

**PRAKTIKUM PEMOGRAMA
BERORIENTASI OBYEK LANJUTAN
KELAS C**



DISUSUN OLEH :

Nama : Johanes Yogtan Wicaksono Raharja
NIM : 215314105

**TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS SANATA DHARMA YOGYAKARTA
2022**

Dasar Teori

1. Buatlah suatu kelas Latihan WriteDataDemo di bawah ini! Kelas ini sudah dilengkapi dengan Throw. Perhatikan isi file latihan.txt.

```
package Contoh;
import java.io.*;
public class WriteDataDemo {

    public WriteDataDemo() {
    }

    public static void main(String[] args) throws IOException {
        // write the data out with FileOutputStream
        // buat file
        File outputFile = new File("latihan.txt");
        // buka stream
        FileOutputStream out = new FileOutputStream(outputFile);
        // sediakan data stringF
        String testOutput = "This is an exercise to write data to a file with FileOutputStream ";
        /* Cara 1 : tulis data ke stream
        for(int i=0;i<testOutput.length();i++){
            out.write((byte)testOutput.charAt(i));
        }
        */
        /*Cara 2 : tulis data ke stream*/
        out.write(testOutput.getBytes());
        // tutup stream
        out.close();
    }
}
```

Name	Status	Date modified	Type	Size
build	🔄	11/10/2022 8:48 PM	File folder	
dist	✅	10/10/2022 9:35 PM	File folder	
nbproject	✅	10/10/2022 9:35 PM	File folder	
src	🔄	10/10/2022 9:35 PM	File folder	
test	✅	11/10/2022 8:48 PM	File folder	
build.xml	✅	9/13/2021 6:11 AM	XML Document	4 KB
latihan.txt	🔄	11/22/2022 8:43 AM	Text Document	1 KB
manifest.mf	✅	9/13/2021 6:11 AM	MF File	1 KB

2. Buatlah suatu kelas Latihan ReadDataDemo di bawah ini! Perhatikan isi file latihan.txt.

```

package Contoh;

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
public class ReadDataDemo {

    public ReadDataDemo() {
    }

    public static void main(String[] args) {
        // write the data out with FileOutputStream
        // buat file
        File inputFile = new File("latihan.txt");
        try {
            // buka stream
            FileInputStream in = new FileInputStream(inputFile);
            // buat buffer pembacaan data
            String testInput = new String();
            // variabel integer untuk membaca data dari file yang dibuka
            int c;
            // cek akhir dari file apakah bernilai -1
            while ((c = in.read()) != -1) {
                // tambahkan data pembacaan ke buffer
                testInput += (char) c;
            }
            // tutup stream
            in.close();
            System.out.println(testInput);
        } catch (IOException ex) {
            System.out.println(ex.getMessage());
        }
    }
}

```

Output - Perpustakaan (run)

```


run:
This is an exercise to write data to a file with FileOutputStream
BUILD SUCCESSFUL (total time: 0 seconds)

```

Tugas

1. Tugas ini meneruskan program Perpustakaan yang pernah anda buat. Buatlah Form Transaksi yang berfungsi untuk mengambil data Peminjam dan Kole

GFrame dTransaksi

 Kelola Data Transaksi Peminjaman

PINJAM

Data Peminjam

ID Peminjam

M1

Cari

Nama

Yogtan

Alamat

Paingan

Jumlah Pinjam

1

NIP

123456789

Transaksi Peminjaman

Proses Peminjaman

Batal

Data Koleksi

ID Koleksi

M1

Cari

Judul

Surga

Penerbit

Tan

Status

Dipinjam

ISBN/ISSN

1234

Volume

1234

Seri

1234

Batal Pesanan

Tambah Pesanan

Daftar Pesanan Koleksi

Id Koleksi	Judul	Penerbit	Status	ISBN/ISSN
M1	Surga	Tan	Dipinjam	1234

Data Transaksi

ID Transaksi	Tgl Pinjam	Tgl Kembali	ID Peminjam
T1669123984	2022-11-22	2022-12-22	M1

Detil

Simpan

Buka

Hapus

Source code untuk mencari dan menampilkan data peminjam

```
private void bCariPActionPerformed(java.awt.event.ActionEvent evt) {  
    // Untuk mencari data peminjam untuk ditampilkan  
    if (cariPeminjam(tfIdP.getText()) == null) {  
        JOptionPane.showMessageDialog(this, "ID Peminjam tidak terdefinisi!!", "Warning", JOptionPane.WARNING_MESSAGE);  
        return;  
    }  
    dtpeminjam = cariPeminjam(tfIdP.getText());  
    tfNamaP.setText(dtpeminjam.getNama());  
    tfAlamatP.setText(dtpeminjam.getAlamat());  
    tfJumlah_PinjamP.setText(Integer.toString(dtpeminjam.getMak_pinjam()));  
    char tipe = dtpeminjam.getId_peminjam().charAt(0);  
    switch (tipe) {  
        case 'D':  
            tfNidP.setText(((Dosen)dtpeminjam).getNip());  
            break;  
        case 'M':  
            tfNidP.setText(((Mahasiswa)dtpeminjam).getNim());  
            break;  
        case 'U':  
            tfNidP.setText(((Umum)dtpeminjam).getNik());  
            break;  
        default:  
            JOptionPane.showMessageDialog(this, "ID Peminjam tidak terdefinisi!!", "Warning", JOptionPane.WARNING_MESSAGE);  
            tfNidP.setText("No ID");  
    }  
}
```

Source code untuk mencari dan menampilkan data koleksi

```
private void bCariKoleksiActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    if (cariKoleksi(tfIdK.getText()) == null) {  
        JOptionPane.showMessageDialog(this, "ID Koleksi tidak terdefinisi!!", "Warning", JOptionPane.WARNING_MESSAGE);  
        return;  
    }  
    dtKoleksi = cariKoleksi(tfIdK.getText());  
    tfJudulK.setText(dtKoleksi.getJudul());  
    tfPenerbitK.setText(dtKoleksi.getPenerbit());  
    tfStatusK.setText(dtKoleksi.isStatus_pinjam() ? "Dipinjam": "Tersedia");  
    bTambahKoleksi.setEnabled(!dtKoleksi.isStatus_pinjam());  
    char tipe = dtKoleksi.getIdKoleksi().charAt(0);  
    switch (tipe) {  
        case 'D':  
            lVolumeK.setText("Format");  
            lSeriK.setText(" ");  
            tfSeriK.setVisible(Boolean.FALSE);  
            tfIsbnK.setText(((Disk)dtKoleksi).getIsbn());  
            tfVolumeK.setText(((Disk)dtKoleksi).getFormat());  
            break;  
        case 'M':  
            lVolumeK.setText("Volume");  
            lSeriK.setText("Seri");  
            tfSeriK.setVisible(Boolean.TRUE);  
            tfIsbnK.setText(((Majalah)dtKoleksi).getIssn());  
            tfVolumeK.setText(Integer.toString(((Majalah)dtKoleksi).getVolume()));  
            tfSeriK.setText(Integer.toString(((Majalah)dtKoleksi).getSeri()));  
            break;  
        case 'B':  
            lVolumeK.setText("Halaman");  
            lSeriK.setText(" ");  
            tfSeriK.setVisible(Boolean.FALSE);  
            tfIsbnK.setText(((Buku)dtKoleksi).getIsbn());  
            tfVolumeK.setText(Integer.toString(((Buku)dtKoleksi).getHalaman()));  
            break;  
        default:  
            JOptionPane.showMessageDialog(this, "ID Koleksi tidak terdefinisi!!", "Warning", JOptionPane.WARNING_MESSAGE);  
            tfNidP.setText("No ID");  
    }  
    bTambahKoleksi.setEnabled(Boolean.TRUE);  
}
```

Source code untuk menampilkan data peminjam dan koleksi (peminjam dan apa yang dipinjam) di table

```
private void tampilPesananKoleksi(){
    String isbnIssn = "";
    String[] kolom = {"Id Koleksi", "Judul", "Penerbit", "Status", "ISBN/ISSN"};
    Object[][] objData = new Object[dtPesananKoleksi.size()][5]; // 5 merupakan jumlah kolom
    int i = 0;
    for(Koleksi n : dtPesananKoleksi){
        if (n instanceof Disk){
            isbnIssn = ((Disk) n).getIsbn();
        }
        if (n instanceof Majalah){
            isbnIssn = ((Majalah) n).getIssn();
        }
        if (n instanceof Buku){
            isbnIssn = ((Buku) n).getIsbn();
        }
        objData[i] = new String[]{n.getIdKoleksi(), n.getJudul(), n.getPenerbit(), n.isStatus_pinjam() ? "Dipinjam" : "Tersedia", isbnIssn};
        i++;
    }
    tbModelPesananKoleksi = new DefaultTableModel(objData, kolom){
        @Override
        public boolean isCellEditable(int rowIndex, int colIndex) {
            return false;
        }
    };
    tbPesananKoleksi.repaint();
    tbPesananKoleksi.setModel(tbModelPesananKoleksi);
    clModelPesananKoleksi = tbPesananKoleksi.getColumnModel();
    clModelPesananKoleksi.getColumn(0).setPreferredWidth(10);
    clModelPesananKoleksi.getColumn(1).setPreferredWidth(300);
    clModelPesananKoleksi.getColumn(2).setPreferredWidth(50);
    clModelPesananKoleksi.getColumn(3).setPreferredWidth(10);
    clModelPesananKoleksi.getColumn(4).setPreferredWidth(20);
}
```

Source code untuk menampilkan trasaksinya di table

```
private void tampilTransaksi(){
    String[] kolom = {"ID Transaksi", "Tgl Pinjam", "Tgl Kembali", "ID Peminjam"};
    Object[][] objData = new Object[dtTrx.size()][4]; // 5 merupakan jumlah kolom
    int i = 0;
    for(Transaksi n : dtTrx){
        objData[i] = new String[]{n.getId_transaksi(), n.getTgl_pinjam().toString(), n.getTgl_kembali().toString(), n.getPeminjam().getId_peminjam()};
        i++;
    }
    tbModelTransaksi = new DefaultTableModel(objData, kolom){
        @Override
        public boolean isCellEditable(int rowIndex, int colIndex) {
            return false;
        }
    };
    tbTransaksi.repaint();
    tbTransaksi.setModel(tbModelTransaksi);
    clModelTransaksi = tbTransaksi.getColumnModel();
    clModelTransaksi.getColumn(0).setPreferredWidth(10);
    clModelTransaksi.getColumn(1).setPreferredWidth(300);
    clModelTransaksi.getColumn(2).setPreferredWidth(50);
    clModelTransaksi.getColumn(3).setPreferredWidth(10);
    //clModelTransaksi.getColumn(4).setPreferredWidth(20);
}
```

2. Saat proses simpan transaksi peminjaman, maka data obyek disimpan dalam sebuah file.

Source code simpan data file

```
private void btBTrxActionPerformed(java.awt.event.ActionEvent evt) {  
    //UNTUK MEMBUKA SEBUAH FILE/DATA  
    FileInputStream inFileStream = null;  
    ObjectInputStream inObjectStream = null;  
    try {  
        File inFile = new File("transaksi.dat");  
        inFileStream = new FileInputStream(inFile);  
        inObjectStream = new ObjectInputStream(inFileStream);  
        try {  
            dtTrx = (ArrayList<Transaksi>)inObjectStream.readObject();  
        } catch (ClassNotFoundException ex) {  
            Logger.getLogger(dTransaksi.class.getName()).log(Level.SEVERE, null, ex);  
        }  
    } catch (FileNotFoundException ex) {  
        Logger.getLogger(dTransaksi.class.getName()).log(Level.SEVERE, null, ex);  
    } catch (IOException ex) {  
        Logger.getLogger(dTransaksi.class.getName()).log(Level.SEVERE, null, ex);  
    } finally {  
        try {  
            inObjectStream.close();  
            inFileStream.close();  
        } catch (IOException ex) {  
            Logger.getLogger(dTransaksi.class.getName()).log(Level.SEVERE, null, ex);  
        }  
    }  
    tampilTransaksi();  
}
```

GFrame di transaksi untuk menyimpan

Daftar Pesanan Koleksi				
Id Koleksi	Judul	Penerbit	Status	ISBN/ISSN
M1	Surga	Tan	Dipinjam	1234

Data Transaksi			
ID Transaksi	Tgl Pinjam	Tgl Kembali	ID Peminjam
T1669123984	2022-11-22	2022-12-22	M1

















Detail

Simpan

Buka

Hapus

Bukti telah disimpan

Name	Status	Date modified	Type	Size
 build		11/10/2022 8:48 PM	File folder	
 dist		10/10/2022 9:35 PM	File folder	
 nbproject		10/10/2022 9:35 PM	File folder	
 src		10/10/2022 9:35 PM	File folder	
 test		11/10/2022 8:48 PM	File folder	
 build.xml		9/13/2021 6:11 AM	XML Document	4 KB
 manifest.mf		9/13/2021 6:11 AM	MF File	1 KB
 transaksi.dat		11/22/2022 1:33 PM	DAT File	1 KB