

**LAPORAN TUGAS BESAR PBO**

**APLIKASI DATA PENJUALAN**



**DISUSUN OLEH :**

**MUHAMMAD SYAFI'I HIDAYATULLAH**

**NIM : TI19220020**

**TEKNIK INFORMATIKA**

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER LOMBOK**

**2023/2024**

## KATA PENGANTAR

Sistem informasi yang baik akan menyimpan dengan baik setiap proses transaksi yang terjadi ke dalam database. Hal ini menjadi hal yang krusial, karena ini menjadi titik vital dari perusahaan, karena setiap transaksi perusahaan tercatat dengan baik. Untuk menunjang itu diperlukannya penggunaan database yang baik.

Dalam proses database saat ini sering menggunakan database relational atau RDBMS. Proses database ini menggunakan query untuk memanggil data dari database. Sebuah sistem informasi akan sering melakukan proses query ke dalam database, hal ini untuk mendapatkan data yang sesuai dan dalam keadaan real. Untuk itu diperlukan sebuah proses query database yang baik.

Proses CRUD ini akan menjadi sangat penting karena, akan memberatkan system jika proses berlangsung secara serampangan. Salah satu prosesnya adalah teknik normalisasi dari database, teknik ini sudah sering digunakan dan menjadi bagian penting. Salah satu teknik penting di dalam proses CRUD ke database, kita bisa menggunakan teknik ORM atau *Object Relational Mapping*, terlebih teknik ini akan sangat bekerja optimal dibawah pengaruh *Object Oriented*.

# DAFTAR ISI

KATA PENGANTAR.....	2
DAFTAR ISI .....	3
BAB 1 .....	4
PENDAHULUAN .....	4
1.1 Latar Belakang Masalah .....	4
BAB II.....	5
2.1 Pengertian .....	5
2.2 Cara Membuat CRUD Dengan PHP Dan MySQL .....	5
1. Membuat Tabel Database.....	5
2. Membuat Project Baru pada NetBeans java .....	6
3. Membuat Form Baru JFrame (produk.Java).....	6
4. Membuat Class Koneksi Database MySQL .....	7
BAB III .....	12
A. KESIMPULAN.....	12
B. DAFTAR PUSAKA .....	12

# **BAB 1**

## **PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

Sistem informasi yang baik akan menyimpan dengan baik setiap proses transaksi yang terjadi ke dalam database. Hal ini menjadi hal yang krusial, karena ini menjadi titik vital dari perusahaan, karena setiap transaksi perusahaan tercatat dengan baik. Untuk menunjang itu diperlukannya penggunaan database yang baik. Dalam proses database saat ini sering menggunakan database relational atau RDBMS. Proses database ini menggunakan query untuk memanggil data dari database. Sebuah sistem informasi akan sering melakukan proses query ke dalam database, hal ini untuk mendapatkan data yang sesuai dan dalam keadaan real.

CRUD adalah singkatan dari Create, Read, Update, dan Delete. Proses ini sangat berkaitan dengan pengambilan atau transaksi data dari atau ke database. Hal ini menjadi krusial apabila berhubungan dengan sistem informasi perusahaan karena data yang diproses biasanya merupakan data transaksi. Bagi para Developer, operasi CRUD biasanya menjadi pillar untuk mempelajari proses pengelolaan data menggunakan JAVA dan tentu saja MySQL database. Nah CRUD ini bisa diibaratkan sebagai jalur atau koneksi yang menghubungkan antara bahasa pemrograman JAVA dengan MySQL.

### **1.2 Rumusan Masalah**

Apa pengertian masing-masing dari CRUD tersebut?

### **1.3 Tujuan Penelitian**

Untuk mengetahui apa pengertian dari masing-masing CRUD tersebut.

## **BAB II**

### **PEMBAHASAN**

#### **2.1 Pengertian**

- Create (C) merupakan proses pembuatan data baru. Proses ini biasanya dilakukan ketika Anda mendaftar pada sebuah halaman website. Data yang Anda masukkan akan disimpan di dalam database menggunakan operasi Create.
- Read (R) merupakan proses pengambilan data dari database. Proses ini biasanya terjadi ketika Anda ingin melakukan proses login di halaman website tertentu. Saat Anda klik “Login” maka website akan menjalankan proses “Read” untuk melakukan verifikasi akun yang Anda gunakan. Contoh operasi yang lebih sederhana yaitu ketika Anda mengakses sebuah halaman website, saat itu pula proses Read berjalan. Hasil dari operasi ini adalah tampilan halaman website yang Anda lihat di browser.
- Update (U) adalah proses mengubah data yang berada di dalam database. Contoh proses ini ketika Anda mengubah profil di dalam akun sosial media. Saat Anda klik “Ubah”, website akan mengirimkan proses Update ke dalam database. Kemudian database meresponnya dengan mengubah data lama menjadi data baru yang Anda tambahkan melalui halaman profil.
- Delete (D) adalah proses untuk menghapus data yang ada di database. Proses ini mirip dengan Update, bedanya, proses Delete akan mengubah data yang ada di database menjadi ‘tidak ada’.

#### **2.2 Cara Membuat CRUD Dengan PHP Dan MySQL**

Developer aplikasi sering menggunakan JAVA dan MySQL sebagai komponen utamanya. Bahasa pemrograman JAVA memproses berbagai fungsi untuk menjalankan operasi CRUD. Sedangkan MySQL bertugas untuk menyimpan database dari aplikasi. MySQL sebenarnya tidak hanya bisa diakses menggunakan JAVA saja, bahasa pemrograman lain seperti Python, JavaScript, dan sejenisnya juga bisa untuk mengakses MySQL. Sedangkan JAVA juga bisa menggunakan server database lain seperti PostgreSQL, MongoDB, dan sejenisnya. Namun pada contoh ini kami menggunakan JAVA dan MySQL karena lebih sederhana dan mudah untuk dipahami.

##### **1. Membuat Tabel Database**

Supaya Anda bisa melakukan operasi CRUD tentu saja perlu sebuah data. Jadi hal pertama yang perlu Anda lakukan untuk membuat CRUD adalah menambahkan data ke dalam database di MySQL. Anda dapat menambahkan data pegawai menggunakan query di bawah ini.

Pertama buat database terlebih dahulu.



Server: 127.0.0.1 » Database: penjualan » Tabel: produk									
Jelajahi	Struktur	SQL	Cari	Tambahkan	Ekspor	Impor	Hak Akses	Operasi	Trigger
Struktur tabel	Tampilan hubungan								
#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/> 1	IDProduk	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
<input type="checkbox"/> 2	KodeProduk	varchar(32)	utf8mb4_general_ci	Ya		NULL			Ubah Hapus Lainnya
<input type="checkbox"/> 3	NamaProduk	varchar(255)	utf8mb4_general_ci	Ya		NULL			Ubah Hapus Lainnya
<input type="checkbox"/> 4	Satuan	varchar(255)	utf8mb4_general_ci	Ya		NULL			Ubah Hapus Lainnya
<input type="checkbox"/> 5	Harga	int(11)		Ya		NULL			Ubah Hapus Lainnya
<input type="checkbox"/> 6	Stok	int(11)		Ya		NULL			Ubah Hapus Lainnya
<input type="checkbox"/> 7	CreateDate	datetime		Ya		NULL			Ubah Hapus Lainnya

Query di atas merupakan SQL yang bertugas untuk menyiapkan database beserta TABEL yang berisi variabel idproduk, kodeproduk, namaproduk, satuan, harga dan stok.

Pastikan database berhasil terbuat dengan mengetikkan perintah “show tables;”.

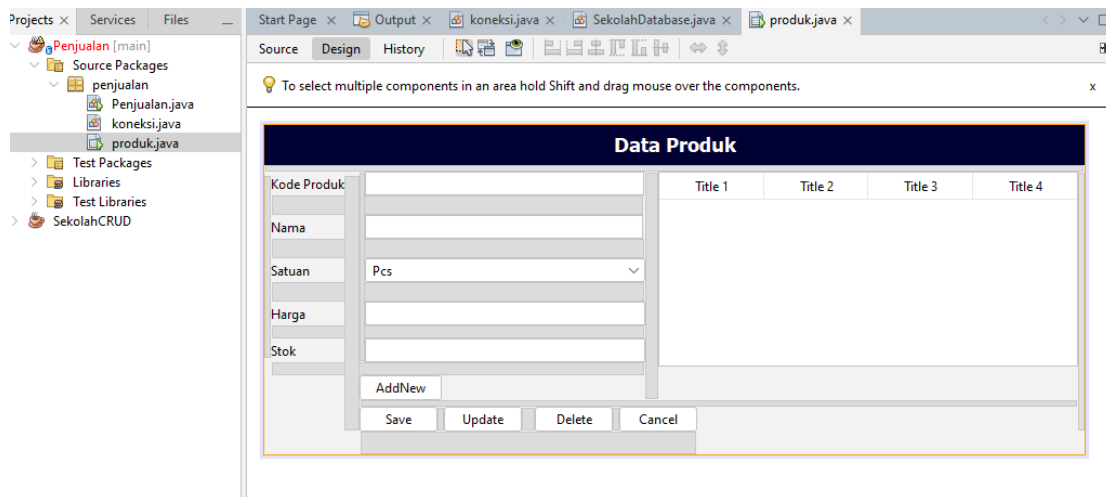
## 2. Membuat Project Baru pada NetBeans java

Buka aplikasi NetBeans dan buat project baru New Project > Java > Java Application > Next > dan buat nama project "Aplikasipenjualan" > finish . Untuk nama project bisa diganti sesuai kebutuhan.

Setelah project dibuat, secara default akan otomatis membuat Main Class dari project dengan nama sesuai nama project java kita, silahkan hapus terlebih dahulu (AplikasiCrud.java). kita akan membuat Form baru dengan JFrame dimana berfungsi untuk menampilkan seluruh data dan interface antara user dan database..

## 3. Membuat Form Baru JFrame (produk.Java)

Buatlah Form baru dengan cara klik kanan pada source package Aplikasipenjualan > New > JFrame > tuliskan nama form baru dengan nama "penjualan.java" dan > Finish. serta desainlah tampilan FrmUtama.java seperti gambar berikut :



Setelah semuanya beres, tinggalkan sejenak FormUtama.Java ini, karena kita harus membuat Class Koneksi yang akan membuat koneksi dari aplikasi ke Database MySQL nantinya

#### 4. Membuat Class Koneksi Database MySQL

Buatlah class baru dengan nama "koneksi.java". Sebelum memulai menuliskan code untuk koneksi java silahkan tambahkan library JDBCdriver ke dalam project dengan cara Klik kanan pada Libraries > ADD Jar / Folder dan pilih file MySQL JDBC

ika semua beres, langsung saja tuliskan code java untuk class koneksi.java berikut ini

Source Code Class Koneksi.Java:

```

    * and open the template in the editor.
    */
package penjualan;
import java.sql.*;
import javax.swing.JOptionPane;

/**
 *
 */
public class koneksi {
    private static Connection KoneksiDatabase;
    public static Connection koneksiDB() throws SQLException {
        try {
            String DB ="jdbc:mysql://localhost/penjualan";
            String user="root";
            String pass="";
            DriverManager.registerDriver(new com.mysql.jdbc.Driver());
            KoneksiDatabase = (Connection) DriverManager.getConnection(url:DB,user,password: pass);
        } catch (Exception e) {
            JOptionPane.showMessageDialog(parentComponent:null,message:"Tidak ada koneksi",title:"Error",
            messageType:JOptionPane.INFORMATION_MESSAGE);
            System.err.println("e.getMessage());
            System.exit(status: 0);
        }
        return KoneksiDatabase;
    }
}

```

Source Code Tombol New

Sebelum menuliskan code pada sub yang lain, silahkan import namespace yang dibutuhkan.

```

package penjualan;
import java.sql.*;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import java.util.Scanner;
import java.awt.*;
/**

```

#### Source Code Tombol New

```

    * Creates new form produk
    */
public produk() {
    initComponents();

    Dimension screenSize=Toolkit.getDefaultToolkit().getScreenSize();
    Dimension frameSize=this.getSize();
    if(frameSize.height > screenSize.height){
        frameSize.height=screenSize.height;
    }
    if(frameSize.width > screenSize.width){
        frameSize.width=screenSize.width;
    }
    this.setLocation((screenSize.width - frameSize.width) / 2,
        (screenSize.height = screenSize.height) / 20);

    //fungsi Tampil data dan set edit off dijalankan saat aplikasi di run
    tampilData();
    SetEditOff();
}

```

#### Source Code Tombol Save

// TODO add your handling code here:

```

String KodeProduk=jTextKodeProduk.getText();
String NamaProduk=jTextNama.getText();
String Satuan=jComboBoxSatuan.getSelectedItem().toString();
String Harga=jTextHarga.getText();
String Stok=jTextStok.getText();

java.util.Date tanggal = new java.util.Date();
java.text.SimpleDateFormat TanggalFormat= new java.text.SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
String CreateDate=TanggalFormat.format(tanggal);

```



```

if (KodeProduk.isEmpty() ) {
    JOptionPane.showMessageDialog(null,"Kode produk tidak boleh kosong");
    jTextKodeProduk.requestFocus();
}else if (NamaProduk.isEmpty()) {
    JOptionPane.showMessageDialog(null,"Nama produk tidak boleh kosong");
    jTextNama.requestFocus();
}else if (Satuan.isEmpty()) {
    JOptionPane.showMessageDialog(null,"Satuan produk tidak boleh kosong");
    jComboBoxSatuan.requestFocus();
}else if (Harga.isEmpty()) {
    JOptionPane.showMessageDialog(null,"Harga produk tidak boleh kosong");
    jTextHarga.requestFocus();
}else if (Stok.isEmpty()) {
    JOptionPane.showMessageDialog(null,"Stok produk tidak boleh kosong");
    jTextStok.requestFocus();
}
else if (!(new Scanner(Harga).hasNextInt())) {
    JOptionPane.showMessageDialog(null,"Harga produk harus angka");
    jTextHarga.requestFocus();
}else if (!(new Scanner(Stok).hasNextInt())) {
    JOptionPane.showMessageDialog(null,"Stok produk harus angka");
    jTextStok.requestFocus();
}else{
    try{
        Connection conn=(Connection)koneksi.koneksiDB();
        Statement stt=conn.createStatement();
        stt.executeUpdate("insert
produk(KodeProduk,NamaProduk,Satuan,Harga,Stok,CreateDate)" +
into
"VALUES('"+KodeProduk+"','"+NamaProduk+"','"+Satuan+"','"+Harga+"','"+Stok+"','"+Cre
ateDate+"')");
        BersihData();
        tampilData();
    }

```

```

        SetEditOff();

        JOptionPane.showMessageDialog(this,"Data
disimpan","Success",JOptionPane.INFORMATION_MESSAGE);

        } catch(SQLException e){

        JOptionPane.showMessageDialog(this,"Simpan
gagal\n"+e.getMessage(),"Error",JOptionPane.ERROR_MESSAGE);

        }

```

berhasil  
  
data

## Source Code Tombol Delete

```

private void jButtonDeleteActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String KodeProduk=jTextKodeProduk.getText();

    if (KodeProduk.isEmpty() ) {
        JOptionPane.showMessageDialog(parentComponent:null,message:"Kode produk tidak boleh kosong");
        jTextKodeProduk.requestFocus();
    }else if (JOptionPane.showConfirmDialog(parentComponent:null,message:"Apakah anda yakin akan menghapus data ini?",
        title: "Informasi",optionType: JOptionPane.OK_CANCEL_OPTION,messageType:JOptionPane.INFORMATION_MESSAGE)==JOptionPane.OK_OPTION) {
        try{
            Connection conn=(Connection) koneksi.koneksiDB();
            Statement stt=conn.createStatement();
            stt.executeUpdate("DELETE FROM produk WHERE KodeProduk='"+KodeProduk+"'");
            BersihData();
            tampilData();
            SetEditOff();
            JOptionPane.showMessageDialog(parentComponent:this,message:"Data berhasil di hapus",title: "Success",messageType:JOptionPane.INFORMATION_MESSAGE);
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(parentComponent:this,"Delete data gagal\n"+e.getMessage(),title: "Error",messageType:JOptionPane.ERROR_MESSAGE);
        }
    }
}

```

## Soure code tombol update

```

// TODO add your handling code here:
String KodeProduk=jTextKodeProduk.getText();
String NamaProduk=jTextNama.getText();
String Satuan=jComboBoxSatuan.getSelectedItem().toString();
String Harga=jTextHarga.getText();
String Stok=jTextStok.getText();

java.util.Date tanggal = new java.util.Date();
java.text.SimpleDateFormat TanggalFormat= new java.text.SimpleDateFormat(pattern:"yyyy-MM-dd HH:mm:ss");
String CreateDate=TanggalFormat.format(date: tanggal);

if (KodeProduk.isEmpty() ) {
    JOptionPane.showMessageDialog(parentComponent:null,message:"Kode produk tidak boleh kosong");
    jTextKodeProduk.requestFocus();
}else if (NamaProduk.isEmpty()) {
    JOptionPane.showMessageDialog(parentComponent:null,message:"Nama produk tidak boleh kosong");
    jTextNama.requestFocus();
}else if (Satuan.isEmpty()) {
    JOptionPane.showMessageDialog(parentComponent:null,message:"Satuan produk tidak boleh kosong");
    jComboBoxSatuan.requestFocus();
}else if (Harga.isEmpty()) {
    JOptionPane.showMessageDialog(parentComponent:null,message:"Harga produk tidak boleh kosong");
    jTextHarga.requestFocus();
}else if (Stok.isEmpty()) {
    JOptionPane.showMessageDialog(parentComponent:null,message:"Stok produk tidak boleh kosong");
    jTextStok.requestFocus();
}

else if (!(new Scanner(source: Harga).hasNextInt())) {
    JOptionPane.showMessageDialog(parentComponent:null,message:"Harga produk harus angka");
    jTextHarga.requestFocus();
}else if (!(new Scanner(source: Stok).hasNextInt())) {
    JOptionPane.showMessageDialog(parentComponent:null,message:"Stok produk harus angka");
    jTextStok.requestFocus();
}

```

```

    }
    else if (!(new Scanner(source: Harga).hasNextInt())) {
        JOptionPane.showMessageDialog(parentComponent: null, message: "Harga produk harus angka");
        jTextFieldHarga.requestFocus();
    } else if (!(new Scanner(source: Stok).hasNextInt())) {
        JOptionPane.showMessageDialog(parentComponent: null, message: "Stok produk harus angka");
        jTextFieldStok.requestFocus();
    } else {
        try {
            Connection conn = (Connection) koneksi.koneksiDB();
            Statement stt = conn.createStatement();
            stt.executeUpdate("UPDATE produk SET NamaProduk='" + NamaProduk + "', Satuan='" + Satuan + "', Harga='" + Harga + "', " +
                "Stok='" + Stok + "' WHERE KodeProduk='" + KodeProduk + "'");
            BersihData();
            tampilData();
            SetEditOff();
            JOptionPane.showMessageDialog(parentComponent: this, message: "Data berhasil diubah", title: "Success", messageType: JOptionPane.INFORMATION_MESSAGE);
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(parentComponent: this, "Ubah data gagal\n" + e.getMessage(), title: "Error", messageType: JOptionPane.ERROR_MESSAGE);
        }
    }
}
}

```

Tampilan aplikasi saat di jalankan

Kode B...	Nama B...	Satuan	Harga	Stok	Create ...
001	mangga	Kg	1000	5	2024-0...
002	aqua	Dus	15000	2	2024-0...

## **BAB III**

### **PENUTUP**

#### **A. KESIMPULAN**

merupakan operasi yang digunakan untuk memanipulasi data database pada sebuah aplikasi. CRUD sebenarnya singkatan Create, Read, Update, Delete. Create digunakan untuk membuat atau menginput data, Read untuk menampilkan data, Update untuk mengedit atau mengubah data sedangkan Delete untuk menghapus data. Dengan CRUD kita bisa membuat form login, register, input, update dan hapus data dan masih banyak lagi. Pada pembuatan form selain CRUD kita juga wajibkan menggunakan session agar nantinya web lebih aman karena dengan menggunakan session maka kita tidak bisa untuk langsung masuk ke halaman sesudah login karena setiap halaman akan dilakukan pengecekan session jadi apabila belum login maka halaman tidak bisa diakses.

#### **B. DAFTAR PUSAKA**

<https://sekolahkoding.com/perpustakaan/pengertian-crud-di-php-dan-mysql>