

IF34348 - PEMROGRAMAN LANJUT

**MEMBUAT APLIKASI
DATABASE**

14

Oleh : Andri Heryandi, M.T.

CONTOH KASUS

IF34348 - Pemrograman Lanjut

- **Buatlah suatu aplikasi yang dapat mengolah data pegawai yang tersimpan di database MySQL.**
- **Nama field di data pegawai adalah NIP, Nama, JenisKelamin, dan Tinggi.**
- **Fitur yang harus ada adalah :**
 - **Menampilkan data pegawai**
 - **Menambah data pegawai baru**
 - **Mengedit data pegawai**
 - **Menghapus data pegawai**
 - **Mencari/Filter data pegawai**

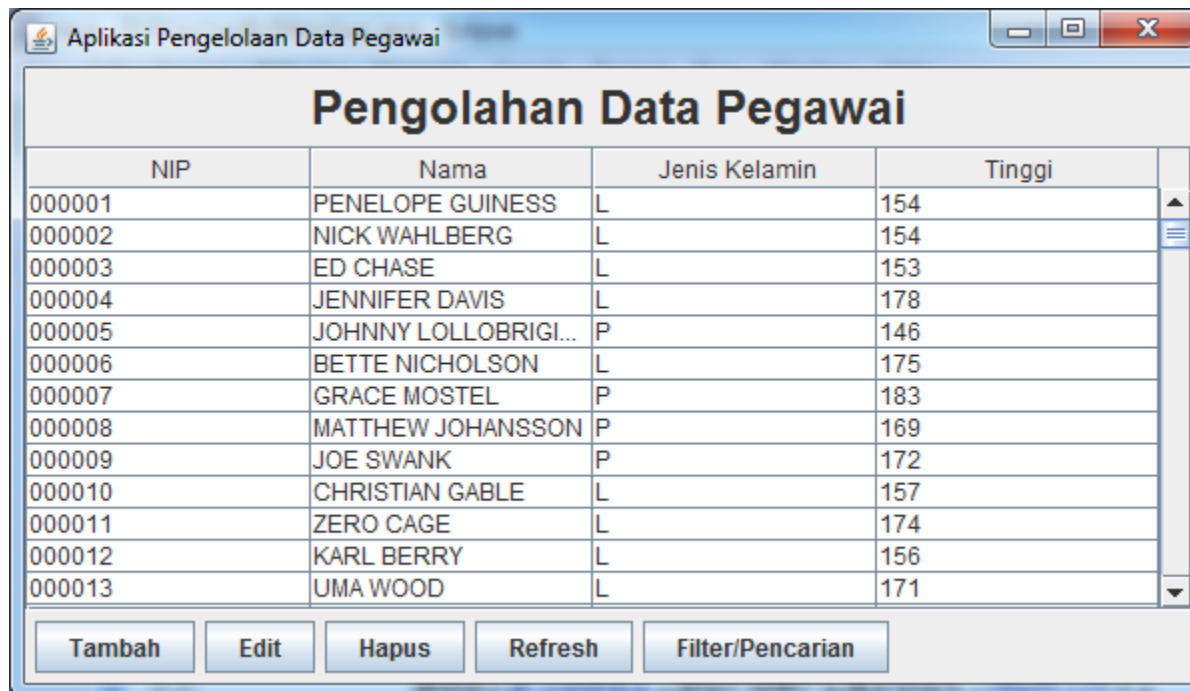


Oleh : Andri Heryandi, M.T.

CONTOH KASUS

IF34348 - Pemrograman Lanjut

■ Tampilan layar aplikasi



The screenshot shows a web application window titled "Aplikasi Pengelolaan Data Pegawai". The main content area is titled "Pengolahan Data Pegawai" and displays a table with the following data:

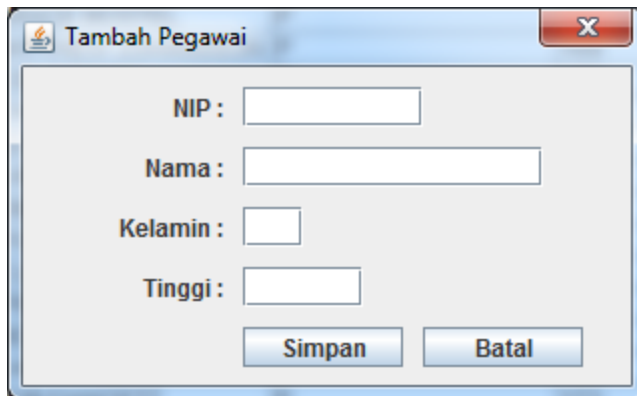
NIP	Nama	Jenis Kelamin	Tinggi
000001	PENELOPE GUINNESS	L	154
000002	NICK WAHLBERG	L	154
000003	ED CHASE	L	153
000004	JENNIFER DAVIS	L	178
000005	JOHNNY LOLLOBRIGI...	P	146
000006	BETTE NICHOLSON	L	175
000007	GRACE MOSTEL	P	183
000008	MATTHEW JOHANSSON	P	169
000009	JOE SWANK	P	172
000010	CHRISTIAN GABLE	L	157
000011	ZERO CAGE	L	174
000012	KARL BERRY	L	156
000013	UMA WOOD	L	171

Below the table, there are five buttons: "Tambah", "Edit", "Hapus", "Refresh", and "Filter/Pencarian".

CONTOH KASUS

IF34348 - Pemrograman Lanjut

■ Layar Penambahan Data



Tambah Pegawai

NIP :

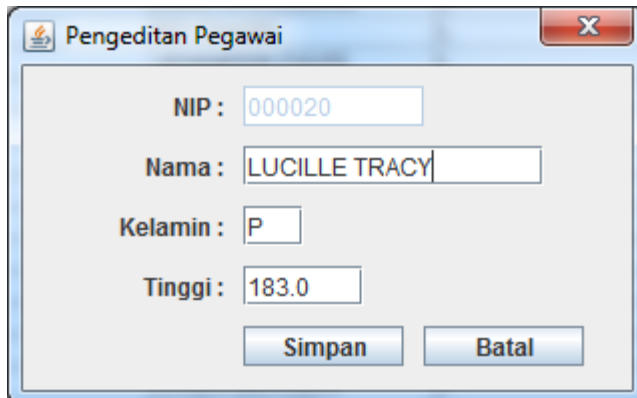
Nama :

Kelamin :

Tinggi :

Simpan Batal

■ Layar Pengeditan Data



Pengeditan Pegawai

NIP :

Nama :

Kelamin :

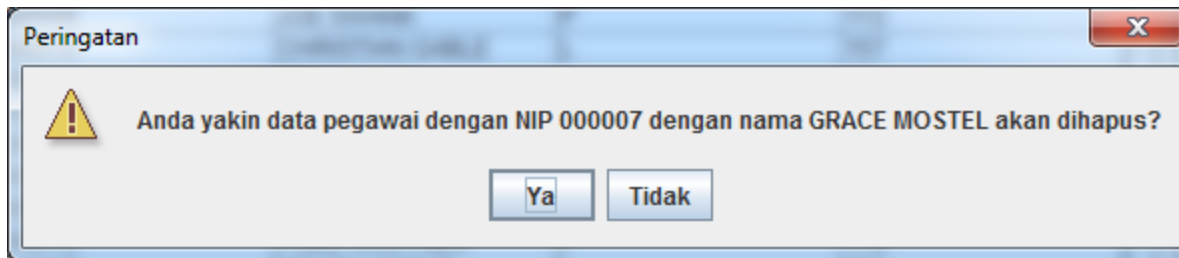
Tinggi :

Simpan Batal

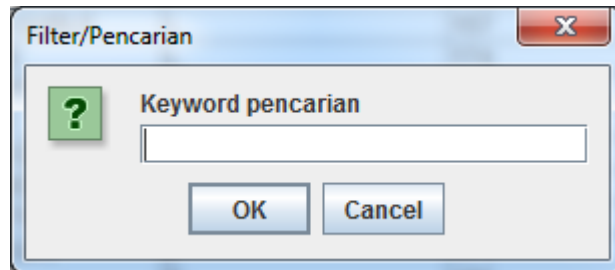
CONTOH KASUS

IF34348 - Pemrograman Lanjut

■ Layar Konfirmasi Penghapusan Data



■ Layar Filtering Data



LANGKAH Pengerjaan

IF34348 - Pemrograman Lanjut

- 1. Membuat Database :**
 - Membuat database dengan nama : dbpegawai
 - Membuat table dengan nama : pegawai
- 2. Buatlah sebuah package, misalnya : dbapp**
- 3. Import library jdbc**
- 4. Buat window-window yang diperlukan :**
 - MainApp : Window Utama
 - FTambahPegawai : Window untuk penambahan Pegawai
 - FEditPegawai : Window untuk pengeditan Pegawai
- 5. Membuat class untuk model Pegawai (class yang mewaliki sebuah data pegawai)**
- 6. Membuat class untuk mengontrol aktivitas database (koneksi, insert, update, delete, select)**



LANGKAH Pengerjaan

IF34348 - Pemrograman Lanjut

7. Membuat Class tabel model untuk data Pegawai
8. Hubungkan class tabel model pegawai ke Jtable
9. Membuat Action untuk tombol Refresh.
10. Membuat Action untuk tombol Filter
11. Membuat Form Tambah Pegawai
12. Membuat Action untuk tombol Tambah
13. Membuat Form Edit Pegawai
14. Membuat Action untuk tombol Edit
15. Membuat Action untuk tombol Hapus



1. MEMBUAT DATABASE

IF34348 - Pemrograman Lanjut

- **Masuk ke mysql client :**

- **Buka shell dari xampp**
- **Tulis : mysql -u root**

- **Isikan sql berikut :**

```
create table dbpegawai;  
use dbpegawai;  
CREATE TABLE pegawai(  
    NIP varchar(10) Primary Key,  
    Nama varchar(30),  
    JenisKelamin char(1),  
    Tinggi int  
);
```

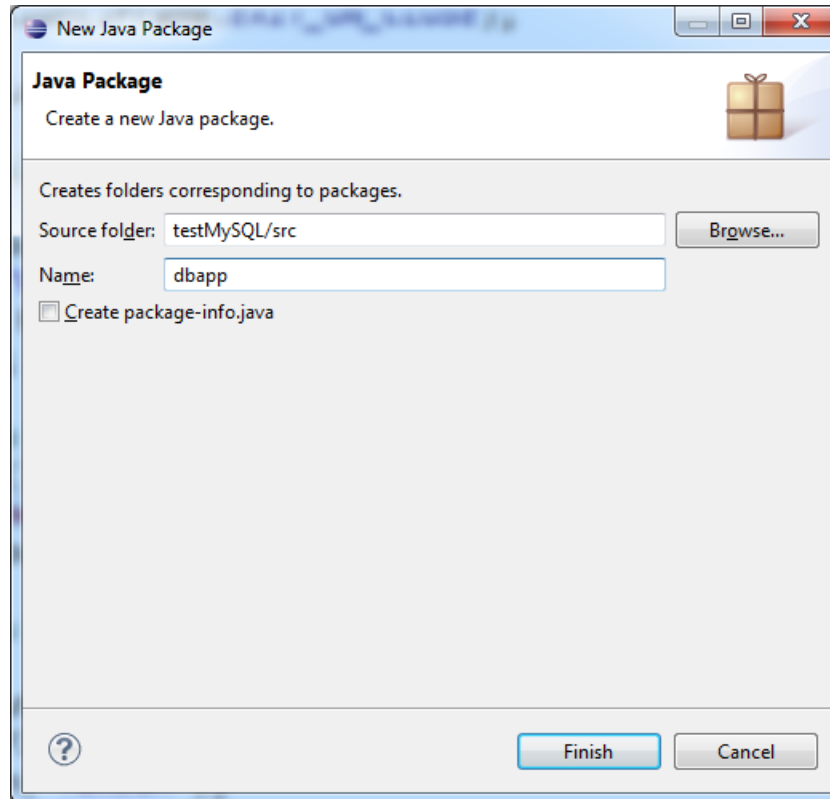


Oleh : Andri Heryandi, M.T.

2. MEMBUAT PACKAGE

IF34348 - Pemrograman Lanjut

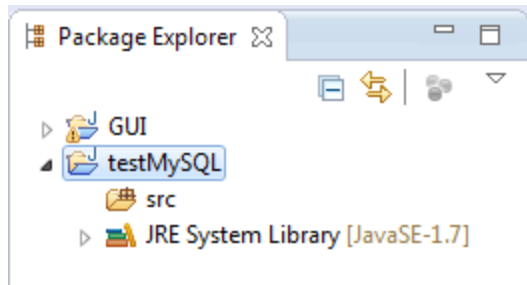
- Klik kanan di nama java project, pilih New → Package.
- Isi dengan nama : “dbapp”. Kemudian klik Finish.



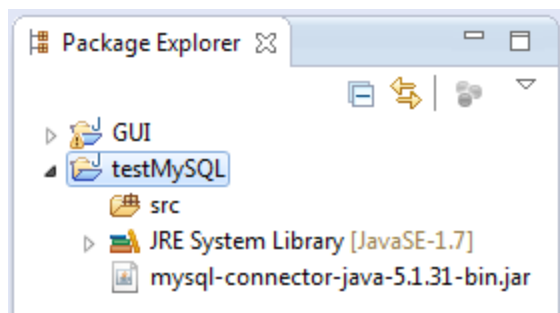
3. IMPORT LIBRARY JDBC

IF34348 - Pemrograman Lanjut

- Perhatikan layar Package Explorer di Eclipse.



- Copy file mysql-connector-java-5.1.31-bin.jar ke file source code anda.
- Refresh folder anda yang ada di Eclipse. Jika berhasil maka akan menampilkan file tersebut.

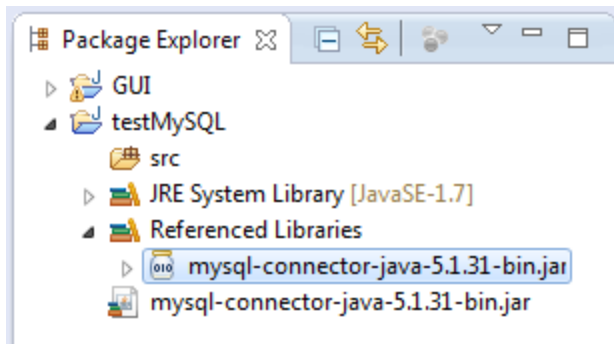


Oleh : Andri Heryandi, M.T.

3. IMPORT LIBRARY JDBC

IF34348 - Pemrograman Lanjut

- Klik kanan di `mysql-connector-java-5.1.31.jar`. Pilih “Build Path”, kemudian pilih “Add to Build Path”.

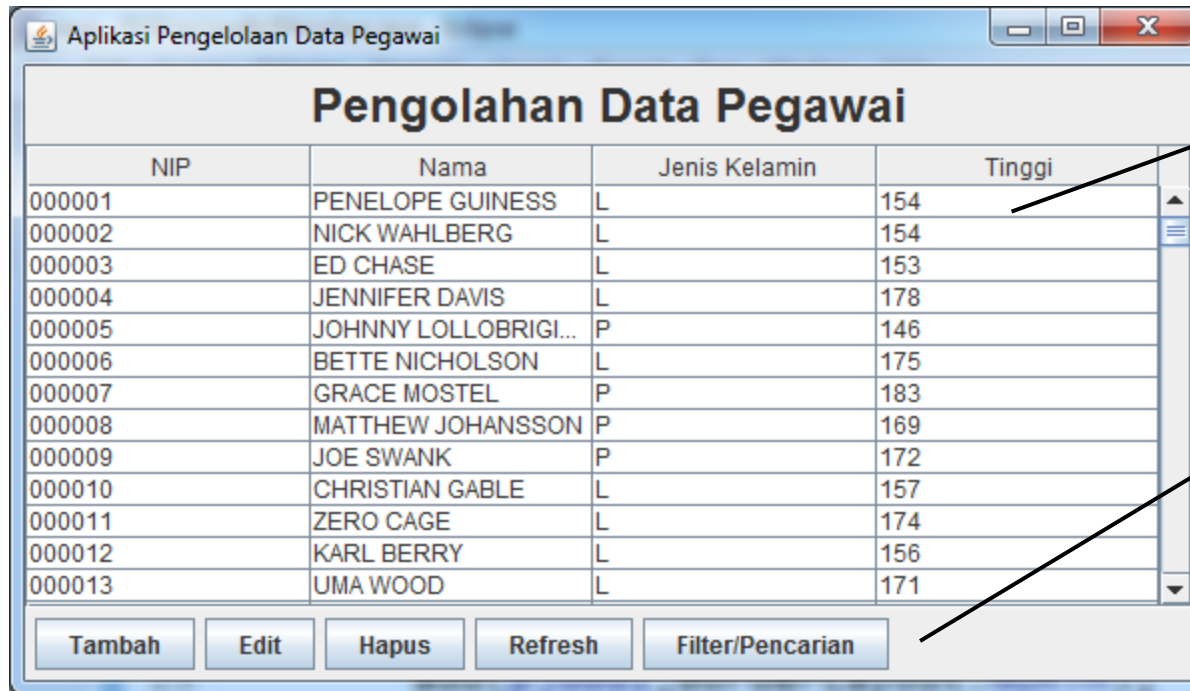


- Jika berhasil maka file tersebut berada pada folder “Referenced Library”.
- JDBC telah bisa digunakan.

4. MEMBUAT WINDOWS UTAMA (MAINAPP)

IF34348 - Pemrograman Lanjut

- Buat Class baru turunan dari JFrame dengan nama MainApp dengan tampilan seperti di bawah ini



Tabeldata:JTable

JButton dengan nama:

- TblTambah
- TblEdit
- TblEdit
- TblRefresh
- TblFilter

4. MEMBUAT WINDOWS UTAMA (MAINAPP)

IF34348 - Pemrograman Lanjut

■ MainApp.java

```
package dbapp;

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class MainApp extends JFrame{
    public JPanel pjudul,pmenu;
    public JScrollPane pdata;
    public JLabel judulapp;
    public JButton tblTambah,tblEdit,tblHapus,tblRefresh,tblFilter;
    public JTable tabeldata;

    public MainApp(){
        // inisialiasasi Frame
        super("Aplikasi Pengelolaan Data Pegawai");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(500,400);
        setLayout(new BorderLayout());
    }
}
```

Oleh : Andri Heryandi, M.T.



4. MEMBUAT WINDOWS UTAMA (MAINAPP)

IF34348 - Pemrograman Lanjut

■ MainApp.java

```
// Membuat Panel Judul
pjudul=new JPanel();
add(pjudul, BorderLayout.NORTH);
judulapp=new JLabel("Pengolahan Data Pegawai", JLabel.CENTER);
judulapp.setFont(new Font("Arial", Font.BOLD, 24));
pjudul.add(judulapp);
```

```
// Membuat Panel Data
tabeldata=new JTable();
JScrollPane pdata = new JScrollPane(tabeldata);
add(pdata, BorderLayout.CENTER);
```

Untuk panel data tidak menggunakan JPanel, tetapi menggunakan JScrollPane agar mendukung adanya Scrollbar.

4. MEMBUAT WINDOWS UTAMA (MAINAPP)

IF34348 - Pemrograman Lanjut

■ MainApp.java

```
// Membuat Panel Menu
pmenu=new JPanel();
add(pmenu, BorderLayout.SOUTH);
pmenu.setLayout(new FlowLayout(FlowLayout.LEFT));
tblTambah=new JButton("Tambah");
tblEdit=new JButton("Edit");
tblHapus=new JButton("Hapus");
tblRefresh=new JButton("Refresh");
tblFilter=new JButton("Filter/Pencarian");
pmenu.add(tblTambah);
pmenu.add(tblEdit);
pmenu.add(tblHapus);
pmenu.add(tblRefresh);
pmenu.add(tblFilter);
```



4. MEMBUAT WINDOWS UTAMA (MAINAPP)

IF34348 - Pemrograman Lanjut

■ MainApp.java

```
// Aksi-aksi tombol di Panel Menu
tblTambah.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
    }
});
tblEdit.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
    }
});
tblHapus.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
    }
});
tblRefresh.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
    }
});
tblFilter.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
    }
});
setVisible(true);
} // Akhir dari public MainApp()
```



4. MEMBUAT WINDOWS UTAMA (MAINAPP)

IF34348 - Pemrograman Lanjut

■ MainApp.java

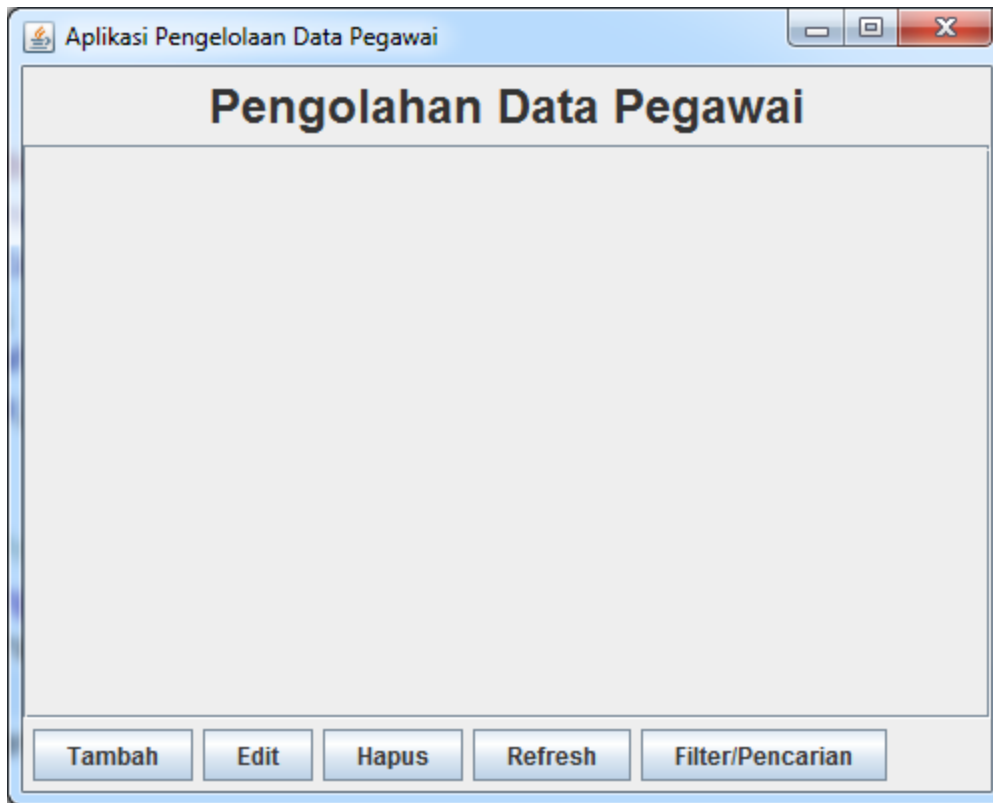
```
public static void main(String[] args) {  
    new MainApp();  
}  
} // Akhir Class MainApp
```



4. MEMBUAT WINDOWS UTAMA (MAINAPP)

IF34348 - Pemrograman Lanjut

■ Run : MainApp.java



- Belum ada data
- Tombol belum ada respon

5. MEMBUAT CLASS MODEL PEGAWAI

IF34348 - Pemrograman Lanjut

- Class Pegawai adalah class yang disebut sebagai class model yang memodelkan/mewakili sebuah model data.



5. MEMBUAT CLASS MODEL PEGAWAI

IF34348 - Pemrograman Lanjut

- Class Pegawai adalah class yang disebut sebagai class model yang memodelkan/mewakili sebuah model data.
- Isi file Pegawai.java

```
package dbapp;
```

```
public class Pegawai {  
    private String NIP;  
    private String Nama;  
    private String JenisKelamin;  
    private int Tinggi;  
    public void setNIP(String nip){  
        NIP=nip;  
    }  
    public void setNama(String nama){  
        Nama=nama;  
    }  
    public void setJenisKelamin(String jk){  
        JenisKelamin=jk;  
    }  
    public void setTinggi(int tinggi){  
        Tinggi=tinggi;  
    }  
}
```



5. MEMBUAT CLASS MODEL PEGAWAI

IF34348 - Pemrograman Lanjut

■ Isi file Pegawai.java

```
public String getNIP(){
    return NIP;
}
public String getNama(){
    return Nama;
}
public String getJenisKelamin(){
    return JenisKelamin;
}
public int getTinggi(){
    return Tinggi;
}
public Pegawai(String nip, String nama, String jk, int tinggi){
    setNIP(nip);
    setNama(nama);
    setJenisKelamin(jk);
    setTinggi(tinggi);
}
}
```



6. MEMBUAT CLASS DATABASE

IF34348 - Pemrograman Lanjut

- **Buat sebuah Class dengan nama Database**
- **Class ini harus dapat melakukan aktivitas seleksi data, tambah data, hapus data, update data dan filtering data.**
- **Method yang ada :**
 - `void tambah_pegawai(Pegawai p)`
Method ini digunakan untuk melakukan insert ke database sesuai parameter `p` yang bertipe class `Pegawai`
 - `void hapus_pegawai(String nip)`
Method ini digunakan untuk melakukan hapus data dari database sesuai parameter `nip` yang bertipe `String`.
 - `void update_pegawai(Pegawai p)`
Method ini digunakan untuk melakukan update data pegawai ke database sesuai parameter `p` yang bertipe class `Pegawai`



6. MEMBUAT CLASS DATABASE

- `Pegawai pilih_1_pegawai(String nip)`
Method ini digunakan untuk melakukan pencarian seorang pegawai berdasarkan nipnya. Method ini digunakan untuk melakukan pencarian pegawai yang akan diedit sebelum ditampilkan dalam form pengeditan.
- `ArrayList<Pegawai> select_seluruh_pegawai()`
Method ini digunakan untuk melakukan pengambilan seluruh data pegawai. Method ini menghasilkan return berupa array of Pegawai (`ArrayList<Pegawai>`). Method ini akan digunakan ketika user mengklik tombol Refresh.
- `ArrayList<Pegawai> filter_pegawai(String keyword)`
Method ini digunakan untuk melakukan pengambilan data pegawai yang namanya sesuai dengan keywordnya. Method ini menghasilkan return berupa array of Pegawai (`ArrayList<Pegawai>`). Method ini akan digunakan ketika user mengklik tombol Filter/Pencarian.

6. MEMBUAT CLASS DATABASE

IF34348 - Pemrograman Lanjut

■ Isi File : Database.java

```
package dbapp;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.ArrayList;

public class Database {
    public final String driver = "com.mysql.jdbc.Driver";
    public final String url = "jdbc:mysql://localhost/dbpegawai";
    public final String username = "root";
    public final String password = "";
```



6. MEMBUAT CLASS DATABASE

IF34348 - Pemrograman Lanjut

■ Method tambah_pegawai

```
public void tambah_pegawai(Pegawai p){
    Connection conn=null;
    Statement stmt=null;
    try{
        Class.forName("com.mysql.jdbc.Driver");
        conn = DriverManager.getConnection(url,username,password);
        stmt = conn.createStatement();
        String sql="insert into pegawai
values ('"+p.getNIP()+"', '"+p.getNama()+"', '"+p.getJenisKelamin()+"', '"+p.getTinggi()+"' )";
        stmt.executeUpdate(sql);
    }
    catch(Exception e){
        System.out.println("Error : "+e.getMessage());
    }
    finally{
        try{
            stmt.close();
        }catch(Exception e){}
        try{
            conn.close();
        }catch(Exception se){}
    }
};
```



6. MEMBUAT CLASS DATABASE

IF34348 - Pemrograman Lanjut

■ Method hapus_pegawai

```
public void hapus_pegawai(String nip){
    Connection conn=null;
    Statement stmt=null;
    try{
        Class.forName("com.mysql.jdbc.Driver");
        conn = DriverManager.getConnection(url,username,password);
        stmt = conn.createStatement();
        String sql="delete from pegawai where nip='"+nip+"'";
        stmt.executeUpdate(sql);
    }
    catch(Exception e){
        System.out.println("Error : "+e.getMessage());
    }
    finally{
        try{
            stmt.close();
        }catch(Exception e){}
        try{
            conn.close();
        }catch(Exception se){}
    }
};
```



6. MEMBUAT CLASS DATABASE

IF34348 - Pemrograman Lanjut

■ Method `update_pegawai`

```
public void update_pegawai(Pegawai p){
    Connection conn=null;
    Statement stmt=null;
    try{
        Class.forName("com.mysql.jdbc.Driver");
        conn = DriverManager.getConnection(url,username,password);
        stmt = conn.createStatement();
        String sql="update pegawai set "
                + "nama='"+p.getNama()+"',"
                + "jeniskelamin='"+p.getJenisKelamin()+"',"
                + "tinggi='"+p.getTinggi()+"'"
                + "where nip='"+p.getNIP()+"'";
        stmt.executeUpdate(sql);
    }
    catch(Exception e){
        System.out.println("Error : "+e.getMessage());
    }
    finally{
        try{
            stmt.close();
        }catch(Exception e){}
        try{
            conn.close();
        }catch(Exception se){}
    }
};
```



6. MEMBUAT CLASS DATABASE

IF34348 - Pemrograman Lanjut

■ Method seleksi 1 pegawai

```
public Pegawai pilih_1_pegawai(String nip){
    Pegawai p=null;
    Connection conn=null;
    Statement stmt=null;
    try{
        Class.forName("com.mysql.jdbc.Driver");
        conn = DriverManager.getConnection(url,username,password);
        stmt = conn.createStatement();
        String sql="select nip,nama,jeniskelamin,tinggi"+
                    " from pegawai where nip='"+nip+"'";
        ResultSet rs = stmt.executeQuery(sql);
        if(rs.next()){
            p=new Pegawai(rs.getString("NIP"), rs.getString("nama"),
                rs.getString("JenisKelamin"), rs.getInt("tinggi"));
        }
        else
            p=null;
        rs.close();
    }
    catch(Exception e){
        System.out.println("Error : "+e.getMessage());
    }
    finally{
        try{
            stmt.close();
        }catch(Exception e){}
        try{
            conn.close();
        }catch(Exception se){}
    }
    return p;
}
```



6. MEMBUAT CLASS DATABASE

IF34348 - Pemrograman Lanjut

■ Method seleksi seluruh pegawai

```
public ArrayList<Pegawai> select_seluruh_pegawai(){
    ArrayList<Pegawai> listPegawai=new ArrayList<Pegawai>();
    Connection conn=null;
    Statement stmt=null;
    try{
        Class.forName("com.mysql.jdbc.Driver");
        conn = DriverManager.getConnection(url,username,password);
        stmt = conn.createStatement();
        String sql="select nip,nama,jeniskelamin,tinggi from pegawai";
        ResultSet rs = stmt.executeQuery(sql);
        while(rs.next()){
            listPegawai.add(new Pegawai(rs.getString("NIP"), rs.getString("Nama"),
                rs.getString("JenisKelamin"),
                rs.getInt("Tinggi")));
        }
        rs.close();
    }
    catch(Exception e){
        System.out.println("Error : "+e.getMessage());
    }
    finally{
        try{
            stmt.close();
        }catch(Exception e){}
        try{
            conn.close();
        }catch(Exception se){}
    }
    return listPegawai;
}
```



6. MEMBUAT CLASS DATABASE

IF34348 - Pemrograman Lanjut

■ Method filter pegawai

```
public ArrayList<Pegawai> filter_pegawai(String keyword){
    ArrayList<Pegawai> listPegawai=new ArrayList<Pegawai>();
    Connection conn=null;
    Statement stmt=null;
    try{
        Class.forName("com.mysql.jdbc.Driver");
        conn = DriverManager.getConnection(url,username,password);
        stmt = conn.createStatement();
        String sql="select nip,nama,jeniskelamin,tinggi from pegawai where nama like '%" +keyword+"%'";
        ResultSet rs = stmt.executeQuery(sql);
        while(rs.next()){
            listPegawai.add(new Pegawai(rs.getString("NIP"),
                                        rs.getString("Nama"),
                                        rs.getString("JenisKelamin"),
                                        rs.getInt("Tinggi")));
        }
        rs.close();
    }
    catch(Exception e){
        System.out.println("Error : "+e.getMessage());
    }
    finally{
        try{
            stmt.close();
        }catch(Exception e){}
        try{
            conn.close();
        }catch(Exception se){}
    }
    return listPegawai;
}
} // Akhir dari class Database
```

7. MEMBUAT CLASS TABLEMODEL UNTUK DATA PEGAWAI

IF34348 - Pemrograman Lanjut

- **TableModel** adalah interface yang menspesifikan method-method yang akan digunakan oleh **Jtabel** untuk bekerja. **Jtabel** dapat menampilkan model data apapun yang mengimplementasikan interface tersebut.



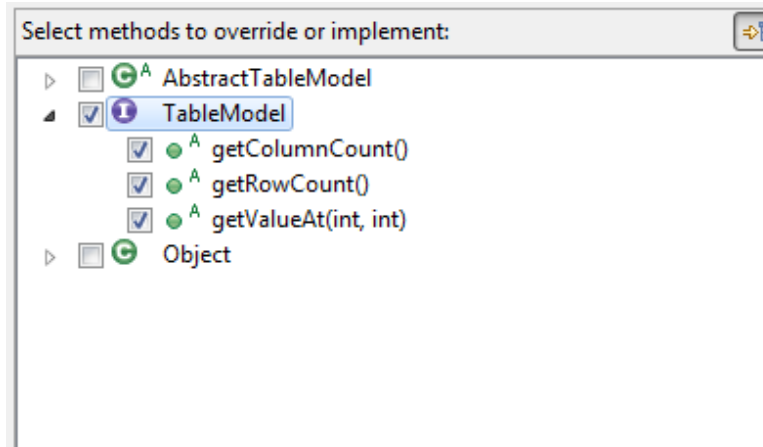
7. MEMBUAT CLASS TABLEMODEL UNTUK DATA PEGAWAI

IF34348 - Pemrograman Lanjut

- Langkah-langkah untuk membuat TabelModel untuk data pegawai adalah :
 - Buat class PegawaiTableModel yang merupakan turunan dari AbstractTableModel

```
public class PegawaiTableModel extends AbstractTableModel {  
  
}
```

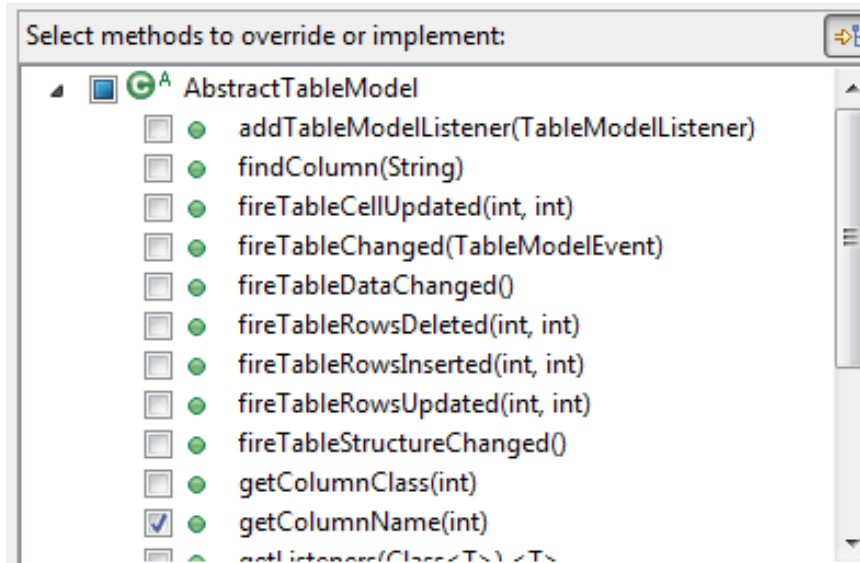
- Implementasikan semua method dari inteface TableModel yang diperlukan dengan cara pilih menu → Source → “Override/Implements method”. Pilih semua method dibawah TableModel.



7. MEMBUAT CLASS TABLEMODEL UNTUK DATA PEGAWAI

IF34348 - Pemrograman Lanjut

- Pilih pula method `getColumnName` yang ada di class `AbstractTableModel`.



- Klik tombol OK

7. MEMBUAT CLASS TABLEMODEL UNTUK DATA PEGAWAI

IF34348 - Pemrograman Lanjut

- Jika berhasil maka ada beberapa method yang harus dioverride dan diimplementasikan

```
public class PegawaiTableModel extends AbstractTableModel {
    public int getColumnCount() {
        return 0;
    }
    public int getRowCount() {
        return 0;
    }
    public Object getValueAt(int arg0, int arg1) {
        return null;
    }
    public String getColumnName(int column) {
        return super.getColumnName(column);
    }
}
```



7. MEMBUAT CLASS TABLEMODEL UNTUK DATA PEGAWAI

IF34348 - Pemrograman Lanjut

- Dikarenakan data yang akan mengisi table ini berupa array dari Pegawai maka tambahkan source code berikut di dalam class PegawaiTableModel:

```
private ArrayList<Pegawai> data;  
public void setData(ArrayList<Pegawai> data){  
    this.data=data;  
}
```

- Tambahkan pula pendefinisian nama-nama kolom berupa array String

```
private String[] namaField = {"NIP", "Nama", "Jenis Kelamin", "Tinggi"};
```

- Modifikasi method getColumnCount(). Method ini digunakan untuk mengetahui berapa kolom yang ada. Karena banyaknya kolom yang akan ditampilkan diambil dari banyaknya field di nama kolom, maka nilai getColumncount() adalah banyaknya data di array namaField.

```
public int getColumnCount() {  
    return namaField.length;  
}
```

Oleh : Andri Heryandi, M.T.



7. MEMBUAT CLASS TABLEMODEL UNTUK DATA PEGAWAI

IF34348 - Pemrograman Lanjut

- Modifikasi method `getRowCount()`. Method ini digunakan untuk mengetahui berapa banyak baris data yang ada. Karena banyaknya baris yang akan ditampilkan diambil dari banyaknya data dari `ArrayList` data, maka nilai `getRowCount()` adalah banyaknya data di `ArrayList` data.

```
public int getRowCount() {  
    return data.size();  
}
```

- Modifikasi method `getColumnName()`. Method ini digunakan oleh `JTable` untuk mengambil nama kolom. Itu artinya method ini harus mereturnkan nama field yang sesuai dengan nomor kolomnya.

```
public String getColumnName(int column) {  
    return namaField[column];  
}
```



7. MEMBUAT CLASS TABLEMODEL UNTUK DATA PEGAWAI

IF34348 - Pemrograman Lanjut

- Modifikasi method `getValueAt`. Method ini digunakan oleh `Jtable` untuk mengambil data pada posisi baris dan kolom yang sesuai. Method ini memberikan 2 buah parameter yaitu `arg0` dan `arg1`, dimana `arg0` adalah nama nomor index baris dan `arg1` adalah nomor index kolom. Untuk mempermudah, ada baiknya jika `arg0` diganti menjadi baris dan `arg1` diganti menjadi kolom. Ada pun isi methodnya adalah :

```
public Object getValueAt(int baris, int kolom) {
    Pegawai p=data.get(baris); // ambil data pada baris ke-baris
    switch(kolom){
        case 0:return p.getNIP();// returnkan nip
        case 1:return p.getNama();// returnkan nama
        case 2:return p.getJenisKelamin();// returnkan jk
        case 3:return p.getTinggi();// returnkan tinggi
        default:return null;// kolom salah, returnkan null
    }
}
```



8. MENGHUBUNGKAN CLASS TABELMODEL PEGAWAI DENGAN JTABLE

IF34348 - Pemrograman Lanjut

1. Tambahkan sebuah object bernama db dengan Class Database di MainApp. Ini dikarenakan untuk menampilkan data kita butuh data yang didapat dari method yang ada di dalam database.
2. Buat pula sebuah object bernama tabelpegawai dengan class PegawaiTableModel di MainApp untuk menyimpan data-data dari database.

```
public Database db;  
public PegawaiTableModel tabelpegawai;
```



8. MENGHUBUNGKAN CLASS TABELMODEL PEGAWAI DENGAN JTABLE

IF34348 - Pemrograman Lanjut

3. Di konstruktor MainApp tambahkan source code untuk membuat object tersebut. Tempatkan sebelum pembuatan JTable.

```
db=new Database(); // buat object database
tabelpegawai=new PegawaiTableModel();// buat object TabelModel
tabelpegawai.setData(db.select_seluruh_pegawai());// ambil
seluruh data, dan isikan ke tabelmodel
```

4. Ubah pembuatan object Jtable agar mengambil data dari TableModelPegawai.

Sebelum :

```
tabeldata=new JTable();
```

Sesudah :

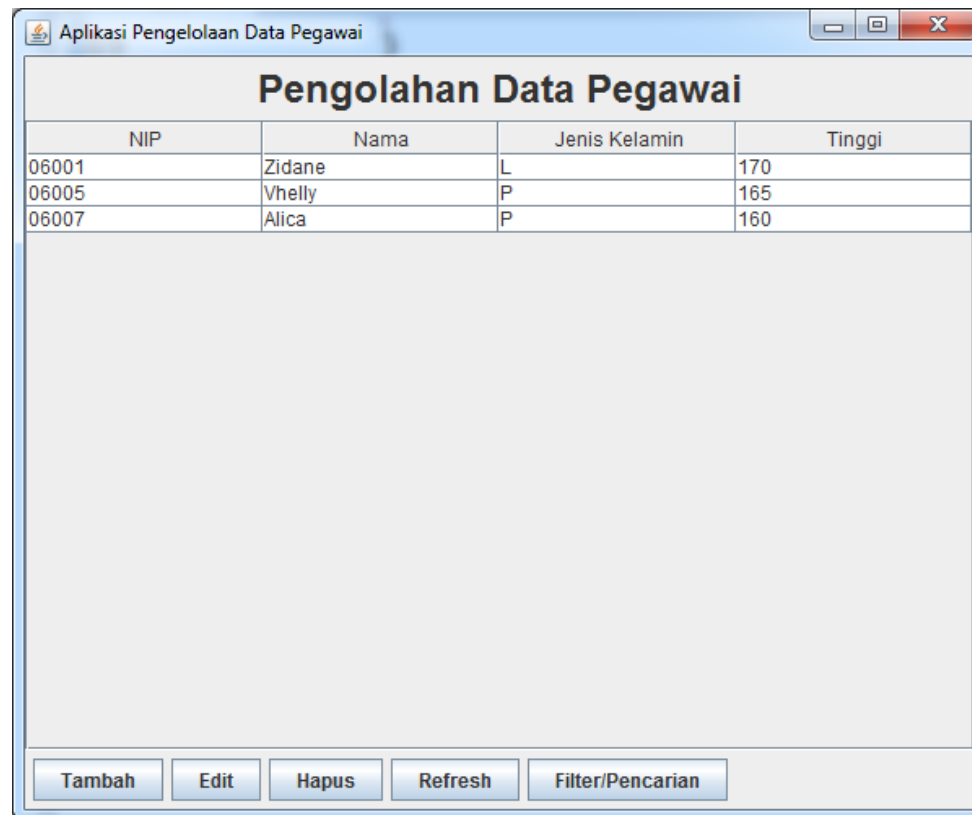
```
tabeldata=new JTable(tabelpegawai);
```



8. MENGHUBUNGKAN CLASS TABELMODEL PEGAWAI DENGAN JTABLE

IF34348 - Pemrograman Lanjut

5. Run Aplikasi. Jika berhasil maka aplikasi sudah dapat menampilkan data.



NIP	Nama	Jenis Kelamin	Tinggi
06001	Zidane	L	170
06005	Vhelly	P	165
06007	Alica	P	160

Oleh : Andri Heryandi, M.T.

9. MEMBUAT ACTION UNTUK TOMBOL REFRESH

IF34348 - Pemrograman Lanjut

- Untuk membuat action tombol Refresh cukup dengan mengisi action pada actionlistener tombol Refresh. Ada pun algoritmanya adalah :
 - Ambil data baru dari database, simpan ke TabelModelPegawai,
 - Refresh JTable

```
tblRefresh.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent arg0) {  
        tabelpegawai.setData(db.select_seluruh_pegawai());  
        tabelpegawai.fireTableDataChanged();  
    }  
});
```

Silahkan ditest dengan cara tambah data di MySQL, kemudian Refresh.



9. MEMBUAT ACTION UNTUK TOMBOL REFRESH

IF34348 - Pemrograman Lanjut

- Tetapi karena fungsi ini tidak hanya digunakan oleh tombol refresh saja (digunakan juga setelah terjadinya tambah, edit atau hapus), maka 2 baris tersebut ada baiknya dijadikan fungsi diluar ActionListener. Jadikan method ini jadi member dari MainApp.

```
public void refreshData() {  
    tabelpegawai.setData(db.select_seluruh_pegawai());  
    tabelpegawai.fireTableDataChanged();  
}
```

- Adapun isi dari action tombol Refresh menjadi :

```
tblRefresh.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent arg0) {  
        refreshData(); // memanggil refreshdata milik MainApp  
    }  
});
```



10. MEMBUAT ACTION UNTUK TOMBOL FILTER

IF34348 - Pemrograman Lanjut

- Action untuk tombol Filter sebenarnya mirip dengan action tombol Refresh. Perbedaannya sebelum melakukan filtering, tombol ini harus menanyakan dulu keyword filteringnya.
- Menanyakan keyword filtering dapat dilakukan dengan `JOptionPane.showInputDialog`

```
tblRefresh.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent arg0) {  
        tabelpegawai.setData(db.select_seluruh_pegawai());  
        tabelpegawai.fireTableDataChanged();  
    }  
});
```

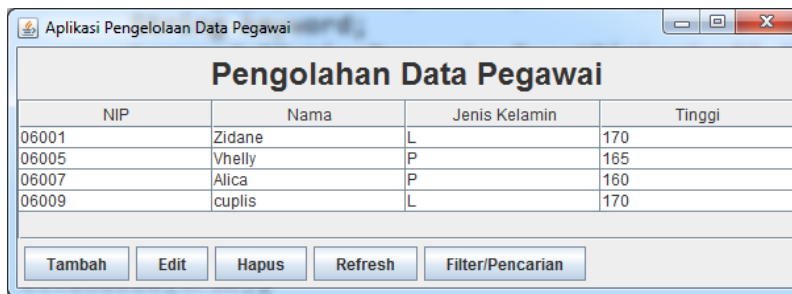
- Jalankan program, test dengan filtering data pegawai.



10. MEMBUAT ACTION UNTUK TOMBOL FILTER

IF34348 - Pemrograman Lanjut

- Tampilan sebelum filter data (semua data muncul)



The screenshot shows a web application window titled "Aplikasi Pengelolaan Data Pegawai". The main content is a table with the following data:

NIP	Nama	Jenis Kelamin	Tinggi
06001	Zidane	L	170
06005	Vhelly	P	165
06007	Alica	P	160
06009	cuplis	L	170

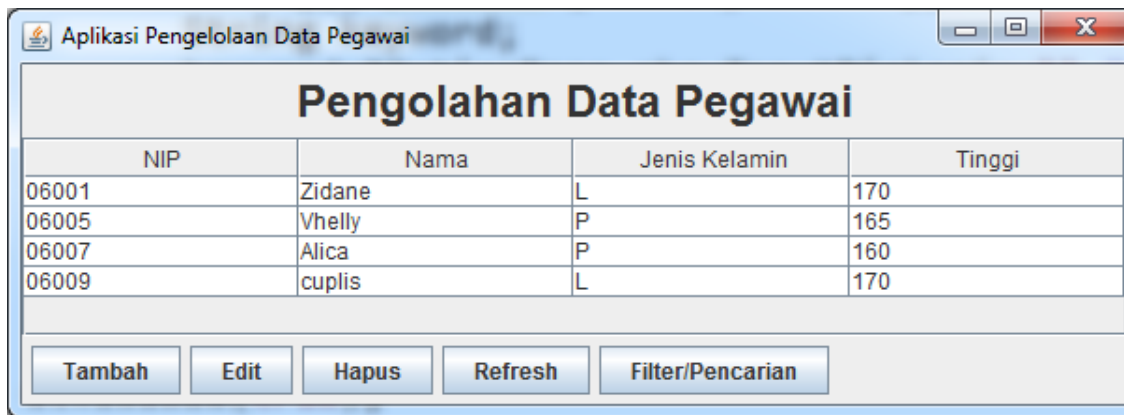
Below the table, there are five buttons: "Tambah", "Edit", "Hapus", "Refresh", and "Filter/Pencarian".

- Isi Filter dengan huruf "a". Maka akan mencari pegawai yang mengandung huruf a

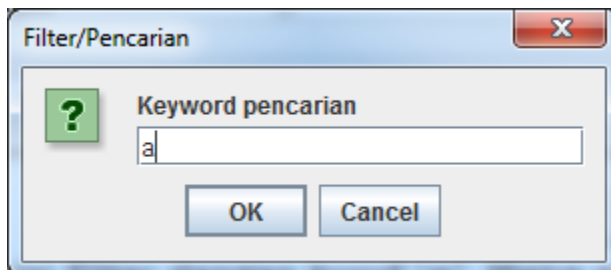
10. MEMBUAT ACTION UNTUK TOMBOL FILTER

IF34348 - Pemrograman Lanjut

- Tampilan sebelum filter data (semua data muncul)



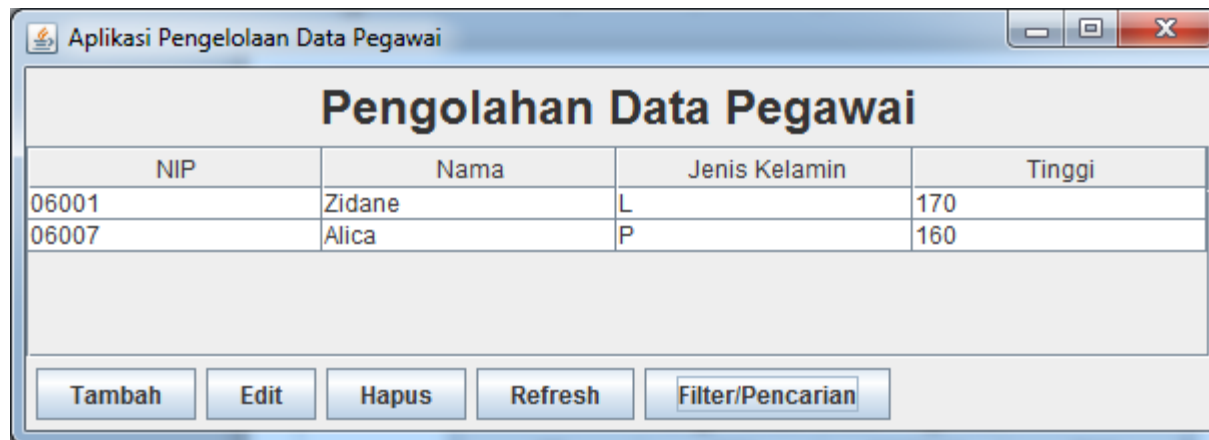
- Isi Filter dengan huruf "a". Maka akan mencari pegawai yang mengandung huruf a



10. MEMBUAT ACTION UNTUK TOMBOL FILTER

IF34348 - Pemrograman Lanjut

- Tampilan setelah filter data (hanya menampilkan nama pegawai yang mengandung huruf a saja).



11. MEMBUAT FORM TAMBAH PEGAWAI

IF34348 - Pemrograman Lanjut

- Buatlah sebuah form sesuai dengan yang diinginkan.
- Buatlah form pengisian dengan membuat class `JDialog`. Sebaiknya jangan gunakan `JFrame`.
- Dengan `JDialog` kita bisa membuat form/window yang sifatnya **modal** yaitu form/window yang tidak memperbolehkan kembali ke window parentnya sebelum window tersebut ditutup.
- Buatlah class dengan nama `FTambahPegawai` yang turunan dari `JDialog`.
- Pada tombol `Simpan` akan berisi algoritma :
 - Ambil data dari form
 - Eksekusi method `tambah_pegawai` yang ada di class `Database`
 - Tutup layar.



Oleh : Andri Heryandi, M.T.

11. MEMBUAT FORM TAMBAH PEGAWAI

IF34348 - Pemrograman Lanjut

■ Isi file : FTambahPegawai.java

```
package dbapp;

import java.awt.Dialog;
import java.awt.event.*;
import javax.swing.*;

public class FTambahPegawai extends JDialog {
    public JLabel lnip,lnama,ljk,ltinggi;
    public JTextField tnip,tnama,tjk,ttinggi;
    public JButton tblsimpan,tblbatal;
    private Database db;
    public FTambahPegawai(JFrame parent){
        super(parent,"Tambah Pegawai",true); // true = dialog modal
        setSize(320,200);
        setLayout(null);
        lnip=new JLabel("NIP :",JLabel.RIGHT);
        lnip.setBounds(10,10,90,20);
        lnama=new JLabel("Nama :",JLabel.RIGHT);
        lnama.setBounds(10,40,90,20);
        ljk=new JLabel("Kelamin :",JLabel.RIGHT);
        ljk.setBounds(10,70,90,20);
        ltinggi=new JLabel("Tinggi :",JLabel.RIGHT);
        ltinggi.setBounds(10,100,90,20);
        tnip=new JTextField();tnip.setBounds(110,10,90,20);
        tnama=new JTextField();tnama.setBounds(110,40,150,20);
        tjk=new JTextField();tjk.setBounds(110,70,30,20);
        ttinggi=new JTextField();ttinggi.setBounds(110,100,60,20);
        tblsimpan=new JButton("Simpan");tblsimpan.setBounds(110,130,80,20);
        tblbatal=new JButton("Batal");tblbatal.setBounds(200,130,80,20);
    }
}
```



11. MEMBUAT FORM TAMBAH PEGAWAI

IF34348 - Pemrograman Lanjut

■ Isi file : FTambahPegawai.java

```
        add(lnip);
        add(lnama);
        add(ljk);
        add(ltinggi);
        add(tnip);
        add(tnama);
        add(tjk);
        add(ttinggi);
        add(tblsimpan);
        add(tblbatal);
        tblbatal.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                setVisible(false);
            }
        });
        tblsimpan.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                db=new Database();
                db.tambah_pegawai(new
Pegawai (tnip.getText(),tnama.getText(),tjk.getText(),Integer.parseInt(ttinggi.getText())));
                setVisible(false);
            }
        });
    }
}
```



12. MEMBUAT ACTION UNTUK TOMBOL TAMBAH

IF34348 - Pemrograman Lanjut

- Buat object bernama formtambah yang mempunyai class FTambahPegawai. Tempatkan definisi atribut MainApp.

```
public FTambahPegawai formtambah;
```

- Buat objectnya sebelum pembuatan Action Tombol Tambah

```
formtambah=new FTambahPegawai(this);
```

- Isilah action untuk tombol tambah dengan method berikut :

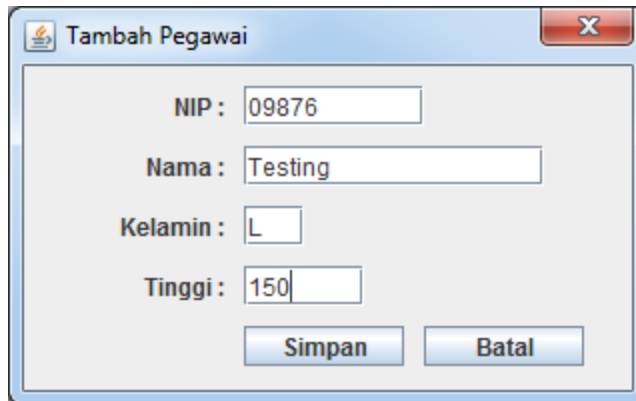
```
tblTambah.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        formtambah.tnip.setText("");  
        formtambah.tnama.setText("");  
        formtambah.tjk.setText("");  
        formtambah.ttinggi.setText("");  
        formtambah.setVisible(true);  
        refreshData();  
    }  
});
```

12. MEMBUAT ACTION UNTUK TOMBOL TAMBAH

IF34348 - Pemrograman Lanjut

■ Test

- Klik tombol Tambah Data, Isi datanya



Tambah Pegawai

NIP : 09876

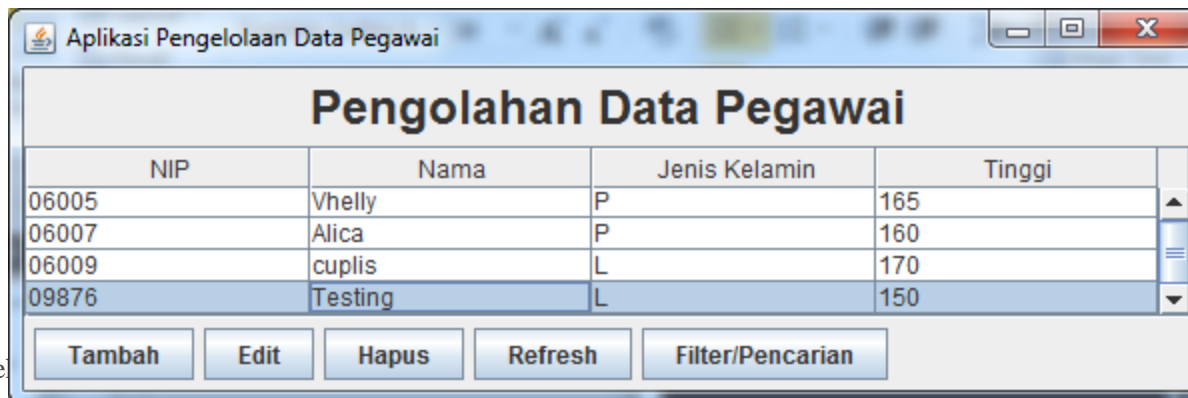
Nama : Testing

Kelamin : L

Tinggi : 150

Simpan Batal

- Klik tombol Simpan.



Aplikasi Pengelolaan Data Pegawai

Pengolahan Data Pegawai

NIP	Nama	Jenis Kelamin	Tinggi
06005	Vhelly	P	165
06007	Alica	P	160
06009	cuplis	L	170
09876	Testing	L	150

Tambah Edit Hapus Refresh Filter/Pencarian

13. MEMBUAT FORM EDIT PEGAWAI

IF34348 - Pemrograman Lanjut

- **Buatlah class dengan nama FEditPegawai yang turunan dari JDialog.**
- **Pada tombol Simpan akan berisi algoritma :**
 - **Ambil data dari form**
 - **Eksekusi method update_pegawai yang ada di class Database**
 - **Tutup layar.**
- **Form Edit Pegawai harus memiliki sebuah method yang menampilkan data sekarang ke Form. Di program ini menggunakan nama setForm.**



13. MEMBUAT FORM EDIT PEGAWAI

IF34348 - Pemrograman Lanjut

■ Isi file : FEditPegawai.java

```
package dbapp;

import java.awt.Dialog;
import java.awt.event.*;
import javax.swing.*;

public class FEditPegawai extends JDialog {
    public JLabel lnip,lnama,ljk,ltinggi;
    public JTextField tnip,tnama,tjk,ttinggi;
    public JButton tblsimpan,tblbatal;
    private Database db;
    public FEditPegawai(JFrame parent){
        super(parent,"Pengeditan Pegawai",true);
        setSize(320,200);
        setLayout(null);
        lnip=new JLabel("NIP :",JLabel.RIGHT);
        lnip.setBounds(10,10,90,20);
        lnama=new JLabel("Nama :",JLabel.RIGHT);
        lnama.setBounds(10,40,90,20);
        ljk=new JLabel("Kelamin :",JLabel.RIGHT);
        ljk.setBounds(10,70,90,20);
        ltinggi=new JLabel("Tinggi :",JLabel.RIGHT);
        ltinggi.setBounds(10,100,90,20);
        tnip=new JTextField();tnip.setBounds(110,10,90,20);tnip.setEnabled(false);
        tnama=new JTextField();tnama.setBounds(110,40,150,20);
        tjk=new JTextField();tjk.setBounds(110,70,30,20);
        ttinggi=new JTextField();ttinggi.setBounds(110,100,60,20);
        tblsimpan=new JButton("Simpan");tblsimpan.setBounds(110,130,80,20);
        tblbatal=new JButton("Batal");tblbatal.setBounds(200,130,80,20);
    }
}
```



13. MEMBUAT FORM EDIT PEGAWAI

IF34348 - Pemrograman Lanjut

■ Isi file : FEditPegawai.java

```
        add(lnip);
        add(lnama);
        add(ljk);
        add(ltinggi);
        add(tnip);
        add(tnama);
        add(tjk);
        add(ttinggi);
        add(tblsimpan);
        add(tblbatal);
        tblbatal.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                setVisible(false);
            }
        });
        tblsimpan.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                db=new Database();
                db.update_pegawai(new
                Pegawai (tnip.getText(),tnama.getText(),tjk.getText(),Integer.parseInt(ttinggi.getText())));
                setVisible(false);
            }
        });
    }
```



13. MEMBUAT FORM EDIT PEGAWAI

IF34348 - Pemrograman Lanjut

■ Isi file : FEditPegawai.java

```
public void setForm(Pegawai p){
    tnip.setText(p.getNIP());
    tnama.setText(p.getNama());
    tjk.setText(p.getJenisKelamin());
    ttinggi.setText(Double.toString(p.getTinggi()));
}
}
```



13. MEMBUAT FORM EDIT PEGAWAI

IF34348 - Pemrograman Lanjut

■ Isi file : FTambahPegawai.java

```
        add(lnip);
        add(lnama);
        add(ljk);
        add(ltinggi);
        add(tnip);
        add(tnama);
        add(tjk);
        add(ttinggi);
        add(tblsimpan);
        add(tblbatal);
        tblbatal.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                setVisible(false);
            }
        });
        tblsimpan.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                db=new Database();
                db.tambah_pegawai(new
Pegawai (tnip.getText(),tnama.getText(),tjk.getText(),Integer.parseInt(ttinggi.getText())));
                setVisible(false);
            }
        });
    }
}
```



14. MEMBUAT ACTION UNTUK TOMBOL EDIT

IF34348 - Pemrograman Lanjut

- Buat object bernama `formedit` yang mempunyai class `FEditPegawai`. Tempatkan definisi atribut `MainApp`.

```
public FEditPegawai formedit;
```

- Buat objectnya sebelum pembuatan Action Tombol Edit

```
formedit=new FEditPegawai(this);
```

- Isilah action untuk tombol edit dengan method berikut :

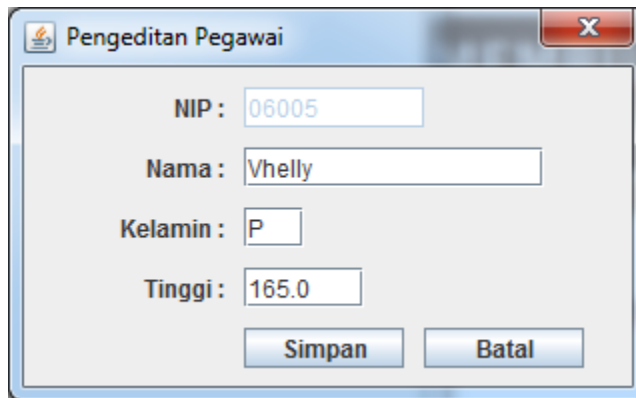
```
tblEdit.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        int baris=tabeldata.getSelectedRow();
        String nip=(String)tabelpegawai.getValueAt (baris,0);
        Pegawai p=db.pilih_1_pegawai (nip);
        if(p!=null) {
            formedit.setForm (p);
            formedit.setVisible(true);
            refreshData ();
        }
        else
            JOptionPane.showMessageDialog (null,"Pegawai dengan
NIP "+nip+" tidak ditemukan.");
    }
});
```

14. MEMBUAT ACTION UNTUK TOMBOL EDIT

IF34348 - Pemrograman Lanjut

■ Test

- Pilih data, klik tombol Edit. Ubah datanya



Pengeditan Pegawai

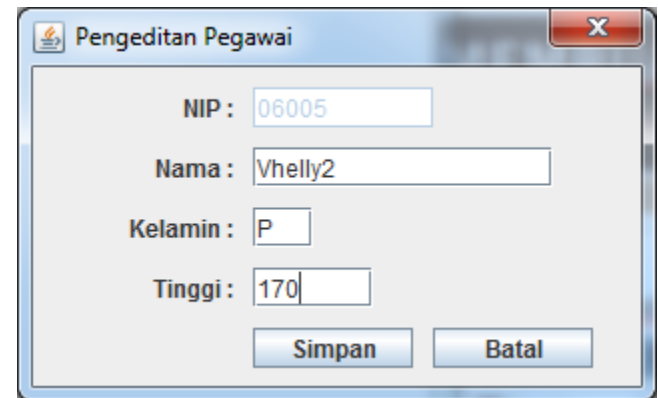
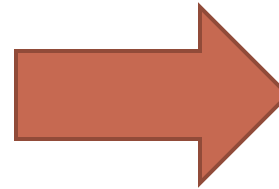
NIP : 06005

Nama : Vhelly

Kelamin : P

Tinggi : 165.0

Simpan Batal



Pengeditan Pegawai

NIP : 06005

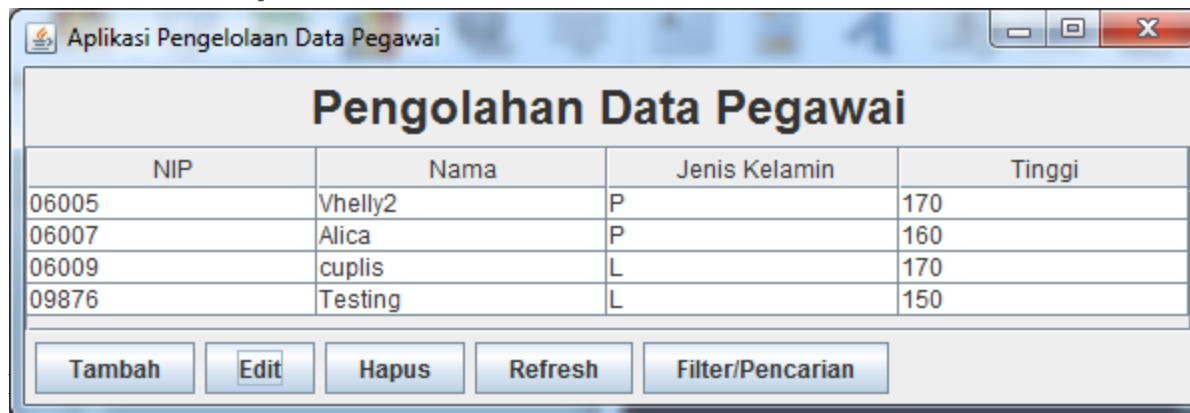
Nama : Vhelly2

Kelamin : P

Tinggi : 170

Simpan Batal

- Klik tombol Simpan.



Aplikasi Pengelolaan Data Pegawai

Pengolahan Data Pegawai

NIP	Nama	Jenis Kelamin	Tinggi
06005	Vhelly2	P	170
06007	Alica	P	160
06009	cuplis	L	170
09876	Testing	L	150

Tambah Edit Hapus Refresh Filter/Pencarian

Oleh : Andri

15. MEMBUAT ACTION UNTUK TOMBOL HAPUS

IF34348 - Pemrograman Lanjut

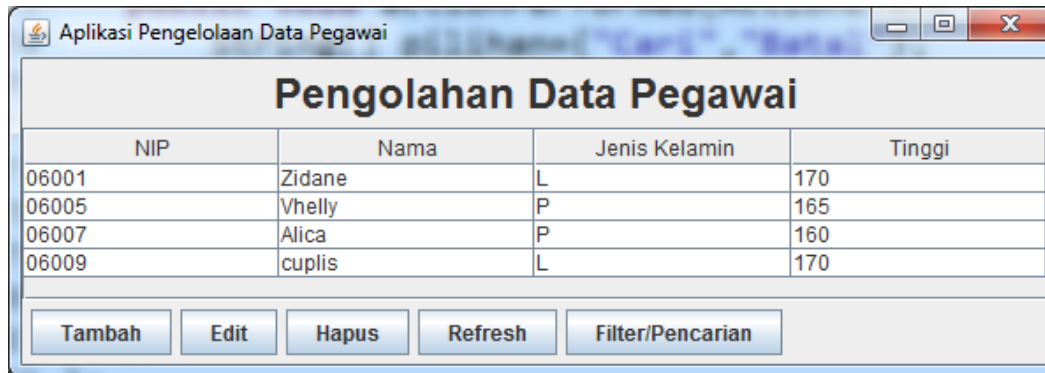
- Isilah action tombol hapus dengan kode berikut yang algoritmanya adalah :
 - Deteksi di baris berapa kursor berada
 - Ambil datanya dari database
 - Tampilkan konfirmasi penghapusan.
 - Jika menjawab Ya, maka hapus data dari database.
 - Refresh data yang tampil di JTable sesuai data dari database

```
tblHapus.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        int baris=tabeldata.getSelectedRow();
        String nip=(String)tabelpegawai.getValueAt (baris,0);
        String nama=(String)tabelpegawai.getValueAt (baris, 1);
        Object[] pilihan = { "Ya", "Tidak" };
        int jawaban=JOptionPane.showOptionDialog(null, "Anda yakin data
pegawai dengan NIP "+nip+" dengan nama "+nama+" akan dihapus?",
"Peringatan",JOptionPane.DEFAULT_OPTION, JOptionPane.WARNING_MESSAGE,null, pilihan, pilihan[0]);
        if(jawaban==0){
            db.hapus_pegawai (nip);
            refreshData ();
        }
    }
});
```

15. MEMBUAT ACTION UNTUK TOMBOL HAPUS

IF34348 - Pemrograman Lanjut

- Data sebelum dihapus :

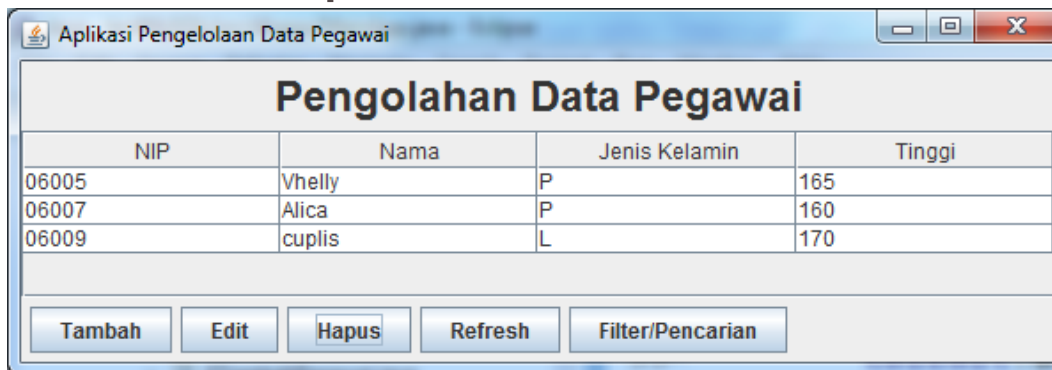


The screenshot shows a web application window titled "Aplikasi Pengelolaan Data Pegawai". The main content area is titled "Pengolahan Data Pegawai" and contains a table with the following data:

NIP	Nama	Jenis Kelamin	Tinggi
06001	Zidane	L	170
06005	Vhelly	P	165
06007	Alica	P	160
06009	cuplis	L	170

Below the table, there are five buttons: "Tambah", "Edit", "Hapus", "Refresh", and "Filter/Pencarian".

- Hapus Data Pertama.
- Data setelah dihapus :



The screenshot shows the same web application window after the first row has been deleted. The table now contains the following data:

NIP	Nama	Jenis Kelamin	Tinggi
06005	Vhelly	P	165
06007	Alica	P	160
06009	cuplis	L	170

The buttons "Tambah", "Edit", "Hapus", "Refresh", and "Filter/Pencarian" are still present at the bottom.

Oleh : Andri Heryandi, M.T.