        Buku b = new Buku();

        b.setIsbn(isbn);
        b.setJudul(judul);
        b.setTahun(tahun);
        b.setPenerbit(penerbit);

        entityManager.persist(b);
        entityManager.getTransaction().commit();

        JOptionPane.showMessageDialog(parentComponent, "Simpan Berhasil");
    } catch (Exception ex) {
        JOptionPane.showMessageDialog(parentComponent, "Simpan Gagal");
    }

    jTextFieldISBN.setText("");
    jTextFieldJudul.setText("");
    jTextFieldTahun.setText("");
    jTextFieldPenerbit.setText("");

    tampil();
}
```

UNTUK TOMBOL EDIT :

```
private void jButtonEditActionPerformed(java.awt.event.ActionEvent evt) {  
  
    try {  
        String isbn = jTextFieldISBN.getText().trim();  
        String judul = jTextFieldJudul.getText();  
        String tahun = jTextFieldTahun.getText();  
        String penerbit = jTextFieldPenerbit.getText();  
        EntityManager entityManager = Persistence.createEntityManagerFactory("Pertemuan9PU")  
  
        entityManager.getTransaction().begin();  
        Buku b = entityManager.find(type: Buku.class, o: isbn);  
  
        b.setIsbn(isbn);  
        b.setJudul(judul);  
        b.setTahun(tahun);  
        b.setPenerbit(penerbit);  
  
        entityManager.persist(o: b);  
        entityManager.getTransaction().commit();  
        JOptionPane.showMessageDialog(parentComponent: null, message: "Ubah Berhasil");  
    } catch (Exception ex) {  
        JOptionPane.showMessageDialog(parentComponent: null, message: "Ubah Gagal");  
    }  
  
    jTextFieldISBN.setText(t: "");  
    jTextFieldJudul.setText(t: "");  
    jTextFieldTahun.setText(t: "");  
    jTextFieldPenerbit.setText(t: "");  
  
    tampil();  
}  
}
```

UNTUK TOMBOL HAPUS :

```
private void jButtonHapusActionPerformed(java.awt.event.ActionEvent evt) {  
  
    try {  
        String isbn = jTextFieldISBN.getText().trim();  
        String judul = jTextFieldJudul.getText();  
        String tahun = jTextFieldTahun.getText();  
        String penerbit = jTextFieldPenerbit.getText();  
        EntityManager entityManager = Persistence.createEntityManagerFactory("Pertemuan9PU")  
  
        entityManager.getTransaction().begin();  
        Buku b = entityManager.find(type: Buku.class,  
                                     o: isbn);  
  
        b.setIsbn(isbn);  
        b.setJudul(judul);  
        b.setTahun(tahun);  
        b.setPenerbit(penerbit);  
  
        entityManager.persist(o: b);  
        entityManager.remove(o: b);  
        entityManager.getTransaction().commit();  
        entityManager.close();  
        JOptionPane.showMessageDialog(parentComponent: null, message: "Hapus Berhasil");  
    } catch (Exception ex) {  
        JOptionPane.showMessageDialog(parentComponent: null, message: "Hapus Gagal");  
    }  
  
    jTextFieldISBN.setText(t: "");  
    jTextFieldJudul.setText(t: "");  
    jTextFieldTahun.setText(t: "");  
    jTextFieldPenerbit.setText(t: "");  
  
    tampil();  
}  
}
```

UNTUK TOMBOL CETAK :

```
private void jButtonCetakActionPerformed(java.awt.event.ActionEvent evt) {  
  
    String jrxmlFile = "src/Pertemuan9/report1.jrxml";  
    EntityManagerFactory emf = Persistence.createEntityManagerFactory(persistenceUnitName: "Pertemuan9PU");  
    EntityManager em = emf.createEntityManager();  
    em.getTransaction().begin();  
  
    CriteriaBuilder cb = em.getCriteriaBuilder();  
    CriteriaQuery<Buku> cq = cb.createQuery(type: Buku.class);  
    Root<Buku> stud = cq.from(type: Buku.class);  
    cq.select(select: stud);  
  
    TypedQuery<Buku> q = em.createQuery(cq);  
    List<Buku> list = q.getResultList();  
    Query query = em.createQuery(string: "SELECT b FROM Buku b");  
    List<Buku> result = query.getResultList();  
  
    JRBeanCollectionDataSource dataSource = new JRBeanCollectionDataSource(beanCollection: result);  
  
    try {  
        // TODO add your handling code here:  
        // JasperPrint jp = JasperFillManager.fillReport(jasperFile.getPath(), null, conn);  
        JasperReport jr = JasperCompileManager.compileReport(sourceFileName: jrxmlFile);  
        JasperPrint jp = JasperFillManager.fillReport(jasperReport: jr, parameters: null, dataSource);  
        JasperViewer jv = new JasperViewer(jasperPrint: jp, isExitOnClose: false);  
        jv.setVisible(b: true);  
    } catch (JRException ex) {  
        Logger.getLogger(name: TampilBuku.class  
            .getName()).log(level: Level.SEVERE, msg: null, thrown: ex);  
    }  
}
```

4. TAMPILAN GUI :

ISBN	<input type="text"/>
Judul Buku	<input type="text"/>
Tahun Terbit	<input type="text"/>
Penerbit	<input type="text"/>
<div>SimpanHapusEdit</div> <div>Cetak</div>	

ISBN	Judul Buku	Tahun Terbit	Penerbit
00003	aljabar	2002	bse
00004	hadist	2020	ganesa
00005	mpb	2000	bse
00002	studi alquran	2011	airlangga
00001	psi	2005	bse

5. OUTPUT REPORT :

Daftar Buku Perpustakaan UINSA



ISBN	Judul Buku	Tahun Terbit	Penerbit
00003	aljabar	2002	bse
00004	hadist	2020	ganesa
00005	mpb	2000	bse
00002	studi alquran	2011	airiangga
00001	psi	2005	bse