

Pertemuan XVIII Database

18.1. Keamanan Aplikasi

Untuk menambahkan fitur keamanan pada aplikasi yang dibuat, dapat dilakukan dengan menambahkan menu login. Sehingga setiap user yang login dapat diberikan hak akses terhadap menu yang tersedia.

Data-data user yang akan kita simpan adalah ID user, nama user, password dan hak akses. Tabel untuk menyimpan data tersebut sudah dibuat pada materi sebelumnya.

Karena password merupakan data rahasia dan tidak boleh diketahui oleh siapapun, maka diperlukan pengkodean untuk menyembunyikannya. Pengkodean yang akan digunakan adalah hash SHA-256, yaitu hash yang menghasilkan nilai heksadesimal 256 bit.

Untuk dapat menggunakan pengkodean hash SHA-256, kita harus menambahkan pernyataan import sebagai berikut :

```
import java.security.MessageDigest;
```

Di bawah ini adalah potongan program yang akan kita buat ketika proses konversi dari data tipe *String* ke bentuk hash SHA-256 dan menyimpannya ke dalam table user (TbUser). Pertama dibuat objek md dari *class MessageDigest* dengan algoritma SHA-256 (catatan : penggunaan *getInstance* harus diberi penanganan *exception*, misalnya digunakan di dalam blok *try... catch*). Data *password* yang akan kita konversi ke tipe *array byte* dengan metode *getBytes*, kemudian kita masukkan ke dalam objek md dengan metode *update* dan dilakukan konversi dengan metode *digest()*.

Hasilnya berupa array byte dikonversi bentuk heksadesimal dan disimpan dalam array karakter (class *StringBuffer*) dengan nama sb. Dengan menggunakan metode *toString* maka kita mendapatkan nilai sb dalam tipe *String*. Selanjutnya kita simpan ke dalam table user (TbUser).

```
MessageDigest md = MessageDigest.getInstance("SHA-256");
md.update(inputPassword.getBytes());
byte bytePassword[] = md.digest();
StringBuffer sb = new StringBuffer();
for (i = 0; i < bytePassword.length; i++) {
    sb.append(Integer.toString((bytePassword[i] & 0xff) + 0x100,
16).substring(1));
}
inputPassword = sb.toString();

SQLStatemen = "insert into TbUser values('"+
TxtIDUserUserAccount.getText()+"','"+TxtNamaUserUserAccount.getText()+
"', '"+inputPassword+"', '"+bufHakAkses+"')";
sta = cn.createStatement();
int simpan = sta.executeUpdate(SQLStatemen);

if (simpan==1){
    JOptionPane.showMessageDialog(null,"Sudah tersimpan");
} else {
    JOptionPane.showMessageDialog(null,"Gagal menyimpan data user",
"Kesalahan",JOptionPane.ERROR_MESSAGE);
}
```

Untuk proses penggantian *password*, langkah konversinya sama dengan proses di atas. Perbedaannya hanya terletak pada proses penyimpanan ke dalam *database*, yaitu menggunakan perintah *update*.

Ketika data user sudah tersimpan, maka user bisa menggunakannya untuk login. Pada proses login, user harus memasukkan ID user dan password. Selanjutnya program akan membandingkan ID user dan password dengan data yang tersimpan, jika sama maka user dapat menggunakan program sesuai data hak akses yang diberikan. Tetapi jika tidak sama, maka user diberi informasi bahwa ID user atau passwordnya salah dan tidak dapat menggunakan programnya.

Hash SHA-256 merupakan konversi satu arah, maka hasilnya tidak dapat dikembalikan ke nilai aslinya. Untuk membandingkan password yang sudah di hash, maka password yang dimasukkan harus di-hash terlebih dulu, kemudian hasilnya dibandingkan dengan nilai yang tersimpan.

Proses mengkonversi password yang dimasukkan ketika login dan membandingkan dengan password yang tersimpan serta penentuan hak akses dapat dilihat pada potongan program di bawah ini :

```
char[] Password = TxtPasswordLogin.getPassword();
String inputPassword = new String(Password);

MessageDigest md=null;
try{
    md = MessageDigest.getInstance("SHA-256");
} catch (Exception ex){
    JOptionPane.showMessageDialog(null,"Kesalahan pada hash SHA-
256\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
}
md.update(inputPassword.getBytes());
byte bytePassword[] = md.digest();

StringBuffer sb = new StringBuffer();
int i;
for (i = 0; i < bytePassword.length; i++) {
    sb.append(Integer.toString((bytePassword[i] & 0xff) + 0x100,
16).substring(1));
}
inputPassword = sb.toString();

Connection cn = null;

try {
    cn = DriverManager.getConnection(StringConnection);
} catch (Exception ex) {
    JDBC_Err=true;
    JOptionPane.showMessageDialog(null,"Koneksi ke database DbTokoABC
gagal\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
}

if (!JDBC_Err){
    try {
        cn = DriverManager.getConnection(StringConnection);
        String SQLStatemen = "Select * from TbUser where
IDUser='"+TxtIDUserLogin.getText().trim()+"' and
PasswordUser='"+inputPassword+"'";
        Statement sta = cn.createStatement();
        ResultSet rset = sta.executeQuery(SQLStatemen);
```

```

        rset.next();
        if (rset.getRow()>0){
            IDUserAktif=TxtIDUserLogin.getText().trim();
            NamaUserAktif=rset.getString("NamaUser").trim();

            String bufHakAkses =rset.getString("HakAkses");
            sta.close();
            rset.close();

            String bufUserID = TxtIDUserLogin.getText().trim();

            while (bufUserID.length() < 10){
                bufUserID = bufUserID.concat(TxtIDUserLogin.getText().trim());
            }

            byte[] ByteBufHakAkses=bufHakAkses.substring(0,10).getBytes();
            byte[] ByteBufUserID=bufUserID.getBytes();
            for (i=0;i<10;i++){
                ByteBufHakAkses[i]= (byte)(ByteBufHakAkses[i] ^ ByteBufUserID[i]);
            }

            if (bufHakAkses.substring(10,20).equals(new
String(ByteBufHakAkses))) {
                HakAksesUserAktif=bufHakAkses.substring(0,10);
            } else {
                HakAksesUserAktif="";
                while (HakAksesUserAktif.length() < 10){
                    HakAksesUserAktif = HakAksesUserAktif.concat("3");
                }
            }

            boolean Status;
            for (i=0;i<10;i++){
                Status=(HakAksesUserAktif.substring(i,i+1).equals("1") ||
HakAksesUserAktif.substring(i,i+1).equals("2"));

                switch (i){
                    case 0:MenuBarang.setEnabled(Status); break;
                    case 1:MenuCustomer.setEnabled(Status); break;
                    case 2:MenuUserAccount.setEnabled(Status); break;
                    case 3:MenuPenjualan.setEnabled(Status); break;
                }
            }

            frmLogin.setVisible(false);
            MenuLogin.setText("Logout");
        } else {
            sta.close();
            rset.close();

            JOptionPane.showMessageDialog(null,"ID User atau password salah");
        }
    } catch (Exception ex){
        JOptionPane.showMessageDialog(null,"Tidak dapat membuka tabel
TbUser\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
    }
}

```

18.2. Kode Program Contoh Aplikasi Database

Kode program contoh aplikasi database secara lengkap dapat dilihat di bawah ini :

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableColumn;
import javax.swing.table.*;
import java.sql.*;
import java.util.*;
import java.text.*;
import java.security.MessageDigest;

public class AplikasiToko extends JFrame{
    /*Untuk koneksi ke MySQL*/
    final static String StringDriver="com.mysql.jdbc.Driver";
    final static String StringConnection =
"jdbc:mysql://localhost:3306/DbTokoABC?user=root&password=root";

    /*Untuk koneksi ke SQLServer*/
    //final static String StringDriver =
"com.microsoft.sqlserver.jdbc.SQLServerDriver";
    //final static String StringConnection =
"jdbc:sqlserver://localhost:1433;databaseName=DbTokoABC; integratedSecurity
= true";

    /*Mendefinisikan variabel untuk user aktif*/
    static String IDUserAktif="";
    static String NamaUserAktif="";
    static String HakAksesUserAktif="";

    /*Mendefinisikan Frame*/
    private JDesktopPane frmMDI;
    private JInternalFrame frmBarang;
    private JInternalFrame frmCustomer;
    private JInternalFrame frmUserAccount;
    private JInternalFrame frmPenjualan;
    private JInternalFrame frmLogin;

    private JFrame frmDaftarBarang;
    private JFrame frmDaftarCustomer;
    private JFrame frmJualDaftarBarang;
    private JFrame frmJualDaftarCustomer;

    /*Mendefinisikan Panel untuk masing-masing frame*/
    private JPanel pnlBarang;
    private JPanel pnlCustomer;
    private JPanel pnlUserAccount;
    private JPanel pnlPenjualan;
    private JPanel pnlLogin;

    private String DaftarMenu[] = {"Master Data","Transaksi"};
    private String DaftarSubMenu[] = {"Barang","Customer","User Account",
"Penjualan"};

    private JMenuBar MenuBar = new JMenuBar();

    /*Mendefinisikan item dari menu bar dan sub menunya*/
    private JMenu MenuMaster = new JMenu("Master Data");
```

```
private JMenuItem MenuBarang = new JMenuItem("Barang"),
    MenuCustomer = new JMenuItem("Customer"),
    MenuUserAccount = new JMenuItem("User Account");

private JMenu MenuTransaksi = new JMenu("Transaksi");
private JMenuItem MenuPenjualan = new JMenuItem("Penjualan");

private JMenu MenuAplikasi = new JMenu("Aplikasi");
private JMenuItem MenuLogin = new JMenuItem("Login");
private JMenuItem MenuExit = new JMenuItem("Exit");

/*Menu Popup User Account*/
private JPopupMenu MenuPopupHakAkses = new JPopupMenu();
private JMenuItem MenuHakAksesFull = new JMenuItem("Full"),
    MenuHakAksesView = new JMenuItem("View"),
    MenuHakAksesDeny = new JMenuItem("Deny");

/*Mendefinisikan komponen-komponen untuk Form DaftarBarang*/
private JPanel pnlDaftarBarang;

    static String JudulKolomTabelDaftarBarang[] = {"Kode", "Nama Barang",
    "Satuan"};
    static DefaultTableModel ModelTabelDaftarBarang = new
DefaultTableModel(null, JudulKolomTabelDaftarBarang);
    static JTable TabelDaftarBarang = new JTable(){
        public boolean isCellEditable(int rowIndex, int colIndex) {
            return false; //Disable Editing
        }
    };
    JScrollPane ScrollBarDaftarBarang = new JScrollPane();

private static JButton TblDaftarBarangPilih = new JButton("Pilih");
private static JButton TblDaftarBarangBatal = new JButton("Batal");

/*Mendefinisikan komponen-komponen untuk Form DaftarCustomer*/
private JPanel pnlDaftarCustomer;

    static String JudulKolomTabelDaftarCustomer[] = {"Kode", "Nama
Customer", "Alamat Customer"};
    static DefaultTableModel ModelTabelDaftarCustomer = new
DefaultTableModel(null, JudulKolomTabelDaftarCustomer);
    static JTable TabelDaftarCustomer = new JTable(){
        public boolean isCellEditable(int rowIndex, int colIndex) {
            return false; //Disable Editing
        }
    };
    JScrollPane ScrollBarDaftarCustomer = new JScrollPane();

private static JButton TblDaftarCustomerPilih = new JButton("Pilih");
private static JButton TblDaftarCustomerBatal = new JButton("Batal");

/*Mendefinisikan komponen-komponen untuk Form JualDaftarBarang*/
private JPanel pnlJualDaftarBarang;

    static String JudulKolomTabelJualDaftarBarang[] = {"Kode", "Nama
Barang", "Satuan"};
    static DefaultTableModel ModelTabelJualDaftarBarang = new
DefaultTableModel(null, JudulKolomTabelJualDaftarBarang);
    static JTable TabelJualDaftarBarang = new JTable(){
        public boolean isCellEditable(int rowIndex, int colIndex) {
            return false; //Disable Editing
```

```

    }
};
JScrollPane ScrollBarJualDaftarBarang = new JScrollPane();

private static JButton TblJualDaftarBarangPilih = new JButton("Pilih");
private static JButton TblJualDaftarBarangBatal = new JButton("Batal");

/*Mendefinisikan komponen-komponen untuk Form DaftarCustomer*/
private JPanel pnlJualDaftarCustomer;

static String JudulKolomTabelJualDaftarCustomer[] = {"Kode", "Nama
Customer", "Alamat Customer"};
static DefaultTableModel ModelTabelJualDaftarCustomer = new
DefaultTableModel(null, JudulKolomTabelJualDaftarCustomer);
static JTable TabelJualDaftarCustomer = new JTable(){
    public boolean isCellEditable(int rowIndex, int colIndex) {
        return false;    //Disable Editing
    }
};
JScrollPane ScrollBarJualDaftarCustomer = new JScrollPane();

private static JButton TblJualDaftarCustomerPilih = new
JButton("Pilih");
private static JButton TblJualDaftarCustomerBatal = new
JButton("Batal");

/*Mendefinisikan komponen-komponen untuk Form Barang*/
private static JLabel LblKodeBarang = new JLabel("Kode Barang");
private static JTextField TxtKodeBarang = new JTextField();
private static JLabel LblNamaBarang = new JLabel("Nama Barang");
private static JTextField TxtNamaBarang = new JTextField();
private static JLabel LblSatuanBarang = new JLabel("Satuan");
private static JTextField TxtSatuanBarang = new JTextField();
private static JLabel LblHargaBarang = new JLabel("Harga Barang");
private static JTextField TxtHargaBarang = new JTextField();
private static JLabel LblStockBarang = new JLabel("Stock Barang");
private static JTextField TxtStockBarang = new JTextField();

private static JButton TblBarangDaftar = new JButton("Daftar");
private static JButton TblBarangDelete = new JButton("Delete");
private static JButton TblBarangSave = new JButton("Save");
private static JButton TblBarangCancel = new JButton("Cancel");

/*Mendefinisikan komponen-komponen untuk Form Customer*/
private static JLabel LblKodeCustomer = new JLabel("Kode Customer");
private static JTextField TxtKodeCustomer = new JTextField();
private static JLabel LblNamaCustomer = new JLabel("Nama Customer");
private static JTextField TxtNamaCustomer = new JTextField();
private static JLabel LblAlamatCustomer = new JLabel("Alamat Customer");
private static JTextField TxtAlamatCustomer = new JTextField();
private static JLabel LblNoTeleponCustomer = new JLabel("No. Telepon");
private static JTextField TxtNoTeleponCustomer = new JTextField();
private static JLabel LblEmailCustomer = new JLabel("Email");
private static JTextField TxtEmailCustomer = new JTextField();

private static JButton TblCustomerDaftar = new JButton("Daftar");
private static JButton TblCustomerDelete = new JButton("Delete");
private static JButton TblCustomerSave = new JButton("Save");
private static JButton TblCustomerCancel = new JButton("Cancel");

/*Komponen untuk Form User Account*/

```

```

private static JLabel LblIDUserUserAccount = new JLabel("ID User");
private static JTextField TxtIDUserUserAccount = new JTextField();
private static JLabel LblNamaUserUserAccount = new JLabel("Nama User");
private static JTextField TxtNamaUserUserAccount = new JTextField();

private static JButton TblUserAccountSave = new JButton("Save");
private static JButton TblUserAccountChangePassword = new
JButton("Change Password");
private static JButton TblUserAccountDelete = new JButton("Delete");
private static JButton TblUserAccountClose = new JButton("Close");
private static JButton TblUserAccountFullAll = new JButton("Full All");
private static JButton TblUserAccountViewAll = new JButton("View All");
private static JButton TblUserAccountDenyAll = new JButton("Deny All");

static String UserAccountTabelJudulKolom[] = {"Menu", "Sub Menu", "Hak
Akses"};
static DefaultTableModel UserAccountModelTabel = new
DefaultTableModel(null, UserAccountTabelJudulKolom);
static JTable UserAccountTabel = new JTable();
JScrollPane UserAccountScrollBar = new JScrollPane();

/*Mendefinisikan komponen-komponen untuk Form Penjualan*/
private JPanel pnlCustomerJual;
private JPanel pnlDetilPenjualan;
private static JLabel LblNoNotaJual = new JLabel("No. Nota");
private static JTextField TxtNoNotaJual = new JTextField();
private static JLabel LblKodeCustomerJual = new JLabel("Kode Customer");
private static JTextField TxtKodeCustomerJual = new JTextField();
private static JLabel LblNamaCustomerJual = new JLabel("Nama Customer");
private static JTextField TxtNamaCustomerJual = new JTextField();
private static JLabel LblKodeBarangJual = new JLabel("Kode Barang");
private static JTextField TxtKodeBarangJual = new JTextField();
private static JLabel LblNamaBarangJual = new JLabel("Nama Barang");
private static JTextField TxtNamaBarangJual = new JTextField();
private static JLabel LblHargaBarangJual = new JLabel("Harga");
private static JTextField TxtHargaBarangJual = new JTextField();
private static JLabel LblJumlahJual = new JLabel("Jumlah");
private static JTextField TxtJumlahJual = new JTextField();
private static JLabel LblSubTotalJual = new JLabel("Sub Total");
private static JTextField TxtSubTotalJual = new JTextField();
private static JLabel LblTotalJual = new JLabel("Total");
private static JTextField TxtTotalJual = new JTextField();
private static JLabel LblBayarJual = new JLabel("Pembayaran");
private static JTextField TxtBayarJual = new JTextField();
private static JLabel LblKembaliJual = new JLabel("Kembalian");
private static JTextField TxtKembaliJual = new JTextField();

static String JudulKolomTabelPenjualan[] = {"No.", "Kode", "Nama Barang",
"Harga", "Jumlah", "Total"};
static DefaultTableModel ModelTabelPenjualan = new
DefaultTableModel(null, JudulKolomTabelPenjualan);
static JTable TabelPenjualan = new JTable(){
    public boolean isCellEditable(int rowIndex, int colIndex) {
        return false; //Disable Editing
    }
};
JScrollPane ScrollBarPenjualan = new JScrollPane();

private static JButton TblJualBarangDaftar = new JButton("Daftar");
private static JButton TblJualCustomerDaftar = new JButton("Daftar");
private static JButton TblPenjualanNoBaru = new JButton("No. Baru");

```

```

private static JButton TblPenjualanTambah = new JButton("Tambah");
private static JButton TblPenjualanSave = new JButton("Save");
private static JButton TblPenjualanCancel = new JButton("Cancel");

/*Menu Popup Penjualan*/
private JPopupMenu MenuPopupPenjualan = new JPopupMenu();
private JMenuItem MenuPenjualanDelete = new JMenuItem("Delete");

/*Mendefinisikan komponen-komponen untuk Form Login*/
private static JLabel LblIDUserLogin = new JLabel("ID User");
private static JTextField TxtIDUserLogin = new JTextField();
private static JLabel LblPasswordLogin = new JLabel("Password");
private static JPasswordField TxtPasswordLogin = new JPasswordField();

private static JButton TblLoginLogin = new JButton("Login");
private static JButton TblLoginCancel = new JButton("Cancel");

Dimension dimensi = Toolkit.getDefaultToolkit().getScreenSize();

AplikasiToko(){
    super("Aplikasi Toko");
    setSize((int)(0.7*dimensi.width),(int)(0.7*dimensi.height));
    setLocation(dimensi.width/2-getWidth()/2,dimensi.height/2-
getHeight()/2);
    setDefaultCloseOperation(EXIT_ON_CLOSE);
    frmMDI = new JDesktopPane ();
    frmMDI.setLayout(null);
    this.add(frmMDI);

    /*Menambahkan menu MDI*/
    MenuMaster.add(MenuBarang);
    MenuMaster.add(MenuCustomer);
    MenuMaster.addSeparator();
    MenuMaster.add(MenuUserAccount);
    MenuBar.add(MenuMaster);

    MenuTransaksi.add(MenuPenjualan);
    MenuBar.add(MenuTransaksi);

    MenuAplikasi.add(MenuLogin);
    MenuAplikasi.add(MenuExit);
    MenuBar.add(MenuAplikasi);

    /*Mendeteksi event pada menu*/
    MenuBarang.addActionListener(new MenuHandler());
    MenuCustomer.addActionListener(new MenuHandler());
    MenuUserAccount.addActionListener(new MenuHandler());

    MenuPenjualan.addActionListener(new MenuHandler());

    MenuLogin.addActionListener(new MenuHandler());
    MenuExit.addActionListener(new MenuHandler());

    /*Mendeteksi event pada Button di Form DaftarBarang*/
    TblDaftarBarangPilih.addActionListener(new
TombolDaftarBarangHandler());
    TblDaftarBarangBatal.addActionListener(new
TombolDaftarBarangHandler());

    /*Mendeteksi event pada Button di Form Barang*/
    TblBarangDaftar.addActionListener(new TombolBarangHandler());

```



```
TblBarangDelete.addActionListener(new TombolBarangHandler());
TblBarangSave.addActionListener(new TombolBarangHandler());
TblBarangCancel.addActionListener(new TombolBarangHandler());

/*Mendeteksi event pada Button di Form DaftarCustomer*/
TblDaftarCustomerPilih.addActionListener(new
TombolDaftarCustomerHandler());
TblDaftarCustomerBatal.addActionListener(new
TombolDaftarCustomerHandler());

/*Mendeteksi event pada Button di Form Customer*/
TblCustomerDaftar.addActionListener(new TombolCustomerHandler());
TblCustomerDelete.addActionListener(new TombolCustomerHandler());
TblCustomerSave.addActionListener(new TombolCustomerHandler());
TblCustomerCancel.addActionListener(new TombolCustomerHandler());

/*Mendeteksi event pada Button di Form User Account*/
TblUserAccountFullAll.addActionListener(new
TombolUserAccountHandler());
TblUserAccountViewAll.addActionListener(new
TombolUserAccountHandler());
TblUserAccountDenyAll.addActionListener(new
TombolUserAccountHandler());
TblUserAccountSave.addActionListener(new TombolUserAccountHandler());
TblUserAccountChangePassword.addActionListener(new
TombolUserAccountHandler());
TblUserAccountDelete.addActionListener(new
TombolUserAccountHandler());
TblUserAccountClose.addActionListener(new TombolUserAccountHandler());

/*Mendeteksi event pada Button di Form JualDaftarBarang*/
TblJualDaftarBarangPilih.addActionListener(new
TombolJualDaftarBarangHandler());
TblJualDaftarBarangBatal.addActionListener(new
TombolJualDaftarBarangHandler());

/*Mendeteksi event pada Button di Form JualDaftarCustomer*/
TblJualDaftarCustomerPilih.addActionListener(new
TombolJualDaftarCustomerHandler());
TblJualDaftarCustomerBatal.addActionListener(new
TombolJualDaftarCustomerHandler());

/*Mendeteksi event pada Button di Form Penjualan*/
TblJualBarangDaftar.addActionListener(new TombolPenjualanHandler());
TblJualCustomerDaftar.addActionListener(new TombolPenjualanHandler());
TblPenjualanNoBaru.addActionListener(new TombolPenjualanHandler());
TblPenjualanTambah.addActionListener(new TombolPenjualanHandler());
TblPenjualanSave.addActionListener(new TombolPenjualanHandler());
TblPenjualanCancel.addActionListener(new TombolPenjualanHandler());

/*Mendeteksi event pada Button di Form Login*/
TblLoginLogin.addActionListener(new TombolLoginHandler());
TblLoginCancel.addActionListener(new TombolLoginHandler());

/*Mendeteksi event pada TxtKodeBarang di Form Barang*/
TxtKodeBarang.addKeyListener(new KeyAdapter() {
    public void keyPressed(KeyEvent e) {
        int keyCode = e.getKeyCode();
        if (keyCode==KeyEvent.VK_ENTER) {
            /*Mulai mencari data barang*/
            Boolean JDBC_Err=false;
```

```

        Connection cn = null;

        try {
            cn = DriverManager.getConnection(StringConnection);
        } catch (Exception ex) {
            JDBC_Err=true;
            JOptionPane.showMessageDialog(null,"Koneksi ke database
DbTokoABC gagal\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
        }

        if (!JDBC_Err){
            try {
                cn = DriverManager.getConnection(StringConnection);
                String SQLStatemen = "Select * from TbBarang where
KodeBarang='"+TxtKodeBarang.getText()+"'";
                Statement sta = cn.createStatement();
                ResultSet rset = sta.executeQuery(SQLStatemen);

                rset.next();
                if (rset.getRow()>0){
                    TxtNamaBarang.setText(rset.getString("NamaBarang"));
                    TxtSatuanBarang.setText(rset.getString("SatuanBarang"));
                    TxtHargaBarang.setText(rset.getString("HargaBarang"));
                    TxtStockBarang.setText(rset.getString("StockBarang"));
                    sta.close();
                    rset.close();
                } else {
                    sta.close();
                    rset.close();
                    ClearFormBarang();
                    JOptionPane.showMessageDialog(null,"Kode barang tidak
ada");
                }
            } catch (Exception ex){
                JOptionPane.showMessageDialog(null,"Tidak dapat membuka
tabel TbBarang\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
            }
        }
        /*selesai mencari data barang*/
    }
}
});

/*Mendeteksi event pada TxtKodeCustomer di Form Customer*/
TxtKodeCustomer.addKeyListener(new KeyAdapter() {
    public void keyPressed(KeyEvent e) {
        int keyCode = e.getKeyCode();
        if (keyCode==KeyEvent.VK_ENTER) {
            /*Mulai mencari data Customer*/
            Boolean JDBC_Err=false;
            Connection cn = null;

            try {
                cn = DriverManager.getConnection(StringConnection);
            } catch (Exception ex) {
                JDBC_Err=true;
                JOptionPane.showMessageDialog(null,"Koneksi ke database
DbTokoABC gagal\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
            }

            if (!JDBC_Err){

```

```

        try {
            cn = DriverManager.getConnection(StringConnection);
            String SQLStatemen = "Select * from TbCustomer where
KodeCustomer='"+TxtKodeCustomer.getText()+"'";
            Statement sta = cn.createStatement();
            ResultSet rset = sta.executeQuery(SQLStatemen);

            rset.next();
            if (rset.getRow()>0){
                TxtNamaCustomer.setText(rset.getString("NamaCustomer"));

                TxtAlamatCustomer.setText(rset.getString("AlamatCustomer"));
                TxtNoTeleponCustomer.setText(rset.getString("NoTelepon"));
                TxtEmailCustomer.setText(rset.getString("Email"));
                sta.close();
                rset.close();
            } else {
                sta.close();
                rset.close();
                ClearFormCustomer();
                JOptionPane.showMessageDialog(null,"Kode customer tidak
ada");
            }
        } catch (Exception ex){
            JOptionPane.showMessageDialog(null,"Tidak dapat membuka
tabel TbCustomer\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
        }
    }
    /*selesai mencari data customer*/
}
}
});

/*Mendeteksi event pada TxtNoNotaJual di Form Penjualan*/
TxtNoNotaJual.addKeyListener(new KeyAdapter() {
    public void keyPressed(KeyEvent e) {
        int keyCode = e.getKeyCode();
        if (keyCode==KeyEvent.VK_ENTER) {
            /*Mulai mencari data barang*/
            Boolean JDBC_Err=false;
            Connection cn = null;

            try {
                cn = DriverManager.getConnection(StringConnection);
            } catch (Exception ex) {
                JDBC_Err=true;
                JOptionPane.showMessageDialog(null,"Koneksi ke database
DbTokoABC gagal\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
            }

            if (!JDBC_Err){
                try {
                    cn = DriverManager.getConnection(StringConnection);
                    String SQLStatemen = "Select TbDetilPenjualan.NoNota,
TbPenjualan.KodeCustomer,TbCustomer>NamaCustomer,"
                        +"TbDetilPenjualan.KodeBarang,TbBarang>NamaBarang,
TbDetilPenjualan.Harga, TbDetilPenjualan.Qty,"
                        +"(TbDetilPenjualan.Harga*TbDetilPenjualan.Qty) as
Subtotal from TbDetilPenjualan inner join "
                        +"TbPenjualan on
TbDetilPenjualan.NoNota=TbPenjualan.NoNota inner join TbCustomer on "

```

```

        + "TbPenjualan.KodeCustomer=TbCustomer.KodeCustomer inner
join TbBarang on "
        + "TbDetailPenjualan.KodeBarang=TbBarang.KodeBarang where
TbDetailPenjualan.NoNota=' "
        + TxtNoNotaJual.getText() + "' ";
        Statement sta = cn.createStatement();
        ResultSet rset = sta.executeQuery(SQLStatemen);

        rset.next();
        if (rset.getRow() > 0) {

            TxtKodeCustomerJual.setText(rset.getString("KodeCustomer"));

            TxtNamaCustomerJual.setText(rset.getString("NamaCustomer"));

            do {

                ModelTabelPenjualan.insertRow(ModelTabelPenjualan.getRowCount(), new
                Object[] { ModelTabelPenjualan.getRowCount() + 1, rset.getString("KodeBarang"), r
                set.getString("NamaBarang"), rset.getString("Harga"), rset.getString("Qty"), r
                set.getString("SubTotal") });
                } while (rset.next());
                sta.close();
                rset.close();

                int Total=0, Bayar=0, SubTtl=0;
                int i;
                for (i=0; i<TabelPenjualan.getRowCount(); i++) {
                    SubTtl=0;
                    try {

                        SubTtl=Integer.parseInt((String)TabelPenjualan.getModel().getValueAt(i,5));
                    } catch (Exception ex){
                    }
                    Total=Total+SubTtl;
                }

                TxtTotalJual.setText(Integer.toString(Total));

                Bayar=0;
                try {
                    Bayar=Integer.parseInt(TxtBayarJual.getText());
                } catch (Exception ex){
                }
                TxtKembaliJual.setText(Integer.toString(Bayar-Total));

            } else {
                sta.close();
                rset.close();
                ClearFormPenjualan();
                JOptionPane.showMessageDialog(null, "No. Nota belum ada");
            }
        } catch (Exception ex) {
            JOptionPane.showMessageDialog(null, "Tidak dapat membuka
tabel TbPenjualan\n"+ex, "Kesalahan", JOptionPane.ERROR_MESSAGE);
        }
    }

    try {

```

```

TxtSubTotalJual.setText(Integer.toString(Integer.parseInt(TxtHargaBarangJual.getText())*Integer.parseInt(TxtJumlahJual.getText())));
    } catch (Exception ex){
        TxtSubTotalJual.setText("0");
    }
    /*selesai mencari data barang*/
}
}
});

/*Mendeteksi event pada TxtKodeBarangJual di Form Penjualan*/
TxtKodeBarangJual.addKeyListener(new KeyAdapter() {
    public void keyPressed(KeyEvent e) {
        int keyCode = e.getKeyCode();
        if (keyCode==KeyEvent.VK_ENTER) {
            /*Mulai mencari data barang*/
            Boolean JDBC_Err=false;
            Connection cn = null;

            try {
                cn = DriverManager.getConnection(StringConnection);
            } catch (Exception ex) {
                JDBC_Err=true;
                JOptionPane.showMessageDialog(null,"Koneksi ke database
DbTokoABC gagal\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
            }

            if (!JDBC_Err){
                try {
                    cn = DriverManager.getConnection(StringConnection);
                    String SQLStatemen = "Select * from TbBarang where
KodeBarang='"+TxtKodeBarangJual.getText()+"'";
                    Statement sta = cn.createStatement();
                    ResultSet rset = sta.executeQuery(SQLStatemen);

                    rset.next();
                    if (rset.getRow()>0){
                        TxtNamaBarangJual.setText(rset.getString("NamaBarang"));
                        TxtHargaBarangJual.setText(rset.getString("HargaBarang"));
                        sta.close();
                        rset.close();
                    } else {
                        sta.close();
                        rset.close();
                        TxtNamaBarangJual.setText("");
                        TxtHargaBarangJual.setText("");
                        JOptionPane.showMessageDialog(null,"Kode barang tidak
ada");
                    }
                } catch (Exception ex){
                    JOptionPane.showMessageDialog(null,"Tidak dapat membuka
tabel TbBarang\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
                }
            }

            try {

TxtSubTotalJual.setText(Integer.toString(Integer.parseInt(TxtHargaBarangJual.getText())*Integer.parseInt(TxtJumlahJual.getText())));
            } catch (Exception ex){

```

```

        TxtSubTotalJual.setText("0");
    }
    /*selesai mencari data barang*/
}
}
});

/*Mendeteksi event pada TxtKodeCustomerJual di Form Penjualan*/
TxtKodeCustomerJual.addKeyListener(new KeyAdapter() {
    public void keyPressed(KeyEvent e) {
        int keyCode = e.getKeyCode();
        if (keyCode==KeyEvent.VK_ENTER) {
            /*Mulai mencari data Customer*/
            Boolean JDBC_Err=false;
            Connection cn = null;

            try {
                cn = DriverManager.getConnection(StringConnection);
            } catch (Exception ex) {
                JDBC_Err=true;
                JOptionPane.showMessageDialog(null,"Koneksi ke database
DbTokoABC gagal\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
            }

            if (!JDBC_Err){
                try {
                    cn = DriverManager.getConnection(StringConnection);
                    String SQLStatemen = "Select * from TbCustomer where
KodeCustomer='"+TxtKodeCustomerJual.getText()+"'";
                    Statement sta = cn.createStatement();
                    ResultSet rset = sta.executeQuery(SQLStatemen);

                    rset.next();
                    if (rset.getRow()>0){

TxtNamaCustomerJual.setText(rset.getString("NamaCustomer"));
                        sta.close();
                        rset.close();
                    } else {
                        sta.close();
                        rset.close();
                        TxtNamaCustomerJual.setText("");
                        JOptionPane.showMessageDialog(null,"Kode customer tidak
ada");
                    }
                } catch (Exception ex){
                    JOptionPane.showMessageDialog(null,"Tidak dapat membuka
tabel TbCustomer\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
                }
            }
            /*selesai mencari data customer*/
        }
    }
});

/*Mendeteksi event pada TxtJumlahJual di Form Penjualan*/
TxtJumlahJual.addKeyListener(new KeyAdapter() {
    public void keyReleased(KeyEvent keyEvent) {
        try {

```

```

TxtSubTotalJual.setText(Integer.toString(Integer.parseInt(TxtHargaBarangJual.getText())*Integer.parseInt(TxtJumlahJual.getText())));
    } catch (Exception ex){
        TxtSubTotalJual.setText("0");
    }
}
});

/*Mendeteksi event pada TxtPembayaranJual di Form Penjualan*/
TxtBayarJual.addKeyListener(new KeyAdapter() {
    public void keyReleased(KeyEvent keyEvent) {
        int Total=0;
        try {
            Total=Integer.parseInt(TxtTotalJual.getText());
        } catch (Exception ex){
            TxtKembaliJual.setText("0");
        }

        try {
            TxtKembaliJual.setText(Integer.toString(Integer.parseInt(TxtBayarJual.getText())-Total));
        } catch (Exception ex){
            TxtKembaliJual.setText("0");
        }
    }
});

setContentPane(frmMDI);

frmBarang = new JInternalFrame();
frmBarang.setTitle("Master Data Barang");
frmCustomer = new JInternalFrame("Master Data Customer");
frmUserAccount = new JInternalFrame("Master Data User Account");
frmPenjualan = new JInternalFrame("Transaksi Penjualan");
frmLogin = new JInternalFrame("Login");

/*Membuat form DaftarBarang*/
frmDaftarBarang = new JFrame("Daftar Barang");
frmDaftarBarang.setBounds(0,0,450,350);
frmDaftarBarang.setLocation(dimensi.width/2-frmDaftarBarang.getWidth()/2,dimensi.height/2-frmDaftarBarang.getHeight()/2);
frmDaftarBarang.setDefaultCloseOperation(DISPOSE_ON_CLOSE);
frmDaftarBarang.setAlwaysOnTop(true);

/*Membuat form DaftarCustomer*/
frmDaftarCustomer = new JFrame("Daftar Customer");
frmDaftarCustomer.setBounds(0,0,450,350);
frmDaftarCustomer.setLocation(dimensi.width/2-frmDaftarCustomer.getWidth()/2,dimensi.height/2-frmDaftarCustomer.getHeight()/2);
frmDaftarCustomer.setDefaultCloseOperation(DISPOSE_ON_CLOSE);
frmDaftarCustomer.setAlwaysOnTop(true);

/*Membuat form JualDaftarBarang*/
frmJualDaftarBarang = new JFrame("Daftar Barang");
frmJualDaftarBarang.setBounds(0,0,450,350);

```

```

        frmJualDaftarBarang.setLocation(dimensi.width/2-
        frmJualDaftarBarang.getWidth()/2,dimensi.height/2-
        frmJualDaftarBarang.getHeight()/2);
        frmJualDaftarBarang.setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        frmJualDaftarBarang.setAlwaysOnTop(true);

        /*Membuat form JualDaftarCustomer*/
        frmJualDaftarCustomer = new JFrame("Daftar Customer");
        frmJualDaftarCustomer.setBounds(0,0,450,350);
        frmJualDaftarCustomer.setLocation(dimensi.width/2-
        frmJualDaftarCustomer.getWidth()/2,dimensi.height/2-
        frmJualDaftarCustomer.getHeight()/2);
        frmJualDaftarCustomer.setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        frmJualDaftarCustomer.setAlwaysOnTop(true);

        /*Pengaturan tampilan Form DaftarBarang*/
        pnlDaftarBarang = new JPanel ();

        /*Membuat tabel di DaftarBarang*/
        TabelDaftarBarang.setModel(ModelTabelDaftarBarang);
        ScrollBarDaftarBarang.getViewPort().add(TabelDaftarBarang);
        // Disable auto resizing
        TabelDaftarBarang.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
        // set Column width
        TableColumn ColumnTabelDaftarBarang =
        TabelDaftarBarang.getColumnModel().getColumn(0);
        ColumnTabelDaftarBarang.setPreferredWidth(80);
        ColumnTabelDaftarBarang =
        TabelDaftarBarang.getColumnModel().getColumn(1);
        ColumnTabelDaftarBarang.setPreferredWidth(230);
        ColumnTabelDaftarBarang =
        TabelDaftarBarang.getColumnModel().getColumn(2);
        ColumnTabelDaftarBarang.setPreferredWidth(100);

        //Header font set to Bold
        TabelDaftarBarang.getTableHeader().setFont( new Font( "Dialog" ,
        Font.BOLD, 13 ));

        TabelDaftarBarang.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);

        TabelDaftarBarang.setEnabled(true);
        ScrollBarDaftarBarang.setBounds(1,1,frmDaftarBarang.getWidth()-
        10,frmDaftarBarang.getHeight()-100);
        pnlDaftarBarang.add(ScrollBarDaftarBarang);
        /*Menampilkan tombol di form DaftarBarang*/
        TblDaftarBarangPilih.setBounds((frmDaftarBarang.getWidth()/2)-
        100,frmDaftarBarang.getHeight()-85,80,30);
        pnlDaftarBarang.add(TblDaftarBarangPilih);

        TblDaftarBarangBatal.setBounds((frmDaftarBarang.getWidth()/2)+20,frmDaftarB
        arang.getHeight()-85,80,30);
        pnlDaftarBarang.add(TblDaftarBarangBatal);
        pnlDaftarBarang.setLayout(null);
        frmDaftarBarang.add(pnlDaftarBarang);

        /*Pengaturan tampilan Form Barang*/
        pnlBarang = new JPanel ();
        LblKodeBarang.setBounds(30,20,80,20);
        pnlBarang.add(LblKodeBarang);
        TxtKodeBarang.setBounds(120,20,100,20);

```



```

        pnlBarang.add(TxtKodeBarang);
        LblNamaBarang.setBounds(30,45,80,20);
        pnlBarang.add(LblNamaBarang);
        TxtNamaBarang.setBounds(120,45,200,20);
        pnlBarang.add(TxtNamaBarang);
        LblSatuanBarang.setBounds(30,70,80,20);
        pnlBarang.add(LblSatuanBarang);
        TxtSatuanBarang.setBounds(120,70,140,20);
        pnlBarang.add(TxtSatuanBarang);
        LblHargaBarang.setBounds(30,95,80,20);
        pnlBarang.add(LblHargaBarang);
        TxtHargaBarang.setBounds(120,95,100,20);
        pnlBarang.add(TxtHargaBarang);
        LblStockBarang.setBounds(30,120,80,20);
        pnlBarang.add(LblStockBarang);
        TxtStockBarang.setBounds(120,120,60,20);
        pnlBarang.add(TxtStockBarang);
        /*Menampilkan tombol di form Barang*/
        TblBarangDaftar.setBounds(225,16,70,25);
        pnlBarang.add(TblBarangDaftar);
        TblBarangDelete.setBounds(50,180,80,30);
        pnlBarang.add(TblBarangDelete);
        TblBarangSave.setBounds(140,180,80,30);
        pnlBarang.add(TblBarangSave);
        TblBarangCancel.setBounds(230,180,80,30);
        pnlBarang.add(TblBarangCancel);
        pnlBarang.setLayout(null);
        frmBarang.add(pnlBarang);

        /*Pengaturan tampilan Form DaftarCustomer*/
        pnlDaftarCustomer = new JPanel ();

        /*Membuat tabel di DaftarCustomer*/
        TabelDaftarCustomer.setModel(ModelTabelDaftarCustomer);
        ScrollBarDaftarCustomer.getViewPort().add(TabelDaftarCustomer);
        // Disable auto resizing
        TabelDaftarCustomer.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
        // set Column width
        TableColumn ColumnTabelDaftarCustomer =
TabelDaftarCustomer.getColumnModel().getColumn(0);
        ColumnTabelDaftarCustomer.setPreferredWidth(80);
        ColumnTabelDaftarCustomer =
TabelDaftarCustomer.getColumnModel().getColumn(1);
        ColumnTabelDaftarCustomer.setPreferredWidth(230);
        ColumnTabelDaftarCustomer =
TabelDaftarCustomer.getColumnModel().getColumn(2);
        ColumnTabelDaftarCustomer.setPreferredWidth(100);

        //Header font set to Bold
        TabelDaftarCustomer.getTableHeader().setFont( new Font( "Dialog" ,
Font.BOLD, 13 ));

TabelDaftarCustomer.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);

        TabelDaftarCustomer.setEnabled(true);
        ScrollBarDaftarCustomer.setBounds(1,1,frmDaftarCustomer.getWidth()-
10,frmDaftarCustomer.getHeight()-100);
        pnlDaftarCustomer.add(ScrollBarDaftarCustomer);
        /*Menampilkan tombol di form DaftarCustomer*/

```

```

        TblDaftarCustomerPilih.setBounds((frmDaftarCustomer.getWidth()/2)-
100,frmDaftarCustomer.getHeight()-85,80,30);
        pnlDaftarCustomer.add(TblDaftarCustomerPilih);

TblDaftarCustomerBatal.setBounds((frmDaftarCustomer.getWidth()/2)+20,frmDaf
tarCustomer.getHeight()-85,80,30);
        pnlDaftarCustomer.add(TblDaftarCustomerBatal);
        pnlDaftarCustomer.setLayout(null);
        frmDaftarCustomer.add(pnlDaftarCustomer);

/*Pengaturan tampilan Form Customer*/
pnlCustomer = new JPanel ();
LblKodeCustomer.setBounds(30,20,90,20);
pnlCustomer.add(LblKodeCustomer);
TxtKodeCustomer.setBounds(135,20,100,20);
pnlCustomer.add(TxtKodeCustomer);
LblNamaCustomer.setBounds(30,45,100,20);
pnlCustomer.add(LblNamaCustomer);
TxtNamaCustomer.setBounds(135,45,200,20);
pnlCustomer.add(TxtNamaCustomer);
LblAlamatCustomer.setBounds(30,70,100,20);
pnlCustomer.add(LblAlamatCustomer);
TxtAlamatCustomer.setBounds(135,70,320,20);
pnlCustomer.add(TxtAlamatCustomer);
LblNoTeleponCustomer.setBounds(30,95,80,20);
pnlCustomer.add(LblNoTeleponCustomer);
TxtNoTeleponCustomer.setBounds(135,95,130,20);
pnlCustomer.add(TxtNoTeleponCustomer);
LblEmailCustomer.setBounds(30,120,60,20);
pnlCustomer.add(LblEmailCustomer);
TxtEmailCustomer.setBounds(135,120,230,20);
pnlCustomer.add(TxtEmailCustomer);
/*Menampilkan tombol di form Customer*/
TblCustomerDaftar.setBounds(240,16,70,25);
pnlCustomer.add(TblCustomerDaftar);
TblCustomerDelete.setBounds(80,180,80,30);
pnlCustomer.add(TblCustomerDelete);
TblCustomerSave.setBounds(170,180,80,30);
pnlCustomer.add(TblCustomerSave);
TblCustomerCancel.setBounds(260,180,80,30);
pnlCustomer.add(TblCustomerCancel);
pnlCustomer.setLayout(null);
frmCustomer.add(pnlCustomer);

/*Pengaturan tampilan Form JualDaftarBarang*/
pnlJualDaftarBarang = new JPanel ();

/*Membuat tabel di JualDaftarBarang*/
TabelJualDaftarBarang.setModel(ModelTabelJualDaftarBarang);
ScrollBarJualDaftarBarang.getViewPort().add(TabelJualDaftarBarang);
// Disable auto resizing
TabelJualDaftarBarang.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
// set Column width
TableColumn ColumnTabelJualDaftarBarang =
TabelJualDaftarBarang.getColumnModel().getColumn(0);
ColumnTabelJualDaftarBarang.setPreferredWidth(80);
ColumnTabelJualDaftarBarang =
TabelJualDaftarBarang.getColumnModel().getColumn(1);
ColumnTabelJualDaftarBarang.setPreferredWidth(230);
ColumnTabelJualDaftarBarang =
TabelJualDaftarBarang.getColumnModel().getColumn(2);

```

```
ColumnTabelJualDaftarBarang.setPreferredWidth(100);

//Header font set to Bold
TabelJualDaftarBarang.getTableHeader().setFont( new Font( "Dialog" ,
Font.BOLD, 13 ));

TabelJualDaftarBarang.setSelectionMode(ListSelectionModel.SINGLE_SELECTION)
;

TabelJualDaftarBarang.setEnabled(true);

ScrollBarJualDaftarBarang.setBounds(1,1,frmJualDaftarBarang.getWidth()-
10,frmJualDaftarBarang.getHeight()-100);
pnlJualDaftarBarang.add(ScrollBarJualDaftarBarang);
/*Menampilkan tombol di form JualDaftarBarang*/
TblJualDaftarBarangPilih.setBounds((frmJualDaftarBarang.getWidth()/2)-
100,frmJualDaftarBarang.getHeight()-85,80,30);
pnlJualDaftarBarang.add(TblJualDaftarBarangPilih);

TblJualDaftarBarangBatal.setBounds((frmJualDaftarBarang.getWidth()/2)+20,fr
mJualDaftarBarang.getHeight()-85,80,30);
pnlJualDaftarBarang.add(TblJualDaftarBarangBatal);
pnlJualDaftarBarang.setLayout(null);
frmJualDaftarBarang.add(pnlJualDaftarBarang);

/*Pengaturan tampilan Form JualDaftarCustomer*/
pnlJualDaftarCustomer = new JPanel ();

/*Membuat tabel di JualDaftarCustomer*/
TabelJualDaftarCustomer.setModel(ModelTabelJualDaftarCustomer);

ScrollBarJualDaftarCustomer.getViewport().add(TabelJualDaftarCustomer);
// Disable auto resizing
TabelJualDaftarCustomer.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
// set Column width
TableColumn ColumnTabelJualDaftarCustomer =
TabelJualDaftarCustomer.getColumnModel().getColumn(0);
ColumnTabelJualDaftarCustomer.setPreferredWidth(80);
ColumnTabelJualDaftarCustomer =
TabelJualDaftarCustomer.getColumnModel().getColumn(1);
ColumnTabelJualDaftarCustomer.setPreferredWidth(230);
ColumnTabelJualDaftarCustomer =
TabelJualDaftarCustomer.getColumnModel().getColumn(2);
ColumnTabelJualDaftarCustomer.setPreferredWidth(100);

//Header font set to Bold
TabelJualDaftarCustomer.getTableHeader().setFont( new Font( "Dialog" ,
Font.BOLD, 13 ));

TabelJualDaftarCustomer.setSelectionMode(ListSelectionModel.SINGLE_SELECTIO
N);

TabelJualDaftarCustomer.setEnabled(true);

ScrollBarJualDaftarCustomer.setBounds(1,1,frmJualDaftarCustomer.getWidth()-
10,frmJualDaftarCustomer.getHeight()-100);
pnlJualDaftarCustomer.add(ScrollBarJualDaftarCustomer);
/*Menampilkan tombol di form JualDaftarCustomer*/
```

```

TblJualDaftarCustomerPilih.setBounds(( frmJualDaftarCustomer.getWidth()/2)-
100,frmJualDaftarCustomer.getHeight()-85,80,30);
    pnlJualDaftarCustomer.add(TblJualDaftarCustomerPilih);

TblJualDaftarCustomerBatal.setBounds(( frmJualDaftarCustomer.getWidth()/2)+2
0,frmJualDaftarCustomer.getHeight()-85,80,30);
    pnlJualDaftarCustomer.add(TblJualDaftarCustomerBatal);
    pnlJualDaftarCustomer.setLayout(null);
    frmJualDaftarCustomer.add(pnlJualDaftarCustomer);

    /*Pengaturan tampilan Form UserAccount*/
    pnlUserAccount = new JPanel ();
    LblIDUserUserAccount.setBounds(30,20,160,20);
    pnlUserAccount.add(LblIDUserUserAccount);
    TxtIDUserUserAccount.setBounds(100,20,100,20);
    pnlUserAccount.add(TxtIDUserUserAccount);
    LblNamaUserUserAccount.setBounds(30,50,160,20);
    pnlUserAccount.add(LblNamaUserUserAccount);
    TxtNamaUserUserAccount.setBounds(100,50,150,20);
    pnlUserAccount.add(TxtNamaUserUserAccount);
    /*Menampilkan tombol di form UserAccount*/
    TblUserAccountSave.setBounds(380,20,140,30);
    pnlUserAccount.add(TblUserAccountSave);
    TblUserAccountChangePassword.setBounds(380,60,140,30);
    pnlUserAccount.add(TblUserAccountChangePassword);
    TblUserAccountDelete.setBounds(380,100,140,30);
    pnlUserAccount.add(TblUserAccountDelete);
    TblUserAccountClose.setBounds(380,140,140,30);
    pnlUserAccount.add(TblUserAccountClose);

    TblUserAccountFullAll.setBounds(40,270,80,30);
    pnlUserAccount.add(TblUserAccountFullAll);
    TblUserAccountViewAll.setBounds(140,270,80,30);
    pnlUserAccount.add(TblUserAccountViewAll);
    TblUserAccountDenyAll.setBounds(240,270,80,30);
    pnlUserAccount.add(TblUserAccountDenyAll);
    pnlUserAccount.setLayout(null);
    frmUserAccount.add(pnlUserAccount);
    /*Membuat tabel di User Account*/
    UserAccountTabel.setModel(UserAccountModelTabel);
    UserAccountScrollBar.getViewport().add(UserAccountTabel);
    // Disable auto resizing
    UserAccountTabel.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
    // set Column width
    TableColumn UserAccountTabelColumn =
UserAccountTabel.getColumnModel().getColumn(0);
    UserAccountTabelColumn.setPreferredWidth(100);
    UserAccountTabelColumn =
UserAccountTabel.getColumnModel().getColumn(1);
    UserAccountTabelColumn.setPreferredWidth(150);
    UserAccountTabelColumn =
UserAccountTabel.getColumnModel().getColumn(2);
    UserAccountTabelColumn.setPreferredWidth(80);

    UserAccountTabel.setEnabled(true);
    UserAccountScrollBar.setBounds(20,100,350,160);
    pnlUserAccount.add(UserAccountScrollBar);

    /*Menambahkan menu PopUp pada tabel HakAkses*/
    MenuPopupHakAkses.add(MenuHakAksesFull);

```

```

MenuPopupHakAkses.add(MenuHakAksesView);
MenuPopupHakAkses.add(MenuHakAksesDeny);
MenuHakAksesFull.addActionListener(new MenuHakAksesHandler());
MenuHakAksesView.addActionListener(new MenuHakAksesHandler());
MenuHakAksesDeny.addActionListener(new MenuHakAksesHandler());

/*Mendeteksi klik mouse untuk menampilkan PopUp*/
UserAccountTabel.addMouseListener(new MouseAdapter(){
    public void mousePressed(MouseEvent me){
        if(me.isPopupTrigger()){
            MenuPopupHakAkses.show(me.getComponent(), me.getX(), me.getY());
        }
    }
    public void mouseReleased(MouseEvent Me){
        if(Me.isPopupTrigger()){
            MenuPopupHakAkses.show(Me.getComponent(), Me.getX(), Me.getY());
        }
    }
});

int i;// test input di tabel user account
for (i=0;i<=3;i++){
    UserAccountModelTabel.insertRow(i,new
Object[] {DaftarMenu[i/3],DaftarSubMenu[i],"Deny"});
}

/*Pengaturan tampilan Form Penjualan*/
pnlPenjualan = new JPanel();
LblNoNotaJual.setBounds(30,10,50,20);
pnlPenjualan.add(LblNoNotaJual);
TxtNoNotaJual.setBounds(80,10,100,20);
pnlPenjualan.add(TxtNoNotaJual);

TblPenjualanNoBaru.setBounds(190,7,85,25);
pnlPenjualan.add(TblPenjualanNoBaru);

pnlCustomerJual = new JPanel();

pnlCustomerJual.setBorder(BorderFactory.createTitledBorder("Customer"));
pnlCustomerJual.setBounds(20,35,350,80);

LblKodeCustomerJual.setBounds(10,20,95,20);
pnlCustomerJual.add(LblKodeCustomerJual);
TxtKodeCustomerJual.setBounds(105,20,100,20);
pnlCustomerJual.add(TxtKodeCustomerJual);
LblNamaCustomerJual.setBounds(10,45,95,20);
pnlCustomerJual.add(LblNamaCustomerJual);
TxtNamaCustomerJual.setBounds(105,45,200,20);
pnlCustomerJual.add(TxtNamaCustomerJual);
TxtNamaCustomerJual.setEditable(false);

TblJualCustomerDaftar.setBounds(210,16,70,25);
pnlCustomerJual.add(TblJualCustomerDaftar);

pnlCustomerJual.setLayout(null);
pnlPenjualan.add(pnlCustomerJual);

pnlDetilPenjualan = new JPanel();
pnlDetilPenjualan.setBorder(BorderFactory.createTitledBorder("Detil
Penjualan"));
pnlDetilPenjualan.setBounds(20,125,630,230);

```

```

        LblKodeBarangJual.setBounds(10,20,90,20);
        pnlDetilPenjualan.add(LblKodeBarangJual);
        TxtKodeBarangJual.setBounds(90,20,100,20);
        pnlDetilPenjualan.add(TxtKodeBarangJual);
        LblNamaBarangJual.setBounds(10,45,90,20);
        pnlDetilPenjualan.add(LblNamaBarangJual);
        TxtNamaBarangJual.setBounds(90,45,200,20);
        pnlDetilPenjualan.add(TxtNamaBarangJual);
        LblHargaBarangJual.setBounds(300,20,50,20);
        pnlDetilPenjualan.add(LblHargaBarangJual);
        TxtHargaBarangJual.setBounds(350,20,100,20);
        pnlDetilPenjualan.add(TxtHargaBarangJual);
        LblJumlahJual.setBounds(300,45,50,20);
        pnlDetilPenjualan.add(LblJumlahJual);
        TxtJumlahJual.setBounds(350,45,60,20);
        pnlDetilPenjualan.add(TxtJumlahJual);
        LblSubTotalJual.setBounds(460,20,60,20);
        pnlDetilPenjualan.add(LblSubTotalJual);
        TxtSubTotalJual.setBounds(520,20,100,20);
        pnlDetilPenjualan.add(TxtSubTotalJual);

        TblJualBarangDaftar.setBounds(195,16,70,25);
        pnlDetilPenjualan.add(TblJualBarangDaftar);

        TxtNamaBarangJual.setEditable(false);
        TxtHargaBarangJual.setEditable(false);
        TxtSubTotalJual.setEditable(false);

        TblPenjualanTambah.setBounds(520,45,80,25);
        pnlDetilPenjualan.add(TblPenjualanTambah);

        /*Membuat tabel di form penjualan*/
        TabelPenjualan.setModel(ModelTabelPenjualan);
        ScrollBarPenjualan.getViewport().add(TabelPenjualan);
        // Disable auto resizing
        TabelPenjualan.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
        // set Column width
        TableColumn ColumnTabelPenjualan =
TabelPenjualan.getColumnModel().getColumn(0);
        ColumnTabelPenjualan.setPreferredWidth(30);
        ColumnTabelPenjualan = TabelPenjualan.getColumnModel().getColumn(1);
        ColumnTabelPenjualan.setPreferredWidth(80);
        ColumnTabelPenjualan = TabelPenjualan.getColumnModel().getColumn(2);
        ColumnTabelPenjualan.setPreferredWidth(230);
        ColumnTabelPenjualan = TabelPenjualan.getColumnModel().getColumn(3);
        ColumnTabelPenjualan.setPreferredWidth(80);
        ColumnTabelPenjualan = TabelPenjualan.getColumnModel().getColumn(4);
        ColumnTabelPenjualan.setPreferredWidth(80);
        ColumnTabelPenjualan = TabelPenjualan.getColumnModel().getColumn(5);
        ColumnTabelPenjualan.setPreferredWidth(100);

        //Header font set to Bold
        TabelPenjualan.getTableHeader().setFont( new Font( "Dialog" ,
Font.BOLD, 13 ));
        // Column Alignment set to Right Alignment
        DefaultTableCellRenderer rightRenderer = new
DefaultTableCellRenderer();
        rightRenderer.setHorizontalAlignment( JLabel.RIGHT );
        TabelPenjualan.getColumnModel().getColumn(0).setCellRenderer(
rightRenderer );

```

```

        TabelPenjualan.getColumnModel().getColumn(3).setCellRenderer(
rightRenderer );
        TabelPenjualan.getColumnModel().getColumn(4).setCellRenderer(
rightRenderer );
        TabelPenjualan.getColumnModel().getColumn(5).setCellRenderer(
rightRenderer );

        TabelPenjualan.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);

        TabelPenjualan.setEnabled(true);
        ScrollBarPenjualan.setBounds(5,80,620,145);
        pnlDetilPenjualan.add(ScrollBarPenjualan);

        /*Menambahkan menu PopUp pada tabel penjualan*/
        MenuPopupPenjualan.add(MenuPenjualanDelete);
        MenuPenjualanDelete.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){
                JMenuItem TblPilih = (JMenuItem)e.getSource();

                if (TblPilih.getText().equals("Delete")) {
                    if (TabelPenjualan.getSelectedRow()>=0){
ModelTabelPenjualan.removeRow(TabelPenjualan.getSelectedRow());

                        int i;
                        for (i=0;i<TabelPenjualan.getRowCount();i++){
                            TabelPenjualan.setValueAt(i+1,i,0);
                        }
                    }
                }
            }
        });

        /*Mendeteksi klik mouse untuk menampilkan PopUp*/
        TabelPenjualan.addMouseListener(new MouseAdapter(){
            public void mousePressed(MouseEvent me){
                if(me.isPopupTrigger()){
                    MenuPopupPenjualan.show(me.getComponent(), me.getX(),
me.getY());
                }
            }
            public void mouseReleased(MouseEvent Me){
                if(Me.isPopupTrigger()){
                    MenuPopupPenjualan.show(Me.getComponent(), Me.getX(),
Me.getY());
                }
            }
        });

        pnlDetilPenjualan.setLayout(null);
        pnlPenjualan.add(pnlDetilPenjualan);

        /*menampilkan Total, Pembayaran dan Kembalian pada Form Penjualan*/
        LblTotalJual.setBounds(320,370,80,20);
        pnlPenjualan.add(LblTotalJual);
        TxtTotalJual.setBounds(400,370,100,20);
        pnlPenjualan.add(TxtTotalJual);
        LblBayarJual.setBounds(320,400,80,20);
        pnlPenjualan.add(LblBayarJual);
        TxtBayarJual.setBounds(400,400,100,20);
        pnlPenjualan.add(TxtBayarJual);

```

```

        LblKembaliJual.setBounds(320,430,80,20);
        pnlPenjualan.add(LblKembaliJual);
        TxtKembaliJual.setBounds(400,430,100,20);
        pnlPenjualan.add(TxtKembaliJual);

        TxtTotalJual.setEditable(false);
        TxtKembaliJual.setEditable(false);

        TblPenjualanSave.setBounds(540,370,80,40);
        pnlPenjualan.add(TblPenjualanSave);
        TblPenjualanCancel.setBounds(540,420,80,30);
        pnlPenjualan.add(TblPenjualanCancel);
        pnlPenjualan.setLayout(null);
        frmPenjualan.add(pnlPenjualan);

        /*Pengaturan tampilan Form Login*/
        pnlLogin = new JPanel();
        LblIDUserLogin.setBounds(30,20,60,20);
        pnlLogin.add(LblIDUserLogin);
        TxtIDUserLogin.setBounds(95,20,200,20);
        pnlLogin.add(TxtIDUserLogin);
        LblPasswordLogin.setBounds(30,50,600,20);
        pnlLogin.add(LblPasswordLogin);
        TxtPasswordLogin.setBounds(95,50,200,20);
        pnlLogin.add(TxtPasswordLogin);

        TblLoginLogin.setBounds(80,100,80,30);
        pnlLogin.add(TblLoginLogin);
        TblLoginCancel.setBounds(180,100,80,30);
        pnlLogin.add(TblLoginCancel);

        pnlLogin.setLayout(null);
        frmLogin.add(pnlLogin);

        /*Menambahkan Form ke MDI*/
        frmMDI.add(frmBarang);
        frmMDI.add(frmCustomer);
        frmMDI.add(frmUserAccount);
        frmMDI.add(frmPenjualan);
        frmMDI.add(frmLogin);

        /*Mengatur posisi tampilan Form di dalam MDI*/
        frmBarang.setBounds(10,10,367,270);
        frmCustomer.setBounds(30,30,500,270);
        frmUserAccount.setBounds(50,50,550,350);
        frmPenjualan.setBounds(70,70,680,500);
        frmLogin.setBounds(90,90,330,200);

        /*Disable semua menu*/
        MenuBarang.setEnabled(false);
        MenuCustomer.setEnabled(false);
        MenuUserAccount.setEnabled(false);
        MenuPenjualan.setEnabled(false);

        setJMenuBar(MenuBar);
        setVisible(true);
    }

    private class MenuHandler implements ActionListener {
        public void actionPerformed(ActionEvent e){
            JMenuItem M = (JMenuItem)e.getSource();

```



```

        if (M.getText().equals("Barang")) {
TblBarangSave.setEnabled(HakAksesUserAktif.substring(0,1).equals("1"));

        TblBarangDelete.setEnabled(HakAksesUserAktif.substring(0,1).equals("1"
));
        frmBarang.setVisible(true);
        } else if (M.getText().equals("Customer")) {
TblCustomerSave.setEnabled(HakAksesUserAktif.substring(1,2).equals("1"));

        TblCustomerDelete.setEnabled(HakAksesUserAktif.substring(1,2).equals(
"1"));
        frmCustomer.setVisible(true);
        } else if (M.getText().equals("User Account")) {
TxtIDUserUserAccount.setText(IDUserAktif);
TxtNamaUserUserAccount.setText(NamaUserAktif);

        int i;
        for (i=0;i<4;i++){
            if (HakAksesUserAktif.substring(i,i+1).equals("1")){
                UserAccountTabel.setValueAt("Full",i,2);
            } else if (HakAksesUserAktif.substring(i,i+1).equals("2")){
                UserAccountTabel.setValueAt("View",i,2);
            } else {
                UserAccountTabel.setValueAt("Deny",i,2);
            }
        }

TxtIDUserUserAccount.setEnabled(HakAksesUserAktif.substring(2,3).equals("1"
));

TblUserAccountSave.setEnabled(HakAksesUserAktif.substring(2,3).equals("1"))
;

        TblUserAccountDelete.setEnabled(HakAksesUserAktif.substring(2,3).equa
ls("1"));
        frmUserAccount.setVisible(true);
        } else if (M.getText().equals("Penjualan")) {
TblPenjualanSave.setEnabled(HakAksesUserAktif.substring(3,4).equals("1"));
        frmPenjualan.setVisible(true);
        } else if (M==MenuLogin) {
            if (M.getText().equals("Login")){
                frmLogin.setBounds((frmMDI.getWidth()/2)-
(frmLogin.getWidth()/2),(frmMDI.getHeight()/2)-
(frmLogin.getHeight()/2),frmLogin.getWidth(),frmLogin.getHeight());
                frmLogin.setVisible(true);
            } else {
                MenuLogin.setText("Login");
                /*Disable semua menu*/
                MenuBarang.setEnabled(false);
                MenuCustomer.setEnabled(false);
                MenuUserAccount.setEnabled(false);
                MenuPenjualan.setEnabled(false);

                /*Tutup semua Form*/
                frmBarang.setVisible(false);
                frmCustomer.setVisible(false);

```

```

        frmUserAccount.setVisible(false);
        frmPenjualan.setVisible(false);
    }
    } else if (M.getText().equals("Exit")) {
        dispose();
    }
}

private void ClearFormBarang(){
    TxtNamaBarang.setText("");
    TxtSatuanBarang.setText("");
    TxtHargaBarang.setText("");
    TxtStockBarang.setText("");
}

private void ClearFormCustomer(){
    TxtNamaCustomer.setText("");
    TxtAlamatCustomer.setText("");
    TxtNoTeleponCustomer.setText("");
    TxtEmailCustomer.setText("");
}

private void ClearFormPenjualan(){
    TxtKodeCustomerJual.setText("");
    TxtNamaCustomerJual.setText("");
    TxtKodeBarangJual.setText("");
    TxtNamaBarangJual.setText("");
    TxtHargaBarangJual.setText("");
    TxtJumlahJual.setText("");
    TxtSubTotalJual.setText("");
    TxtTotalJual.setText("");
    TxtBayarJual.setText("");
    TxtKembaliJual.setText("");
    while (TabelPenjualan.getRowCount()>0){
        ModelTabelPenjualan.removeRow(0);
    }
}

private class MenuHakAksesHandler implements ActionListener {
    public void actionPerformed(ActionEvent e){
        JMenuItem TblPilih = (JMenuItem)e.getSource();

        if (UserAccountTabel.getSelectedRow()>=0){
            if (TblPilih==MenuHakAksesFull) {
                UserAccountTabel.setValueAt("Full",UserAccountTabel.getSelectedRow(),2);
            } else if (TblPilih==MenuHakAksesView) {
                UserAccountTabel.setValueAt("View",UserAccountTabel.getSelectedRow(),2);
            } else if (TblPilih==MenuHakAksesDeny) {
                UserAccountTabel.setValueAt("Deny",UserAccountTabel.getSelectedRow(),2);
            }
        }
    }
}

/*class untuk memproses ActionEvent dari tombol di form Daftar Barang di
form Barang*/
private class TombolDaftarBarangHandler implements ActionListener {

```

```

public void actionPerformed(ActionEvent e){
    JButton TblPilih = (JButton)e.getSource();
    Boolean JDBC_Err = false;

    if (TblPilih.getText().equals("Pilih")) {
        if (TabelDaftarBarang.getSelectedRow()>=0){

            TxtKodeBarang.setText((String)TabelDaftarBarang.getModel().getValueAt(Tabel
            DaftarBarang.getSelectedRow(),0));
            frmDaftarBarang.setVisible(false);
            frmDaftarBarang.dispose();

            /*Mulai mencari data barang*/
            Connection cn = null;
            try {
                cn = DriverManager.getConnection(StringConnection);
            } catch (Exception ex) {
                JDBC_Err=true;
                JOptionPane.showMessageDialog(null,"Koneksi ke database
                DbTokoABC gagal\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
            }

            if (!JDBC_Err){
                try {
                    cn = DriverManager.getConnection(StringConnection);
                    String SQLStatemen = "Select * from TbBarang where
                    KodeBarang='"+TxtKodeBarang.getText()+"'";
                    Statement sta = cn.createStatement();
                    ResultSet rset = sta.executeQuery(SQLStatemen);

                    rset.next();
                    if (rset.getRow()>0){
                        TxtNamaBarang.setText(rset.getString("NamaBarang"));
                        TxtSatuanBarang.setText(rset.getString("SatuanBarang"));
                        TxtHargaBarang.setText(rset.getString("HargaBarang"));
                        TxtStockBarang.setText(rset.getString("StockBarang"));
                        sta.close();
                        rset.close();
                    } else {
                        sta.close();
                        rset.close();
                        ClearFormBarang();
                        JOptionPane.showMessageDialog(null,"Kode barang tidak
                        ada");
                    }
                } catch (Exception ex){
                    JOptionPane.showMessageDialog(null,"Tidak dapat membuka
                    tabel TbBarang\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
                }
            }
            /*selesai mencari data barang*/
        } else {
            JOptionPane.showMessageDialog(null,"Belum ada yang dipilih, klik
            item di tabel");
        }
    } else if (TblPilih.getText().equals("Batal")) {
        frmDaftarBarang.setVisible(false);
        frmDaftarBarang.dispose();
    }
}
}

```

```

    /*class untuk memproses ActionEvent dari tombol di form Daftar Customer
    di form Customer*/
    private class TombolDaftarCustomerHandler implements ActionListener {
        public void actionPerformed(ActionEvent e){
            JButton TblPilih = (JButton)e.getSource();
            Boolean JDBC_Err = false;

            if (TblPilih.getText().equals("Pilih")) {
                if (TabelDaftarCustomer.getSelectedRow() >= 0){
                    TxtKodeCustomer.setText((String)TabelDaftarCustomer.getModel().getValueAt(T
                    abelDaftarCustomer.getSelectedRow(),0));
                    frmDaftarCustomer.setVisible(false);
                    frmDaftarCustomer.dispose();

                    /*Mulai mencari data Customer*/
                    Connection cn = null;
                    try {
                        cn = DriverManager.getConnection(StringConnection);
                    } catch (Exception ex) {
                        JDBC_Err=true;
                        JOptionPane.showMessageDialog(null,"Koneksi ke database
                        DbTokoABC gagal\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
                    }

                    if (!JDBC_Err){
                        try {
                            cn = DriverManager.getConnection(StringConnection);
                            String SQLStatemen = "Select * from TbCustomer where
                            KodeCustomer='"+TxtKodeCustomer.getText()+"'";
                            Statement sta = cn.createStatement();
                            ResultSet rset = sta.executeQuery(SQLStatemen);

                            rset.next();
                            if (rset.getRow() > 0){
                                TxtNamaCustomer.setText(rset.getString("NamaCustomer"));

                                TxtAlamatCustomer.setText(rset.getString("AlamatCustomer"));
                                TxtNoTeleponCustomer.setText(rset.getString("NoTelepon"));
                                TxtEmailCustomer.setText(rset.getString("Email"));
                                sta.close();
                                rset.close();
                            } else {
                                sta.close();
                                rset.close();
                                ClearFormCustomer();
                                JOptionPane.showMessageDialog(null,"Kode customer tidak
                                ada");
                            }
                        } catch (Exception ex){
                            JOptionPane.showMessageDialog(null,"Tidak dapat membuka
                            tabel TbCustomer\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
                        }
                    }
                    /*selesai mencari data customer*/
                } else {
                    JOptionPane.showMessageDialog(null,"Belum ada yang dipilih, klik
                    item di tabel");
                }
            } else if (TblPilih.getText().equals("Batal")) {

```

```

        frmDaftarCustomer.setVisible(false);
        frmDaftarCustomer.dispose();
    }
}

/*class untuk memproses ActionEvent dari tombol di form Daftar Barang di
form penjualan*/
private class TombolJualDaftarBarangHandler implements ActionListener {
    public void actionPerformed(ActionEvent e){
        JButton TblPilih = (JButton)e.getSource();
        Boolean JDBC_Err = false;

        if (TblPilih.getText().equals("Pilih")) {
            if (TabelJualDaftarBarang.getSelectedRow()>=0){

                TxtKodeBarangJual.setText((String)TabelJualDaftarBarang.getModel().getValue
                At(TabelJualDaftarBarang.getSelectedRow(),0));
                frmJualDaftarBarang.setVisible(false);
                frmJualDaftarBarang.dispose();

                /*Mulai mencari data barang*/
                Connection cn = null;
                try {
                    cn = DriverManager.getConnection(StringConnection);
                } catch (Exception ex) {
                    JDBC_Err=true;
                    JOptionPane.showMessageDialog(null,"Koneksi ke database
                    DbTokoABC gagal\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
                }

                if (!JDBC_Err){
                    try {
                        cn = DriverManager.getConnection(StringConnection);
                        String SQLStatemen = "Select * from TbBarang where
                        KodeBarang='"+TxtKodeBarangJual.getText()+"'";
                        Statement sta = cn.createStatement();
                        ResultSet rset = sta.executeQuery(SQLStatemen);

                        rset.next();
                        if (rset.getRow()>0){
                            TxtNamaBarangJual.setText(rset.getString("NamaBarang"));
                            TxtHargaBarangJual.setText(rset.getString("HargaBarang"));
                            sta.close();
                            rset.close();
                        } else {
                            sta.close();
                            rset.close();
                            ClearFormBarang();
                            JOptionPane.showMessageDialog(null,"Kode barang tidak
                            ada");
                        }
                    } catch (Exception ex){
                        JOptionPane.showMessageDialog(null,"Tidak dapat membuka
                        tabel TbBarang\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
                    }
                }
                /*selesai mencari data barang*/

                try {

```

```

TxtSubTotalJual.setText(Integer.toString(Integer.parseInt(TxtHargaBarangJual.getText())*Integer.parseInt(TxtJumlahJual.getText())));
    } catch (Exception ex){
        TxtSubTotalJual.setText("0");
    }
    } else {
        JOptionPane.showMessageDialog(null,"Belum ada yang dipilih, klik item di tabel");
    }
    } else if (TblPilih.getText().equals("Batal")) {
        frmJualDaftarBarang.setVisible(false);
        frmJualDaftarBarang.dispose();
    }
}
}

/*class untuk memproses ActionEvent dari tombol di form Daftar Customer di form penjualan*/
private class TombolJualDaftarCustomerHandler implements ActionListener
{
    public void actionPerformed(ActionEvent e){
        JButton TblPilih = (JButton)e.getSource();
        Boolean JDBC_Err = false;

        if (TblPilih.getText().equals("Pilih")) {
            if (TabelJualDaftarCustomer.getSelectedRow()>=0){
                TxtKodeCustomerJual.setText((String)TabelJualDaftarCustomer.getModel().getValueAt(TabelJualDaftarCustomer.getSelectedRow(),0));
                frmJualDaftarCustomer.setVisible(false);
                frmJualDaftarCustomer.dispose();

                /*Mulai mencari data Customer*/
                Connection cn = null;
                try {
                    cn = DriverManager.getConnection(StringConnection);
                } catch (Exception ex) {
                    JDBC_Err=true;
                    JOptionPane.showMessageDialog(null,"Koneksi ke database DbTokoABC gagal\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
                }

                if (!JDBC_Err){
                    try {
                        cn = DriverManager.getConnection(StringConnection);
                        String SQLStatemen = "Select * from TbCustomer where KodeCustomer='"+TxtKodeCustomerJual.getText()+"'";
                        Statement sta = cn.createStatement();
                        ResultSet rset = sta.executeQuery(SQLStatemen);

                        rset.next();
                        if (rset.getRow()>0){
                            TxtNamaCustomerJual.setText(rset.getString("NamaCustomer"));

                            sta.close();
                            rset.close();
                        } else {
                            sta.close();
                            rset.close();
                        }
                    }
                }
            }
        }
    }
}

```

```

        ClearFormCustomer();
        JOptionPane.showMessageDialog(null,"Kode customer tidak
ada");
    }
    } catch (Exception ex){
        JOptionPane.showMessageDialog(null,"Tidak dapat membuka
tabel TbCustomer\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
    }
}
/*selesai mencari data customer*/
} else {
    JOptionPane.showMessageDialog(null,"Belum ada yang dipilih, klik
item di tabel");
}
} else if (TblPilih.getText().equals("Batal")) {
    frmJualDaftarCustomer.setVisible(false);
    frmJualDaftarCustomer.dispose();
}
}
}

/*class untuk memproses(ActionEvent dari tombol di form Barang*/
private class TombolBarangHandler implements ActionListener {
    public void actionPerformed(ActionEvent e){
        JButton TblPilih = (JButton)e.getSource();
        Boolean JDBC_Err = false;

        if (TblPilih.getText().equals("Daftar")) {
            /*Mulai mencari data barang*/
            Connection cn = null;

            try {
                cn = DriverManager.getConnection(StringConnection);
            } catch (Exception ex) {
                JDBC_Err=true;
                JOptionPane.showMessageDialog(null,"Koneksi ke database
DbTokoABC gagal\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
            }

            if (!JDBC_Err){
                try {
                    cn = DriverManager.getConnection(StringConnection);
                    String SQLStatemen = "Select KodeBarang>NamaBarang,
SatuanBarang from TbBarang order by NamaBarang";
                    Statement sta = cn.createStatement();
                    ResultSet rset = sta.executeQuery(SQLStatemen);

                    ModelTabelDaftarBarang.setRowCount(0);
                    while (rset.next()){

ModelTabelDaftarBarang.insertRow(ModelTabelDaftarBarang.getRowCount(),new
Object[]{rset.getString("KodeBarang"),rset.getString("NamaBarang"),
rset.getString("SatuanBarang")});
                    }

                    sta.close();
                    rset.close();
                } catch (Exception ex){
                    JOptionPane.showMessageDialog(null,"Tidak dapat membuka tabel
TbBarang\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
                }
            }
        }
    }
}

```

```

    }

    /*selesai mencari data barang*/
    frmDaftarBarang.setVisible(true);
} else if (TblPilih.getText().equals("Delete")) {
    /*Mulai menghapus data barang*/
    Connection cn = null;

    try {
        cn = DriverManager.getConnection(StringConnection);
    } catch (Exception ex) {
        JDBC_Err=true;
        JOptionPane.showMessageDialog(null,"Koneksi ke database
DbTokoABC gagal\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
    }

    if (!JDBC_Err){
        try {
            cn = DriverManager.getConnection(StringConnection);
            String SQLStatemen = "Select * from TbBarang where
KodeBarang='"+TxtKodeBarang.getText()+"'";
            Statement sta = cn.createStatement();
            ResultSet rset = sta.executeQuery(SQLStatemen);

            rset.next();
            if (rset.getRow()>0){
                sta.close();
                rset.close();

                SQLStatemen = "delete from TbBarang where KodeBarang='"+
TxtKodeBarang.getText()+"'";
                sta = cn.createStatement();
                int simpan = sta.executeUpdate(SQLStatemen);

                if (simpan==1){
                    TxtKodeBarang.setText("");
                    ClearFormBarang();
                    JOptionPane.showMessageDialog(null,"Sudah dihapus");
                } else {
                    JOptionPane.showMessageDialog(null,"Gagal menghapus data
barang","Kesalahan",JOptionPane.ERROR_MESSAGE);
                }
            } else {
                sta.close();
                rset.close();

                JOptionPane.showMessageDialog(null,"Kode barang tidak ada");
            }
        } catch (Exception ex){
            JOptionPane.showMessageDialog(null,"Tidak dapat membuka tabel
TbBarang\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
        }
    }

    /*Selesai menghapus data barang*/
} else if (TblPilih.getText().equals("Save")) {
    /*Mulai menyimpan data barang*/
    Connection cn = null;

    try {
        cn = DriverManager.getConnection(StringConnection);
    } catch (Exception ex) {

```



```

        JDBC_Err=true;
        JOptionPane.showMessageDialog(null,"Koneksi ke database
        DbTokoABC gagal\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
    }

    if (!JDBC_Err){
        try {
            cn = DriverManager.getConnection(StringConnection);
            String SQLStatemen = "Select * from TbBarang where
KodeBarang='"+TxtKodeBarang.getText()+"'";
            Statement sta = cn.createStatement();
            ResultSet rset = sta.executeQuery(SQLStatemen);

            rset.next();
            if (rset.getRow()>0){
                sta.close();
                rset.close();
                Object[] arrOpsi = {"Ya","Tidak"};
                int pilih=JOptionPane.showOptionDialog(null,"Kode Barang
sudah ada\nApakah data diupdate?","Konfirmasi",JOptionPane.YES_NO_OPTION,
JOptionPane.QUESTION_MESSAGE,null,arrOpsi,arrOpsi[0]);
                if (pilih==0){
                    SQLStatemen = "update TbBarang set NamaBarang='"+
TxtNamaBarang.getText()+
                        "', SatuanBarang='"+TxtSatuanBarang.getText()+
                        "', HargaBarang='"+TxtHargaBarang.getText()+
                        "', StockBarang='"+TxtStockBarang.getText()+
                        "' where KodeBarang='"+TxtKodeBarang.getText()+"'";
                    sta = cn.createStatement();
                    int simpan = sta.executeUpdate(SQLStatemen);

                    if (simpan==1){
                        TxtKodeBarang.setText("");
                        ClearFormBarang();
                        JOptionPane.showMessageDialog(null,"Sudah tersimpan");
                    } else {
                        JOptionPane.showMessageDialog(null,"Gagal menyimpan data
barang","Kesalahan",JOptionPane.ERROR_MESSAGE);
                    }
                } else {
                    sta.close();
                    rset.close();

                    SQLStatemen = "insert into TbBarang values ('"+
TxtKodeBarang.getText()+"', '"+TxtNamaBarang.getText()+"', '"+
TxtSatuanBarang.getText()+"', '"+TxtHargaBarang.getText()+"', '"+
TxtStockBarang.getText()+"')";
                    sta = cn.createStatement();
                    int simpan = sta.executeUpdate(SQLStatemen);

                    if (simpan==1){
                        TxtKodeBarang.setText("");
                        ClearFormBarang();
                        JOptionPane.showMessageDialog(null,"Sudah tersimpan");
                    } else {
                        JOptionPane.showMessageDialog(null,"Gagal menyimpan data
barang","Kesalahan",JOptionPane.ERROR_MESSAGE);
                    }
                }
            } catch (Exception ex){

```

```

        JOptionPane.showMessageDialog(null,"Tidak dapat membuka tabel
        TbBarang\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
    }
    }
    /*selesai menyimpan data barang*/
} else if (TblPilih.getText().equals("Cancel")) {
    frmBarang.setVisible(false);
}
}
}

/*class untuk memproses ActionEvent dari tombol di form Customer*/
private class TombolCustomerHandler implements ActionListener {
    public void actionPerformed(ActionEvent e){
        JButton TblPilih = (JButton)e.getSource();
        Boolean JDBC_Err = false;

        if (TblPilih.getText().equals("Daftar")) {
            /*Mulai mencari data Customer*/
            Connection cn = null;

            try {
                cn = DriverManager.getConnection(StringConnection);
            } catch (Exception ex) {
                JDBC_Err=true;
                JOptionPane.showMessageDialog(null,"Koneksi ke database
                DbTokoABC gagal\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
            }

            if (!JDBC_Err){
                try {
                    cn = DriverManager.getConnection(StringConnection);
                    String SQLStatemen = "Select KodeCustomer,NamaCustomer,
                    AlamatCustomer from TbCustomer order by NamaCustomer";
                    Statement sta = cn.createStatement();
                    ResultSet rset = sta.executeQuery(SQLStatemen);

                    ModelTabelDaftarCustomer.setRowCount(0);
                    while (rset.next()){

ModelTabelDaftarCustomer.insertRow(ModelTabelDaftarCustomer.getRowCount(),
new Object[]{rset.getString("KodeCustomer"),
rset.getString("NamaCustomer"),rset.getString("AlamatCustomer")});
                }

                sta.close();
                rset.close();
            } catch (Exception ex){
                JOptionPane.showMessageDialog(null,"Tidak dapat membuka tabel
                TbCustomer\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
            }
        }

        /*selesai mencari data Customer*/
        frmDaftarCustomer.setVisible(true);
    } else if (TblPilih.getText().equals("Delete")) {
        /*Mulai menghapus data Customer*/
        Connection cn = null;

        try {
            cn = DriverManager.getConnection(StringConnection);

```

```

        } catch (Exception ex) {
            JDBC_Err=true;
            JOptionPane.showMessageDialog(null,"Koneksi ke database
DbTokoABC gagal\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
        }

        if (!JDBC_Err){
            try {
                cn = DriverManager.getConnection(StringConnection);
                String SQLStatemen = "Select * from TbCustomer where
KodeCustomer='"+TxtKodeCustomer.getText()+"'";
                Statement sta = cn.createStatement();
                ResultSet rset = sta.executeQuery(SQLStatemen);

                rset.next();
                if (rset.getRow()>0){
                    sta.close();
                    rset.close();

                    SQLStatemen = "delete from TbCustomer where KodeCustomer='"+
+TxtKodeCustomer.getText()+"'";
                    sta = cn.createStatement();
                    int simpan = sta.executeUpdate(SQLStatemen);

                    if (simpan==1){
                        TxtKodeCustomer.setText("");
                        ClearFormCustomer();
                        JOptionPane.showMessageDialog(null,"Sudah dihapus");
                    } else {
                        JOptionPane.showMessageDialog(null,"Gagal menghapus data
customer", "Kesalahan",JOptionPane.ERROR_MESSAGE);
                    }
                } else {
                    sta.close();
                    rset.close();
                    JOptionPane.showMessageDialog(null,"Kode customer tidak
ada");
                }
            } catch (Exception ex){
                JOptionPane.showMessageDialog(null,"Tidak dapat membuka tabel
TbCustomer\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
            }
        }
        /*Selesai menghapus data Customer*/
    } else if (TblPilih.getText().equals("Save")) {
        /*Mulai menyimpan data Customer*/
        Connection cn = null;

        try {
            cn = DriverManager.getConnection(StringConnection);
        } catch (Exception ex) {
            JDBC_Err=true;
            JOptionPane.showMessageDialog(null,"Koneksi ke database
DbTokoABC gagal\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
        }

        if (!JDBC_Err){
            try {
                cn = DriverManager.getConnection(StringConnection);
                String SQLStatemen = "Select * from TbCustomer where
KodeCustomer='"+TxtKodeCustomer.getText()+"'";

```

```

        Statement sta = cn.createStatement();
        ResultSet rset = sta.executeQuery(SQLStatemen);

        rset.next();
        if (rset.getRow()>0){
            sta.close();
            rset.close();
            Object[] arrOpsi = {"Ya","Tidak"};
            int pilih=JOptionPane.showOptionDialog(null,"Kode customer
sudah ada\nApakah data diupdate?","Konfirmasi",JOptionPane.YES_NO_OPTION,
JOptionPane.QUESTION_MESSAGE,null,arrOpsi,arrOpsi[0]);
            if (pilih==0){
                SQLStatemen = "update TbCustomer set
NamaCustomer='"+TxtNamaCustomer.getText()+
                "', AlamatCustomer='"+TxtAlamatCustomer.getText()+
                "', NoTelepon='"+TxtNoTeleponCustomer.getText()+
                "', Email='"+TxtEmailCustomer.getText()+
                "' where KodeCustomer='"+
TxtKodeCustomer.getText()+"'";
                sta = cn.createStatement();
                int simpan = sta.executeUpdate(SQLStatemen);

                if (simpan==1){
                    TxtKodeCustomer.setText("");
                    ClearFormCustomer();
                    JOptionPane.showMessageDialog(null,"Sudah tersimpan");
                } else {
                    JOptionPane.showMessageDialog(null,"Gagal menyimpan data
customer","Kesalahan",JOptionPane.ERROR_MESSAGE);
                }
            } else {
                sta.close();
                rset.close();
                SQLStatemen = "insert into TbCustomer values
('"+TxtKodeCustomer.getText()+"', '"+TxtNamaCustomer.getText()+"', '"+TxtAlam
atCustomer.getText()+"', '"+TxtNoTeleponCustomer.getText()+"', '"+TxtEmailCus
tomer.getText()+"'";
                sta = cn.createStatement();
                int simpan = sta.executeUpdate(SQLStatemen);

                if (simpan==1){
                    TxtKodeCustomer.setText("");
                    ClearFormCustomer();
                    JOptionPane.showMessageDialog(null,"Sudah tersimpan");
                } else {
                    JOptionPane.showMessageDialog(null,"Gagal menyimpan data
customer","Kesalahan",JOptionPane.ERROR_MESSAGE);
                }
            }
        } catch (Exception ex){
            JOptionPane.showMessageDialog(null,"Tidak dapat membuka tabel
TbCustomer\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
        }
    }
    /*selesai menyimpan data Customer*/
} else if (TblPilih.getText().equals("Cancel")) {
    frmCustomer.setVisible(false);
}
}
}

```

```

/*class untuk memproses ActionEvent dari tombol di form User Account*/
private class TombolUserAccountHandler implements ActionListener {
    public void actionPerformed(ActionEvent e){
        JButton TblPilih = (JButton)e.getSource();
        Boolean JDBC_Err = false;

        if (TblPilih==TblUserAccountSave) {
            String bufUserID = TxtIDUserUserAccount.getText().trim();
            if (bufUserID.length()>0){
                /*Mulai menyimpan data User Account*/
                Connection cn = null;
                String bufHakAkses = "";

                while (bufUserID.length() < 10){
                    bufUserID = bufUserID.concat(TxtIDUserUserAccount.getText());
                }

                int i;
                for (i=0;i<=(UserAccountModelTabel.getRowCount()-1);i++){
                    if
(((String)UserAccountTabel.getModel().getValueAt(i,2)).equals("Full")) {
                        bufHakAkses = bufHakAkses.concat("1");
                    } else if
(((String)UserAccountTabel.getModel().getValueAt(i,2)).equals("View")) {
                        bufHakAkses = bufHakAkses.concat("2");
                    } else {
                        bufHakAkses = bufHakAkses.concat("3");
                    }
                }

                while (bufHakAkses.length() < 10) {
                    bufHakAkses =
bufHakAkses.concat(Character.toString((char)Math.round(Math.random()*255)))
;
                }

                byte[] ByteBufHakAkses=bufHakAkses.getBytes();
                byte[] ByteBufUserID=bufUserID.getBytes();
                for (i=0;i<10;i++){
                    ByteBufHakAkses[i]= (byte)(ByteBufHakAkses[i] ^
ByteBufUserID[i]);
                }
                bufHakAkses=bufHakAkses.concat(new String(ByteBufHakAkses));

                try {
                    cn = DriverManager.getConnection(StringConnection);
                } catch (Exception ex) {
                    JDBC_Err=true;
                    JOptionPane.showMessageDialog(null,"Koneksi ke database
DbTokoABC gagal\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
                }

                if (!JDBC_Err){
                    try {
                        cn = DriverManager.getConnection(StringConnection);
                        String SQLStatemen = "select * from TbUser where
IDUser='"+TxtIDUserUserAccount.getText()+"'";
                        Statement sta = cn.createStatement();
                        ResultSet rset = sta.executeQuery(SQLStatemen);

```

```

        rset.next();
        if (rset.getRow()>0){
            sta.close();
            rset.close();
            Object[] arrOpsi = {"Ya","Tidak"};
            int pilih=JOptionPane.showOptionDialog(null,"ID User sudah
ada\nApakah data
diupdate?","Konfirmasi",JOptionPane.YES_NO_OPTION,JOptionPane.QUESTION_MESS
AGE,null,arrOpsi,arrOpsi[0]);
            if (pilih==0){
                SQLStatemen = "update TbUser set NamaUser='"+
TxtNamaUserUserAccount.getText()+
                "', HakAkses='"+bufHakAkses+
                "' where IDUser='"+
TxtIDUserUserAccount.getText()+"'";
                sta = cn.createStatement();
                int simpan = sta.executeUpdate(SQLStatemen);

                if (simpan==1){
                    JOptionPane.showMessageDialog(null,"Sudah diupdate");
                } else {
                    JOptionPane.showMessageDialog(null,"Gagal meng-update
data user","Kesalahan",JOptionPane.ERROR_MESSAGE);
                }
            } else {
                sta.close();
                rset.close();

                JPasswordField PasswordField = new JPasswordField();
                int TombolOpsi=JOptionPane.showConfirmDialog(null,
PasswordField, "Ketik Password", JOptionPane.OK_CANCEL_OPTION);
                if (TombolOpsi==JOptionPane.OK_OPTION){
                    char[] Password = PasswordField.getPassword();
                    PasswordField = new JPasswordField();
                    TombolOpsi=JOptionPane.showConfirmDialog(null,
PasswordField, "Ketik Ulang Password", JOptionPane.OK_CANCEL_OPTION);
                    if (TombolOpsi==JOptionPane.OK_OPTION){
                        char[] RetypePassword = PasswordField.getPassword();

                        String inputPassword = new String(Password);
                        String retypePassword = new String(RetypePassword);
                        if (inputPassword.equals(retypePassword)){
                            MessageDigest md = MessageDigest.getInstance("SHA-
256");

                            md.update(inputPassword.getBytes());
                            byte bytePassword[] = md.digest();

                            StringBuffer sb = new StringBuffer();
                            for (i = 0; i < bytePassword.length; i++) {
                                sb.append(Integer.toString((bytePassword[i] &
0xff) + 0x100, 16).substring(1));
                            }
                            inputPassword = sb.toString();

                            SQLStatemen = "insert into TbUser values
('"+TxtIDUserUserAccount.getText()+"', '"+TxtNamaUserUserAccount.getText()+"'
', '"+inputPassword+"', '"+bufHakAkses+"')";
                            sta = cn.createStatement();
                            int simpan = sta.executeUpdate(SQLStatemen);

```

```

        if (simpan==1){
            JOptionPane.showMessageDialog(null,"Sudah
tersimpan");
        } else {
            JOptionPane.showMessageDialog(null,"Gagal
menyimpan data user","Kesalahan",JOptionPane.ERROR_MESSAGE);
        }
        } else {
            JOptionPane.showMessageDialog(null,"Password baru
tidak sama","Kesalahan",JOptionPane.ERROR_MESSAGE);
        }
    }
}
} catch (Exception ex){
    JOptionPane.showMessageDialog(null,"Tidak dapat membuka
tabel TbUserAccount\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
}
}
/*selesai menyimpan data Customer*/
} else {
    JOptionPane.showMessageDialog(null,"ID user tidak boleh
kosong","Kesalahan",JOptionPane.ERROR_MESSAGE);
}
} else if (TblPilih==TblUserAccountChangePassword) {
    /*Mulai ganti Password*/
    JPasswordField PasswordField = new JPasswordField();
    int TombolOpsi=JOptionPane.showConfirmDialog(null, PasswordField,
    "Ketik Password Lama", JOptionPane.OK_CANCEL_OPTION);
    if (TombolOpsi==JOptionPane.OK_OPTION){
        char[] Password = PasswordField.getPassword();
        String inputPassword = new String(Password);

        MessageDigest md=null;
        try{
            md = MessageDigest.getInstance("SHA-256");
        } catch (Exception ex){
            JOptionPane.showMessageDialog(null,"Kesalahan pada hash SHA-
256\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
        }
        md.update(inputPassword.getBytes());
        byte bytePassword[] = md.digest();

        StringBuffer sb = new StringBuffer();
        int i;
        for (i = 0; i < bytePassword.length; i++) {
            sb.append(Integer.toString((bytePassword[i] & 0xff) + 0x100,
16).substring(1));
        }
        inputPassword = sb.toString();

        Connection cn = null;

        try {
            cn = DriverManager.getConnection(StringConnection);
        } catch (Exception ex) {
            JDBC_Err=true;
            JOptionPane.showMessageDialog(null,"Koneksi ke database
DbTokoABC gagal\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
        }
    }
}

```

```

        if (!JDBC_Err){
            try {
                cn = DriverManager.getConnection(StringConnection);
                String SQLStatemen = "Select * from TbUser where
IDUser='"+TxtIDUserUserAccount.getText()+"' and
PasswordUser='"+inputPassword+"'";
                Statement sta = cn.createStatement();
                ResultSet rset = sta.executeQuery(SQLStatemen);

                rset.next();
                if (rset.getRow()>0){
                    sta.close();
                    rset.close();

                    PasswordField = new JPasswordField();
                    TombolOpsi=JOptionPane.showConfirmDialog(null,
PasswordField, "Ketik Password Baru", JOptionPane.OK_CANCEL_OPTION);
                    if (TombolOpsi==JOptionPane.OK_OPTION){
                        Password = PasswordField.getPassword();
                        PasswordField = new JPasswordField();
                        TombolOpsi=JOptionPane.showConfirmDialog(null,
PasswordField, "Ketik Ulang Password Baru", JOptionPane.OK_CANCEL_OPTION);
                        if (TombolOpsi==JOptionPane.OK_OPTION){
                            char[] RetypePassword = PasswordField.getPassword();

                            inputPassword = new String(Password);
                            String retypePassword = new String(RetypePassword);
                            if (inputPassword.equals(retypePassword)){
                                md = MessageDigest.getInstance("SHA-256");
                                md.update(inputPassword.getBytes());
                                bytePassword = md.digest();

                                sb = new StringBuffer();
                                for (i = 0; i < bytePassword.length; i++) {
                                    sb.append(Integer.toString((bytePassword[i] &
0xff) + 0x100, 16).substring(1));
                                }
                                inputPassword = sb.toString();

                                SQLStatemen = "update TbUser set
PasswordUser='"+inputPassword+
                                "' where IDUser='"+
TxtIDUserUserAccount.getText()+"'";
                                sta = cn.createStatement();
                                int simpan = sta.executeUpdate(SQLStatemen);

                                if (simpan==1){
                                    JOptionPane.showMessageDialog(null,"Password sudah
diganti");
                                } else {
                                    JOptionPane.showMessageDialog(null,"Gagal
mengganti password","Kesalahan",JOptionPane.ERROR_MESSAGE);
                                }
                                } else {
                                    JOptionPane.showMessageDialog(null,"Password baru
tidak sama","Kesalahan",JOptionPane.ERROR_MESSAGE);
                                }
                            }
                        }
                    } else {
                        sta.close();

```



```

        rset.close();
        JOptionPane.showMessageDialog(null,"ID User atau password
salah");
    }
    } catch (Exception ex){
        JOptionPane.showMessageDialog(null,"Tidak dapat membuka
tabel TbUser\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
    }
}
}
/*Selesai ganti Password*/
} else if (TblPilih==TblUserAccountDelete) {
    /*Mulai menghapus data user*/
    Connection cn = null;

    try {
        cn = DriverManager.getConnection(StringConnection);
    } catch (Exception ex) {
        JDBC_Err=true;
        JOptionPane.showMessageDialog(null,"Koneksi ke database
DbTokoABC gagal\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
    }

    if (!JDBC_Err){
        try {
            cn = DriverManager.getConnection(StringConnection);
            String SQLStatemen = "Select * from TbUser where
IDUser='"+TxtIDUserUserAccount.getText()+"'";
            Statement sta = cn.createStatement();
            ResultSet rset = sta.executeQuery(SQLStatemen);

            rset.next();
            if (rset.getRow()>0){
                sta.close();
                rset.close();

                SQLStatemen = "delete from TbUser where
IDUser='"+TxtIDUserUserAccount.getText()+"'";
                sta = cn.createStatement();
                int simpan = sta.executeUpdate(SQLStatemen);

                if (simpan==1){
                    TxtIDUserUserAccount.setText(IDUserAktif);
                    TxtNamaUserUserAccount.setText>NamaUserAktif);
                    JOptionPane.showMessageDialog(null,"Sudah dihapus");
                } else {
                    JOptionPane.showMessageDialog(null,"Gagal menghapus data
user","Kesalahan",JOptionPane.ERROR_MESSAGE);
                }
            } else {
                sta.close();
                rset.close();

                JOptionPane.showMessageDialog(null,"ID User tidak ada");
            }
        } catch (Exception ex){
            JOptionPane.showMessageDialog(null,"Tidak dapat membuka tabel
TbUser\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
        }
    }
}

```

```

        /*Selesai menghapus data user*/
    } else if (TblPilih==TblUserAccountClose) {
        frmUserAccount.setVisible(false);
    } else if (TblPilih==TblUserAccountFullAll) {
        int i;
        for (i=0;i<=(UserAccountModelTabel.getRowCount()-1);i++){
            UserAccountTabel.setValueAt("Full",i,2);
        }
    } else if (TblPilih==TblUserAccountViewAll) {
        int i;
        for (i=0;i<=(UserAccountModelTabel.getRowCount()-1);i++){
            UserAccountTabel.setValueAt("View",i,2);
        }
    } else if (TblPilih==TblUserAccountDenyAll) {
        int i;
        for (i=0;i<=(UserAccountModelTabel.getRowCount()-1);i++){
            UserAccountTabel.setValueAt("Deny",i,2);
        }
    }
}
}

/*class untuk memproses ActionEvent dari tombol di form Penjualan*/
private class TombolPenjualanHandler implements ActionListener {
    public void actionPerformed(ActionEvent e){
        JButton TblPilih = (JButton)e.getSource();
        Boolean JDBC_Err = false;

        if (TblPilih==TblJualCustomerDaftar) {
            /*Mulai mencari data Customer*/
            Connection cn = null;

            try {
                cn = DriverManager.getConnection(StringConnection);
            } catch (Exception ex) {
                JDBC_Err=true;
                JOptionPane.showMessageDialog(null,"Koneksi ke database
                DbTokoABC gagal\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
            }

            if (!JDBC_Err){
                try {
                    cn = DriverManager.getConnection(StringConnection);
                    String SQLStatemen = "Select KodeCustomer,NamaCustomer,
                    AlamatCustomer from TbCustomer order by NamaCustomer";
                    Statement sta = cn.createStatement();
                    ResultSet rset = sta.executeQuery(SQLStatemen);

                    ModelTabelJualDaftarCustomer.setRowCount(0);
                    while (rset.next()){

ModelTabelJualDaftarCustomer.insertRow(ModelTabelJualDaftarCustomer.getRowC
ount(),new
Object[] {rset.getString("KodeCustomer"),rset.getString("NamaCustomer"),rset
.getString("AlamatCustomer")});
                    }

                    sta.close();
                    rset.close();
                } catch (Exception ex){

```

```

        JOptionPane.showMessageDialog(null,"Tidak dapat membuka tabel
TbCustomer\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
    }

    /*selesai mencari data Customer*/
    frmJualDaftarCustomer.setVisible(true);
} else if (TblPilih==TblJualBarangDaftar) {
    /*Mulai mencari data barang*/
    Connection cn = null;

    try {
        cn = DriverManager.getConnection(StringConnection);
    } catch (Exception ex) {
        JDBC_Err=true;
        JOptionPane.showMessageDialog(null,"Koneksi ke database
DbTokoABC gagal\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
    }

    if (!JDBC_Err){
        try {
            cn = DriverManager.getConnection(StringConnection);
            String SQLStatemen = "Select
KodeBarang>NamaBarang,SatuanBarang from TbBarang order by NamaBarang";
            Statement sta = cn.createStatement();
            ResultSet rset = sta.executeQuery(SQLStatemen);

            ModelTabelJualDaftarBarang.setRowCount(0);
            while (rset.next()){

ModelTabelJualDaftarBarang.insertRow(ModelTabelJualDaftarBarang.getRowCount
(),new Object[]{rset.getString("KodeBarang"),
rset.getString("NamaBarang"),rset.getString("SatuanBarang")});
            }

            sta.close();
            rset.close();
        } catch (Exception ex){
            JOptionPane.showMessageDialog(null,"Tidak dapat membuka tabel
TbBarang\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
        }
    }

    /*selesai mencari data barang*/
    frmJualDaftarBarang.setVisible(true);
} else if (TblPilih.getText().equals("No. Baru")) {
    /*Mulai membuat no. baru*/
    ClearFormPenjualan();
    Connection cn = null;

    try {
        cn = DriverManager.getConnection(StringConnection);
    } catch (Exception ex) {
        JDBC_Err=true;
        JOptionPane.showMessageDialog(null,"Koneksi ke database
DbTokoABC gagal\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
    }

    if (!JDBC_Err){
        java.util.Date WaktuSekarang;
        SimpleDateFormat FormatWaktu;

```

```

        String tgl="",bln="",thn="",cari="";
        FormatWaktu = new SimpleDateFormat ("HH:mm:ss EEEE, dd-MM-
yyyy",Locale.getDefault());
        WaktuSekarang = new java.util.Date();

        FormatWaktu.applyPattern("dd");
        try {
            tgl = FormatWaktu.format(WaktuSekarang);
        } catch (NumberFormatException n) {
            tgl = "00";
        }

        FormatWaktu.applyPattern("MM");
        try {
            bln = FormatWaktu.format(WaktuSekarang);
        } catch (NumberFormatException n) {
            bln = "00";
        }

        FormatWaktu.applyPattern("yy");
        try {
            thn = FormatWaktu.format(WaktuSekarang);
        } catch (NumberFormatException n) {
            thn = "00";
        }

        cari=thn+bln+tgl;

        try {
            cn = DriverManager.getConnection(StringConnection);
            String SQLStatemen = "Select * from TbPenjualan where
Substring(NoNota,1,6)='" +cari+ "' order by NoNota Desc";
            Statement sta = cn.createStatement();
            ResultSet rset = sta.executeQuery(SQLStatemen);

            rset.next();
            if (rset.getRow()>0){
                String NoBaru=
Integer.toString(Integer.parseInt(rset.getString("NoNota").substring(6,10))
+1);

                while (NoBaru.length()<4){
                    NoBaru="0"+NoBaru;
                }

                TxtNoNotaJual.setText(cari+NoBaru);
            } else {
                TxtNoNotaJual.setText(cari+"0001");
            }
            sta.close();
            rset.close();
        } catch (Exception ex){
            JOptionPane.showMessageDialog(null,"Tidak dapat membuka tabel
TbPenjualan\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
        }
    }
    /*selesai membuat no. baru*/
} else if (TblPilih.getText().equals("Tambah")) {
    /*Mulai menambah item penjualan*/
    boolean ada=false;
    int i=0;

```

```

        while (!ada && (i<TabelPenjualan.getRowCount())){
            if
(TxtKodeBarangJual.getText().trim().equals(((String)TabelPenjualan.getModel
().getValueAt(i,1)).trim())){
                ada=true;
            } else {
                i++;
            }
        }

        if (ada){
            TabelPenjualan.setValueAt(TxtNamaBarangJual.getText(),i,2);
            TabelPenjualan.setValueAt(TxtHargaBarangJual.getText(),i,3);
            TabelPenjualan.setValueAt(TxtJumlahJual.getText(),i,4);
            TabelPenjualan.setValueAt(TxtSubTotalJual.getText(),i,5);
        } else {

ModelTabelPenjualan.insertRow(ModelTabelPenjualan.getRowCount(),new
Object[] {ModelTabelPenjualan.getRowCount()+1,TxtKodeBarangJual.getText(),Tx
tNamaBarangJual.getText(),TxtHargaBarangJual.getText(),TxtJumlahJual.getTex
t(),TxtSubTotalJual.getText()});
        }

        int Total=0,SubTtl,Bayar;

        for (i=0;i<TabelPenjualan.getRowCount();i++){
            SubTtl=0;
            try {

SubTtl=Integer.parseInt(((String)TabelPenjualan.getModel().getValueAt(i,5));
            } catch (Exception ex){
            }
            Total=Total+SubTtl;
        }

        TxtTotalJual.setText(Integer.toString(Total));
        Bayar=0;
        try {
            Bayar=Integer.parseInt(TxtBayarJual.getText());
        } catch (Exception ex){
        }
        TxtKembaliJual.setText(Integer.toString(Bayar-Total));
        /*Selesai menambah item penjualan*/
    } else if (TblPilih.getText().equals("Save")) {
        /*Mulai menyimpan data Customer*/
        Connection cn = null;
        try {
            cn = DriverManager.getConnection(StringConnection);
        } catch (Exception ex) {
            JDBC_Err=true;
            JOptionPane.showMessageDialog(null,"Koneksi ke database
DbTokoABC gagal\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
        }

        if (!JDBC_Err){
            java.util.Date WaktuSekarang;
            SimpleDateFormat FormatWaktu;
            FormatWaktu = new SimpleDateFormat ("yyyy/MM/dd
HH:mm:ss",Locale.getDefault());
            WaktuSekarang = new java.util.Date();

```

```

String Tanggal="";
Boolean diSimpan=true;

try {
    Tanggal = FormatWaktu.format(WaktuSekarang);
} catch (NumberFormatException n) {
    Tanggal = "1900/01/01 01:01:01";
}

try {
    cn = DriverManager.getConnection(StringConnection);
    String SQLStatemen = "Select * from TbPenjualan where
NoNota='"+TxtNoNotaJual.getText()+"'";
    Statement sta = cn.createStatement();
    ResultSet rset = sta.executeQuery(SQLStatemen);
    int simpan=0;

    rset.next();
    if (rset.getRow()>0){
        sta.close();
        rset.close();

        Object[] arrOpsi = {"Ya","Tidak"};
        int pilih=JOptionPane.showOptionDialog(null,"No. Nota sudah
ada\nApakah data
diupdate?","Konfirmasi",JOptionPane.YES_NO_OPTION,JOptionPane.QUESTION_MESS
AGE,null,arrOpsi,arrOpsi[0]);
        if (pilih==0){
            SQLStatemen = "update TbPenjualan set Tanggal='"+Tanggal
            +"' , KodeCustomer='"+TxtKodeCustomerJual.getText()+"' ,
IDUser='"+B"
            +"' where NoNota='"+TxtNoNotaJual.getText()+"'";
            sta = cn.createStatement();
            simpan = sta.executeUpdate(SQLStatemen);
            sta.close();
        } else {
            diSimpan=false;
        }
    } else {
        sta.close();
        rset.close();
        SQLStatemen = "insert into TbPenjualan values
('"+TxtNoNotaJual.getText()+"' , '"+Tanggal+"' , '"+TxtKodeCustomerJual.getText
()+' , '"+B+"')";
        sta = cn.createStatement();
        simpan = sta.executeUpdate(SQLStatemen);
        sta.close();
    }

    if (diSimpan){
        if (simpan==1){
            int i;
            for (i=0;i<TabelPenjualan.getRowCount();i++){
                SQLStatemen = "Select * from TbDetilPenjualan where
NoNota='"+TxtNoNotaJual.getText()+"' and
KodeBarang='"+(String)TabelPenjualan.getModel().getValueAt(i,1)+"'";
                sta = cn.createStatement();
                rset = sta.executeQuery(SQLStatemen);

                rset.next();
                if (rset.getRow()>0){

```

```

        SQLStatemen = "update TbDetilPenjualan set Harga='" +
Integer.parseInt((String)TabelPenjualan.getModel().getValueAt(i,3))
        + "', Qty='" +
Integer.parseInt((String)TabelPenjualan.getModel().getValueAt(i,4))
        + "' where NoNota='" + TxtNoNotaJual.getText() + "'
and KodeBarang='" + (String)TabelPenjualan.getModel().getValueAt(i,1) + "'";
        sta = cn.createStatement();
        simpan = sta.executeUpdate(SQLStatemen);
        sta.close();
    } else {
        SQLStatemen = "insert into TbDetilPenjualan values('" +
TxtNoNotaJual.getText()
        + "', '" +
(String)TabelPenjualan.getModel().getValueAt(i,1)
        + "', '" +
Integer.parseInt((String)TabelPenjualan.getModel().getValueAt(i,3))
        + "', '" +
Integer.parseInt((String)TabelPenjualan.getModel().getValueAt(i,4)) + "')";
        sta = cn.createStatement();
        simpan = sta.executeUpdate(SQLStatemen);
        sta.close();
    }
}

JOptionPane.showMessageDialog(null,"Sudah tersimpan");
} else {
    JOptionPane.showMessageDialog(null,"Gagal menyimpan data
penjualan", "Kesalahan", JOptionPane.ERROR_MESSAGE);
}
} catch (Exception ex){
    JOptionPane.showMessageDialog(null,"Tidak dapat membuka tabel
TbPenjualan\n"+ex, "Kesalahan", JOptionPane.ERROR_MESSAGE);
}
}
/*selesai menyimpan data Customer*/
} else if (TblPilih.getText().equals("Cancel")) {
    frmPenjualan.setVisible(false);
}
}

/*class untuk memproses(ActionEvent dari tombol di form Login*/
private class TombolLoginHandler implements ActionListener {
    public void actionPerformed(ActionEvent e){
        JButton TblPilih = (JButton)e.getSource();
        Boolean JDBC_Err = false;

        if (TblPilih==TblLoginLogin) {
            /*Mulai proses Login*/
            IDUserAktif="";
            NamaUserAktif="";
            HakAksesUserAktif="";
            if (TxtIDUserLogin.getText().trim().length()>0){
                char[] Password = TxtPasswordLogin.getPassword();
                String inputPassword = new String(Password);

                MessageDigest md=null;
                try{
                    md = MessageDigest.getInstance("SHA-256");
                } catch (Exception ex){

```

```

        JOptionPane.showMessageDialog(null, "Kesalahan pada hash SHA-
256\n"+ex, "Kesalahan", JOptionPane.ERROR_MESSAGE);
    }
    md.update(inputPassword.getBytes());
    byte bytePassword[] = md.digest();

    StringBuffer sb = new StringBuffer();
    int i;
    for (i = 0; i < bytePassword.length; i++) {
        sb.append(Integer.toString((bytePassword[i] & 0xff) + 0x100,
16).substring(1));
    }
    inputPassword = sb.toString();

    Connection cn = null;
    try {
        cn = DriverManager.getConnection(StringConnection);
    } catch (Exception ex) {
        JDBC_Err=true;
        JOptionPane.showMessageDialog(null, "Koneksi ke database
DbTokoABC gagal\n"+ex, "Kesalahan", JOptionPane.ERROR_MESSAGE);
    }

    if (!JDBC_Err){
        try {
            cn = DriverManager.getConnection(StringConnection);
            String SQLStatemen = "Select * from TbUser where IDUser='"+
TxtIDUserLogin.getText().trim()+"' and PasswordUser='"+inputPassword+"'";
            Statement sta = cn.createStatement();
            ResultSet rset = sta.executeQuery(SQLStatemen);

            rset.next();
            if (rset.getRow()>0){
                IDUserAktif=TxtIDUserLogin.getText().trim();
                NamaUserAktif=rset.getString("NamaUser").trim();

                String bufHakAkses =rset.getString("HakAkses");
                sta.close();
                rset.close();

                String bufUserID = TxtIDUserLogin.getText().trim();
                while (bufUserID.length() < 10){
                    bufUserID =
bufUserID.concat(TxtIDUserLogin.getText().trim());
                }

                byte[] ByteBufHakAkses =
bufHakAkses.substring(0,10).getBytes();
                byte[] ByteBufUserID=bufUserID.getBytes();
                for (i=0;i<10;i++){
                    ByteBufHakAkses[i]= (byte)(ByteBufHakAkses[i] ^
ByteBufUserID[i]);
                }

                if (bufHakAkses.substring(10,20).equals(new
String(ByteBufHakAkses))){
                    HakAksesUserAktif=bufHakAkses.substring(0,10);
                } else {
                    HakAksesUserAktif="";
                    while (HakAksesUserAktif.length() < 10){
                        HakAksesUserAktif = HakAksesUserAktif.concat("3");
                    }
                }
            }
        }
    }

```



```

        }
    }

    boolean Status;
    for (i=0;i<10;i++){
        Status=(HakAksesUserAktif.substring(i,i+1).equals("1")
|| HakAksesUserAktif.substring(i,i+1).equals("2"));

        switch (i){
            case 0:MenuBarang.setEnabled(Status); break;
            case 1:MenuCustomer.setEnabled(Status); break;
            case 2:MenuUserAccount.setEnabled(Status); break;
            case 3:MenuPenjualan.setEnabled(Status); break;
        }
    }

    frmLogin.setVisible(false);
    MenuLogin.setText("Logout");
} else {
    sta.close();
    rset.close();
    JOptionPane.showMessageDialog(null,"ID User atau password
salah");
}
} catch (Exception ex){
    JOptionPane.showMessageDialog(null,"Tidak dapat membuka
tabel TbUser\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
}
}
}
/*Selesai proses Login*/
} else if (TblPilih==TblLoginCancel) {
    frmLogin.setVisible(false);
}
}
}

public static void main(String args[]){
    AplikasiToko frameku = new AplikasiToko();
    Boolean JDBC_Err = false;

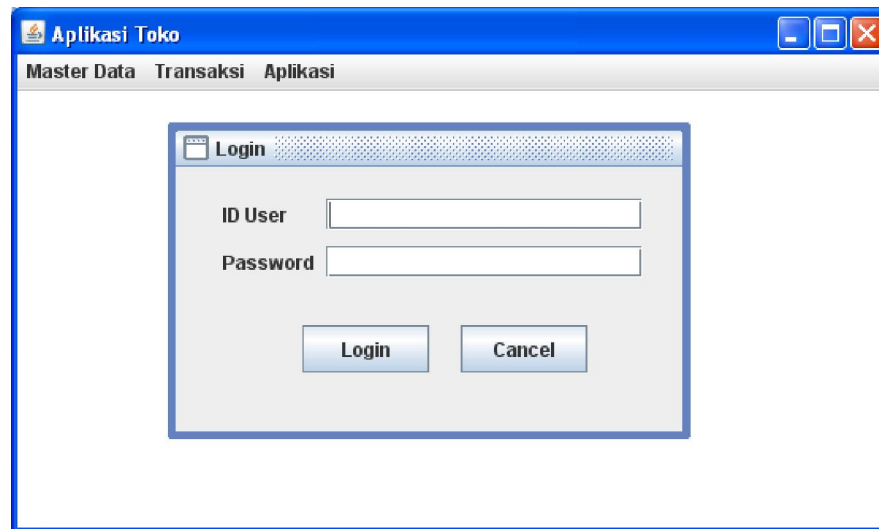
    try{
        Class.forName(StringDriver);
    } catch (Exception ex){
        JDBC_Err=true;
        JOptionPane.showMessageDialog(null,"JDBC Driver tidak ditemukan
atau rusak\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
    }

    if (!JDBC_Err){
        try {
            Connection cn = DriverManager.getConnection(StringConnection);
        } catch (Exception ex) {
            JOptionPane.showMessageDialog(null,"Koneksi ke database DbTokoABC
gagal\n"+ex,"Kesalahan",JOptionPane.ERROR_MESSAGE);
        }
    }
}
}
}
}

```

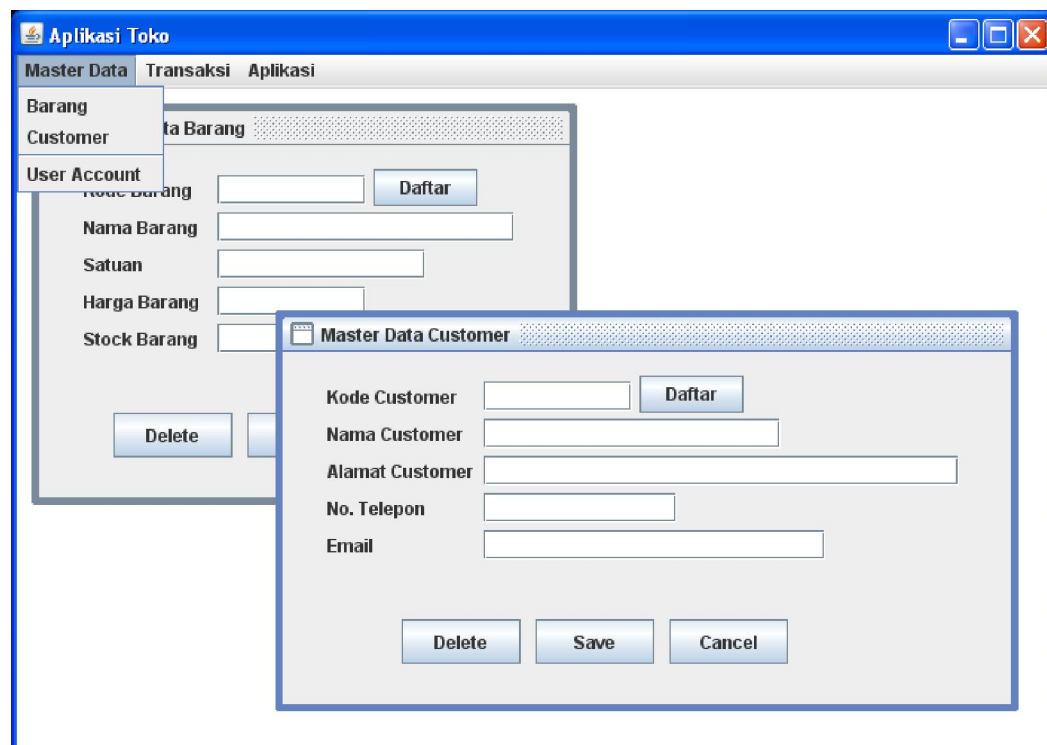
18.3. Tampilan Contoh Aplikasi Database

18.3.1. Form Login



Gambar 18.1. Tampilan Menu Login

18.3.2. Form Master Data Barang dan Customer



Gambar 18.2. Tampilan Menu Master Data

18.3.3. Form Transaksi Penjualan

The screenshot shows a Windows-style application window titled "Aplikasi Toko". It has a menu bar with "Master Data", "Transaksi", and "Aplikasi". The "Transaksi" menu is active, showing a sub-menu "Transaksi Penjualan".

The "Transaksi Penjualan" form contains the following sections:

- Header:** "No. Nota" (Invoice Number) with a text box and a "No. Baru" (New) button.
- Customer Section:** "Customer" label, "Kode Customer" (Customer Code) with a text box and a "Daftar" (List) button, and "Nama Customer" (Customer Name) with a text box.
- Detail Penjualan Section:** "Detail Penjualan" label, "Kode Barang" (Item Code) with a text box and a "Daftar" button, "Nama Barang" (Item Name) with a text box, "Harga" (Price) with a text box, "Jumlah" (Quantity) with a text box, "Sub Total" with a text box, and a "Tambah" (Add) button.
- Table:** A table with 6 columns: "No.", "Kode", "Nama Barang", "Harga", "Jumlah", and "Total". The table is currently empty.
- Summary Section:** "Total" with a text box, "Pembayaran" (Payment) with a text box, "Kembalian" (Change) with a text box, a "Save" button, and a "Cancel" button.

Gambar 18.3. Tampilan Form Transaksi

Referensi:

1. Hariyanto, Bambang, (2007), *Esensi-esensi Bahasa Pemrograman Java*, Edisi 2, Informatika Bandung, November 2007.
2. Utomo, Eko Priyo, (2009), *Panduan Mudah Mengenal Bahasa Java*, Yrama Widya, Juni 2009.
3. Tim Pengembang JENI, JENI 1-6, Depdiknas, 2007