

LISTING PROGRAM

Login.java

```
public class Login extends
javax.swing.JDialog {
    int noPx;
    int yah;
    String usl,usdf = "admin", pasl =
"admin",pasdf, yhl;

    private final Sounderror
PlaySound = new Sounderror();
    /**
     * Creates new form mnAwal
     */

    public Login(java.awt.Frame
parent, boolean modal) {
        super(parent, modal);
        initComponents();
        this.setTitle("User dan
Password masuk");

this.setLocationRelativeTo(null);
    }
    private void kOsongLog()
    {
        tfUser.setText("");
        tfPass.setText("");

    }
    private void ambilUser(){
        String nmU = tfUser.getText();
        if
("".equals(tfUser.getText())) {
            PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Nama User masih kosong");
            tfUser.requestFocus();
            return;

        }

        try {
            Connection cpbar =
Libra.KonekZahwa.getkoneksi();
            Statement spb =
cpbar.createStatement();
            String sqlpbar =
"SELECT * FROM login WHERE User =
'" + nmU + "'";
            ResultSet rsambarpes =
spb.executeQuery(sqlpbar);

            if (rsambarpes.next())
            {
                usl =
rsambarpes.getString("User");
                pasl =
rsambarpes.getString("Pass");

                yhl =
rsambarpes.getString("Hak_Akses");

tfPass.requestFocus();
            }
            else
            {
                PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Nama User "+tfUser.getText()+"
Tidak
ada!!!", "INFORMASI",JOptionPane.IN
FORMATION_MESSAGE);
                tfUser.setText("");

tfUser.requestFocus();
                kOsongLog();
            }
        }
        catch(SQLException ePb)
        {
            System.err.println("Error data
tidak bisa diAmbil!!!");
        }
    }

    private void
tfPassKeyPressed(java.awt.event.Ke
yEvent evt) {
        // TODO add your handling
code here:
        String usr =
tfUser.getText();
        String pas =
tfPass.getText();

        if(tfUser.getText()!=null){
            ambilUser();
        }
        int
ascii=evt.getKeyCode();
        if (ascii==10)
        {
            if
("".equals(tfUser.getText())) {
                PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Nama User masih kosong");

tfUser.requestFocus();
                return;
            }
        }
    }
}
```

```

        if
        ("".equals(tfPass.getText())) {

PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Password User masih kosong");

tfPass.requestFocus();
        return;
    }
    if (usr.equals(usr) &&
pas.equals(pas1)) {

        yah =
Integer.parseInt(yhl);
        mnUtama menuUta =
new mnUtama(yah, usr);

menuUta.setAkt(yah);

menuUta.setVisible(true);
        this.dispose();
    }
    else
    if (!usr.equals(usr)) {
        PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Nama User Salah");
        tfUser.setText("");
        tfUser.requestFocus();
    }
    else
    if (!pas.equals(pas1)) {

PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Password User Salah");

tfPass.setText("");

tfPass.requestFocus();
    }
    }

    private void
btMasukActionPerformed(java.awt.ev
ent.ActionEvent evt) {
        // TODO add your handling
code here:
        String usr =
tfUser.getText();
        String pas =
tfPass.getText();

        if
        ("".equals(tfUser.getText())) {

```

```

        PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Nama User masih kosong");
        tfUser.requestFocus();
        return;
    }
    if
    ("".equals(tfPass.getText())) {
        PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Password User masih kosong");
        tfPass.requestFocus();
        return;
    }

    if (tfUser.getText() != null) {
        ambilUser();
    }
    if (usr.equals(usr) &&
pas.equals(pas1)) {

        yah =
Integer.parseInt(yhl);
        mnUtama menuUta = new
mnUtama(yah, usr);
        menuUta.setAkt(yah);

menuUta.setVisible(true);
        this.dispose();
    }
    else if (!usr.equals(usr)) {
        PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Nama User Salah");
        tfUser.setText("");
        tfUser.requestFocus();
    }
    else
    if (!pas.equals(pas1)) {
        PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Password User Salah");
        tfPass.setText("");
        tfPass.requestFocus();
    }
    }

    private void
tfUserKeyPressed(java.awt.event.Ke
yEvent evt) {
        // TODO add your handling
code here:
        int
ascii=evt.getKeyCode();
        if (ascii==10)
        {

```

```

        String nmU =
tfUser.getText();
        if
("".equals(tfUser.getText())) {

PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Nama User masih kosong");

tfUser.requestFocus();
        return;
    }

    try {
        Connection cpbar =
        Libra.KonekZahwa.getkoneksi();
        Statement spb =
        cpbar.createStatement();
        String sqlpbar =
        "SELECT * FROM login WHERE User =
        '" + nmU + "'";
        ResultSet
        rsambarpes =
        spb.executeQuery(sqlpbar);

        if
        (rsambarpes.next()) {
            usl =
            rsambarpes.getString("User");
            pas1 =
            rsambarpes.getString("Pass");
            yhl =
            rsambarpes.getString("Hak_Akses");

            tfPass.requestFocus();
        }
        else
        {

PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Nama User "+tfUser.getText()+ "
Tidak
ada!!!", "INFORMASI", JOptionPane.IN
FORMATION_MESSAGE);

tfUser.setText("");

tfUser.requestFocus();
            kOsongLog();
        }
    }
    catch (SQLException
ePb)
    {

System.err.println("Error data
tidak bisa diAmbil!!!");
    }
}

```

```

    }

    private void
btBatalActionPerformed(java.awt.ev
ent.ActionEvent evt) {
        // TODO add your handling
        code here:
        System.exit(0);
    }

```

UserPass.java

```

public class UserPass extends
javax.swing.JDialog {
    private final Sounderror PlaySound
    = new Sounderror();
    String [] hak = {"admin","kasir"};
    String [] hak2 = {"kasir"};
    int noid, no, pilhak, hk;
    String usl,pas1 = "admin", yhl;
    String awal,awal2,akhir,akhir2;
    String ussr;
    StringBuffer
    pas,pas2,pas3,pas4,pas5,pas6;
    ImageIcon gb;
    int cen = 0;
    Object[] o ;

    /**
     * Creates new form UserPass
     */
    public UserPass(java.awt.Frame
parent, boolean modal) {
        super(parent, modal);
        initComponents();
        ldKar();
        loadHak();
        kOsongkanU();
        akU1(false);
        this.setTitle("Data User
Zahwa");
        this.setLocation(600,
250);
        pas3 = new
StringBuffer("");
        pas6 = new
StringBuffer("");
    }
    private void kOsongkanU()
    {
        tfUserbaru.setText("");
        tfPassword.setText("");
        tfPassword2.setText("");
        cbKaryawan.setSelectedIndex(-
1);
        cbHak.setSelectedIndex(-1);
        lbCentang.setVisible(false);
    }
    private void akU1(boolean k)

```

```

{
    btEdit.setEnabled(k);
    btHapus.setEnabled(k);
    btSimpan.setEnabled(k);
    tfUserbaru.setEnabled(k);
    tfPassword.setEnabled(k);
    tfPassword2.setEnabled(k);
    cbKaryawan.setEnabled(k);
    cbHak.setEnabled(k);
}
private void akU2(boolean z)
{
    btSimpan.setEnabled(z);
    tfUserbaru.setEnabled(z);
    cbKaryawan.setEnabled(z);
    cbHak.setEnabled(z);
}
private void akU3(boolean y)
{
    btTambah.setEnabled(y);
    btEdit.setEnabled(y);
    btHapus.setEnabled(y);
    tfUserbaru.setEnabled(y);
    tfPassword.setEnabled(y);
    cbKaryawan.setEnabled(y);
    cbHak.setEnabled(y);
}
private void loadNo() {
    try {
        Connection conlg =
        Libra.KonekZahwa.getkoneksi();
        Statement slg =
        conlg.createStatement();
        String sqllg = "select No
as nousr from login ORDER by No
desc ";
        ResultSet rslg =
        slg.executeQuery(sqllg);
        if(rslg.next())
        {
            String noPs =
            rslg.getString("nousr");
            int noPsx =
            Integer.parseInt(noPs)+1;
            no = noPsx;
        }
    } catch (SQLException ex) {

        System.out.println("Terjadi error
di ambil no login");
    }
}
private void loadDataUser()
{
    if
    ("".equals(tfUserbaru.getText()))
    {
        PlaySound.PlaySound();

        JOptionPane.showMessageDialog(this
, "Nama User mohon diisi");
    }
}

```

```

tfUserbaru.requestFocus();
return;
}
ussr = tfUserbaru.getText();
o = new Object[1];
try
{
    Connection con =
    Libra.KonekZahwa.getkoneksi();
    Statement sj =
    con.createStatement();
    String sql = "select
User from login ";
    ResultSet rsj =
    sj.executeQuery(sql);
    while(rsj.next())
    {
        o[0] =
        rsj.getString("User");

        if(ussr.equals(o[0])){

            JOptionPane.showMessageDialog(this
, "Sudah Ada !", "Warning !",
JOptionPane.WARNING_MESSAGE);

            tfUserbaru.requestFocus();
            tfUserbaru.setText("");
            return;
        }
        else

            tfPassword.setEnabled(true);

            tfPassword.requestFocus();
        }
        rsj.close();
        sj.close();
    }
    catch(SQLException e)
    {

        System.out.println("Terjadi error
di load data login");
    }
}

private void ldKar()
{
    cbKaryawan.removeAllItems();
    try
    {
        Connection cKpb =
        Libra.KonekZahwa.getkoneksi();
        Statement skp =
        cKpb.createStatement();
        String sqKp = "select *
from Karyawan ";
        ResultSet rsk =
        skp.executeQuery(sqKp);
    }
}

```

```

        while(rsk.next())
        {
            Object[] oK = new
Object[2];
            oK[0] =
rsk.getString("Kd_Karyawan");
            oK[1] =
rsk.getString("Nama");

cbKaryawan.addItem("(" +oK[0]+"
"+oK[1] );
        }
        rsk.close();
        skp.close();
    }
    catch(SQLException e)
    {

System.out.println("Terjadi error
di load data Karyawan Combo");
    }
}
private void loadAdmin()
{
    try
    {
        Connection con =
Libra.KonekZahwa.getkoneksi();
        Statement sj =
con.createStatement();
        String sql ="select
No,Hak_Akses from login ";
        ResultSet rsj =
sj.executeQuery(sql);
        if(rsj.next())
        {
            String ha =
rsj.getString("No");
            String hal =
rsj.getString("Hak_Akses");
            int iha =
Integer.parseInt(hal);
            if(iha < 1)
                pilhak = 2;

System.out.println("iha = "+iha);
        }
        else {
            pilhak = 1;
        }
        rsj.close();
        sj.close();
    }
    catch(SQLException e)
    {

System.out.println("Terjadi error
di load Hak_Akses");
    }
}

```

```

        System.out.println("pilhak =
"+pilhak);
    }
    private void loadHak(){
        cbHak.removeAllItems();
        if(pilhak == 1){
            for(int c = 0; c <
hak.length; c++){
                cbHak.addItem(hak[c]);
                cbHak.setSelectedIndex(0);
            }
        }
        if(pilhak == 2){
            for(int c = 0; c <
hak2.length; c++){
                cbHak.addItem(hak2[c]);
                cbHak.setSelectedIndex(0);
            }
        }
    }
    private void centang(){
        String ban = pas3.toString();
        String ban2= pas6.toString();
        if(ban != ban2){
            cen = 0;
        }
        if (ban.equals(ban2)){
            cen = 1;
        }
        if(cen < 1){
            btSimpan.setEnabled(false);

lbCentang.setVisible(false);
        }
        else if(cen > 0){
            btSimpan.setEnabled(true);

lbCentang.setVisible(true);
        }
    }

    private void
btHapusActionPerformed(java.awt.ev
ent.ActionEvent evt) {
        // TODO add your handling
code here:
        if
        ("".equals(tfUserbaru.getText()))
        {
            PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Silahkan pilih User yang akan
dihapus!");
            return;
        }
        String nmJh =
tfUserbaru.getText();
        int jwb =
JOptionPane.showConfirmDialog(this

```

```

, "Data "+nmJh+" akan di hapus ?\n
Ya / Tidak", "Konfirmasi",

JOptionPane.YES_NO_OPTION);
    switch (jwb)
    {
        case 0 :
JOptionPane.showMessageDialog(this
, "Jawab : Yes");
        try
        {
            Connection cJh =
Libra.KonekZahwa.getkoneksi();
            String sqlJh =
"DELETE FROM login WHERE User =
?";

            PreparedStatement
pJh = cJh.prepareStatement(sqlJh);
            pJh.setString(1 ,
nmJh);

pJh.executeUpdate();
            pJh.close();
            cJh.close();

JOptionPane.showMessageDialog(this
, "Data "+nmJh+ " Telah DiHapus
!!!");
        }
        catch (SQLException
eJh)
        {

System.err.println("Error tidak
bisa diHapus !!!");
        }
        finally
        {
            kOsongkanU();
        }
        break;
        case 1 :
JOptionPane.showMessageDialog(this
, "Jawab : No");
        break;

default:JOptionPane.showMessagDia
log(this, "Kok gak jawab Bro
!!!");
    }
}

private void
btSimpanActionPerformed(java.awt.e
vent.ActionEvent evt) {
    // TODO add your handling
code here:

    if
("".equals(tfUserbaru.getText()))
    {

```

```

        PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Nama User mohon diisi");

tfUserbaru.requestFocus();
        return;
    }
    if
("".equals(tfPassword.getText()))
    {
        PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Password mohon diisi");

tfPassword.requestFocus();
        return;
    }
    if
(cbKaryawan.getSelectedIndex() == -
1)
    {
        PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Kode karyawan mohon dipilih");

cbKaryawan.requestFocus();
        return;
    }
    if
(cbHak.getSelectedIndex() == -1)
    {
        PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Hak Akses mohon dipilih");
        cbHak.requestFocus();
        return;
    }
    String nou =
String.valueOf(no);
    String ub =
tfUserbaru.getText();
    String pas = akhir2;
    String kkd =
cbKaryawan.getSelectedItem().toStr
ing();
    String kkd1 =
kkd.substring(1, 5);
    if (cbHak.getSelectedItem()
== "admin") {
        hk = 0;
    }
    if (cbHak.getSelectedItem()
== "kasir") {
        hk = 1;
    }
    try
    {

```

```

        Connection cinJ =
        Libra.KonekZahwa.getkoneksi();
        String sqlJ = "INSERT
        INTO login set No = ?, User = ?,
        Pass = ?, Kd_Karyawan = ?,
        Hak_Akses = ?";
        PreparedStatement pJ =
        cinJ.prepareStatement(sqlJ);
        pJ.setString(1,nou);
        pJ.setString(2,ub);
        pJ.setString(3,pas);
        pJ.setString(4, kkd1);
        pJ.setString(5,
        String.valueOf(hk));
        pJ.executeUpdate();
        pJ.close();

        JOptionPane.showMessageDialog(this
        , "Data Telah DiSimpan");
    }
    catch(SQLException ex)
    {

        System.out.println("Proses Tambah
        Error");
    }
    finally
    {
        kOsongkanU();
        awal = "";
        awal2 = "";
        akU1(false);
        lbInfo.setText("");
    }

    }

    private void
    btEditActionPerformed(java.awt.eve
    nt.ActionEvent evt) {
        // TODO add your handling
        code here:
        String nmu =
        tfUserbaru.getText();
        String pu =
        tfPassword.getText();
        String ku =
        cbKaryawan.getSelectedItem().toStr
        ing();
        String kul =
        ku.substring(1, 5);
        int hku =
        cbHak.getSelectedIndex();
        try
        {
            Connection cJb =
            Libra.KonekZahwa.getkoneksi();
            String sqlJb = "UPDATE
            login set User = ?, Pass = ?,

```

```

            Kd_Karyawan = ? , Hak_Akses = ?
            where No = '"+no+"'";
            PreparedStatement pJb
            = cJb.prepareStatement(sqlJb);
            pJb.setString(1 ,
            nmu);
            pJb.setString(2 , pu);
            pJb.setString(3 ,
            kul);
            pJb.setString(4 ,
            String.valueOf(hku));
            pJb.executeUpdate();

            JOptionPane.showMessageDialog(this
            , "Data "+nmu+ " Telah DiUbah
            !!!");
            pJb.close();
        }
        catch(SQLException eJh)
        {

            System.err.println("Error tidak
            bisa diUbah !!!");
        }
        finally
        {
            kOsongkanU();
            akU1(false);
        }
    }

    private void
    btTambahActionPerformed(java.awt.e
    vent.ActionEvent evt) {
        // TODO add your handling
        code here:
        loadNo();
        akU1(false);
        akU2(true);
        kOsongkanU();
        loadAdmin();
        loadHak();
        awal = "";
        awal2 = "";
        tfUserbaru.requestFocus();
    }

    private void
    tfUserbaruKeyPressed(java.awt.even
    t.KeyEvent evt) {
        // TODO add your handling
        code here:
        int
        ascii=evt.getKeyCode();
        if (ascii==10)
        {
            loadDataUser();
        }
    }
}

```

```

        private void
cbKaryawanKeyPressed(java.awt.event.
KeyEvent evt) {
    // TODO add your handling
code here:
    int
ascii=evt.getKeyCode();
    if (ascii==10)
    {
        cbHak.requestFocus();
    }
}

    private void
tfPasswordKeyReleased(java.awt.event.
KeyEvent evt) {
    // TODO add your handling
code here:
    awal = tfPassword.getText();
    int e = evt.getKeyCode();
    pas = new StringBuffer(awal);
    int itu = pas.length();
    pas2 = new StringBuffer(itu);
    String [] kr = new
String[itu];
    if(e != 8){
        for (int bb = 0; bb <
pas.length(); bb++){
            kr[bb] = "#";
            pas2.append(kr[bb]);
        }
        pas.reverse();
        String aw;
        if(itu > 0){
            aw = pas.substring(0, 1);
            pas3.append(aw);
        }
    }
    if(e == 8){
        awal = "";
        pas = new
StringBuffer(awal);
        itu = pas.length();
        pas2 = new
StringBuffer(itu);
        pas3 = new
StringBuffer(itu);
        kr = new String[itu];
    }
    if(cbTampil.isSelected()){

tfPassword.setText(String.valueOf(
pas3));
    }
    else

tfPassword.setText(String.valueOf(
pas2));
    lbInfo.setText("Password
Setidaknya 6 Karakter");
    tfPassword2.setEnabled(true);

```

```

        if
("".equals(tfPassword.getText())){

tfPassword2.setEnabled(false);
        }
        if(itu > 5){

lbInfo.setForeground(Color.green);
        }
        if(itu < 6 && itu > 1){

lbInfo.setForeground(Color.yellow)
;
        }
        if(itu < 1 ){

lbInfo.setForeground(Color.red);
        }

if("".equals(tfPassword.getText()))
){
    lbInfo.setText("");
}
if(tfPassword.getText() !=
null)
    centang();
}

    private void
tfPassword2KeyReleased(java.awt.event.
KeyEvent evt) {
    // TODO add your handling
code here:
    awal2 = tfPassword2.getText();
    int e = evt.getKeyCode();
    pas4 = new
StringBuffer(awal2);
    int it = pas4.length();
    pas5 = new StringBuffer(it);
    String [] kr = new String[it];
    if(e != 8){
        for (int bb = 0; bb <
pas4.length(); bb++){
            kr[bb] = "#";
            pas5.append(kr[bb]);
        }
        pas4.reverse();
        String aw;
        if(it > 0){
            aw = pas4.substring(0, 1);
            pas6.append(aw);
        }
    }
    if(e == 8){
        awal2 = "";
        pas4 = new
StringBuffer(awal);
        it = pas4.length();
        pas5 = new
StringBuffer(it);

```



```

        pas6 = new
StringBuffer(it);
        kr = new String[it];
    }
    if(cbTampil.isSelected()){
tfPassword2.setText(String.valueOf
(pas6));
    }
    else

tfPassword2.setText(String.valueOf
(pas5));
    lbInfo.setText("Password
Setidaknya 6 Karakter");
    if(it > 5){

lbInfo.setForeground(Color.green);
    }
    if(it < 6 && it > 1){

lbInfo.setForeground(Color.yellow)
;
    }
    if(it < 1 ){

lbInfo.setForeground(Color.red);
    }

if("").equals(tfPassword2.getText()
) ){
    lbInfo.setText("");
    }
    centang();
    }

    private void
cbTampilActionPerformed(java.awt.e
vent.ActionEvent evt) {
        // TODO add your handling
code here:
        akhir = pas3.toString();
        akhir2 = pas6.toString();
        if(cbTampil.isSelected()){
            tfPassword.setText(akhir);

tfPassword2.setText(akhir2);
        }
        else {

tfPassword.setText(String.valueOf(
pas2));

tfPassword2.setText(String.valueOf
(pas5));
        }
    }
}

```

mnUtama.java

```

public class mnUtama extends
javax.swing.JFrame {
    int akt;
    String user;
    /**
     * Creates new form mnUtama
     */
    public mnUtama() {

        public mnUtama(int akt, String
user) {
            this.akt = akt;
            this.user = user;
            initComponents();
            this.setTitle("SISTEM
INFORMASI DAN TRANSAKSI ZAHWA");

            this.setLocationRelativeTo(null);
            aktif();
        }

        public void setUser(String
user) {
            this.user = user;
        }

        public void setAkt(int akt) {
            this.akt = akt;
        }
        private void aktif(){

            if(akt ==1){
                kasir(true);
                boolean x = false;
                mnAdmin.setEnabled(x);
                mnKaryawan.setEnabled(x);
                mnPerawatan.setEnabled(x);
                lbAdmin1.setText(user);

lbStatus1.setText("Kasir");
            }
            if(akt ==0){
                admin(true);
                boolean xi = false;

boTransaksi.setEnabled(xi);

mnTransaksi.setEnabled(xi);
                lbAdmin1.setText(user);

lbStatus1.setText("Admin");
            }
        }
        private void kasir(boolean yd){

mnTransaksi.setEnabled(yd);

mnPelanggan.setEnabled(yd);
            mnLaporan.setEnabled(yd);
        }
    }
}

```

```

        mnKeluar.setEnabled(yd);
    }
    private void admin(boolean xi){
        mnAdmin.setEnabled(xi);
        mnKaryawan.setEnabled(xi);

        mnPerawatan.setEnabled(xi);

        mnPelanggan.setEnabled(xi);
        mnLaporan.setEnabled(xi);
        mnKeluar.setEnabled(xi);
    }

    private void
miLogoutActionPerformed(java.awt.e
vent.ActionEvent evt) {
        // TODO add your handling
code here:
        this.dispose();
        Login lg1 = new Login(this,
true);
        lg1.setVisible(true);
    }

    private void
miTransaksiActionPerformed(java.aw
t.event.ActionEvent evt) {
        // TODO add your handling
code here:
        Pembayaran pbyr = new
Pembayaran(akt, user);
        pbyr.setVisible(true);
        this.dispose();
    }

    private void
miPelangganActionPerformed(java.aw
t.event.ActionEvent evt) {
        // TODO add your handling
code here:
        DataPelanggan plg = new
DataPelanggan(akt, user);
        plg.setVisible(true);
        this.dispose();
    }

    private void
miJenisActionPerformed(java.awt.ev
ent.ActionEvent evt) {
        // TODO add your handling
code here:
        JenisPerawatan jpr = new
JenisPerawatan();
        jpr.setVisible(true);
    }

    private void
miPerawatanActionPerformed(java.aw
t.event.ActionEvent evt) {

```

```

        // TODO add your handling
code here:
        DataPerawatan dpr = new
DataPerawatan();
        dpr.setVisible(true);
    }

    private void
miKaryawanActionPerformed(java.awt
.event.ActionEvent evt) {
        // TODO add your handling
code here:
        DataKaryawan dkar = new
DataKaryawan();
        dkar.setVisible(true);
    }

    private void
miLapTransaksiActionPerformed(java
.awt.event.ActionEvent evt) {
        // TODO add your handling
code here:
        LaporanHarian lhar = new
LaporanHarian();
        lhar.setVisible(true);
    }

    private void
miLoginActionPerformed(java.awt.ev
ent.ActionEvent evt) {
        // TODO add your handling
code here:
        UserPass ups = new
UserPass(this, true);
        ups.setVisible(true);
    }

    private void
miKeluarActionPerformed(java.awt.e
vent.ActionEvent evt) {
        // TODO add your handling
code here:
        System.exit(0);
    }

    private void
miLapTransaksiBulananActionPerformed
(java.awt.event.ActionEvent evt)
{
        // TODO add your handling
code here:
        LaporanBulanan lptb = new
LaporanBulanan();
        lptb.setVisible(true);
    }

    private void
boTransaksiActionPerformed(java.aw
t.event.ActionEvent evt) {

```

```

        // TODO add your handling
        code here:
        Pembayaran pbyr = new
        Pembayaran(akt, user);
        pbyr.setVisible(true);
        this.dispose();
    }

    private void
    boLaporanActionPerformed(java.awt.
    event.ActionEvent evt) {
        // TODO add your handling
        code here:
        LaporanHarian lpt = new
        LaporanHarian();
        lpt.setVisible(true);
    }

    private void
    boLogoutActionPerformed(java.awt.e
    vent.ActionEvent evt) {
        // TODO add your handling
        code here:
        this.dispose();
        Login lg1 = new
        Login(this, true);
        lg1.setVisible(true);
    }

    private void
    boKeluarActionPerformed(java.awt.e
    vent.ActionEvent evt) {
        // TODO add your handling
        code here:
        System.exit(0);
    }

    private void
    boPelangganActionPerformed(java.aw
    t.event.ActionEvent evt) {
        // TODO add your handling
        code here:
        DataPelanggan plg = new
        DataPelanggan(akt, user);
        plg.setVisible(true);
        this.dispose();
    }

```

Pembayaran.java

```

public final class Pembayaran
extends javax.swing.JFrame {
    String[] nmKolmas = {"Kode", "Nama
    Perawatan", "Harga", "Sub Total"};
    JTable tabel;
    private TableCellRenderer kanan =
    new renderingKanan();

```

```

    private TableCellRenderer merah =
    new WarnaKodeMerah();
    private DefaultTableModel
    modelMasuk;
    private DefaultTableModel
    modelPembayaran;
    private DefaultTableModel
    modelTampilmasuk;
    String
    kdjprw, kdbp, nmbp, hjbp, hjbp2, kdpela
    ng, keterangan;
        String kdl;
        String nmperl;
        String jns1;
        String hrpl;
        String ketl;
        String nmpl1, nmpl2;
        String caripl, sqlplc;
        String [] kodepl;
        int i=0;
        int totpemJ2 = 0, totpemJ = 0,
        totak = 0;
        int um ;
        int sis ;
        int diskon;
        int disk;
        int iPsp, itbM;
        Date now = new Date();

        SimpleDateFormat frm = new
        SimpleDateFormat("dd/MM/yyyy");
        String utuh =
        frm.format(now);
        String tgl =
        utuh.substring(0, 10);
        String tgl1 =
        utuh.substring(0, 2);
        String tgl2 =
        utuh.substring(1, 2);
        String bln =
        utuh.substring(3, 5);
        String thn =
        utuh.substring(6, 10);
        String [] cari =
        {"Kode", "Nama", "Alamat", "Baru"};
        String ket;
        int x = -1;
        private Sounderror PlaySound = new
        Sounderror();
        private Jam jamOn;
        int akt, disq, panDt, a = -1;
        String user, sqldis;

```

```

    /**
     * Creates new form
    pemesananBarang
    */
    public Pembayaran() {
        initComponents();
    }

```

```

    }

    public Pembayaran(int akt,
String user) {
        this.akt = akt;
        this.user = user;
        initComponents();
        modelPembayaran = new
DefaultTableModel() {
            @Override
            public boolean
isCellEditable (int row, int
column){
                return false;
            }
        };

modelPembayaran.addColumn("Kd_Tran
saksi");

modelPembayaran.addColumn("Tgl_Tra
nsaksi");

modelPembayaran.addColumn("Kd_Kary
awan");

modelPembayaran.addColumn("Kd_Kons
umen");

modelPembayaran.addColumn("Total")
;

modelPembayaran.addColumn("Diskon"
);

tbPembayaran.setModel(modelPembaya
ran);

tbPembayaran.setAutoResizeMode(JTa
ble.AUTO_RESIZE_OFF);
        modelMasuk = new
DefaultTableModel() {
            @Override
            public boolean
isCellEditable (int row, int
column){
                return false;
            }
        };

modelMasuk.addColumn(nmKolmas[0]);

modelMasuk.addColumn(nmKolmas[1]);

modelMasuk.addColumn(nmKolmas[2]);

modelMasuk.addColumn(nmKolmas[3]);

tbMasuk.setModel(modelMasuk);
        modelTampilmasuk = new
DefaultTableModel() {

```

```

            @Override
            public boolean
isCellEditable (int row, int
column){
                return false;
            }
        };

modelTampilmasuk.addColumn("Kode_P
erawatan");

modelTampilmasuk.addColumn("Nama_P
erawatan");

modelTampilmasuk.addColumn("Harga"
);

tbTampilData.setModel(modelTampilm
asuk);

        TableColumn column0 =
tbMasuk.getColumnModel().getColumn
(0);

        TableColumn column1 =
tbMasuk.getColumnModel().getColumn
(1);

        TableColumn column2 =
tbMasuk.getColumnModel().getColumn
(2);

        TableColumn column3 =
tbMasuk.getColumnModel().getColumn
(3);

        TableColumn col0 =
tbPembayaran.getColumnModel().getC
olumn(0);

        TableColumn col1 =
tbPembayaran.getColumnModel().getC
olumn(1);

        TableColumn col2 =
tbPembayaran.getColumnModel().getC
olumn(2);

        TableColumn col3 =
tbPembayaran.getColumnModel().getC
olumn(3);

        TableColumn col4 =
tbPembayaran.getColumnModel().getC
olumn(4);

        TableColumn col5 =
tbPembayaran.getColumnModel().getC
olumn(5);

        TableColumn colA =
tbTampilData.getColumnModel().getC
olumn(0);

        TableColumn colB =
tbTampilData.getColumnModel().getC
olumn(1);

        TableColumn colC =
tbTampilData.getColumnModel().getC
olumn(2);

        for(int a = 0; a <
tbMasuk.getColumnCount();a++)
        {

```

```

tbMasuk.getColumnModel().getColumn
(a).setCellRenderer(new
WarnaKolom());
    }
    for(int b = 0; b <
tbPembayaran.getColumnModel().b++)
    {

tbPembayaran.getColumnModel().getC
olumn(b).setCellRenderer(new
WarnaKolom());
    }
    for(int c = 0; c <
tbTampilData.getColumnModel().c++)
    {

tbTampilData.getColumnModel().getC
olumn(c).setCellRenderer(new
WarnaKolom());
    }

column0.setPreferredWidth(40);
column1.setPreferredWidth(170);
column2.setPreferredWidth(70);
column3.setPreferredWidth(70);
column0.setCellRenderer(merah);
column2.setCellRenderer(kanan);
column3.setCellRenderer(kanan);

col0.setPreferredWidth(90);
col1.setPreferredWidth(120);
col2.setPreferredWidth(130);
col3.setPreferredWidth(140);
col4.setPreferredWidth(105);
col5.setPreferredWidth(105);
col0.setCellRenderer(merah);
col2.setCellRenderer(merah);
col3.setCellRenderer(merah);
col4.setCellRenderer(kanan);
col5.setCellRenderer(kanan);
colA.setCellRenderer(merah);
colC.setCellRenderer(kanan);

```

```

colA.setPreferredWidth(50);

colC.setPreferredWidth(40);
    this.setTitle("Pembayaran
Transaksi Zahwa");
    this.setLocation(600,
250);

ftTglbayar.setText(tgl1+"/"+b1n+"/
"+thn);
    loadPem();
    loadPel();
    loadKar();
    loadJenis();
    loadPerawatan();
    loadCari();
    aktPem1(false);
    aktiPem3(false);
    kOsongkanPem();
    jamOn = new Jam();
    jamOn.setLb(lbJam);
    jamOn.jamOn();
    lbAdmin.setText(user);
    cbDisk.setEnabled(false);
}

public String getString(){
    return kdbp;
}

    public void setAkt(int akt) {
        this.akt = akt;
    }

    public void setUser(String
user) {
        this.user = user;
    }

    private void datalook()
    {

tfKdperawatan.setText(lookPerawata
n.kdL);

tfNmperawatan.setText(lookPerawata
n.nmL);

tfHarga.setText(lookPerawatan.hL);
    }
    private void Next()
    {
        int row =
tbPembayaran.getSelectedRow();
        int b=
tbPembayaran.getRowCount();
        b-=1;
        if(row==b)
        {

```

```

        row=b;
    }
    else
    {
        row+=1;
    }

    tbPembayaran.changeSelection(row,
    WIDTH, false, false);
    }
    private void Akhir()
    {
        int row =
        tbPembayaran.getSelectedRow();
        int b=
        tbPembayaran.getRowCount();
        b-=1;
        if(row==b)
        {
            row=b;
        }
        else
        {
            row=b;
        }
    }

    tbPembayaran.changeSelection(row,
    WIDTH, false, false);
    }
    private void Before()
    {
        int row =
        tbPembayaran.getSelectedRow();
        int b=
        tbPembayaran.getRowCount();
        if(row==b)
        {
            row=0;
        }
        else
        {
            if(row==-1)
                b=0;
            else
                row-=1;
        }
    }

    tbPembayaran.changeSelection(row,
    WIDTH, false, false);
    System.out.println("row :
    "+row);
    }
    private void Awal()
    {
        int row =
        tbPembayaran.getSelectedRow();
        int b=
        tbPembayaran.getRowCount();
        if(row==b)
        {
            row=0;

```

```

    }
    else
    {
        row=0;
    }

    tbPembayaran.changeSelection(row,
    WIDTH, false, false);
    }
    private void bacakdPem()
    {
        modelTampilmasuk.getDataVector().r
        emoveAllElements();

        modelTampilmasuk.fireTableDataChan
        ged();
        iPsp =
        tbPembayaran.getSelectedRow();
        if (iPsp == -1)
        {
            return;
        }
        String noPs =
        modelPembayaran.getValueAt(iPsp,
        0).toString();
        try {
            Connection cpper =
            Libra.KonekZahwa.getkoneksi();
            Statement sp =
            cpper.createStatement();
            String sqlpper =
            "SELECT * FROM dtransaksi WHERE
            Kd_Transaksi = '" + noPs + "'";
            ResultSet rsambarpe =
            sp.executeQuery(sqlpper);
            while
            (rsambarpe.next())
            {
                Object[] oPs = new
                Object[3];
                oPs[0] =
                rsambarpe.getString("Kd_Prw");
                oPs[1] =
                rsambarpe.getString("Nama");
                oPs[2] =
                rsambarpe.getString("Harga");
                modelTampilmasuk.addRow(oPs);
            }
            rsambarpe.close();
            sp.close();
        }
        catch (SQLException ePb)
        {
            System.err.println("Error data
            dtransaksi tidak bisa
            diAmbil!!!");
        }
    }
}

```

```

private void bacakol()
{
    String zkps =
lbNobayar.getText();
    String zkps1 =
zkps.substring(3, 7);
    int iMmsk =
tbMasuk.getRowCount();
    if( i == 0 )
    {
        PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Kolom Masuk masih kosong");
        return;
    }
    for(int z = 0; z<iMmsk; z++)
    {
        String zkdPb = (String)
modelMasuk.getValueAt(z, 0);
        String znmPb = (String)
modelMasuk.getValueAt(z, 1);
        String zhPb =
(String) modelMasuk.getValueAt(z,
2);
        try
        {
            Connection cDps =
Libra.KonekZahwa.getkoneksi();
            String sqlD = "INSERT
INTO dtransaksi VALUES ( ?, ?, ?,
?)";
            PreparedStatement pD =
cDps.prepareStatement(sqlD);
            pD.setString(1, zkps1 );
            pD.setString(2, zkdPb );
            pD.setString(3, znmPb );
            pD.setString(4, zhPb );
            pD.executeUpdate();
            pD.close();
        }
        catch(SQLException ex)
        {

System.out.println("Proses Tambah
dtransaksi Error");
        }
        finally
        {
        }
    }

JOptionPane.showMessageDialog(this
, "Data DTransaksi Telah
DiSimpan");
}
private void aktPem1(boolean i)
{
    cbKar.setEnabled(i);
    cbPel.setEnabled(i);
    cbCari.setEnabled(i);

```

```

    tfCari.setEnabled(i);
    tfKet.setEnabled(i);
    cbJenis.setEnabled(i);
    cbPerawatan.setEnabled(i);
    btKeluar.setEnabled(i);
    btHapem.setEnabled(i);
    btSimpem.setEnabled(i);
    btBatpem.setEnabled(i);
    tfKdperawatan.setEnabled(i);
    tfNmperawatan.setEnabled(i);
    tfNama.setEnabled(i);
    tfHarga.setEnabled(i);
    tfDiskon.setEnabled(i);
    tfTotal.setEnabled(i);
    tfBayar.setEnabled(i);
    tfKembali.setEnabled(i);
}
private void aktiPem2(boolean
j)
{
    cbKar.setEnabled(j);
    cbPel.setEnabled(j);
    cbCari.setEnabled(j);
    tfCari.setEnabled(j);
    tfKet.setEnabled(j);
    cbJenis.setEnabled(j);
    cbPerawatan.setEnabled(j);
    btSimpem.setEnabled(j);
    btBatpem.setEnabled(j);
    tfKdperawatan.setEnabled(j);
    tfNmperawatan.setEnabled(j);
    tfNama.setEnabled(j);
    tfHarga.setEnabled(j);
    tfTotal.setEnabled(j);
    tfDiskon.setEnabled(j);
    tfBayar.setEnabled(j);
    tfKembali.setEnabled(j);
    btKeluar.setEnabled(j);
}
private void aktiPem3(boolean
k)
{
    cbJenis.setEnabled(k);
    cbPerawatan.setEnabled(k);
    tfKdperawatan.setEnabled(k);
    tfNmperawatan.setEnabled(k);
    tfHarga.setEnabled(k);
    btambil.setEnabled(k);
    cbDisk.setEnabled(k);
}
private void kOsongkanPem()
{
    x=-1;
    tfKdperawatan.setText("");
    tfCari.setText("");
    tfKet.setText("");
    tfNmperawatan.setText("");
    tfNama.setText("");
    tfHarga.setText("");
    tfBayar.setText("0");
    tfDiskon.setText("0");

```

```

        tfTotal.setText("0");
        tfKembali.setText("0");
        cbKar.setSelectedIndex(-1);
        cbPel.setSelectedIndex(-1);
        cbCari.setSelectedIndex(-1);
        cbJenis.setSelectedIndex(-1);

        cbPerawatan.setSelectedIndex(-1);
    }
    private void kOsongkanpera()
    {
        tfKdperawatan.setText("");
        tfNmperawatan.setText("");
        tfHarga.setText("");
        cbJenis.setSelectedIndex(-1);

        cbPerawatan.setSelectedIndex(-1);
    }
    private void nomorPeby()
    {
        try
        {
            Connection conPbyr =
            Libra.KonekZahwa.getkoneksi();
            Statement sPbyr =
            conPbyr.createStatement();
            String sqlbyr ="select
            Kd_Transaksi as nokdyr from
            transaksi ORDER by Kd_Transaksi
            desc ";
            ResultSet rsbyr =
            sPbyr.executeQuery(sqlbyr);
            if(rsbyr.next())
            {
                String noPs =
                rsbyr.getString("nokdyr");
                String noPsa =
                noPs.substring(1 , 4);
                int noPsx =
                Integer.parseInt(noPsa)+1;
                if (noPsx > 999)
                {

                    PlaySound.PlaySound();

                    JOptionPane.showMessageDialog(this
                    , "Data Transaksi Over, telah
                    mencapai maksimal!");

                    lbNobayar.setText("Data Penuh");

                    tfKdperawatan.setEnabled(false);

                    tfNmperawatan.setEnabled(false);

                    btSimpem.setEnabled(false);
                }
                if (noPsx > 0)
                {
                    lbNobayar.setText("[
                    N00" +noPsx+" ]");

```

```

                }
                if (noPsx >= 10)
                {
                    lbNobayar.setText("[
                    N0" +noPsx+" ]");
                }
                if (noPsx >= 100)
                {
                    lbNobayar.setText("[
                    N" +noPsx+" ]");
                }
            }
            else
            {
                lbNobayar.setText("[
                    N000 ]");
            }
            rsbyr.close();
            sPbyr.close();
        }
        catch(SQLException ep)
        {
            System.out.println("Terjadi error
            di ambil data nomor Transaksi");
        }
        cbPel.requestFocus();
    }
    private void hitTotmas()
    {
        int uBP = 0;
        if
        ("".equals(tfTotal.getText()))
        {
            PlaySound.PlaySound();

            JOptionPane.showMessageDialog(this
            , "Total Bayar Masih Kosong !!");

            tfKdperawatan.requestFocus();
            return;
        }
        int toT =
        Integer.parseInt(tfTotal.getText()
        .toString());
        if
        ("".equals(tfBayar.getText()))
        {
            int uBMin = toT;
            uBP = uBP + uBMin;
            sis = uBP - toT;
            tfBayar.setText(""+uBP);

            tfKembali.setText(""+sis);
            return;
        }
        else
        {
            int umMin = toT;
            uBP = uBP + umMin;

```



```

        sis = uBP - toT;
        tfBayar.setText(""+uBP);
        tfKembali.setText(""+sis);
        uBP =
Integer.parseInt(tfBayar.getText()
.toString());
        toT =
Integer.parseInt(tfTotal.getText()
.toString());
        }
        if(uBP < toT)
        {
            PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Jumlah Uang yang di bayarkan
masih belum cukup !");
            tfBayar.setText(" ");
            tfBayar.requestFocus();
        }
    }
private void Batalambil()
{
    int dis,plh,ttl;
    plh =
Integer.parseInt(modelMasuk.getVal
ueAt(itbM, 3).toString());
    dis = diskon;
    ttl =
Integer.parseInt(tfTotal.getText()
.toString());
    totpemJ2 = ttl - plh;
    tfTotal.setText(""+totpemJ2);
    tfBayar.setText(""+totpemJ2);
    tfKembali.setText("");
}
private void loadPel()
{
    cbPel.removeAllItems();
    try
    {
        Connection cPb =
        Libra.KonekZahwa.getkoneksi();
        Statement sp =
        cPb.createStatement();
        String sqP ="select *
        from konsumen ";
        ResultSet rsp =
        sp.executeQuery(sqP);
        while(rsp.next())
        {
            Object[] oP = new
            Object[2];
            oP[0] =
            rsp.getString("Kd_Konsumen");
            oP[1] =
            rsp.getString("Nama");

            cbPel.addItem(""+oP[0]+"
            "+oP[1] );

```

```

        }
        rsp.close();
        sp.close();
    }
    catch(SQLException e)
    {
        System.out.println("Terjadi error
        di load data Pelanggan Combo");
    }
}
private void cariPl(){
    cbPel.removeAllItems();
    try {
        Connection c =
        Libra.KonekZahwa.getkoneksi();
        Statement st =
        c.createStatement();
        ResultSet rs =
        st.executeQuery(sqlplc);
        while
        (rs.next()) {
            Object[] oP1
            = new Object[2];
            oP1[0] =
            rs.getString("a.Kd_Konsumen");
            oP1[1] =
            rs.getString("a.nama");

            cbPel.addItem(""+oP1[0]+"
            "+oP1[1] );
        }
    } catch (SQLException ex) {
        System.out.println("Terjadi error
        di cari data Pelanggan");
    }
}
private void loadKar()
{
    cbKar.removeAllItems();
    try
    {
        Connection cKpb =
        Libra.KonekZahwa.getkoneksi();
        Statement skp =
        cKpb.createStatement();
        String sqKp ="select *
        from Karyawan ";
        ResultSet rsk =
        skp.executeQuery(sqKp);
        while(rsk.next())
        {
            Object[] oK = new
            Object[2];
            oK[0] =
            rsk.getString("Kd_Karyawan");
            oK[1] =
            rsk.getString("Nama");

```

```

cbKar.addItem("(" + oK[0] + ")
"+oK[1] );
    }
    rsk.close();
    skp.close();
}
catch(SQLException e)
{

System.out.println("Terjadi error
di load data Karyawan Combo");
    }
}
private void loadJenis()
{
    cbJenis.removeAllItems();
    try
    {
        Connection cJ =
        Libra.KonekZahwa.getkoneksi();
        Statement sj =
        cJ.createStatement();
        String sqlJ ="select *
from jenis_prw ";
        ResultSet rsj =
        sj.executeQuery(sqlJ);
        while(rsj.next())
        {
            Object[] oJ = new
Object[2];
            oJ[0] =
rsj.getString("Kd_Jenis");
            oJ[1] =
rsj.getString("Nama");

cbJenis.addItem("(" + oJ[0] + ")
"+oJ[1] );
        }
        rsj.close();
        sj.close();
    }
    catch(SQLException e)
    {

System.out.println("Terjadi error
di load data Jenis Combo");
    }
}
private void loadPerawatan()
{
    cbPerawatan.removeAllItems();
    try
    {
        Connection cJ =
        Libra.KonekZahwa.getkoneksi();
        Statement sj =
        cJ.createStatement();
        String sqlJ ="select *
from perawatan";

```

```

        ResultSet rsj =
        sj.executeQuery(sqlJ);
        while(rsj.next())
        {
            Object[] oJ = new
Object[2];
            oJ[0] =
rsj.getString("Kd_Prw");
            oJ[1] =
rsj.getString("Nama");

cbPerawatan.addItem("(" + oJ[0] + ")
"+oJ[1] );
        }
        rsj.close();
        sj.close();
    }
    catch(SQLException e)
    {

System.out.println("Terjadi error
di load data Perawatan Combo");
    }
}
private void
loadPilihJenisPerawatan()
{
    cbPerawatan.removeAllItems();
    if(cbJenis.getSelectedIndex()
!=-1){
        kdjprw = (String)
cbJenis.getSelectedItem();
        String kdjprw2 =
kdjprw.substring(1, 5);
        try
        {
            Connection cJ =
            Libra.KonekZahwa.getkoneksi();
            Statement sj =
            cJ.createStatement();
            String sqlpper =
            "SELECT * FROM perawatan WHERE
            Kd_Jenis = '" + kdjprw2 + "'";
            ResultSet rsj =
            sj.executeQuery(sqlpper);
            while(rsj.next())
            {
                Object[] oJ = new
Object[2];
                oJ[0] =
                rsj.getString("Kd_Prw");
                oJ[1] =
                rsj.getString("Nama");

cbPerawatan.addItem("(" + oJ[0] + ")
"+oJ[1] );
            }
            rsj.close();
            sj.close();
        }
        catch(SQLException e)

```

```

        {
            System.out.println("Terjadi error
            di load data PilihPerawatan
            Combo");
        }
    }
}
private void pilihPerawatan() {
    if (cbPerawatan.getSelectedIndex() != -1) {
        String kdperl = (String)
        cbPerawatan.getSelectedItem();
        String kdper2 =
        kdperl.substring(1, 5);
        String nmpe;
        String hrge;
        try {
            Connection cpbar =
            Libra.KonekZahwa.getkoneksi();
            Statement spb =
            cpbar.createStatement();
            String sqlpbar =
            "SELECT * FROM perawatan WHERE
            Kd_Prw = '" + kdper2 + "'";
            ResultSet rsambarpes =
            spb.executeQuery(sqlpbar);
            if (rsambarpes.next()) {
                nmpe =
                rsambarpes.getString("Nama");
                hrge =
                rsambarpes.getString("Harga");

                tfKdperawatan.setText(""+kdper2.to
                UpperCase());

                tfNmperawatan.setText(""+nmpe);

                tfHarga.setText(""+hrge);

                btambil.requestFocus();
            }
        } catch (SQLException ePb) {
        }

        System.err.println("Error data
        tidak bisa diAmbil!!!");
    }
}
private void loadPem() {
    modelPembayaran.getDataVector().re
    moveAllElements();

    modelPembayaran.fireTableDataChang
    ed();
}

```

```

        try {
            Connection cPe =
            Libra.KonekZahwa.getkoneksi();
            Statement spe =
            cPe.createStatement();
            String sqlpe = "SELECT
            a.Kd_Transaksi, a.Tanggal,
            a.Total, a.Diskon, b.nama, "
            + " c.nama
            FROM transaksi a , karyawan b,
            konsumen c WHERE a.Kd_Karyawan =
            b.Kd_Karyawan "
            + "and
            a.Kd_Konsumen = c.Kd_Konsumen and
            a.Tanggal like '"+thn+"-"+bln+"-
            "+tg11+"'";
            ResultSet rspe =
            spe.executeQuery(sqlpe);
            while (rspe.next()) {
                Object[] ope = new
                Object[6];
                ope[0] =
                rspe.getString("a.Kd_Transaksi");
                String tglt =
                rspe.getString("a.Tanggal");
                String th1 =
                tglt.substring(0, 4);
                String bl1 =
                tglt.substring(5, 7);
                String tg1 =
                tglt.substring(8, 10);
                String tglt2 =
                tg1+"-"+bl1+"-"+th1;
                ope[1] = tglt2;
                ope[2] =
                rspe.getString("b.nama");
                ope[3] =
                rspe.getString("c.nama");
                ope[4] =
                rspe.getString("a.Total");
                ope[5] =
                rspe.getString("a.Diskon");

                modelPembayaran.addRow(ope);
            }
            rspe.close();
            spe.close();
        } catch (SQLException ep) {
        }

        System.out.println("Terjadi error
        di load data transaksi");
    }
}
private void loadCari() {
    cbCari.removeAllItems();
    for (int c = 0; c <
    cari.length; c++) {

```

```

        cbCari.addItem(cari[c]);
    }
}

private void aturtext(KeyEvent
ke, JTextField jt, int jml){
    char e = ke.getKeyChar();
    if(jt.getText().length()>jml){
        ke.consume();
    }
    if(!Character.isDigit(e)){
        ke.consume();
    }
}

private void cetak(){
    int y = 0;
    Frame cetak = new Frame();
    PrintJob print =
cetak.getToolkit().getPrintJob(cet
ak, "Mencetak", null, null);
    if(print!=null){
        Graphics grap =
print.getGraphics();
        if(grap!=null){
            grap.setFont(new
Font("Dialog", 1, 10));
            grap.drawString("Spa
dan Salon Zahwa", 30, 9);
            grap.setFont(new
Font("Dialog", 1, 7));
            grap.drawString("Jl.
Nitikan Baru No. 27 Umbulharjo ",
30, 19);
            grap.drawString("
Yogyakarta 0274-6632877", 30, 29);
            grap.setFont(new
Font("Dialog", 1, 8));
            String notr =
lbNobayar.getText();
            grap.drawString("Nota
: "+notr, 30, 42);

            grap.drawString("Pelanggan :
"+tftNama.getText(), 30, 52);
            String ky =
cbKar.getSelectedItem().toString()
;
            String kyl =
ky.substring(6);

            grap.drawString("Karyawan :
"+kyl, 30, 61);

            grap.drawString("Tanggal :
"+ftTglbayar.getText()+" Jam :
"+lbJam.getText(), 30, 71);
            grap.setFont(new
Font("Dialog", 0, 7));
            grap.drawLine(20, 78,
180, 78);

```

```

        int nn = 0;
        for (int xx = 0;xx <
modelMasuk.getRowCount(); xx++ ){
            nn = nn + 10;

            grap.drawString(modelMasuk.getValu
eAt(xx, 0).toString()+"
"+modelMasuk.getValueAt(xx,
1).toString().toString(), 30,
82+nn );

            grap.drawString(modelMasuk.getValu
eAt(xx, 2).toString(), 150, 82+nn
);
        }
        grap.drawLine(20,
88+nn, 180, 88+nn);

        grap.drawString("Total      : ", 30,
99+nn);

        grap.drawString(tftTotal.getText(),
150, 99+nn);

        grap.drawString("Bayar      : ", 30,
109+nn);

        grap.drawString(tftBayar.getText(),
150, 109+nn);

        grap.drawString("Kembali   : ", 30,
119+nn);

        grap.drawString(tftKembali.getText(
), 150, 119+nn);

        grap.drawString("Diskon    : ", 30,
129+nn);

        grap.drawString(String.valueOf(dis
kon), 150, 129+nn);
        grap.drawLine(20,
133+nn, 180, 133+nn);
        grap.setFont(new
Font("Dialog", 2, 11));

        grap.drawString("Terima Kasih",
55, 145+nn);
        grap.setFont(new
Font("Dialog", 1, 9));

        grap.drawString("Kepuasan Anda ,
Tujuan Kami ", 30, 157+nn);
    }
    print.end();
    print.end();
}

btTamPem.requestFocus();
}

private void tambahTran(){
    aktPem1(false);
}

```

```

        aktiPem2(true);
        kOsongkanPem();
        nomorPeby();

btTamPem.setEnabled(false);
        btBatpem.setEnabled(true);
    }
    private void hapusPerawatan() {
        Batalambil();
        modelMasuk.removeRow(itbM);
        btHapem.setEnabled(false);
    }
    private void batalTran() {
        aktPem1(false);
        kOsongkanPem();
        nomorPeby();

btBatpem.setEnabled(false);
        btTamPem.setEnabled(true);

modelMasuk.getDataVector().removeAllElements();

modelMasuk.fireTableDataChanged();
        btTamPem.requestFocus();
        i = 0 ;
        totpemJ2 = 0;
        um = 0 ;
        sis = 0;
        diskon = 0;
    }
    private void pekkon() {
        try
        {
            Connection cJ =
Libra.KonekZahwa.getkoneksi();
            Statement sj =
cJ.createStatement();
            String sqlJ ="select
a.Pekerjaan from konsumen a where
Kd_Konsumen like '"+kdpelang+"' ";
            ResultSet rsj =
sj.executeQuery(sqlJ);
            while(rsj.next())
            {
                Object[] oJ = new
Object[1];
                oJ[0] =
rsj.getString("a.Pekerjaan");
                ket =
oJ[0].toString();
            }
            rsj.close();
            sj.close();
        }
        catch (SQLException e)
        {

System.out.println("Terjadi error
di load Pekerjaan konsumen");
        }

```

```

        System.out.println("ket :
"+ket);
    }

    private void keluarTran() {
        int jwb=
JOptionPane.showConfirmDialog(this
, "Anda akan membatalkan transaksi
?",
        "Konfirmasi",
JOptionPane.YES_NO_OPTION,
JOptionPane.QUESTION_MESSAGE);
        switch(jwb){
            case 0 : this.dispose();
                break;
            case 1 :
btBatpem.requestFocus();
                break;
            default :
btBatpem.requestFocus();
        }
    }
    private void nildiskon() {
        try {
            Connection c =
Libra.KonekZahwa.getkoneksi();
            Statement st =
c.createStatement();
            sqldis = "select diskon
from diskon where Keterangan like
'"+ket+"'";
            ResultSet rs =
st.executeQuery(sqldis);
            while(rs.next())
            {
                disq =
Integer.parseInt(rs.getString(1));
            }
            rs.close();
            st.close();
        }
        catch (SQLException ex) {

System.out.println("Terjadi error
di cari panj data diskon");
        }
        System.out.println("disq :
"+disq);
    }
    private void Simpan() {
        int cbhit1 =
cbKar.getSelectedIndex();
        if (cbhit1!=-1)
        {
            PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Karyawan mohon dipilih !",
"PERHATIAN",JOptionPane.WARNING_ME
SSAGE);

```

```

        cbKar.requestFocus();
        return;
    }
    if
    ("".equals(tfNama.getText()))
    {
        PlaySound.PlaySound();

        JOptionPane.showMessageDialog(this
        , "Pelanggan mohon dipilih !",

        "PERHATIAN",JOptionPane.WARNING_ME
        SSAGE);

        cbPel.requestFocus();
        return;
    }
    if
    ("".equals(tfTotal.getText()))
    {
        PlaySound.PlaySound();

        JOptionPane.showMessageDialog(this
        , "Jumlah total mohon diisi !",

        "PERHATIAN",JOptionPane.WARNING_ME
        SSAGE);

        tfKdperawatan.requestFocus();
        return;
    }
    if
    ("".equals(tfBayar.getText()))
    {
        PlaySound.PlaySound();

        JOptionPane.showMessageDialog(this
        , "Nilai Pembayaran mohon diisi
        !",

        "PERHATIAN",JOptionPane.WARNING_ME
        SSAGE);

        tfBayar.requestFocus();
        return;
    }
    String kt =
    lbNobayar.getText();
    String kt1 =
    kt.substring(3, 7);
    String tgl1 =
    ftTglbayar.getText();
    String tgl =
    tgl1.substring(0, 2);
    String b11 =
    tgl1.substring(3, 5);
    String th1 =
    tgl1.substring(6, 10);
    String tgl2 = th1+"-
    "+b11+"-"+tgl;
    String kart = (String)
    cbKar.getSelectedItem();

```

```

        String kart1 =
        kart.substring(1, 5);
        String pelt = kdpelang;
        String admin =
        lbAdmin.getText();
        String tott =
        tfTotal.getText();
        try
        {
            Connection cPsn =
            Libra.KonekZahwa.getkoneksi();
            String sqPs = "INSERT
            INTO transaksi VALUES ( ?, ?, ?,
            ?, ?, ?, ?)";
            PreparedStatement pPs
            = cPsn.prepareStatement(sqPs);
            pPs.setString(1, kt1);
            pPs.setString(2,
            tgl2);
            pPs.setString(3,
            kart1);
            pPs.setString(4,
            pelt);
            pPs.setString(5,
            tott);
            pPs.setString(6,
            String.valueOf(diskon));
            pPs.setString(7,
            admin);
            pPs.executeUpdate();
            pPs.close();

            JOptionPane.showMessageDialog(this
            , "Pembayaran Telah DiSimpan");
        }
        catch (SQLException ex)
        {
            System.out.println("Proses Simpan
            Transaksi Error");
        }
        finally
        {
            bacakol();
            cetak();
        }
        aktPem1(false);
        loadPem();
        kOsongkanPem();

        btBatpem.setEnabled(false);
        btTamPem.setEnabled(true);

        modelMasuk.getDataVector().removeA
        llElements();

        modelMasuk.fireTableDataChanged();
        btTamPem.requestFocus();
        i = 0 ;
        totpemJ2 = 0;
        um = 0 ;

```

```

        sis = 0;
        diskon = 0;
        totpemJ = 0;
        totak = 0;
        disk = 0;
        iPsp = 0;
        itbM = 0;
    }

    private void
    formWindowClosed(java.awt.event.Wi
    ndowEvent evt) {
        // TODO add your handling
        code here:
            mnUtama menuUta =
            new mnUtama(akt, user);

        menuUta.setAkt(akt);

        menuUta.setVisible(true);
    }

    private void
    btTamPemActionPerformed(java.awt.e
    vent.ActionEvent evt) {
        // TODO add your handling
        code here:
            tambahTran();
    }

    private void
    btHapemActionPerformed(java.awt.ev
    ent.ActionEvent evt) {
        // TODO add your handling
        code here:
            hapusPerawatan();
    }

    private void
    btSimpemActionPerformed(java.awt.e
    vent.ActionEvent evt) {
        // TODO add your handling
        code here:
            Simpan();
            cbDisk.setSelected(false);
    }

    private void
    btBatpemActionPerformed(java.awt.e
    vent.ActionEvent evt) {
        // TODO add your handling
        code here:
            batalTran();
            cbDisk.setEnabled(false);
            cbDisk.setSelected(false);
    }

    private void
    cbKarKeyPressed(java.awt.event.Key
    Event evt) {

```

```

        // TODO add your handling
        code here:
            int
            ascii=evt.getKeyCode();
            if (ascii==10)
            {
                cbPel.requestFocus();
            }
        }

    private void
    cbPelActionPerformed(java.awt.even
    t.ActionEvent evt) {
        // TODO add your handling
        code here:

        if(cbPel.getSelectedIndex()>-1){
            nmpl = (String)
            cbPel.getSelectedItem();
            String nmpl1 =
            nmpl.substring(1,5);
            nmpl2 =
            nmpl.substring(8);
            kdpelang = nmpl1;
            tfNama.setText(nmpl2);

            cbCari.setSelectedIndex(-1);
            tfCari.setText("");
            pekkon();
            tfKet.setText(ket);
            aktiPem3(true);
        }
    }

    private void
    cbPelKeyPressed(java.awt.event.Key
    Event evt) {
        // TODO add your handling
        code here:
            int
            ascii=evt.getKeyCode();
            if (ascii==10)
            {

            tfKdperawatan.requestFocus();
            }
        }

    private void
    cbCariActionPerformed(java.awt.eve
    nt.ActionEvent evt) {
        // TODO add your handling
        code here:
            cbPel.setSelectedIndex(-
            1);
            tfCari.requestFocus();
        }
    }

```

```

        private void
tfKdperawatanKeyPressed(java.awt.e
vent.KeyEvent evt) {
    // TODO add your handling
code here:
        int
ascii=evt.getKeyCode();
        if (ascii==10)
        {
            String kdbarl =
tfKdperawatan.getText();
            if
("".equals(tfKdperawatan.getText()
)) {

PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Kode Perawatan masih kosong");

tfKdperawatan.requestFocus();
        return;
    }
    try {
        Connection cpbar =
Libra.KonekZahwa.getkoneksi();
        Statement spb =
cpbar.createStatement();
        String sqlpbar =
"SELECT * FROM perawatan WHERE
Kd_Prw = '" + kdbarl + "'";
        ResultSet
rsambarpes =
spb.executeQuery(sqlpbar);
        if
(rsambarpes.next()) {
            nmperl =
rsambarpes.getString("Nama");
            jns1 =
rsambarpes.getString("Kd_Jenis");
            hrgr1 =
rsambarpes.getString("Harga");
            ket1 =
rsambarpes.getString("Keterangan")
;

tfKdperawatan.setText(""+kdbarl.to
UpperCase());

tfNmperawatan.setText(""+nmperl);

tfHarga.setText(""+hrgr1);

JOptionPane.showMessageDialog(this
, "Kode Perawatan
"+kdbarl.toUpperCase()+ " Harganya
"+hrgr1);
    }
    else
    {

```

```

PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Data "+tfKdperawatan.getText()+
" Tidak
ada!!!", "INFORMASI", JOptionPane.IN
FORMATION_MESSAGE);

tfKdperawatan.setText("");

tfKdperawatan.requestFocus();

kOsongkanpera();
    }
    catch (SQLException
ePb)
    {

System.err.println("Error data
tidak bisa diAmbil!!!");
    }
}

private void
tfKdperawatanKeyTyped(java.awt.eve
nt.KeyEvent evt) {
    // TODO add your handling
code here:

if(tfKdperawatan.getText().length(
)>3){
    evt.consume();
}

private void
tfHargaKeyPressed(java.awt.event.K
eyEvent evt) {
    // TODO add your handling
code here:
        int
ascii=evt.getKeyCode();
        if (ascii==10)
        {
            btambil.doClick();
        }
    }

private void
btlookPerawatanActionPerformed(jav
a.awt.event.ActionEvent evt) {
    // TODO add your handling
code here:
        lookPerawatan lBp = new
lookPerawatan();
        lBp.setVisible(true);
        if(lookPerawatan.o ==
true){

```



```

        datalook();
    }
}

private void
btambilActionPerformed(java.awt.ev
ent.ActionEvent evt) {
    // TODO add your handling
code here:
    i=i+1;
    int tempTot =0;
    if(tbMasuk.getRowCount() <
1){
cbDisk.setEnabled(false);
    }
    cbDisk.setEnabled(true);
    int iMmsk =
tbMasuk.getRowCount();
    int iMmsk1 = iMmsk - 1;
    for (int m = 0; m <=
iMmsk1; m++ )
    {
        String kdpr = (String)
tbMasuk.getValueAt(m, 0);
        String kdpr1 =
tfKdperawatan.getText();
        if(kdpr.equals(kdpr1))
        {

PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Data Perawatan "+kdpr1+" sudah
ada");

        kOsongkanpera();

tfKdperawatan.requestFocus();
        return;
    }
    }
    if
("".equals(tfKdperawatan.getText()
))
    {
        PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Kode Perawatan masih kosong");

tfKdperawatan.requestFocus();
        return;
    }
    if
("".equals(tfNmperawatan.getText()
))
    {
        PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Nama Perawatan masih kosong");

```

```

tfNmperawatan.requestFocus();
        return;
    }
    if
("".equals(tfHarga.getText()))
    {
        PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Harga Perawatan masih kosong");

tfHarga.requestFocus();
        return;
    }

    kdbp =
tfKdperawatan.getText();
    nmbp =
tfNmperawatan.getText();
    hjbp = tfHarga.getText();
    Object[] omasbar = new
Object[4];
    omasbar[0] =
kdbp.toString();
    omasbar[1] =
nmbp.toString();
    omasbar[2] =
hjbp.toString();
    omasbar[3] = hjbp;

modelMasuk.addRow(omasbar);
    tempTot = tempTot +
Integer.parseInt(hjbp);
    kOsongkanpera();
    totpemJ2 = totpemJ2 +
tempTot;

tfTotal.setText(""+totpemJ2);

tfKdperawatan.requestFocus();
    hitTotmas();
}

private void
btambilKeyPressed(java.awt.event.K
eyEvent evt) {
    // TODO add your handling
code here:
    int
ascii=evt.getKeyCode();
    if (ascii==10)
    {
        btambil.doClick();
    }
}

```

```

        private void
        cbJenisActionPerformed(java.awt.ev
        ent.ActionEvent evt) {
            // TODO add your handling
            code here:
            loadPilihJenisPerawatan();
        }

```

```

        private void
        cbPerawatanActionPerformed(java.aw
        t.ActionEvent evt) {
            // TODO add your handling
            code here:
            pilihPerawatan();
        }

```

```

        private void
        cbDiskActionPerformed(java.awt.eve
        nt.ActionEvent evt) {
            // TODO add your handling
            code here:
            nildiskon();
            if(cbDisk.isSelected()){
                diskon = totpemJ2 *
                disq /100;
                totpemJ = totpemJ2 -
                diskon;
                tfDiskon.setText(""+diskon);
                tfTotal.setText(""+totpemJ2);
                tfBayar.setText(""+totpemJ);
            }
            aktiPem3(false);
        }

```

```

        private void
        tbMasukMouseClicked(java.awt.event
        .MouseEvent evt) {
            // TODO add your handling
            code here:
            itbM =
            tbMasuk.getSelectedRow();
            if(itbM == -1){
                return;
            }
            btHapem.setEnabled(true);
        }

```

```

        private void
        tfBayarMouseClicked(java.awt.event
        .MouseEvent evt) {
            // TODO add your handling
            code here:
            String pattern= "[0-
            9]{1,8}";
            String bayar =
            JOptionPane.showInputDialog(this,

```

```

            "Masukkan Angka",
            "Input",JOptionPane.OK_OPTION);
            if("").equals(bayar)||bayar
            ==null){
                bayar =
                JOptionPane.showInputDialog(this,
                "Masukkan Angka",
                "Input",JOptionPane.OK_OPTION);
                return;
            }

```

```

            if(bayar.matches(pattern)){
                long byr =
                Integer.parseInt(bayar);

                tfBayar.setText(String.valueOf(byr
                ));
            }
        }

```

```

        private void
        tfBayarKeyPressed(java.awt.event.K
        eyEvent evt) {
            // TODO add your handling
            code here:
            int
            ascii=evt.getKeyCode();
            long tot1, ub1,sisa;
            if (ascii==10)
            {
                tot1 =totpemJ;
                ub1 =
                Integer.parseInt(tfBayar.getText()
                );
                sisa = ub1 - tot1;
                if (ub1 > tot1)
                {

```

```

                tfKembali.setText(""+sisa);

                JOptionPane.showMessageDialog(this
                , "Jumlah      Rp.
                "+totpemJ2+",00\n"
                + "Diskon
                Rp. "+diskon+",00\n"
                + "Total
                Rp. "+totpemJ+",00\n"
                + "Bayar
                Rp. "+ub1+",00\n"
                + "Kembali
                Rp. "+sisa+",00","Total Pembayaran
                Transaksi",JOptionPane.INFORMATION
                _MESSAGE);

                btSimpem.setEnabled(true);
            }
            if (ub1 < tot1)
            {

```

```

                PlaySound.PlaySound();
            }

```

```
JOptionPane.showMessageDialog(this
, "Uang Bayar kurang !!!");
```

```
JOptionPane.showMessageDialog(this
, "Kurang Rp.
"+sis+"00\n", "Perhatian
!!!", JOptionPane.ERROR_MESSAGE);
```

```
tfbayar.setText("0");
```

```
tfkembali.setText("");
```

```
tfbayar.requestFocus();
```

```
btSimpem.setEnabled(false);
```

```
    }
}
```

```
private void
tfbayarKeyPressed(java.awt.event.Key
Event evt) {
    // TODO add your handling
code here
    aturtext(evt, tfbayar, 7);
}
```

```
private void
btKirimMouseClicked(java.awt.eve
nt.MouseEvent evt) {
    // TODO add your handling
code here:
    Awal();
    bacakdPem();
}
```

```
private void
btKirimMouseClicked(java.awt.ev
ent.MouseEvent evt) {
    // TODO add your handling
code here:
    Akhir();
    bacakdPem();
}
```

```
private void
btKiriMouseClicked(java.awt.event.
MouseEvent evt) {
    // TODO add your handling
code here:
    Before();
    bacakdPem();
}
```

```
private void
btKananMouseClicked(java.awt.event
.MouseEvent evt) {
    // TODO add your handling
code here:
    Next();
}
```

```
    bacakdPem();
}
```

```
private void
tbPembayaranMouseClicked(java.awt.
event.MouseEvent evt) {
    // TODO add your handling
code here:
    bacakdPem();
}
```

```
private void
btKeluarActionPerformed(java.awt.e
vent.ActionEvent evt) {
    // TODO add your handling
code here:
    keluarTran();
}
```

```
private void
tfCariKeyPressed(java.awt.event.Ke
yEvent evt) {
    // TODO add your handling
code here:
    int
ascii=evt.getKeyCode();
    if (ascii==10)
    {
```

```
        if("".equals(cbCari.getSelectedIte
m())){
```

```
            PlaySound.PlaySound();
```

```
JOptionPane.showMessageDialog(this
, "Pilih Jenis Pencarian");
```

```
cbCari.requestFocus();
```

```
tfCari.setText("");
return;
```

```
        }
        x =
cbCari.getSelectedIndex();
        caripl =
tfCari.getText();
        if(x ==0){
            sqlplc = "SELECT
a.Kd_Konsumen, a.nama FROM
konsumen a WHERE Kd_Konsumen LIKE
'" + caripl+ "'";
```

```
            cariPl();
```

```
        }
        if(x ==1){
            sqlplc = "SELECT
a.Kd_Konsumen, a.nama FROM
konsumen a WHERE Nama LIKE '" +
caripl+ "%'";
```

```

        cariPl();
    }
    if(x ==2){
        sqlplc = "SELECT
a.Kd_Konsumen, a.nama FROM
konsumen a WHERE Alamat LIKE '%"
+ cariPl+ "%'";

        cariPl();
    }
    if(x ==3){
        sqlplc = "SELECT
a.Kd_Konsumen, a.nama FROM
konsumen a WHERE status LIKE
'n'";

        cariPl();
    }
}
}

```

lookPerawatan.java

```

public final class lookPerawatan
extends javax.swing.JFrame {
private DefaultTableModel
modelLookper;
public static String kdL, nmL, jL,
hL, ketL ;
public static boolean o = false;
String kdbp ;
String nmbp ;
String hjbp ;
String jbp ;
public TableCellRenderer merah
= new merahKode();
/**
 * Creates new form lookBrgpes
 */
public lookPerawatan() {
    initComponents();
    modelLookper = new
DefaultTableModel(){
        @Override
        public boolean
isCellEditable (int row, int
coloumn){
            return false;
        }
    };

modelLookper.addColumn("Kd.
Prwtn");

modelLookper.addColumn("Nm.
Perawatan");

modelLookper.addColumn("Jns.
Prwtn");

```

```

modelLookper.addColumn("Harga");

modelLookper.addColumn("Keterangan
");

tbLookper.setModel(modelLookper);

tbLookper.setAutoResizeMode(JTable
.AUTO_RESIZE_SUBSEQUENT_COLUMNS);
    TableColumn col0 =
tbLookper.getColumnModel().getColu
mn(0);
    TableColumn col1 =
tbLookper.getColumnModel().getColu
mn(1);
    TableColumn col2 =
tbLookper.getColumnModel().getColu
mn(2);
    TableColumn col3 =
tbLookper.getColumnModel().getColu
mn(3);
    TableColumn col4 =
tbLookper.getColumnModel().getColu
mn(4);

    col0.setWidth(40);
    col1.setWidth(150);
    col2.setWidth(40);
    col3.setWidth(70);
    col4.setWidth(200);

col0.setCellRenderer(merah);

col3.setCellRenderer(merah);

    this.setTitle("Tabel Look
Perawatan Zahwa");
    this.setLocation(300,
175);
    loadPerawatanL();
}
public void loadPerawatanL()
{
modelLookper.getDataVector().remov
eAllElements();

modelLookper.fireTableDataChanged(
);

    try
    {
        Connection cper =
Libra.KonekZahwa.getkoneksi();
        Statement sper =
cper.createStatement();

        String sqlper ="select
* from perawatan ORDER by Kd_Jenis
ASC";

```

```

        ResultSet rsper =
sper.executeQuery(sqlper);

        while(rsper.next())
        {
            Object[] obarLpes
= new Object[5];
            obarLpes[0] =
rsper.getString("Kd_Prw");
            obarLpes[1] =
rsper.getString("Nama");
            obarLpes[2] =
rsper.getString("Kd_Jenis");
            obarLpes[3] =
rsper.getString("Harga");
            obarLpes[4] =
rsper.getString("Keterangan");
modelLookper.addRow(obarLpes);
        }
        rsper.close();
        sper.close();
    }
    catch (SQLException eLpes)
    {

```

```

System.out.println("Terjadi error
di load data Perawatan
LookPerawatan ");
    }
}

```

```

        private void
tbLookperMouseClicked(java.awt.eve
nt.MouseEvent evt) {
            // TODO add your handling
code here:
            o = true;
            int iLp =
tbLookper.getSelectedRow();
            if (iLp == -1)
            {
                return;
            }
            kdL =
modelLookper.getValueAt(iLp,
0).toString();
            nmL =
modelLookper.getValueAt(iLp,
1).toString();
            jL =
modelLookper.getValueAt(iLp,
2).toString();
            hL =
modelLookper.getValueAt(iLp,
3).toString();
            ketL =
modelLookper.getValueAt(iLp,
4).toString();

```

```

        System.out.println(kdL);

```

```

        System.out.println(nmL);
        System.out.println(jL);
        System.out.println(hL);
        System.out.println(ketL);
    }

```

```

        private void
formWindowClosed(java.awt.event.Wi
ndowEvent evt) {
            // TODO add your handling
code here:
            o = true;
        }

        private void
formWindowActivated(java.awt.event
.WindowEvent evt) {
            // TODO add your handling
code here:
            o = false;
        }

```

JenisPerawatan.java

```

public final class JenisPerawatan
extends javax.swing.JFrame {
    private DefaultTableModel
modelJenis;
    private TableCellRenderer merah =
new WarnaKodeMerah();
    private Sounderror PlaySound = new
Sounderror();
    boolean ckj;
    String nokd;

```

```

    public JenisPerawatan() {
        initComponents();
        modelJenis = new
DefaultTableModel(){
            @Override
            public boolean
isCellEditable (int row, int
column){
                return false;
            }
        };
    }

```

```

    modelJenis.addColumn("Kode
Jenis");

    modelJenis.addColumn("Jenis
Perawatan");

    tblJenis.setModel(modelJenis);
    TableColumn colj0 =
tblJenis.getColumnModel().getColum
n(0);

```

```

        TableColumn colj1 =
tblJenis.getColumnModel().getColumn
n(1);
        for(int a = 0; a <
tblJenis.getColumnCount();a++)
        {

tblJenis.getColumnModel().getColumn
n(a).setCellRenderer(new
WarnaKolom());
        }

colj0.setPreferredWidth(40);
colj1.setPreferredWidth(150);

colj0.setCellRenderer(merah);

tblJenis.setSelectionBackground(Co
lor.LIGHT_GRAY);

tblJenis.setGridColor(Color.yellow
);

        loadData();
        kOsongkanJ();
        aktJ1(false);
        this.setTitle("Data Jenis
Perawatan Zahwa");
        this.setLocation(600,
250);
    }

private void kOsongkanJ()
{
    tfKdJ.setText("");
    tfNmJ.setText("");
}
private void aktJ1(boolean k)
{
    btEdit.setEnabled(k);
    btHapus.setEnabled(k);
    btSimpan.setEnabled(k);
    tfKdJ.setEnabled(k);
    tfNmJ.setEnabled(k);
}
private void aktiJ2(boolean z)
{
    btSimpan.setEnabled(z);
    tfKdJ.setEnabled(z);
    tfNmJ.setEnabled(z);
}
private void aktiJ3(boolean y)
{
    btTambah.setEnabled(y);
    btEdit.setEnabled(y);
    btHapus.setEnabled(y);
    tfKdJ.setEnabled(y);
    tfNmJ.setEnabled(y);

```

```

    }
private void nomorJ()
{
    try
    {
        Connection con =
Libra.KonekZahwa.getkoneksi();
        Statement sNJ =
con.createStatement();
        String sqlNj ="select
kd_Jenis as nokd from jenis_prw
ORDER by kd_Jenis desc ";
        ResultSet rsj =
sNJ.executeQuery(sqlNj);
        if(rsj.next())
        {
            String noJ =
rsj.getString("nokd");
            String noJa =
noJ.substring(1 , 4);
            int noJx =
Integer.parseInt(noJa)+1;

            if (noJx > 999)
            {

PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Data Over, telah mencapai
maksimal!");

tfKdJ.setEnabled(false);

tfNmJ.setEnabled(false);

btSimpan.setEnabled(false);
            }
            if (noJx > 0)
            {
                tfKdJ.setText("J00"
+noJx);
            }
            if (noJx >= 10)
            {
                tfKdJ.setText("J0"
+noJx);
            }
            if (noJx >= 100)
            {
                tfKdJ.setText("J"
+noJx);
            }
        }
        else
        {

tfKdJ.setText("J000");
        }
        rsj.close();
        sNJ.close();
    }
}

```

```

    }
    catch(SQLException e)
    {

System.out.println("Terjadi error
di ambil data nomor Jenis");
    }
    tfNmJ.requestFocus();
}
private void loadData()
{

modelJenis.getDataVector().removeA
llElements();

modelJenis.fireTableDataChanged();
    try
    {
        Connection con =
        Libra.KonekZahwa.getkoneksi();
        Statement sj =
        con.createStatement();
        String sql ="select *
        from jenis_prw ORDER by kd_Jenis
        ASC ";
        ResultSet rsj =
        sj.executeQuery(sql);
        while(rsj.next())
        {
            Object[] o = new
Object[2];
            o[0] =
            rsj.getString("kd_Jenis");
            o[1] =
            rsj.getString("Nama");

modelJenis.addRow(o);
        }
        rsj.close();
        sj.close();
    }
    catch(SQLException e)
    {

System.out.println("Terjadi error
di load data jenis");
    }

}
private void cekHps(){
    if
    ("".equals(tfKdJ.getText()))
    {
        PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Kode Jenis belum dipilih");

tblJenis.requestFocus();
        return;
    }
    nokd = tfKdJ.getText();

```

```

    try
    {
        Connection con =
        Libra.KonekZahwa.getkoneksi();
        Statement sj =
        con.createStatement();
        String sql ="select *
        from perawatan WHERE Kd_Jenis LIKE
        '" + nokd+"'";

        ResultSet rsj =
        sj.executeQuery(sql);
        if(rsj.next())
        {
            String n =
            rsj.getString("Kd_Jenis");

            if(nokd.equals(n)){

System.out.println("Kd Jenis =
"+n" , digunakan !!!");

                ckj = false;

            btHapus.setEnabled(false);

                return;
            }
            else{
                ckj = true;

            btHapus.setEnabled(true);
            }
        }
        rsj.close();
        sj.close();
    }
    catch(SQLException e)
    {

System.out.println("Terjadi error
di load perawatan");
    }

}

private void
btTambahActionPerformed(java.awt.e
vent.ActionEvent evt) {
    // TODO add your handling
code here:
    aktJ1(false);
    aktiJ2(true);
    kOsongkanJ();
    nomorJ();

}

```

```

        private void
tblJenisMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling
code here:
    aktiJ2(false);
    aktiJ3(true);
    int iJ =
tblJenis.getSelectedRow();
    if (iJ == -1)
    {
        return;
    }
    String kdJs =
modelJenis.getValueAt(iJ,
0).toString();
    String nmjs =
modelJenis.getValueAt(iJ,
1).toString();

    tfKdJ.setText(kdJs);
    tfNmJ.setText(nmjs);
    cekHps();
}

```

```

        private void
btEditActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling
code here:
    String kdJb =
tfKdJ.getText();
    String jnBr =
tfNmJ.getText();
    try
    {
        Connection cJb =
Libra.KonekZahwa.getkoneksi();

        String sqlJb = "UPDATE
jenis_prw set Nama = ? where
kd_Jenis = '"+tfKdJ.getText()+"'";
        PreparedStatement pJb =
cJb.prepareStatement(sqlJb);

        pJb.setString(1 , jnBr);
        pJb.executeUpdate();

```

```

JOptionPane.showMessageDialog(this
, "Data "+kdJb+ " Telah DiUbah
!!!");

    pJb.close();
}

catch(SQLException eJh)
{

```

```

System.err.println("Error tidak
bisa diUbah !!!");

```

```

    }

finally
{
    loadData();
    kOsongkanJ();
}
}

```

```

        private void
btSimpanActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling
code here:
    if
("".equals(tfKdJ.getText()))
    {
        PlaySound.PlaySound();

```

```

JOptionPane.showMessageDialog(this
, "Kode Jenis Perawatan mohon
diisi");

    tfKdJ.requestFocus();
    return;
}

if
("".equals(tfNmJ.getText()))
{
    PlaySound.PlaySound();

```

```

JOptionPane.showMessageDialog(this
, "Nama Jenis Perawatan mohon
diisi");

    tfNmJ.requestFocus();
    return;
}

```

```

    String kdj =
tfKdJ.getText();
    String nmj =
tfNmJ.getText();
    try
    {
        Connection cinJ =
Libra.KonekZahwa.getkoneksi();
        String sqlJ = "INSERT
INTO jenis_prw VALUES ( ?, ?)";
        PreparedStatement pJ =
cinJ.prepareStatement(sqlJ);

```

```

        pJ.setString(1,kdj );
        pJ.setString(2,nmj );
        pJ.executeUpdate();
        pJ.close();

```

```

JOptionPane.showMessageDialog(this
, "Data Telah DiSimpan");
}

catch(SQLException ex)
{

```



```

        System.out.println("Proses Tambah
        Error");
    }
    finally
    {
        loadData();
        kOsongkanJ();
    }
}

private void
btHapusActionPerformed(java.awt.ev
ent.ActionEvent evt) {
    // TODO add your handling
    code here:
    int iJh =
tblJenis.getSelectedRow();
    if (iJh == -1)
    {
        PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Silahkan pilih data yang akan
dihapus!");
        return;
    }

    String kdJh = (String)
modelJenis.getValueAt(iJh, 0);

        try
        {
            Connection cJh =
Libra.KonekZahwa.getkoneksi();
            String sqlJh =
"DELETE FROM jenis_prw WHERE
kd_Jenis = ?";
            PreparedStatement
pJh = cJh.prepareStatement(sqlJh);

            pJh.setString(1 ,
kdJh);

pJh.executeUpdate();
            pJh.close();

JOptionPane.showMessageDialog(this
, "Data "+kdJh+ " Telah DiHapus
!!!");
        }
        catch (SQLException
eJh)
        {

System.err.println("Error tidak
bisa diHapus !!!");
        }
    finally
    {

```

```

        loadData();
        kOsongkanJ();
    }
}

```

DataPerawatan.java

```

public final class DataPerawatan
extends javax.swing.JFrame {
    private DefaultTableModel
modelPerawatan;
    private TableCellRenderer kanan =
new renderingKanan();
    private TableCellRenderer merah =
new WarnaKodeMerah();
    private Sounderror PlaySound = new
Sounderror();
    String [] kat = {"A","B","C"};
    String prtsr, harga = "0", kate
, nokd;
    String awal;
    StringBuilder awall;
    Object [] har;
    boolean ckh, cnm = false;
    /**
     * Creates new form dataBarang
     */
    public DataPerawatan() {
        initComponents();
        modelPerawatan = new
DefaultTableModel(){
            @Override
            public boolean
isCellEditable (int row, int
coloumn){
                return false;
            }
        };

modelPerawatan.addColumn("Kode
Prwt");

modelPerawatan.addColumn("Nama
Perawatan");

modelPerawatan.addColumn("Jenis
Prwt");

modelPerawatan.addColumn("Harga");

modelPerawatan.addColumn("Kategori
");

modelPerawatan.addColumn("Keterang
an");

tbDaPer.setModel(modelPerawatan);

```

```

tbDaPer.setAutoResizeMode(JTable.A
UTO_RESIZE_OFF);
    TableColumn col0 =
tbDaPer.getColumnModel().getColumn
(0);
    TableColumn col1 =
tbDaPer.getColumnModel().getColumn
(1);
    TableColumn col2 =
tbDaPer.getColumnModel().getColumn
(2);
    TableColumn col3 =
tbDaPer.getColumnModel().getColumn
(3);
    TableColumn col4 =
tbDaPer.getColumnModel().getColumn
(4);
    TableColumn col5 =
tbDaPer.getColumnModel().getColumn
(5);
    for(int a = 0; a <
tbDaPer.getColumnCount();a++)
    {

tbDaPer.getColumnModel().getColumn
(a).setCellRenderer(new
WarnaKolom());
    }

col0.setPreferredWidth(70);
col1.setPreferredWidth(150);
col2.setPreferredWidth(70);
col3.setPreferredWidth(70);
col4.setPreferredWidth(70);
col5.setPreferredWidth(200);
col3.setCellRenderer(kanan);
col0.setCellRenderer(merah);
col2.setCellRenderer(merah);
    aktP1(false);
    kate();
    loadJper();
    loadPerawatan();
    kOsongkanPer();

    this.setTitle("Data
Perawatan Zahwa");
    this.setLocation(600,
250);
    }

private void kOsongkanPer()
{

    cbJper.setSelectedIndex(-1);
    cbkat.setSelectedIndex(-1);
    tfKdper.setText("");
    tfNmper.setText("");
    tfHaper.setText("");
    tfKetper.setText("");
}
private void aktP1(boolean a)
{
    btEdper.setEnabled(a);
    btHapper.setEnabled(a);
    btSimper.setEnabled(a);
    tfKdper.setEnabled(a);
    tfNmper.setEnabled(a);
    cbJper.setEnabled(a);
    cbkat.setEnabled(a);
    tfHaper.setEnabled(a);
    tfKetper.setEnabled(a);
}
private void aktiP2(boolean b)
{
    tfHaper.setEnabled(b);
    tfKetper.setEnabled(b);
    btSimper.setEnabled(b);
    tfKdper.setEnabled(b);
    tfNmper.setEnabled(b);
    cbJper.setEnabled(b);
    cbkat.setEnabled(b);
}
private void aktiP3(boolean c)
{
    btTamper.setEnabled(c);
    btEdper.setEnabled(c);
    btHapper.setEnabled(c);
    tfKdper.setEnabled(c);
    tfNmper.setEnabled(c);
    cbJper.setEnabled(c);
    cbkat.setEnabled(c);
    tfHaper.setEnabled(c);
    tfKetper.setEnabled(c);
}
private void nomorPer()
{
    try
    {
        Connection conPer =
Libra.KonekZahwa.getkoneksi();
        Statement sNP =
conPer.createStatement();

        String sqlNp ="select Kd_Prw
as nokdper from perawatan ORDER by
Kd_Prw desc ";
        ResultSet rsp =
sNP.executeQuery(sqlNp);

        if(rsp.next())
        {
            String noP =
rsp.getString("nokdper");

```

```

        String noPa =
noP.substring(1 , 4);
        int noPx =
Integer.parseInt(noPa)+1;

        if (noPx > 199)
        {

JOptionPane.showMessageDialog(this
, "Data Perawatan Over, telah
mencapai maksimal!");

tfKdper.setEnabled(false);

tfNmper.setEnabled(false);

btSimper.setEnabled(false);
        }
        if (noPx > 0)
        {
            tfKdper.setText("P00"
+noPx);
        }
        if (noPx >= 10)
        {
            tfKdper.setText("P0"
+noPx);
        }
        if (noPx >= 100)
        {
            tfKdper.setText("P"
+noPx);
        }
        }
        else
        {
            tfKdper.setText("P000");
        }
        rsp.close();
        sNP.close();
    }
    catch(SQLException e)
    {

System.out.println("Terjadi error
di ambil data nomor Perawatan");
    }
    tfNmper.requestFocus();
}
private void loadJper()
{
    cbJper.removeAllItems();
    try
    {
        Connection con =
        Libra.KonekZahwa.getkoneksi();
        Statement sj =
        con.createStatement();
        String sql ="select *
        from jenis_prw ";

```

```

        ResultSet rsj =
        sj.executeQuery(sql);
        while(rsj.next())
        {
            Object[] ob = new
            Object[2];
            ob[0] =
            rsj.getString("kd_Jenis");

            cbJper.addItem(ob[0]);
        }
        rsj.close();
        sj.close();
    }
    catch(SQLException e)
    {

System.out.println("Terjadi error
di load data jenis per");
    }
}
private void loadPerawatan()
{
    modelPerawatan.getDataVector().rem
oveAllElements();

    modelPerawatan.fireTableDataChange
d();
    try
    {
        Connection cper =
        Libra.KonekZahwa.getkoneksi();
        Statement sper =
        cper.createStatement();

        String sqlper ="select
        * from perawatan ORDER by Kd_Jenis
        ASC ";

        ResultSet rsper =
        sper.executeQuery(sqlper);

        while(rsper.next())
        {
            Object[] oper =
            new Object[6];
            oper[0] =
            rsper.getString("Kd_Prw");
            oper[1] =
            rsper.getString("Nama");
            oper[2] =
            rsper.getString("kd_Jenis");
            oper[3] =
            rsper.getString("Harga");
            String pr =
            rsper.getString("Kategori");
            if (pr == null ||
            "".equals(pr)) {
                oper[4] = "-";
            }
            else

```

```

        oper[4] =
rsper.getString("Kategori");
        oper[5] =
rsper.getString("Keterangan");

modelPerawatan.addRow(oper);
    }
    rsper.close();
    sper.close();
}
catch (SQLException e)
{

System.out.println("Terjadi error
di load data perawatan");
}
}
private void loadnmPrwt()
{
    if
    ("".equals(tfNmper.getText()))
    {
        PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Nama Perawatan mohon diisi");

tfNmper.requestFocus();
        return;
    }
    prtsr = tfNmper.getText();

    try
    {
        Connection con =
        Libra.KonekZahwa.getkoneksi();
        Statement sj =
        con.createStatement();
        String sql ="select *
from perawatan WHERE Nama LIKE '"
+ prtsr+"'";

        ResultSet rsj =
        sj.executeQuery(sql);
        if(rsj.next())
        {
            String n =
            rsj.getString("Nama");
            String k =
            rsj.getString("Kd_Jenis");
            kate =
            rsj.getString("Kategori");
            harga =
            rsj.getString("Harga");
            String kr =
            rsj.getString("Keterangan");

            if(prtsr.equals(n)){

JOptionPane.showMessageDialog(this

```

```

, "Sudah Ada !", "Warning !",
JOptionPane.WARNING_MESSAGE);
        cnm = true;

        cbJper.setSelectedItem(k);

        cbkat.setEnabled(true);

        cbkat.requestFocus();

        tfHaper.setText(harga);

        tfKetper.setText(kr);
            return;
        }
        else
            cnm = false;

        cbkat.requestFocus();

    }
    rsj.close();
    sj.close();
}
catch (SQLException e)
{

System.out.println("Terjadi error
di load Nm Prw");
}
}
private void KatPil(){

        if(kate == null){
            awal = prtsr;
            tfNmper.setText(awal+"
"+cbkat.getSelectedItem());
        }
        if("A".equals(kate)){
            awal = prtsr;
            awal1 = new
            StringBuilder(awal);
            if(awal != null&&
            awal1.length() != 0){
                int jum =
                awal.length();
                int jum2 =
                awal1.length();

                awal1.deleteCharAt(jum2-1);
                awal =
                String.valueOf(awal1);

                tfNmper.setText(awal+""+cbkat.getS
                electedItem());
                tfHaper.requestFocus();
            }
        }
        if("B".equals(kate)){
            awal = prtsr;

```

```

        awal1 = new
StringBuilder(awal);
        if(awal != null&&
awal1.length() != 0){
            int jum =
awal.length();
            int jum2 =
awal1.length();

awal1.deleteCharAt(jum2-1);
            awal =
String.valueOf(awal1);

tfNmper.setText(awal+""+cbkat.getS
electedItem());
            tfHaper.requestFocus();
        }
        if("C".equals(kate)){
            awal = prtsr;
            awal1 = new
StringBuilder(awal);
            if(awal != null&&
awal1.length() != 0){
                int jum =
awal.length();
                int jum2 =
awal1.length();

awal1.deleteCharAt(jum2-1);
                    awal =
String.valueOf(awal1);

tfNmper.setText(awal+""+cbkat.getS
electedItem());
                    tfHaper.requestFocus();
            }
        }
    }
    private void kate(){
        cbkat.removeAllItems();
        for (int yu = 0; yu <
kat.length; yu++){
            cbkat.addItem(kat[yu]);
        }
    }
    private void hargaBaru(){
        String hrg =
tfHaper.getText();
        int hrg2 =
Integer.parseInt(hrg);
        int harga2 =
Integer.parseInt(harga);
        if(hrg2 <= harga2){

JOptionPane.showMessageDialog(this
,"Harga yang di masukkan kurang"
,"Perhatian",JOptionPane.WARNING_M
ESSAGE);
            tfHaper.requestFocus();

```

```

        tfHaper.setText("");
    }
}
private void cekHapus(){
    if
("".equals(tfKdper.getText()))
    {
        PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
,"Kode Perawatan belum dipilih");

tbDaPer.requestFocus();
        return;
    }
    nokd = tfKdper.getText();

    try
    {
        Connection con =
Libra.KonekZahwa.getkoneksi();
        Statement sj =
con.createStatement();
        String sql ="select *
from dtransaksi WHERE Kd_Prw LIKE
'" + nokd+"'";

        ResultSet rsj =
sj.executeQuery(sql);
        if(rsj.next())
        {
            String n =
rsj.getString("Kd_Prw");

            if(nokd.equals(n)){

System.out.println("Kd Prw = "+n+"
, Ada transaksi !!!");

                ckh = false;

btHapper.setEnabled(false);

                return;
            }
            else
                ckh = true;

btHapper.setEnabled(true);

        }
        rsj.close();
        sj.close();
    }
    catch(SQLException e)
    {

System.out.println("Terjadi error
di load dtransaksi");
    }
}

```

```

}

private void
btTamperActionPerformed(java.awt.e
vent.ActionEvent evt) {
    // TODO add your handling
code here:
    aktP1(false);
    aktiP2(true);
    kOsongkanPer();
    nomorPer();
    cnm = false;
}

private void
jTabbedPaneMouseClicked(java.awt.
event.MouseEvent evt) {
    // TODO add your handling
code here:
    btTamper.requestFocus();
}

private void
tbDaPerMouseClicked(java.awt.event
.MouseEvent evt) {
    // TODO add your handling
code here:
    aktiP2(false);
    aktiP3(true);
    int iP =
tbDaPer.getSelectedRow();
    if (iP == -1)
    {
        return;
    }
    String kdPs =
modelPerawatan.getValueAt(iP,
0).toString();
    String nmPs =
modelPerawatan.getValueAt(iP,
1).toString();
    String jPs =
modelPerawatan.getValueAt(iP,
2).toString();
    String hgP =
modelPerawatan.getValueAt(iP,
3).toString();
    String katP =
modelPerawatan.getValueAt(iP,
4).toString();
    String ketP =
modelPerawatan.getValueAt(iP,
5).toString();
    if( "-".equals(katP) ){

cbkat.setSelectedIndex(-1);

cbkat.setEnabled(false);
    }

else

cbkat.setSelectedItem(katP);
    int hgPs =
Integer.parseInt(hgP);
    tfKdper.setText(kdPs);
    tfNmper.setText(nmPs);

cbJper.setSelectedItem(jPs);
    tfHaper.setText(""+hgPs);
    tfKetper.setText(ketP);
    cekHapus();
}

private void
btSimperActionPerformed(java.awt.e
vent.ActionEvent evt) {
    // TODO add your handling
code here:
    int cph1 =
cbJper.getSelectedIndex();
    String hBaru =
tfHaper.getText();
    int hBaru2 =
Integer.parseInt(hBaru);
    int harga2 =
Integer.parseInt(harga);
    if
("".equals(tfNmper.getText()))
    {
        PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Nama Perawatan mohon diisi");
        tfNmper.requestFocus();
        return;
    }
    if (cph1==-1)
    {
        PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Jenis Perawatan mohon
dipilih");
        cbJper.requestFocus();
        return;
    }
    if
("".equals(tfHaper.getText()))
    {
        PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Harga Perawatan mohon diisi");
        tfHaper.requestFocus();
        return;
    }
    loadnmPrwt();
    if(cnm == false){

```

```

        String kdp =
tfKdper.getText();
        String nmp =
tfNmper.getText();
        String jp = (String)
cbJper.getSelectedItem();
        String ktt = (String)
cbkat.getSelectedItem();
        String hap =
tfHaper.getText();
        String kup =
tfKetper.getText();

        if(hBaru2 <= harga2){

JOptionPane.showMessageDialog(this
, "Harga terlalu rendah dari harga
normal","Perhatian
!",JOptionPane.WARNING_MESSAGE);

tfHaper.requestFocus();
        tfHaper.setText("
");
        return;
    }
    try
    {
        Connection cinP =
Libra.KonekZahwa.getkoneksi();
        String sqlP = "INSERT
INTO perawatan VALUES ( ?, ?, ?,
?, ?, ?)";

        PreparedStatement pP =
cinP.prepareStatement(sqlP);

        pP.setString(1,kdp );
        pP.setString(2,nmp );
        pP.setString(3,jp );
        pP.setString(4, ktt);
        pP.setString(5,hap );
        pP.setString(6, kup);

        pP.executeUpdate();
        pP.close();

JOptionPane.showMessageDialog(this
, "Data Perawatan Telah
DiSimpan");
    }
    catch(SQLException ex)
    {

System.out.println("Proses Tambah
Error");
    }
    finally
    {
        aktP1(false);
        loadPerawatan();
        kOsongkanPer();
    }

```

```

        cnm = false;
    }
}

private void
btHapperActionPerformed(java.awt.e
vent.ActionEvent evt) {
    // TODO add your handling
code here:
    int iPh =
tbDaPer.getSelectedRow();
    if (iPh == -1)
    {
        PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Silahkan pilih data pada Tabel
Perawatan!");
        return;
    }
    String kdPh = (String)
modelPerawatan.getValueAt(iPh, 0);
    try
    {
        Connection cPh =
Libra.KonekZahwa.getkoneksi();
        String sqlPh = "DELETE
FROM perawatan WHERE Kd_Prw = ?";
        PreparedStatement pPh =
cPh.prepareStatement(sqlPh);

        pPh.setString(1 , kdPh);
        pPh.executeUpdate();
        pPh.close();

JOptionPane.showMessageDialog(this
, "Data "+kdPh+ " Telah DiHapus
!!!");
    }
    catch(SQLException ePh)
    {

System.err.println("Error Data
Perawatan tidak bisa diHapus
!!!");
    }
    finally
    {
        loadPerawatan();
        kOsongkanPer();
    }
}

private void
btCekActionPerformed(java.awt.even
t.ActionEvent evt) {
    // TODO add your handling
code here:
    lookJenis lJ = new
lookJenis();

```

```

        lJ.setVisible(true);
    }

    private void
    btEdperActionPerformed(java.awt.ev
ent.ActionEvent evt) {
        // TODO add your handling
code here:
        String kdP =
    tfKdper.getText();
        String nmP =
    tfNmper.getText();
        String jP =
    cbJper.getSelectedItem().toString(
);
        String hjP =
    tfHaper.getText();
        String kP =
    tfKetper.getText();
        int cph1 =
    cbJper.getSelectedIndex();
        if
    ("".equals(tfNmper.getText()))
        {
            PlaySound.PlaySound();

    JOptionPane.showMessageDialog(this
, "Nama Perawatan mohon diisi");
            tfNmper.requestFocus();
            return;
        }
        if (cph1==-1)
        {
            PlaySound.PlaySound();

    JOptionPane.showMessageDialog(this
, "Jenis Perawatan mohon
dipilih");
            cbJper.requestFocus();
            return;
        }
        if
    ("".equals(tfHaper.getText()))
        {
            PlaySound.PlaySound();

    JOptionPane.showMessageDialog(this
, "Harga Perawatan mohon diisi");
            tfHaper.requestFocus();
            return;
        }
        try
        {
            Connection cdpr =
    Libra.KonekZahwa.getkoneksi();

            String sqldPr = "UPDATE
perawatan set Nama = ? , Kd_Jenis
= ? , "

```

```

        + "Harga = ? ,
Keterangan = ? where Kd_Prw =
'" + tfKdper.getText() + "'";
        PreparedStatement pDp =
    cdpr.prepareStatement(sqldPr);

        pDp.setString(1 , nmP);
        pDp.setString(2 , jP);
        pDp.setString(3 ,
    hjP);
        pDp.setString(4 ,
    kP);
        pDp.executeUpdate();

    JOptionPane.showMessageDialog(this
, "Data "+kdP+ " Telah DiUbah
!!!");
        pDp.close();
    }

    catch(SQLException eDh)
    {

        System.err.println("Error Data
Perawatan tidak bisa diUbah !!!");
    }

    finally
    {
        loadPerawatan();
        kOsongkanPer();
    }
}

    private void
    tfNmperKeyPressed(java.awt.event.K
eyEvent evt) {
        // TODO add your handling
code here:
        int ascii=evt.getKeyCode();
        if (ascii==10)
        {
            loadnmPrwt();
        }
    }

    private void
    cbkatActionPerformed(java.awt.even
t.ActionEvent evt) {
        // TODO add your handling
code here:
        tfHaper.setText("");
        cnm = false;
        KatPil();
    }

    private void
    tfHaperKeyPressed(java.awt.event.K
eyEvent evt) {
        // TODO add your handling
code here:

```



```

int ascii=evt.getKeyCode();
if (ascii==10)
{
    hargaBaru();
}
}

```

lookJenis.java

```

public final class lookJenis
extends javax.swing.JFrame {
private DefaultTableModel
modelLookJ;
/**
 * Creates new form lookJenis
 */
public lookJenis() {
    initComponents();
    modelLookJ = new
DefaultTableModel(){
    @Override
    public boolean
isCellEditable (int row, int
coloumn){
        return false;
    }
};
    modelLookJ.addColumn("Kd
Jenis");

modelLookJ.addColumn("Jenis
Perawatan");

tbLookJ.setModel(modelLookJ);
loadLookJ();
this.setTitle("Lihat Data
Jenis Perawatan");
this.setLocation(600,
250);
}
private void loadLookJ()
{
modelLookJ.getDataVector().removeA
llElements();

modelLookJ.fireTableDataChanged();
try
{
    Connection clj =
Libra.KonekZahwa.getkoneksi();
    Statement slj =
clj.createStatement();
    String sqllj ="select
* from jenis_prw";
    ResultSet rslj =
slj.executeQuery(sqllj);

```

```

while(rslj.next())
{
    Object[] olj = new
Object[2];
        olj[0] =
rslj.getString("kd_Jenis");
        olj[1] =
rslj.getString("Nama");

modelLookJ.addRow(olj);
}
    rslj.close();
    slj.close();
}
catch(SQLException e)
{
    System.out.println("Terjadi error
di load data jenis");
}
}

```

DataPelanggan.java

```

public final class DataPelanggan
extends javax.swing.JFrame {
private DefaultTableModel
modelPelanggan;
private TableCellRenderer merah =
new WarnaKodeMerah();
private TableCellRenderer
kolomWarna = new WarnaKodeMerah();
String [] bulan =
{"","JANUARI","FEBRUARI","MARET","
APRIL","MEI","JUNI","JULI"
,"AGUSTUS","SEPTEMBER","OKTOBER","
NOVEMBER","DESEMBER"};
String tglp ;
int blnp ;
String thnp, nokd ;
Date now;
SimpleDateFormat frm = new
SimpleDateFormat("yyyy/MM/dd");
String kerPs, alPs, koPs, ntPs,
hpPs,mmPs;
private Sounderror PlaySound = new
Sounderror();
boolean ckp;
int akt,panDt, a = -1;
String user;
String [] ket;

/**
 * Creates new form
dataPelanggan
 */
public DataPelanggan() {
    initComponents();

```

```

    }

    public DataPelanggan(int akt,
String user) {
        this.akt = akt;
        this.user = user;
        initComponents();
        modelPelanggan = new
DefaultTableModel() {
            @Override
            public boolean
isCellEditable (int row, int
coloumn){
                return false;
            }
        };

modelPelanggan.addColumn("Kd
Plgn.");

modelPelanggan.addColumn("Nama
Plgn");

modelPelanggan.addColumn("Tgl.
Lahir");

modelPelanggan.addColumn("Gender")
;

modelPelanggan.addColumn("Pekerjaa
n");

modelPelanggan.addColumn("Alamat
Pelanggan");

modelPelanggan.addColumn("Kota");

modelPelanggan.addColumn("Telepon"
);

modelPelanggan.addColumn("Hp");

modelPelanggan.addColumn("Status")
;

tbPlgn.setModel(modelPelanggan);

tbPlgn.setAutoResizeMode(JTable.AU
TO_RESIZE_OFF);
        TableColumn colpe0 =
tbPlgn.getColumnModel().getColumn(
0);
        TableColumn colpe1 =
tbPlgn.getColumnModel().getColumn(
1);
        TableColumn colpe2 =
tbPlgn.getColumnModel().getColumn(
2);

        TableColumn colpe3 =
tbPlgn.getColumnModel().getColumn(
3);
        TableColumn colpe4 =
tbPlgn.getColumnModel().getColumn(
4);
        TableColumn colpe5 =
tbPlgn.getColumnModel().getColumn(
5);
        TableColumn colpe6 =
tbPlgn.getColumnModel().getColumn(
6);
        TableColumn colpe7 =
tbPlgn.getColumnModel().getColumn(
7);
        TableColumn colpe8 =
tbPlgn.getColumnModel().getColumn(
8);
        for(int a = 0; a <
tbPlgn.getColumnCount();a++)
        {
            tbPlgn.getColumnModel().getColumn(
a).setCellRenderer(new
WarnaKolom());
        }

colpe0.setPreferredWidth(60);

colpe1.setPreferredWidth(130);

colpe2.setPreferredWidth(80);

colpe3.setPreferredWidth(60);

colpe4.setPreferredWidth(85);

colpe5.setPreferredWidth(200);

colpe6.setPreferredWidth(75);

colpe7.setPreferredWidth(90);

colpe8.setPreferredWidth(110);

colpe0.setCellRenderer(merah);

tbPlgn.setSelectionBackground(Colo
r.LIGHT_GRAY);

tbPlgn.setGridColor(Color.yellow);

tbPlgn.setBackground(Color.pink);
        btGjk.add(rdPria);
        btGjk.add(rdWanita);
        itemTgl();
        itemBln();
        itemThn();
        panjData();
        ketdisk();
        itemkerja();

```

```

        loadPel();
        kOsongkanPel();
        aktP1(false);

        this.setTitle("Data
Pelanggan Zahwa");
        this.setLocation(600,
250);
        lbAdmin.setText(user);
        lbAdmin.setVisible(false);
    }

    public void setAkt(int akt) {
        this.akt = akt;
    }

    public void setUser(String
user) {
        this.user = user;
    }
    private void onakt(){
        if(akt == 0){

btHappel.setEnabled(true);
        btEdpel.setEnabled(true);
        cbMember.setEnabled(true);
        }
        if(akt == 1){

cbMember.setEnabled(false);
        }
    }
    private void kOsongkanPel()
    {
        tfKdpel.setText("");
        tfNmpel.setText("");
        tfAlpel.setText("");
        tfKopel.setText("");
        tfNopel.setText("");
        tfHp.setText("");
        btGjk.clearSelection();
        cbTgl.setSelectedIndex(-1);
        cbBln.setSelectedIndex(0);
        cbThn.setSelectedIndex(-1);

cbPekerjaan.setSelectedIndex(-1);
        cbMember.setSelected(false);
    }
    private void aktP1(boolean f)
    {
        btEdpel.setEnabled(f);
        btHappel.setEnabled(f);
        btSimpel.setEnabled(f);
        tfKdpel.setEnabled(f);
        tfNmpel.setEnabled(f);
        tfAlpel.setEnabled(f);
        tfKopel.setEnabled(f);
        tfNopel.setEnabled(f);
        tfHp.setEnabled(f);
        rdPria.setEnabled(f);

        rdWanita.setEnabled(f);
        cbTgl.setEnabled(f);
        cbBln.setEnabled(f);
        cbThn.setEnabled(f);
        cbPekerjaan.setEnabled(f);
        cbMember.setEnabled(f);
    }
    private void aktiP2(boolean g)
    {
        btSimpel.setEnabled(g);
        tfKdpel.setEnabled(g);
        tfNmpel.setEnabled(g);
        tfAlpel.setEnabled(g);
        tfKopel.setEnabled(g);
        tfNopel.setEnabled(g);
        tfHp.setEnabled(g);
        rdPria.setEnabled(g);
        rdWanita.setEnabled(g);
        cbTgl.setEnabled(g);
        cbBln.setEnabled(g);
        cbThn.setEnabled(g);
        cbPekerjaan.setEnabled(g);
        cbMember.setEnabled(g);
    }
    private void aktiP3(boolean h)
    {
        btTampel.setEnabled(h);
        btEdpel.setEnabled(h);
        btHappel.setEnabled(h);
        tfKdpel.setEnabled(h);
        tfNmpel.setEnabled(h);
        tfAlpel.setEnabled(h);
        tfKopel.setEnabled(h);
        tfNopel.setEnabled(h);
        tfHp.setEnabled(h);
        rdPria.setEnabled(h);
        rdWanita.setEnabled(h);
        cbTgl.setEnabled(h);
        cbBln.setEnabled(h);
        cbThn.setEnabled(h);
        cbPekerjaan.setEnabled(h);
        cbMember.setEnabled(h);
    }
    private void itemTgl(){
        cbTgl.removeAllItems();
        for(int i = 1; i < 32; i++){
            cbTgl.addItem(i);
        }
    }
    private void itemBln(){
        cbBln.removeAllItems();
        for(int i = 0; i < 13; i++){
            cbBln.addItem(bulan[i]);
        }
    }
    private void itemThn(){
        cbThn.removeAllItems();
        for(int i = 1950; i < 2012;
i++){
            cbThn.addItem(i);

```

```

    }
    }
    private void itemkerja(){
        cbPekerjaan.removeAllItems();
        for(int i = 0; i <
ket.length; i++){
cbPekerjaan.addItem(ket[i]);
    }
    }
    private void nomorPel()
    {
        try
        {
            Connection conPel =
Libra.KonekZahwa.getkoneksi();
            Statement sNP =
conPel.createStatement();

            String sqlNp ="select
Kd_Konsumen as nokdpel from
konsumen ORDER by Kd_Konsumen desc
";
            ResultSet rsp =
sNP.executeQuery(sqlNp);

            if(rsp.next())
            {
                String noP =
rsp.getString("nokdpel");
                String noPa =
noP.substring(1 , 4);
                int noPx =
Integer.parseInt(noPa)+1;
                if (noPx > 999)
                {

PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Data Pelanggan Over, telah
mencapai maksimal!");

tfKdpel.setEnabled(false);

tfNmpel.setEnabled(false);

btSimpel.setEnabled(false);
                }
                if (noPx > 0)
                {
                    tfKdpel.setText("P00"
+noPx);
                }
                if (noPx >= 10)
                {
                    tfKdpel.setText("P0"
+noPx);
                }
                if (noPx >= 100)
                {

```

```

                    tfKdpel.setText("P"
+noPx);
                }
            }
            else
            {
                tfKdpel.setText("P000");
            }
            rsp.close();
            sNP.close();
        }
        catch(SQLException ep)
        {

System.out.println("Terjadi error
di ambil data nomor Pelanggan");
        }
        tfNmpel.requestFocus();
    }
    private void loadPel()
    {

modelPelanggan.getDataVector().rem
oveAllElements();

modelPelanggan.fireTableDataChange
d();
        try
        {
            Connection cPel =
Libra.KonekZahwa.getkoneksi();
            Statement spel =
cPel.createStatement();

            String sqlpel ="select
* from konsumen";
            ResultSet rspel =
spel.executeQuery(sqlpel);

            while(rspel.next())
            {
                Object[] opel =
new Object[10];
                opel[0] =
rspel.getString("Kd_Konsumen");
                opel[1] =
rspel.getString("Nama");
                opel[2] =
rspel.getString("Tanggal_Lahir");
                opel[3] =
rspel.getString("Kelamin");
                String p =
rspel.getString("Pekerjaan");
                if (p == null ||
"".equals(p)) {
                    opel[4] = "---
";
                }
                else
                {
                    opel[4] =
rspel.getString("Pekerjaan");

```

```

        String A =
rspel.getString("Alamat");
        if (A == null ||
"".equals(A)){
            opel[5] = "---
";
        }
        else
            opel[5] =
rspel.getString("Alamat");

        String K =
rspel.getString("Kota");
        if (K == null ||
"".equals(K)){
            opel[6] = "---
";
        }
        else
            opel[6] =
rspel.getString("Kota");

        String T =
rspel.getString("Telepon");
        if (T == null ||
"".equals(T)){
            opel[7] = "---
";
        }
        else
            opel[7] =
rspel.getString("Telepon");

        String H =
rspel.getString("Hp");
        if (H == null ||
"".equals(H)){
            opel[8] = "---
";
        }
        else
            opel[8] =
rspel.getString("Hp");

        String z =
rspel.getString("status");
        if (
"m".equals(z)){
            opel[9] =
"Member";
        }
        else
            if ("n".equals(z)){
                opel[9] = "Non
Member";
            }
        }

        modelPelanggan.addRow(opel);

        }
        rspel.close();
        spel.close();
    }
    catch(SQLException e)
    {

        System.out.println("Terjadi error
di load data pelanggan");
    }
    }

    private void aturtext(KeyEvent
ke, JTextField jt, int jml){
        char e = ke.getKeyChar();
        if(jt.getText().length()>jml){
            ke.consume();
        }
        if(!Character.isDigit(e)){
            ke.consume();
        }
    }

    private void jarakText(KeyEvent
k, JTextField j, int jm){
        if(j.getText().length()>jm){
            k.consume();
        }
    }

    private void cekHps(){
        if
("".equals(tfKdpel.getText()))
        {
            PlaySound.PlaySound();

            JOptionPane.showMessageDialog(this
, "Kode Pelanggan belum dipilih");
            tbPlgn.requestFocus();
            return;
        }
        nokd = tfKdpel.getText();

        try
        {
            Connection con =
Libra.KonekZahwa.getkoneksi();
            Statement sj =
con.createStatement();
            String sql ="select *
from transaksi WHERE Kd_Konsumen
LIKE '" + nokd+"'";

            ResultSet rsj =
sj.executeQuery(sql);
            if(rsj.next())
            {
                String n =
rsj.getString("Kd_Konsumen");

                if(nokd.equals(n)){

                    ckp = false;

```

```

        btHappel.setEnabled(false);

        return;
    }
    else
        ckp = true;

    btHappel.setEnabled(true);

    }
    rsj.close();
    sj.close();
}
catch (SQLException e)
{

System.out.println("Terjadi error
di load transaksi");
}
}
private void ketdisk(){
    ket = new String[panDt];
    try
    {
        Connection cJ =
        Libra.KonekZahwa.getkoneksi();
        Statement sj =
        cJ.createStatement();
        String sqlJ ="select
        a.Keterangan from diskon a";
        ResultSet rsj =
        sj.executeQuery(sqlJ);
        while(rsj.next())
        {
            a +=1;
            Object[] oJ = new
            Object[1];
            oJ[0] =
            rsj.getString("a.Keterangan");
            ket [a] = (String)
            oJ[0];

        }
        rsj.close();
        sj.close();
    }
    catch (SQLException e)
    {

System.out.println("Terjadi error
di load keterangan diskon");
}
    System.out.println("ket :
    "+ket);
}
private void panjData(){

    try {
        Connection c =
        Libra.KonekZahwa.getkoneksi();

```

```

        Statement st =
        c.createStatement();

        ResultSet rs =
        st.executeQuery("select
        COUNT(kd_diskon)from diskon ");

        while(rs.next())
        {
            panDt =
            Integer.parseInt(rs.getString(1));
        }
        rs.close();
        st.close();
    }
    catch (SQLException ex) {

System.out.println("Terjadi error
di cari panj data diskon");
}
    System.out.println("panDt :
    "+panDt);
}

private void
tbPlgnMouseClicked(java.awt.event.
MouseEvent evt) {
    // TODO add your handling
    code here:
    aktiP2(false);
    aktiP3(true);
    onakt();
    int iP =
    tbPlgn.getSelectedRow();
    if (iP == -1)
    {
        return;
    }

    String kdPs =
    modelPelanggan.getValueAt(iP,
    0).toString();
    String nmPs =
    modelPelanggan.getValueAt(iP,
    1).toString();
    String tglPs =
    modelPelanggan.getValueAt(iP,
    2).toString();
    String kelps =
    modelPelanggan.getValueAt(iP,
    3).toString();
    kerPs =
    modelPelanggan.getValueAt(iP,
    4).toString();
    if( "---".equals(kerPs) ){

cbPekerjaan.setSelectedIndex(-1);
    }
    else

```

```

cbPekerjaan.setSelectedItem(kerPs)
;

        alPs =
modelPelanggan.getValueAt(iP,
5).toString();
        if( "---".equals(alPs) ){
            tfAlpel.setText("");
        }
        else
            tfAlpel.setText(alPs);

        koPs =
modelPelanggan.getValueAt(iP,
6).toString();
        if( "---".equals(koPs) ){
            tfKopel.setText("");
        }
        else
            tfKopel.setText(koPs);

        ntPs =
modelPelanggan.getValueAt(iP,
7).toString();
        if( "---".equals(ntPs) ){
            tfNopel.setText("");
        }
        else
            tfNopel.setText(ntPs);

        hpPs =
modelPelanggan.getValueAt(iP,
8).toString();
        if( "---".equals(hpPs) ){
            tfHp.setText("");
        }
        else
            tfHp.setText(hpPs);

        mmPs =
modelPelanggan.getValueAt(iP,
9).toString();
        if( "Member".equals(mmPs)
){
cbMember.setSelected(true);
        }
        else
cbMember.setSelected(false);

            tfKdpel.setText(kdPs);
            tfNmpel.setText(nmPs);

            int tgl =
Integer.parseInt(tglPs.substring(8
, 10));

```

```

            int bul =
Integer.parseInt(tglPs.substring(5
, 7));

            int thn =
Integer.parseInt(tglPs.substring(0
, 4));

cbTgl.setSelectedItem(tgl);

cbBln.setSelectedItem(String.valueOf(bulan[bul]));

cbThn.setSelectedItem(thn);

            if(kelPs.equals("P"))
rdPria.setSelected(true);
            if(kelPs.equals("W"))
rdWanita.setSelected(true);
            cekHps();
        }

        private void
cbBlnActionPerformed(java.awt.event.ActionEvent evt) {
            // TODO add your handling
code here:
            blnp =
cbBln.getSelectedIndex();
            System.out.println("Bulan
: "+blnp);
        }

        private void
btTampelActionPerformed(java.awt.event.ActionEvent evt) {
            // TODO add your handling
code here:
            aktP1(false);
            aktiP2(true);
            onakt();
            kOsongkanPel();
            nomorPel();
        }

        private void
btEdpelActionPerformed(java.awt.event.ActionEvent evt) {
            // TODO add your handling
code here:
            String kdK =
tfKdpel.getText();
            String nmK =
tfNmpel.getText();
            String tglK =
String.valueOf(cbTgl.getSelectedItem());

```

```

        String blnK =
String.valueOf(cbBln.getSelectedIn
dex());
        String thnK =
String.valueOf(cbThn.getSelectedIt
em());
        String lhrK = thnK+"-
"+blnK+"-"+tglK;

        String kelpK ="";
        if(rdPria.isSelected()){
            kelpK = "P";
        }
        if(rdWanita.isSelected()){
            kelpK = "W";
        }

        String alK;

        if("").equals(tfAlpel.getText())){
            alK = "";
        }
        else
            alK = tfAlpel.getText();

        String koK;

        if("").equals(tfKopel.getText())){
            koK = "";
        }
        else
            koK = tfKopel.getText();

        String noK;

        if("").equals(tfNopel.getText())){
            noK = "";
        }
        else
            noK = tfNopel.getText();

        String hpK;

        if("").equals(tfHp.getText())){
            hpK = "";
        }
        else
            hpK = tfHp.getText();

        String kjK = "";

        if(cbPekerjaan.getSelectedIndex()=
=-1){
            kjK = "";
        }

        if(cbPekerjaan.getSelectedIndex())>
-1){
            kjK =
cbPekerjaan.getSelectedItem().toSt
ring();

```

```

        }
        String mmo = "";
        if(cbMember.isSelected()){
            mmo = "m";
        }
        //        else mmo = "n";
        if(cbMember.isSelected()==
false){
            mmo = "n";
        }
        System.out.println("
Status : "+mmo);
        try
        {
            Connection cpel =
Libra.KonekZahwa.getkoneksi();

            String sqlpel =
"UPDATE konsumen set Nama = ? ,
Tanggal_Lahir = ? , "
                + "Kelamin = ?
, Pekerjaan = ? , Alamat = ? , Kota
= ? , Telepon = ? , Hp = ?"
                + " , status = ?
where Kd_Konsumen =
'" + tfKdpel.getText() + "'";
            PreparedStatement ppel
= cpel.prepareStatement(sqlpel);

            ppel.setString(1 ,
nmK);
            ppel.setString(2 ,
lhrK);
            ppel.setString(3 ,
kelpK);
            ppel.setString(4 ,
kjK);
            ppel.setString(5 ,
alK);
            ppel.setString(6 ,
koK);
            ppel.setString(7 ,
noK);
            ppel.setString(8 ,
hpK);
            ppel.setString(9 ,
mmo);
            ppel.executeUpdate();

            JOptionPane.showMessageDialog(this
, "Data "+kdK+ " Telah DiUbah
!!!");
            ppel.close();
        }
        catch(SQLException epe)
        {
            System.err.println("Error Data
Pelanggan tidak bisa diUbah !!!");
        }

```



```

        finally
        {
            loadPel();
            kOsongkanPel();
        }
    }

    private void
    btHappelActionPerformed(java.awt.e
    vent.ActionEvent evt) {
        // TODO add your handling
        code here:
        int iKh =
        tbPlgn.getSelectedRow();
        if (iKh == -1)
        {
            PlaySound.PlaySound();

            JOptionPane.showMessageDialog(this
            , "Silahkan pilih data pada Tabel
            Pelanggan!");
            return;
        }
        String kdKh = (String)
        modelPelanggan.getValueAt(iKh, 0);
        try
        {
            Connection cKh =
            Libra.KonekZahwa.getkoneksi();
            String sqlKh = "DELETE
            FROM konsumen WHERE Kd_Konsumen =
            ?";
            PreparedStatement pKh
            = cKh.prepareStatement(sqlKh);
            pKh.setString(1 ,
            kdKh);
            pKh.executeUpdate();
            pKh.close();

            JOptionPane.showMessageDialog(this
            , "Data "+kdKh+ " Telah DiHapus
            !!!");
        }
        catch (SQLException eKh)
        {
            System.err.println("Error Data
            Pelanggan tidak bisa diHapus
            !!!");
        }
        finally
        {
            loadPel();
            kOsongkanPel();
        }
    }

    private void
    btSimpelActionPerformed(java.awt.e
    vent.ActionEvent evt) {

```

```

        // TODO add your handling
        code here:
        String kelpp = "";
        if (rdPria.isSelected()) {
            kelpp = "P";
        }
        if (rdWanita.isSelected()) {
            kelpp = "W";
        }
        if
        ("".equals(tfNmpel.getText()))
        {
            PlaySound.PlaySound();

            JOptionPane.showMessageDialog(this
            , "Nama Pelanggan mohon diisi");
            tfNmpel.requestFocus();
            return;
        }

        if (cbTgl.getSelectedIndex() == -1) {
            PlaySound.PlaySound();

            JOptionPane.showMessageDialog(this
            , "Tanggal Lahir mohon diisi");
            cbTgl.requestFocus();
            return;
        }

        if (cbBln.getSelectedIndex() == -1) {
            PlaySound.PlaySound();

            JOptionPane.showMessageDialog(this
            , "Bulan Lahir mohon diisi");
            cbBln.requestFocus();
            return;
        }

        if (cbThn.getSelectedIndex() == -1) {
            PlaySound.PlaySound();

            JOptionPane.showMessageDialog(this
            , "Tahun Lahir mohon diisi");
            cbThn.requestFocus();
            return;
        }

        if ("".equals(kelpp)) {
            PlaySound.PlaySound();

            JOptionPane.showMessageDialog(this
            , "Jenis Gender mohon diisi");
            rdPria.requestFocus();
            return;
        }

        String kdp =
        tfKdpel.getText();
        String nmp =
        tfNmpel.getText();

```

```

        String lhr =
thnp+"/"+blnp+"/"+tglp;
        String kej = "";
        String mm = "";

if (cbPekerjaan.getSelectedIndex()=
=-1){
            kej = "";
        }

if (cbPekerjaan.getSelectedIndex()>
-1){
            kej =
cbPekerjaan.getSelectedItem().toSt
ring();
        }
        if (cbMember.isSelected()){
            mm = "m";
        }
        else mm = "n";

        String alp ;

if ("".equals(tfAlpel.getText())){
            alp = "";
        }
        else
            alp = tfAlpel.getText();

        String kop;

if ("".equals(tfKopel.getText())){
            kop = "";
        }
        else
            kop = tfKopel.getText();

        String ntp;

if ("".equals(tfNopel.getText())){
            ntp = "";
        }
        else
            ntp = tfNopel.getText();

        String hpp;

if ("".equals(tfHp.getText())){
            hpp = "";
        }
        else
            hpp = tfHp.getText();

        try
        {
            Connection cinP =
Libra.KonekZahwa.getkoneksi();
            String sqlP = "INSERT
INTO konsumen VALUES ( ?, ?, ?, ?,
?, ?, ?, ? ,? ,?)";

```

```

        PreparedStatement pP =
cinP.prepareStatement(sqlP);
        pP.setString(1,kdp );
        pP.setString(2,nmp );
        pP.setString(3, lhr);
        pP.setString(4,kelpp
);

        pP.setString(5, kej);
        pP.setString(6,alp );
        pP.setString(7,kop );
        pP.setString(8, ntp);
        pP.setString(9, hpp);
        pP.setString(10, mm);
        pP.executeUpdate();
        pP.close();

JOptionPane.showMessageDialog(this
, "Data Pelanggan Telah
DiSimpan");
        System.out.println("
Lahir : "+lhr);
    }
    catch (SQLException exp)
    {

System.out.println("Proses Tambah
Pelanggan Error");
    }
    finally
    {
        aktPl(false);
        loadPel();
        kOsongkanPel();
    }
}

private void
cbThnActionPerformed(java.awt.even
t.ActionEvent evt) {
    // TODO add your handling
code here:
    thnp =
String.valueOf(cbThn.getSelectedIt
em());
}

private void
tfNopelKeyTyped(java.awt.event.Key
Event evt) {
    // TODO add your handling
code here:
    aturtext(evt, tfNopel, 10);
}

private void
tfHpKeyTyped(java.awt.event.KeyEve
nt evt) {
    // TODO add your handling
code here:
    aturtext(evt, tfHp, 11);
}

```

```

        private void
        tfKopelKeyTyped(java.awt.event.Key
        Event evt) {
            // TODO add your handling
            code here:
            jarakText(evt, tfKopel, 19);
        }

```

```

        private void
        cbTglActionPerformed(java.awt.even
        t.ActionEvent evt) {
            // TODO add your handling
            code here:
            tglp =
            String.valueOf(cbTgl.getSelectedIt
            em());
        }

```

```

        private void
        tfNmpelKeyTyped(java.awt.event.Key
        Event evt) {
            // TODO add your handling
            code here:
            jarakText(evt, tfNmpel, 53);
        }

```

```

        private void
        formWindowClosed(java.awt.event.Wi
        ndowEvent evt) {
            // TODO add your handling
            code here:
            mnUtama menuUta =
            new mnUtama(akt, user);

            menuUta.setAkt(akt);

            menuUta.setVisible(true);
        }

```

```

        private void
        cbMemberActionPerformed(java.awt.e
        vent.ActionEvent evt) {
            // TODO add your handling
            code here:

            if (cbMember.isSelected()) {

            cbPekerjaan.setSelectedItem(cbMemb
            er.getText());
            }
        }

```

DataKaryawan.java

```

public final class DataKaryawan
extends javax.swing.JFrame {

```

```

    private DefaultTableModel
    modelKaryawan;
    private TableCellRenderer merah =
    new WarnaKodeMerah();
    private TableCellRenderer
    kolomWarna = new WarnaKodeMerah();
    String [] bulan =
    {"", "JANUARI", "FEBRUARI", "MARET", "
    APRIL", "MEI", "JUNI", "JULI"

```

```

    , "AGUSTUS", "SEPTEMBER", "OKTOBER", "
    NOVEMBER", "DESEMBER"};

```

```

    String ntKs;
    public Sounderror PlaySound = new
    Sounderror();

```

```

    boolean ckh;

```

```

    /**
     * Creates new form
     dataKaryawan
     */

```

```

    public DataKaryawan() {
        initComponents();
        modelKaryawan = new
        DefaultTableModel() {
            @Override
            public boolean
            isCellEditable (int row, int
            coloumn){
                return false;
            }
        };

```

```

        modelKaryawan.addColumn("Kd
        Kary.");

```

```

        modelKaryawan.addColumn("Nama
        Karyawan");

```

```

        modelKaryawan.addColumn("Tgl
        Lahir");

```

```

        modelKaryawan.addColumn("Alamat");

```

```

        modelKaryawan.addColumn("Kota");

```

```

        modelKaryawan.addColumn("Telepon")
        ;

```

```

        modelKaryawan.addColumn("Hp");

```

```

        modelKaryawan.addColumn("Ktp");

```

```

        tbKar.setModel(modelKaryawan);

```

```

        tbKar.setAutoResizeMode(JTable.AUT
        O_RESIZE_OFF);

```

```

        TableColumn colka0 =
        tbKar.getColumnModel().getColumn(0
        );

```

```

        TableColumn colka1 =
tbKar.getColumnModel().getColumn(1
);
        TableColumn colka2 =
tbKar.getColumnModel().getColumn(2
);
        TableColumn colka3 =
tbKar.getColumnModel().getColumn(3
);
        TableColumn colka4 =
tbKar.getColumnModel().getColumn(4
);
        TableColumn colka5 =
tbKar.getColumnModel().getColumn(5
);
        TableColumn colka6 =
tbKar.getColumnModel().getColumn(6
);
        TableColumn colka7 =
tbKar.getColumnModel().getColumn(7
);
        for(int a = 0; a <
tbKar.getColumnCount();a++)
        {

tbKar.getColumnModel().getColumn(a
).setCellRenderer(new
WarnaKolom());
        }

```

```

colka0.setPreferredWidth(60);
colka1.setPreferredWidth(120);
colka2.setPreferredWidth(80);
colka3.setPreferredWidth(150);
colka4.setPreferredWidth(80);
colka5.setPreferredWidth(85);
colka6.setPreferredWidth(95);
colka7.setPreferredWidth(100);
colka0.setCellRenderer(merah);
        itemTgl();
        itemBln();
        itemThn();
        loadKary();
        kOsongkanKar();
        aktK1(false);
        this.setTitle("Data
Karyawan Zahwa");
        this.setLocation(600,
250);
    }

```

```

private void kOsongkanKar()
{

```

```

        tfKdkar.setText("");
        tfNmkar.setText("");
        tfAlkar.setText("");
        tfKokar.setText("");
        tfNotelkar.setText("");
        tfHp.setText("");
        tfKtp.setText("");
        cbTgl.setSelectedIndex(-1);
        cbBln.setSelectedIndex(0);
        cbThn.setSelectedIndex(-1);
    }
private void aktK1(boolean d)
{
    btEdkar.setEnabled(d);
    btHapkar.setEnabled(d);
    btSimkar.setEnabled(d);
    tfKdkar.setEnabled(d);
    tfNmkar.setEnabled(d);
    tfAlkar.setEnabled(d);
    tfKokar.setEnabled(d);
    tfNotelkar.setEnabled(d);
    tfHp.setEnabled(d);
    tfKtp.setEnabled(d);
    cbTgl.setEnabled(d);
    cbBln.setEnabled(d);
    cbThn.setEnabled(d);
}
private void aktiK2(boolean e)
{
    btSimkar.setEnabled(e);
    tfKdkar.setEnabled(e);
    tfNmkar.setEnabled(e);
    tfAlkar.setEnabled(e);
    tfKokar.setEnabled(e);
    tfNotelkar.setEnabled(e);
    tfHp.setEnabled(e);
    tfKtp.setEnabled(e);
    cbTgl.setEnabled(e);
    cbBln.setEnabled(e);
    cbThn.setEnabled(e);
}
private void aktiK3(boolean c)
{
    btTamkar.setEnabled(c);
    btEdkar.setEnabled(c);
    btHapkar.setEnabled(c);
    tfKdkar.setEnabled(c);
    tfNmkar.setEnabled(c);
    tfAlkar.setEnabled(c);
    tfKokar.setEnabled(c);
    tfNotelkar.setEnabled(c);
    tfHp.setEnabled(c);
    tfKtp.setEnabled(c);
    cbTgl.setEnabled(c);
    cbBln.setEnabled(c);
    cbThn.setEnabled(c);
}
private void itemTgl(){
    cbTgl.removeAllItems();
    for(int i = 1; i < 32; i++){

```

```

        cbTgl.addItem(i);
    }
}
private void itemBln(){
    cbBln.removeAllItems();
    for(int i = 0; i < 13; i++){
        cbBln.addItem(bulan[i]);
    }
}
private void itemThn(){
    cbThn.removeAllItems();
    for(int i = 1974; i < 2000;
i++){
        cbThn.addItem(i);
    }
}
private void nomorKar()
{
    try
    {
        Connection conKar =
        Libra.KonekZahwa.getkoneksi();
        Statement sNK =
        conKar.createStatement();

        String sqlNk ="select
Kd_Karyawan as nokdkar from
Karyawan ORDER by Kd_Karyawan desc
";

        ResultSet rsk =
        sNK.executeQuery(sqlNk);

        if(rsk.next())
        {
            String noK =
            rsk.getString("nokdkar");
            String noKa =
            noK.substring(1 , 4);
            int noKx =
            Integer.parseInt(noKa)+1;
            String noKb =
            noK.substring(2 , 4);
            String noKc =
            noK.substring(3 , 4);

            if (noKx > 999)
            {

                PlaySound.PlaySound();

                JOptionPane.showMessageDialog(this
, "Data Karyawan Over, telah
mencapai maksimal!");

                tfKdkar.setEnabled(false);

                tfNmkar.setEnabled(false);

                btSimkar.setEnabled(false);
            }
            if (noKx > 0)

```

```

        {
            tfKdkar.setText("K00"
+noKx);
        }
        if (noKx >= 10)
        {
            tfKdkar.setText("K0"
+noKx);
        }
        if (noKx >= 100)
        {
            tfKdkar.setText("K"
+noKx);
        }
        else
        {
            tfKdkar.setText("K000");
        }
        rsk.close();
        sNK.close();
    }
    catch(SQLException e)
    {

        System.out.println("Terjadi error
di ambil data nomor Karyawan");
    }
    tfNmkar.requestFocus();
}
private void loadKary()
{
    modelKaryawan.getDataVector().remo
veAllElements();

    modelKaryawan.fireTableDataChanged
();
    try
    {
        Connection ckar =
        Libra.KonekZahwa.getkoneksi();
        Statement skar =
        ckar.createStatement();

        String sqlkar ="select
* from karyawan ";
        ResultSet rskar =
        skar.executeQuery(sqlkar);

        while(rskar.next())
        {
            Object[] okar =
            new Object[8];
            okar[0] =
            rskar.getString("Kd_Karyawan");
            okar[1] =
            rskar.getString("Nama");
            okar[2] =
            rskar.getString("Tanggal_Lahir");

```

```

                okar[3] =
rskar.getString("Alamat");
                okar[4] =
rskar.getString("Kota");
                String kos =
rskar.getString("Telepon");
                if (kos == null ||
kos == "") {
                    okar[5] = "---
";
                }
                else
                    okar[5] =
rskar.getString("Telepon");
                    okar[6] =
rskar.getString("Hp");
                    okar[7] =
rskar.getString("Ktp");

modelKaryawan.addRow(okar);
            }
            rskar.close();
            skar.close();
        }
        catch (SQLException e)
        {
            System.out.println("Terjadi error
            di load data karyawan");
        }
    }

    private void cekHapus() {
        if
        ("".equals(tfKdkar.getText()))
        {
            PlaySound.PlaySound();

            JOptionPane.showMessageDialog(this
            , "Kode Karyawan belum dipilih");
            tbKar.requestFocus();
            return;
        }

        String nokd =
        tfKdkar.getText();

        try
        {
            Connection con =
            Libra.KonekZahwa.getkoneksi();
            Statement sj =
            con.createStatement();
            String sql = "select *
            from transaksi WHERE Kd_Karyawan
            LIKE '" + nokd + "'";

            ResultSet rsj =
            sj.executeQuery(sql);
            if (rsj.next())
            {
                String n =
                rsj.getString("Kd_Karyawan");

```

```

            if (nokd.equals(n)) {

                System.out.println("Kd Kar = "+n+"
                , Ada transaksi !!!");

                ckh = false;

                btHapkar.setEnabled(false);

                return;
            }
            else
                ckh = true;

            btHapkar.setEnabled(true);

        }
        rsj.close();
        sj.close();
    }
    catch (SQLException e)
    {
        System.out.println("Terjadi error
        di load transaksi");
    }
}

    private void
    tbKarMouseClicked(java.awt.event.M
    ouseEvent evt) {
        // TODO add your handling
        code here:
        aktiK2(false);
        aktiK3(true);
        int iK =
        tbKar.getSelectedRow();
        if (iK == -1)
        {
            return;
        }

        String kdKs =
        modelKaryawan.getValueAt(iK,
        0).toString();
        String nmKs =
        modelKaryawan.getValueAt(iK,
        1).toString();
        String tglKs =
        modelKaryawan.getValueAt(iK,
        2).toString();
        String alKs =
        modelKaryawan.getValueAt(iK,
        3).toString();
        String koKs =
        modelKaryawan.getValueAt(iK,
        4).toString();

```

```

        ntKs =
modelKaryawan.getValueAt(iK,
5).toString();
        if( ntKs == "---" ){
            ntKs = "";
        }
        else

tfNotelkar.setText(ntKs);
        String hpKs =
modelKaryawan.getValueAt(iK,
6).toString();
        String ktpKs =
modelKaryawan.getValueAt(iK,
7).toString();
        tfKdkar.setText(kdKs);
        tfNmkar.setText(nmKs);
        int tgl =
Integer.parseInt(tglKs.substring(8
, 10));
        int bul =
Integer.parseInt(tglKs.substring(5
, 7));
        int thn =
Integer.parseInt(tglKs.substring(0
, 4));

cbTgl.setSelectedItem(tgl);

cbBln.setSelectedItem(bulan[bul]);

cbThn.setSelectedItem(thn);
        tfAlkar.setText(alKs);
        tfKokar.setText(koKs);
        tfHp.setText(hpKs);
        tfKtp.setText(ktpKs);
        cekHapus();
    }

    private void
btTamkarActionPerformed(java.awt.e
vent.ActionEvent evt) {
        // TODO add your handling
code here:
        aktK1(false);
        aktiK2(true);
        kOsongkanKar();
        nomorKar();
    }

    private void
btSimkarActionPerformed(java.awt.e
vent.ActionEvent evt) {
        // TODO add your handling
code here:
        if
        ("".equals(tfNmkar.getText()))
        {
            PlaySound.PlaySound();

```

```

JOptionPane.showMessageDialog(this
, "Nama Karyawan mohon diisi");
        tfNmkar.requestFocus();
        return;
    }

    if(cbTgl.getSelectedIndex()==-1){
        PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Tanggal Lahir mohon diisi");
        cbTgl.requestFocus();
        return;
    }

    if(cbBln.getSelectedIndex()==-1){
        PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Bulan Lahir mohon diisi");
        cbBln.requestFocus();
        return;
    }

    if(cbThn.getSelectedIndex()==-1){
        PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Tahun Lahir mohon diisi");
        cbThn.requestFocus();
        return;
    }
    if
    ("".equals(tfAlkar.getText()))
    {
        PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Alamat Karyawan mohon diisi");
        tfAlkar.requestFocus();
        return;
    }
    if
    ("".equals(tfKokar.getText()))
    {
        PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Kota Karyawan mohon diisi");
        tfKokar.requestFocus();
        return;
    }
    if
    ("".equals(tfHp.getText()))
    {
        PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Nomor Hp Karyawan mohon
diisi");

```

```

        tfHp.requestFocus();
        return;
    }
    if
    ("".equals(tfKtp.getText()))
    {
        PlaySound.PlaySound();

        JOptionPane.showMessageDialog(this
        , "Nomor Ktp Karyawan mohon
        diisi");
        tfKtp.requestFocus();
        return;
    }
    String kdk =
    tfKdkar.getText();
    String nmk =
    tfNmkar.getText();
    String tglk =
    String.valueOf(cbTgl.getSelectedIt
    em());
    int blnk =
    cbBln.getSelectedIndex();
    String thnk =
    String.valueOf(cbThn.getSelectedIt
    em());
    String lhr = thnk+"-
    "+blnk+"-"+tglk;
    String alk =
    tfAlkar.getText();
    String kok =
    tfKokar.getText();
    String ntk = "";

    if ("".equals(tfNotelkar.getText()))
    {
        ntk = "";
    }
    else
        ntk =
    tfNotelkar.getText();
    String hpk =
    tfHp.getText();
    String ktpk =
    tfKtp.getText();
    try
    {
        Connection cinK =
        Libra.KonekZahwa.getkoneksi();
        String sqlK = "INSERT
        INTO Karyawan VALUES ( ?, ?, ?, ?,
        ?, ?, ?, ?)";
        PreparedStatement pK =
        cinK.prepareStatement(sqlK);
        pK.setString(1,kdk );
        pK.setString(2,nmk );
        pK.setString(3, lhr);
        pK.setString(4,alk );
        pK.setString(5,kok );
        pK.setString(6,ntk );
        pK.setString(7, hpk);

```

```

        pK.setString(8, ktpk);
        pK.executeUpdate();
        pK.close();

        JOptionPane.showMessageDialog(this
        , "Data Karyawan Telah DiSimpan");
    }
    catch(SQLException exk)
    {

        System.out.println("Proses Tambah
        Karyawan Error");
    }
    finally
    {
        aktK1(false);
        loadKary();
        kOsongkanKar();
    }

    private void
    btHapkarActionPerformed(java.awt.e
    vent.ActionEvent evt) {
        // TODO add your handling
        code here:
        int iKh =
        tbKar.getSelectedRow();
        if (iKh == -1)
        {
            PlaySound.PlaySound();

            JOptionPane.showMessageDialog(this
            , "Silahkan pilih data pada Tabel
            Karyawan!");
            return;
        }
        String kdKh = (String)
        modelKaryawan.getValueAt(iKh, 0);
        try
        {
            Connection cKh =
            Libra.KonekZahwa.getkoneksi();
            String sqlKh = "DELETE
            FROM Karyawan WHERE Kd_Karyawan =
            ?";
            PreparedStatement pKh =
            cKh.prepareStatement(sqlKh);

            pKh.setString(1 , kdKh);
            pKh.executeUpdate();
            pKh.close();

            JOptionPane.showMessageDialog(this
            , "Data "+kdKh+ " Telah DiHapus
            !!!");
        }
        catch(SQLException eKh)
        {

```



```

System.err.println("Error Data
Karyawan tidak bisa diHapus !!!");
    }
    finally
    {
        loadKary();
        kOsongkanKar();
    }
}

private void
btEdkarActionPerformed(java.awt.ev
ent.ActionEvent evt) {
    // TODO add your handling
    code here:
        String kdK =
tfKdkar.getText();
        String nmK =
tfNmkar.getText();
        String tglK =
String.valueOf(cbTgl.getSelectedIt
em());
        String blnK =
String.valueOf(cbBln.getSelectedIn
dex());
        String thnK =
String.valueOf(cbThn.getSelectedIt
em());
        String lhrK = thnK+"-
"+blnK+"-"+tglK;
        String alK =
tfAlkar.getText();
        String koK =
tfKokar.getText();
        String noK = "";

if("").equals(tfNotelkar.getText())
){
        noK = "";
    }
    else
        noK =
tfNotelkar.getText();

        String hpK =
tfHp.getText();
        String ktpK =
tfKtp.getText();
        try
        {
            Connection ckar =
Libra.KonekZahwa.getkoneksi();

            String sqlkar = "UPDATE
Karyawan set Nama = ?
,Tanggal_Lahir = ?, "
+ " Alamat = ? ,
Kota = ? , Telepon = ? , Hp = ? ,
"

```

```

+ "Ktp = ? where
Kd_Karyawan = '"+kdK+"'";
        PreparedStatement pkar =
ckar.prepareStatement(sqlkar);
        pkar.setString(1 ,
nmK);
        pkar.setString(2 ,
lhrK);
        pkar.setString(3 ,
alK);
        pkar.setString(4 ,
koK);
        pkar.setString(5 ,
noK);
        pkar.setString(6 ,
hpK);
        pkar.setString(7 ,
ktpK);
        pkar.executeUpdate();

JOptionPane.showMessageDialog(this
, "Data "+kdK+ " Telah DiUbah
!!!");
        pkar.close();
    }
    catch(SQLException eka)
    {

System.err.println("Error Data
Karyawan tidak bisa diUbah !!!");
    }
    finally
    {
        loadKary();
        kOsongkanKar();
    }
}

private void
cbBlnActionPerformed(java.awt.even
t.ActionEvent evt) {
    // TODO add your handling
    code here:
        int blnk =
cbBln.getSelectedIndex();
        System.out.println("Bulan
: "+blnk);
    }

private void
tfHpKeyTyped(java.awt.event.KeyEve
nt evt) {
    // TODO add your handling
    code here:

if(tfHp.getText().length()>11)
{
        evt.consume();
    }
    char c = evt.getKeyChar();

```

```

        if(!Character.isDigit(c))
        {
            evt.consume();
        }
    }

    private void
    tfNotelkarKeyTyped(java.awt.event.
    KeyEvent evt) {
        // TODO add your handling
        code here:

    if(tfNotelkar.getText().length()>1
    0)
        {
            evt.consume();
        }
        char c = evt.getKeyChar();
        if(!Character.isDigit(c))
        {
            evt.consume();
        }
    }

    private void
    tfKtpKeyTyped(java.awt.event.KeyEv
    ent evt) {
        // TODO add your handling
        code here:

    if(tfKtp.getText().length()>15)
        {
            evt.consume();
        }
        char c = evt.getKeyChar();
        if(!Character.isDigit(c))
        {
            evt.consume();
        }
    }

    private void
    tfKokarKeyTyped(java.awt.event.Key
    Event evt) {
        // TODO add your handling
        code here:

    if(tfKokar.getText().length()>19)
        {
            evt.consume();
        }
    }

```

FDiskon.java

```

public class FDiskon extends
    javax.swing.JFrame {
    private final DefaultTableModel
    modeldiskon;

```

```

    private TableCellRenderer merah =
    new WarnaKodeMerah();
    private Sounderror PlaySound = new
    Sounderror();
    boolean ckd;
    String nokd;
    int no = 0;
    String [] keterangan = {"Rumah
    Tangga","Belum
    Menikah","Bekerja","Tidak
    Bekerja","Wiraswasta","Mahasiswa",
    "Pelajar","Member"};
    /**
     * Creates new form FDiskon
     */
    public FDiskon() {
        initComponents();
        modeldiskon = new
    DefaultTableModel(){
        @Override
        public boolean
    isCellEditable (int row, int
    coloumn){
            return false;
        }
    };

    modeldiskon.addColumn("Kode
    Diskon");

    modeldiskon.addColumn("Pekerjaan")
    ;

    modeldiskon.addColumn("Diskon");

    tblDiskon.setModel(modeldiskon);
        TableColumn cold0 =
    tblDiskon.getColumnModel().getColu
    mn(0);
        TableColumn cold1 =
    tblDiskon.getColumnModel().getColu
    mn(1);
        TableColumn cold2 =
    tblDiskon.getColumnModel().getColu
    mn(2);

        for(int a = 0; a <
    tblDiskon.getColumnCount();a++)
        {

    tblDiskon.getColumnModel().getColu
    mn(a).setCellRenderer(new
    WarnaKolom());
        }

    cold0.setPreferredWidth(80);

    cold1.setPreferredWidth(150);

    cold2.setPreferredWidth(80);

```

```

cold0.setCellRenderer(merah);

tblDiskon.setSelectionBackground(C
olor.LIGHT_GRAY);

tblDiskon.setGridColor(Color.yello
w);
        loadData();
        kOsongD();
        ldker();

        aktD1(false);
        this.setTitle("Data
Diskon");
        this.setLocation(600,
250);
    }
private void kOsongD()
{
    tfDiskon.setText("");

cbPekerjaan.setSelectedIndex(-1);
}
private void aktD1(boolean k)
{
    btEdit.setEnabled(k);
    btHapus.setEnabled(k);
    btSimpan.setEnabled(k);

    tfDiskon.setEnabled(k);
    cbPekerjaan.setEnabled(k);
}
private void aktiD2(boolean z)
{
    btSimpan.setEnabled(z);

    tfDiskon.setEnabled(z);
    cbPekerjaan.setEnabled(z);
}
private void aktiD3(boolean y)
{
    btTambah.setEnabled(y);
    btEdit.setEnabled(y);
    btHapus.setEnabled(y);

    tfDiskon.setEnabled(y);
    cbPekerjaan.setEnabled(y);
}
private void loadData()
{
    modeldiskon.getDataVector().remove
AllElements();

    modeldiskon.fireTableDataChanged()
;
    try
    {

```

```

        Connection con =
        Libra.KonekZahwa.getkoneksi();
        Statement sj =
        con.createStatement();
        String sql ="select *
        from diskon ORDER by kd_diskon ASC
        ";
        ResultSet rsj =
        sj.executeQuery(sql);
        while(rsj.next())
        {
            Object[] o = new
            Object[3];
            o[0] =
            rsj.getString("kd_diskon");
            o[1] =
            rsj.getString("Keterangan");
            o[2] =
            rsj.getString("diskon");

            modeldiskon.addRow(o);
        }
        rsj.close();
        sj.close();
    }
    catch(SQLException e)
    {
        System.out.println("Terjadi error
        di load diskon");
    }
}

private void loadNo(){
    try {
        Connection conlg =
        Libra.KonekZahwa.getkoneksi();
        Statement slg =
        conlg.createStatement();
        String sqllg ="select
        kd_diskon as no from diskon ORDER
        by kd_diskon desc ";
        ResultSet rslg =
        slg.executeQuery(sqllg);
        if(rslg.next())
        {
            String noPs =
            rslg.getString("no");

            int noPsx =
            Integer.parseInt(noPs)+1;
            no = noPsx;

        }
        else no = 1;
    } catch (SQLException ex) {
        System.out.println("Terjadi error
        di ambil kd_diskon");
    }
}

```

```

    }
    private void ldker()
    {
        cbPekerjaan.removeAllItems();
        for(int c = 0; c <
keterangan.length; c++){

cbPekerjaan.addItem(keterangan[c])
;

cbPekerjaan.setSelectedIndex(0);
    }

}

```

```

    private void
btSimpanActionPerformed(java.awt.e
vent.ActionEvent evt) {
    // TODO add your handling
code here:

    if
("".equals(cbPekerjaan.getSelected
Item()))
    {
        PlaySound.PlaySound();

JOptionPane.showMessageDialog(this
, "Keterangan Diskon mohon
diisi");

cbPekerjaan.requestFocus();
        return;
    }

    String dis =
tfDiskon.getText();
    String pek =
cbPekerjaan.getSelectedItem().toSt
ring();
    try
    {
        Connection cinD =
Libra.KonekZahwa.getkoneksi();
        String sqlD = "INSERT
INTO diskon VALUES ( ?, ?, ?)";
        PreparedStatement pD =
cinD.prepareStatement(sqlD);

pD.setString(1,String.valueOf(no)
);

        pD.setString(2,dis );
        pD.setString(3,pek );
        pD.executeUpdate();
        pD.close();

```

```

JOptionPane.showMessageDialog(this
, "Data Telah DiSimpan");
    }
    catch(SQLException ex)
    {

System.out.println("Proses Tambah
Error");
    }
    finally
    {
        loadData();
        kOsongD();
    }

}

```

```

    private void
btEditActionPerformed(java.awt.eve
nt.ActionEvent evt) {
    // TODO add your handling
code here:

        String disk =
tfDiskon.getText();
        String pek2 =
cbPekerjaan.getSelectedItem().toSt
ring();
        try
        {
            Connection cb =
Libra.KonekZahwa.getkoneksi();

            String sqlb = "UPDATE
diskon set diskon = ? , keterangan
= ? where kd_diskon = '"+no+"'";
            PreparedStatement pb =
cb.prepareStatement(sqlb);

            pb.setString(1 ,
disk);
            pb.setString(2 ,
pek2);

            pb.executeUpdate();

JOptionPane.showMessageDialog(this
, "Data "+no+ " Telah DiUbah
!!!");
            pb.close();
        }

        catch(SQLException eJh)
        {

System.err.println("Error tidak
bisa diUbah !!!");
        }

```

```

        finally
        {
            loadData();
            kOsongD();
        }
    }

    private void
    btHapusActionPerformed(java.awt.ev
    ent.ActionEvent evt) {
        // TODO add your handling
        code here:
        int iJh =
        tblDiskon.getSelectedRow();
        if (iJh == -1)
        {
            PlaySound.PlaySound();

            JOptionPane.showMessageDialog(this
            , "Silahkan pilih data yang akan
            dihapus!");
            return;
        }

        String kdh = (String)
        modeldiskon.getValueAt(iJh, 0);

        try
        {
            Connection ch =
            Libra.KonekZahwa.getkoneksi();
            String sqlh = "DELETE
            FROM diskon WHERE kd_diskon = ?";
            PreparedStatement ph =
            ch.prepareStatement(sqlh);

            ph.setString(1 , kdh);
            ph.executeUpdate();
            ph.close();

            JOptionPane.showMessageDialog(this
            , "Data "+kdh+ " Telah DiHapus
            !!!");
        }
        catch (SQLException eh)
        {

            System.err.println("Error tidak
            bisa diHapus !!!");
        }
        finally
        {
            loadData();
            kOsongD();
        }
    }

    private void
    tblDiskonMouseClicked(java.awt.eve
    nt.MouseEvent evt) {

```

```

        // TODO add your handling
        code here:
        aktiD2(false);
        aktiD3(true);
        int iJ =
        tblDiskon.getSelectedRow();
        if (iJ == -1)
        {
            return;
        }
        String kdJs =
        modeldiskon.getValueAt(iJ,
        0).toString();
        String peker =
        modeldiskon.getValueAt(iJ,
        1).toString();
        String disk =
        modeldiskon.getValueAt(iJ,
        2).toString();
        no =
        Integer.parseInt(kdJs);
        tfDiskon.setText(disk);

        cbPekerjaan.setSelectedItem(peker)
        ;
        //          cekHps();
    }

    private void
    btTambahActionPerformed(java.awt.e
    vent.ActionEvent evt) {
        // TODO add your handling
        code here:
        loadNo();
        aktD1(false);
        aktiD2(true);
        kOsongD();
        tfDiskon.requestFocus();
        System.out.println("no :
        "+no);
    }

```

LaporanHarian.java

```

public class LaporanHarian extends
    javax.swing.JFrame {
    private DefaultTableModel
    modLaporan;
    private TableCellRenderer kanan =
    new renderingKanan();
    private TableCellRenderer merah =
    new WarnaKodeMerah();
    String [] bulan =
    {"", "JANUARI", "FEBRUARI", "MARET", "
    APRIL", "MEI", "JUNI", "JULI"

    , "AGUSTUS", "SEPTEMBER", "OKTOBER", "
    NOVEMBER", "DESEMBER"};

```

```

String
caripl,sqlplc,sqlpla,sqlplb;
String tglp,thnp ,kodpel;
int blnp ,a = -1,b = -1;
int panDt;
boolean y1,y2,y3;
Date now = new Date();
SimpleDateFormat frm = new
SimpleDateFormat("dd/MM/yyyy");
    String utuh =
frm.format(now);
    String thn =
utuh.substring(6, 10);
String [] notrans ,nmpel,namapel;
int [] panDtdet;
String[] Dat1,no2,no;
/**
 * Creates new form
LaporanHarian
 */
    public LaporanHarian() {
        initComponents();
        modLaporan = new
DefaultTableModel();

modLaporan.addColumn("Nomor");

modLaporan.addColumn("Nama");
        modLaporan.addColumn("Kode
Transaksi");

modLaporan.addColumn("Perawatan");

modLaporan.addColumn("Total");

tbTransaksi.setModel(modLaporan);

tbTransaksi.setAutoResizeMode(JTab
le.AUTO_RESIZE_OFF);
        TableColumn co0 =
tbTransaksi.getColumnModel().getCo
lumn(0);

        TableColumn co1 =
tbTransaksi.getColumnModel().getCo
lumn(1);
        TableColumn co2 =
tbTransaksi.getColumnModel().getCo
lumn(2);
        TableColumn co3 =
tbTransaksi.getColumnModel().getCo
lumn(3);
        TableColumn co4 =
tbTransaksi.getColumnModel().getCo
lumn(4);
        for(int r = 0; r <
tbTransaksi.getColumnCount();r++)
        {

tbTransaksi.getColumnModel().getCo

```

```

lumn(r).setCellRenderer(new
WarnaKolom());
        }

co4.setCellRenderer(kanan);

co2.setCellRenderer(merah);
        co0.setPreferredWidth(50);

co1.setPreferredWidth(185);

co2.setPreferredWidth(110);

co3.setPreferredWidth(215);

co4.setPreferredWidth(100);

        itemTgl();
        itemBln();
        itemThn();
        kosongkan();
        this.setTitle("Laporan
Transaksi Harian");
    }
private void itemTgl(){
    cbtg.removeAllItems();
    for(int i = 1; i < 32; i++){
        cbtg.addItem(i);
    }
}
private void itemBln(){
    cbbl.removeAllItems();
    for(int i = 0; i < 13; i++){
        cbbl.addItem(bulan[i]);
    }
}
private void itemThn(){
    int th =
Integer.parseInt(thn);
    cbth.removeAllItems();
    for(int i = 2009; i <= th;
i++){
        cbth.addItem(i);
    }
}
private void kosongkan(){

y1 = false;
y2 = false;
y3 = false;
a = -1;
b = -1;
panDt = 0;

}
private void tglData(){
    int tgl2 =
Integer.parseInt(tglp);
    if( tgl2 < 10){
        tglp =
"0"+cbtg.getSelectedItem();

```

```

        }
        if( tgl2 > 10){
            tglp =
            ""+cbtg.getSelectedItemAt();
        }
    }
    private void panjData() {
        tglData();
        try {
            Connection c =
            Libra.KonekZahwa.getkoneksi();
            Statement st =
            c.createStatement();

            ResultSet rs =
            st.executeQuery("select
            COUNT(Kd_Transaksi)from transaksi
            "
            + "where Tanggal like
            '"+thnp+"-"+blnp+"-"+tglp+"' ");

            while(rs.next())
            {
                panDt =
                Integer.parseInt(rs.getString(1));
            }
            rs.close();
            st.close();
        }
        catch (SQLException ex) {

            System.out.println("Terjadi error
            di cari panj data Transaksi");
        }
    }
    private void ambilData() {
        tglData();
        notrans = new String[panDt];
        nmpel = new String[panDt];
        try {
            Connection c =
            Libra.KonekZahwa.getkoneksi();
            Statement st =
            c.createStatement();
            ResultSet rs =
            st.executeQuery("select * from
            transaksi where Tanggal "
            + "like
            '"+thnp+"-"+blnp+"-"+tglp+"' ");

            while(rs.next())
            {
                a +=1;
                Object[] cr = new
                Object[3];

                String tgl2
                = tglp+"-"+blnp+"-"+thnp;
                cr[0] =
                tgl2;

```

```

                cr[1] =
                rs.getString("Kd_Transaksi");
                cr[2] =
                rs.getString("Kd_Konsumen");
                notrans[a] =
                rs.getString("Kd_Transaksi");
                nmpel[a] =
                rs.getString("Kd_Konsumen");

                modLaporan.addRow(cr);
            }
            rs.close();
            st.close();
        }
        catch (SQLException ex) {

            System.out.println("Terjadi error
            di cari data Transaksi");
        }
    }
    private void bacakdpel()
    {
        modLaporan.getDataVector().removeA
        llElements();

        modLaporan.fireTableDataChanged();
        namapel = new String[panDt];
        for(int yi = 0; yi < panDt;
        yi++){
            try {
                Connection cpel =
                Libra.KonekZahwa.getkoneksi();
                Statement s =
                cpel.createStatement();
                String sqlpel =
                "SELECT Nama FROM konsumen WHERE
                Kd_Konsumen = '" + nmpel[yi] +
                "'";

                ResultSet rpel =
                s.executeQuery(sqlpel);

                while (rpel.next())
                {

                    namapel[yi] =
                    rpel.getString("Nama");
                }
                rpel.close();
                s.close();
            }
            catch (SQLException ePb)
            {

                System.err.println("Error data
                dtransaksi tidak bisa
                diAmbil!!!");
            }
        }
    }
    private void panjDataDetail() {

```

```

panDtdet = new int[panDt];
for(int yy = 0; yy < panDt; yy++){
    try {
        Connection c =
        Libra.KonekZahwa.getkoneksi();
        Statement st =
        c.createStatement();
        ResultSet rs =
        st.executeQuery("select
        COUNT(Kd_Transaksi)from dtransaksi
        "
        + "where Kd_Transaksi
        like '"+notrans[yy]+' '");
        while(rs.next())
        {
            b +=1;

            panDtdet[b] =
            Integer.parseInt(rs.getString(1));
        }
        rs.close();
        st.close();
    }
    catch (SQLException ex) {

        System.out.println("Terjadi error
        di cari panj data Detail
        Transaksi");
    }
}
private void namaPel()
{
    int panBaris =0;
    for(int o = 0; o <
    panDtdet.length ; o++){
        panBaris =
        panBaris+panDtdet[o];
    }
    Dat1 = new String[panBaris];
    no = new String[panDt];
    for(int itu = 0; itu < panDt ;
    itu++){
        no[itu] =
        String.valueOf(itu+1);
    }
    int isi =panBaris;
    no2 = new String[panBaris];
    for(int ok = panDt-1; ok >=
    0; ok--){
        isi = isi -
        panDtdet[ok];
        no2 [isi]= no[ok];
        Dat1 [isi]= namapel[ok];
        if(null ==
        no[ok]||"".equals(no[ok])){
            no2 [isi]= "";
        }
    }
}

```

```

        if(null ==
        namapel[ok]||"P---
        ".equals(namapel[ok])){
            Dat1 [isi]=
            "Pelanggan Baru";
        }
    }
    private void bacaDetT()
    {
        int panBaris =0;

        modLaporan.getDataVector().removeA
        llElements();

        modLaporan.fireTableDataChanged();

        for(int o = 0; o <
        panDtdet.length ; o++){
            panBaris =
            panBaris+panDtdet[o];
        }

        int bn = -1;
        Object[] oPs;

        for(int yy = 0; yy < panDt;
        yy++){
            try {
                Connection cpper =
                Libra.KonekZahwa.getkoneksi();
                Statement sp =
                cpper.createStatement();
                String sqlpper =
                "SELECT * FROM dtransaksi WHERE
                Kd_Transaksi = '" +notrans[yy]+
                "'";

                ResultSet rsambarpe =
                sp.executeQuery(sqlpper);

                while
                (rsambarpe.next())
                {
                    bn +=1;
                    oPs = new
                    Object[6];

                    oPs[0] = no2[bn];

                    oPs[1] = Dat1[bn];
                    oPs[2] =
                    rsambarpe.getString("Kd_Transaksi"
                    );

                    oPs[3] =
                    rsambarpe.getString("Nama");
                    oPs[4] =
                    rsambarpe.getString("Harga");

                    modLaporan.addRow(oPs);
                }
            }
        }
    }
}

```



```

        }
        rsambarpe.close();
        sp.close();
    }
    catch (SQLException ePb)
    {

System.err.println("Error data
dtransaksi tidak bisa
diAmbil!!!");
    }
    }
}

private void hittotal(){
    String [] total = null;
    int baris =
modLaporan.getRowCount();
    int totalUang = 0;
    total = new String[baris];

    for(int xy = 0; xy < baris;
xy++){
        total[xy]=
String.valueOf(modLaporan.getValue
At(xy, 4));
    }
    for(int xy = 0; xy < baris;
xy++){
        totalUang = totalUang +
Integer.parseInt(total[xy]);
    }

tfPemasukan.setText(String.valueOf
(totalUang));

tfJumtran.setText(String.valueOf(p
anDt));
}

private void
cbtgActionPerformed(java.awt.event
.ActionEvent evt) {
    // TODO add your handling
code here:
    tglp =
String.valueOf(cbtg.getSelectedIte
m());
}

private void
cbblActionPerformed(java.awt.event
.ActionEvent evt) {
    // TODO add your handling
code here:
    blnp =
cbbl.getSelectedIndex();
}

private void
cbthActionPerformed(java.awt.event
.ActionEvent evt) {

```

```

        // TODO add your handling
code here:
        thnp =
String.valueOf(cbth.getSelectedIte
m());
    }

```

```

private void
btTampilActionPerformed(java.awt.e
vent.ActionEvent evt) {
    // TODO add your handling
code here:

```

```

        panjData();
        ambilData();
        panjDataDetail();
        bacakdpel();
        namaPel();
        bacaDetT();
        hittotal();
        kosongkan();
    }

```

```

private void
btKeluarActionPerformed(java.awt.e
vent.ActionEvent evt) {
    // TODO add your handling
code here:
        this.dispose();
    }

```

LaporanBulanan.java

```

public class LaporanBulanan
extends javax.swing.JFrame {
    String [] bulan =
    {"", "JANUARI", "FEBRUARI", "MARET", "
APRIL", "MEI", "JUNI", "JULI"

, "AGUSTUS", "SEPTEMBER", "OKTOBER", "
NOVEMBER", "DESEMBER"};
    private TableCellRenderer kanan =
new renderingKanan();
    private DefaultTableModel
modellap;
    String sqlplc;
    String thnp ;
    int blnp, jmlH;
    Date now = new Date();
    SimpleDateFormat frm = new
SimpleDateFormat("dd/MM/yyyy");
        String utuh =
frm.format(now);
        String thn =
utuh.substring(6, 10);
        String bln =
utuh.substring(3, 5);

```

```

private Sounderror PlaySound = new
Sounderror();
/**
 * Creates new form
LaporanHarian
 */
public LaporanBulanan() {
    initComponents();
    modelLap = new
DefaultTableModel();
    modelLap.addColumn("No");

modelLap.addColumn("Tanggal");
    modelLap.addColumn("Jumlah
Transaksi");
    modelLap.addColumn("Sub
Total");
    modelLap.addColumn("Sub
Diskon");

tbBulanan.setModel(modelLap);

tbBulanan.setAutoResizeMode(JTable
.AUTO_RESIZE_OFF);
    TableColumn cl0 =
tbBulanan.getColumnModel().getColu
mn(0);
    TableColumn cl1 =
tbBulanan.getColumnModel().getColu
mn(1);
    TableColumn cl2 =
tbBulanan.getColumnModel().getColu
mn(2);
    TableColumn cl3 =
tbBulanan.getColumnModel().getColu
mn(3);
    TableColumn cl4 =
tbBulanan.getColumnModel().getColu
mn(4);
    for(int a = 0; a <
tbBulanan.getColumnCount();a++)
    {

tbBulanan.getColumnModel().getColu
mn(a).setCellRenderer(new
WarnaKolom());
        cl0.setPreferredWidth(50);

cl1.setPreferredWidth(100);

cl2.setPreferredWidth(120);

cl3.setPreferredWidth(105);

cl4.setPreferredWidth(105);

cl2.setCellRenderer(kanan);

cl3.setCellRenderer(kanan);

```

```

cl4.setCellRenderer(kanan);
        this.setTitle("Laporan
Transaksi Bulanan ");
        this.setLocation(600,
250);
        itemBln();
        itemThn();
        kosongkan();
    }
private void kosongkan(){
    cbBln.setSelectedIndex(-1);
    cbThn.setSelectedIndex(-1);
    tfTotPend.setText("");
    tfTotSeluruh.setText("");
    tfDiskon.setText("");

modelLap.getDataVector().removeAll
Elements();
    modelLap.fireTableDataChanged();
}

private void itemBln(){
    cbBln.removeAllItems();
    for(int i = 0; i < 13; i++){
        cbBln.addItem(bulan[i]);
    }
}
private void itemThn(){
    int th = Integer.parseInt(thn);
    cbThn.removeAllItems();
    for(int i = 2009; i <= th; i++){
        cbThn.addItem(i);
    }
}
private void jumHari(){
    try {
        SimpleDateFormat SDF=new
SimpleDateFormat("MM-yyyy");
        String str=" "+ blnp+"-
"+thnp;
        Date d = SDF.parse(str);
        Calendar
Cal=Calendar.getInstance();
        Cal.setTime(d);
        jmlH =
Cal.getActualMaximum(Calendar.DAY_
OF_MONTH);

System.out.println(Cal.getActualMa
ximum(Calendar.DAY_OF_MONTH));
    } catch (ParseException ex)
    {

    }

}

private void hitTotal(){

```

```

modellLap.getDataVector().removeAll
Elements();

modellLap.fireTableDataChanged();
    String totalperHari ;
    String totalDiskon ;
    String jumlahTransaksi;
    for(int hr = 1; hr <= jmlH;
hr++){

        try {
            Connection c =
Libra.KonekZahwa.getkoneksi();
            Statement st =
c.createStatement();

            if( hr < 10){
                sqlplc
="select
SUM(Total),SUM(Diskon),COUNT(Kd_Tr
ansaksi)from transaksi where
Tanggal like '"+thnp+"-"+blnp+"-
0"+hr+"' ";
            }
            if( hr > 10){
                sqlplc
="select
SUM(Total),SUM(Diskon),COUNT(Kd_Tr
ansaksi)from transaksi where
Tanggal like '"+thnp+"-"+blnp+"-
"+hr+"' ";
            }

            ResultSet rs =
st.executeQuery(sqlplc);

            while(rs.next())
            {
                totalperHari =
rs.getString(1);
                totalDiskon =
rs.getString(2);
                jumlahTransaksi =
rs.getString(3);
                Object[] maslap =
new Object[5];
                if(totalperHari
== null){
totalperHari = "0";
                }
                if(totalDiskon
== null){
totalDiskon = "0";
                }
                if( hr < 10){
                    maslap[0] =
                    "0"+hr+"-"+blnp+"-"+thnp;
                    maslap[1] =
                    jumlahTransaksi;
                    maslap[2] =
                    totalperHari;
                    maslap[3] =
                    totalDiskon;
                    modellLap.addRow(maslap);
                }
            }
            if( hr > 10){
                maslap[0] =
                hr;
                maslap[1] =
                hr+"-"+blnp+"-"+thnp;
                maslap[2] =
                jumlahTransaksi;
                maslap[3] =
                totalperHari;
                maslap[4] =
                totalDiskon;
                modellLap.addRow(maslap);
            }
        }
        rs.close();
        st.close();
    }

    catch (SQLException ex) {
        System.out.println("Terjadi error
di Hittotal Transaksi Bulanan");
    }
}

private void jmlTot(){
    Object[] diskonx = new
Object[tbBulanan.getRowCount()];
    Object[] ttseinya= new
Object[tbBulanan.getRowCount()];
    int ttdis = 0;
    int ttstl = 0;
    for(int e = 0 ; e <
tbBulanan.getRowCount(); e++){
        diskonx[e] =
tbBulanan.getValueAt(e, 4);
        ttseinya[e] =
tbBulanan.getValueAt(e, 3);
        if(diskonx[e] != null){
            ttdis = ttdis +
Integer.parseInt((String)
diskonx[e]);
        }
        if(ttseinya[e] != null){

```

```

        ttstl = ttstl +
Integer.parseInt((String)
ttstlnya[e]);
    }
}

tfTotPend.setText(String.valueOf(t
tsl));

tfDiskon.setText(String.valueOf(tt
dis));
    int seltot = ttstl + ttdis;

tfTotSeluruh.setText(String.valueO
f(seltot));
}
private void cetakbln(){
    int y = 0;
    Frame cetak = new Frame();

    PrintJob print =
cetak.getToolkit().getPrintJob(cet
ak, "Mencetak", null, null);
    if(print!=null){
        Graphics grap =
print.getGraphics();

        if(grap!=null){

            grap.setFont(new
Font("Dialog", 1, 10));

            grap.drawString("Spa
dan Salon Zahwa", 30, 19);
            grap.setFont(new
Font("Dialog", 1, 9));
            grap.drawString("Jl.
Nitikan Baru No. 27 Umbulharjo ",
30, 29);
            grap.drawString("
Yogyakarta 0274-6632877", 30, 39);
            grap.setFont(new
Font("Dialog", 1, 14));
            String titl =
lbTitle.getText();
            grap.drawString(titl,
30, 62);
            grap.setFont(new
Font("Dialog", 1, 10));

            String blnx =
cbBln.getSelectedItem().toString()
;
            String thnx =
cbThn.getSelectedItem().toString()
;

            grap.drawString(blnx,
30, 92);
            grap.drawString(thnx,
120, 92);

```

```

        grap.setFont(new
Font("Dialog", 0, 12));
        grap.drawLine(20,
122, 300, 122);
        int nn = 0;
        for (int xx = 0;xx <
modelLap.getRowCount(); xx++ ){
            nn = nn + 15;

            grap.drawString(modelLap.getValueA
t(xx, 0).toString()+"
"+modelLap.getValueAt(xx,
1).toString(), 30, 132+nn );

            grap.drawString(modelLap.getValueA
t(xx, 2).toString(), 135, 132+nn
);

            grap.drawString(modelLap.getValueA
t(xx, 3).toString(), 160, 132+nn
);

            grap.drawString(modelLap.getValueA
t(xx, 4).toString(), 250, 132+nn
);
        }
        grap.drawLine(20,
162+nn, 300, 162+nn);
        grap.setFont(new
Font("Dialog", 2, 12));

        grap.drawString("Total      : ", 30,
175+nn);

        grap.drawString(tfTotPend.getText(
), 150, 175+nn);

        grap.drawString("Diskon : ", 30,
195+nn);

        grap.drawString(tfDiskon.getText()
, 150, 195+nn);

        grap.drawString("Total Seluruhnya
: ", 30, 215+nn);

        grap.drawString(tfTotSeluruh.getTe
xt(), 150, 215+nn);
        grap.drawLine(20,
225+nn, 300, 225+nn);
    }

    print.end();
    print.end();

}

cbBln.requestFocus();
}

```

```

        private void
        cbBlnActionPerformed(java.awt.event.ActionEvent evt) {
            // TODO add your handling
            code here:
            blnp =
            cbBln.getSelectedIndex();
        }

        private void
        cbThnActionPerformed(java.awt.event.ActionEvent evt) {
            // TODO add your handling
            code here:
            thnp =
            String.valueOf(cbThn.getSelectedItem());
        }

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

            if (cbBln.getSelectedIndex() == -1) {
                PlaySound.PlaySound();

                JOptionPane.showMessageDialog(this, "Pilihan Bulan belum terpilih !", "PERHATIAN", JOptionPane.WARNING_MESSAGE);
                cbBln.requestFocus();
            }
        }

```

```

            return;
        }
        if
        (cbThn.getSelectedIndex() == -1)
        {
            PlaySound.PlaySound();

            JOptionPane.showMessageDialog(this, "Pilihan Tahun belum terpilih !", "PERHATIAN", JOptionPane.WARNING_MESSAGE);
            cbThn.requestFocus();
            return;
        }
        jumHari();
        hitTotal();
        jmlTot();
    }

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

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

```

LISTING DATABASE

phpMyAdmin

Database: dtzahwa (8)

dtzahwa (8)

admin, diskon, dtransaksi, jenis_prw, karyawan, konsumen, perawatan, transaksi

localhost > dtzahwa > jenis_prw

Browse Structure SQL Search Insert Export Import Operations Empty Drop

Showing rows 0 - 8 (9 total, Query took 0.0002 sec)

SELECT * FROM `jenis_prw` LIMIT 0 - 30

Profiling [Edit] [Explain SQL] [Create PHP Code] [Refresh]

Show: 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

Sort by key: None

+ Options

		k_d_Jenis	Nama	
<input type="checkbox"/>			J000	Gunting Rambut
<input type="checkbox"/>			J001	Perawatan Rambut
<input type="checkbox"/>			J002	Style/Trend Rambut
<input type="checkbox"/>			J003	Perawatan Wajah
<input type="checkbox"/>			J004	Make Over
<input type="checkbox"/>			J005	Make Up
<input type="checkbox"/>			J006	Perawatan Tangan & Kaki
<input type="checkbox"/>			J007	Paket Perawatan
<input type="checkbox"/>			J008	Perawatan Tubuh

Check All / Uncheck All With selected:

phpMyAdmin

Database: dtzahwa (8)

dtzahwa (8)

admin, diskon, dtransaksi, jenis_prw, karyawan, konsumen, perawatan, transaksi

localhost > dtzahwa > perawatan

Browse Structure SQL Search Insert Export Import Operations Empty Drop

Showing rows 0 - 29 (30 total, Query took 0.0005 sec)

SELECT * FROM `perawatan` LIMIT 0 - 30

Profiling [Edit] [Explain SQL] [Create PHP Code] [Refresh]

Show: 30 row(s) starting from record # 30

in horizontal mode and repeat headers after 100 cells

Sort by key: None

+ Options

		Kd_Prw	Nama	Kd_Jenis	Kategori	Harga	Keterangan	
<input type="checkbox"/>			P000	Cuci Gunting Pria	J000	NULL	25000	
<input type="checkbox"/>			P001	Cuci Gunting + Blow Wanita	J000	NULL	25000	
<input type="checkbox"/>			P002	Gunting Poni	J000	NULL	10000	
<input type="checkbox"/>			P003	Cuci Blow + Tonic	J001	NULL	25000	
<input type="checkbox"/>			P041	Sewa Jank	J004	NULL	15000	
<input type="checkbox"/>			P004	Cuci Catok / Blow Variasi	J001	NULL	35000	
<input type="checkbox"/>			P005	Creambath Tradisional	J001	NULL	30000	massage punggung dan tangan kurang lebih 60 menit
<input type="checkbox"/>			P006	Creambath NR	J001	NULL	50000	massage punggung dan tangan kurang lebih 60 menit
<input type="checkbox"/>			P007	Creambath Makarizo	J001	NULL	57000	massage punggung dan tangan kurang lebih 60 menit
<input type="checkbox"/>			P040	Sewa Kebaya	J004	NULL	40000	
<input type="checkbox"/>			P008	Hair Spa Tradisional	J001	NULL	45000	massage punggung dan tangan kurang lebih 60 menit
<input type="checkbox"/>			P009	Hair Spa Makarizo	J001	NULL	50000	massage punggung dan tangan kurang lebih 60 menit
<input type="checkbox"/>			P010	Hair Spa Loreal	J001	NULL	55000	massage punggung dan tangan kurang lebih 60 menit
<input type="checkbox"/>			P011	Hair Mask Milk	J001	NULL	50000	massage punggung dan tangan kurang lebih 60 menit
<input type="checkbox"/>			P012	Hair Mask Makarizo	J001	NULL	60000	massage punggung dan tangan kurang lebih 60 menit
<input type="checkbox"/>			P039	Hair Mask Loreal	J001	NULL	65000	massage punggung dan tangan kurang lebih 60 menit
<input type="checkbox"/>			P013	Smoothing Matrix	J002	NULL	200000	

phpMyAdmin

Database: dtzahwa (8)

dtzahwa (8)

admin, diskon, dtransaksi, jenis_prw, karyawan, konsumen, perawatan, transaksi

localhost > dtzahwa > karyawan

Browse Structure SQL Search Insert Export Import Operations Empty Drop

Showing rows 0 - 4 (5 total, Query took 0.0005 sec)

SELECT * FROM `karyawan` LIMIT 0 - 30

Profiling [Edit] [Explain SQL] [Create PHP Code] [Refresh]

Show: 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

Sort by key: None

+ Options

		Kd_Karyawan	Nama	Tanggal_Lahir	Alamat	Kota	Telepon	Hp	Ktp	
<input type="checkbox"/>			K000	Harhisma	1979-03-31	Nitikan	Yogyakarta	0274380749	081904244274	5364854789467
<input type="checkbox"/>			K001	Anita Herawati	1979-06-06	Nitikan Baru Gg. Leo No. 26	Yogyakarta	0274380749	081904244275	123456789
<input type="checkbox"/>			K002	Rochma Wati	1991-05-14	Mangkuyudan MJ III / 365 RT 27 RW 08	Yogyakarta		085729020093	341000000000003
<input type="checkbox"/>			K003	Vitha	1980-06-30	Nitikan	Yogyakarta	0274232323	081212121212	453454365675465
<input type="checkbox"/>			K004	Tria Aprilia	1991-05-25	Jl. Glagah Sari	Yogyakarta		0274778975	033311165697785

phpMyAdmin

Database: dtzahwa (8)

dtzahwa (8)

- admin
- diskon
- dtransaksi
- jenis_prv
- karyawan
- konsumen
- perawatan
- transaksi

Showing rows 0 - 29 (37 total, Query took 0.0004 sec)

SELECT * FROM `karyawan` LIMIT 0, 30

Profiling | Edit | Explain SQL | Create PHP Code | Refresh

Show: 30 row(s) starting from record # 30

in horizontal mode and repeat headers after 100 cells

Sort by key: None

+ Options

	Kd_Transaksi	Tanggal	Kd_Karyawan	Kd_Konsumen	Total	Diskon	User
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	N030	2015-02-16	K003	P003	335000	0	yudhi
	N029	2015-02-13	K001	P012	475000	71250	yudhi
	N028	2015-02-13	K002	P012	315000	0	yudhi
	N027	2015-02-01	K001	P001	135000	0	yudhi
	N026	2015-01-11	K001	P005	241750	38250	Rohma
	N025	2015-01-03	K003	P002	125000	0	Rohma
	N024	2015-01-03	K000	P---	254000	0	Rohma
	N023	2014-12-11	K002	P003	170000	0	Rohma
	N022	2014-12-10	K002	P006	140000	0	Rohma
	N021	2014-12-09	K002	P003	16750	2250	Rohma
	N020	2014-12-01	K002	P006	75000	0	Rohma
	N019	2014-12-01	K001	P008	180000	20000	Rohma
	N018	2014-12-01	K001	P002	35000	0	Rohma
	N017	2014-11-25	K001	P---	124500	5500	Rohma
	N016	2014-11-25	K002	P004	88000	24250	Rohma
	N015	2014-11-22	K001	P003	321750	8250	Rohma
	N014	2014-11-16	K003	P---	252500	7500	Rohma
	N013	2014-11-11	K001	P004	276250	3750	Rohma

phpMyAdmin

Database: dtzahwa (8)

dtzahwa (8)

- admin
- diskon
- dtransaksi
- jenis_prv
- karyawan
- konsumen
- perawatan
- transaksi

Showing rows 0 - 29 (69 total, Query took 0.0005 sec)

SELECT * FROM `dtransaksi` LIMIT 0, 30

Profiling | Edit | Explain SQL | Create PHP Code | Refresh

Show: 30 row(s) starting from record # 30

in horizontal mode and repeat headers after 100 cells

+ Options

	Kd_Transaksi	Kd_Prw	Nama	Harga
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	N016	P000	Cuci Gunting Pria	25000
	N017	P043	Latulip	75000
	N017	P047	Bleaching Tangan	49500
	N018	P084	Cuci Gunting + Blow Wanita B	35000
	N026	P047	Bleaching Tangan	46750
	N026	P013	Smoothing Matrix	170000
	N026	P003	Cuci Blow + Tonic	25000
	N025	P045	Lt Pro	100000
	N025	P003	Cuci Blow + Tonic	25000
	N024	P062	Paket Treatment	145000
	N024	P043	Latulip	75000
	N024	P082	Cuci Gunting Pria C	34000
	N023	P061	Paket Lulur	110000
	N023	P049	Foot Spa + Massage Bubble	45000
	N023	P041	Sewa Jarik	15000
	N022	P000	Cuci Gunting Pria	25000
	N022	P047	Bleaching Tangan	55000
	N022	P022	Facial Biokos	60000

phpMyAdmin

Database: dtzahwa (8)

dtzahwa (8)

admin
diskon
dtransaksi
jenis_prv
karyawan
konsumen
perawatan
transaksi

localhost ▶ dtzahwa ▶ konsumen

Browse Structure SQL Search Insert Export Import Operations Empty Drop

Showing rows 0 - 13 (14 total, Query took 0.0004 sec)

SELECT * FROM `konsumen`
LIMIT 0 - 28

Profiling [Edit] [Explain SQL] [Create PHP Code] [Refresh]

Show: 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

Sort by key: None

+ Options

	Kd_Konsumen	Nama	Tanggal_Lahir	Kelamin	Pekerjaan	Alamat	Kota	Telepon	
<input type="checkbox"/>	P000	Fitri Azzahra Syanfah	2004-04-21	W	Pelajar	Nitikan Baru	Yogyakarta	0274360749	08
<input type="checkbox"/>	P001	Fitria Syalwa Dyastari	1999-02-20	W	Pelajar	Nitikan Baru Gg. Leo No. 26	Yogyakarta	0274360749	08
<input type="checkbox"/>	P002	Anita Herawati	1979-06-06	W	Wiraswasta	Nitikan Baru Gg. leo	Yogyakarta	0274360749	08
<input type="checkbox"/>	P003	Harlia Ika Pratiwi Rais	1986-07-19	W	Bekerja	Golo Gg. Anyelir No. 1014 Umbulhargo 5 Pandeyan	Yogyakarta	0274360748	08
<input type="checkbox"/>	P005	Yayuk	1955-06-12	W	Rumah Tangga	Seturan	Yogyakarta	0274	08
<input type="checkbox"/>	P004	Harhisma Yudhistira Rais	1979-03-31	P	Tidak Bekerja	Nitikan Baru Gg. Leo 26 Umbulhargo 6	Yogyakarta	0274360749	08
<input type="checkbox"/>	P006	Fuad Abror Siagian	1977-12-31	P	Bekerja	Kemuning Condong Catur JTIII Sleman	Yogyakarta	0274881498	08
<input type="checkbox"/>	P007	Risma Yanti	1992-04-07	W	Belum Menikah	Godean	Bantul	02745656897	08

phpMyAdmin

Database: dtzahwa (8)

dtzahwa (8)

admin
diskon
dtransaksi
jenis_prv
karyawan
konsumen
perawatan
transaksi

localhost ▶ dtzahwa ▶ diskon

Browse Structure SQL Search Insert Export Import Operations Empty Drop

Showing rows 0 - 7 (8 total, Query took 0.0004 sec)

SELECT * FROM `diskon`
LIMIT 0 - 28

Profiling [Edit] [Explain SQL] [Create PHP Code] [Refresh]

Show: 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

Sort by key: None

+ Options

	kd_diskon	diskon	Keterangan
<input type="checkbox"/>	1	0	Rumah Tangga
<input type="checkbox"/>	2	0	Belum Menikah
<input type="checkbox"/>	3	0	Bekerja
<input type="checkbox"/>	4	0	Wiraswasta
<input type="checkbox"/>	5	10	Mahasiswa
<input type="checkbox"/>	6	10	Pelajar
<input type="checkbox"/>	7	15	Member
<input type="checkbox"/>	8	0	Tidak Bekerja

phpMyAdmin

Database: dtzahwa (8)

dtzahwa (8)

admin
diskon
dtransaksi
jenis_prv
karyawan
konsumen
perawatan
transaksi

localhost ▶ dtzahwa ▶ admin

Browse Structure SQL Search Insert Export Import Operations Empty Drop

Showing rows 0 - 3 (4 total, Query took 0.0004 sec)

SELECT * FROM `admin`
LIMIT 0 - 28

Profiling [Edit] [Explain SQL] [Create PHP Code] [Refresh]

Show: 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

Sort by key: None

+ Options

	No	User	Pass	Kd_Karyawan	Hak_Akses
<input type="checkbox"/>	0	admin	yudhi0215	k000	0
<input type="checkbox"/>	1	yudhi	yudhi0215	k000	1
<input type="checkbox"/>	2	aih	k001		1
<input type="checkbox"/>	3	rohma	kasir123	k002	1