

MANUAL BOOK
APLIKASI CRUD (CREATE, RESET, UPDATE, DELETE)
PRAKTIKUM REKAYASA PERANGKAT LUNAK 2



Disusun oleh:

1. Fira Febriyanti (50420498)
2. Ibrahim Bramullah (50420562)
3. Shafa Nurul Umayya (51420188)

TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS GUNADARMA
DEPOK
2023

DAFTAR ISI

DAFTAR ISI	2
BAB I	3
PENDAHULUAN	3
1.1 Latar Belakang	3
1.2 Tujuan	3
BAB II	4
PEMBAHASAN	4
2.1 Perangkat lunak	4
2.1.1 CRUD Operations	4
2.1.2 Karakteristik Perangkat Lunak	5
Perangkat lunak memiliki karakteristik khusus yang relevan untuk implementasi CRUD:	5
2.2 Software Pendukung	5
2.2.1 Phpmyadmin	5
2.2.2 XAMPP	6
2.2.3 Aspect Oriented Programming (AOP)	6
2.2.4 Spring Framework	7
BAB III	8
ANALISA DAN PERANCANGAN	8
3.1 Tahapan Pembangunan Sistem	8
3.1.1 Implementasi bagian-bagian project	8
3.1.1.1 Add Library dan Add RAR	8
3.1.1.2 Java package springHibernate	8
3.1.1.3 Java package springHibernate.configuration	9
3.1.1.4 Java package springHibernate.controller	10
3.1.1.5 Java package springHibernate.dao	14
3.1.1.6 Java package springHibernate.model	15
3.1.1.7 Java package springHibernate.service	16
3.1.1.8 Java package springHibernate.view	18
3.1.1.9 <default package>	26
3.1.2 Implementasi CRUD Operations	29
3.1.3 Database	33
3.1.3.1 Xampp	33
3.1.3.2 MySQL (phpmyadmin)	33
3.1.4 Hasil Output	34
3.1.4.1 Tampilan awal	34
3.1.4.2 Save	34
3.1.4.3 Ubah	35
3.1.4.4 Delete	35
3.1.4.5 DialogMessage (Error)	36
BAB IV	37
PENUTUP	37
4.1 Kesimpulan	37
4.2 Saran	37

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kemajuan teknologi kian semakin maju, banyak perusahaan atau bahkan usaha mikro dan menengah (UMKM) menggantungkan usahanya pada teknologi. Dengan memanfaatkan perangkat lunak, perusahaan atau usaha mikro dan menengah (UMKM) dapat mengurangi pekerjaan yang sebelumnya menggunakan sistem manual atau bahkan tulis tangan. Sistem manual memang dinilai dapat lebih akurat yang artinya kecil kemungkinan adanya kecurangan, tetapi untuk perusahaan maju dan berkembang haruslah menggunakan sistem yang terotomatisasi karena tidak memungkinkan jika menggunakan sistem manual secara terus-menerus, karena suatu data pasti bertambah banyak seiring dengan waktunya.

Perangkat Lunak saat ini muncul karena adanya kebutuhan pasar serta masyarakat yang semakin tinggi dibidang teknologi. Perangkat lunak dapat digunakan sebagai wadah untuk memproses data, memerintah atau menginstruksi khusus agar pengguna dapat mengoperasikan komputer sesuai dengan hasil informasi yang diinginkan.

Tampilan akhir pada perangkat lunak yang telah diproses dapat digunakan oleh perusahaan atau usaha mikro dan menengah (UMKM). tampilan akhir tersebut akan menjadi informasi yang dapat berbentuk teks, grafik, gambar, dan lain-lain.

1.2 Tujuan

Tujuan dari penulisan ini adalah merancang sebuah perangkat lunak sederhana yang mengandung CRUD (Create, Reset, Update, Delete) pada daftar Karyawan yang disimpan dalam database.

BAB II

PEMBAHASAN

2.1 Perangkat lunak

RPL merupakan sebuah disiplin ilmu yang berhubungan dengan seluruh aspek produk perangkat lunak baik dari tahapan awal hingga ke pemeliharaan dari perangkat lunak pasca produksi atau proses yang terintegrasi dan menyeluruh dari segala aspek, mulai dari sebelum perangkat lunak itu dibuat hingga selesai dan bahkan hingga tahap segala aspek. (Rizky Wicaksono, 2017)

Karakteristik Perangkat Lunak lebih dikenal sebagai elemen logik dari pada fisik, oleh karena itu perangkat lunak memiliki karakteristik yang berbeda dari perangkat keras :

- Perangkat lunak dikembangkan atau direayasa, jadi tidak diproduksi dalam pengertian klasik.
- Merupakan produk yang unik (tidak ada seri produksi).
- Perangkat lunak tidak pernah akan rusak karena selalu diperbaharui.
- Perangkat lunak pada umumnya dibangun sesuai keinginan, jadi tidak dibentuk dari komponen yang sudah ada.
- Fleksibel, sehingga mudah dimodifikasi.
- Dihubungkan (linked) dengan sistem komputer.

2.1.1 CRUD Operations

Operasi CRUD (Create, Read, Update, Delete) adalah prinsip dasar dalam pengelolaan data. Mari kita bahas setiap operasi:

- **Create (Buat)**

Ini melibatkan pembuatan data baru dalam sistem. Dalam konteks aplikasi ini, ini mungkin terkait dengan menambahkan entri baru ke dalam daftar karyawan.

- **Read (Baca)**

Ini melibatkan membaca atau menampilkan data yang sudah ada. Dalam aplikasi CRUD ini, ini mungkin terkait dengan menampilkan daftar karyawan dari database.

- **Update (Perbarui)**

Ini melibatkan modifikasi data yang sudah ada. Dalam konteks aplikasi ini, ini dapat mencakup memperbarui informasi karyawan seperti alamat, nomor telepon, dll.

- Delete (Hapus)

Ini melibatkan penghapusan data dari sistem. Dalam aplikasi ini, ini mungkin terkait dengan menghapus entri karyawan dari daftar.

2.1.2 Karakteristik Perangkat Lunak

Perangkat lunak memiliki karakteristik khusus yang relevan untuk implementasi CRUD:

- Dikembangkan atau Direkayasa

Proses pengembangan perangkat lunak melibatkan rekayasa perangkat lunak yang mencakup perencanaan, desain, pengkodean, pengujian, dan pemeliharaan.

- Produk Unik

Setiap perangkat lunak adalah produk unik yang dibangun sesuai kebutuhan spesifik.

- Fleksibilitas

Perangkat lunak harus mudah dimodifikasi, mencakup perubahan dalam fitur atau pembaruan data.

2.2 Software Pendukung

2.2.1 Phpmyadmin

Phpmyadmin adalah sebuah aplikasi open source yang berfungsi untuk memudahkan manajemen MySQL. Dengan menggunakan phpmyadmin, anda dapat membuat database, membuat tabel, menginsert, menghapus dan mengupdate data dengan GUI dan terasa lebih mudah, tanpa perlu mengetikkan perintah SQL secara manual.

PHPMyAdmin merupakan front-end MySQL berbasis web. PHPMyAdmin dibuat dengan menggunakan PHP. Saat ini, PHPMyAdmin banyak digunakan dalam hampir semua penyedia hosting yang ada di internet. PHPMyAdmin

mendukung berbagai fitur administrasi MySQL termasuk manipulasi database, tabel, index dan juga dapat mengekspor data ke dalam berbagai format data. PHPMyAdmin juga tersedia dalam 50 bahasa lebih, termasuk bahasa Indonesia.

2.2.2 XAMPP

XAMPP adalah software gratis dan open source yang dapat diinstal di berbagai platform, seperti Windows, Linux dan OS X. Pasalnya, di dalam aplikasi XAMPP terdapat puluhan paket modul, bahasa pemrograman dan komponen lainnya. Ini akan dibahas lebih detail di bagian fitur XAMPP. XAMPP mewakili lima komponen utama, yaitu:

- X – Cross Platform. X adalah singkatan dari XAMPP yang merupakan singkatan dari Cross. Ini merupakan indikasi bahwa XAMPP adalah aplikasi lintas platform yang kompatibel dengan berbagai jenis sistem operasi.
- A – Apache. Apache adalah server web default yang digunakan oleh XAMPP. Server web adalah perangkat lunak yang mengelola transmisi data antara website dan pengunjugnya. Selain itu, Apache juga berperan sebagai distributor fitur atau komponen lainnya.
- M – MariaDB. MariaDB adalah perangkat lunak DBMS default XAMPP. DBMS adalah perangkat lunak untuk mengelola database dan data yang berisi mereka. MariaDB menggantikan perangkat lunak MySQL DBMS sebelumnya pada XAMPP 5.5.30 dan 5.6.14.
- P – PHP. PHP adalah bahasa pemrograman untuk membuat website atau aplikasi web dari back-end.
- P – Perl. Kepanjangan yang terakhir yaitu Perl. Perl adalah contoh bahasa pemrograman untuk tujuan yang lebih kompleks. Misalnya membuat aplikasi GUI, pemrograman jaringan dan masih banyak lagi.

2.2.3 Aspect Oriented Programming (AOP)

Aspect Oriented Programming (AOP) merupakan paradigma yang memiliki target meningkatkan modularitas aplikasi dengan menggunakan pemisahan *cross-cutting concern* (aspek).

2.2.4 Spring Framework

Spring *framework* adalah sebuah *framework* atau kerangka kerja yang bersifat *open-source*.

Beberapa keunggulan yang dimiliki Spring adalah:

- Mudah dipelajari
- Tidak terlalu invasif
- Helper class yang banyak
- Mendukung AOP
- Menyediakan integrasi dengan framework lain

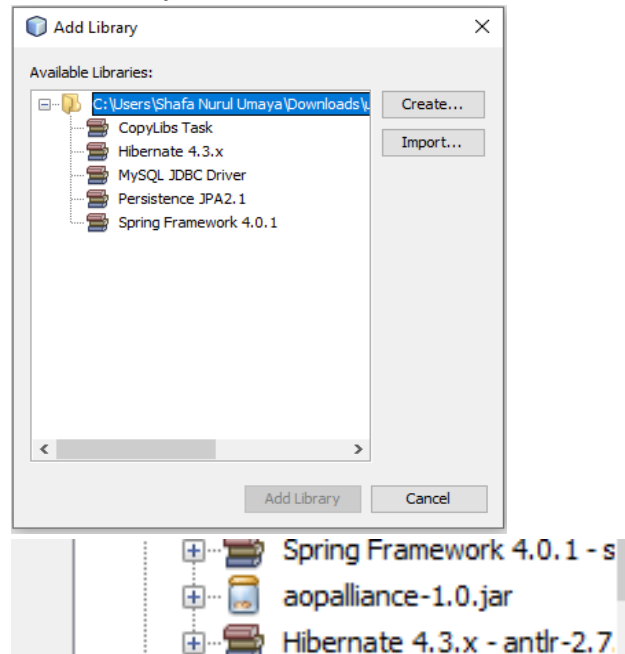
BAB III

ANALISA DAN PERANCANGAN

3.1 Tahapan Pembangunan Sistem

3.1.1 Implementasi bagian-bagian project

3.1.1.1 Add Library dan Add RAR



3.1.1.2 Java package springHibernate

Class: **App.java**

```
package springHibernate;

import springHibernate.service.KaryawanService;
import springHibernate.view.KaryawanView;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
/**
 *
 * @author ibrah
 */
public class App {
    private static ApplicationContext applicationContext;

    public static void main(String[] args) {
```



```

        /*konfigurasi spring*/
        applicationContext = new
ClassPathXmlApplicationContext("classpath:spring-configuration.xml");
        new KaryawanView().setVisible(true);
    }

    public static KaryawanService getKaryawanService() {
        return (KaryawanService)
applicationContext.getBean("KaryawanService");
    }
}

```

3.1.1.3 Java package springHibernate.configuration

Class: **KaryawanTableModel.java**

```

package springHibernate.configuration;

import springHibernate.model.Karyawan;
import java.util.ArrayList;
import java.util.List;
import javax.swing.table.AbstractTableModel;

public class KaryawanTableModel extends AbstractTableModel {

    private List<Karyawan> karyawans = new ArrayList<>();
    private final String HEADER[] = {"Id", "Nama", "JenisKelamin",
"HariKerja"};

    public KaryawanTableModel (List<Karyawan> karyawans) {
        this.karyawans = karyawans;
    }

    @Override
    public int getRowCount() {
        return karyawans.size();
    }

    @Override
    public int getColumnCount() {
        return HEADER.length;
    }
}

```

```

@Override
public String getColumnName(int columnIndex) {
    return HEADER[columnIndex];
}

@Override
public Object getValueAt(int rowIndex, int columnIndex) {
    Karyawan karyawan = karyawanans.get(rowIndex);

    switch (columnIndex) {
        case 0:
            return karyawan.getId();
        case 1:
            return karyawan.getNama();
        case 2:
            return karyawan.getJenisKelamin();
        case 3:
            return karyawan.getHariKerja();
        default:
            return null;
    }
}
}
}

```

3.1.1.4 Java package springHibernate.controller

Class: **KaryawanController.java**

```

package springHibernate.controller;

import springHibernate.App;
import springHibernate.configuration.KaryawanTableModel;
import springHibernate.model.Karyawan;
import springHibernate.view.KaryawanView;
import java.util.List;
import javax.swing.JOptionPane;

public class KaryawanController {
    private final KaryawanView karyawanView;
    private KaryawanTableModel karyawanTableModel;
    private List<Karyawan> karyawanans;

    public KaryawanController(KaryawanView karyawanView) {

```

```

        this.karyawanView = karyawanView;
    }

    public void tampilData() {
        karyawans = App.getKaryawanService().getKaryawans();
        karyawanTableModel = new KaryawanTableModel(karyawans);
        this.karyawanView.getTabel().setModel(karyawanTableModel);
    }

    public void show() {
        int index = this.karyawanView.getTabel().getSelectedRow();

        this.karyawanView.getId().setText(String.valueOf(this.karyawanView.getTabel().getValueAt(index, 0)));

        this.karyawanView.getNama().setText(String.valueOf(this.karyawanView.getTabel().getValueAt(index, 1)));
        //
        this.karyawanView.getJenisKelamin().setText(String.valueOf(this.karyawanView.getTabel().getValueAt(index, 2)));

        this.karyawanView.getJenisKelaminComboBox().setSelectedItem(String.valueOf(this.karyawanView.getTabel().getValueAt(index, 2)));
        //
        this.karyawanView.getHariKerja().setText(String.valueOf(this.karyawanView.getTabel().getValueAt(index, 3)));

        this.karyawanView.getHarikerjaComboBox().setSelectedItem(String.valueOf(this.karyawanView.getTabel().getValueAt(index, 3)));
    }

    public void clear() {
        this.karyawanView.getId().setText("");
        this.karyawanView.getNama().setText("");
        // this.karyawanView.getJenisKelamin().setText("");
        this.karyawanView.getJenisKelaminComboBox().setSelectedIndex(0); //
Reset the combo box selection
        // this.karyawanView.getHariKerja().setText("");
        this.karyawanView.getHarikerjaComboBox().setSelectedIndex(0); //
Reset the combo box selection
    }

    public void saveKaryawan() {
        if (this.karyawanView.getJenisKelaminComboBox().getSelectedIndex()
== 0) {
            // "Select Date" is selected, show a message dialog

```

```

        JOptionPane.showMessageDialog(null, "Pilih Jenis Kelamin", "Error",
JOptionPane.ERROR_MESSAGE);
        return; // Stop further execution
    }

    if (this.karyawanView.getHarikerjaComboBox().getSelectedIndex() ==
0) {
        // "Select Date" is selected, show a message dialog
        JOptionPane.showMessageDialog(null, "Pilih hari kerja terlebih
dahulu", "Error", JOptionPane.ERROR_MESSAGE);
        return; // Stop further execution
    }

    Karyawan karyawan = new Karyawan();
    karyawan.setId(this.karyawanView.getId().getText());
    karyawan.setNama(this.karyawanView.getNama().getText());

    //
    karyawan.setJenisKelamin(this.karyawanView.getJenisKelamin().getText());
    karyawan.setJenisKelamin((String)
this.karyawanView.getJenisKelaminComboBox().getSelectedItem());

    //
    karyawan.setHariKerja(this.karyawanView.getHariKerja().getText());
    karyawan.setHariKerja((String)
this.karyawanView.getHarikerjaComboBox().getSelectedItem());

    App.getKaryawanService().save(karyawan);
    JOptionPane.showMessageDialog(null, "Data Berhasil di Simpan",
"Info", JOptionPane.INFORMATION_MESSAGE);
    clear();
    tampilData();
}

public void updateKaryawan() {
    if (this.karyawanView.getJenisKelaminComboBox().getSelectedIndex()
== 0) {
        // "Select Date" is selected, show a message dialog
        JOptionPane.showMessageDialog(null, "Pilih Jenis Kelamin", "Error",
JOptionPane.ERROR_MESSAGE);
        return; // Stop further execution
    }

    if (this.karyawanView.getHarikerjaComboBox().getSelectedIndex() ==
0) {
        // "Select Date" is selected, show a message dialog

```

```

        JOptionPane.showMessageDialog(null, "Pilih hari kerja terlebih
dahulu", "Error", JOptionPane.ERROR_MESSAGE);
        return; // Stop further execution
    }

    Karyawan karyawan = new Karyawan();
    karyawan.setId(this.karyawanView.getId().getText());
    karyawan.setNama(this.karyawanView.getNama().getText());

    //
    karyawan.setJenisKelamin(this.karyawanView.getJenisKelamin().getText());
    karyawan.setJenisKelamin((String)
this.karyawanView.getJenisKelaminComboBox().getSelectedItem());

    karyawan.setHariKerja((String)
this.karyawanView.getHarikerjaComboBox().getSelectedItem());

    App.getKaryawanService().update(karyawan);
    JOptionPane.showMessageDialog(null, "Data Berhasil di Ubah",
"Info", JOptionPane.INFORMATION_MESSAGE);
    clear();
    tampilData();
}

public void deleteKaryawan() {
    if (this.karyawanView.getId().getText().isEmpty()) {
        JOptionPane.showMessageDialog(null, "Data Karyawan Kosong / Belum
Dipilih", "Error", JOptionPane.ERROR_MESSAGE);
    } else {
        Karyawan karyawan = new Karyawan();
        karyawan.setId(this.karyawanView.getId().getText());
        int pilih = JOptionPane.showConfirmDialog(null, "Apakah data ingin
dihapus ?", "Warning", JOptionPane.YES_NO_OPTION,
JOptionPane.WARNING_MESSAGE);
        if (pilih == JOptionPane.YES_OPTION) {
            App.getKaryawanService().delete(karyawan);
            JOptionPane.showMessageDialog(null, "Data Berhasil di Hapus",
"Info", JOptionPane.INFORMATION_MESSAGE);
            clear();
            tampilData();
        }
    }
}
}
}

```

3.1.1.5 Java package springHibernate.dao

Interface: KaryawanDao.java

```
package springHibernate.dao;

import springHibernate.model.Karyawan;
import java.util.List;
/**
 *
 * @author ibrah
 */
public interface KaryawanDao {
    public void save(Karyawan karyawan); /*perintah save*/
    public void update(Karyawan karyawan); /*perintah update*/
    public void delete(Karyawan karyawan); /*perintah delete*/
    public Karyawan getKaryawan(String id); /*getter karyawan*/
    public List<Karyawan> getKaryawans(); /*getter semua data karyawan yang
ada di databse*/
}
```

Class: KaryawanDaoImpl.java

```
package springHibernate.dao;

import springHibernate.model.Karyawan;
import java.util.List;
import org.hibernate.SessionFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Repository;

@Repository
public class KaryawanDaoImpl implements KaryawanDao {

    @Autowired
    private SessionFactory sessionFactory;

    @Override
    public void save(Karyawan karyawan) {
        sessionFactory.getCurrentSession().save(karyawan);
    }

    @Override
```

```

public void update(Karyawan karyawan) {
    sessionFactory.getCurrentSession().update(karyawan);
}

@Override
public void delete(Karyawan karyawan) {
    sessionFactory.getCurrentSession().delete(karyawan);
}

@Override
public Karyawan getKaryawan(String id) {
    return (Karyawan) sessionFactory.getCurrentSession().get(Karyawan.class, id);
}

@Override
public List<Karyawan> getKaryawans() {
    return sessionFactory.getCurrentSession().createCriteria(Karyawan.class).list();
}
}

```

3.1.1.6 Java package springHibernate.model

Class: **Karyawan.java**

```

package springHibernate.model;

import java.io.Serializable;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name = "tb_karyawan")
public class Karyawan implements Serializable{
    /*memasukkan data untuk data karyawan*/
    @Id
    @Column(name = "id", length = 10)
    private String id;

    @Column(name = "nama", length = 50)
    private String nama;
}

```

```

@Column(name = "jeniskelamin", length = 15)
private String jeniskelamin;

@Column(name = "harikerja", length = 15)
private String harikerja;

/*getter dan setter*/
public String getId() {
    return id;
}

public void setId(String id) {
    this.id = id;
}

public String getName() {
    return nama;
}

public void setName(String nama) {
    this.nama = nama;
}

public String getJenisKelamin() {
    return jeniskelamin;
}

public void setJenisKelamin(String jeniskelamin) {
    this.jeniskelamin = jeniskelamin;
}

public String getHariKerja() {
    return harikerja;
}

public void setHariKerja(String harikerja) {
    this.harikerja = harikerja;
}
}

```

3.1.1.7 Java package springHibernate.service

Interface: **KaryawanService.java**


```

package springHibernate.service;

import springHibernate.model.Karyawan;
import java.util.List;
/**
 *
 * @author ibrah
 */
public interface KaryawanService {
    public void save(Karyawan karyawan);
    public void update(Karyawan karyawan);
    public void delete(Karyawan karyawan);
    public Karyawan getKaryawan(String id);
    public List<Karyawan> getKaryawans();
}

```

Class: KaryawanServiceImpl.java

```

package springHibernate.service;

import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;
import springHibernate.dao.KaryawanDao;
import springHibernate.model.Karyawan;

@Service("KaryawanService") /*yang akan di jalankan springframework*/
@Transactional(readOnly = true) /*langsung terhubung ke database*/
public class KaryawanServiceImpl implements KaryawanService {

    @Autowired
    private KaryawanDao karyawanDao;

    @Transactional
    @Override
    public void save(Karyawan karyawan) {
        karyawanDao.save(karyawan);
    }

    @Transactional
    @Override
    public void update(Karyawan karyawan) {

```

```

        karyawanDao.update(karyawan);
    }

    @Transactional
    @Override
    public void delete(Karyawan karyawan) {
        karyawanDao.delete(karyawan);
    }

    @Transactional
    @Override
    public Karyawan getKaryawan(String id) {
        return karyawanDao.getKaryawan(id);
    }

    @Transactional
    @Override
    public List<Karyawan> getKaryawans() {
        return karyawanDao.getKaryawans();
    }
}

```

3.1.1.8 Java package springHibernate.view

JFrame: KaryawanView.java

```

package springHibernate.view;

import javax.swing.JCheckBox;
import springHibernate.controller.KaryawanController;
import javax.swing.JTable;
import javax.swing.JTextArea;
import javax.swing.JTextField;

/**
 *
 * @author ibrah
 */
public class KaryawanView extends javax.swing.JFrame {
    private final KaryawanController karyawanController =new
    KaryawanController(this);

    /**

```

```

    * Creates new form KaryawanView
    */
public KaryawanView() {
    initComponents();
    karyawanController.tampilData();
}

/**
 * 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() {

    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    id = new javax.swing.JTextField();
    nama = new javax.swing.JTextField();
    simpan = new javax.swing.JButton();
    update = new javax.swing.JButton();
    hapus = new javax.swing.JButton();
    jScrollPane1 = new javax.swing.JScrollPane();
    tabel = new javax.swing.JTable();
    jLabel5 = new javax.swing.JLabel();
    harikerja = new javax.swing.JComboBox<>();
    jeniskelamin = new javax.swing.JComboBox<>();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    jLabel1.setFont(new java.awt.Font("Arial", 1, 20)); // NOI18N
    jLabel1.setText("DAFTAR KARYAWAN");

    jLabel2.setText("Id                :");

    jLabel3.setText("Nama                :");

    jLabel4.setText("Jenis Kelamin      :");

    simpan.setText("Simpan");
    simpan.addActionListener(new java.awt.event.ActionListener() {

```

```

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

    update.setText("Ubah");
    update.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            updateActionPerformed(evt);
        }
    });

    hapus.setText("Hapus");
    hapus.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            hapusActionPerformed(evt);
        }
    });

    tabel.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"
        }
    ));
    tabel.addMouseListener(new java.awt.event.MouseAdapter() {
        public void mouseClicked(java.awt.event.MouseEvent evt) {
            tabelMouseClicked(evt);
        }
    });
    jScrollPane1.setViewportViewView(tabel);

    jLabel5.setText("Hari Kerja      :");

    harikerja.setModel(new javax.swing.DefaultComboBoxModel<>(new
String[] { "Pilih Hari Kerja", "SENIN", "SELASA", "RABU", "KAMIS", "JUMAT",
"SABTU", "MINGGU" }));
    harikerja.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            harikerjaActionPerformed(evt);
        }
    });
}

```

```

        jeniskelamin.setModel(new javax.swing.DefaultComboBoxModel<>(new
String[] { "Pilih Jenis Kelamin", "Laki - laki", "Perempuan" }));

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                .addGap(0, 0, Short.MAX_VALUE)
                .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 207,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(136, 136, 136))
            .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGap(78, 78, 78)
                    .addComponent(simpan,
javax.swing.GroupLayout.PREFERRED_SIZE, 82,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(62, 62, 62)
                    .addComponent(update,
javax.swing.GroupLayout.PREFERRED_SIZE, 82,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(57, 57, 57)
                    .addComponent(hapus,
javax.swing.GroupLayout.PREFERRED_SIZE, 82,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(layout.createSequentialGroup()
                    .addGap(30, 30, 30)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 450,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

```

```

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING))

        .addComponent(jLabel13)
        .addComponent(jLabel12)
        .addComponent(jLabel15))

.addGroup(layout.createSequentialGroup())

        .addComponent(jLabel14)
        .addGap(1, 1, 1))
        .addGap(49, 49, 49)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false))

        .addComponent(harikerja,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.PREFERRED_SIZE, 121,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jeniskelamin,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.PREFERRED_SIZE, 185,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(id,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, 295, Short.MAX_VALUE)
        .addComponent(nama,
javax.swing.GroupLayout.Alignment.LEADING))))))
        .addContainerGap(15, Short.MAX_VALUE))
);
layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup()

            .addContainerGap()
            .addComponent(jLabel11,
javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(17, 17, 17)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(id,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jLabel12))

```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(nama,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel3))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel4)
    .addComponent(jeniskelamin,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel5)
    .addComponent(harikerja,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(33, 33, 33)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(simpan,
        javax.swing.GroupLayout.PREFERRED_SIZE, 32,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(update,
        javax.swing.GroupLayout.PREFERRED_SIZE, 32,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(hapus,
        javax.swing.GroupLayout.PREFERRED_SIZE, 32,
        javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(44, 44, 44)
    .addComponent(jScrollPane1,
        javax.swing.GroupLayout.PREFERRED_SIZE, 355,
        javax.swing.GroupLayout.PREFERRED_SIZE)
```

```

        .addContainerGap(30, Short.MAX_VALUE))

    );

    harikerja.getAccessibleContext().setAccessibleName("");

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

private void updateActionPerformed(java.awt.event.ActionEvent evt) {
    karyawanController.updateKaryawan();
}

private void simpanActionPerformed(java.awt.event.ActionEvent evt) {
    karyawanController.saveKaryawan();
}

private void hapusActionPerformed(java.awt.event.ActionEvent evt) {
    karyawanController.deleteKaryawan();
}

private void tabelMouseClicked(java.awt.event.MouseEvent evt) {
    karyawanController.show();
}

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

public JTable getTabel() {
    return tabel;
}

public JTextField getId() {
    return id;
}

public JTextField getNama() {
    return nama;
}

public javax.swing.JComboBox<String> getHarikerjaComboBox() {
    return harikerja;
}

public javax.swing.JComboBox<String> getJenisKelaminComboBox() {
    return jeniskelamin;
}

```



```

}

/**
 * @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(KaryawanView.class.getName()).log(java.u
til.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(KaryawanView.class.getName()).log(java.u
til.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(KaryawanView.class.getName()).log(java.u
til.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(KaryawanView.class.getName()).log(java.u
til.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>

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

```

```

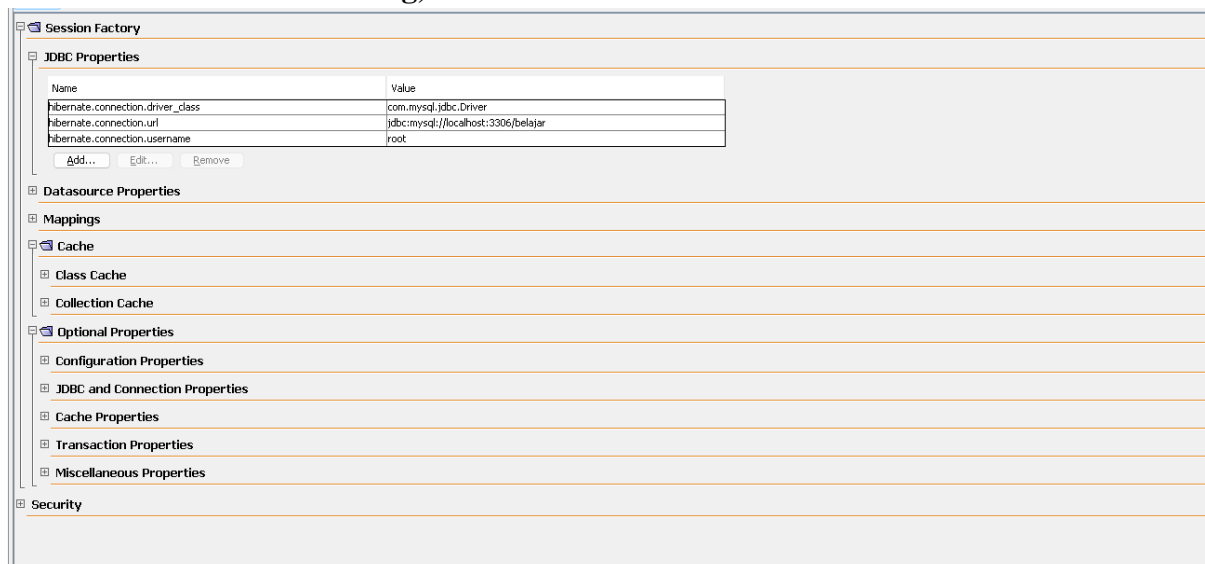
        new KaryawanView().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JButton hapus;
private javax.swing.JComboBox<String> harikerja;
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.JScrollPane jScrollPane1;
private javax.swing.JComboBox<String> jeniskelamin;
private javax.swing.JTextField nama;
private javax.swing.JButton simpan;
private javax.swing.JTable tabel;
private javax.swing.JButton update;
// End of variables declaration
}

```

3.1.1.9 <default package>

XML Document: hibernate.cfg.xml



<hibernate-configuration>

```

    <session-factory>
        <property
name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
        <property
name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
        <property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/belajar</proper
ty>
        <property name="hibernate.connection.username">root</property>

        <property name="hibernate.hbm2ddl.auto">update</property>
        <property name="hibernate.show_sql">true</property>
        <property name="hibernate.format_sql">true</property>
        <mapping class="springHibernate.model.Karyawan" />
    </session-factory>
</hibernate-configuration>

```

Properties: **jdbc.properties**

```

jdbc.driver = com.mysql.jdbc.Driver
jdbc.url = jdbc:mysql://localhost:3306/belajar
jdbc.username = root
jdbc.password =

```

Spring Config: **spring-configuration.xml**

```

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:context="http://www.springframework.org/schema/context"
    xmlns:p="http://www.springframework.org/schema/p"
    xmlns:tx="http://www.springframework.org/schema/tx"

    xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-4.0.xsd
    http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-4.0.xsd
    http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx-4.0.xsd
">
    <context:annotation-config/>

```

```
<tx:annotation-driven transaction-manager="transactionManager"/>

<context:component-scan base-package="springHibernate"/>
<context:property-placeholder location="classpath:jdbc.properties"/>

<bean id="dataSource"
class="org.springframework.jdbc.datasource.DriverManagerDataSource"
    p:driverClassName="${jdbc.driver}"
    p:url="${jdbc.url}"
    p:username="${jdbc.username}"
    p:password="${jdbc.password}"/>

<bean id="sessionFactory"
class="org.springframework.orm.hibernate4.LocalSessionFactoryBean"
    p:dataSource-ref="dataSource"
    p:configLocation="classpath:hibernate.cfg.xml"/>

<bean id="transactionManager"
class="org.springframework.orm.hibernate4.HibernateTransactionManager"
    p:sessionFactory-ref="sessionFactory"/>

</beans>
```

3.1.2 Implementasi CRUD Operations

Setelah memahami konsep CRUD, kita dapat melihat bagaimana operasi-operasi ini diimplementasikan dalam kode. Berikut adalah penjelasan singkat untuk setiap operasi dalam implementasi aplikasi kita.

- Create (Buat) - Penambahan Karyawan Baru:

Class: **KaryawanController.java**

```
public void saveKaryawan() {
    if (this.karyawanView.getJenisKelaminComboBox().getSelectedIndex()
    == 0) {
        // "Select Date" is selected, show a message dialog
        JOptionPane.showMessageDialog(null, "Pilih Jenis Kelamin", "Error",
        JOptionPane.ERROR_MESSAGE);
        return; // Stop further execution
    }

    if (this.karyawanView.getHarikerjaComboBox().getSelectedIndex() ==
    0) {
        // "Select Date" is selected, show a message dialog
        JOptionPane.showMessageDialog(null, "Pilih hari kerja terlebih
        dahulu", "Error", JOptionPane.ERROR_MESSAGE);
        return; // Stop further execution
    }

    Karyawan karyawan = new Karyawan();
    karyawan.setId(this.karyawanView.getId().getText());
    karyawan.setNama(this.karyawanView.getNama().getText());

    //
    karyawan.setJenisKelamin(this.karyawanView.getJenisKelamin().getText());
    karyawan.setJenisKelamin((String)
    this.karyawanView.getJenisKelaminComboBox().getSelectedItem());

    //
    karyawan.setHariKerja(this.karyawanView.getHariKerja().getText());
    karyawan.setHariKerja((String)
    this.karyawanView.getHarikerjaComboBox().getSelectedItem());

    App.getKaryawanService().save(karyawan);
    JOptionPane.showMessageDialog(null, "Data Berhasil di Simpan",
    "Info", JOptionPane.INFORMATION_MESSAGE);
}
```

```
clear();  
tampilData();  
}
```

- Read (Baca) - Menampilkan Daftar Karyawan:

Class: **KaryawanController.java**

```
public void tampilData() {  
    karyawans = App.getKaryawanService().getKaryawans();  
    karyawanTableModel = new KaryawanTableModel(karyawans);  
    this.karyawanView.getTabel().setModel(karyawanTableModel);  
}  
  
public void show() {  
    int index = this.karyawanView.getTabel().getSelectedRow();  
  
    this.karyawanView.getId().setText(String.valueOf(this.karyawanView.getTabel()  
().getValueAt(index, 0)));  
  
    this.karyawanView.getNama().setText(String.valueOf(this.karyawanView.getTab  
el().getValueAt(index, 1)));  
  
    this.karyawanView.getJenisKelaminComboBox().setSelectedItem(String.valueOf(  
this.karyawanView.getTabel().getValueAt(index, 2)));  
  
    this.karyawanView.getHarikerjaComboBox().setSelectedItem(String.valueOf(thi  
s.karyawanView.getTabel().getValueAt(index, 3)));  
}
```

- Update (Perbarui) - Memodifikasi Informasi Karyawan:

Class: **KaryawanController.java**

```
public void updateKaryawan() {  
    if (this.karyawanView.getJenisKelaminComboBox().getSelectedIndex()  
== 0) {  
        // "Select Date" is selected, show a message dialog  
        JOptionPane.showMessageDialog(null, "Pilih Jenis Kelamin", "Error",  
JOptionPane.ERROR_MESSAGE);  
        return; // Stop further execution  
    }  
}
```

```

    }

    if (this.karyawanView.getHarikerjaComboBox().getSelectedIndex() ==
0) {
        // "Select Date" is selected, show a message dialog
        JOptionPane.showMessageDialog(null, "Pilih hari kerja terlebih
dahulu", "Error", JOptionPane.ERROR_MESSAGE);
        return; // Stop further execution
    }

    Karyawan karyawan = new Karyawan();
    karyawan.setId(this.karyawanView.getId().getText());
    karyawan.setNama(this.karyawanView.getNama().getText());

    //
karyawan.setJenisKelamin(this.karyawanView.getJenisKelamin().getText());
    karyawan.setJenisKelamin((String)
this.karyawanView.getJenisKelaminComboBox().getSelectedItem());

    karyawan.setHariKerja((String)
this.karyawanView.getHarikerjaComboBox().getSelectedItem());

    App.getKaryawanService().update(karyawan);
    JOptionPane.showMessageDialog(null, "Data Berhasil di Ubah", "Info",
JOptionPane.INFORMATION_MESSAGE);
    clear();
    tampilData();
}

```

- Delete (Hapus) - Menghapus Karyawan dari Daftar:

Class: **KaryawanController.java**

```

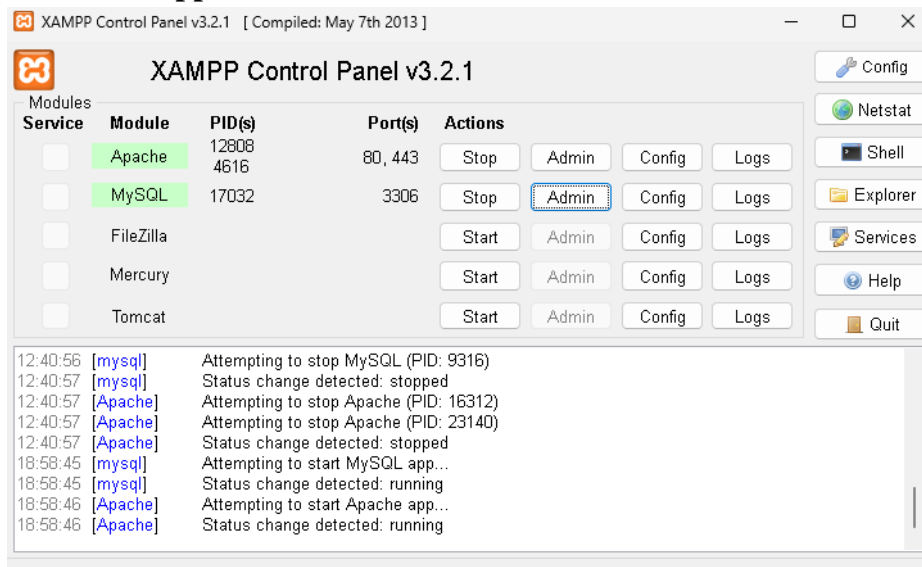
public void deleteKaryawan() {
    if (this.karyawanView.getId().getText().isEmpty()) {
        JOptionPane.showMessageDialog(null, "Data Karyawan Kosong / Belum
Dipilih", "Error", JOptionPane.ERROR_MESSAGE);
    } else {
        Karyawan karyawan = new Karyawan();
        karyawan.setId(this.karyawanView.getId().getText());
        int pilih = JOptionPane.showConfirmDialog(null, "Apakah data ingin
dihapus", "?", "Warning", JOptionPane.YES_NO_OPTION,
JOptionPane.WARNING_MESSAGE);
        if (pilih == JOptionPane.YES_OPTION) {

```

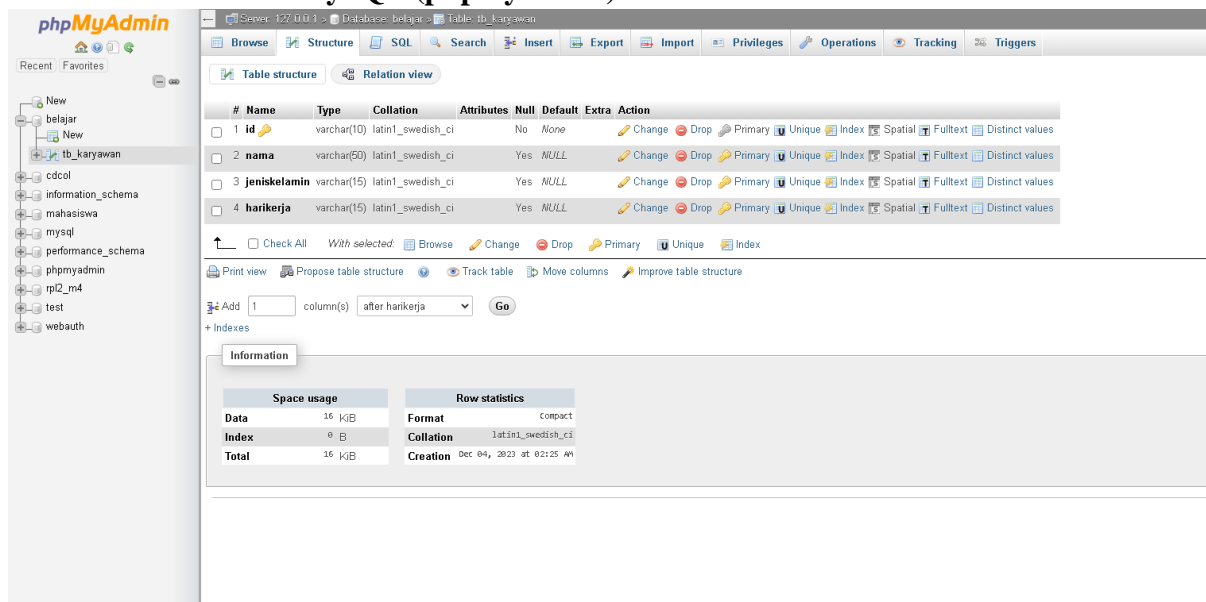
```
App.getKaryawanService().delete(karyawan);
OptionPane.showMessageDialog(null, "Data Berhasil di Hapus",
"Info", JOptionPane.INFORMATION_MESSAGE);
clear();
tampilData();
}
}
}
```


3.1.3 CRUD Database

3.1.3.1 Xampp



3.1.3.2 MySQL (phpmyadmin)



3.1.4 Hasil Output

3.1.4.1 Tampilan awal

DAFTAR KARYAWAN

Id :

Nama :

Jenis Kelamin :

Hari Kerja :

Id	Nama	JenisKelamin	HariKerja
----	------	--------------	-----------

3.1.4.2 Save

DAFTAR KARYAWAN

Id :

Nama :

Jenis Kelamin :

Hari Kerja :

Id	Nama	JenisKelamin	HariKerja
----	------	--------------	-----------

Info
Data Berhasil di Simpan
OK

DAFTAR KARYAWAN

Id :

Nama :

Jenis Kelamin :

Hari Kerja :

Id	Nama	JenisKelamin	HariKerja
1	Ibrahim Bramullah	Laki - laki	SELASA

3.1.4.3 Ubah

DAFTAR KARYAWAN

Id :

Nama :

Jenis Kelamin :

Hari Kerja :

Id	Nama	JenisKelamin	HariKerja
1	Ibrahim Bramullah	Laki - laki	SELASA

Info

Data Berhasil di Ubah

3.1.4.4 Delete

DAFTAR KARYAWAN

Id :

Nama :

Jenis Kelamin :

Hari Kerja :

Id	Nama	JenisKelamin	HariKerja
1	Ibrahim	Laki - laki	SELASA

Warning

Apakah data ingin dihapus ?

Info

Data Berhasil di Hapus

DAFTAR KARYAWAN

Id :

Nama :

Jenis Kelamin :

Hari Kerja :

Id	Nama	JenisKelamin	HariKerja
1	Ibrahim	Laki - laki	SELASA

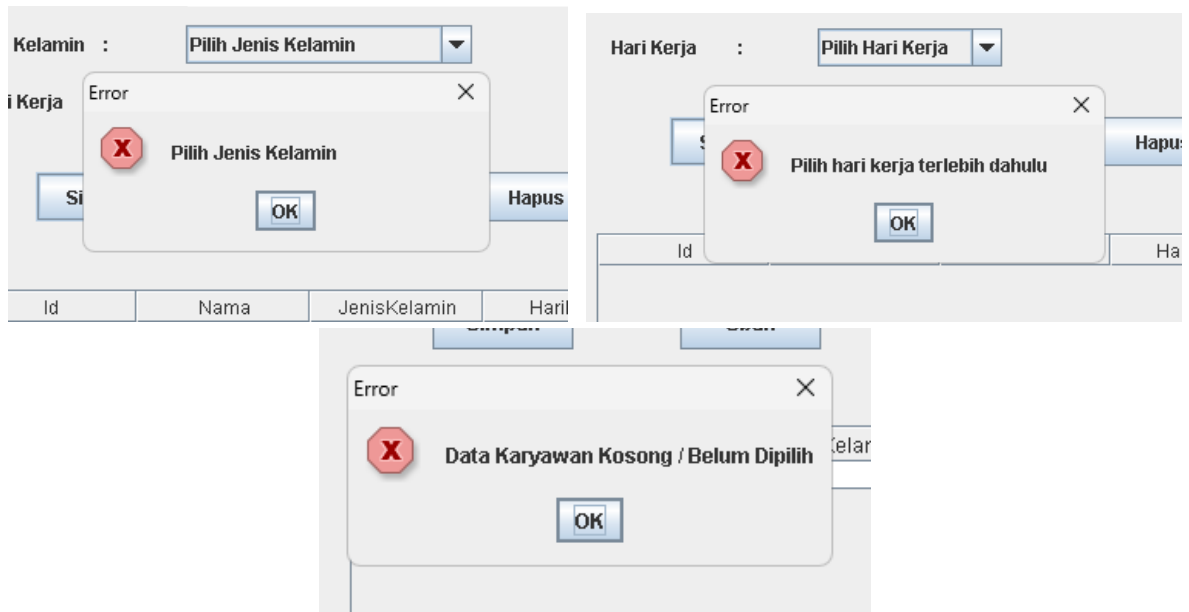
Warning

Apakah data ingin dihapus ?

Info

Data Berhasil di Hapus

3.1.4.5 DialogMessage (Error)



BAB IV

PENUTUP

4.1 Kesimpulan

Dengan adanya Rekayasa Perangkat Lunak 2 ini dapat menambah pengetahuan dalam membuat perangkat lunak, sehingga dalam penulisan ini dibuatlah sebuah project perangkat lunak sederhana yang mengelola data karyawan ke database dengan sistem real time yang dapat melakukan create, input, update, dan delete.

4.2 Saran

Dalam penulisan ini masih terdapat banyak kekurangan, dalam penulisannya maupun pembuatan web yang memiliki tampilan masih sederhana.

DAFTAR PUSTAKA

<https://www.neliti.com/publications/172978/class-library-untuk-pembuatan-aplikasi-crud>

<https://www.exabytes.co.id/blog/apa-itu-xampp-adalah/>

https://perpustakaan.pancabudi.ac.id/dl_file/penelitian/19723_2_BAB_II.pdf