**LAPORAN PRAKTIKUM  
PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL IX**

**JAVA DATABASE**



Oleh:

Ramadhan Wijaya

2211102208

IF-10-M

**S1 TEKNIK INFORMATIKA**

**INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

**2024**

# DASAR TEORI

## API untuk aplikasi GUI java

API atau application programing interface adalah seperangkat alat dan protocol yang digunakan untuk membuat perangkat lunak pada kondisi ini di java application. API menyediakan fungsionalitas untuk membuat antar muka grafis atau GUI(graphical User Interface). pada java application API yang klasik digunakan adalah java AWT(abstract window toolkit) API, Swing API, dan javaFX API. Berikut adalah penjelasannya :

* Java AWT

java AWT adalah toolkit GUI asli untuk java, namun karena perkembangan zaman java AWT digantikan oleh Swing dan javaFX karena keterbatasannya dalam fleksibilitas dan tampilannya yang lebih menarik. komponen AWT termasuk frame, Button, Label, TextField, dan TextArea.

Berikut adalah contoh sederhana aplikasi yang menggunakan java AWT menggunakan grid LayOut.

A screenshot of a computer

Description automatically generated

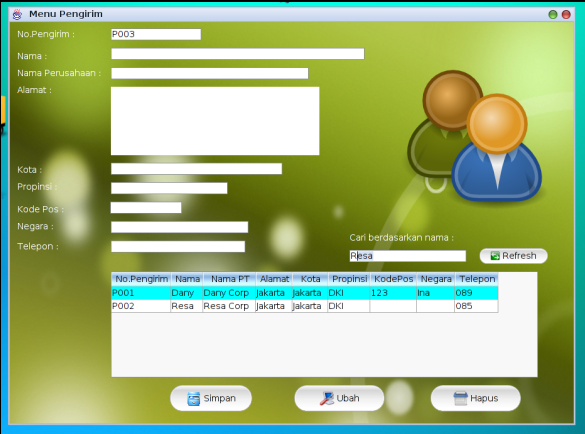
* Java Swing

Swing adalaha bagian dari java foundation classes (JFC) yang digunakan untuk membangun aplikasi GUI di java, sama halnya dengan AWT. Swing menyediakan komponen GUI yang lebih kaya dari AWT dan lebih mudah disesuiakan. Berikut adalah komponen komponen yang disediakan oleh java Swing

A screenshot of a computer menu

Description automatically generated

Berikut adalah contoh sederhana GUI dari aplikasi yang dibuat dari Java Swing

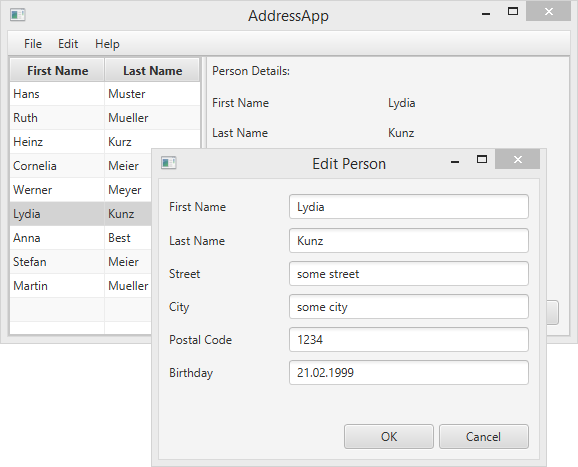


Teradapat table di bawah keterangan pengirim yang Dimana table tidak tersedia di java AWT.

* JavaFX

javaFX adalah framework yang lebih modern untuk membangun aplikasi di java GUI. javaFX dirancang untuk menggantikkan Swing dan menawarkan lebih banyak fitur dalam pengembangan antarmuka pengguna yang lebih canggih. Cara kerja JavaFX melibatkan beberapa komponen utama dan konsep yang perlu dipahami untuk mengembangkan aplikasi GUI yang efektif.

Berikut adalah contoh aplikasi yang dibuat oleh JavaFX



## JDBC dan JDBC API

JDBC (Java Database Connectivity) adalah sebuah API (Application Programming Interface) di Java yang memungkinkan aplikasi Java untuk berinteraksi dengan berbagai jenis sistem manajemen basis data (DBMS). JDBC menyediakan metode dan mekanisme standar untuk mengakses dan mengelola data yang tersimpan dalam basis data, sehingga pengembang dapat membuat aplikasi database-independen yang dapat bekerja dengan berbagai jenis DBMS.

JDBC API adalah sekumpulan kelas dan antarmuka dalam paket java.sql dan javax.sql yang menyediakan metode untuk berinteraksi dengan basis data. Komponen yang penting ada :

* DriverManager: Kelas ini digunakan untuk mengelola daftar driver basis data. Menggunakan metode getConnection() untuk membuat koneksi ke basis data.
* Connection: Antarmuka ini mewakili koneksi ke basis data. Menggunakan metode untuk membuat statement, mengelola transaksi, dan menutup koneksi.
* Statement: Antarmuka ini digunakan untuk menjalankan query SQL

Gambaran pengimportan JDBC

A black background with white text

Description automatically generated

Dari connection sampai statement adalah komponen komponen dari JDBC

1. JDBC Components dan Architecture

A diagram of a software application

Description automatically generated

Diatas merupakan gambar arsitektur relasi JDBC API dan JDBC Driver.

JDBC API berisi 2 set utama yaitu JDBC API untuk GUI dan JDBC API Driver untuk antarmuka database dari developer(tidak ditunjukan kepada user).

A black and white text on a white background

Description automatically generated

Diatas merupakan class class yang terdapat pada JDBC API.

A table of text with black text

Description automatically generated

Sedangkan yang ini adalah interfaces yang terdapat pada JDBC API.

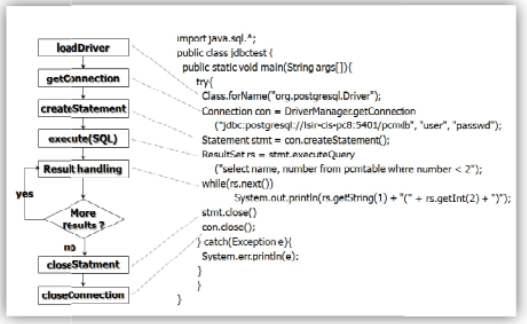
1. Cara Kerja JDBC

A diagram of software components

Description automatically generated

Diatas merupakan map yang menunjukan cara kerja JDBC. Simplenya pada saat memuat JDBC kita akan membuat koneksi, dan Driver manager pada tahap ini dilakukan di coding bagian Source. Selanjutnya ada tahap pemrosesan yang membuat dan mengeksekusi pernataan SQL menggunaakn “Statement” atau “PreparedStatement”, dan terakhir menggunakan Resultset untuk memproses hasil querinya. Pada masa terminasi menutup semua resource yang digunakan (koneksi, statement, result set) untuk melepaskan memori dan sumber daya lainnya.

Berikut adalah gambaran membuat koneksi, membuat statement, result handling condition dan terakhir terminasi di bagian closeConnection.



1. Tahapan Akses Database

A computer screen shot of a computer

Description automatically generated

Driver memingkinkan java untuk berkomunikasi dengan basis data tertentu dari yang kita maksud. Dari banyaknya basisdata seperti PostgreSQL, oracle, MySQL. Selanjutnya ke driver manager, di driver manager ini kita akan mengoneksikan ke basis data. Setelah itu ke connection untuk mewakili koneksi fisik ke basis data. Yang ke-4 ada statement untuk menjalankan pernyataan SQL terhadap basis data, pernyataan seperti select,insert,update, delete dll. Digunakan untuk mengelola hasil querinya. Terakhir masuk ke resultset, resultset adalah antarmuka yang mewakili hasil dari query SQL(yang ditunjukan ke user). Selain itu resultset juga membaca proses hasil queri yang dikembalikan oleh basis data ke user agar dapat melihat dengan jembata antarmuka yang friendly dengan user.

# GUIDED

Pertama kita buat databasenya di myadminphp dengan XAMPP.

A screenshot of a computer

Description automatically generated

Diatas adalah datababase yang saya buat untuk guided. Jadi ada database “dbdata” yang memiliki table “tbdata”. Dalam tbdata terdapat ketentuan atribut atribut yang sesuai yang kita jalankan di praktikum.

Setelah membuat database dan table yang sesuai ketentuan selanjutnya kita buka netbeansnya.

A black text on a white background

Description automatically generated

Dalam netbeasn kita membuat package projek yang didalamnya terdapat apk.java yang meruakan Jframe form dan koneksi.java yang merupakan java class

A black text on a white background

Description automatically generated

Selanjutnya pada folder libraries kita masukan mysql connector jarnya untuk mengedit GUI.

A screenshot of a computer

Description automatically generated

Selanjutnya kita masuk ke window->service. Lalu membuat menyambungkan phpMyAdmin dengan netbeansnya seperti diatas.



Ketika sudah akan muncul gambar seperti diatas di bagian service di dalam database.

|  |
| --- |
| /\*\*  \*  \* @author rama  \* 2211102208  \*/  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.SQLException;  import javax.swing.JOptionPane;  public class koneksi{  private static Connection mysqlkoneksi;  public static Connection koneksiDB() throws SQLException{  if(mysqlkoneksi == null){  try{  String DB = "jdbc:mysql://localhost:3306/dbdata";  String user = "root";  String pass = "";  DriverManager.registerDriver(new com.mysql.jdbc.Driver());  mysqlkoneksi = (Connection) DriverManager.getConnection(DB, user, pass);  }catch(Exception e){    }  }return mysqlkoneksi;  }  } |

Selanjutnya kita masuk ke class koneksi.java untuk memasukan program diatas yang Dimana fungsinya adalah membuat koneksi dengan mengimport *import java.sql.Connection;*

Setelah membuat koneksi ke phpMyAdmin sesuai gambar diatas kita selanjutnya membuat GUI untuk user di class APK.java

A screenshot of a computer

Description automatically generated

Berikut adalah UI buatan saya sesuai pertemuan di kelas praktikum.

Selanjutnya kita akan membuat setiap table, fieldText, dan tombol menjadi responsive agar bisa digunakan oleh user.

Agar responsive kita perlu menambahkan program di bagian source di kiri atas Ketika di class APK.java.



Diatas klik bagian source di class APK.java

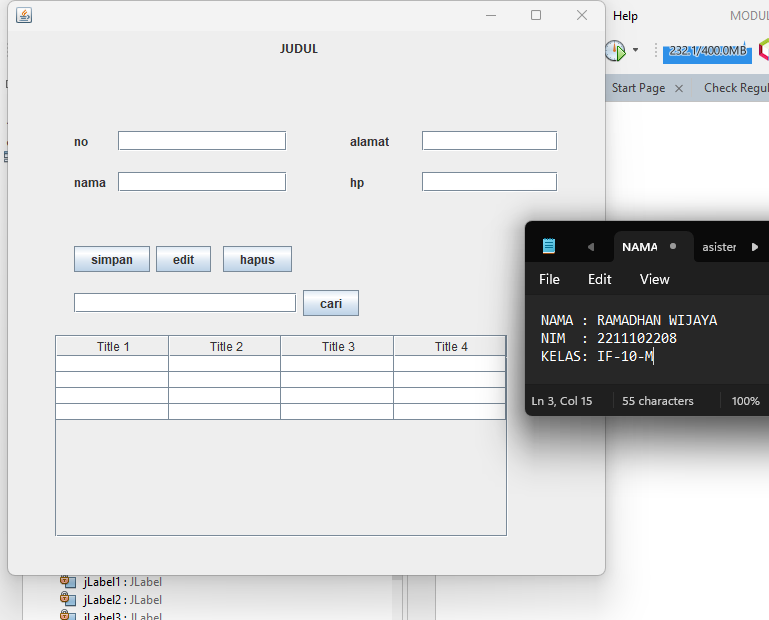
Setelah masuk ke bagian source kita selanjutnya memasukan code untuk setiap button, textfield, dan tablenya seperti dibawah ini programnya

|  |
| --- |
| package projek;  /\*\*  \*  \* @author rama  \* 2211102208  \*/  import java.sql.\*;  import javax.swing.table.\*;  import java.text.SimpleDateFormat;  import javax.swing.JOptionPane;  import com.mysql.cj.xdevapi.Table;  public class apk extends javax.swing.JFrame {  /\*\*  \* Creates new form apk  \*/  public apk() {  initComponents();  }  /\*\*  \* This method is called from within the constructor to initialize the form.  \* WARNING: Do NOT modify this code. The content of this method is always  \* regenerated by the Form Editor.  \*/  @SuppressWarnings("unchecked")  public void tampil\_data() {  DefaultTableModel tabel = new DefaultTableModel();  tabel.addColumn("NIM");  tabel.addColumn("NAMA");  tabel.addColumn("ALAMAT");  tabel.addColumn("TELP/HP");  try {  java.sql.Connection conn = (java.sql.Connection) projek.koneksi.koneksiDB();  String sql = "select \*from tbdata";  java.sql.PreparedStatement pst = conn.prepareStatement(sql);  ResultSet rs = pst.executeQuery(sql);  while (rs.next()) {  tabel.addRow(new Object[]{  rs.getString(1),  rs.getString(2),  rs.getString(3),  rs.getString(4),});  }  jTable2.setModel(tabel);  } catch (Exception e) {  }  }  public static void main(String args[]) {  java.awt.EventQueue.invokeLater(new Runnable() {  public void run() {  new apk().setVisible(true);  }  });  }  // <editor-fold defaultstate="collapsed" desc="Generated Code">  private void initComponents() {  jScrollPane1 = new javax.swing.JScrollPane();  jTable1 = new javax.swing.JTable();  jLabel1 = new javax.swing.JLabel();  jLabel2 = new javax.swing.JLabel();  jLabel3 = new javax.swing.JLabel();  jLabel4 = new javax.swing.JLabel();  jLabel5 = new javax.swing.JLabel();  no = new javax.swing.JTextField();  alamat = new javax.swing.JTextField();  nama = new javax.swing.JTextField();  hp = new javax.swing.JTextField();  tb\_simpan = new javax.swing.JButton();  tb\_edit = new javax.swing.JButton();  tb\_hapus = new javax.swing.JButton();  tb\_cari = new javax.swing.JButton();  cari = new javax.swing.JTextField();  jScrollPane2 = new javax.swing.JScrollPane();  jTable2 = new javax.swing.JTable();  jTable1.setModel(new javax.swing.table.DefaultTableModel(  new Object [][] {  {null, null, null, null},  {null, null, null, null},  {null, null, null, null},  {null, null, null, null}  },  new String [] {  "Title 1", "Title 2", "Title 3", "Title 4"  }  ));  jScrollPane1.setViewportView(jTable1);  setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);  jLabel1.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N  jLabel1.setText("JUDUL");  jLabel2.setText("no");  jLabel3.setText("nama");  jLabel4.setText("hp");  jLabel5.setText("alamat");  nama.addActionListener(new java.awt.event.ActionListener() {  public void actionPerformed(java.awt.event.ActionEvent evt) {  namaActionPerformed(evt);  }  });  tb\_simpan.setText("simpan");  tb\_simpan.addActionListener(new java.awt.event.ActionListener() {  public void actionPerformed(java.awt.event.ActionEvent evt) {  tb\_simpanActionPerformed(evt);  }  });  tb\_edit.setText("edit");  tb\_hapus.setText("hapus");  tb\_cari.setText("cari");  tb\_cari.addActionListener(new java.awt.event.ActionListener() {  public void actionPerformed(java.awt.event.ActionEvent evt) {  tb\_cariActionPerformed(evt);  }  });  cari.addActionListener(new java.awt.event.ActionListener() {  public void actionPerformed(java.awt.event.ActionEvent evt) {  cariActionPerformed(evt);  }  });  jTable2.setModel(new javax.swing.table.DefaultTableModel(  new Object [][] {  {null, null, null, null},  {null, null, null, null},  {null, null, null, null},  {null, null, null, null}  },  new String [] {  "Title 1", "Title 2", "Title 3", "Title 4"  }  ));  jScrollPane2.setViewportView(jTable2);  javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());  getContentPane().setLayout(layout);  layout.setHorizontalGroup(  layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  .addGroup(layout.createSequentialGroup()  .addGap(66, 66, 66)  .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  .addGroup(layout.createSequentialGroup()  .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  .addComponent(jLabel3)  .addComponent(jLabel2))  .addGap(12, 12, 12)  .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)  .addComponent(no, javax.swing.GroupLayout.DEFAULT\_SIZE, 169, Short.MAX\_VALUE)  .addComponent(nama))  .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)  .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  .addComponent(jLabel5)  .addComponent(jLabel4))  .addGap(33, 33, 33)  .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)  .addComponent(alamat, javax.swing.GroupLayout.DEFAULT\_SIZE, 136, Short.MAX\_VALUE)  .addComponent(hp))  .addGap(47, 47, 47))  .addGroup(layout.createSequentialGroup()  .addComponent(tb\_simpan)  .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  .addComponent(tb\_edit)  .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)  .addComponent(tb\_hapus)  .addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))  .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()  .addComponent(cari, javax.swing.GroupLayout.PREFERRED\_SIZE, 223, javax.swing.GroupLayout.PREFERRED\_SIZE)  .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  .addComponent(tb\_cari)  .addGap(246, 246, 246))))  .addGroup(layout.createSequentialGroup()  .addGap(47, 47, 47)  .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)  .addGap(0, 0, Short.MAX\_VALUE))  .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()  .addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)  .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 61, javax.swing.GroupLayout.PREFERRED\_SIZE)  .addGap(264, 264, 264))  );  layout.setVerticalGroup(  layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  .addGroup(layout.createSequentialGroup()  .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 34, javax.swing.GroupLayout.PREFERRED\_SIZE)  .addGap(66, 66, 66)  .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)  .addComponent(jLabel2)  .addComponent(jLabel5)  .addComponent(no, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)  .addComponent(alamat, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))  .addGap(20, 20, 20)  .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)  .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED\_SIZE, 22, javax.swing.GroupLayout.PREFERRED\_SIZE)  .addComponent(jLabel4)  .addComponent(nama, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)  .addComponent(hp, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))  .addGap(53, 53, 53)  .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)  .addComponent(tb\_simpan)  .addComponent(tb\_edit)  .addComponent(tb\_hapus))  .addGap(18, 18, 18)  .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)  .addComponent(cari, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)  .addComponent(tb\_cari))  .addGap(19, 19, 19)  .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED\_SIZE, 202, javax.swing.GroupLayout.PREFERRED\_SIZE)  .addContainerGap(38, Short.MAX\_VALUE))  );  pack();  }// </editor-fold>  private void namaActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  }  private void cariActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  }  private void tb\_simpanActionPerformed(java.awt.event.ActionEvent evt) {  try {  String sql = "insert into tbdata values('"  + no.getText() + "','"  + nama.getText() + "','"  + alamat.getText() + "','"  + hp.getText() + "')";  java.sql.Connection conn = (java.sql.Connection) projek.koneksi.koneksiDB();  java.sql.PreparedStatement pst = conn.prepareStatement(sql);  pst.execute();  JOptionPane.showMessageDialog(null, "Berhasil disimpan");  tampil\_data();  } catch (Exception e) {  JOptionPane.showMessageDialog(null, "Gagal disimpan");  System.out.println(e.getMessage());  }  }  private void tb\_cariActionPerformed(java.awt.event.ActionEvent evt) {  try {  java.sql.Connection conn = (java.sql.Connection) projek.koneksi.koneksiDB();  String sql = "Select \* from tbdata where no='" + cari.getText() + "' or nama='" + cari.getText() + "'";  java.sql.PreparedStatement pst = conn.prepareStatement(sql);  ResultSet rs = pst.executeQuery(sql);  if (rs.next()) {  no.setText(rs.getString(1));  nama.setText(rs.getString(2));  alamat.setText(rs.getString(3));  hp.setText(rs.getString(4));  JOptionPane.showMessageDialog(null, "Data tbdata : "  + cari.getText() + " ditemukan, Apakah Anda Ingin Mengedit atau Menghapus?");  }  } catch (Exception e) {  }  cari.requestFocus();  }  // Variables declaration - do not modify  private javax.swing.JTextField alamat;  private javax.swing.JTextField cari;  private javax.swing.JTextField hp;  private javax.swing.JLabel jLabel1;  private javax.swing.JLabel jLabel2;  private javax.swing.JLabel jLabel3;  private javax.swing.JLabel jLabel4;  private javax.swing.JLabel jLabel5;  private javax.swing.JScrollPane jScrollPane1;  private javax.swing.JScrollPane jScrollPane2;  private javax.swing.JTable jTable1;  private javax.swing.JTable jTable2;  private javax.swing.JTextField nama;  private javax.swing.JTextField no;  private javax.swing.JButton tb\_cari;  private javax.swing.JButton tb\_edit;  private javax.swing.JButton tb\_hapus;  private javax.swing.JButton tb\_simpan;  // End of variables declaration  } |

Program diatas menjelaskan komponen apa saja yang kita gunakan dalam mendesain GUI untuk class APK.java dan tempat memasukan fungsi untuk membuat setiap komponen komponen bisa responsive.

Setelah diisikan program diatas kita bisa langsung run di class APK.java di sourcenya

Berikut tampilan hasil runningnya



Dapat terlihat bahwa hasil runnya akan sesuai dengan apa yang kita desain. Program yang kita masukan tadi di APK.java akan membuat setiap komponen, sekarang saya coba masukan random data ke tabel tersebut apakah terhubung ke phpMyAdmin ?

A screenshot of a computer

Description automatically generated

Berikut adalah data yang saya masukan

A screenshot of a computer

Description automatically generated

Terlihat masuk di tampilan dari yang dikerjakan oleh program resultset

Berikut adalah database phpmyadminya

A screenshot of a computer

Description automatically generated

Terlihat data “UCUP” sudah tersimpan di database phpMyAdmin.

Sekian guided kali ini… terimakasih!!