

***LAPORAN TUGAS PRAKTIKUM PEMROGRAMAN
BERORIENTASI OBJEK***

| | |
|--|--|
| Pertemuan Ke-: | |
| 14 | |
| Pembahasan: GUI dan Database | |
| NIM:1841720013 | Dosen Pengampu: Septian Enggar Sukmana, Spd, MT. |
| Nama Mahasiswa:M.iqbaluddin Al Huda | Nilai: |

TUJUAN PRAKTIKUM (10 points)

Petunjuk: tulis tujuan dari praktikum yang telah dilaksanakan

Setelah melakukan percobaan pada modul ini, mahasiswa memahami konsep:

1. Menggunakan paradigma berorientasi objek untuk interaksi dengan database
2. Membuat backend dan frontend
3. Membuat form sebagai frontend

JAWABAN PERTANYAAN (30 points)

Petunjuk: jawaban dari pertanyaan yang terdapat pada modul 6 dituliskan pada bagian kolom kotak yang telah disediakan

| | |
|--|--|
| | |
|--|--|

KODE PROGRAM DAN PENJELASAN TIAP METHODNYA (30 points)

| |
|--|
| Nama Class: Peminjaman |
| <pre>package Backend; import java.sql.ResultSet; import java.util.ArrayList; /** * * @author IQBAL */ public class Peminjaman { private int idpeminjaman, idanggota, idbuku; private Anggota ang = new Anggota(); private Buku buku = new Buku(); private String tanggalPinjam; private String tanggalKembali; public Peminjaman() { } public Peminjaman(Anggota ang, Buku buku, String tanggalPinjam, String tanggalKembali) { this.ang = ang; this.buku = buku; } }</pre> |

```

        this.tanggalPinjam = tanggalPinjam;
        this.tanggalKembali = tanggalKembali;
    }

    public int getIdanggota() {
        return idanggota;
    }

    public void setIdanggota(int idanggota) {
        this.idanggota = idanggota;
    }

    public int getIdbuku() {
        return idbuku;
    }

    public void setIdbuku(int idbuku) {
        this.idbuku = idbuku;
    }

    public int getIdpeminjaman() {
        return idpeminjaman;
    }

    public void setIdpeminjaman(int idpeminjaman) {
        this.idpeminjaman = idpeminjaman;
    }

    public Anggota getAng() {
        return ang;
    }

    public void setAng(Anggota ang) {
        this.ang = ang;
    }

    public Buku getBuku() {
        return buku;
    }

    public void setBuku(Buku buku) {
        this.buku = buku;
    }

    public String getTanggalPinjam() {
        return tanggalPinjam;
    }

```

```

    }

    public void setTanggalPinjam(String tanggalPinjam) {
        this.tanggalPinjam = tanggalPinjam;
    }

    public String getTanggalKembali() {
        return tanggalKembali;
    }

    public void setTanggalKembali(String tanggalKembali) {
        this.tanggalKembali = tanggalKembali;
    }

    public Peminjaman getById(int id) {
        Peminjaman pem = new Peminjaman();
        ResultSet rs = DBHelper.selectQuery("SELECT "
            + " p.idpeminjaman AS idpeminjaman, "
            + " p.tanggalpinjam AS tanggalpinjam, "
            + " p.tanggalkembali AS tanggalkembali, "
            + " a.idanggota AS idanggota, "
            + " a.nama AS nama, "
            + " a.alamat AS alamat, "
            + " a.telepon AS telepon, "
            + " b.idbuku AS idbuku, "
            + " b.judul AS judul, "
            + " b.penerbit AS penerbit, "
            + " b.penulis AS penulis, "
            + " k.idkategori AS idkategori, "
            + " k.nama AS nama, "
            + " k.keterangan AS keterangan "
            + " FROM peminjaman p "
            + " LEFT JOIN buku b ON b.idbuku = p.idbuku "
            + " LEFT JOIN anggota a ON a.idanggota = p.idanggota "
            + " LEFT JOIN kategori k ON b.idkategori = k.idkategori "
            + " WHERE p.idpeminjaman = " + id + "");

        try {
            while (rs.next()) {
                pem = new Peminjaman();
                pem.setIdpeminjaman(rs.getInt("idpeminjam"));
                pem.setIdanggota(rs.getInt("idanggota"));
                pem.setIdbuku(rs.getInt("idbuku"));
                pem.setTanggalPinjam(rs.getString("tanggalpinjam"));
                pem.setTanggalKembali(rs.getString("tanggalkembali"));
                pem.setAng(new Anggota().getById(pem.getIdanggota()));
            }
        }
    }
}

```

```

        pem.setBuku(new Buku().getById(pem.getIdbuku()));

    }
    } catch (Exception e) {
        e.printStackTrace();
    }
    }
    return pem;
}

public ArrayList<Peminjaman> getAll() {
    ArrayList<Peminjaman> ListPinjam = new ArrayList();
    ResultSet rs = DBHelper.selectQuery("SELECT "
        + " p.idpeminjaman AS idpeminjaman, "
        + " p.tanggalpinjam AS tanggalpinjam, "
        + " p.tanggalkembali AS tanggalkembali, "
        + " a.idanggota AS idanggota, "
        + " a.nama AS nama, "
        + " a.alamat AS alamat, "
        + " a.telepon AS telepon, "
        + " b.idbuku AS idbuku, "
        + " b.judul AS judul, "
        + " b.penerbit AS penerbit, "
        + " b.penulis AS penulis, "
        + " k.idkategori AS idkategori, "
        + " k.nama AS nama, "
        + " k.keterangan AS keterangan "
        + " FROM peminjaman p "
        + " LEFT JOIN buku b ON b.idbuku = p.idbuku "
        + " LEFT JOIN anggota a ON a.idanggota = p.idanggota"
        + " LEFT JOIN kategori k ON k.idkategori = p.idkategori ");

    try {
        while (rs.next()) {
            Peminjaman pem = new Peminjaman();
            pem.setIdpeminjaman(rs.getInt("idpeminjaman"));
            pem.setIdanggota(rs.getInt("idanggota"));
            pem.setIdbuku(rs.getInt("idbuku"));
            pem.setTanggalPinjam(rs.getString("tanggalpinjam"));
            pem.setTanggalKembali(rs.getString("tanggalkembali"));
            pem.setAng(new Anggota().getById(pem.getIdanggota()));
            pem.setBuku(new Buku().getById(pem.getIdbuku()));

            ListPinjam.add(pem);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

```

    }
    return ListPinjam;
}

public void save() {
    if (getById(idpeminjaman).getIdpeminjaman() == 0) {
        String sql = "Insert into peminjaman (idanggota, idbuku, tanggalpinjam,
tanggalkembali) "
            + "values ("
            + "" + this.getAng().getIdanggota() + ", "
            + "" + this.getBuku().getIdbuku() + ", "
            + "" + this.tanggalPinjam + ", "
            + "" + this.tanggalKembali + " "
            + ")";
        this.idpeminjaman = DBHelper.insertQueryGetId(sql);
    } else {
        String sql = "Update buku set "
            + "idanggota = " + this.getAng().getIdanggota() + ", "
            + "idbuku = " + this.getBuku().getIdbuku() + ", "
            + "tanggalpinjam = " + this.tanggalPinjam + ", "
            + "tanggalkembali = " + this.tanggalKembali + """;
        DBHelper.executeQuery(sql);
    }
}

public void cariAnggota(int id) {
    ResultSet rs = DBHelper.selectQuery("Select * from anggota where idanggota = "
+ id + "");
    try {
        while (rs.next()) {
            getAng().setIdanggota(rs.getInt("idanggota"));
            getAng().setNama(rs.getString("nama"));
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public void cariBuku(int id) {
    ResultSet rs = DBHelper.selectQuery("Select * from buku where idbuku = " + id
+ "");
    try {
        while (rs.next()) {
            getBuku().setIdbuku(rs.getInt("idbuku"));
            getBuku().setJudul(rs.getString("judul"));
        }
    }
}

```

| | | |
|---|--|--|
| <pre> } catch (Exception e) { e.printStackTrace(); } } public void delete() { String sql = "Delete from peminjaman where idpeminjam = '" + this.idpeminjaman + ""'; DBHelper.executeQuery(sql); } } </pre> | | |
| public Peminjaman() | | Merupakan konstruktor default |
| public Peminjaman(Anggota ang, Buku buku, String tanggalPinjam, String tanggalKembali) | | Merupakan konstruktor berparameter Anggota ang, Buku buku, String tanggalPinjam, String tanggalKembali |
| public Peminjaman getById(int id) | | mendapatkan objek Peminjaman yang ada di database berdasarkan id-nya |
| public ArrayList<Peminjaman> getAll() | | mendapatkan semua data Peminjaman yang ada di database, dan ditampung ke ArrayList<Peminjaman>. |
| public void save() | | memiliki dua fungsi, yaitu insert dan update. Jika data yang diinputkan belum ada (idkategori = 0) maka akan otomatis insert. Jika data yang diinputkan sudah ada, maka otomatis update. |
| public void cariAnggota(int id) | | Untuk melakukan pencarian data dengan id anggota |
| public void cariBuku(int id) | | Untuk melakukan pencarian data dengan id judul |
| public ArrayList<Peminjaman> search(String keyword) | | melakukan pencarian data. Method ini mirip dengan method getAll() namun querynya berbeda |
| public void delete() | | melakukan operasi penghapusan pada tabel peminjaman pada database. |
| Nama Class : FrmPeminjaman | | |
| <pre> package Forntend; import Backend.*; import java.util.ArrayList; import javax.swing.table.DefaultTableModel; /** * * @author IQBAL */ </pre> | | |

```

public class FrmPeminjaman extends javax.swing.JFrame {

    /**
     * Creates new form FrmPeminjaman
     */
    public FrmPeminjaman() {
        initComponents();
        tampilkanData();
        kosongkanForm();
    }

    public void kosongkanForm() {
        this.id.setText("0");
        this.cariAnggota.setText("");
        this.cariBuku.setText("");
        this.tglPinjam.setText("");
        this.tglKembali.setText("");
    }

    public void tampilkanData() {
        String[] kolom = {"ID", "Peminjam", "Judul Buku", "Tanggal Pinjam", "Tanggal
Kembali"};
        ArrayList<Peminjaman> list = new Peminjaman().getAll();
        Object rowData[] = new Object[5];

        this.tabelData.setModel(new DefaultTableModel(new Object[][]{ }, kolom));
        for (int i = 0; i < list.size(); i++) {
            rowData[0] = list.get(i).getIdpeminjaman();
            rowData[1] = list.get(i).getAng().getNama();
            rowData[2] = list.get(i).getBuku().getJudul();
            rowData[3] = list.get(i).getTanggalPinjam();
            rowData[4] = list.get(i).getTanggalKembali();

            ((DefaultTableModel) this.tabelData.getModel()).addRow(rowData);
        }
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

```



```

btnSimpan = new javax.swing.JButton();
cariAnggota = new javax.swing.JTextField();
btnTmbhBaru = new javax.swing.JButton();
cariBuku = new javax.swing.JTextField();
btnHapus = new javax.swing.JButton();
tglPinjam = new javax.swing.JTextField();
jLabel8 = new javax.swing.JLabel();
tglKembali = new javax.swing.JTextField();
jLabel9 = new javax.swing.JLabel();
btnCariAnggota = new javax.swing.JButton();
jLabel1 = new javax.swing.JLabel();
btnCariBuku = new javax.swing.JButton();
jLabel2 = new javax.swing.JLabel();
cariAng = new javax.swing.JLabel();
jLabel3 = new javax.swing.JLabel();
cariBu = new javax.swing.JLabel();
jLabel4 = new javax.swing.JLabel();
jScrollPane1 = new javax.swing.JScrollPane();
tabelData = new javax.swing.JTable();
jLabel5 = new javax.swing.JLabel();
id = new javax.swing.JTextField();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

btnSimpan.setText("Simpan");
btnSimpan.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnSimpanActionPerformed(evt);
    }
});

cariAnggota.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cariAnggotaActionPerformed(evt);
    }
});

btnTmbhBaru.setText("Tambah Baru");
btnTmbhBaru.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnTmbhBaruActionPerformed(evt);
    }
});

btnHapus.setText("Hapus");
btnHapus.addActionListener(new java.awt.event.ActionListener() {

```

```

        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnHapusActionPerformed(evt);
        }
    });

    jLabel8.setText("Format : YYYY/MM/DD");

    jLabel9.setText("Format : YYYY/MM/DD");

    btnCariAnggota.setText("Cari");
    btnCariAnggota.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnCariAnggotaActionPerformed(evt);
        }
    });

    jLabel1.setText("ID");

    btnCariBuku.setText("Cari");
    btnCariBuku.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnCariBukuActionPerformed(evt);
        }
    });

    jLabel2.setText("ID Anggota");

    cariAng.setText("Nama Anggota");

    jLabel3.setText("ID Buku");

    cariBu.setText("Judul Buku");

    jLabel4.setText("Tanggal Pinjam");

    tabelData.setModel(new javax.swing.table.DefaultTableModel(
        new Object [][] {
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null}
        },
        new String [] {
            "Title 1", "Title 2", "Title 3", "Title 4"
        }
    ));

```

```

        tabelData.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                tabelDataMouseClicked(evt);
            }
        });
        jScrollPane1.setViewportViewView(tabelData);

        jLabel5.setText("Tanggal Kembali");

        id.setEditable(false);

        javax.swing.GroupLayout layout = new
        javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .add(jScrollPane1)
                        .addGroup(layout.createSequentialGroup()
                            .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                .add(layout.createSequentialGroup()
                                    .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                        .add(btnTmbhBaru)
                                        .add(btnHapus))
                                    .addComponent(btnSimpan)
                                )
                                .add(layout.createSequentialGroup()
                                    .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                        .add(layout.createSequentialGroup()
                                            .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                                .add(jLabel1)
                                                .add(jLabel2)
                                                .add(jLabel3)
                                                .add(jLabel4)
                                                .add(jLabel5))
                                            .addGap(21, 21, 21)
                                            .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                                                .add(layout.createSequentialGroup()
                                                    .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                                                        .add(layout.createSequentialGroup()
                                                            .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                                                                .add(layout.createSequentialGroup()
                                                                    .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                                                                        .add(layout.createSequentialGroup()
                                                                            .addComponent(id,
                                    javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addComponent(cariBuku,
javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(cariAnggota,
javax.swing.GroupLayout.Alignment.LEADING))
        .addGap(18, 18, 18)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
        .addComponent(btnCariBuku,
javax.swing.GroupLayout.Alignment.TRAILING)
        .addComponent(btnCariAnggota,
javax.swing.GroupLayout.Alignment.TRAILING)))
        .addComponent(tglPinjam,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tglKembali,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(29, 29, 29)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(cariBu)
        .addComponent(cariAng)
        .addComponent(jLabel8))
        .addComponent(jLabel9,
javax.swing.GroupLayout.Alignment.TRAILING))))
        .addGap(0, 139, Short.MAX_VALUE)))
        .addContainerGap()
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
        .addGap(20, 20, 20)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
        .addComponent(jLabel1)
        .addComponent(id, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
        .addComponent(jLabel2)

```

```

        .addComponent(cariAnggota,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(btnCariAnggota)
        .addComponent(cariAng))
    .addGap(18, 18, 18)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
        .addComponent(jLabel3)
        .addComponent(cariBuku, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(btnCariBuku)
        .addComponent(cariBu))
    .addGap(18, 18, 18)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
        .addComponent(jLabel4)
        .addComponent(tglPinjam,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel8))
    .addGap(18, 18, 18)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
        .addComponent(jLabel5)
        .addComponent(tglKembali,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel9))
    .addGap(28, 28, 28)
    .addComponent(btnSimpan)
    .addGap(18, 18, 18)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
        .addComponent(btnTmbhBaru)
        .addComponent(btnHapus))
    .addGap(18, 18, 18)
    .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))

```

109,

```

    );

    pack();
} // </editor-fold>

private void btnSimpanActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Peminjaman pem = new Peminjaman().getById(Integer.parseInt(this.id.getText()));
    pem.setAnggota(new Anggota().getById(Integer.parseInt(this.cariAnggota.getText())));
    pem.setBuku(new Buku().getById(Integer.parseInt(this.cariBuku.getText())));
    pem.setTanggalPinjam(this.tglPinjam.getText());
    pem.setTanggalKembali(this.tglKembali.getText());
    pem.save();
    tampilkanData();
}

private void cariAnggotaActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void btnTmbhBaruActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    kosongkanForm();
}

private void btnHapusActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    DefaultTableModel model = (DefaultTableModel) this.tabelData.getModel();
    int row = this.tabelData.getSelectedRow();

    Peminjaman pem = new Peminjaman().getById(Integer.parseInt(model.getValueAt(row, 0).toString()));
    pem.delete();
    kosongkanForm();
    tampilkanData();
}

private void btnCariAnggotaActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Peminjaman pem = new Peminjaman();
    pem.cariAnggota(Integer.parseInt(this.cariAnggota.getText()));
    this.cariAng.setText((pem.getAng().getNama()));
}

```

```

private void btnCariBukuActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Peminjaman pem = new Peminjaman();
    pem.cariBuku(Integer.parseInt(this.cariBuku.getText()));
    this.cariBu.setText((pem.getBuku().getJudul()));
}

private void tabelDataMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    DefaultTableModel model = (DefaultTableModel) this.tabelData.getModel();
    int row = this.tabelData.getSelectedRow();

    Peminjaman pem = new
Peminjaman().getById(Integer.parseInt(model.getValueAt(row, 0).toString()));

    this.id.setText(String.valueOf(pem.getIdpeminjaman()));
    this.cariAnggota.setText(String.valueOf(pem.getIdanggota()));
    this.cariBuku.setText(String.valueOf(pem.getIdbuku()));
    this.tglPinjam.setText(String.valueOf(pem.getTanggalPinjam()));
    this.tglKembali.setText(String.valueOf(pem.getTanggalKembali()));
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional)
">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look
and feel.
    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(FrmPeminjaman.class.getName()).log(java.util.log
ging.Level.SEVERE, null, ex);

```

```

        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(FrmPeminjaman.class.getName()).log(java.util.log
ging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(FrmPeminjaman.class.getName()).log(java.util.log
ging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(FrmPeminjaman.class.getName()).log(java.util.log
ging.Level.SEVERE, null, ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new FrmPeminjaman().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton btnCariAnggota;
private javax.swing.JButton btnCariBuku;
private javax.swing.JButton btnHapus;
private javax.swing.JButton btnSimpan;
private javax.swing.JButton btnTmbhBaru;
private javax.swing.JLabel cariAng;
private javax.swing.JTextField cariAnggota;
private javax.swing.JLabel cariBu;
private javax.swing.JTextField cariBuku;
private javax.swing.JTextField id;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable tabelData;
private javax.swing.JTextField tglKembali;
private javax.swing.JTextField tglPinjam;
// End of variables declaration

```


| | | |
|--|--|---|
| } | | |
| Public FrmPeminjaman() | | Memanggil method () tampilkanData(), kosongkanForm, agar ketika form ditampilkan pertama kali, maka form isian akan kosong dan data buku langsung ditampilkan |
| Public void kosongkanForm() | | Mengosongkan isian textbox pada form |
| Public void tampilkanData() | | mengambil semua data buku dari database dan menampilkannya ke JTable tblPeminjaman |
| public Peminjaman cari(String keyword) | | melakukan pencarian berdasarkan keyword tertentu. |
| private void btnCariNamaAnggActionPerformed (java.awt.event.ActionEvent evt) | | melakukan pencarian terhadap keyword yang dimasukkan pada txtIdAnggota |
| private void btnSimpanActionPerformed (java.awt.event.ActionEvent evt) | | untuk menambahkan kode untuk menyimpan data. Aksi menyimpan ini secara otomatis menentukan apakah insert atau update, karena pada method save() dari objek pem, sudah dicek apakah data baru atau sudah ada |
| private void btnTambahBaruActionPerformed | | untuk mengosongkan form sehingga dapat digunakan untuk menginputkan data baru |
| private void btnHapusActionPerformed | | untuk menghapus data |
| private void tblPeminjamanMouseClicked | | Agar kita dapat memilih data yang ada pada tblPeminjaman, agar dapat ditampilkan di textfield untuk diedit atau dihapus |
| private void btnCariJudulBukuActionPerformed | | melakukan pencarian terhadap keyword yang dimasukkan pada txtIdJudul |
| Nama Class : TestBackendPeminjaman | | |
| package Forntend; | | |
| import Backend.*; | | |
| /** | | |
| * | | |
| * @author IQBAL | | |
| */ | | |
| public class TestBackendPeminjaman { | | |
| public static void main(String[] args) { | | |
| Kategori novel = new Kategori().getById(27); | | |
| Kategori referensi = new Kategori().getById(28); | | |

```

    Buku buku1 = new Buku(novel, "Ayat-ayat Cinta", " Bangku Kosong", "Laskar
Pelangi");
    Buku buku2 = new Buku(referensi, "Metode Linier", "Springer",
        "Alex Baldwin");
    Buku buku3 = new Buku(novel, "Bintang Terang", "Erlangga", "Mat Sewoot");
    Anggota anggota1 = new Anggota("Iqbal", "Mojokerto", "0982363624");
    Anggota anggota2 = new Anggota("Fanani", "Qurtil", "007871231");
    Anggota anggota3 = new Anggota("Agung", "Aceh", "0865152112");

    Peminjaman peminjaman1 = new Peminjaman(anggota1, buku1, "2019/11/27",
"2019/11/29");
    Peminjaman peminjaman2 = new Peminjaman(anggota2, buku2, "2019/11/27",
"2019/11/29");
    Peminjaman peminjaman3 = new Peminjaman(anggota3, buku3, "2019/11/27",
"2019/11/29");
// test insert

    novel.save();
    referensi.save();

    buku1.save();
    buku2.save();
    buku3.save();

    anggota1.save();
    anggota2.save();
    anggota3.save();


    peminjaman1.save();
    peminjaman2.save();
    peminjaman3.save();
// test select all
    for (Peminjaman b : new Peminjaman().getAll()) {
        System.out.println("Id Anggota : " + b.getAng().getIdanggota() + ", idbuku: " +
b.getBuku().getIdbuku() + ", Tgl Pinjam: " + b.getTanggalPinjam() + ", Tgl Kembali: "
+ b.getTanggalKembali());
    }
}
}
//      // test search
//      for (Peminjaman b : new Peminjaman().search("40")) {

```

Petunjuk: tempatkan kode program di dalam kotak yang telah disediakan dan berikan penjelasan tentang kegunaan dari method-method yang telah dibuat, jika class lebih dari satu, buat dalam kotak yang berbeda (kotak dapat digandakan).

HASIL (15 points)

Petunjuk: tampilkan screenshot hasil program pada bagian ini.


— □ ×

ID

ID Anggota Fanani

ID Buku Metode Linier

Tanggal Pinjam Format : YYYY/MM/DD

Tanggal Kembali Format : YYYY/MM/DD

| ID | Peminjam | Judul Buku | Tanggal Pinjam | Tanggal Kembali |
|----|----------|-----------------|----------------|-----------------|
| 1 | Iqbal | Ayat-ayat Cinta | 2019-11-27 | 2019-11-29 |
| 2 | Fanani | Metode Linier | 2019-11-27 | 2019-11-29 |
| 3 | Agung | Bintang Terang | 2019-11-27 | 2019-11-29 |
| 4 | Iqbal | Ayat-ayat Cinta | 2019-11-27 | 2019-11-29 |
| 5 | Fanani | Metode Linier | 2019-11-27 | 2019-11-29 |

ID

ID Anggota Fanani

ID Buku Metode Linier

Tanggal Pinjam Format : YYYY/MM/DD

Tanggal Kembali Format : YYYY/MM/DD

| ID | Peminjam | Judul Buku | Tanggal Pinjam | Tanggal Kembali |
|----|----------|-----------------|----------------|-----------------|
| 3 | Agung | Bintang Terang | 2019-11-27 | 2019-11-29 |
| 4 | Iqbal | Ayat-ayat Cinta | 2019-11-27 | 2019-11-29 |
| 5 | Fanani | Metode Linier | 2019-11-27 | 2019-11-29 |
| 6 | Agung | Bintang Terang | 2019-11-27 | 2019-11-29 |
| 7 | Fanani | Metode Linier | 2019-11-29 | 2019-11-30 |

KESIMPULAN (15 points)

Tips: tuliskan kesimpulan dari hasil tugas praktikum yang telah dilakukan.

Paradigma berorientasi objek untuk interaksi dengan database yang telah dipelajari dengan cara membuat aplikasi berbasis database dan dilengkapi dengan form sebagai Graphical User Interface (GUI). Memiliki tahapan sebagai berikut:

1. Membuat database dan tabel-tabelnya.
2. Membuat backend yang berisi class-class yang mewakili data yang ada pada database, dan class helper untuk melakukan eksekusi query database.
3. Membuat frontend yang merupakan antarmuka kepada pengguna. Frontend ini bisa berbasis teks (console), GUI, web, mobile, dan sebagainya.

Library yang digunakan untuk praktikum ini antara lain:

1. JDBC, untuk melakukan interaksi ke database.
2. ArrayList, untuk menampung data hasil query ke database.
3. Swing, untuk membuat tampilan GUI.

Di dalam percobaan ini terdapat class untuk mengeksekusi query SQL yang di dalamnya ada method-method antara lain:

- a) bukaKoneksi(), untuk membuka koneksi ke database
- b) insertQueryGetId(String query), untuk melakukan insert ke tabel dan mengembalikan nilai ID yang digenerate oleh database (hasil Auto Increment).
- c) executeQuery(String query), untuk mengeksekusi query yang tidak mengembalikan nilai (misal: insert, update, delete).
- d) selectQuery(String query), untuk mengeksekusi select query yang mengembalikan nilai hasil query.

Dari Percobaan ini dapat mengetahui pengertian front end dan back end, dimana Front End Developer atau Pengembang front-end bertanggung jawab untuk menyesuaikan tampilan, nuansa, dan perilaku situs web. Antarmuka pengguna mencakup semuanya mulai dari ukuran dan warna font, formulir, hingga menu dan bentuk dropdown. Sedangkan Back End Developer atau pengembang web bagian belakang, lebih ke database