

**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 : Menambahkan upload csv dengan 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, message: "Simpan Berhasil");  
    } catch (Exception ex) {  
        JOptionPane.showMessageDialog(parentComponent, message: "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(Buku.class, isbn);  
  
        b.setIsbn(isbn);  
        b.setJudul(judul);  
        b.setTahun(tahun);  
        b.setPenerbit(penerbit);  
  
        entityManager.persist(b);  
        entityManager.getTransaction().commit();  
        JOptionPane.showMessageDialog(parentComponent, message: "Ubah Berhasil");  
    } catch (Exception ex) {  
        JOptionPane.showMessageDialog(parentComponent, message: "Ubah Gagal");  
    }  
  
    jTextFieldISBN.setText("");  
    jTextFieldJudul.setText("");  
    jTextFieldTahun.setText("");  
    jTextFieldPenerbit.setText("");  
  
    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(Buku.class,  
            isbn);  
  
        b.setIsbn(isbn);  
        b.setJudul(judul);  
        b.setTahun(tahun);  
        b.setPenerbit(penerbit);  
  
        entityManager.persist(b);  
        entityManager.remove(b);  
        entityManager.getTransaction().commit();  
        entityManager.close();  
        JOptionPane.showMessageDialog(parentComponent, message: "Hapus Berhasil");  
    } catch (Exception ex) {  
        JOptionPane.showMessageDialog(parentComponent, message: "Hapus Gagal");  
    }  
    jTextFieldISBN.setText("");  
    jTextFieldJudul.setText("");  
    jTextFieldTahun.setText("");  
    jTextFieldPenerbit.setText("");  
  
    tampil();  
}
```

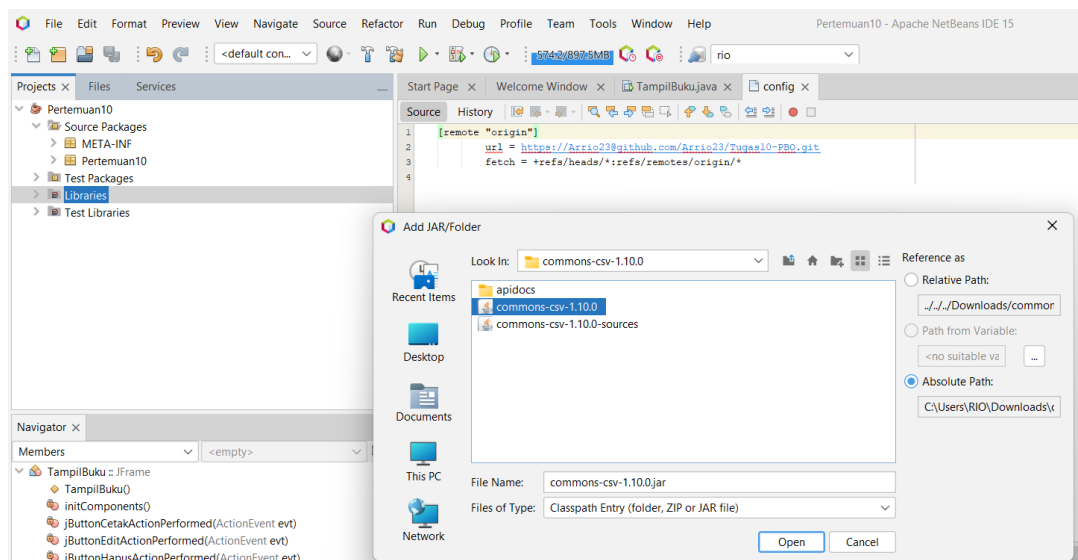
## UNTUK TOMBOL CETAK :

```
private void jButtonCetakActionPerformed(java.awt.event.ActionEvent evt) {  
  
    String jxmlFile = "src/Pertemuan9/report1.jxml";  
    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);  
  
    TypedQuery<Buku> q = em.createQuery(cq);  
    List<Buku> list = q.getResultList();  
    Query query = em.createQuery("SELECT b FROM Buku b");  
    List<Buku> result = query.getResultList();  
  
    JRBeanCollectionDataSource dataSource = new JRBeanCollectionDataSource(result);  
  
    try {  
        // TODO add your handling code here:  
        // JasperPrint jp = JasperFillManager.fillReport(jasperFile.getPath(), null, conn);  
        JasperReport jr = JasperCompileManager.compileReport(jxmlFile);  
        JasperPrint jp = JasperFillManager.fillReport(jr, parameters: null, dataSource);  
        JasperViewer jv = new JasperViewer(jp, isExitOnClose: false);  
        jv.setVisible(true);  
    } catch (JRException ex) {  
        Logger.getLogger(TampilBuku.class.getName()).log(Level.SEVERE, msg: null, thrown: ex);  
    }  
}
```

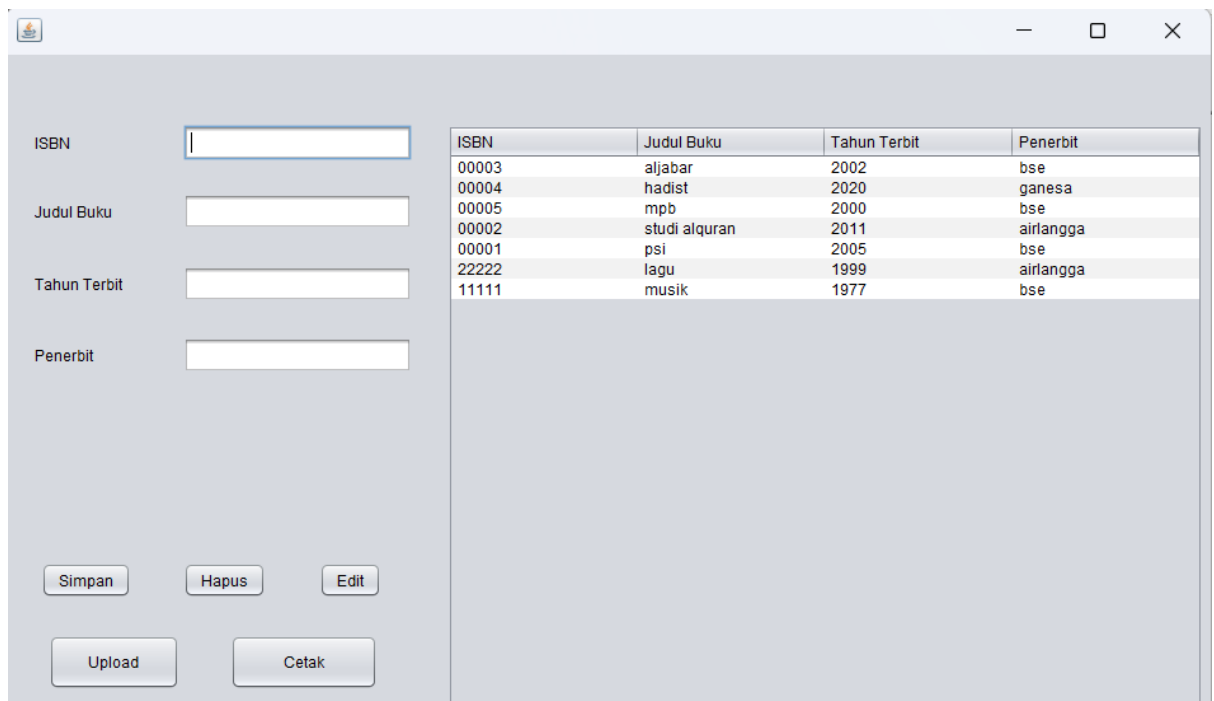
## UNTUK TOMBOL UPLOAD

```
private void jButtonUploadActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    JFileChooser filechooser = new JFileChooser();  
    EntityManager entityManager = Persistence.createEntityManagerFactory("Pertemuan9PU").createEntityManager();  
  
    entityManager.getTransaction().begin();  
  
    int i = filechooser.showOpenDialog(parent: null);  
    if (i == JFileChooser.APPROVE_OPTION) {  
        File f = filechooser.getSelectedFile();  
        String filepath = f.getPath();  
        String fi = f.getName();  
        //Parsing CSV Data  
  
        try {  
            InputStreamReader inputStreamReader = new InputStreamReader(new FileInputStream(string: filepath));  
            org.apache.commons.csv.CSVParser csvParser = CSVFormat.DEFAULT.parse(reader: inputStreamReader);  
            for (CSVRecord csvRecord : csvParser) {  
                Buku b = new Buku();  
  
                b.setIsbn(isbn: csvRecord.get(i: 0));  
                b.setJudul(judul: csvRecord.get(i: 1));  
                b.setTahun(tahun: csvRecord.get(i: 2));  
                b.setPenerbit(penerbit: csvRecord.get(i: 3));  
                entityManager.persist(o: b);  
            }  
        } catch (Exception ex) {  
            System.out.println("Error in Parsing CSV File");  
        }  
    }  
  
    entityManager.getTransaction().commit();  
    tampil();  
}
```

## 5. TAMBAHKAN JAR CSV PADA LIBRARY

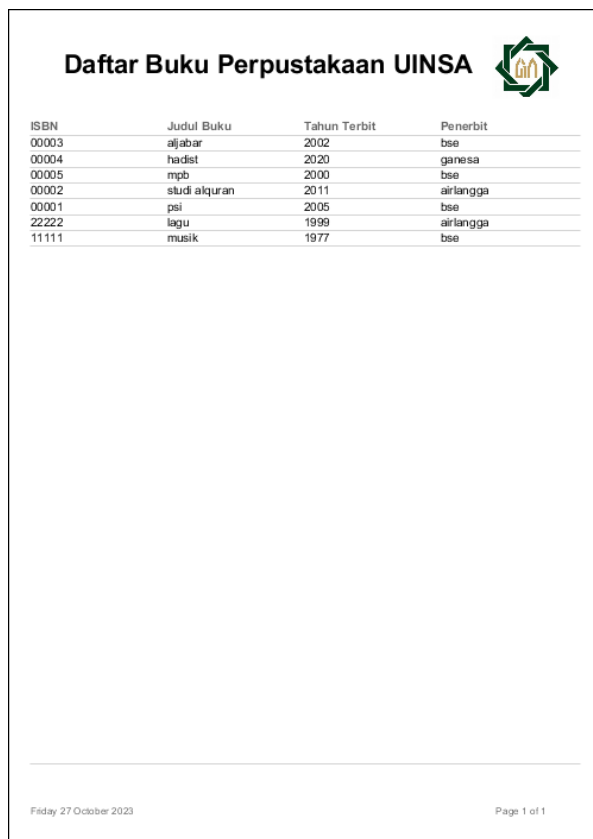


#### 4. TAMPILAN GUI :



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
22222	lagu	1999	airlangga
11111	musik	1977	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	airlangga
00001	psi	2005	bse
22222	lagu	1999	airlangga
11111	musik	1977	bse

Friday 27 October 2023 Page 1 of 1

**LINK GITHUB :**

**TUGAS 10 : <https://github.com/Arrio23/Tugas10-PBO>**

**TUGAS 9 : <https://github.com/Arrio23/Tugas9-PBO>**

**UTS : <https://github.com/Arrio23/UTS-PBO>**

**TUGAS 6 : <https://github.com/Arrio23/Tugas6-PBO>**

**TUGAS 5 : <https://github.com/Arrio23/Tugas5-PBO>**

**TUGAS 4 : <https://github.com/Arrio23/Tugas4-PBO>**

**TUGAS 3 : <https://github.com/Arrio23/Tugas3-PBO>**

**TUGAS 2 : <https://github.com/Arrio23/Tugas2-PBO>**

**TUGAS 1 : <https://github.com/Arrio23/Tugas1-PBO>**