LAMPIRAN

```
private  void  jMenuItem2ActionPerformed(java.awt.event.ActionEvent
evt) {
       // TODO add your handling code here:
        PerkiraanPersalinanDialog       dialog       =       new
PerkiraanPersalinanDialog(this, true);
        dialog.setTitle("Menghitung Perkiraan Persalianan");
        dialog.setLocationRelativeTo(this);
        dialog.setVisible(true);
    }

    private                                                 void
jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        Trimester1Dialog  dialog  =  new  Trimester1Dialog(this,
true);
        dialog.setTitle("Menghitung Trimester 1");
        dialog.setLocationRelativeTo(this);
        dialog.setVisible(true);
    }

    private                                                 void
jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        Trimester2Dialog  dialog  =  new  Trimester2Dialog(this,
true);
        dialog.setTitle("Menghitung Trimester 2");
        dialog.setLocationRelativeTo(this);
        dialog.setVisible(true);
    }

    private                                                 void
jMenuItem5ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        Trimester3Dialog  dialog  =  new  Trimester3Dialog(this,
true);
        dialog.setTitle("Menghitung Trimester 3");
        dialog.setLocationRelativeTo(this);
        dialog.setVisible(true);
    }

    private                                                 void
jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        PenggunaDialog dialog = new PenggunaDialog(this, true);
        int i = dialog.showDialog();
        if (i == 1) {
            HitungApp.setPengguna(dialog.getPengguna());
            load();
        }
    }
    private boolean home = false;

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

        if (!home) {
            WelcomeDialog  welcomeFrame  =  new  WelcomeDialog(this,
true);
            welcomeFrame.setSize(700, 400);
            welcomeFrame.setLocationRelativeTo(this);
            welcomeFrame.setVisible(true);
            home = true;
            /*
            if (home) {
                if (HitungApp.getPengguna() == null) {
                    PenggunaDialog     penggunaDialog     =     new
PenggunaDialog(this, true);
                    penggunaDialog.setSize(700, 400);
                    int i = penggunaDialog.showDialog();
                    if (i == 1 && penggunaDialog.getPengguna() !=
null) {

HitungApp.setPengguna(penggunaDialog.getPengguna());

setTitle(HitungApp.getPengguna().getNama());
                        load();
                        penggunaDialog.setVisible(false);
                    }
                }
            }
             * *
             */
        }


    }

    private                                                 void
jMenuItem6ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        dispose();
    }

    private                                                 void
jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        PenggunaDialog dialog = new PenggunaDialog(this, true);
        int i = dialog.showDialog();
        if (i == 1) {
            HitungApp.setPengguna(dialog.getPengguna());
            load();
        }


    }

    private                                                 void
jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
```

```
        if (table.getSelectedRow() < 0) {
            return;
        }
        EntityManager em = HitungApp.getEntityManager();
        em.getTransaction().begin();
        em.remove(model.list.get(table.getSelectedRow()));
        em.getTransaction().commit();
        load();
    }

private  void  jMenuItem7ActionPerformed(java.awt.event.ActionEvent
evt) {
// TODO add your handling code here:
    MakanandanMinuman       obMakanandanMinuman       =       new
MakanandanMinuman();
    obMakanandanMinuman.setLocationRelativeTo(this);
    obMakanandanMinuman.setVisible(true);
}

private  void  jMenuItem8ActionPerformed(java.awt.event.ActionEvent
evt) {
// TODO add your handling code here:
    JasmanidanRohani obJasmanidanRohani = new JasmanidanRohani();
    obJasmanidanRohani.setLocationRelativeTo(this);
    obJasmanidanRohani.setVisible(true);
}

private  void  jMenuItem9ActionPerformed(java.awt.event.ActionEvent
evt) {
// TODO add your handling code here:
    AwalKehamilan obAwalKehamilan = new AwalKehamilan();
    obAwalKehamilan.setLocationRelativeTo(this);
    obAwalKehamilan.setVisible(true);
}

private void jMenuItem10ActionPerformed(java.awt.event.ActionEvent
evt) {
// TODO add your handling code here:
    KehamilanTri1 obKehamilanTri1 = new KehamilanTri1();
    obKehamilanTri1.setLocationRelativeTo(this);
    obKehamilanTri1.setVisible(true);
}

private void jMenuItem11ActionPerformed(java.awt.event.ActionEvent
evt) {
// TODO add your handling code here:
     KehTri2 obKehTri2 = new KehTri2();
    obKehTri2.setLocationRelativeTo(this);
    obKehTri2.setVisible(true);
}

private void jMenuItem12ActionPerformed(java.awt.event.ActionEvent
evt) {
// TODO add your handling code here:
    KehTri5 obKehTri5 = new KehTri5();
    obKehTri5.setLocationRelativeTo(this);
    obKehTri5.setVisible(true);
```

```
}

ation

    public void load() {
        EntityManager em = HitungApp.getEntityManager();
        Query query = em.createQuery("SELECT p FROM Proses p WHERE
p.pengguna       =       :pengguna       ORDER       BY       p.id
DESC").setParameter("pengguna", HitungApp.getPengguna());
        model.setData(query.getResultList());
    }

    class ProsesTableModel extends AbstractTableModel {

        private List<Proses> list;

        public ProsesTableModel() {
            list = new ArrayList<Proses>();
        }

        public void setData(List list) {
            this.list = list;
            fireTableDataChanged();
        }

        @Override
        public int getColumnCount() {
            return 5;
        }

        @Override
        public Object getValueAt(int rowIndex, int columnIndex) {
            switch (columnIndex) {
                case 0:
                    Integer jenis = list.get(rowIndex).getJenis();
                    switch (jenis) {
                        case 0:
                            return "Perkiraan Persalinan";
                        case 1:
                            return "Trimester 1";
                        case 2:
                            return "Trimester 2";
                        case 3:
                            return "Trimester 3";
                        default:
                            return "";

                    }
                case 1:
                    //return list.get(rowIndex).getHpht();
                    return      new      SimpleDateFormat("dd-MM-
yyyy").format(list.get(rowIndex).getHpht());
                case 2:
                    return      new      SimpleDateFormat("dd-MM-
yyyy").format(list.get(rowIndex).getTanggal());
                case 3:
                    return list.get(rowIndex).getSiklus();
                case 4:
                    return list.get(rowIndex).getHasil();
                default:
                    return null;
            }
        }

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

        @Override
        public String getColumnName(int column) {
            switch (column) {
                case 0:
                    return "Jenis";
                case 1:
                    return "HPHT";
                case 2:
                    return "Tanggal Hitung";
                case 3:
                    return "Siklus";
                case 4:
                    return "Hasil";
                default:
                    return "";
            }
        }
    }
}
```