

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Pada proses validasi program perencanaan fondasi telapak, selisih paling besar dari perbandingan perhitungan manual dan program adalah 1,4473 %. Selisih perhitungan tersebut terjadi karena adanya pembulatan pada proses perhitungan manual, pada proses menentukan tebal efektif fondasi. Pembulatan pada program hanya bisa dilakukan pada akhir proses perhitungan dan dilakukan di luar program, untuk kepentingan menentukan spasi tulangan fondasi. Dari hasil perancangan aplikasi, penulis dapat membuat beberapa kesimpulan :

1. Perencanaan secara manual fondasi telapak yang berbentuk segi empat momen uniaksial dengan beban konsentris dan eksentris akan membutuhkan waktu yang relative lama.
2. Aplikasi *mobile* perencanaan fondasi telapak berbasis *android* tersebut bisa diandalkan sebagai program hitung yang akurat dan efektif.
3. Ukuran file yang kecil ( $\pm 1$  MB) membuat aplikasi tersebut mudah untuk dibagikan ke perangkat lain.

## 5.2 Saran

Aplikasi *mobile* perencanaan fondasi telapak berbasis *android* tersebut tentu saja masih jauh dari sempurna dan harus banyak mengalami pembenahan, diantaranya :

1. Aplikasi tersebut harus bisa memberi ilustrasi gambar fondasi dengan skala tertentu.
2. Membuat aplikasi tersebut bisa dijalankan di berbagai macam sistem operasi *smartphone*, bukan hanya android saja.
3. Memperbaiki tampilan dan fungsi pada aplikasi, sehingga menjadi lebih menarik dan mudah digunakan.
4. Membuat sistem penyimpanan data pada aplikasi, yang bisa memudahkan pengguna untuk meninjau ulang pekerjaan yang sudah pernah dilakukan.
5. Aplikasi tersebut harus bisa digunakan untuk berbagai macam bentuk fondasi telapak.

## DAFTAR PUSTAKA

- Asroni, A. 2010. "*Kolom Fondasi & Balok T Bertulang*". GRAHA ILMU: Yogyakarta
- Buyens, J. 2001. "*Web Database Development*". Elex Media Komputindo: Jakarta
- Hashimi, S.Y., et al., 2010, *Pro Android 2*, Apress, New York
- Janner, S. 2009. Perancangan Basis Data. Andi: Yogyakarta.
- Nakazawa, K dkk. 1994. "*MEKANIKA TANAH & TEKNIK FONDASI*". Pradnya Paramita: Jakarta
- Riyanto. 2010. "*Membuat Sendiri Aplikasi Mobile GIS Platform Java ME, Blackberry & Android*". ANDI OFFSET: Yogyakarta
- Signorini, E., Hochmuth, P., 2010. Consumerization of the Mobile Enterprise. Boston: Yankee Group
- Panitia Teknik Konstruksi dan Bangunan. 2002. Perencanaan Struktur Beton Bertulang (SNI 03-2847-2002). Badan Standardisasi Nasional

Layout Home XML

```

<ScrollView
xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:tools="http://schemas.android.com/tools"

    android:orientation="vertical"

    android:layout_width="fill_parent"

    android:layout_height="fill_parent"

    android:gravity="center">
<RelativeLayout

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:paddingBottom="@dimen/activity_vertical_margin"

    android:paddingLeft="@dimen/activity_horizontal_margin"

    android:paddingRight="@dimen/activity_horizontal_margin"

    android:paddingTop="@dimen/activity_vertical_margin"

    tools:context=".Home" >
<TextView

    android:id="@+id/textView1"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:layout_alignParentLeft="true"

    android:layout_alignParentTop="true"

    android:text="Petunjuk Penggunaan : "

    android:textAlignment="center" />
<TextView

    android:id="@+id/textView1"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:layout_alignParentLeft="true"

    android:layout_alignParentTop="true"

    android:layout_below="@+id/textView1"

```

```

    android:text=" "

/>
<TextView

    android:id="@+id/textView2"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:layout_alignLeft="@+id/textView1"

    android:layout_below="@+id/textView1"

    android:text="COBOW merupakan program
perhitungan perencanaan fondasi telapak berbasis
android yang dibuat untuk mempermudah melakukan
proses trial and error pada perhitungan fondasi.

Program ini membutuhkan beberapa input dengan
satuan panjang meter (m), beban kilo Newton
(kN),berat jenis tanah dan beton kilo Newton per meter
kubik (kN/m3), kuat tekan dan tarik MegaPascal
(MPa), dan diameter tulangan milimeter (mm).

Acuan yang digunakan dalam perencanaan adalah
SNI 03-2847-2002.

" />
<TextView

    android:id="@+id/textView3"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:layout_alignLeft="@+id/textView2"

    android:layout_below="@+id/textView2"

    android:text="Input data struktur dan tanah : " />
<ImageView

    android:id="@+id/imageView1"

    android:layout_width="wrap_content"

    android:layout_height="500dp"

    android:layout_alignLeft="@+id/textView3"

    android:layout_below="@+id/textView3"

    android:src="@drawable/pictutorial" />
<TextView

    android:id="@+id/textView4"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

```

```

    android:layout_alignLeft="@+id/imageView1"

    android:layout_below="@+id/imageView1"

    android:text="Untuk memulai menghitung
    fondasi, diperlukan beberapa input data struktur dan
    tanah seperti pada gambar di atas.

```

Gambar tersebut merupakan patokan pada perhitungan program dan harus diisi secara lengkap menurut data yang ada.

Jika hendak mengosongkan salah satu baris, gunakan nilai nol (0).

```

    " />
<TextView
    android:id="@+id/textView5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/textView1"
    android:layout_below="@+id/textView4"
    android:text=" " />
<Button
    android:id="@+id/Lanjut"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/textView1"
    android:layout_alignParentBottom="true"
    android:layout_alignRight="@+id/textView2"
    android:layout_below="@+id/textView5"
    android:text="Lanjutkan" />
</RelativeLayout>
</ScrollView>

```

#### Home JAVA

```

package com.cobow;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;

```

```

import android.view.View;
import android.widget.Button;

public class Home extends Activity {

    @Override

    protected void onCreate(Bundle
    savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_home);

        Button Lanjut1 = (Button)
        findViewById(R.id.Lanjut);

        Lanjut1.setOnClickListener(new
        View.OnClickListener() {

            @Override

            public void
            onClick(View v) {

                // TODO Auto-generated method stub

                Intent intent = new Intent(v.getContext(),
                Home2.class);

                startActivity(intent);

            }

        });

    }

    @Override

    public boolean onCreateOptionsMenu(Menu
    menu) {

        // Inflate the menu; this adds
        items to the action bar if it is present.

        getMenuInflater().inflate(R.menu.home, menu);

        return true;

    }

}

```

#### Layout Home 2 XML

```

<ScrollView
xmlns:android="http://schemas.android.com/apk/res/an
droid"

xmlns:tools="http://schemas.android.com/tools"

```

```

android:orientation="vertical"

android:layout_width="fill_parent"

android:layout_height="fill_parent"

android:gravity="center">
<RelativeLayout

    android:layout_width="match_parent"

    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin">
    <TextView

        android:id="@+id/textView1"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:text="Input Data Beban :"/>
    <ImageView

        android:id="@+id/imageView1"

        android:layout_width="wrap_content"

        android:layout_height="250dp"

        android:layout_alignLeft="@+id/textView1"

        android:layout_below="@+id/textView1"

        android:src="@drawable/pictutorialbeban"/>
    <TextView

        android:id="@+id/textView2"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:layout_alignLeft="@+id/imageView1"

        android:layout_below="@+id/imageView1"

        android:text="Pada proses input data beban,
disediakan baris Pd dan Pl untuk beban hidup dan mati,
dengan kombinasi beban 1.2Pd + 1.6Pl.

```

Momen yang dipakai merupakan momen ultimate dengan arah sumbu x dan y.

Seperti pada gambar di atas, apabila pengguna menggunakan eksentrisitas sebagai perhitungannya,

maka momen tidak digunakan dan sebaiknya diisi dengan nilai nol (0), berlaku juga kebalikannya.

```

"/>
<TextView

    android:id="@+id/textView3"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:layout_below="@+id/textView2"

    android:layout_alignLeft="@+id/imageView1"

    android:text="Output Perencanaan :"/>
<TextView

    android:id="@+id/textView4"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:layout_below="@+id/textView3"

    android:layout_alignLeft="@+id/imageView1"

    android:text="Hasil perhitungan program akan
menunjukkan nilai ketahanan dan gaya yang bekerja,
untuk cek keamanan bisa dilakukan dengan
membandingkan nilai tersebut.

    Hasil yang ditampilkan merupakan perhitungan
tanpa menggunakan pembulatan, sehingga diperoleh
nilai yang asli dan pembulatan bisa dilakukan sendiri
oleh pengguna.

"/>
<TextView

    android:id="@+id/textView5"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:layout_below="@+id/textView4"

    android:layout_alignLeft="@+id/imageView1"

    android:text=""/>
<Button

    android:id="@+id/Input"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:layout_alignLeft="@+id/textView4"

```

```

        android:layout_alignParentBottom="true"
        android:layout_alignRight="@+id/textView4"
        android:layout_below="@+id/textView5"
        android:text="Input" />
</RelativeLayout>
</ScrollView>

Home 2 JAVA
package com.cobow;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class Home2 extends Activity{
    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home2);
        Button Input = (Button)
findViewById(R.id.Input);
        Input.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                Intent intent = new Intent(v.getContext(), Input.class);
                startActivity(intent);
            }
        });
    }
}

//
// @Override
        public boolean onCreateOptionsMenu(Menu
menu) {
        //
        // Inflate the menu; this adds
items to the action bar if it is present.
        //
        getMenuInflater().inflate(R.menu.home,
menu);
        //
        return true;
        //
    }
}

Layout Input XML
<ScrollView
xmlns:android="http://schemas.android.com/apk/res/an
droid"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="center">
    <TableLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <TextView
            android:layout_width="fill_parent"
            android:layout_height="20dip"
            android:text=""/>
        <LinearLayout
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:paddingLeft="40dip"
            android:paddingRight="40dip">
            <TextView
                android:layout_width="fill_parent"

```

```

    android:layout_height="wrap_content"
    android:text="Data Struktur"
    android:textSize="10sp"
    android:textColor="#000000"/>
<EditText
    android:id="@+id/textXkolom"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:hint="x kolom (m)"
    android:inputType="numberDecimal" />
<EditText
    android:id="@+id/textYkolom"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:hint="y kolom (m)"
    android:inputType="numberDecimal" />
<EditText
    android:id="@+id/textTipe"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:hint="tipe kolom 20,30,40"
    android:inputType="numberDecimal" />
<EditText
    android:id="@+id/textXfond"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:hint="X fondasi (m)"
    android:inputType="numberDecimal" />
<EditText
    android:id="@+id/textYfond"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:inputType="numberDecimal"
    android:hint="Y fondasi (m)"/>
<EditText
    android:id="@+id/textEXfond"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:hint="ex (m)"
    android:inputType="numberDecimal" />
<EditText
    android:id="@+id/textEYfond"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:inputType="numberDecimal"
    android:hint="ey (m)"/>
<EditText
    android:id="@+id/textTfond"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:inputType="numberDecimal"
    android:hint="Tebal fondasi (m)"/>
<EditText
    android:id="@+id/textFc"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:inputType="numberDecimal"
    android:hint="fc' (MPa)"/>
<EditText
    android:id="@+id/textFy"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:inputType="numberDecimal"
    android:hint="fy (MPa)"/>
<EditText
    android:id="@+id/textBesi"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"

```



```

        android:inputType="numberDecimal"
        android:hint="diameter"/>
<EditText
    android:id="@+id/textBjbeton"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:inputType="numberDecimal"
    android:hint="bj beton (KN/m3)"/>
<TextView
    android:layout_width="fill_parent"
    android:layout_height="20dip"
    android:text=""/>
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Data Tanah"
    android:textSize="10sp"
    android:textColor="#000000"/>
<EditText
    android:id="@+id/textDdt"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:inputType="numberDecimal"
    android:hint="daya dukung tanah (KN/m2)"/>
<EditText
    android:id="@+id/textBerattanah"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:inputType="numberDecimal"
    android:hint="berat tanah (KN/m3)"/>
<EditText
    android:id="@+id/textHa"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"

        android:inputType="numberDecimal"
        android:hint="tebal tanah di atas fondasi (m)"/>
<TextView
    android:layout_width="fill_parent"
    android:layout_height="20dip"
    android:text=""/>
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Data Beban"
    android:textSize="10sp"
    android:textColor="#000000"/>
<EditText
    android:id="@+id/textPd"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:hint="Dead load (KN)"
    android:inputType="numberDecimal" />
<EditText
    android:id="@+id/textPl"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:hint="Live load (KN)"
    android:inputType="numberDecimal" />
<EditText
    android:id="@+id/textMx"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:inputType="numberDecimal"
    android:hint="momen x (KNm)"/>
<EditText
    android:id="@+id/textMy"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"

```

```

        android:inputType="numberDecimal"
        android:hint="momen y (KNm)"/>
</LinearLayout>
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:layout_marginLeft="39sp"
    android:layout_marginRight="37sp">
<Button
    android:id="@+id/btnHitung"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="HITUNG"/>
</LinearLayout>
</TableLayout>
</ScrollView>

Input JAVA
package com.cobow;

import java.text.DecimalFormat;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class Input extends Activity{

    EditText xk, yk, ti, pd, pl, mx, my, Tf, ht, gt, gc, ddt,
    dia, Xf, Yf, Ex, Ey, fc, fy;

    String hasil;

    Button hitung;

        //        double D =
        Double.parseDouble(dia.getText().toString());

        //        double xf =
        Double.parseDouble(Xf.getText().toString());

        //        double yf =
        Double.parseDouble(Yf.getText().toString());

        //        double Pd =
        Double.parseDouble(pd.getText().toString());

        //        double Pl =
        Double.parseDouble(pl.getText().toString());

        //        double Mx =
        Double.parseDouble(mx.getText().toString());

        //        double My =
        Double.parseDouble(my.getText().toString());

        //        double Gt =
        Double.parseDouble(gt.getText().toString());

        //        double Ht =
        Double.parseDouble(ht.getText().toString());

        //        double tf =
        Double.parseDouble(Tf.getText().toString());

        //        double Gc =
        Double.parseDouble(gc.getText().toString());

        //        double ex =
        Double.parseDouble(Ex.getText().toString());

        //        double ey =
        Double.parseDouble(Ey.getText().toString());

        //        double DDT =
        Double.parseDouble(ddt.getText().toString());

        //        double Fc =
        Double.parseDouble(fc.getText().toString());

        //        double XK =
        Double.parseDouble(xk.getText().toString());

        //        double YK =
        Double.parseDouble(yk.getText().toString());

        //        double Ti =
        Double.parseDouble(ti.getText().toString());

        //        double Fy =
        Double.parseDouble(fy.getText().toString());

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        // TODO Auto-generated method stub

        super.onCreate(savedInstanceState);

        setContentView(R.layout.layout_input);

```

```

        xk = (EditText)
findViewById(R.id.textXkolom);

        yk = (EditText)
findViewById(R.id.textYkolom);

        ti = (EditText)
findViewById(R.id.textTipe);

        Xf = (EditText)
findViewById(R.id.textXfond);

        Yf = (EditText)
findViewById(R.id.textYfond);

        Ex = (EditText)
findViewById(R.id.textEXfond);

        Ey = (EditText)
findViewById(R.id.textEYfond);

        Tf = (EditText)
findViewById(R.id.textTfond);

        fc = (EditText)
findViewById(R.id.textFc);

        fy = (EditText)
findViewById(R.id.textFy);

        dia = (EditText)
findViewById(R.id.textBesi);

        gc = (EditText)
findViewById(R.id.textBjeton);

        ddt = (EditText)
findViewById(R.id.textDdt);

        gt = (EditText)
findViewById(R.id.textBerattallah);

        ht = (EditText)
findViewById(R.id.textHa);

        pd = (EditText)
findViewById(R.id.textPd);

        pl = (EditText)
findViewById(R.id.textPl);

        mx = (EditText)
findViewById(R.id.textMx);

        my = (EditText)
findViewById(R.id.textMy);

        //my.setTe

        hitung = (Button)
findViewById(R.id.btnHitung);

        hitung.setOnClickListener(new
View.OnClickListener() {

@Override
        public void
onClick(View v) {

// TODO
        Auto-generated method stub

        Bundle bun = new Bundle();

        bun.putString("sigma", hitungSigma());

        bun.putString("sigmaexover", hitungSigmaexover());

        bun.putString("sigmaekurang",
hitungSigmaekurang());

        bun.putString("sigmamin", hitungSigmamin());

        bun.putString("sigmaexovermin",
hitungSigmaexovermin());

        bun.putString("sigmaekurangmin",
hitungSigmaekurangmin());

//
        bun.putDouble("ddt", DDT);

        bun.putString("ddt", hitungDdt());

        bun.putString("vcx", hitungVcx());

        bun.putString("vux", hitungVux());

        bun.putString("vuxexs", hitungVuxexs());

        bun.putString("vuxover", hitungVuxover());

        bun.putString("vcy", hitungVcy());

        bun.putString("vuy", hitungVuy());

        bun.putString("vuyexs", hitungVuyexs());

        bun.putString("vuyover", hitungVuyover());

        bun.putString("vc_2", hitungVc_2());

        bun.putString("vu_2", hitungVu_2());

        bun.putString("vu_2exs", hitungVu_2exs());

        bun.putString("vu_2over",
hitungVu_2exs());

        bun.putString("tul_x_long",
hitungTul_x_long());

        bun.putString("tul_y_short",
hitungTul_y_short());

        bun.putString("tul_x_short",
hitungTul_x_short());

        bun.putString("tul_y_long",
hitungTul_y_long());
}
}

```

```

        bun.putString("diameter",
tampil_diameter());
        bun.putString("ex", tampil_ex());
        bun.putString("ey", tampil_ey());
        bun.putString("xf", tampil_xf());
        bun.putString("yf", tampil_yf());

        Intent intent = new Intent(v.getContext(),
Output.class);

        intent.putExtras(bun);

        startActivity(intent);
    });
}

public String tampil_diameter(){
    try{
        double D =
Double.parseDouble(dia.getText().toString());

        hasil = (String.valueOf(D));

        return hasil;
    } catch (Exception e) {
        e.printStackTrace();
    }

    return "";
}

public String tampil_ex(){
    try{
        double ex =
Double.parseDouble(Ex.getText().toString());

        hasil =
(String.valueOf(ex));

        return hasil;
    } catch (Exception e) {
        e.printStackTrace();
    }

    return "";
}

public String tampil_ey(){
    try{
        double ey =
Double.parseDouble(Ey.getText().toString());

        hasil =
(String.valueOf(ey));

        return hasil;
    } catch (Exception e) {
        e.printStackTrace();
    }

    return "";
}

public String tampil_xf(){
    try{
        double xf =
Double.parseDouble(Xf.getText().toString());

        hasil =
(String.valueOf(xf));

        return hasil;
    } catch (Exception e) {
        e.printStackTrace();
    }

    return "";
}

public String tampil_yf(){
    try{
        double yf =
Double.parseDouble(Yf.getText().toString());

        hasil =
(String.valueOf(yf));

        return hasil;
    } catch (Exception e) {
        e.printStackTrace();
    }

    return "";
}

```

```

    }

    public String hitungSigma() {
        try {
            double Puk = (double) (1.2*Pd + 1.6*Pl);
            DecimalFormat df = new DecimalFormat("###.####");
            hasil =
            (String.valueOf(df.format(Puk)));
            return hasil;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return "";
    }

    public String hitungSigmaexover() {
        try {
            double xf =
            Double.parseDouble(Xf.getText().toString());
            double yf =
            Double.parseDouble(Yf.getText().toString());
            double Pd =
            Double.parseDouble(pd.getText().toString());
            double Pl =
            Double.parseDouble(pl.getText().toString());
            double ex =
            Double.parseDouble(Ex.getText().toString());
            double sigmaexover =
            (double) (2/3)*((1.2*Pd + 1.6*Pl)/((0.5*xf-ex)*yf));
            DecimalFormat df =
            new DecimalFormat("###.####");
            hasil =
            (String.valueOf(df.format(sigmaexover)));
            return hasil;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return "";
    }
}

    public String hitungSigmaexkurang() {
        try {
            double xf =
            Double.parseDouble(Xf.getText().toString());
            double yf =
            Double.parseDouble(Yf.getText().toString());
            double Pd =
            Double.parseDouble(pd.getText().toString());
            double Pl =
            Double.parseDouble(pl.getText().toString());
            double Gt =
            Double.parseDouble(gt.getText().toString());
            double Ht =
            Double.parseDouble(ht.getText().toString());
            double tf =
            Double.parseDouble(Tf.getText().toString());
            double Gc =
            Double.parseDouble(gc.getText().toString());
            double ex =
            Double.parseDouble(Ex.getText().toString());
            double ey =
            Double.parseDouble(Ey.getText().toString());
            double
            sigmaexskurang = (double) ((1.2*Pd +
            1.6*Pl)/(xf*yf))*(1+((6*ex)/(xf))+((6*ey)/(yf))+((Ht*
            Gt)+(tf*Gc)));
            DecimalFormat df = new DecimalFormat("###.####");
            hasil = (String.valueOf(df.format(sigmaexskurang)));
            return hasil;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return "";
    }

    public String hitungSigmamin() {
        try {
            double xf =
            Double.parseDouble(Xf.getText().toString());
            double yf =
            Double.parseDouble(Yf.getText().toString());
            double Pd =
            Double.parseDouble(pd.getText().toString());

```

```

        double Pl =
Double.parseDouble(pl.getText().toString());

        double Mx =
Double.parseDouble(mx.getText().toString());

        double My =
Double.parseDouble(my.getText().toString());

        double Gt =
Double.parseDouble(gt.getText().toString());

        double Ht =
Double.parseDouble(ht.getText().toString());

        double tf =
Double.parseDouble(Tf.getText().toString());

        double Gc =
Double.parseDouble(gc.getText().toString());

        double sigmamin =
(double) ((1.2*Pd + 1.6*Pl)/(xf*yf) -
((Mx)/((yf*yf*xf/6))) - ((My)/((yf*xf*xf/6))) +
((Ht*Gt)+(tf*Gc));

        DecimalFormat df =
new DecimalFormat("###.####");

hasil = (String.valueOf(df.format(sigmamin)));

        return hasil;

    } catch (Exception e) {

        e.printStackTrace();

    }

    return "";

}

public String hitungSigmaexovermin() {

    try {

        double sigmaexovermin = (double) 0;

        DecimalFormat df = new DecimalFormat("###.####");

        hasil = (String.valueOf(df.format(sigmaexovermin)));

        return hasil;

    } catch (Exception e) {

        e.printStackTrace();

    }

    return "";

}

public String hitungSigmaexkurangmin() {

    try {

        double xf =
Double.parseDouble(Xf.getText().toString());

        double yf =
Double.parseDouble(Yf.getText().toString());

        double Pd =
Double.parseDouble(pd.getText().toString());

        double Pl =
Double.parseDouble(pl.getText().toString());

        double Gt =
Double.parseDouble(gt.getText().toString());

        double Ht =
Double.parseDouble(ht.getText().toString());

        double tf =
Double.parseDouble(Tf.getText().toString());

        double Gc =
Double.parseDouble(gc.getText().toString());

        double ex =
Double.parseDouble(Ex.getText().toString());

        double ey =
Double.parseDouble(Ey.getText().toString());

        double sigmaexkurangmin = (double) ((1.2*Pd +
1.6*Pl)/(xf*yf))*((1-((6*ex)/(xf))-
((6*ey)/(yf)))+(Ht*Gt)+(tf*Gc));

        DecimalFormat df = new DecimalFormat("###.####");

        hasil =
(String.valueOf(df.format(sigmaexkurangmin)));

        return hasil;

    } catch (Exception e) {

        e.printStackTrace();

    }

    return "";

}

public String hitungDdt() {

    try {

        double Fc =
Double.parseDouble(fc.getText().toString());

        double XK =
Double.parseDouble(xk.getText().toString());

        double YK =
Double.parseDouble(yk.getText().toString());

        double Pu = (double)
0.7*0.85*Fc*XK*YK*1000;

    }

```

```

    }
    return "";
}

DecimalFormat df =
new DecimalFormat("###.####");

    hasil =
(String.valueOf(df.format(Pu)));

    return hasil;
} catch (Exception e) {
    e.printStackTrace();
}
return "";
}

public String hitungVcx() {
    try {
        double D =
Double.parseDouble(dia.getText().toString());

        double yf =
Double.parseDouble(Yf.getText().toString());

        double tf =
Double.parseDouble(Tf.getText().toString());

        double Fc =
Double.parseDouble(fc.getText().toString());

        double ds = (double)
(75+(D/2));

        double d = (double)
(1000 * tf - ds);

        double Vc1 =
(double) (Math.sqrt(Fc)) / 6 * 1000 * yf * d / 1000;

        double piVc =
(double) Vc1*0.75;

        // format decimal
        DecimalFormat df =
new DecimalFormat("###.##");

        // format decimal

        hasil =
(String.valueOf(df.format(piVc)));

        return hasil;
    } catch (Exception e) {
        e.printStackTrace();
    }
}

}

return "";
}

public String hitungVux() {
    try {
        double XK =
Double.parseDouble(xk.getText().toString());

        double xf =
Double.parseDouble(Xf.getText().toString());

        double yf =
Double.parseDouble(Yf.getText().toString());

        double Pd =
Double.parseDouble(pd.getText().toString());

        double Pl =
Double.parseDouble(pl.getText().toString());

        double Mx =
Double.parseDouble(mx.getText().toString());

        double My =
Double.parseDouble(my.getText().toString());

        double Gt =
Double.parseDouble(gt.getText().toString());

        double Ht =
Double.parseDouble(ht.getText().toString());

        double tf =
Double.parseDouble(Tf.getText().toString());

        double Gc =
Double.parseDouble(gc.getText().toString());

        double sigma_min =
(double) (1.2*Pd + 1.6*Pl) / (xf*yf) - (Mx)/(yf*yf*xf/6)
- (My)/(yf*xf*xf/6) + ((Ht*Gt)+(tf*Gc));

        double D =
Double.parseDouble(dia.getText().toString());

        double ds = (double)
(75 + (D/2));

        double d = (double)
(1000 * tf - ds);

        double a = (double)
((1000*xf / 2) - (1000*XK / 2) - d) / 1000;

        double sigma_max =
(double) (1.2*Pd + 1.6*Pl) / (xf*yf) +
(Mx)/(yf*yf*xf/6) + (My)/(yf*xf*xf/6) +
((Ht*Gt)+(tf*Gc));
    }
}

```

```

        double sigma_a =
(double) sigma_min + (((xf-a)*(sigma_max-
sigma_min))/xf);

        double Vu1 =
(double) (a * yf * ((sigma_max + sigma_a) / 2));

        DecimalFormat df =
new DecimalFormat("###.#####");

        hasil =
(String.valueOf(df.format(Vu1)));

        return hasil;
    } catch (Exception e) {
        e.printStackTrace();
    }
    return "";
}

public String hitungVuxexs() {
    try {

        double D =
Double.parseDouble(dia.getText().toString());

        double xf =
Double.parseDouble(Xf.getText().toString());

        double yf =
Double.parseDouble(Yf.getText().toString());

        double Pd =
Double.parseDouble(pd.getText().toString());

        double Pl =
Double.parseDouble(pl.getText().toString());

        double Gt =
Double.parseDouble(gt.getText().toString());

        double Ht =
Double.parseDouble(ht.getText().toString());

        double tf =
Double.parseDouble(Tf.getText().toString());

        double Gc =
Double.parseDouble(gc.getText().toString());

        double ex =
Double.parseDouble(Ex.getText().toString());

        double ey =
Double.parseDouble(Ey.getText().toString());

        double XK =
Double.parseDouble(xk.getText().toString());

        double sigma_min =
(double) ((1.2*Pd + 1.6*Pl)/(xf*yf))*(1-((6*ex)/(xf)-
((6*ey)/(yf)))+(Ht*Gt)+(tf*Gc));

        double ds = (double)
(75 + (D/2));

        double d = (double)
(1000 * tf - ds);

        double aa = (double)
((1000*xf/2) - (ex*1000) - (1000*XK/2) - d) / 1000;

        double a = (double)
((1000*xf / 2) + (ex*1000) - (1000*XK / 2) - d) / 1000;

        double sigma_max =
(double) ((1.2*Pd +
1.6*Pl)/(xf*yf))*(1+((6*ex)/(xf))+((6*ey)/(yf)))+(Ht*
Gt)+(tf*Gc));

        double sigma_aa =
(double) sigma_min + (((xf-aa)*(sigma_max-
sigma_min))/xf);

        double sigma_a =
(double) sigma_min + (((a)*(sigma_max-
sigma_min))/xf);

        // a dari kiri ke tepi kol ,,,, aa dari
kanan ke tepi kol

        double Vu1 =
(double) (a * yf * ((sigma_a + sigma_min) / 2));

        double Vu12 =
(double) (aa * yf * ((sigma_max + sigma_aa) / 2));

        double V = Vu1;

        if (V < Vu12)
        { V = Vu12; }
        else if (V > Vu12)
        { V = Vu1; }

        DecimalFormat df =
new DecimalFormat("###.#####");

        hasil =
(String.valueOf(df.format(V)));

        return hasil;
    } catch (Exception e) {
        e.printStackTrace();
    }
    return "";
}

```



```

    }

    public String hitungVuxover() {

        try {

            double D =
            Double.parseDouble(dia.getText().toString());

            double xf =
            Double.parseDouble(Xf.getText().toString());

            double yf =
            Double.parseDouble(Yf.getText().toString());

            double Pd =
            Double.parseDouble(pd.getText().toString());

            double Pl =
            Double.parseDouble(pl.getText().toString());

            double tf =
            Double.parseDouble(Tf.getText().toString());

            double ex =
            Double.parseDouble(Ex.getText().toString());

            double XK =
            Double.parseDouble(xk.getText().toString());

            double sigma_min =
            (double) 0;

            double ds = (double)
            (75 + (D/2));

            double d = (double)
            (1000 * tf - ds);

            double x = (double)
            3*(0.5*xf - ex);

            double aa = (double)
            ((1000*xf/2) - (ex*1000) - (1000*XK/2) - d) / 1000;

            double a = (double)
            ((1000*x) - (aa*1000) - (1000*XK) - (2*d)) / 1000;

            double sigma_max =
            (double) (2/3)*((1.2*Pd + 1.6*Pl)/((0.5*xf-ex)*yf));

            double sigma_a =
            (double) sigma_min + (((x-aa)*(sigma_max-
            sigma_min))/x);

            double sigma_aa =
            (double) sigma_min + (((a)*(sigma_max-
            sigma_min))/x);

            double Vu1 =
            (double) (a * yf * ((sigma_a + sigma_min) / 2));

            double Vu11 =
            (double) (aa * yf * ((sigma_max + sigma_aa) / 2));

            double V = Vu1;

            if (V < Vu11)

                { V = Vu11; }

            else if (V > Vu11)

                { V = Vu1; }

            DecimalFormat df = new
            DecimalFormat("###.####");

            hasil =
            (String.valueOf(df.format(V)));

            return hasil;

        } catch (Exception e) {

            e.printStackTrace();

        }

        return "";

    }

    public String hitungVcy() {

        try {

            double D =
            Double.parseDouble(dia.getText().toString());

            double xf =
            Double.parseDouble(Xf.getText().toString());

            double tf =
            Double.parseDouble(Tf.getText().toString());

            double Fc =
            Double.parseDouble(fc.getText().toString());

            double ds = (double)
            (75+(D/2));

            double d = (double)
            (1000 * tf - ds);

            double Vc1 =
            (double) (Math.sqrt(Fc)) / 6 * 1000 * xf * d / 1000;

            double piVc =
            (double) Vc1*0.75;

            // format decimal

            DecimalFormat df =
            new DecimalFormat("###.##");

            // format decimal

            hasil =
            (String.valueOf(df.format(piVc)));

        }

    }

```

```

        return hasil;
    } catch (Exception e) {
        e.printStackTrace();
    }
    return "";
}

public String hitungVuy() {
    try {
        double D =
Double.parseDouble(dia.getText().toString());
        double xf =
Double.parseDouble(Xf.getText().toString());
        double yf =
Double.parseDouble(Yf.getText().toString());
        double Pd =
Double.parseDouble(pd.getText().toString());
        double Pl =
Double.parseDouble(pl.getText().toString());
        double Mx =
Double.parseDouble(mx.getText().toString());
        double My =
Double.parseDouble(my.getText().toString());
        double Gt =
Double.parseDouble(gt.getText().toString());
        double Ht =
Double.parseDouble(ht.getText().toString());
        double tf =
Double.parseDouble(Tf.getText().toString());
        double Gc =
Double.parseDouble(gc.getText().toString());
        double YK =
Double.parseDouble(yk.getText().toString());
        double sigma_min =
(double) (1.2*Pd + 1.6*Pl) / (xf*yf) - (Mx)/(yf*yf*xf/6)
- (My)/(yf*xf*xf/6) + ((Ht*Gt)+(tf*Gc));
        double ds = (double)
(75 + (D/2));
        double d = (double)
(1000 * tf - ds);
        double a = (double)
((1000*yf / 2) - (1000*YK / 2) - d) / 1000;
        double sigma_max =
(double) (1.2*Pd + 1.6*Pl) / (xf*yf) +
(Mx)/(yf*yf*xf/6) + (My)/(yf*xf*xf/6) +
((Ht*Gt)+(tf*Gc));
        double sigma_a =
(double) sigma_min + (((yf-a)*(sigma_max-
sigma_min))/yf);
        double Vu1 =
(double) (a * xf * ((sigma_max + sigma_a) / 2));
        DecimalFormat df =
new DecimalFormat("###.####");
        hasil =
(String.valueOf(df.format(Vu1)));
        return hasil;
    } catch (Exception e) {
        e.printStackTrace();
    }
    return "";
}

public String hitungVuyexs() {
    try {
        double D =
Double.parseDouble(dia.getText().toString());
        double xf =
Double.parseDouble(Xf.getText().toString());
        double yf =
Double.parseDouble(Yf.getText().toString());
        double Pd =
Double.parseDouble(pd.getText().toString());
        double Pl =
Double.parseDouble(pl.getText().toString());
        double Gt =
Double.parseDouble(gt.getText().toString());
        double Ht =
Double.parseDouble(ht.getText().toString());
        double tf =
Double.parseDouble(Tf.getText().toString());
        double Gc =
Double.parseDouble(gc.getText().toString());
        double ex =
Double.parseDouble(Ex.getText().toString());
        double ey =
Double.parseDouble(Ey.getText().toString());

```

```

        double YK =
Double.parseDouble(yk.getText().toString());

        double sigma_min =
(double) ((1.2*Pd + 1.6*Pl)/(xf*yf))*((1-((6*ex)/(xf))-
((6*ey)/(yf)))+(Ht*Gt)+(tf*Gc));

        double ds = (double)
(75 + (D/2));

        double d = (double)
(1000 * tf - ds);

        double aa = (double)
((1000*yf/2) - (ey*1000) - (1000*YK/2) - d) / 1000;

        double a = (double)
((1000*yf / 2) + (ey*1000) - (1000*YK / 2) - d) / 1000;

        double sigma_max =
(double) ((1.2*Pd +
1.6*Pl)/(xf*yf))*((1+((6*ex)/(xf))+((6*ey)/(yf)))+(Ht*
Gt)+(tf*Gc));

        double sigma_aa =
(double) sigma_min + (((yf-aa)*(sigma_max-
sigma_min))/yf);

        double sigma_a =
(double) sigma_min + (((a)*(sigma_max-
sigma_min))/yf);

        double Vu1 =
(double) (a * xf * ((sigma_min + sigma_a) / 2));

        double Vu12 =
(double) (aa * xf * ((sigma_max + sigma_aa) / 2));

        double V = Vu1;

        if (V < Vu12)
        { V = Vu12; }

        else if (V > Vu12)
        { V = Vu1; }

        DecimalFormat df = new
DecimalFormat("###.####");

        hasil =
(String.valueOf(df.format(V)));

        return hasil;

    } catch (Exception e) {

        e.printStackTrace();

    }

    return "";

}

public String hitungVuyover() {

    try {

        double D =
Double.parseDouble(dia.getText().toString());

        double xf =
Double.parseDouble(Xf.getText().toString());

        double yf =
Double.parseDouble(Yf.getText().toString());

        double Pd =
Double.parseDouble(pd.getText().toString());

        double Pl =
Double.parseDouble(pl.getText().toString());

        // double Gt =
Double.parseDouble(gt.getText().toString());

        // double Ht =
Double.parseDouble(ht.getText().toString());

        double tf =
Double.parseDouble(Tf.getText().toString());

        // double Gc =
Double.parseDouble(gc.getText().toString());

        // double ex =
Double.parseDouble(Ex.getText().toString());

        double ey =
Double.parseDouble(Ey.getText().toString());

        double YK =
Double.parseDouble(yk.getText().toString());

        double sigma_min = (double) 0;

        double ds = (double)
(75 + (D/2));

        double d = (double)
(1000 * tf - ds);

        double y = (double)
3*(0.5*yf - ey);

        double aa = (double)
((1000*yf/2) - (ey*1000) - (1000*YK/2) - d) / 1000;

        double a = (double)
((1000*y) - (aa*1000) - (1000*YK) - (2*d)) / 1000;

        double sigma_max =
(double) (2/3)*((1.2*Pd + 1.6*Pl)/((0.5*yf-ey)*xf));

        double sigma_a =
(double) sigma_min + (((y-a)*(sigma_max-
sigma_min))/y);

        double sigma_aa =
(double) sigma_min + (((aa)*(sigma_max-
sigma_min))/y);

```

```

        double Vu1 =
(double) (a * xf * ((sigma_min + sigma_a) / 2));

        double Vu11 =
(double) (aa * xf * ((sigma_max + sigma_aa) / 2));

        double V = Vu1;

        if (V < Vu11)
        { V = Vu11; }

        else if (V > Vu11)
        { V = Vu1; }

        DecimalFormat df = new
DecimalFormat("###.####");

        hasil =
(String.valueOf(df.format(V)));

        return hasil;

    } catch (Exception e) {

        e.printStackTrace();

    }

    return "";

}

public String hitungVc_2() {

    try {

        double D =
Double.parseDouble(dia.getText().toString());

        double xf =
Double.parseDouble(Xf.getText().toString());

        double yf =
Double.parseDouble(Yf.getText().toString());

        double tf =
Double.parseDouble(Tf.getText().toString());

        double ex =
Double.parseDouble(Ex.getText().toString());

        double ey =
Double.parseDouble(Ey.getText().toString());

        double Fc =
Double.parseDouble(fc.getText().toString());

        double XK =
Double.parseDouble(xk.getText().toString());

        double YK =
Double.parseDouble(yk.getText().toString());

        double Ti =
Double.parseDouble(ti.getText().toString());

        double XKol =
(double) 1000*XK;

        double YKol =
(double) 1000*YK;

        double ds = (double)
(75+(D/2));

        double d = (double)
(1000*tf-ds);

        //jarak dari titik pusat
fondasai

        double AB = (double)
((1000*ex) + (1000*0.5*XK) + (0.5*d));

        double CD = (double)
((1000*ey) + (1000*0.5*YK) + (0.5*d));

        //selisih jarak dengan
setengah fondasi

        double XX = (double)
AB - (1000*0.5*xf);

        double YY = (double)
CD - (1000*0.5*yf);

        //setengah bentang
fondasi

        double XXx =
(double) (1000*0.5*xf);

        double YYy =
(double) (1000*0.5*yf);

        //keliling kritis geser
2 arah kasus

        double c1 = (double)
(XKol+d)+(XKol+d)+(YKol+d)+(YKol+d);

        double c2 = (double)
(XKol+d-XX)+(XKol+d-XX)+(YKol+d)+(YKol+d);

        double c3 = (double)
(XKol+d)+(XKol+d)+(YKol+d-YY)+(YKol+d-YY);

        double c4 = (double)
(XKol+d-XX)+(XKol+d-XX)+(YKol+d-YY)+(YKol+d-YY);

        double bo = c1;

        if (AB <= (XXx) &&
CD <= (YYy))

        { bo = c1; }

```

```

else if (AB > (XXx) && CD <=
(Yy))
    { bo = c2; }
else if (AB <= (XXx) && CD >
(Yy))
    { bo = c3; }
else if (AB > (XXx) && CD > (Yy))
    { bo = c4; }

(double) XK / YK;
(double) YK / XK;

double bet1 =
double bet2 =
double betaC = bet1;

if (XK > YK)
    {betaC = bet1;}
else if (XK < YK)
    {betaC = bet2;}

double Vc1 =
(double) 0.75 * (1 + 2 / betaC) * (Math.sqrt(Fc)) * bo *
d / 6000;
double Vc2 =
(double) 0.75 * (2 + Ti*d/bo) * (Math.sqrt(Fc)) * bo * d
/12000;
double Vc3 =
(double) 0.75 * 1/3 * (Math.sqrt(Fc)) * bo * d / 1000;

double V = Vc1;
if (V < Vc2 && V < Vc3)
    { V = Vc1; }
else if (V > Vc2 && Vc2 < Vc3)
    { V = Vc2; }
else if (V > Vc3 && Vc3 < Vc2)
    { V = Vc3; }

DecimalFormat df =
new DecimalFormat("###.####");

// dipilih yang paling
kecil dari v1 v2 v3

hasil =
(String.valueOf(df.format(V)));

return hasil;
} catch (Exception e) {
e.printStackTrace();
}
return "";
}

public String hitungVu_2() {
try {
double D =
Double.parseDouble(dia.getText().toString());
double xf =
Double.parseDouble(Xf.getText().toString());
double yf =
Double.parseDouble(Yf.getText().toString());
double Pd =
Double.parseDouble(pd.getText().toString());
double Pl =
Double.parseDouble(pl.getText().toString());
double Mx =
Double.parseDouble(mx.getText().toString());
double My =
Double.parseDouble(my.getText().toString());
double Gt =
Double.parseDouble(gt.getText().toString());
double Ht =
Double.parseDouble(ht.getText().toString());
double tf =
Double.parseDouble(Tf.getText().toString());
double Gc =
Double.parseDouble(gc.getText().toString());
double XK =
Double.parseDouble(xk.getText().toString());
double YK =
Double.parseDouble(yk.getText().toString());

double sigma_min =
(double) (1.2*Pd + 1.6*Pl) / (xf*yf) - (Mx)/(yf*xf*xf/6)
- (My)/(yf*yf*xf/6) + ((Ht*Gt)+(tf*Gc));

double sigma_max =
(double) (1.2*Pd + 1.6*Pl) / (xf*yf) +
(Mx)/(yf*xf*xf/6) + (My)/(yf*yf*xf/6) +
((Ht*Gt)+(tf*Gc));

double ds = (double)
(75+(D/2));
}
}

```

```

double d = (double)
(1000*tf-ds);

double Vu2 =
(double) (yf * xf - (YK + (d/1000)) * (XK + (d/1000)))
* ((sigma_max + sigma_min)/2);

DecimalFormat df =
new DecimalFormat("###.####");

hasil =
(String.valueOf(df.format(Vu2)));

return hasil;

} catch (Exception e) {
e.printStackTrace();
}

return "";
}

public String hitungVu_2exs() {
try {
double D =
Double.parseDouble(dia.getText().toString());

double xf =
Double.parseDouble(Xf.getText().toString());

double yf =
Double.parseDouble(Yf.getText().toString());

double Pd =
Double.parseDouble(pd.getText().toString());

double Pl =
Double.parseDouble(pl.getText().toString());

double Gt =
Double.parseDouble(gt.getText().toString());

double Ht =
Double.parseDouble(ht.getText().toString());

double Tf =
Double.parseDouble(Tf.getText().toString());

double Gc =
Double.parseDouble(gc.getText().toString());

double ex =
Double.parseDouble(Ex.getText().toString());

double ey =
Double.parseDouble(Ey.getText().toString());

double XK =
Double.parseDouble(xk.getText().toString());

double YK =
Double.parseDouble(yk.getText().toString());

double ds = (double)
(75+(D/2));

double d = (double)
(1000*tf-ds)/1000;

double AB = (double)
(ex + (0.5*XK) + (0.5*d));

double CD = (double)
(ey + (0.5*YK) + (0.5*d));

double XX = (double)
AB - (0.5*xf);

double YY = (double)
CD - (0.5*yf);

double Lo = XX;

if (AB <= xf && CD
<= yf)
{ Lo =
(XK+d)*(YK+d); }

else if (AB > xf && CD <= yf)
{ Lo = (XK+d-
XX)*(YK+d); }

else if (AB <= xf && CD > yf)
{ Lo = (XK+d)*(YK+d-YY); }

else if (AB > xf && CD > yf)
{ Lo = (XK+d-XX)*(YK+d-YY); }

double sigma_min =
(double) ((1.2*Pd + 1.6*Pl)/(xf*yf))*
(1-(((6*ex)/(xf))-
((6*ey)/(yf)))+(Ht*Gt)+(tf*Gc));

double sigma_max =
(double) ((1.2*Pd +
1.6*Pl)/(xf*yf))*
(1+(((6*ex)/(xf))+((6*ey)/(yf)))+(Ht*
Gt)+(tf*Gc));

double Vu2 =
(double) (yf * xf - Lo) * ((sigma_max + sigma_min)/2);

DecimalFormat df =
new DecimalFormat("###.####");

hasil =
(String.valueOf(df.format(Vu2)));

return hasil;

} catch (Exception e) {

e.printStackTrace();

}

return "";
}

```

```

    }

    public String hitungVu_2exsover() {
        try {
            double D =
Double.parseDouble(dia.getText().toString());

            double xf =
Double.parseDouble(Xf.getText().toString());

            double yf =
Double.parseDouble(Yf.getText().toString());

            double Pd =
Double.parseDouble(pd.getText().toString());

            double Pl =
Double.parseDouble(pl.getText().toString());

            double tf =
Double.parseDouble(Tf.getText().toString());

            double ex =
Double.parseDouble(Ex.getText().toString());

            double ey =
Double.parseDouble(Ey.getText().toString());

            double XK =
Double.parseDouble(xk.getText().toString());

            double YK =
Double.parseDouble(yk.getText().toString());

            double ds = (double)
(75+(D/2));

            double d = (double)
(1000*tf-ds)/1000;

            double AB = (double)
(ex + (0.5*XK) + (0.5*d));

            double CD = (double)
(ey + (0.5*YK) + (0.5*d));

            double XX = (double)
AB - (0.5*xf);

            double YY = (double)
CD - (0.5*yf);

            double Lo = XX;

            if (AB <= xf && CD
<= yf)
                { Lo =
(XK+d)*(YK+d); }

            else if (AB > xf && CD <= yf)
                { Lo = (XK+d-
XX)*(YK+d); }

            else if (AB <= xf && CD > yf)
                { Lo = (XK+d)*(YK+d-YY); }

            else if (AB > xf && CD > yf)
                { Lo = (XK+d-XX)*(YK+d-YY); }

            double sigma_min =
(double) 0;

            double sigma_max =
(double) (2/3)*((1.2*Pd + 1.6*Pl)/((0.5*xf-ex)*yf));

            double Vu2 =
(double) (yf * xf - Lo) * ((sigma_max + sigma_min)/2);

            DecimalFormat df =
new DecimalFormat("###.####");

            hasil =
(String.valueOf(df.format(Vu2)));

            return hasil;

        } catch (Exception e) {
            e.printStackTrace();
        }

        return "";
    }

    public String hitungTul_x_long() {
        try {
            double D =
Double.parseDouble(dia.getText().toString());

            double xf =
Double.parseDouble(Xf.getText().toString());

            double yf =
Double.parseDouble(Yf.getText().toString());

            double Pd =
Double.parseDouble(pd.getText().toString());

            double Pl =
Double.parseDouble(pl.getText().toString());

            double Mx =
Double.parseDouble(mx.getText().toString());

            double My =
Double.parseDouble(my.getText().toString());

            double Gt =
Double.parseDouble(gt.getText().toString());

            double Ht =
Double.parseDouble(ht.getText().toString());

            double tf =
Double.parseDouble(Tf.getText().toString());

```

```

double Gc =
Double.parseDouble(gc.getText().toString());

double ex =
Double.parseDouble(Ex.getText().toString());

double ey =
Double.parseDouble(Ey.getText().toString());

double Fc =
Double.parseDouble(fc.getText().toString());

double XK =
Double.parseDouble(xk.getText().toString());

double Fy =
Double.parseDouble(fy.getText().toString());

double
sigma_minover = (double) 0;

double
sigma_maxover = (double) (2/3)*((1.2*Pd +
1.6*Pl)/((0.5*xf-ex)*yf));

double sigma_maxexs
= (double) ((1.2*Pd +
1.6*Pl)/(xf*yf))*((6*ex)/(xf))+((6*ey)/(yf))+((Ht*
Gt)+(tf*Gc));

double sigma_minexs
= (double) ((1.2*Pd + 1.6*Pl)/(xf*yf))*((6*ex)/(xf))-
((6*ey)/(yf))+((Ht*Gt)+(tf*Gc));

double sigma_mina =
(double) (1.2*Pd + 1.6*Pl) / (xf*yf) - (Mx)/(yf*xf*xf/6)
- (My)/(yf*yf*xf/6) + ((Ht*Gt)+(tf*Gc));

double sigma_maxa =
(double) (1.2*Pd + 1.6*Pl) / (xf*yf) +
(Mx)/(yf*xf*xf/6) + (My)/(yf*yf*xf/6) +
((Ht*Gt)+(tf*Gc));

double sigma_max =
sigma_maxa;

if (ex == 0 && ey ==
0)

{ sigma_max =
sigma_maxa; }

else if (ex <= (xf/6) && ey == 0)

{ sigma_max =
sigma_maxexs; }

else if (ex == 0 && ey <= (yf/6))

{ sigma_max =
sigma_maxexs; }

else if (ex <= (xf/6) && ey <= (yf/6))

{ sigma_max =
sigma_maxexs; }

else if (ex > (xf/6) && ey > (yf/6))

{ sigma_max =
sigma_maxover; }

else if (ex > (xf/6) && ey == 0)

{ sigma_max =
sigma_minover; }

else if (ex == 0 && ey > (yf/6))

{ sigma_max =
sigma_minover; }

else if (ex > (xf/6) && ey <= (yf/6))

```



```

        { sigma_min =
sigma_minover; }
        else if (ex <= (xf/6) && ey > (yf/6))
        { sigma_min =
sigma_minover; }
        double x = (double)
(xf/2 - XK/2);
        double sigma_x =
(double) sigma_min + ((xf - x)*(sigma_max -
sigma_min)/xf);
        double Mu1 =
(double) (sigma_x * x * x *0.5) + ((sigma_max -
sigma_x)*x*x/3);
        // +
((1/3)*(sigma_max - sigma_x)*x*x)
        //((sigma_x * x * x
*0.5)
        double ds = (double)
(75 + (D/2));
        double d = (double)
(1000*tf-ds);
        double K = (double)
(Mu1 * 1000000) / (1000 * d * d * 0.8);
        // (1000 * d * d * 0.8)
        // (Mu1 * 1000000)
        double fcroperlu =
(double) (1-(2*K)/(0.85*Fc));
        double ro_min =
(double) 0.0018;
        double ro_max =
(double) 0.75*(((0.85*Fc)/Fy)*0.85*(600/(600+Fy)));
        double ro_perlu =
(double) ((0.85*Fc)/Fy)*(1- Math.sqrt(fcroperlu));
        double Asmin =
(double) ro_min*1000*tf*1000;
        double Asmax =
(double) ro_max*1000*tf*1000;
        double Asperlu =
(double) ro_perlu*1000*d;
        double ro = ro_perlu;
        double Asu = Asmin;
        if (ro <= ro_min)
        { Asu = Asmin; }
        else if (ro > ro_min && ro < ro_max)
        { sigma_min =
sigma_minover; }
        else if (ro >= ro_max)
        { Asu = Asmax; }
        double SP1 =
(double) (0.25 * (Math.PI) * (Math.pow(D,2)) *
1000)/Asu;
        double SP2 =
(double) 2 * tf * 1000;
        double SP3 =
(double) 450;
        double SP = SP1;
        if (SP < SP2 && SP <
SP3)
        { SP = SP1; }
        else if (SP2 < SP &&
SP2 < SP3)
        { SP = SP2; }
        else if (SP3 < SP &&
SP3 < SP2)
        { SP = SP3; }
        DecimalFormat df =
new DecimalFormat("###.#####");
        // "D" + D + " - " +
yang dulu..
        hasil =
(String.valueOf ("D" + D + " - " + df.format(SP)));
        return hasil;
    } catch (Exception e) {
        e.printStackTrace();
    }
    return "";
}

public String hitungTul_y_short() {
    try {
        double D =
Double.parseDouble(dia.getText().toString());
        double xf =
Double.parseDouble(Xf.getText().toString());
        double yf =
Double.parseDouble(Yf.getText().toString());
    }
}

```

```

double Pd = Double.parseDouble(pd.getText().toString());
double Pl = Double.parseDouble(pl.getText().toString());
double Mx = Double.parseDouble(mx.getText().toString());
double My = Double.parseDouble(my.getText().toString());
double Gt = Double.parseDouble(gt.getText().toString());
double Ht = Double.parseDouble(ht.getText().toString());
double tf = Double.parseDouble(tf.getText().toString());
double Gc = Double.parseDouble(gc.getText().toString());
double ex = Double.parseDouble(Ex.getText().toString());
double ey = Double.parseDouble(Ey.getText().toString());
double Fc = Double.parseDouble(fc.getText().toString());
double YK = Double.parseDouble(yk.getText().toString());
double Fy = Double.parseDouble(fy.getText().toString());
double sigma_maxover = (double) (2/3)*((1.2*Pd + 1.6*Pl)/((0.5*xf-ex)*yf));
double sigma_maxexs = (double) ((1.2*Pd + 1.6*Pl)/(xf*yf))*((1+((6*ex)/(xf))+((6*ey)/(yf)))+(Ht*Gt)+(tf*Gc));
double sigma_maxa = (double) (1.2*Pd + 1.6*Pl) / (xf*yf) + (Mx)/(yf*xf*xf/6) + (My)/(yf*yf*xf/6) + ((Ht*Gt)+(tf*Gc));
double x6 = (double) xf/6;
double y6 = (double) yf/6;
double sigma_max = sigma_maxa;
if (ex == 0 && ey == 0)
{ sigma_max = sigma_maxa; }
else if (ex == 0 && ey <= (y6))
{ sigma_max = sigma_maxexs; }
else if (ex <= (x6) && ey == 0)
{ sigma_max = sigma_maxover; }
else if (ex <= (x6) && ey <= (y6))
{ sigma_max = sigma_maxover; }
else if (ex > (x6) && ey > (y6))
{ sigma_max = sigma_maxover; }
else if (ex > (x6) && ey == 0)
{ sigma_max = sigma_maxover; }
else if (ex == 0 && ey > (y6))
{ sigma_max = sigma_maxover; }
else if (ex > (x6) && ey <= (y6))
{ sigma_max = sigma_maxover; }
else if (ex <= (x6) && ey > (y6))
{ sigma_max = sigma_maxover; }
double x = (double) (yf/2 - YK/2);
double Mu2 = (double) 0.5 * sigma_max * Math.pow(x,2);
double ds = (double) (75+ D + (D/2));
double d = (double) (1000*tf-ds);
double K = (double) (Mu2 * 1000000) / (1000 * d * d * 0.8);
double fcroperlu = (double) (1 - (2*K)/(0.85*Fc));
double ro_min = (double) 0.0018;
double ro_max = (double) 0.75*(((0.85*Fc)/Fy)*0.85*(600/(600+Fy)));
double ro_perlu = (double) ((0.85*Fc)/Fy)*(1 - Math.sqrt(fcroperlu));
double Asmin = (double) ro_min*1000*tf*1000;
double Asmax = (double) ro_max*1000*tf*1000;
double Asperlu = (double) ro_perlu*1000*d;
double ro = ro_perlu;
double Asu = Asmin;
if (ro <= ro_min)
{ Asu = Asmin; }
else if (ro > ro_min && ro < ro_max)

```

```

        { Asu = Asperlu; }
    else if (ro >= ro_max)
        { Asu = Asmax; }
    double Asp = (double) (2*yf*Asu) / (yf+xf);
    double Ast = (double) (Asu - Asp);
    double SP1 = (double) (0.25 * (Math.PI) *
(Math.pow(D,2)) * 1000)/Asp;
    double SP2 = (double) 2 * tf * 1000;
    double SP3 = (double) 450;
    double SP = SP1;
    if (SP < SP2 && SP <
SP3)
        {SP = SP1;}
    else if (SP2 < SP &&
SP2 < SP3)
        {SP = SP2;}
    else if (SP3 < SP &&
SP3 < SP2)
        {SP = SP3;}
    double ST1 = (double) (0.25 * (Math.PI) *
(Math.pow(D,2)) * 1000)/Ast;
    double ST2 = (double) 2 * tf * 1000;
    double ST3 = (double) 450;
    double ST = ST1;
    if (ST < ST2 && ST < ST3)
        {ST = ST1;}
    else if (ST2 < ST &&
ST2 < ST3)
        {ST = ST2;}
    else if (ST3 < ST &&
ST3 < ST2)
        {ST = ST3;}
    DecimalFormat df =
new DecimalFormat("###.####");
    // "D" + D + " - "
    hasil =
(String.valueOf("D" + D + " - " + df.format(SP)+ " / "
+ST));
    //+" / " +ST
    return hasil;
} catch (Exception e) {
    e.printStackTrace();
}
return "";
}
}
public String hitungTul_y_long() {
    try {
        double D =
Double.parseDouble(dia.getText().toString());
        double xf =
Double.parseDouble(Xf.getText().toString());
        double yf =
Double.parseDouble(Yf.getText().toString());
        double Pd =
Double.parseDouble(pd.getText().toString());
        double Pl =
Double.parseDouble(pl.getText().toString());
        double Mx =
Double.parseDouble(mx.getText().toString());
        double My =
Double.parseDouble(my.getText().toString());
        double Gt =
Double.parseDouble(gt.getText().toString());
        double Ht =
Double.parseDouble(ht.getText().toString());
        double tf =
Double.parseDouble(Tf.getText().toString());
        double Gc =
Double.parseDouble(gc.getText().toString());
        double ex =
Double.parseDouble(Ex.getText().toString());
        double ey =
Double.parseDouble(Ey.getText().toString());
        double Fc =
Double.parseDouble(fc.getText().toString());
        double YK =
Double.parseDouble(yk.getText().toString());
        double Fy =
Double.parseDouble(fy.getText().toString());
    }
}
}

```

```

double
sigma_minover = (double) 0;

double
sigma_maxover = (double) (2/3)*((1.2*Pd +
1.6*PI)/((0.5*xf-ex)*yf));

double sigma_maxexs
= (double) ((1.2*Pd +
1.6*PI)/(xf*yf))*(1+((6*ex)/(xf))+((6*ey)/(yf)))+(Ht*
Gt)+(tf*Gc));

double sigma_minexs
= (double) ((1.2*Pd + 1.6*PI)/(xf*yf))*(1-((6*ex)/(xf))-
((6*ey)/(yf)))+(Ht*Gt)+(tf*Gc));

double sigma_minina =
(double) (1.2*Pd + 1.6*PI) / (xf*yf) - (Mx)/(yf*yf*xf/6)
- (My)/(yf*xf*xf/6) + ((Ht*Gt)+(tf*Gc));

double sigma_maxa =
(double) (1.2*Pd + 1.6*PI) / (xf*yf) +
(Mx)/(yf*yf*xf/6) + (My)/(yf*xf*xf/6) +
((Ht*Gt)+(tf*Gc));

double y6 = (double)
yf/6;

double x6 = (double)
xf/6;

double sigma_max =
sigma_maxa;

if (ex == 0 && ey ==
0)
{ sigma_max =
sigma_maxa; }

else if (ex <= (x6)
&& ey <= (y6))
{ sigma_max =
sigma_maxexs; }

else if (ex == 0 && ey <= (y6))
{ sigma_max =
sigma_maxexs; }

else if (ex <= (x6) && ey == 0)
{ sigma_max =
sigma_maxexs; }

else if (ex > (x6) && ey > (y6))
{ sigma_max =
sigma_maxover; }

else if (ex > (x6) && ey == 0)
{ sigma_max =
sigma_maxover; }

else if (ex == 0 && ey > (y6))
{ sigma_min =
sigma_minover; }

else if (ex <= (x6) && ey == 0)
{ sigma_min =
sigma_minexs; }

else if (ex > (x6) && ey > (y6))
{ sigma_min =
sigma_minover; }

else if (ex <= (x6) && ey > (y6))
{ sigma_min =
sigma_minover; }

else if (ex > (x6) && ey <= (y6))
{ sigma_min =
sigma_minover; }

else if (ex <= (x6) && ey > (y6))
{ sigma_min =
sigma_minover; }

else if (ex == 0 && ey > (y6))
{ sigma_min =
sigma_minover; }

else if (ex > (x6) && ey <= (y6))
{ sigma_min =
sigma_minover; }

else if (ex <= (x6) && ey > (y6))
{ sigma_min =
sigma_minover; }

else if (ex > (x6) && ey == 0)
{ sigma_min =
sigma_minover; }

else if (ex == 0 && ey > (y6))
{ sigma_min =
sigma_minover; }

```

```

double x = (double)
(yf/2 - YK/2);

double sigma_x =
(double) sigma_min + ((yf - x)*(sigma_max -
sigma_min)/yf);

double Mu1 =
(double) (sigma_x * x * x * 0.5) + ((sigma_max -
sigma_x)*x*x/3);

// +
((1/3)*(sigma_max - sigma_x)*x*x)
*0.5)

double ds = (double)
(75 + (D/2));

double d = (double)
(1000*tf-ds);

double K = (double)
(Mu1 * 1000000) / (1000 * d * d * 0.8);

// (1000 * d * d * 0.8)
// (Mu1 * 1000000)

double fcroperlu =
(double) (1-(2*K)/(0.85*Fc));

double ro_min =
(double) 0.0018;

double ro_max =
(double) 0.75*(((0.85*Fc)/Fy)*0.85*(600/(600+Fy)));

double ro_perlu =
(double) ((0.85*Fc)/Fy)*(1 - Math.sqrt(fcroperlu));

double Asmin =
(double) ro_min*1000*tf*1000;

double Asmax =
(double) ro_max*1000*tf*1000;

double Asperlu =
(double) ro_perlu*1000*d;

double ro = ro_perlu;

double Asu = Asmin;

if (ro <= ro_min)
    { Asu = Asmin; }

else if (ro > ro_min && ro < ro_max)
    { Asu = Asperlu; }

else if (ro >= ro_max)
    { Asu = Asmax; }

double SP1 =
(double) (0.25 * (Math.PI) * (Math.pow(D,2)) *
1000)/Asu;

double SP2 =
(double) 2 * tf * 1000;

double SP3 =
(double) 450;

SP3)
SP2 < SP3)
SP3 < SP2)

double SP = SP1;
if (SP < SP2 && SP <
{ SP = SP1; }
else if (SP2 < SP &&
{ SP = SP2; }
else if (SP3 < SP &&
{ SP = SP3; }

DecimalFormat df =
new DecimalFormat("###.#####");

hasil =
(String.valueOf("D" + D + " - " +df.format(SP)));

return hasil;

} catch (Exception e) {
e.printStackTrace();
}

return "";
}

public String hitungTul_x_short() {
try {

double D =
Double.parseDouble(dia.getText().toString());

double xf =
Double.parseDouble(Xf.getText().toString());

double yf =
Double.parseDouble(Yf.getText().toString());

double Pd =
Double.parseDouble(pd.getText().toString());

double Pl =
Double.parseDouble(pl.getText().toString());

double Mx =
Double.parseDouble(mx.getText().toString());

```

```

double My =
Double.parseDouble(my.getText().toString());

double Gt =
Double.parseDouble(gt.getText().toString());

double Ht =
Double.parseDouble(ht.getText().toString());

double tf =
Double.parseDouble(Tf.getText().toString());

double Gc =
Double.parseDouble(gc.getText().toString());

double ex =
Double.parseDouble(Ex.getText().toString());

double ey =
Double.parseDouble(Ey.getText().toString());

double Fc =
Double.parseDouble(fc.getText().toString());

double XK =
Double.parseDouble(xk.getText().toString());

double Fy =
Double.parseDouble(fy.getText().toString());

double
sigma_maxover = (double) (2/3)*((1.2*Pd +
1.6*PI)/((0.5*xf-ex)*yf));

double sigma_maxexs
= (double) ((1.2*Pd +
1.6*PI)/(xf*yf))*(1+((6*ex)/(xf))+((6*ey)/(yf)))+((Ht*
Gt)+(tf*Gc));

double sigma_maxa =
(double) (1.2*Pd + 1.6*PI) / (xf*yf) +
(Mx)/(yf*xf*xf/6) + (My)/(yf*yf*xf/6) +
((Ht*Gt)+(tf*Gc));

double sigma_max = sigma_maxa;
if (ex == 0 && ex == 0)
{ sigma_max = sigma_maxa; }
else if (ex <= (xf/6) && ey <= (yf/6))
{ sigma_max = sigma_maxexs; }
else if (ex == 0 && ey <= (yf/6))
{ sigma_max = sigma_maxexs; }
else if (ex <= (xf/6) && ey == 0)
{ sigma_max = sigma_maxexs; }
else if (ex > (xf/6) && ey > (yf/6))
{ sigma_max = sigma_maxover; }
else if (ex > (xf/6) && ey == 0)
{ sigma_max = sigma_maxover; }

{ sigma_max =
sigma_maxover; }
else if (ex == 0 && ey > (yf/6))
{ sigma_max =
sigma_maxover; }
else if (ex > (xf/6) && ey <= (yf/6))
{ sigma_max =
sigma_maxover; }
else if (ex <= (xf/6) && ey > (yf/6))
{ sigma_max =
sigma_maxover; }
double x = (double)
(xf/2 - XK/2);
double Mu2 =
(double) 0.5 * sigma_max * Math.pow(x,2);
double ds = (double)
(75+ D + (D/2));
double d = (double)
(1000*tf-ds);
double K = (double)
(Mu2 * 1000000) / (1000 * d * d * 0.8);
double fcroperlu =
(double) (1-(2*K)/(0.85*Fc));
double ro_min =
(double) 0.0018;
double ro_max =
(double) 0.75*(((0.85*Fc)/Fy)*0.85*(600/(600+Fy)));
double ro_perlu =
(double) ((0.85*Fc)/Fy)*(1- Math.sqrt(fcroperlu));
double Asmin =
(double) ro_min*1000*tf*1000;
double Asmax =
(double) ro_max*1000*tf*1000;
double Asperlu =
(double) ro_perlu*1000*d;
double ro = ro_perlu;
double Asu = Asmin;
if (ro <= ro_min)
{ Asu = Asmin; }
else if (ro > ro_min && ro < ro_max)
{ Asu = Asperlu; }
else if (ro >= ro_max)

```

```

        { Asu = Asmax; }
double Asp =
(double) (2*xf*Asu) / (yf+xf);
        double Ast = (double)
(Asu - Asp);
        double SP1 =
(double) (0.25 * (Math.PI) * (Math.pow(D,2)) *
1000)/Asp;
        double SP2 = (double) 2 * tf * 1000;
        double SP3 = (double) 450;
        double SP = SP1;

        if (SP < SP2 && SP <
SP3)
            { SP = SP1; }
        else if (SP2 < SP &&
SP3 < SP2)
            { SP = SP2; }
        else if (SP3 < SP &&
SP2 < SP3)
            { SP = SP3; }

        double ST1 =
(double) (0.25 * (Math.PI) * (Math.pow(D,2)) *
1000)/Ast;
        double ST2 =
(double) 2 * tf * 1000;
        double ST3 =
(double) 450;
        double ST = ST1;
        if (ST < ST2 && ST
< ST3)
            { ST = ST1; }
        else if (ST2 < ST &&
ST3 < ST2)
            { ST = ST2; }
        else if (ST3 < ST &&
ST2 < ST3)
            { ST = ST3; }

        DecimalFormat df =
new DecimalFormat("###.####");
        // "D"+D+" - "+SP+"
/ "+"
    } catch (Exception e) {
        e.printStackTrace();
    }
    return "";
}
}
}
}

Layout Output XML
<ScrollView
xmlns:android="http://schemas.android.com/apk/res/an
droid"

    android:layout_width="fill_parent"

    android:layout_height="fill_parent"

    android:gravity="center"

    android:orientation="vertical" >

    <TableLayout

        android:layout_width="fill_parent"

        android:layout_height="wrap_content"

        android:layout_marginTop="10dp"

        android:orientation="horizontal" >

        <TableRow

            android:id="@+id/tableRow1"

            android:layout_width="match_parent"

            android:layout_height="wrap_content"

            android:layout_marginLeft="20dp"

            android:layout_marginRight="15dp"

            android:gravity="center_horizontal|fill_horizontal"

        >

            <TextView

                android:id="@+id/texttegangan"

                android:layout_width="wrap_content"

                android:layout_height="wrap_content"

                android:textSize="10dp"

```

```

        android:text="Kuat Dukung :." />
        </TableRow>
<TableRow
    android:id="@+id/tableRow2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="15dp"
    android:gravity="center_horizontal|fill_horizontal"
>
    <TextView
        android:id="@+id/txtSig"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Puk" />
    <EditText
        android:id="@+id/textSigma"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10" >
        <requestFocus />
    </EditText>
</TableRow>
<TableRow
    android:id="@+id/tableRow3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
        android:layout_marginLeft="20dp"
    android:layout_marginRight="15dp"
    android:gravity="center_horizontal|fill_horizontal"
>
    <TextView
        android:id="@+id/txtddt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Pu" />
    <EditText
        android:id="@+id/textDdt"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10" />
</TableRow>
<TableRow
    android:id="@+id/tableRow4"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="15dp"
    android:layout_marginTop="15dp"
    android:gravity="center_horizontal|fill_horizontal"
>
    <TextView
        android:id="@+id/textkuatgeser1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="10dp"
        android:text="Geser 1 arah :." />
</TableRow>
<TableRow
    android:id="@+id/tableRow5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="15dp"
    android:gravity="center_horizontal|fill_horizontal"
>
    <TextView
        android:id="@+id/textVcx"

```



```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Vc x" />
    <EditText
        android:id="@+id/textVc_x"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10" />
</TableRow>
<TableRow
    android:id="@+id/tableRow6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="15dp"
    android:gravity="center_horizontal|fill_horizontal"
    >
    <TextView
        android:id="@+id/textVux"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Vu x" />
    <EditText
        android:id="@+id/textVu_x"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10" />
</TableRow>
<TableRow
    android:id="@+id/tableRow7"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="15dp"
    android:gravity="center_horizontal|fill_horizontal"
    >
    <TextView
        android:id="@+id/txtVcy"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Vc y" />
    <EditText
        android:id="@+id/textVc_y"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10" >
    </EditText>
</TableRow>
<TableRow
    android:id="@+id/tableRow8"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="15dp"
    android:gravity="center_horizontal|fill_horizontal"
    >
    <TextView
        android:id="@+id/txtVuy"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Vu y" />
    <EditText
        android:id="@+id/textVu_y"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10" />
</TableRow>
<TableRow
    android:id="@+id/tableRow9"

```

```

    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="15dp"
    android:layout_marginRight="5dp"
    android:gravity="center_horizontal|fill_horizontal"
  >
  <TextView
    android:id="@+id/textkuatgeser2"
    android:layout_width="wrap_content"
    android:textSize="10dp"
    android:layout_height="wrap_content"
    android:text="Geser 2 arah : " />
</TableRow>
<TableRow
  android:id="@+id/tableRow10"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_marginLeft="20dp"
  android:layout_marginRight="15dp"
  android:gravity="center_horizontal|fill_horizontal"
  >
  <TextView
    android:id="@+id/textVc2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Vc 2" />
  <EditText
    android:id="@+id/textVc_2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10" />
</TableRow>
<TableRow
  android:id="@+id/tableRow11"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_marginLeft="20dp"
  android:layout_marginRight="15dp"
  android:gravity="center_horizontal|fill_horizontal"
  >
  <TextView
    android:id="@+id/textVu2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Vu 2" />
  <EditText
    android:id="@+id/textVu_2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10" />
</TableRow>
<TableRow
  android:id="@+id/tableRow12"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_marginLeft="20dp"
  android:layout_marginTop="15dp"
  android:layout_marginRight="15dp"
  android:gravity="center_horizontal|fill_horizontal"
  >
  <TextView
    android:id="@+id/textTulangan"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="10dp"
    android:text="Penulangan : " />
</TableRow>
<TableRow
  android:id="@+id/tableRow13"

```

```

android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginLeft="20dp"
android:layout_marginRight="15dp"
android:gravity="center_horizontal|fill_horizontal"
>
<TextView
    android:id="@+id/textTulx"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Tul x" />
<EditText
    android:id="@+id/textTul_x"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:ems="10" />
</TableRow>
<TableRow
    android:id="@+id/tableRow14"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="15dp"
    android:gravity="center_horizontal|fill_horizontal"
>
<TextView
    android:id="@+id/textTuly"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Tul y" />
<EditText
    android:id="@+id/textTul_y"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:ems="10" />
</TableRow>
</LinearLayout
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:layout_marginLeft="20sp"
    android:layout_marginRight="23sp">
<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="" />
<Button
    android:id="@+id/btnBack1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:textSize="15sp"
    android:text="Kembali" />
</LinearLayout>
</TableLayout>
</ScrollView>

```

Output JAVA

```

package com.cobow;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
public class Output extends Activity {

```

```

String sigma, sigmaexover, sigmaekurang,
sigmamin, sigmaexovermin, sigmaekurangmin, ddt,
vcx, vux, vuxexs, vuxover,
vcy, vuy, vuyexs, vuyover,
vc_2, vu_2, vu_2exs, vu_2over,
tul_x_long, tul_y_long, tul_x_short,
tul_y_short,
hasil,D, xf, yf, ex, ey;
EditText textSigma, textDdt, txtvcx, txtvux,
txtvcy, txtvuy, txtvc2, txtvu2, txttulx, txttuly;
Button back;
@Override
protected void onCreate(Bundle
savedInstanceState) {
    // TODO Auto-generated method
    stub
    super.onCreate(savedInstanceState);
    setContentView(R.layout.layout_output);
    Bundle b =
    getIntent().getExtras();
    D = b.getString("diameter");
    xf = b.getString("xf");
    yf = b.getString("yf");
    ex = b.getString("ex");
    ey = b.getString("ey");
    ddt = b.getString("ddt");
//1 arah x
    vcx = b.getString("vcx");
    vux = b.getString("vux");
    vuxexs = b.getString("vuxexs");
    vuxover =
    b.getString("vuxover");
//1 arah y
    vcy = b.getString("vcy");
    vuy = b.getString("vuy");
    vuyexs = b.getString("vuyexs");
    vuyover =
    b.getString("vuyover");
//2arah
    vc_2 = b.getString("vc_2");
    vu_2 = b.getString("vu_2");
    vu_2exs =
    b.getString("vu_2exs");
    vu_2over =
    b.getString("vu_2over");
    sigmamin =
    b.getString("sigmamin");
    sigmaexovermin =
    b.getString("sigmaexovermin");
    sigmaekurangmin =
    b.getString("sigmaekurangmin");
    sigmaekurang =
    b.getString("sigmaekurang");
    sigmaexover =
    b.getString("sigmaexover");
    sigma = b.getString("sigma");
    tul_x_long =
    b.getString("tul_x_long");
    tul_y_long =
    b.getString("tul_y_long");
    tul_x_short =
    b.getString("tul_x_short");
    tul_y_short =
    b.getString("tul_y_short");
    // memberi inisial "" pada string
    biar gak menampilkan nilai kosong
    textSigma = (EditText)
    findViewById(R.id.textSigma);
    textSigma.setText(sigma);
    textDdt = (EditText)
    findViewById(R.id.textDdt);
    textDdt.setText(String.valueOf(ddt));

```

```

        txtvcx = (EditText)
findViewById(R.id.textVc_x);

        txtvcx.setText(String.valueOf(vcx));

        double exxx =
Double.parseDouble(xf);

        double eyyy = Double.parseDouble(yf);

        double exx = (double) exxx/6;

        double eyy = (double) eyyy/6;

        String Vuallx = "";

        // memberi inisial "" pada string
        biar gak menampilkan nilai kosong

        if (Double.parseDouble(ex) == 0 &&
Double.parseDouble(ey) == 0)

            { Vuallx = vux; }

        else if (Double.parseDouble(ex) == 0 &&
Double.parseDouble(ey) <= eyy)

            { Vuallx = vuxexs; }

        else if (Double.parseDouble(ex) <= exx &&
Double.parseDouble(ey) == 0)

            { Vuallx = vuxexs; }

        else if (Double.parseDouble(ex) <= exx)

            { Vuallx = vuxexs; }

        else if (Double.parseDouble(ex) >= (-exx))

            { Vuallx = vuxexs; }

        else if (Double.parseDouble(ex) > exx)

            { Vuallx = vuxover; }

        else if (Double.parseDouble(ex) < (-exx))

            { Vuallx = vuxover; }

        txtvux = (EditText)
findViewById(R.id.textVu_x);

        txtvux.setText(String.valueOf(Vuallx));

        txtvcy = (EditText)
findViewById(R.id.textVc_y);

        txtvcy.setText(String.valueOf(vcy));

        String Vually = "";

        // memberi inisial "" pada string
        biar gak menampilkan nilai kosong

```

```

        if (Double.parseDouble(ex) == 0 &&
Double.parseDouble(ey) == 0)

            { Vually = vuy; }

        else if (Double.parseDouble(ey) == 0 &&
Double.parseDouble(ex) <= exx)

            { Vually = vuyexs; }

        else if (Double.parseDouble(ey) <= eyy &&
Double.parseDouble(ex) == 0)

            { Vually = vuyexs; }

        else if (Double.parseDouble(ey) <= eyy)

            { Vually = vuyexs; }

        else if (Double.parseDouble(ey) <= (-eyy))

            { Vually = vuyexs; }

        else if (Double.parseDouble(ey) > eyy &&
Double.parseDouble(ey) < (-eyy))

            { Vually = vuyover; }

        txtvuy = (EditText)
findViewById(R.id.textVu_y);

        txtvuy.setText(String.valueOf(Vually));

        txtvc2 = (EditText)
findViewById(R.id.textVc_2);

        txtvc2.setText(String.valueOf(vc_2));

        String Vuall2 = "";

        // memberi inisial "" pada string
        biar gak menampilkan nilai kosong

        if (Double.parseDouble(ex) == 0 &&
Double.parseDouble(ey) == 0)

            { Vuall2 = vu_2; }

        else if (Double.parseDouble(ex) <= (exx) &&
Double.parseDouble(ex) >= (-
exx))

            { Vuall2 = vu_2exs; }

        else if (Double.parseDouble(ey) <=
(eyy) &&
Double.parseDouble(ey) >= (-eyy))

            { Vuall2 = vu_2exs; }

        else if (Double.parseDouble(ex) > (exx) &&

```

```

        Double.parseDouble(ex) < (-
ex))
        { Vuall2 = vu_2over; }

        else if (Double.parseDouble(ey) > (ey) &&
                Double.parseDouble(ey) < (-
ey))
        { Vuall2 = vu_2over; }

        txtvu2 = (EditText)
findViewById(R.id.textVu_2);

        txtvu2.setText(String.valueOf(Vuall2));

        String Tullx = "";
        // memberi inisial "" pada string
        biar gak menampilkan nilai kosong

        if (Double.parseDouble(xf) ==
Double.parseDouble(yf))
        { Tullx = tul_x_long; }

        else if (Double.parseDouble(xf) >
Double.parseDouble(yf))
        { Tullx = tul_x_long; }

        else if (Double.parseDouble(xf) <
Double.parseDouble(yf))
        { Tullx = tul_x_short; }

        txttulx = (EditText)
findViewById(R.id.textTul_x);

        txttulx.setText(String.valueOf(Tullx));

        //"D" + D + " - " +

        String Tully = "";
        // memberi inisial "" pada string
        biar gak menampilkan nilai kosong

        if (Double.parseDouble(xf) ==
Double.parseDouble(yf))
        { Tully = tul_y_long; }

        else if (Double.parseDouble(yf) >
Double.parseDouble(xf))
        { Tully = tul_y_long; }

        else if (Double.parseDouble(yf) <
Double.parseDouble(xf))
        { Tully = tul_y_short; }

        txttuly = (EditText)
findViewById(R.id.textTul_y);

        txttuly.setText(String.valueOf(Tully));

        //"D" + D + " - " +

        back = (Button)
findViewById(R.id.btnBack1);

        back.setOnClickListener(new
View.OnClickListener() {

        @Override
        public void
onClick(View v) {

        // TODO
        Auto-generated method stub

        Intent
intent = new Intent(v.getContext(), Input.class);

        startActivity(intent);

        }

        });
}
}

```