

BAB IV

IMPLEMENTASI SISTEM

Pada tahap implementasi ini merupakan penerapan aplikasi dari hasil perancangan sistem yang sudah dibuat untuk mencapai suatu tujuan yang diinginkan. Implementasi melaksanakan perintah-perintah *coding* yang secara terstruktur dari awal sampai akhir. Berdasarkan rancangan yang telah diuraikan pada bab dan sebelumnya, maka dalam bab ini akan membahas hasil implementasi *database* dan implementasi *interface* (antarmuka) :

4.1. Implementasi Database

4.1.1. Tampilan Database Login

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	username	varchar(10)			No	None	
2	password	varchar(8)			No	None	

Gambar 4.1. Tampilan Database Login

Pada tampilan *database login*, terdapat dua atribut yaitu *username* sebagai *Primary Key* dan *Password*.

4.1.2. Tampilan Database Pegawai

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id	varchar(5)			No	None	
2	bulan	text			No	None	
3	namapegawai	text			No	None	
4	gajipokok	int(8)			No	None	
5	tunjangan	int(8)			No	None	
6	potongan	int(8)			No	None	
7	hutang	mediumint(8)			No	None	
8	target	int(8)			No	None	

Gambar 4.2. Tampilan Database Pegawai

Pada tampilan *database* pegawai, terdapat 8 atribut yaitu id sebagai *primary key*, bulan, namapegawai, gajipokok, tunjangan, potongan, hutang, dan target.

4.1.3. Tampilan *Database* Simulasi

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id	varchar(5)			No	None	
2	namapegawai	text			No	None	
3	bulan	varchar(12)			No	None	
4	bobot	varchar(11)			No	None	
5	nilai	double			No	None	
6	nilai_y	double			No	None	
7	max	double			No	None	
8	min	double			No	None	

Gambar 4.3. Tampilan *Database* Simulasi

Pada tampilan *database* simulasi, terdapat 8 atribut yaitu id sebagai *foreign key*, namapegawai, bulan, bulan, bobot, nilai, nilai_y, *max* dan *min*.

4.1.4. Tampilan *Database* Pengujian

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id	varchar(5)			No	None	
2	namapegawai	varchar(20)			No	None	
3	bulan	varchar(10)			No	None	
4	target	varchar(20)			No	None	
5	output	double			No	None	
6	selisih	double			No	None	
7	akurasi	double			No	None	
8	warning	longtext			No	None	

Gambar 4.4. Tampilan *Database* Pengujian

Pada tampilan *database* pengujian, terdapat 8 atribut yaitu *id* sebagai *primary key*, *namapegawai*, *bulan*, *target*, *output*, *selisih*, *akurasi* dan *warning*.

4.2. Implementasi *Interface*

4.2.1. Implementasi *Form Login*



Gambar 4.5. Implementasi *Form Login*

Sebelum pengguna masuk ke menu utama sistem, pengguna perlu melakukan *login* dengan mengisi *Username* dan *Password*, jika benar maka akan tampil menu utama sistem. Jika tidak berhasil *login* maka akan muncul *message box* yang menyatakan *login* gagal. Pengguna harus mengisi ulang *Username* dan *Password* untuk bisa masuk ke menu utama sistem.

```

private void masukan (){
    try {
        String nama = fuser.getText();
        String pass = new String(fpass.getPassword());
        asd.ss = asd.cc.createStatement();
        String sql = "Select * From loginn Where username = '"+nama+"'
And password = '"+pass+"'";
        asd.rr=asd.ss.executeQuery(sql);
        if (asd.rr.next()){

if(fpass.getText().equals(asd.rr.getString("Password"))){
    JOptionPane.showMessageDialog(rootPane, "Login Anda
Berhasil");
        new MenuUtama().show();
        this.dispose();
    }
    else {
        fpass.setText("");
        fpass.requestFocus();
    }
}
else {
    JOptionPane.showMessageDialog(rootPane, "Password salah,
silahkan coba lagi");
}
} catch (SQLException e) {
    JOptionPane.showMessageDialog(null, e);
}
}

```

4.2.2. Impelementasi Menu Utama



Gambar 4.6. Implementasi Menu Utama

Dalam tampilan menu utama, terdapat beberapa item yaitu Master yang berisi *Form* Pegawai sebagai inputan, JST yang berisi Simulasi dan pengujian *Backpropagation* untuk prosesnya dan Keluar untuk keluar dari Menu Utama.

4.2.3. Implementasi *Form* Pegawai

The screenshot shows a Java Swing window titled "FORM PEGAWAI". At the top, there is a table displaying employee records. Below the table, on the left, is a section labeled "Isi Data" containing several text input fields for: ID, Bulan, Nama Pegawai, Gaji Pokok, Tunjangan, Potongan, Hutang, and Call Call target. On the right side of the window, there is a section labeled "Tombol" containing five buttons: "Simpan", "Hapus", "Baru", "Edit", and "Keluar".

ID	Bulan	Nama Pegawai	Gaji Pokok	Tunjangan	Potongan	Hutang	target
1	Januari	Isak	200000	50000	50000	100000	100000
2	Februari	Isak	200000	50000	50000	100000	100000
3	Maret	Katrina	200000	50000	50000	100000	100000
4	April	Katrina	200000	50000	50000	100000	100000
5	Mei	Katrina	200000	50000	50000	100000	100000
6	Juni	Katrina	200000	50000	50000	100000	100000
7	Juli	Katrina	200000	50000	50000	100000	100000
8	Agustus	Katrina	200000	50000	50000	100000	100000
9	September	Katrina	200000	50000	50000	100000	100000
10	Oktober	Katrina	200000	50000	50000	100000	100000
11	November	Katrina	200000	50000	50000	100000	100000
12	Desember	Katrina	200000	50000	50000	100000	100000

Gambar 4.7. Implementasi *Form* Pegawai

Pada tampilan implementasi *form* pegawai, *User* melakukan inputan data pegawai perbulannya untuk melakukan proses pengambilan kredit. Data diisi pada *panel* Isi data dan akan di proses melalui tombol yang ada seperti Simpan, Hapus, Baru, Edit dan Keluar. Kemudian data pegawai yang sudah disimpan akan masuk ke dalam *database* yang sudah dibuat.

```
private void simpan(){
    String id = this.jTextField1.getText();
    String bulan = this.jTextField2.getText();
    String namapegawai = this.jTextField3.getText();
    String gaji pokok = this.jTextField5.getText();
    String tunjangan = this.jTextField6.getText();
    String potongan = this.jTextField7.getText();
    String hutang = this.jTextField8.getText();
    String target = this.jTextField9.getText();
    try {
        String sql= "Insert into datapegawai values(?,?,?,?,?,?,?,?)";
        PreparedStatement p =(PreparedStatement) con.cc.prepareStatement(sql);
        p.setString(1, id);
        p.setString(2, bulan);
        p.setString(3, namapegawai);
        p.setString(4, gaji pokok);
        p.setString(5, tunjangan);
        p.setString(6, potongan);
        p.setString(7, hutang);
        p.setString(8, target);
        p.executeUpdate();
        BacaTabel();
        JOptionPane.showMessageDialog(this,"Data Sukses Disimpan");
    } catch (Exception e) {
        System.out.println();
    }
}
```

```

private void edit(){
    String id = this.jTextField1.getText();
    String bulan = this.jTextField2.getText();
    String namapegawai = this.jTextField3.getText();
    String gajipokok = this.jTextField5.getText();
    String tunjangan = this.jTextField6.getText();
    String potongan = this.jTextField7.getText();
    String hutang = this.jTextField8.getText();
    String target = this.jTextField9.getText();
    try {
        String sql= "Update datapegawai Set bulan=?, namapegawai=?, gajipokok=?,
tunjangan=?, potongan=?, hutang=?, target=? Where id=?";
        PreparedStatement p =(PreparedStatement) con.cc.prepareStatement(sql);
        p.setString(8, id);
        p.setString(1, bulan);
        p.setString(2, namapegawai);
        p.setString(3, gajipokok);
        p.setString(4, tunjangan);
        p.setString(5 , potongan);
        p.setString(6, hutang);
        p.setString(7, target);
        p.executeUpdate();
        BacaTabel();
        JOptionPane.showMessageDialog(this,"Data Sukses Diubah");
    } catch (Exception e) {
        System.out.println();
    } }

```

```

private void hapus(){
    try {
        String sql= "Delete from datapegawai Where id='"+jTextField1.getText()+"'";
        con.ss.executeUpdate(sql);
        con.ss.close();
        JOptionPane.showMessageDialog(null,"Apakah data akan dihapus?");
        BacaTabel();
        JOptionPane.showMessageDialog(null,"Data Berhasil Dihapus");
        jTextField1.requestFocus();
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
}
}
}

```

```

private void baru(){
    jTextField1.setText("");
    jTextField2.setText("");
    jTextField3.setText("");
    jTextField5.setText("");
    jTextField6.setText("");
    jTextField7.setText("");
    jTextField8.setText("");
    jTextField9.setText("");
}

```

[illegible]

Pada tahap *form* simulasi, setelah data pegawai dimasukan pada *form* pegawai, data sebelumnya dinormalisasikan, selanjutnya data setiap pegawai tersebut disimulasi per satu bulan dan diproses dengan mengklik tombol proses dan setelah itu data disimpan dengan mengklik tombol simpan. Data akan otomatis tersimpan pada *database*.

```

public void backpropagation()
{
    isialisasi_bobot();
    while (epoh < max_epoh)
    {
        System.out.println();
        System.out.println("Iterasi Ke = " + epoh);
        hidden_layer=0;
        for (i=1; i <= 12 ;i++)
        {
            for (j=1; j <= 4 ;j++)
            {
                hidden_layer= (v[j][i] * x[i][j]) + hidden_layer;
            }
            z_in[i]=v0[i]+hidden_layer;
            System.out.println("Nilai Z_in " + i + " = " + z_in[i]);
            hidden_layer= 0;
        }
        epoh++;
        System.out.println();
        y_ink=0;
        for (i=1; i<=12;i++)
        {
            z[i] = 1/(1+ (double)Math.exp(-z_in[i]));
            System.out.println("Nilai z "+ i + " = " + z[i]);
            y_ink = y_ink + (w[i]*z[i]);
        }
        y_ink = w0 + y_ink;
        System.out.println("Nilai y_ink = " + y_ink);
        y = 1/(1+ (double)Math.exp(-y_ink));
        System.out.println("Nilai y = " + y);
        error = y-target;
        System.out.println("Nilai error = " + error);
        kuadrat_error= Math.pow(error,2);
        System.out.println("Nilai Kuadrat Error = " + kuadrat_error);
        delta= ( (target-y) * (y) * (1-y) ) ;
        System.out.println("Nilai Delta = " + delta);
        for (i=1; i<=12;i++)
        {
            delta_w[i]=alfa * delta * z[i];
            System.out.println("Nilai delta w "+i+" = " + delta_w[i]);
            delta_in[i]=delta*w[i];
            System.out.println("Nilai delta_in "+i+" = " + delta_in[i]);
            delta_error[i]= ( delta_in[i] * (z[i]) * (1- z[i]) );
            System.out.println("Nilai delta_error "+i+" = " + delta_error[i]);
        }
    }
}

```

```

        bobot="w0";
        nilai=Double.toString(w0);
        try {
            String sql= "Insert into pelatihan values(?,?,?,?,?,?,?)";
            PreparedStatement p =(PreparedStatement) con.cc.prepareStatement(sql);
            p.setString(1, id_p);
            p.setString(2, namapegawai);
            p.setString(3, b);
            p.setString(4, bobot);
            p.setString(5, nilai);
            p.setString(6, Double.toString(y));
            p.setString(7, Double.toString(max));
            p.setString(8, Double.toString(min));
            p.executeUpdate();
            //JOptionPane.showMessageDialog(this, "Data Sukses Disimpan");
        } catch (Exception e) {
            System.out.println();
        }
        JOptionPane.showMessageDialog(this, "Data Sukses Disimpan");
    }
}

```

4.2.5 Implementasi *Form* Pengujian

ID	Nama Pegawai	Bulan	Target	Output	Salah	Akurasi (%)	Warning
48	Anni B. R.	Januari	327383	323506.1598462683	223.15984626830323	99.9337760353591	Warning : Tidak ada ...
48	Anni B. R.	Juli	550483	612152.41250118956	6.609271250118956	88.7917524115.5368	Warning : Tidak ada ...

Gambar 4.9. Implementasi *Form* Pengujian

Pada tahap pengujian, data pegawai yang sudah disimulasi selanjutnya diuji. Data yang diuji akan diproses dengan mengklik tombol proses dan akan menghasilkan data peringatan/*warning* dalam melakukan pengambilan kredit. Setelah itu, data disimpan di dalam *database* dan akan menampilkan data laporannya.

```

private void panggil_data_uji()
{
    try {
        Object header[]={ "id", "Nama
Pegawai", "Bulan", "Target", "Output", "Selisih", "Akurasi(%)", "Warning"};
        DefaultTableModel data = new DefaultTableModel(null, header);
        jTable3.setModel(data);
        con.ss = con.cc.createStatement();
        String sql = "Select * From pengujian";
        con.rr= con.ss.executeQuery(sql);
        con.rr.beforeFirst();
        while (con.rr.next()){
            String c8= con.rr.getString("id");
            String c1 = con.rr.getString("namapegawai");
            String c2= con.rr.getString("bulan");
            String c3= con.rr.getString("target");
            String c4= con.rr.getString("output");
            String c5= con.rr.getString("selisih");
            String c6= con.rr.getString("akurasi");
            String c7= con.rr.getString("warning");
            String c[] = {c8,c1, c2, c3, c4, c5, c6, c7};
            data.addRow(c);
        }
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, e);
    }
    this.panggil_data();
    hasil_akhir= ((max-min) * (nilaiy-0.1))/0.8) + min;
    System.out.println("Nilai Hasil Akhir = " + hasil_akhir);
    akurasi=Double.parseDouble(tt);
    nilai_akurasi=( (hasil_akhir-akurasi)/akurasi ) * 100;
    nilai_akurasi=100-(Math.abs(nilai_akurasi));
    selisih=hasil_akhir-akurasi;
    if(hasil_akhir<1500000)
    {
        pesan="Warning ! Tidak bisa melakukan kredit tambahan";
    }else
    {
        if ( (hasil_akhir>=1600000)&&(hasil_akhir<=2000000) )
        {
            pesan="Dapat melakukan kredit tambahan dengan angsuran minimal 100.000
dan maksimal 600.000";
        } else
        {
            if ( (hasil_akhir>=2000000)&&(hasil_akhir<=2500000) )
            {
                pesan="Dapat melakukan kredit tambahan dengan angsuran minimal 700.000
dan maksimal 1.000.000";
            }
            else
            {
                if ( (hasil_akhir > 2500000) )
                {
                    pesan="Dapat melakukan kredit tambahan dengan angsuran minimal
1.000.000 dan maksimal 1500.000";
                }
            }
            JOptionPane.showMessageDialog(this, "Pegawai An. " +jTextField3.getText()+"
"+pesan);
        }
    }
}

```

```

String b = String.valueOf(jComboBox1.getSelectedItem()); //jComboBox1.getSelectedItem();
String np = this.jTextField3.getText();
String target = this.jTextField5.getText();
try {
    String sql= "Insert into pengujian values(?,?,?,?,?,?,?)";
    PreparedStatement p =(PreparedStatement) con.cc.prepareStatement(sql);
    p.setString(1, np);
    p.setString(2, b);
    p.setString(3, target);
    p.setString(4, Double.toString(hasil_akhir));
    p.setString(5, Double.toString(selisih));
    p.setString(6, Double.toString(nilai_akurasi));
    p.setString(7, pesan);
    p.executeUpdate();
    JOptionPane.showMessageDialog(this, "Data Sukses Disimpan");
} catch (Exception e) {
    System.out.println();
}
}
panggil_data_uji();
}

```

 LAPORAN PENGAMBILAN KREDIT PNS PADA SMPN 4 KOTA KUPANG 							
Nama Pegawai	Bulan	Gaji Pokok (Rp.)	Tunjangan (Rp.)	Potongan (Rp.)	Hutang (Rp.)	Target (Rp.)	Warning
Anni B. R.	Juli	4,475,700	5,076,037	668,837	4,251,042	156,158	Warning ! Tidak bisa melakukan kredit tambahan

4.2.6. Implementasi Laporan Cetak Data

Gambar 4.10. Implementasi *Report* Pegawai

Pada tampilan *report* pengambilan kredit Pegawai Negeri Sipil tersebut, isinya terdiri dari atribut data pegawai dan pengujian antara lain nama pegawai, bulan, gaji pokok, tunjangan, potongan, hutang, target dan warning. *Report* berfungsi sebagai *output* dari sistem yang menjadi pegangan Bendahara dan pegawai yang bersangkutan. Laporan tersebut menggunakan format .DOC.

```
public void setLaporan(String file, String nama, String bulan){
    try{
        reportSource=System.getProperty("user.dir")+"/src/report/template/laporan_pengujian.
        jrxml";

        reportDest=System.getProperty("user.dir")+"/src/report/template/laporan_pengujian.ht
        ml";
        //      System.out.println(reportSource);
        //      System.out.println(reportDest);
        //      System.out.println(System.getProperty("user.dir"));
        Map<String,Object>params=new HashMap<String,Object>();
        params.put("datapegawai_namapegawai", nama);
        params.put("datapegawai_bulan", bulan);
        try{
            JasperReport
            jasperReport=JasperCompileManager.compileReport(reportSource);
            JasperPrint
            jasperPrint=JasperFillManager.fillReport(jasperReport, params,conn);

            JasperExportManager.exportReportToHtmlFile(jasperPrint,reportDest);
            JasperViewer.viewReport(jasperPrint,false);
        }catch(Exception e){
            System.out.println(e);
            JOptionPane.showMessageDialog(null,"Cetak File Gagal " +
            e,"Gagal",JOptionPane.ERROR_MESSAGE);
        }
        }catch(Exception e){
            JOptionPane.showMessageDialog(null,"Kesalahan Property Pada Jasper.  " +
            e,"Gagal",JOptionPane.ERROR_MESSAGE);
        }
    }
}
```