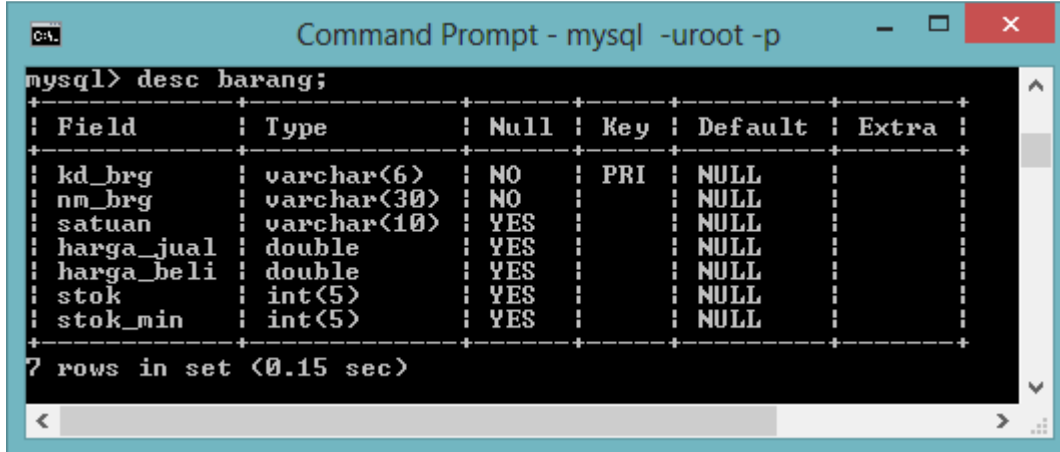


PRAKTIKUM #11

Buat Database Penjualan dan Tabel Barang berikut :



The screenshot shows a MySQL Command Prompt window with the title "Command Prompt - mysql -uroot -p". The command entered is "mysql> desc barang;". The output is a table with 7 rows and 6 columns: Field, Type, Null, Key, Default, and Extra. The fields are kd_brg, nm_brg, satuan, harga_jual, harga_beli, stok, and stok_min. The types are varchar(6), varchar(30), varchar(10), double, double, int(5), and int(5) respectively. The null values are NO, NO, YES, YES, YES, YES, and YES. The key values are PRI, and the default values are NULL, NULL, NULL, NULL, NULL, NULL, and NULL. The extra values are empty for all fields.

Field	Type	Null	Key	Default	Extra
kd_brg	varchar(6)	NO	PRI	NULL	
nm_brg	varchar(30)	NO		NULL	
satuan	varchar(10)	YES		NULL	
harga_jual	double	YES		NULL	
harga_beli	double	YES		NULL	
stok	int(5)	YES		NULL	
stok_min	int(5)	YES		NULL	

7 rows in set (0.15 sec)

Koneksi Java – Database MySQL :

```
package penjualan;
import java.sql.*;

public class KoneksiMysql {
    String url, usr, pwd, dbn;
    public KoneksiMysql (String dbn) {
        this.url = "jdbc:mysql://localhost/" + dbn;
        this.usr = "root";
        this.pwd = "";
    }

    public KoneksiMysql (String host, String user, String pass,
        String dbn) {
        this.url = "jdbc:mysql://" + host + "/" + dbn;
        this.usr = user;
        this.pwd = pass;
    }

    public Connection getConnection() {
        Connection con = null;
        try {
            Class.forName("com.mysql.jdbc.Driver");
            con = DriverManager.getConnection(this.url, this.usr,
this.pwd);
        } catch (ClassNotFoundException e) {
            System.out.println ("Error #1 : " + e.getMessage());
        }
    }
}
```

```
        System.exit(0);
    } catch (SQLException e) {
        System.out.println ("Error #2 : " + e.getMessage());
        System.exit(0);
    }
    return con;
}

/*
public static void main (String args[]) {
    KoneksiMysql kon = new KoneksiMysql ("barang");
    Connection c = kon.getConnection();
}

*/
}
```

Form Master Barang

The screenshot shows a Java Swing window titled "Form Data Barang". The window contains a form for entering item data and a table showing existing items. Callouts identify UI components: jTextField, JComboBox, jTable, and jButton.

Data Barang

Kode Barang: 00001

Nama Barang: Durian Bang...

Satuan: Buah

Harga: 10000.0

Stok: 10

Stok Minimal: 5

Kode	Nama Barang	Satuan	Harga	Stok	Stok Min
00001	Durian Bang...	Buah	10000.0	10	5
00002	Buku	Buah	2000.0	20	3

Tambah Simpan Koreksi Hapus Batal Keluar

Setelah membuat table barang, buat desain form seperti gambar di atas dengan menggunakan NetBean. Sesuaikan semua nama obyek dengan kebutuhan anda.

Alur program :

1. frmbarang created
2. panggil konstruktor frmbarang sekaligus open database
3. isian jtextfiled off
4. table barang diisi data dari table barang yang ada di server MySQL
5. button tambah, koreksi, hapus dan keluar dalam keadaan on sedang simpan dan batal off.
6. Jika ditekan tombol tambah, maka actionPerformed pada button tambah dijalankan.
7. Jika ditekan tombol simpan, maka actionPerformed pada button simpan dijalankan.
8. Jika ditekan tombol koreksi, maka actionPerformed pada button koreksi dijalankan.
9. Jika ditekan tombol hapus, maka actionPerformed pada button hapus dijalankan.
10. Jika ditekan tombol keluar, maka actionPerformed pada button keluar dijalankan.

Code dan Method-method yang dibuat

Tambahkan code di bawah

public class frmBarang extends javax.swing.JFrame {

```
Connection Con;
ResultSet RsBrg;
Statement stm;

Boolean ada = false;
String sSatuan;
Boolean edit=false;

private Object[][] dataTable = null;
private String[] header =
{"Kode", "Nama Barang", "Satuan", "Harga", "Stok", "Stok Min"};
```

Tambahkan code dibawah knstruktur

```
public frmBarang() {
    initComponents();
    open_db();
    baca_data();
    aktif(false);
    setTombol(true);
}
```

Buat Method-method berikut :

//method untuk memindahkan data dr table ke form

```
private void setField()
{
    int row=tblBrg.getSelectedRow();
    txtKode.setText((String)tblBrg.getValueAt(row,0));
    txtNama.setText((String)tblBrg.getValueAt(row,1));
    cmbSatuan.setSelectedItem((String)tblBrg.getValueAt(row,2));
}
```

```
String harga = Double.toString(((Double)tblBrg.getValueAt(row,3));
txtHarga.setText(harga);
String stok=Integer.toString(((Integer)tblBrg.getValueAt(row,4));
txtStok.setText(stok);
String stokmin=Integer.toString(((Integer)tblBrg.getValueAt(row,5));
txtStokMin.setText(stokmin);
}

//method membuka database server, user, pass, database disesuaikan
private void open_db()
{
    try{
        KoneksiMysql kon = new KoneksiMysql
("localhost","root","root","penjualan");
        Con = kon.getConnection();
        //System.out.println("Berhasil ");
    }catch (Exception e) {
        System.out.println("Error : "+e);
    }
}

//method baca data dari Mysql dimasukkan ke table pada form
private void baca_data()
{
    try{
        stm = Con.createStatement();
        RsBrg = stm.executeQuery("select * from barang");

        ResultSetMetaData meta = RsBrg.getMetaData();
        int col = meta.getColumnCount();
        int baris = 0;
        while(RsBrg.next()) {
            baris = RsBrg.getRow();
        }

        dataTable = new Object[baris][col];
        int x = 0;
        RsBrg.beforeFirst();
        while(RsBrg.next()) {
            dataTable[x][0] = RsBrg.getString("kd_brg");
            dataTable[x][1] = RsBrg.getString("nm_brg");
            dataTable[x][2] = RsBrg.getString("satuan");
            dataTable[x][3] = RsBrg.getDouble("harga");
            dataTable[x][4] = RsBrg.getInt("stok");
            dataTable[x][5] = RsBrg.getInt("stok_min");
            x++;
        }
        tblBrg.setModel(new DefaultTableModel(dataTable,header));
    }
    catch(SQLException e)
    {
        JOptionPane.showMessageDialog(null, e);
    }
}

//untuk mengkosongkan isian data
private void kosong()
{

```

```
txtKode.setText("");
txtNama.setText("");
txtHarga.setText("");
txtStok.setText("");
txtStokMin.setText("");
}

//mengset aktif tidak isian data
private void aktif(boolean x)
{
    txtKode.setEditable(x);
    txtNama.setEditable(x);
    //cmbSatuan.setEditable(x);
    cmbSatuan.setEnabled(x);
    txtHarga.setEditable(x);
    txtStok.setEditable(x);
    txtStokMin.setEditable(x);
}
//mengset tombol on/off
private void setTombol(boolean t)
{
    cmdTambah.setEnabled(t);
    cmdKoreksi.setEnabled(t);
    cmdHapus.setEnabled(t);
    cmdSimpan.setEnabled(!t);
    cmdBatal.setEnabled(!t);
    cmdKeluar.setEnabled(t);
}
}
```

Event pada masing-masing obyek yang perlu disesuaikan :

```
private void cmdKeluarActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}

private void cmdSimpanMouseClicked(java.awt.event.MouseEvent evt) {
    String tKode=txtKode.getText();
    String tNama=txtNama.getText();

    double hrg=Double.parseDouble(txtHarga.getText());
    int stk=Integer.parseInt(txtStok.getText());
    int stkMin=Integer.parseInt(txtStokMin.getText());
    try{
        if (edit==true)
        {
            stm.executeUpdate("update barang set
nm_brg='"+tNama+"', satuan='"+sSatuan+"', harga="+hrg+", stok="+stk+", stok
_min="+stkMin+" where kd_brg='"+ tKode + "'");
        }else
        {
            stm.executeUpdate("INSERT into barang
VALUES ('"+tKode+"', '"+tNama+"', '"+sSatuan+"', "+hrg+", "+stk+", "+stkMin+"
)");
        }
        tblBrg.setModel(new DefaultTableModel(dataTable,header));
    }
```

```
        baca_data();
        aktif(false);
        setTombol(true);
    }catch(SQLException e) {
        JOptionPane.showMessageDialog(null, e);
    }
}

private void cmbSatuanActionPerformed(java.awt.event.ActionEvent evt) {
    JComboBox cSatuan = (javax.swing.JComboBox)evt.getSource();

    //Membaca Item Yang Terpilih - > String

    sSatuan = (String)cSatuan.getSelectedItem();
}

private void cmdTambahActionPerformed(java.awt.event.ActionEvent evt) {
    aktif(true);
    setTombol(false);
    kosong();
}

private void cmdBatalActionPerformed(java.awt.event.ActionEvent evt) {
    aktif(false);
    setTombol(true);
}

private void tblBrgMouseClicked(java.awt.event.MouseEvent evt) {
    setField();
}

private void cmdKoreksiActionPerformed(java.awt.event.ActionEvent evt)
{
    edit=true;
    aktif(true);
    setTombol(false);
    txtKode.setEditable(false);
}

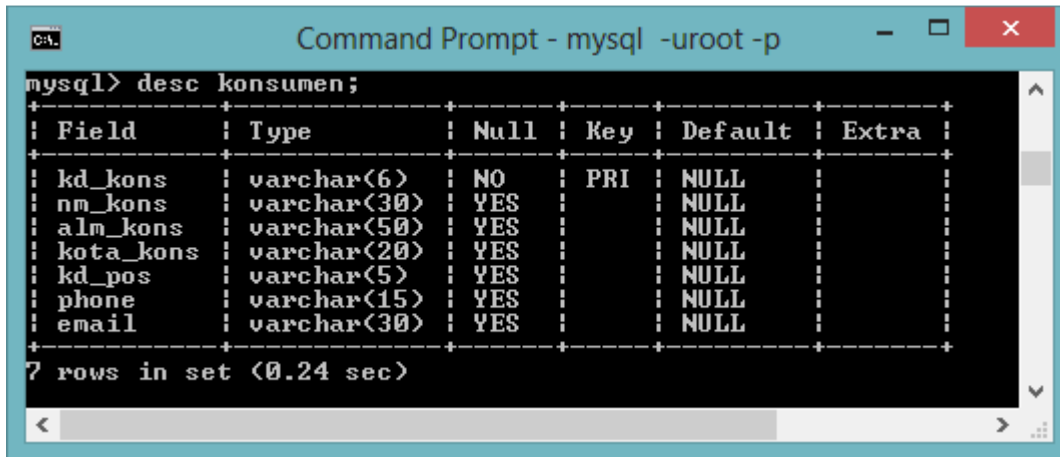
private void cmdHapusActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        String sql="delete from barang where kd_brg='" + txtKode.getText()
+ "'";
        stm.executeUpdate(sql);
        baca_data();
    }
    catch(SQLException e)
    {
        JOptionPane.showMessageDialog(null, e);
    }
}
```

Latihan 1 :

Tambahkan pada desain form barang field Harga Beli sesuai dengan Tabel Barang, update di beberapa code untuk penyesuaian dengan field Harga Beli.

Latihan 2:

a. Buat Tabel Konsumen berikut :

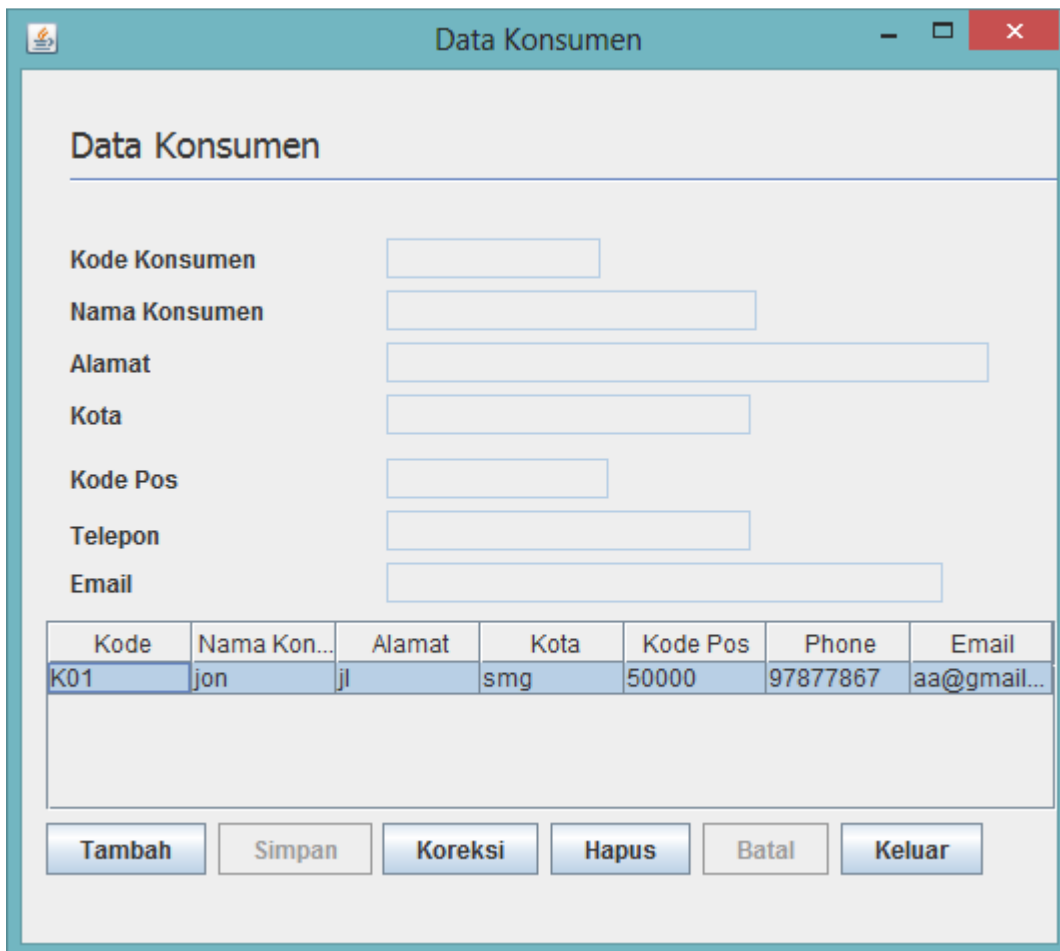


```
mysql> desc konsumen;
```

Field	Type	Null	Key	Default	Extra
kd_kons	varchar(6)	NO	PRI	NULL	
nm_kons	varchar(30)	YES		NULL	
alm_kons	varchar(50)	YES		NULL	
kota_kons	varchar(20)	YES		NULL	
kd_pos	varchar(5)	YES		NULL	
phone	varchar(15)	YES		NULL	
email	varchar(30)	YES		NULL	

7 rows in set (0.24 sec)

b. Buat form konsumen



Data Konsumen

Kode Konsumen

Nama Konsumen

Alamat

Kota

Kode Pos

Telepon

Email

Kode	Nama Kon...	Alamat	Kota	Kode Pos	Phone	Email
K01	jon	jl	smg	50000	97877867	aa@gmail...

Tambah Simpan Koreksi Hapus Batal Keluar

Latihan 3:

Buat form dari table master sesuai tugas akhir masing-masing.