

LAPORAN TUGAS PROYEK
“APLIKASI PENDATAAN DATA DAN PERHITUNGAN GAJI BERSIH
PEGAWAI CV.XXXI”



Paramadina Mulya Majid

13020180006

B1

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MUSLIM INDONESIA
MAKASSAR
2020

BAB 1

PENDAHULUAN

Kasus yang saya ambil yaitu mengenai perhitungan gaji bersih pegawai CV. XXXI. Perusahaan ini masih melakukan proses pendataan data pegawai dan perhitungan gaji dengan cara menuliskannya pada buku perusahaan. Untuk mempermudah pegawai dalam melakukan proses tersebut maka dibuatlah sebuah aplikasi yang bernama “**Aplikasi Pendataan Data Dan Perhitungan Gaji Bersih Pegawai CV.XXXI**”. Kasus tersebut memiliki ketentuan sebagai berikut :

1. Pengguna terlebih dahulu menginputkan jumlah pegawai yang ingin dimasukkan datanya kedalam form pegawai. Form tersebut berisi NIP (Nomor Induk Pegawai), Nama Pegawai, Gaji Pokok, Golongan, Jam Kerja dan Lembur yang diinput melalui keyboard.
2. Untuk menghitung gaji bersih pegawai terdapat beberapa ketentuan sebagai berikut :
 - a. Terlebih dahulu menghitung besar tunjangan menggunakan rumus :
Tunjangan = Gaji pokok*15%
 - b. Kemudian menghitung besar pajak dengan menggunakan rumus :
Pajak = 10% * (Gaji Pokok + Tunjangan)
 - c. Lalu untuk menghitung gaji bersih dapat menggunakan rumus :
Gaji Bersih = Gaji Pokok + Tunjangan – Pajak
3. Untuk menghitung total gaji pegawai dapat menggunakan rumus :
Total Gaji = Gaji Tetap Sesuai Golongan + (Lembur* Tarif Lembur)

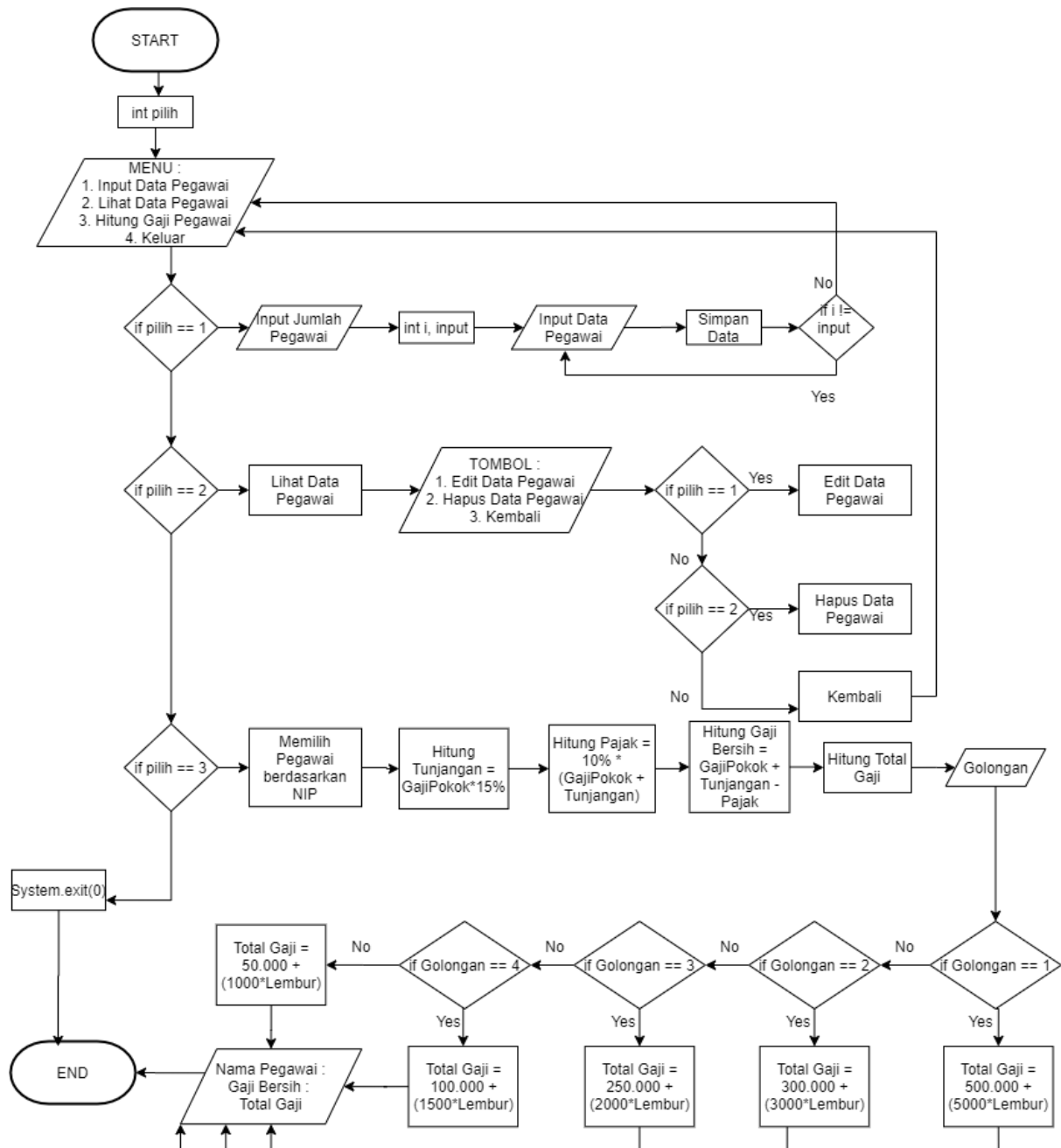
Adapun ketentuan dalam menentukan gaji tetap dan gaji per jam pegawai berdasarkan golongan :

Golongan	Gaji Tetap (Rp)	Upah Lembur per Jam (Rp)
1	500.000	5000
2	300.000	3000
3	250.000	2000
4	100.000	1500
5	50.000	1000

BAB 2

ANALISIS DAN DESAIN PROSEDUR

2.1 Flowchart Alur Program



Gambar 1. Flowchart Alur Program

Pada Gambar 1 merupakan sebuah alur program ketika program tersebut dijalankan. Dimulai dari pemilihan menu. Jika pengguna memilih menu 1 yaitu input data pegawai maka program tersebut akan beralih ke penginputan jumlah pegawai untuk dimasukkan datanya kedalam form data pegawai. Selanjutnya pengisian form data pegawai dan menyimpan data. Apabila masih ada data pegawai yang ingin

diinputkan / batas jumlah pegawai pada saat penginputan belum mencapai batas maka proses penginputan data akan terus dilakukan hingga sudah tidak ada lagi data pegawai yang ingin dimasukkan. Jika sudah tidak ada maka program akan otomatis kembali pada menu utama.

Jika pengguna memilih menu lihat data maka didalamnya pengguna dapat melakukan beberapa aktivitas seperti melihat data dan beberapa pilihan tombol seperti edit data, hapus data dan kembali ke menu awal. Jika pengguna menekan tombol edit maka pengguna akan melakukan proses perubahan data pegawai. Jika pengguna menekan tombol hapus maka pengguna akan melakukan proses penghapusan data pegawai dan jika pilihan terakhir pengguna adalah menekan tombol kembali maka program akan beralih ke menu utama.

Jika pengguna memilih menu hitung gaji pegawai maka didalamnya pengguna dapat memilih pegawai yang akan dihitung gajinya dan proses perhitungannya juga memiliki ketentuan dan untuk total gaji dapat dilihat dari golongan pegawai. ketika proses perhitungan telah selesai maka program akan otomatis menampilkan nama pegawai, hasil perhitungan gaji bersih dan hasil perhitungan total gaji pegawai.

Jika pengguna menekan tombol keluar maka program tersebut akan otomatis keluar/menutup programnya sendiri menggunakan *system.exit(0)*.

2.1 Konsep Pemrograman Berorientasi Objek

2.1.1 Encapsulation

Encapsulation merupakan sebuah proses pembungkusan variable dan method dalam sebuah objek dalam baguan yang terlindungi. Pembungkusan ini mendefinisikan perilaku dan melindungi program dan data yang sedang diolah agar tidak diakses sembarangan oleh program lain.

Pada aplikasi yang saya buat, class yang menerapkan konsep encapsulation ini yaitu class “Pegawai”. Berikut penggalan kode program pada class tersebut :

```

/*
    Nama      : Paramadina Mulya Majid
    Stambuk   : 13020180006
    Hari/Tgl  : Kamis, 18 Juni 2020
    Waktu     : 23.30 WITA
*/

package Source;

import java.util.Objects;

public class Pegawai {
    String nip;
    String nama_pegawai;
    String gaji_pokok;
    String golongan;
    String jam_kerja;
    String lembur;

    public Pegawai() { }

    public Pegawai(String nip, String nama_pegawai, String gaji_pokok, String golongan, String jam_kerja, String lembur) {
        this.nip = nip;
        this.nama_pegawai = nama_pegawai;
        this.gaji_pokok = gaji_pokok;
        this.golongan = golongan;
        this.jam_kerja = jam_kerja;
        this.lembur = lembur;
    }

    public String getNip() {
        return nip;
    }

    public void setNip(String nip) {
        this.nip = nip;
    }
}

```

Gambar 2. Penggalan Kode Program yang Menerapkan Konsep Encapsulation

Pada Gambar 2 merupakan penggalan dari kode program yang menerapkan encapsulation pada class pegawai. Untuk kode program selanjutnya itu sama dengan method getNip dan setNip hanya saja penamaannya yang berbeda sesuai dengan variable yang telah dibuat seperti nama_pegawai, gaji_pokok, golongan, jam_kerja dan lembur

2.1.2 Inheritance

Inheritance digunakan untuk beberapa class turunan dari sebuah class lain. Setiap sub classnya dapat memiliki perbedaan bentuk diantara sub class yang lainnya.

Pada aplikasi yang saya buat, salah satu class yang menerapkan konsep inheritance adalah class “InputData”. Berikut contoh penggalan kode program pada class tersebut :

```

/*
    Nama      : Paramadina Mulya Majid
    Stambuk   : 13020180006
    Hari/Tgl  : Kamis, 18 Juni 2020
    Waktu     : 23.30 WITA
*/

package View;

import Source.ImplementPegawai;
import Source.InterfacePegawai;
import Source.KoneksiDB;
import Source.Pegawai;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

public class InputData extends javax.swing.JFrame {

    DefaultTableModel model = new DefaultTableModel();
    InterfacePegawai interfacePegawai;
    List record = new ArrayList();
    int baris;
    String kGol;
    String PindahKesebelah;
    int i = 1;
}

```

Gambar 3. Penggalan Kode Program yang Menerapkan Konsep Inheritance

Pada Gambar 3 merupakan penggalan dari kode program yang menerapkan encapsulation pada class inputData. Maksud dari *extends* disini adalah kita meng-extend class java.swing.JFrame sehingga dapat mewariskan semua properti dan method yang ada pada class tersebut.

2.1.3 Polymorphism

Polymorphism merupakan sub class yang berasal dari super class yang sama namun memiliki bentuk yang berbeda diantara sub-sub class tersebut.

Pada aplikasi yang saya buat, class yang menerapkan konsep polymorphism ini adalah class “Implement Pegawai” Berikut penggalan kode program dari class tersebut :

```
/*
    Nama      : Paramadina Mulya Majid
    Stambuk   : 13020180006
    Hari/Tgl  : Kamis, 18 Juni 2020
    Waktu     : 23.49 WITA
*/
package Source;

import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;

public class ImplementPegawai implements InterfacePegawai {

    @Override
    public Pegawai insert(Pegawai o) throws SQLException {
        PreparedStatement st = KoneksiDB.getConnection().prepareStatement("insert into pegawai values(?, ?, ?, ?, ?, ?)");
        st.setString(1, o.getNip());
        st.setString(2, o.getNama_Pegawai());
        st.setString(3, o.getGaji_Pokok());
        st.setString(4, o.getGolongan());
        st.setString(5, o.getJam_Kerja());
        st.setString(6, o.getLembur());
        st.executeUpdate();
        return o;
    }

    @Override
    public void update(Pegawai o, String nipAwal) throws SQLException {
        PreparedStatement st = KoneksiDB.getConnection().prepareStatement("update pegawai set nama_pegawai='"+o.getNama_Pegawai()+"', gaji_pokok='"+o.getGaji_Pokok()+"', nip='"+o.getNip()+"', golongan='"+o.getGolongan()+"', jam_kerja='"+o.getJam_Kerja()+"', lembur='"+o.getLembur()+"' where nip='"+nipAwal+"'");
        st.executeUpdate();
    }
}
```

Gambar 4. Penggalan Kode Program yang Menerapkan Konsep Polymorphism

Pada Gambar 4 terdapat sebuah keyword *implement* yang berarti sifat yang terdapat pada interfacePegawai akan diimplementasikan kedalam class Implement Pegawai.

BAB 3

HASIL IMPLEMENTASI

1. Database

Untuk menyimpan data-data pegawai yang telah diinputkan akan masuk kedalam database yang didalamnya terdapat sebuah table “pegawai”.

```
1  /*
2      Nama      : Paramadina Mulya Majid
3      Stambuk   : 13020180006
4      Hari/Tgl  : Kamis, 18 Juni 2020
5      Waktu     : 22.12 WITA
6  */
7  • create database proyekPBO;
8  • use proyekPBO;
9
10 • create table pegawai (nip varchar(5) primary key,
11     nama_pegawai varchar(50),
12     gaji_pokok int,
13     golongan enum('1','2','3','4','5'),
14     jam_kerja int,
15     lembur int);
```

Gambar 5. Database proyekPBO

2. Class Koneksi Database

Pembuatan Class Koneksi Database agar database terhubung dengan aplikasi yang dibuat menggunakan netbeans.

```
1  /*
2      Nama      : Paramadina Mulya Majid
3      Stambuk   : 13020180006
4      Hari/Tgl  : Kamis, 18 Juni 2020
5      Waktu     : 23.28 WITA
6  */
7  package Source;
8
9  import java.sql.Connection;
10 import java.sql.DriverManager;
11 import java.sql.SQLException;
12 import java.util.logging.Level;
13 import java.util.logging.Logger;
14
15 public class KoneksiDB {
16     private static Connection con;
17     public static Connection getConnection(){
18         if(con==null){
19             try {
20                 DriverManager.registerDriver(new com.mysql.jdbc.Driver());
21                 con=DriverManager.getConnection("jdbc:mysql://localhost:3306/proyekPBO","root","");
22             } catch (SQLException ex) {
23                 Logger.getLogger(KoneksiDB.class.getName()).log(Level.SEVERE, null, ex);
24             }
25         }
26         return con;
27     }
28 }
```

Gambar 6. Koneksi Database proyekPBO

3. Class Menu

Pembuatan class ini terdiri atas 3 button yaitu :

a. Button Input Data Pegawai

Ketika kita ingin menginputkan data pegawai, maka tekanlah tombol seperti pada Gambar 7.



Gambar 7. Button Input Data Pegawai

Untuk Action dari button pada Gambar 8 maksudnya adalah ketika kita menekan button input data maka tampilan menu awal hilang dan akan beralih ke tampilan class JmlPegawai yang dimana berisi inputan jumlah pegawai yang ingin kita masukkan datanya.

```
82 private void InputDataPegawaiActionPerformed(java.awt.event.ActionEvent evt) {  
83     // TODO add your handling code here:  
84     JmlPegawai jp = new JmlPegawai();  
85     jp.setVisible(true);  
86     this.setVisible(false);  
87 }
```

Gambar 8. Action Button Input Data Pegawai

b. Button Lihat Data Pegawai

Ketika kita ingin menginputkan data pegawai, maka tekanlah tombol seperti pada Gambar 9.



Gambar 9. Button Lihat Data Pegawai

Untuk Action dari button pada Gambar 10 maksudnya adalah ketika kita menekan button lihat data maka tampilan menu awal hilang dan akan beralih ke tampilan class DataPegawai yang dimana berisi data pegawai yang telah dimasukkan.

```
89 private void LihatDataPegawaiActionPerformed(java.awt.event.ActionEvent evt) {  
90     try {  
91         // TODO add your handling code here:  
92         DataPegawai dp = new DataPegawai();  
93         dp.setVisible(true);  
94         this.setVisible(false);  
95     } catch (SQLException ex) {  
96         Logger.getLogger(Menu.class.getName()).log(Level.SEVERE, null, ex);  
97     }  
98 }
```

Gambar 10. Action Button Lihat Data Pegawai

c. Button Hitung Gaji Pegawai

Ketika kita ingin menginputkan data pegawai, maka tekanlah tombol seperti pada Gambar 11.



Gambar 11. Button Hitung Gaji Pegawai

Untuk Action dari button pada Gambar 12 maksudnya adalah ketika kita menekan button hitung gaji maka tampilan menu awal hilang dan akan beralih ke tampilan class HitungGajiPegawai yang dimana berisi perhitungan gaji bersih dan total gaji pegawai.

```
100 private void HitungGajiPegawaiActionPerformed(java.awt.event.ActionEvent evt) {  
101     // TODO add your handling code here:  
102     HitungGajiPegawai hgp = new HitungGajiPegawai();  
103     hgp.setVisible(true);  
104     this.setVisible(false);  
105 }
```

Gambar 12. Action Button Hitung Gaji Pegawai

4. Class Jumlah Pegawai

Class ini digunakan untuk memasukkan jumlah pegawai yang akan diinputkan pada form input data pegawai. Pada Gambar 13 merupakan Action dari button OK yang berfungsi untuk mengalihkan tampilan class `jmlPegawai` ke class `InputData`. Terdapat `this.setVisible(false)` yang berarti tampilan `jmlPegawai` akan hilang ketika menekan tombol OK.

```
65 private void OKActionPerformed(java.awt.event.ActionEvent evt) {  
66     // TODO add your handling code here:  
67     String Sinput = txtJumlahPegawai.getText();  
68     new InputData(Sinput).setVisible(true);  
69     this.setVisible(false);  
70  
71 }
```

Gambar 13. Action Button OK

5. Class Input Data

Pada class ini terdapat button simpan yang digunakan untuk menyimpan data yang telah diinputkan. Untuk Action pada button simpan dapat dilihat pada gambar dibawah ini :

```
private void SimpanActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try {  
        if (txtNip.getText().equals("") && txtNamaPegawai.getText().equals("") && txtGajiPokok.getText().equals("")  
            && CbGolongan.getSelectedItem().equals("") && txtJamKerja.getText().equals("") && txtLembur.getText().equals("")) {  
            JOptionPane.showMessageDialog(this, "silahkan isi data yang akan ditambah");  
        } else {  
            //Integer.parseInt(txtGajiPokok);  
            Pegawai p = new Pegawai();  
            p.setNip(txtNip.getText());  
            p.setNama_Pegawai(txtNamaPegawai.getText());  
            p.setGaji_Pokok(txtGajiPokok.getText());  
            p.setGolongan((String) CbGolongan.getSelectedItem());  
            p.setJam_Kerja(txtJamKerja.getText());  
            p.setLembur(txtLembur.getText());  
            interfacePegawai.insert(p);  
            JOptionPane.showMessageDialog(this, "Data tersimpan");  
            //bersih();  
            this.setVisible(false);  
        }  
        int input = Integer.parseInt(PindahKesebelah);  
  
        if (i != input) {  
            i++;  
            this.setVisible(true);  
            txtNip.setText("");  
            txtGajiPokok.setText("");  
            txtNamaPegawai.setText("");  
            txtJamKerja.setText("");  
            txtLembur.setText("");  
            CbGolongan.setSelectedIndex(0);  
        } else {  
            Menu m = new Menu();  
            m.setVisible(true);  
        }  
    } catch (SQLException ex) {  
        JOptionPane.showMessageDialog(this, ex);  
    }  
}
```

Gambar 14. Action Button Simpan

Pada Gambar 11 selain terdapat statement untuk mengambil data yang telah diinputkan, juga terdapat sebuah kondisi yang dimaksudkan untuk melakukan perulangan / meminta penginputan sebanyak jumlah pegawai yang telah dimasukkan sebelumnya. Tak lupa juga ketika kita sudah menyimpan 1 data dan masih ada data lagi yang ingin diinputkan maka kolom inputan akan dikosongkan terlebih dahulu. Ketika telah mencapai batas jumlah pegawai, maka akan otomatis keluar dari form input data pegawai dan beralih kembali ke tampilan menu awal aplikasi.

6. Class Data Pegawai

Class ini digunakan untuk melihat data pegawai yang telah diinputkan sebelumnya. Terdapat 3 button pada tampilan ini yaitu :

a. Button Edit

Tombol ini digunakan untuk mengubah data yang kita inginkan baik itu NIP, Nama Pegawai, Gaji Pokok, Golongan, Jam Kerja dan Lembur. Untuk Action dari button ini dapat dilihat pada gambar dibawah :

```
private void EditActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        Pegawai pgw = new Pegawai();
        pgw.setNip(txtNip.getText().trim());
        pgw.setNama_Pegawai(txtNamaPegawai.getText().trim());
        pgw.setGaji_Pokok(txtGajiPokok.getText().trim());
        pgw.setGolongan(String.valueOf(CbGolongan.getSelectedItem()).trim());
        pgw.setJam_Kerja(txtJamKerja.getText().trim());
        pgw.setLembur(txtLembur.getText().trim());
        interfacePegawai.update(pgw, nipAwal);
        JOptionPane.showMessageDialog(this, "data berhasil diubah");
        ambil_data();
    } catch (SQLException ex) {
        System.out.println(ex.getMessage());
        Logger.getLogger(DataPegawai.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

Gambar 15. Action Button Edit

b. Button Hapus

Tombol ini digunakan untuk menghapus data yang diinginkan. Untuk Action dari button ini dapat dilihat pada gambar dibawah :

```
private void HapusActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        String Nip = txtNip.getText();
        interfacePegawai.delete(Nip);
        ambil_data();
        txtNip.setText("");
        txtGajiPokok.setText("");
        txtNamaPegawai.setText("");
        txtJamKerja.setText("");
        txtLembur.setText("");
        CbGolongan.setSelectedIndex(0);
        JOptionPane.showMessageDialog(this, "data berhasil dihapus");
    } catch (SQLException ex) {
        Logger.getLogger(DataPegawai.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

Gambar 16. Action Button Hapus

c. Button Kembali

Tombol ini digunakan untuk kembali ke menu awal aplikasi dan tampilan data pegawai akan hilang. Untuk action dari button ini dapat dilihat pada gambar dibawah :

```
private void KembaliActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    Menu m = new Menu();  
    m.setVisible(true);  
    this.setVisible(false);  
}
```

Gambar 17. Action Button Kembali

7. Class Hitung Gaji Pegawai

Class ini digunakan untuk menghitung gaji bersih dan total gaji pegawai. Pada tampilan ini terdapat 4 button yaitu :

a. Button Lihat Data Pegawai

Tombol ini digunakan untuk melihat data-data pegawai yang telah diinputkan agar kita dapat mengetahui NIP dari pegawai yang ingin kita hitung gajinya. Untuk action dari button ini dapat dilihat pada gambar dibawah :

```
private void LihatActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    DataPegawai2 dp = new DataPegawai2();  
    dp.setVisible(true);  
    this.setVisible(false);  
}
```

Gambar 18. Action Button Lihat Data Pegawai

b. Button Hitung

Tombol ini digunakan untuk menghitung gaji bersih dan total gaji pegawai. untuk action pada button ini dapat dilihat pada gambar dibawah :

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

Gambar 19. Action Button Hitung

Pada Gambar 19 terdapat pemanggilan method itung data yang dimana berisikan proses perhitungan gaji bersih dan total gaji pegawai tersebut. Didalamnya juga terdapat statement untuk pengonversian tipe data int dan terdapat sebuah kondisi yang digunakan untuk menentukan gaji tetap dan upah lembur per jam berdasarkan berdasarkan golongan pegawai. Untuk statement iniNamaPegawai sampai iniTotalGaji pada Gambar 20 digunakan untuk

menampilkan nama pegawai, gaji bersih dan total gaji berdasarkan hasil perhitungan yang telah dilakukan.

```

public void itungdata(){
    int lemburreal = Integer.parseInt(lembur);
    int gajipokokreal = Integer.parseInt(gaji_pokok);
    int golonganreal = Integer.parseInt(golongan);
    double tunjangan = (gajipokokreal * 0.15);
    double pajak = (0.1 * (gajipokokreal + tunjangan));
    double gajiBersih = (gajipokokreal + tunjangan - pajak);
    int totalGaji = 0;
    if(golonganreal == 1){
        totalGaji = (500000 + (5000 * lemburreal));
    }else if(golonganreal == 2){
        totalGaji = (300000 + (3000 * lemburreal));
    }else if(golonganreal == 3){
        totalGaji = (250000 + (2000 * lemburreal));
    }else if(golonganreal == 4){
        totalGaji = (100000 + (1500 * lemburreal));
    }else if(golonganreal == 5){
        totalGaji = (50000 + (1000 * lemburreal));
    }

    iniNamaPegawai.setText(namaPegawai);
    iniGajiBersih.setText(String.valueOf(gajiBersih));
    iniTotalGaji.setText(String.valueOf(totalGaji));
}

```

Gambar 20. Method Itungdata

c. Button Reset

Tombol ini digunakan untuk menghapus / mengosongkan seluruh isi kolom inputan atau dapat dikatakan tampilan seperti pertama kali masuk ke menu hitung data. Untuk action dari button ini dapat dilihat pada gambar dibawah :

```

private void ResetActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    CbNip.setSelectedIndex(0);
    txtGajiPokok.setText("");
    txtGolongan.setText("");
    iniNamaPegawai.setText("");
    iniGajiBersih.setText("");
    iniTotalGaji.setText("");
}

```

Gambar 21. Action Reset

d. Button Kembali

Tombol ini digunakan untuk kembali ke menu awal aplikasi dan terdapat statement *this.setVisible(false)* yang digunakan agar tampilan hitung gaji akan hilang. Untuk action dari button ini dapat dilihat pada gambar dibawah :

```

private void KembaliActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Menu m = new Menu();
    m.setVisible(true);
    this.setVisible(false);
}

```

Gambar 22. Action Kembali

BAB 4

HASIL PENGUJIAN

1. Menu Utama Aplikasi

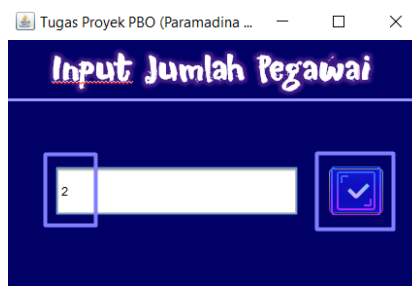
Pada saat program pertama kali dijalankan maka akan tampil menu utama aplikasi yang telah dibuat terdapat 3 pilihan menu yaitu Input Data Pegawai, Lihat Data Pegawai dan Hitung Gaji Pegawai. Pada menu tersebut juga terdapat tombol untuk keluar dari aplikasi. Berikut capture dari tampilan menu utama aplikasi :



Gambar 23. Tampilan Menu Utama Aplikasi

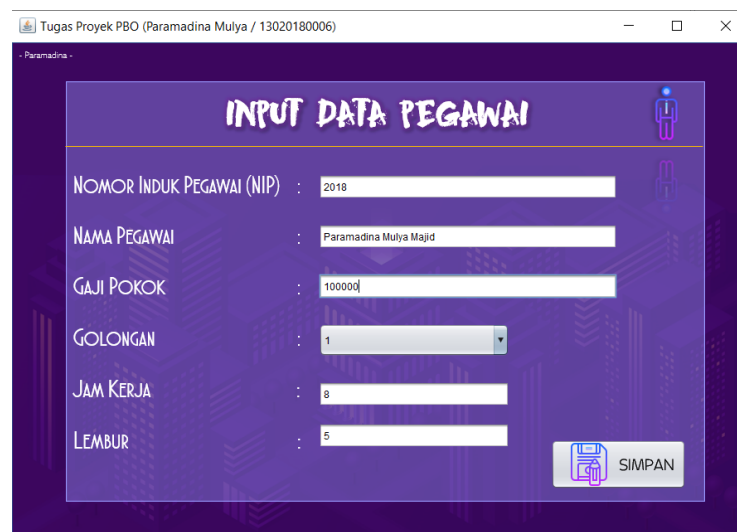
2. Pilihan Menu “Input Data Pegawai”

Ketika kita memilih menu “Input Data Pegawai” maka tampilan menu awal aplikasi akan hilang dan akan beralih ke tampilan penginputan jumlah pegawai yang ingin dimasukkan kedalam form data pegawai. pada gambar dibawah saya akan menginputkan sebanyak 2 pegawai.



Gambar 24. Tampilan Input Jumlah Pegawai

Kemudian ketika kita telah menginputkan jumlah pegawai dan menekan tombol ceklis maka tampilan penginputan jumlah pegawai akan hilang dan akan beralih ke tampilan selanjutnya yaitu form penginputan data pegawai yang terdiri atas NIP (Nomor Induk Pegawai), Nama Pegawai, Gaji Pokok, Golongan, Jam Kerja dan Lembur. Sebelumnya saya memasukkan jumlah pegawai yaitu 2 pegawai maka pada form ini akan meminta pengisian form sebanyak jumlah pegawai yang diinputkan. Ketika telah menginputkan data pegawai sebanyak jumlah pegawai yang diinputkan sebelumnya dan menekan tombol simpan maka akan tampil pemberitahuan “data telah tersimpan” dan otomatis tampilan akan beralih ke menu utama.

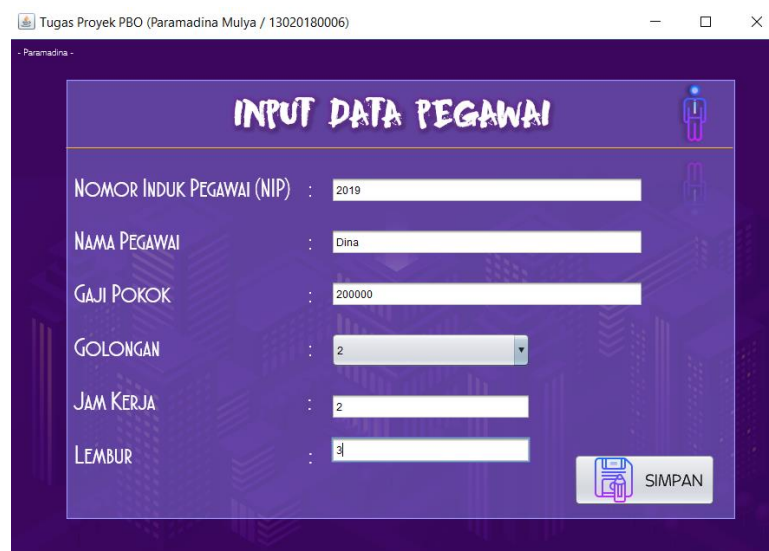


The screenshot shows a web application window titled 'Tugas Proyek PBO (Paramadina Mulya / 13020180006)'. The main form is titled 'INPUT DATA PEGAWAI' and has a purple background with a pattern. It contains the following fields and values:

Field	Value
NOMOR INDUK PEGAWAI (NIP)	2018
NAMA PEGAWAI	Paramadina Mulya Majid
GAJI POKOK	100000
GOLONGAN	1
JAM KERJA	8
LEMBUR	5

A 'SIMPAN' button with a floppy disk icon is located at the bottom right of the form.

Gambar 25. Tampilan Input Data Pegawai 1

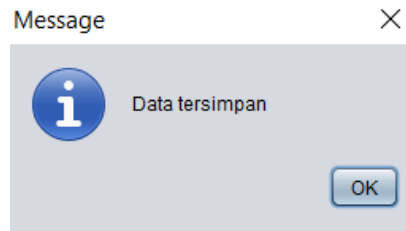


The screenshot shows the same 'INPUT DATA PEGAWAI' form, but with data for the second employee. The fields and values are:

Field	Value
NOMOR INDUK PEGAWAI (NIP)	2019
NAMA PEGAWAI	Dina
GAJI POKOK	200000
GOLONGAN	2
JAM KERJA	2
LEMBUR	3

The 'SIMPAN' button is still present at the bottom right.

Gambar 26. Tampilan Input Data Pegawai 2



Gambar 27. Tampilan Pesan Ketika Menekan Tombol Simpan

3. Pilihan Menu “Lihat Data Pegawai”

Pada saat memilih menu “Lihat Data Pegawai” maka menu utama aplikasi akan hilang dan akan beralih ke tampilan untuk melihat data pegawai yang telah diinputkan sebelumnya.

NIP	NAMA PEGAWAI	GAJI POKOK	GOLONGAN	JAM KERJA	LEMBUR
00004	Majid	800000	2	2	5
00005	Aldi	1000000	4	3	1
00006	Dina	420000	2	5	5
2018	Paramadina Mulya Majid	100000	1	8	5
2019	Dina	200000	2	2	3

Gambar 28. Tampilan Data Pegawai

. Pada tampilan tersebut terdapat tombol-tombol sebagai berikut :

a. Edit Data :

Ketika kita ingin mengubah data pegawai, terlebih dahulu menekan data pegawai yang ingin diubah maka akan muncul data pegawai pada bagian atas kemudian kita dapat mengubah data tersebut. Kali ini saya mengubah nama “Paramadina Mulya Majid” menjadi “Paramadina” dan NIP “2018” menjadi “2000”. Ketika sudah tidak ada lagi yang ingin diubah maka silahkan untuk menekan tombol “EDIT” dan akan ada pesan “Data berhasil diubah”.

Tugas Proyek PBO (Paramadina Mulya / 13020180006)

Paramadina -

DATA PEGAWAI

NOMOR INDUK PEGAWAI (NIP)

NAMA PEGAWAI

GAJI POKOK

GOLONGAN

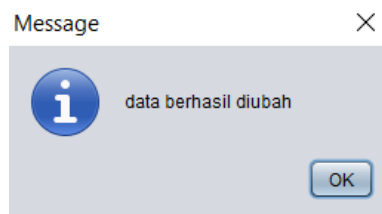
JAM KERJA

LEMBUR

NIP	NAMA PEGAWAI	GAJI POKOK	GOLONGAN	JAM KERJA	LEMBUR
00004	Majid	800000	2	2	5
00005	Aldi	1000000	4	3	1
00006	Dina	420000	2	5	5
2018	Paramadina Mulya Majid	100000	1	8	5
2019	Dina	200000	2	2	3

EDIT
 HAPUS
 KEMBALI

Gambar 29. Tampilan Mengubah Data Pegawai



Gambar 30. Tampilan Pesan Ketika Menekan Tombol Edit

Tugas Proyek PBO (Paramadina Mulya / 13020180006)

Paramadina -

DATA PEGAWAI

NOMOR INDUK PEGAWAI (NIP)

NAMA PEGAWAI

GAJI POKOK

GOLONGAN

JAM KERJA

LEMBUR

NIP	NAMA PEGAWAI	GAJI POKOK	GOLONGAN	JAM KERJA	LEMBUR
00004	Majid	800000	2	2	5
00005	Aldi	1000000	4	3	1
00006	Dina	420000	2	5	5
2000	Paramadina	100000	1	8	5
2019	Dina	200000	2	2	3

EDIT
 HAPUS
 KEMBALI

Gambar 31. Tampilan Perubahan yang Terjadi Setelah Mengedit Data Pegawai

b. Hapus Data

Ketika kita ingin menghapus data pegawai, maka terlebih dahulu kita memilih data yang ingin dihapus kemudian menekan tombol “HAPUS”. Ketika telah menekan tombol hapus maka akan tampil pesan “data telah dihapus”.

The screenshot shows a web application window titled 'Tugas Proyek PBO (Paramadina Mulya / 13020180006)'. The main content area is titled 'DATA PEGAWAI' and features a form with the following fields:

- NOMOR INDUK PEGAWAI (NIP): 2019
- NAMA PEGAWAI: Dina
- GAJI POKOK: 200000
- GOLONGAN: 2 (dropdown menu)
- JAM KERJA: 2
- LEMBUR: 3

Below the form is a table with the following data:

NIP	NAMA PEGAWAI	GAJI POKOK	GOLONGAN	JAM KERJA	LEMBUR
00004	Majid	800000	2	2	5
00005	Aldi	1000000	4	3	1
00006	Dina	420000	2	5	5
2000	Paramadina	100000	1	8	5
2019	Dina	200000	2	2	3

Gambar 32. Tampilan Pemilihan Data Pegawai yang Ingin Dihapus

The screenshot shows the same 'DATA PEGAWAI' form, but with a 'Message' dialog box overlaid in the center. The dialog box contains an information icon and the text 'data berhasil dihapus', with an 'OK' button at the bottom right. Below the form, there are three buttons: 'EDIT' (with a pencil icon), 'HAPUS' (with a trash can icon), and 'KEMBALI' (with a left arrow icon). The table data is the same as in Gambar 32.

Gambar 33. Tampilan Pesan dan Perubahan Data Ketika Menekan Tombol Hapus

c. Kembali ke Menu Utama

Ketika kita menekan tombol ini maka tampilan data pegawai akan hilang dan akan beralih kembali ke menu utama.

4. Pilihan Menu “Hitung Gaji Pegawai”

Pada saat menekan menu “Hitung Gaji Pegawai” maka tampilan menu awal akan hilang dan akan beralih ke tampilan hitung gaji pegawai.



Gambar 34. Tampilan Hitung Gaji Bersih

Pada tampilan diatas terdapat beberapa tombol sebagai berikut :

a. Melihat Data Pegawai

Ketika kita ingin menghitung gaji pegawai, kita bisa melihat data pegawai terlebih dahulu untuk melihat NIP-nya.



NIP	NAMA PEGAWAI	GAJI POKOK	GOLONGAN	JAM KERJA	LEMBUR
00001	Bahri	100000	1	8	5
00002	Paramadina	250000	5	2	1
00003	Mulya	100000	1	1	1
00004	Majid	800000	2	2	5
00005	Aldi	1000000	4	3	1
00006	Dina	420000	2	5	5
2000	Paramadina	100000	1	8	5
2010	Dina	200000	2	2	2

Gambar 35. Tampilan Button “Tampil Data Pegawai”

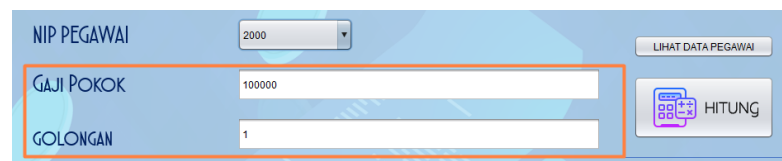
Ketika sudah melihatnya, kita dapat menekan tombol kembali maka akan kembali beralih ke tampilan hitung gaji pegawai.

b. Menghitung Gaji Bersih dan Total Gaji Pegawai

Untuk menghitung gaji bersih dan total gaji pegawai, terlebih dahulu kita akan memilih NIP pegawai yang ingin dihitung gajinya. Ketika kita telah memilih NIP pegawai maka otomatis gaji pokok dan golongan akan muncul berdasarkan data dari NIP yang dipilih.



Gambar 36. Tampilan Pemilihan NIP



Gambar 37. Tampilan Data yang Otomatis Muncul Ketika Telah Memilih NIP

Ketika kita menekan tombol “HITUNG” maka otomatis Nama Pegawai, hasil hitung Gaji Pokok dan hasil hitung Total Gaji akan tampil.



Gambar 38. Tampilan Hasil Perhitungan Gaji Bersih dan Total Gaji

Berikut proses hitungannya :

Tunjangan	: $100.000 * (15/100) = 15.000$
Pajak	: $(10/100) * (100.000 + 15.000) = 11.500$
Gaji Bersih	: $100.000 + 15.000 - 11.500 = 103.500$
Total Gaji	: $500.000 + (5 * 5000) = 525.000$

c. Reset

Ketika kita ingin menghapus isi data hitungan, maka kita dapat menekan tombol reset maka otomatis data sebelumnya sudah tidak ada / tampilan sama seperti awal pertama membuka menu “Hitung Gaji Pegawai”.

The screenshot shows a web application titled "HITUNG GAJI BERSIH". It has a light blue background with a faint watermark. The form includes the following elements:

- Inputs:** "NIP PEGAWAI" (with a dropdown menu), "GAJI POKOK", and "GOLONGAN".
- Buttons:** "LIHAT DATA PEGAWAI", "HITUNG" (with a calculator icon), "RESET", and "KEMBALI" (with a circular arrow icon).
- Formulas (KETERANGAN):**
 - Tunjangan = $\text{Gaji pokok} \times 15\%$
 - Pajak = $10\% \times (\text{Gaji Pokok} + \text{Tunjangan})$
 - Gaji Bersih = $\text{Gaji Pokok} + \text{Tunjangan} - \text{Pajak}$
 - Total Gaji = $\text{Gaji Tetap Sesuai Golongan} + (\text{Lembur} \times \text{Tarif Lembur})$
- Outputs:** "NAMA PEGAWAI", "GAJI BERSIH", and "TOTAL GAJI" with corresponding input fields.

Gambar 39. Tampilan Setelah di Reset

d. Kembali ke Menu Utama

Ketika kita menekan tombol ini maka tampilan data pegawai akan hilang dan akan beralih kembali ke menu utama.

5. Button Keluar

Pada tombol ini digunakan apabila kita ingin keluar dari aplikasi. Pada tombol tersebut memiliki action yang didalamnya terdapat statement *System.exit(0)* yang dimaksudkan apabila kita menekan tombol tersebut maka aplikasi akan secara otomatis keluar.

BAB 5

PENUTUP

A. Kesimpulan

Dari hasil pembuatan aplikasi perhitungan gaji bersih dan total gaji untuk pegawai CV.XXXI dapat ditarik kesimpulan sebagai berikut :

1. Aplikasi telah mampu melakukan penyimpanan dan melihat data pegawai.
2. Aplikasi telah mampu melakukan proses pengubahan dan penghapusan data pegawai.
3. Aplikasi telah mampu melakukan proses perhitungan gaji bersih dan total gaji pegawai berdasarkan ketentuan yang ada
4. Aplikasi berhasil menerapkan konsep Pemrograman Berorientasi Objek seperti enkapsulasi, inheritance dan polimorphism.

B. Saran

Berdasarkan hasil pembuatan aplikasi ini terdapat saran untuk pengembangan selanjutnya yaitu diharapkan aplikasi ini dapat menerapkan lebih banyak fitur lainnya seperti pembuatan history/laporan perhitungan gaji bersih dan total gaji pegawai secara otomatis agar dapat memudahkan perusahaan dalam melakukan pelaporan penggajian pegawai.

DAFTAR PUSTAKA

Bim. 2013. membuat perintah exit program pada java. <https://ambilgratis.com/2013/08/28/membuat-perintah-exit-program-pada-java/>. Diakses pada tanggal 19 Juni 2020 (Online).

JARDIKNAS. 2020. Encapsulation, Inheritance, polymorphism, dan interface