

LAPORAN TUGAS
PEMROGRAMAN BERBASIS OBJEK
PERTEMUAN KESEMBILAN 17 OKTOBER 2023



UIN SUNAN AMPEL
S U R A B A Y A

DOSEN PEMBIMBING

Bayu Adhi Nugroho, Ph.D.

(197905182014031001)

DISUSUN OLEH

Mochamad Roiyan Rintiarno

(09020622033)

UIN SUNAN AMPEL SURABAYA

TAHUN 2023

1. Tugas

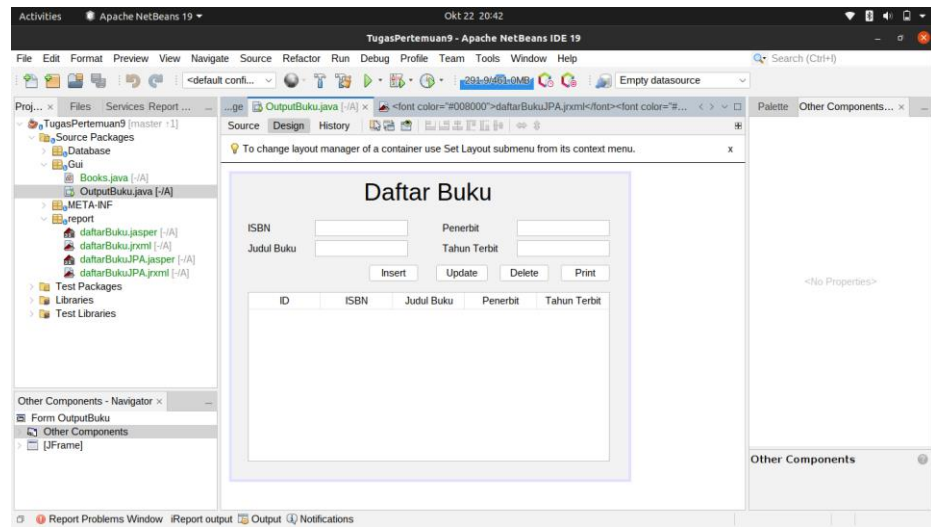
→ Membuat CRUD berbasis Persistence berdasarkan soal UTS

2. Pembahasan dan Isi

a. Persiapan dan Langkah-langkah

1. Membuat Desain

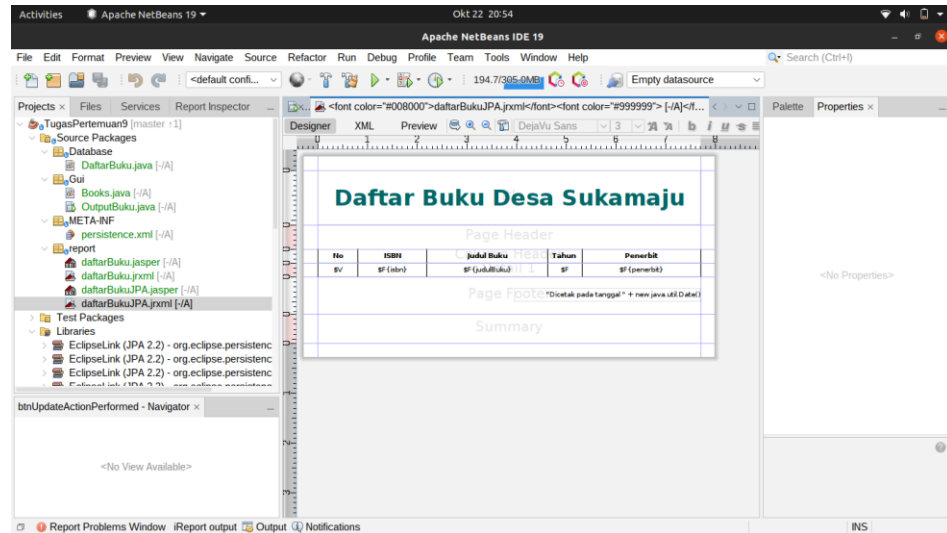
a. Desain Aplikasi Java



i. Source Code

[illegible]

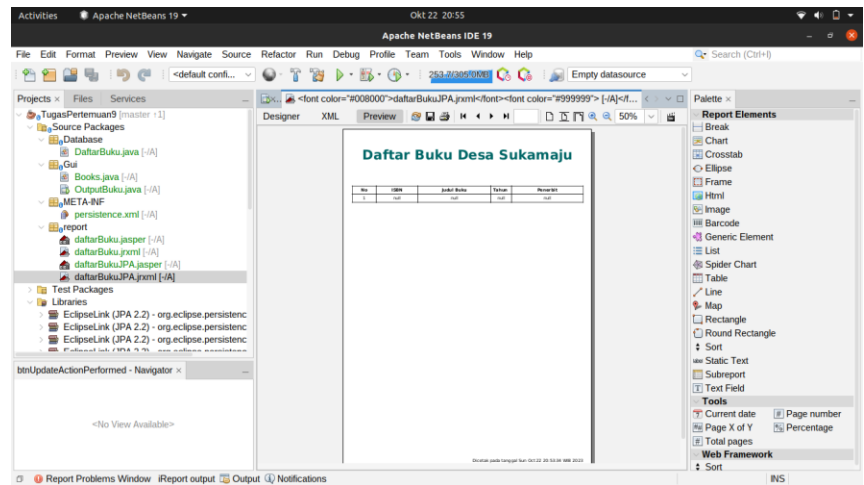
b. Desain Report



i. Source Code

[illegible]

ii. Preview



2. Menambahkan Source Code

- Membuat kelas persistence dengan cara New => Entity Class from Database, dalam hal ini namanya adalah kelas DaftarBuku. Java

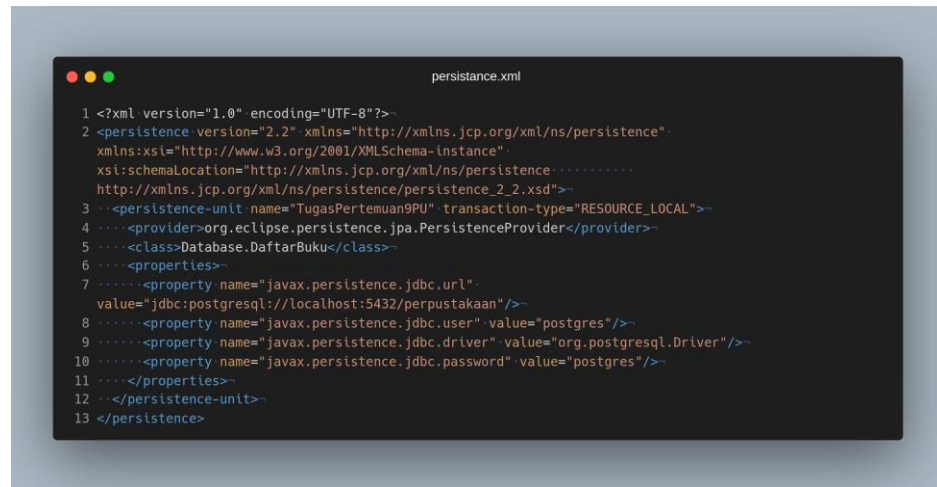
```

DaftarBuku.java

1 import java.io.Serializable;
2 import javax.persistence.Basic;
3 import javax.persistence.Column;
4 import javax.persistence.Entity;
5 import javax.persistence.GeneratedValue;
6 import javax.persistence.GenerationType;
7 import javax.persistence.Id;
8 import javax.persistence.NamedQueries;
9 import javax.persistence.NamedQuery;
10 import javax.persistence.Table;
11
12 /**
13  *
14  * @author milea
15  */
16 @Entity
17 @Table(name = "daftar_buku")
18 @NamedQueries({
19     @NamedQuery(name = "DaftarBuku.findAll", query = "SELECT d FROM DaftarBuku d"),
20     @NamedQuery(name = "DaftarBuku.findByIsbn", query = "SELECT d FROM DaftarBuku d WHERE d.isbn = :isbn"),
21     @NamedQuery(name = "DaftarBuku.findByJudulBuku", query = "SELECT d FROM DaftarBuku d WHERE d.judulBuku = :judulBuku"),
22     @NamedQuery(name = "DaftarBuku.findByTahunTerbit", query = "SELECT d FROM DaftarBuku d WHERE d.tahunTerbit = :tahunTerbit"),
23     @NamedQuery(name = "DaftarBuku.findByPenerbit", query = "SELECT d FROM DaftarBuku d WHERE d.penerbit = :penerbit"),
24     @NamedQuery(name = "DaftarBuku.findByIdBuku", query = "SELECT d FROM DaftarBuku d WHERE d.idBuku = :idBuku")
25 })
26 public class DaftarBuku implements Serializable {
27     private static final long serialVersionUID = 1L;
28     @Column(name = "isbn")
29     private String isbn;
30     @Column(name = "judul_buku")
31     private String judulBuku;
32     @Column(name = "tahun_terbit")
33     private String tahunTerbit;
34     @Column(name = "penerbit")
35     private String penerbit;
36     @Id
37     @GeneratedValue(strategy = GenerationType.IDENTITY)
38     @Basic(optional = false)
39     @Column(name = "id_buku")
40     private Integer idBuku;
41
42     public DaftarBuku() {}
43
44     public DaftarBuku(Integer idBuku) {
45         this.idBuku = idBuku;
46     }
47
48     public String getIsbn() {
49         return isbn;
50     }
51
52     public void setIsbn(String isbn) {
53         this.isbn = isbn;
54     }
55
56     public String getJudulBuku() {
57         return judulBuku;
58     }
59
60     public void setJudulBuku(String judulBuku) {
61         this.judulBuku = judulBuku;
62     }
63
64     public String getTahunTerbit() {
65         return tahunTerbit;
66     }
67
68     public void setTahunTerbit(String tahunTerbit) {
69         this.tahunTerbit = tahunTerbit;
70     }
71
72     public String getPenerbit() {
73         return penerbit;
74     }
75
76     public void setPenerbit(String penerbit) {
77         this.penerbit = penerbit;
78     }
79
80     public Integer getIdBuku() {
81         return idBuku;
82     }
83
84     public void setIdBuku(Integer idBuku) {
85         this.idBuku = idBuku;
86     }
87
88     @Override
89     public int hashCode() {
90         int hash = 0;
91         hash += (idBuku != null ? idBuku.hashCode() : 0);
92         return hash;
93     }
94
95     @Override
96     public boolean equals(Object object) {
97         // TODO: Warning - this method won't work in the case the id fields are not set
98         if (!(object instanceof DaftarBuku)) {
99             return false;
100         }
101         DaftarBuku other = (DaftarBuku) object;
102         if ((this.idBuku == null && other.idBuku != null) || (this.idBuku != null && !this.idBuku.equals(other.idBuku))) {
103             return false;
104         }
105         return true;
106     }
107
108     @Override
109     public String toString() {
110         return "Database.DaftarBuku[" + idBuku + " ]";
111     }
112
113 }
114

```

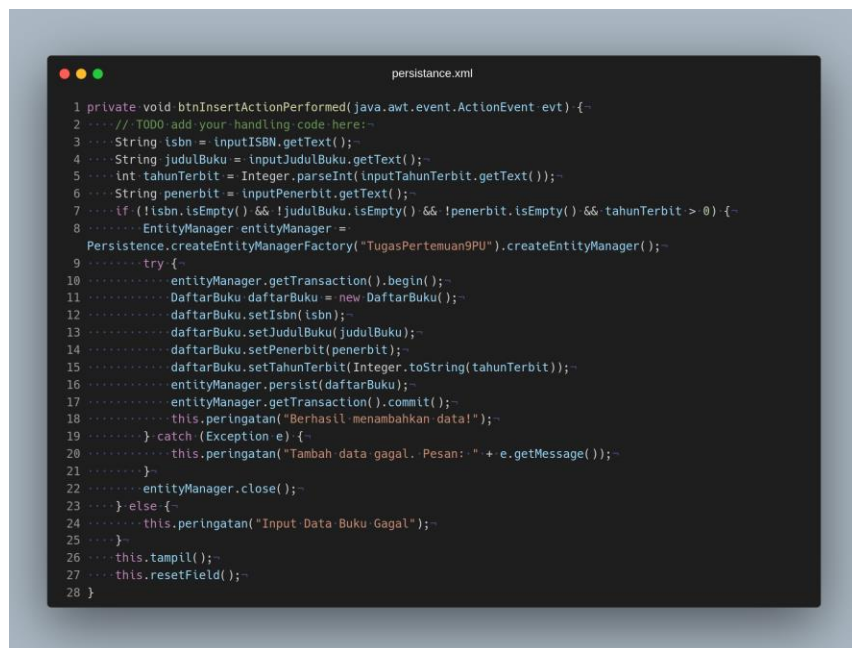
- b. Kemudian akan langsung muncul file yang bernama persistence.xml yang berada pada folder META-INF



```
persistence.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <persistence version="2.2" xmlns="http://xmlns.jcp.org/xml/ns/persistence"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
    http://xmlns.jcp.org/xml/ns/persistence/persistence_2_2.xsd">
3   <persistence-unit name="TugasPertemuan9PU" transaction-type="RESOURCE_LOCAL">
4     <provider>org.eclipse.persistence.jpa.PersistenceProvider</provider>
5     <class>Database.DaftarBuku</class>
6     <properties>
7       <property name="javax.persistence.jdbc.url"
8         value="jdbc:postgresql://localhost:5432/perpustakaan"/>
9       <property name="javax.persistence.jdbc.user" value="postgres"/>
9       <property name="javax.persistence.jdbc.driver" value="org.postgresql.Driver"/>
10      <property name="javax.persistence.jdbc.password" value="postgres"/>
11    </properties>
12  </persistence-unit>
13 </persistence>
```

- c. Langkah selanjutnya adalah menambahkan kode pada masing masing button, seperti insert, update, delete, cetak, serta lainnya

i. Source Code Insert



```
persistence.xml
1 private void btnInsertActionPerformed(java.awt.event.ActionEvent evt) {
2   // TODO: add your handling code here:
3   String isbn = inputISBN.getText();
4   String judulBuku = inputJudulBuku.getText();
5   int tahunTerbit = Integer.parseInt(inputTahunTerbit.getText());
6   String penerbit = inputPenerbit.getText();
7   if (!isbn.isEmpty() && !judulBuku.isEmpty() && !penerbit.isEmpty() && tahunTerbit > 0) {
8     EntityManager entityManager =
9       Persistence.createEntityManagerFactory("TugasPertemuan9PU").createEntityManager();
10    try {
11      entityManager.getTransaction().begin();
12      DaftarBuku daftarBuku = new DaftarBuku();
13      daftarBuku.setIsbn(isbn);
14      daftarBuku.setJudulBuku(judulBuku);
15      daftarBuku.setPenerbit(penerbit);
16      daftarBuku.setTahunTerbit(Integer.toString(tahunTerbit));
17      entityManager.persist(daftarBuku);
18      entityManager.getTransaction().commit();
19      this.peringatan("Berhasil menambahkan data!");
20    } catch (Exception e) {
21      this.peringatan("Tambah data gagal. Pesan: " + e.getMessage());
22    }
23    entityManager.close();
24  } else {
25    this.peringatan("Input Data Buku Gagal");
26  }
27  this.tampil();
28  this.resetField();
29 }
```

ii. Source Code Update


```

OutputBuku.java

1 private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) {~
2     // TODO add your handling code here:~
3     if (this.id != 0) {~
4         String isbn = inputISBN.getText();~
5         String judulBuku = inputJudulBuku.getText();~
6         int tahunTerbit = Integer.parseInt(inputTahunTerbit.getText());~
7         String penerbit = inputPenerbit.getText();~
8         if (!isbn.isEmpty() && !judulBuku.isEmpty() && !penerbit.isEmpty() && tahunTerbit > 0) {~
9             EntityManager entityManager =
10                 Persistence.createEntityManagerFactory("TugasPertemuan9PU").createEntityManager();~
11             try {~
12                 entityManager.getTransaction().begin();~
13                 DaftarBuku daftarBuku = entityManager.find(DaftarBuku.class, this.id);~
14                 daftarBuku.setIsbn(isbn);~
15                 daftarBuku.setJudulBuku(judulBuku);~
16                 daftarBuku.setPenerbit(penerbit);~
17                 daftarBuku.setTahunTerbit(Integer.toString(tahunTerbit));~
18                 entityManager.merge(daftarBuku);~
19                 entityManager.getTransaction().commit();~
20                 this.peringatan("Berhasil mengubah data!");~
21             } catch (Exception e) {~
22                 entityManager.getTransaction().rollback();~
23                 this.peringatan("Gagal mengubah data. Pesan peringatan: " + e.getMessage());~
24             }~
25             entityManager.close();~
26         } else {~
27             this.peringatan("Update Data Mahasiswa Gagal!");~
28         }~
29         this.tampil();~
30         this.resetField();~
31     } else {~
32         this.peringatan("Pilih data terlebih dahulu!");~
33     }~
34 }

```

iii. Source Code Delete

```

OutputBuku.java

1 private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {~
2     // TODO add your handling code here:~
3     if (this.id > 0) {~
4         EntityManager entityManager =
5             Persistence.createEntityManagerFactory("TugasPertemuan9PU").createEntityManager();~
6         try {~
7             entityManager.getTransaction().begin();~
8             DaftarBuku daftarBuku = entityManager.find(DaftarBuku.class, this.id);~
9             entityManager.remove(daftarBuku);~
10            entityManager.getTransaction().commit();~
11            this.peringatan("Berhasil menghapus data!");~
12        } catch (Exception e) {~
13            this.peringatan("Gagal menghapus data. Pesan: " + e.getMessage());~
14        }~
15        entityManager.close();~
16    } else {~
17        this.peringatan("Pilih data terlebih dahulu!");~
18    }~
19    this.resetField();~
20    this.tampil();~
21 }

```

iv. Source Code Print

```
OutputBuku.java

1 private void btnPrintActionPerformed(java.awt.event.ActionEvent evt) {.....
2 ....//TODO: add your handling code here:-
3 .... try {-
4 ....     EntityManager entityManager =
5 ....         Persistence.createEntityManagerFactory("TugasPertemuan9PU").createEntityManager();~
6 ....     entityManager.getTransaction().begin();~
7 ....     TypedQuery<DaftarBuku> querySelectAll = entityManager.createNamedQuery("DaftarBuku.findAll",
8 ....         DaftarBuku.class);~
9 ....     List<DaftarBuku> results = querySelectAll.getResultList();~
10 ....     entityManager.getTransaction().commit();~
11 ....     entityManager.close();~
12 ....     String jrxmlFile = new String("src/report/daftarBukuJPA.jrxml");~
13 ....     JasperReport jr = JasperCompileManager.compileReport(jrxmlFile);~
14 ....     JasperPrint jp = JasperFillManager.fillReport(jr, null, new
15 ....         JRBeanCollectionDataSource(results));~
16 ....     JasperViewer.viewReport(jp, false);~
17 .... } catch (JRException ex) {-
18 ....     Logger.getLogger(OutputBuku.class.getName()).log(Level.SEVERE, null, ex);~
19 .... } catch (Exception e) {-
20 ....     Logger.getLogger(OutputBuku.class.getName()).log(Level.SEVERE, null, e);~
21 .... }
22 }
```

v. Source Code Tampil

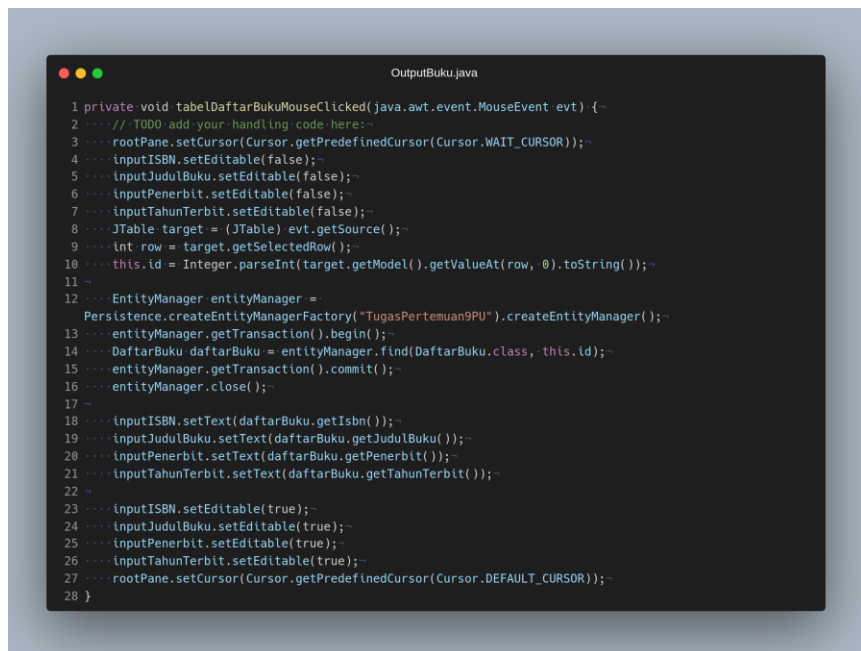
```
OutputBuku.java

1 private void tampil() {-
2 .... EntityManager entityManager =
3 ....     Persistence.createEntityManagerFactory("TugasPertemuan9PU").createEntityManager();~
4 ....     entityManager.getTransaction().begin();~
5 ....     TypedQuery<DaftarBuku> querySelectAll = entityManager.createNamedQuery("DaftarBuku.findAll",
6 ....         DaftarBuku.class);~
7 ....     List<DaftarBuku> results = querySelectAll.getResultList();~
8 ....     DefaultTableModel model = (DefaultTableModel) tabelDaftarBuku.getModel();~
9 ....     model.setRowCount(0);~
10 ....     for (DaftarBuku data : results) {-
11 ....         Object[] baris = new Object[5];~
12 ....         baris[0] = data.getIdBuku();~
13 ....         baris[1] = data.getIsbn();~
14 ....         baris[2] = data.getJudulBuku();~
15 ....         baris[3] = data.getTahunTerbit();~
16 ....         baris[4] = data.getPenerbit();~
17 ....         model.addRow(baris);~
18 ....     }~
19 ....     entityManager.getTransaction().commit();~
20 ....     entityManager.close();~
21 }
22 }
```

vi. Source Code reset field



vii. Source Code



3. Output

a. Tampilan Awal

Output Buku.jpg 5/10/2020

Daftar Buku

ISBN Penerbit

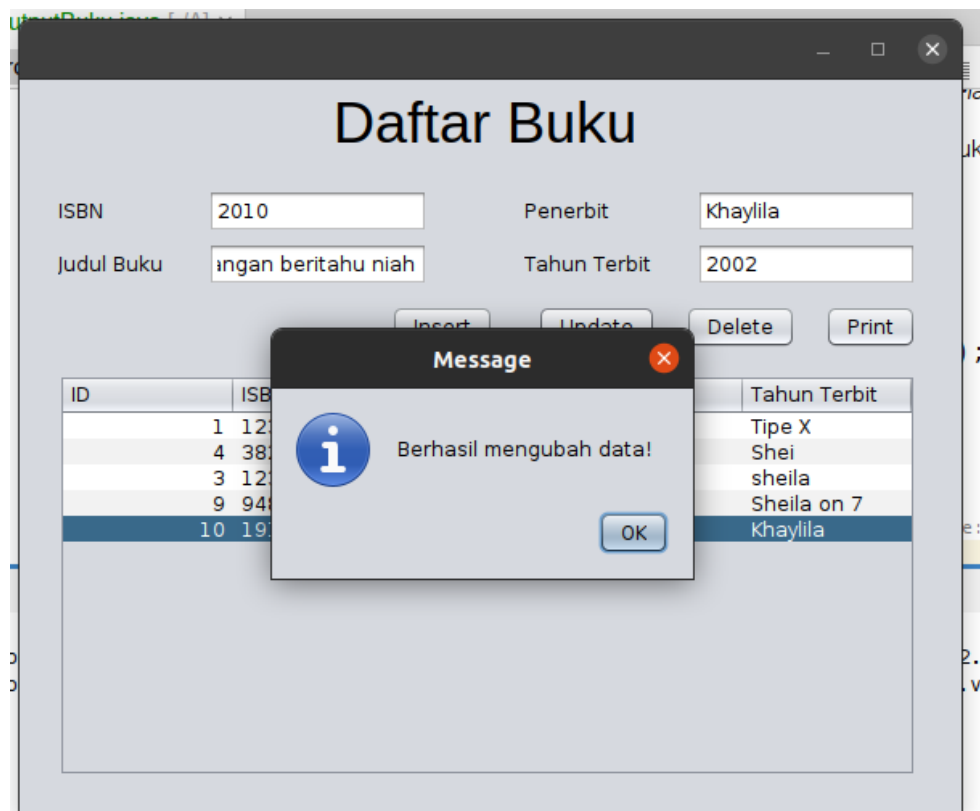
Judul Buku Tahun Terbit

ID	ISBN	Judul Buku	Penerbit	Tahun Terbit
1	12312	Mereka tak p...	2019	Tipe X
4	3829	merea	2004	Shei
3	12341	dan	2001	sheila
9	9483	Lapang Dada	2014	Sheila on 7

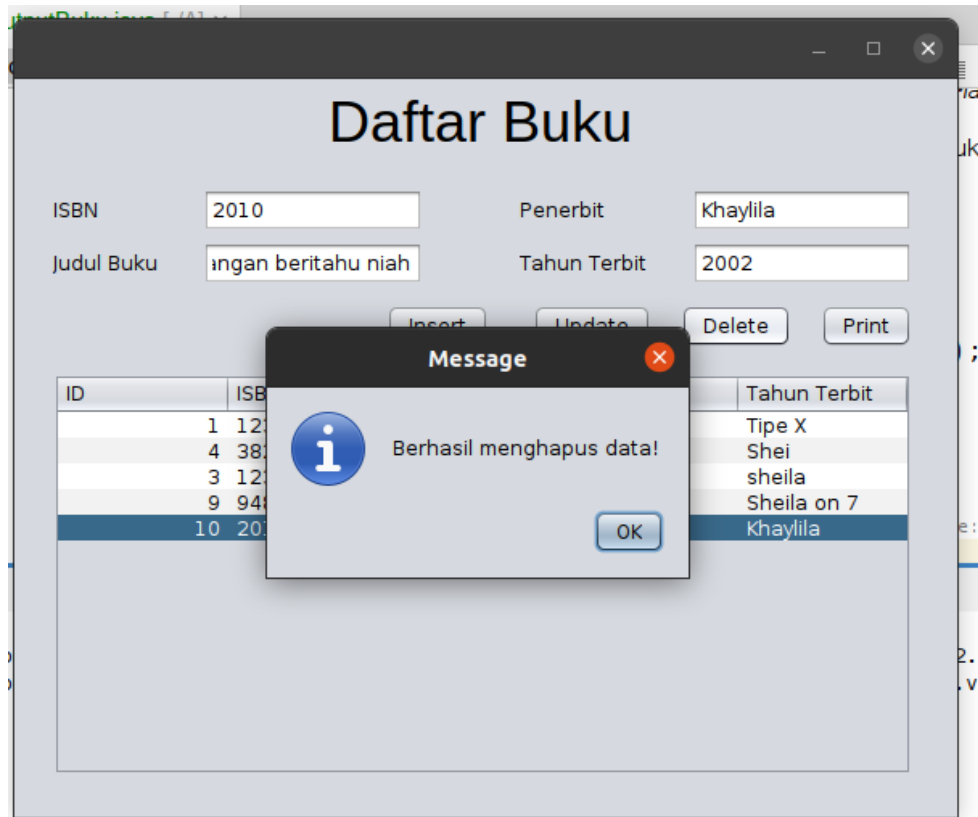
b. Ketika menambahkan data



c. Ketika mengubah data



d. Ketika menghapus data



e. Ketika mencetak data

