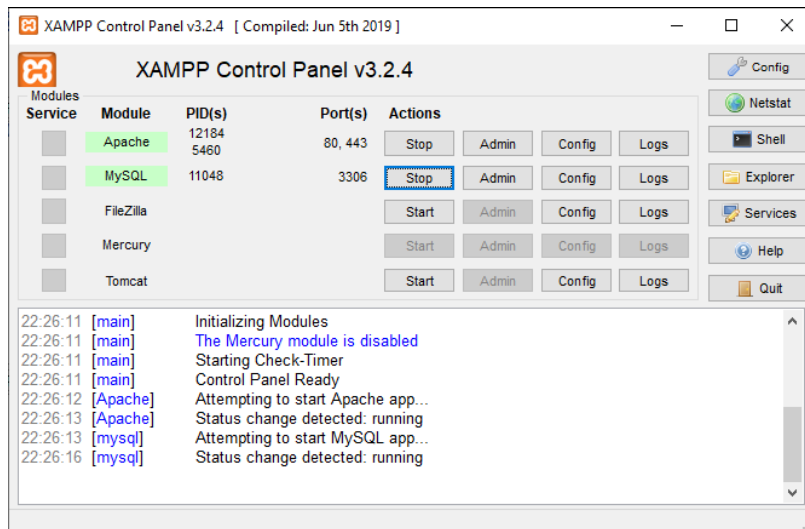
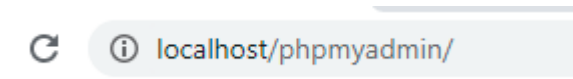


## Form Pegawai (Pertemuan ke 9)

- Jalankan XAMPP terlebih dahulu



- Buka browser dan jalankan phpmyadmin

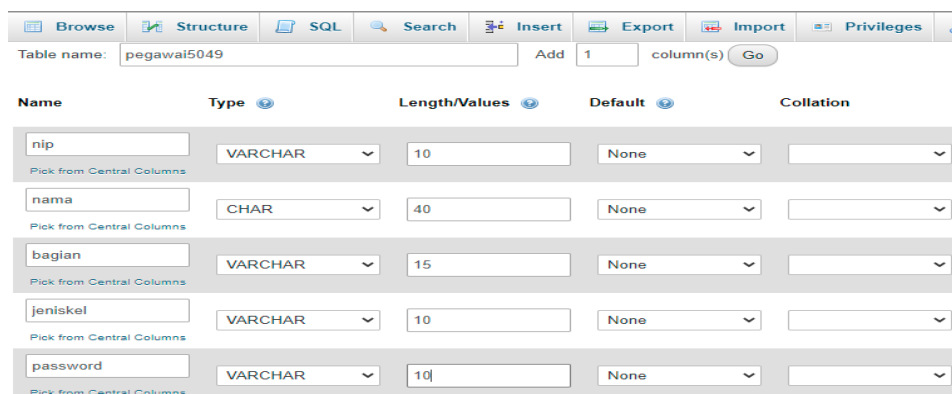


- Buat file tabel pegawai :

Buat Tabel : pegawaiXXXX (sesuai nim, misal : **pegawai5049**)

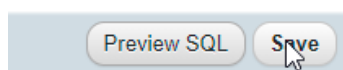
Berikan nilai untuk Default -> NULL

Name	Type	Length
nip	varchar	10
nama	char	50
bagian	varchar	15
jeniskel	varchar	10
password	varchar	10



Name	Type	Length/Values	Default	Collation
nip	VARCHAR	10	None	
nama	CHAR	40	None	
bagian	VARCHAR	15	None	
jeniskel	VARCHAR	10	None	
password	VARCHAR	10	None	

Selanjutnya pilih save



Struktur tabel yang terbentuk sebagai berikut :

[Browse](#)
[Structure](#)
[SQL](#)
[Search](#)
[Insert](#)
[Export](#)
[Import](#)
[Privileges](#)

[Table structure](#)
[Relation view](#)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	nip	varchar(10)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a>
<input type="checkbox"/> 2	nama	char(40)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a>
<input type="checkbox"/> 3	bagian	varchar(15)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a>
<input type="checkbox"/> 4	jeniskel	varchar(10)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a>
<input type="checkbox"/> 5	password	varchar(10)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a>

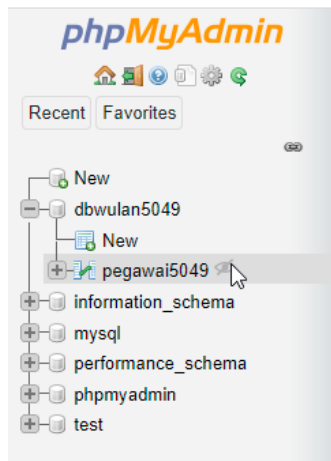
☐ Check all    With selected: [Browse](#) [Change](#) [Drop](#) [Primary](#) [Unique](#) [Index](#)

[Add to central columns](#) [Remove from central columns](#)

---

[Print](#) [Propose table structure](#) [Track table](#) [Move columns](#) [Normalize](#)

[Add](#)  column(s)  [Go](#)



[Browse](#)
[Structure](#)
[SQL](#)
[Search](#)
[Insert](#)
[Export](#)

[Table structure](#)
[Relation view](#)

#	Name	Type	Collation	Attributes	Null	Default	Comments
<input type="checkbox"/> 1	nip	varchar(10)	utf8mb4_general_ci		No	None	
<input type="checkbox"/> 2	nama	char(40)	utf8mb4_general_ci		No	None	
<input type="checkbox"/> 3	bagian	varchar(15)	utf8mb4_general_ci		No	None	
<input type="checkbox"/> 4	jeniskel	varchar(10)	utf8mb4_general_ci		No	None	
<input type="checkbox"/> 5	password	varchar(10)	utf8mb4_general_ci		No	None	

☐ Check all    With selected: [Browse](#) [Change](#) [Drop](#)

[Add to central columns](#) [Remove from central columns](#)

Lakukan pengisian data sebagai berikut :

Column	Type	Function	Null	Value
nip	varchar(10)			11111
nama	char(40)			Awalina Damayanti
bagian	varchar(15)			Pemasaran
jeniskel	varchar(10)			Perempuan
password	varchar(10)			11111

**Go**

- Output program yang harus dibuat :

Data Pegawai

NIP

Nama

Bagian

Jns Kelamin

Password

Nip	Nama	Bagian
11111	Awalina Damayanti	Pemasaran
22222	Mulyani Sari	Pemasaran
44444	Guna Wijaya	Produksi
12345	Lani Damayanti	Pemasaran
12346	Sri Rejeki Wijayanti	Administrasi

Add

Cari

Koreksi

Hapus

Selesai

Contoh program sebagai berikut :

Data Pegawai

NIP

Nama

Bagian

Nip	Nama	Bagian
11111	Awalina Damayanti	Pemasaran
22222	Mulyani Sari	Pemasaran
44444	Guna Wijaya	Produksi
12345	Lani Damayanti	Pemasaran
12346	Sri Rejeki Wijayanti	Administrasi

Add

Cari

Koreksi

Hapus

Selesai

### Kode program :

```
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import javax.swing.*;
import java.sql.*;

import javax.swing.table.DefaultTableModel;

public class FPegawai05049 extends JFrame implements ActionListener
{
    private JPanel      pPgw      = new JPanel();
    private JLabel      lNip       = new JLabel ("N I P      "),
                    lNama        = new JLabel ("Nama      "),
                    lBagian      = new JLabel ("Bagian     "),
                    lcreate      = new JLabel ("nama");

    private JTextField  fNip       = new JTextField (),
                    fNama        = new JTextField (),
                    fBagian      = new JTextField ();

    private JButton      btnAdd     = new JButton (),
                    btnCari      = new JButton (),
                    btnKoreksi    = new JButton (),
                    btnHapus     = new JButton (),
                    btnSelesai    = new JButton ();

    //tambah
    private ResultSet rs;
    String[] header = {"Nip","Nama","Bagian"};

    DefaultTableModel tabModel;
    JTable tabel = new JTable();
    JScrollPane skrTabel = new JScrollPane();

    public FPegawai05049() {
        setPreferredSize(new Dimension(570,410));
        setTitle("Data Pegawai");
        //setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JDesktopPane pPgw = new JDesktopPane();

        /* Mengatur letak objek Label Di container*/
        lNip.setBounds (20, 20, 100, 25);
        lNama.setBounds (20, 55, 100, 25);
        lBagian.setBounds (20, 90, 100, 25);
        lcreate.setBounds (230, 225, 200, 25);
        lcreate.setText("create by nama dan nim Anda");

        /* Mengatur letak objek Text Di Container */
        fNip.setBounds (115, 20, 100, 25);
        fNama.setBounds (115, 55, 205, 25);
        fBagian.setBounds (115, 90, 92, 25);

        fNip.setToolTipText("Isi Nip dengan Angka !");

        /* Mengatur letak objek Button di Container */
        btnAdd.setBounds (20, 320, 85, 25);
        btnAdd.setLabel("Add");
        btnAdd.setBackground(Color.green);
        btnAdd.setForeground(Color.getHSBColor(250,0,255));
        btnCari.setBounds (120, 320, 85, 25);
        btnCari.setLabel("Cari");
```

```

        btnKoreksi.setBounds (220, 320, 85, 25);
        btnKoreksi.setLabel("Koreksi");
        btnHapus.setBounds (320, 320, 85, 25);
        btnHapus.setLabel("Hapus");
        btnSelesai.setBounds (435, 320, 85, 25);
        btnSelesai.setLabel("Selesai");
        btnSelesai.setToolTipText("Mengakhiri Program");

        //membuat tabel.....
        tabModel = new DefaultTableModel(null,header);
        tabel.setModel(tabModel);
        tabel.setBackground(Color.orange);
        skrTabel.getViewport().add(tabel);
        tabel.setEnabled(true);
        skrTabel.setBounds(20,130,500,170);

        /* Objek Button di Non Aktifkan dan di aktifkan */
        btnAdd.setEnabled(true);
        btnSelesai.setEnabled(true);

        // Mengatur objek untuk dapat berinteraksi
        btnAdd.addActionListener (this);
        btnCari.addActionListener (this);
        btnKoreksi.addActionListener (this);
        btnHapus.addActionListener (this);
        btnSelesai.addActionListener (this);

        // Meletakkan seluruh kontrol pada objek panel */
        pPgw.add (skrTabel);
        pPgw.add (lNip);
        pPgw.add (fNip);
        pPgw.add (lNama);
        pPgw.add (fNama);
        pPgw.add (lBagian);
        pPgw.add (fBagian);
        pPgw.add (lcreate);

        pPgw.add (btnAdd);
        pPgw.add (btnCari);
        pPgw.add (btnKoreksi);
        pPgw.add (btnHapus);
        pPgw.add (btnSelesai);

        /* Menambahkan objek panel (pPgr) ke container frame */
        getContentPane().add (pPgw).setBackground(Color.getHSBColor(100,150,75));

        /* Menampilkan frame ke layar monitor */
        pack();
        setLocationRelativeTo(null);
        tampiltabel();
        Kosong();
        settombol(1, 1, 0, 0, 1);
    }

    public static void main(String[] args) {
        new FPegawai05049().setVisible(true);
    }

    /* Fungsi jika user melakukan action penekanan tombol Button */
    public void actionPerformed (ActionEvent ae) {
        Object obj = ae.getSource();

        if (obj == btnAdd)

```

```

{
    if (btnAdd.getText()=="Simpan")
    {
        String xnip = fNip.getText();
        if (xnip.length() != 0) Add();
        else
            JOptionPane.showMessageDialog(this, "Nip kosong !");

        btnAdd.setText("Add");
        Kosong();
        tampiltabel();
        settombol(1, 1, 0, 0, 1);
    }
    else
    {
        Kosong();
        settombol(1, 0, 0, 0, 1);
        btnAdd.setText("Simpan");
        btnAdd.setForeground(Color.getHSBColor(100,10,0));
        btnAdd.setBackground(Color.getHSBColor(200,100,100));
    }
}

if (obj == btnCari)
{
    String xcari = JOptionPane.showInputDialog(this,"Masukkan Nip !");
    if (xcari != null)
    {
        fNip.setText(xcari);
        int xx = Cari();
        if (xx == 1) settombol(1, 1, 1, 1, 1);
    }
    else
        JOptionPane.showMessageDialog(this, "Nip kosong !");
}

if (obj == btnKoreksi)
{
    if (btnKoreksi.getText()=="Simpan")
    {
        if (fNip.getText() != "")
            Koreksi();
        btnKoreksi.setText("Koreksi");
        tampiltabel();
        Kosong();
        settombol(1, 1, 0, 0, 1);
    }
    else
    {
        btnKoreksi.setText("Simpan");
        settombol(0, 0, 1, 0, 1);
        fNama.requestFocus();
    }
}

if (obj == btnHapus)
{
    int stt = JOptionPane.showConfirmDialog(this,"Yakin dihapus ?");
    System.out.print(stt);

    if (stt == 0)
    {
        Hapus();
        tampiltabel();
        Kosong();
    }
}

if (obj == btnSelesai)

```

```

        {
            //System.exit(0);
            this.dispose();
        }
    }

    // Fungsi untuk menambahkan data ke tabel barang
    void Add()
    {
        try
        {
            KoneksiDBMS CnPenjualan = new KoneksiDBMS();
            Connection con = CnPenjualan.BukaCn();
            Statement stat = con.createStatement();

String strSQLcr = "SELECT * FROM pegawai5049 WHERE nip='" + fNip.getText () + "'";
            ResultSet rs = stat.executeQuery(strSQLcr);

            if(rs.next())
            {
                JOptionPane.showMessageDialog(this, "Nip Sudah Ada !");
                return;
            }

String strSQL = "insert into pegawai5049(nip, nama, bagian) values " +
                " ('"+fNip.getText() + "', '"+fNama.getText()+"' " +
                " , '"+fBagian.getText()+"')";
            int stsproses = stat.executeUpdate(strSQL);

            if (stsproses == 1)
                JOptionPane.showMessageDialog(this, "Sukses Di Tambahkan!!!");
            con.close();
        }
        catch(SQLException e)
        {
            JOptionPane.showMessageDialog(this, "Penambahan Gagal!!!");
            System.err.println("Kesalahan perintah SQL : " + e.getMessage());
        }
    }

    // Fungsi untuk mencari Kode_Barang ke tabel barang
    int Cari(){
        int x = 0;
        try
        {
            KoneksiDBMS CnPenjualan = new KoneksiDBMS();
            Connection con = CnPenjualan.BukaCn();
            Statement stat = con.createStatement();

String strSQL = "SELECT * FROM pegawai5049 WHERE nip='" + fNip.getText () + "'";
            ResultSet rs = stat.executeQuery(strSQL);

            if(rs.next())
            {
                // Jika Nip Ditemukan Di Tabel Pegawai
                fNip.setText (rs.getString ("nip"));
                fNama.setText (rs.getString ("nama"));
                fBagian.setText (rs.getString ("bagian"));
                x = 1;
            }
            else
            {
                // Jika Nip tidak ditemukan
                JOptionPane.showMessageDialog(this, "Data tidak ditemukan !");
                fNip.requestFocus ();
            }
        }
    }

```

```

        }
        rs.close();
        con.close();
    }
    catch(SQLException e)
    {
    }
    return x;
}

// Fungsi untuk menambahkan data ke tabel barang
void Koreksi()
{
    try
    {
        KoneksiDBMS CnPenjualan = new KoneksiDBMS();
        Connection con = CnPenjualan.BukaCn();
        Statement stat = con.createStatement();

        String strsql = "update pegawai5049 set nama='"+fNama.getText()+
            "',bagian='"+ fBagian.getText()+
            "' where nip='"+fNip.getText()+"' ";
        int stsproses = stat.executeUpdate(strsql);

        if (stsproses == 1)
            JOptionPane.showMessageDialog(this, "Sukses DiEdit!");
        con.close();
    }
    catch(SQLException e)
    {
        JOptionPane.showMessageDialog(this, "Koreksi Gagal !");
        System.err.println("Kesalahan perintah SQL : " + e.getMessage());
    }
}

// Fungsi untuk menambahkan data ke tabel barang
void Hapus()
{
    try
    {
        KoneksiDBMS CnPenjualan = new KoneksiDBMS();
        Connection con = CnPenjualan.BukaCn();
        Statement stat = con.createStatement();

        String strsql = "delete from pegawai5049 where nip='"+fNip.getText()+"' ";
        int stsproses = stat.executeUpdate(strsql);

        if (stsproses == 1)
            JOptionPane.showMessageDialog(this, "Data Terhapus !");
        con.close();
    }
    catch(SQLException e)
    {
        JOptionPane.showMessageDialog(this, "Penghapusan Gagal!!!");
        System.err.println("Kesalahan perintah SQL : " + e.getMessage());
    }
}

//tambah
void tampiltabel()
{
    try {
        bersihtabel();
        KoneksiDBMS CnPenjualan = new KoneksiDBMS();

```



```

        Connection con = CnPenjualan.BukaCn();
        Statement stat = con.createStatement();
        String strsql = "select * from pegawai5049";
        rs = stat.executeQuery(strsql);
        ResultSetMetaData meta = rs.getMetaData();

        rs.beforeFirst();
        while(rs.next()) {
            String Nip = rs.getString("nip");
            String Nama = rs.getString("nama");
            String Bagian = rs.getString("bagian");
            String[] data = {Nip, Nama, Bagian};
            tabModel.addRow(data);
        }
        stat.close();
        rs.close();
        con.close();

    } catch (SQLException se) {
        System.err.println("Kesalahan perintah SQL : " + se.getMessage());
    }
}

void bersihtabel()
{
    int brs = tabModel.getRowCount();
    for(int i=0;i<brs;i++)
    {
        tabModel.removeRow(0);
    }
}

// Fungsi untuk mengkosongkan Objek masukan
void Kosong ()
{
    fNip.setText("");
    fNama.setText("");
    fBagian.setText("");
    fNip.requestFocus();
}

void settombol(int a, int b, int c, int d, int e)
{
    btnAdd.setEnabled(a>=1 ? true: false);
    btnCari.setEnabled(b>=1 ? true: false);
    btnKoreksi.setEnabled(c>=1 ? true: false);
    btnHapus.setEnabled(d>=1 ? true: false);
    btnSelesai.setEnabled(e>=1 ? true: false);
}
}

```