

Nama : Danang Aji Nugroho

NIM : L200180015

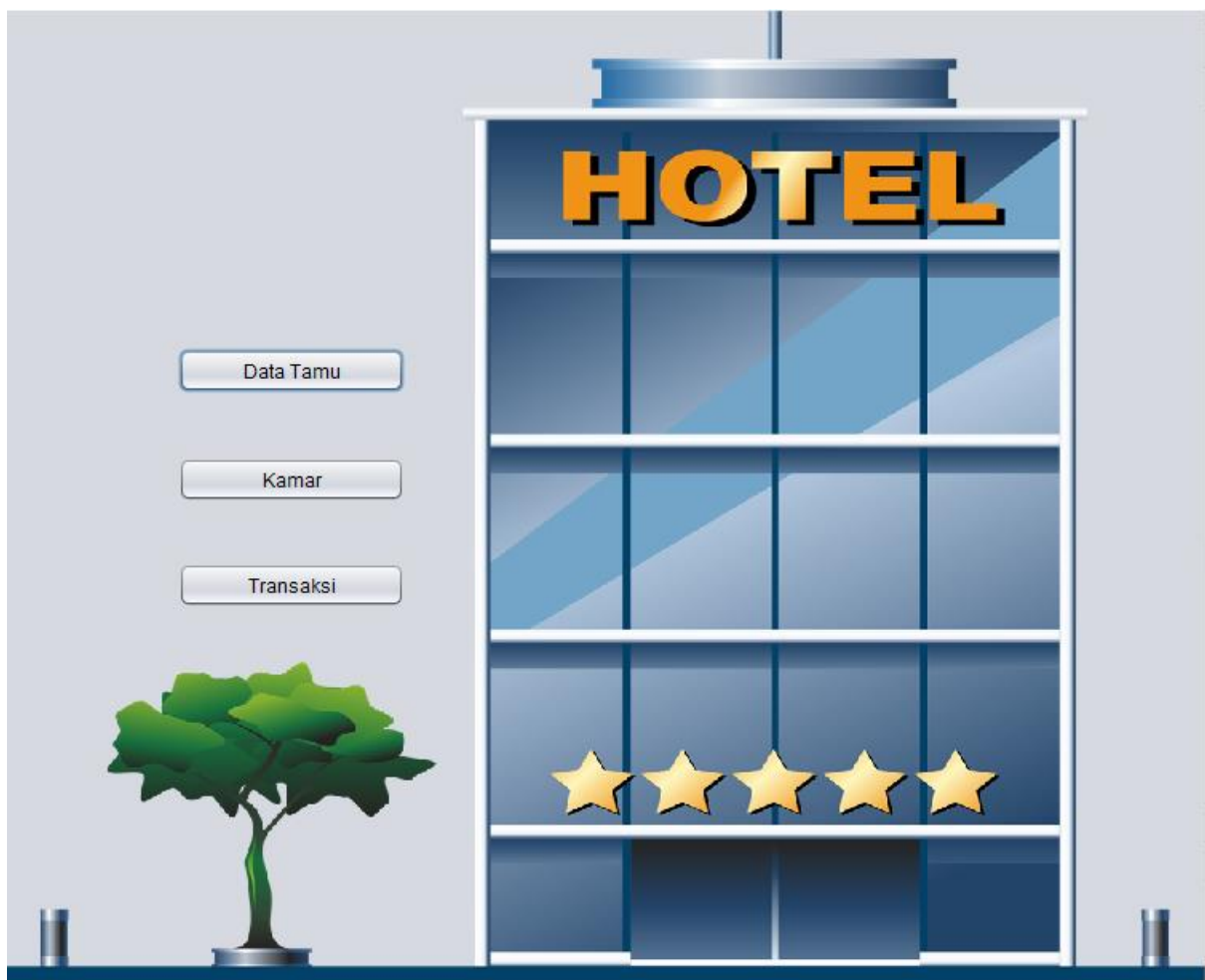
Kelas : A

SBD Modul 12

HOTEL

1.

Home



Identitas Tamu

FORM IDENTITAS TAMU

[Home](#)

No KTP: 2001800155

Nama: danang

Alamat: cangkol

No HP: 08122333

Message

Proses Simpan Data Berhasil..

OK

Cancel
Simpan
Edit
Delete
Next

No.	No KTP	Nama Tamu	Alamat	No HP
1	1	danang	sukoharjo	8222540
2	3857464	aji	jetis	085878222488
3	200180015	AJI NUGROHO	SUKOHARJO	0828282828

Kamar

FORM KAMAR

[Home](#)

No Kamar: A015

Kelas Kamar: VIP

Harga: 1000000

Message

Proses Simpan Data Berhasil..

OK

Cancel
Simpan
Edit
Delete
Next

No.	No Kamar	Kelas Kamar	Harga
1	A007	Reguler	450000
2	B07	VIP	20000
3	V09	Reguler	450000

Transaksi

FORM TRANSAKSI

Home

HOTEL

No Transaksi:

No KTP:

No Kamar:

Harga:

Jumlah Kamar:

Jumlah Hari:

Tanggal:

Total Harga:

Uang Bayar:

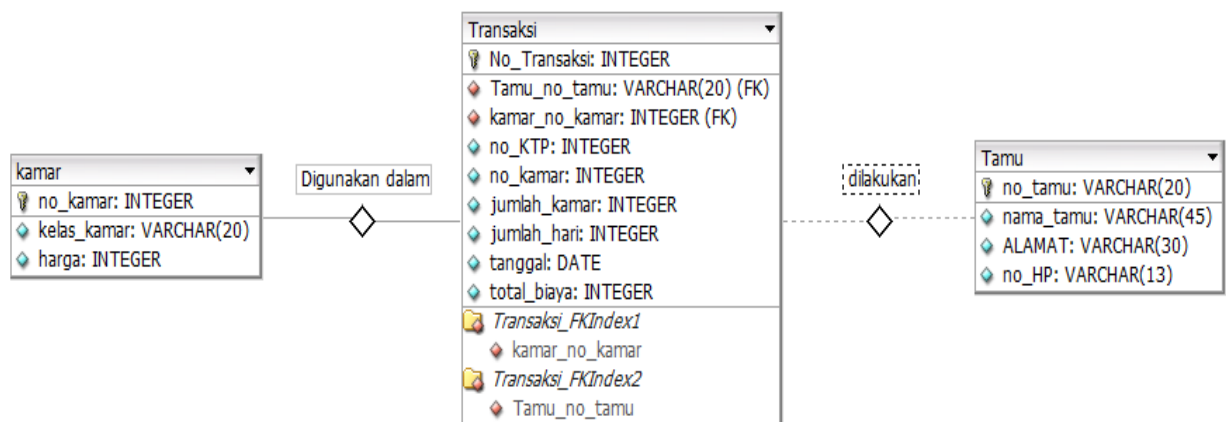
Kembalian:

Message

Proses Simpan Data Berhasil..

No Transaksi	No KTP	No Kamar	Jumlah Kamar	Jumlah Hari	Tanggal	Total Biaya
2	1	A15	1	2	2019-06-14 1...	2000000
7	1	A15	1	2	2020-06-24 0...	2000000

2.



Pada Hotel tersedia Kamar. Kamar memiliki nomor kamar dan Kelas kamar. Terdapat dua Kelas kamar: Reguler dan VIP. Untuk kamar Reguler tarif per malam senilai 450.000 rupiah, sedangkan VIP 1.000.000 rupiah.

Tamu yang ingin memesan kamar harus mengisi Data Tamu.

Data yang diperlukan adalah no, no KTP, nama tamu, Alamat, dan nomor HP.

Setelah registrasi, langsung dilanjutkan transaksi. Pada setiap transaksi memiliki nomor transaksi yang nilainya otomatis bertambah. Data untuk transaksi adalah No Transaksi, No KTP, No Kamar, Jumlah Kamar, Jumlah Hari, Tanggal, dan Total Biaya.

TAMU

- **(SELECT)**

```
private void tampilkan_data() {  
  
    DefaultTableModel model = new DefaultTableModel();  
    model.addColumn("No.");  
    model.addColumn("No KTP");  
    model.addColumn("Nama Tamu");  
    model.addColumn("Alamat");  
    model.addColumn("No HP");  
  
    try {  
        int no = 1;  
        String sql = "SELECT * FROM tamu";  
        java.sql.Connection conn = (Connection)Konfig.configDB();  
        java.sql.Statement stm = conn.createStatement();  
        java.sql.ResultSet res = stm.executeQuery(sql);  
  
        while(res.next()) {  
            model.addRow(new Object[] {no++, res.getString(1), res.getString(2), res.getString(3), res.getString(4)});  
        }  
        tabelTamu.setModel(model);  
    } catch (SQLException e) {  
        System.out.println("Error : " + e.getMessage());  
    }  
}
```

- **(INSERT)**

```
private void btn_simpanActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try {  
        String sql = "INSERT INTO tamu VALUES ('"+txt_noktp.getText()+"', '"+txt_nama.getText()+"', '"+txt_alamat.getText()+"', '"  
        java.sql.Connection conn = (Connection)Konfig.configDB();  
        java.sql.PreparedStatement pstmt = conn.prepareStatement(sql);  
        pstmt.execute();  
        JOptionPane.showMessageDialog(null, "Proses Simpan Data Berhasil..");  
        tampilkan_data();  
        kosongkan_form();  
    } catch (HeadlessException | SQLException e) {  
        JOptionPane.showMessageDialog(this, e.getMessage());  
    }  
}
```

- **(UPDATE)**

```
private void btn_editActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{
        String sql = "UPDATE tamu SET no_ktp='"+txt_noktp.getText()+"',nama='"+txt_nama.getText()+"',alamat='"+txt_alamat.getText()+"'";
        java.sql.Connection conn = (Connection)Konfig.configDB();
        java.sql.PreparedStatement pstm = conn.prepareStatement(sql);
        pstm.execute();
        JOptionPane.showMessageDialog(null, "Edit Data Berhasil...");
    }catch(HeadlessException | SQLException e){
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
    tampilkan_data();
    kosongkan_form();
}
}
```

- **(DELETE)**

```
private void btn_deleteActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{
        String sql = "DELETE FROM tamu WHERE no_ktp='"+txt_noktp.getText()+"'";
        java.sql.Connection conn = (Connection)Konfig.configDB();
        java.sql.PreparedStatement pstm = conn.prepareStatement(sql);
        pstm.execute();
        JOptionPane.showMessageDialog(null, "Hapus Data Berhasil...");
    }catch(HeadlessException | SQLException e){
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
    tampilkan_data();
    kosongkan_form();
}
}
```

KAMAR

- **(SELECT)**

```

private void tampilkan_data(){

    DefaultTableModel model = new DefaultTableModel();
    model.addColumn("No.");
    model.addColumn("No Kamar");
    model.addColumn("Kelas Kamar");
    model.addColumn("Harga");

    try{
        int no = 1;
        String sql = "SELECT * FROM kamar";
        java.sql.Connection conn = (Connection)Konfig.configDB();
        java.sql.Statement stm = conn.createStatement();
        java.sql.ResultSet res = stm.executeQuery(sql);

        while(res.next()){
            model.addRow(new Object[]{no++,res.getString(1),res.getString(2),res.getString(3)});
        }
        tabelKamar.setModel(model);
    }catch (SQLException e){
        System.out.println("Error : " + e.getMessage());
    }
}

```

- **(INSERT)**

```

private void btn_simpanActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{
        String sql = "INSERT INTO kamar VALUES ('"+txt_nokamar.getText()+"','"+cb_kelas.getSelectedItem()+"','"+txt_harga.getText()+"')";
        java.sql.Connection conn = (Connection)Konfig.configDB();
        java.sql.PreparedStatement pstmt = conn.prepareStatement(sql);
        pstmt.execute();
        JOptionPane.showMessageDialog(null, "Proses Simpan Data Berhasil..");
        tampilkan_data();
        kosongkan_form();
    }catch (HeadlessException | SQLException e){
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
}

```

- **(UPDATE)**

```

private void btn_editActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{
        String sql = "UPDATE kamar SET no_kamar='"+txt_nokamar.getText()+"', kelas_kamar='"+cb_kelas.getSelectedItem()+"', harga='"+txt_harga.getText()+"' WHERE no_kamar='"+txt_nokamar.getText()+"'";
        java.sql.Connection conn = (Connection)Konfig.configDB();
        java.sql.PreparedStatement pstmt = conn.prepareStatement(sql);
        pstmt.execute();
        JOptionPane.showMessageDialog(null, "Edit Data Berhasil...");
    }catch (HeadlessException | SQLException e){
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
    tampilkan_data();
    kosongkan_form();
}

```

- **(DELETE)**

```

private void btn_deleteActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{
        String sql = "DELETE FROM kamar WHERE no_kamar='"+txt_nokamar.getText()+"'";
        java.sql.Connection conn = (Connection)Konfig.configDB();
        java.sql.PreparedStatement pstmt = conn.prepareStatement(sql);
        pstmt.execute();
        JOptionPane.showMessageDialog(null, "Hapus Data Berhasil...");
    }catch(HeadlessException | SQLException e){
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
    tampilkan_data();
    kosongkan_form();
}

```

TRANSAKSI

- **(SELECT)**

```

private void tampilkan_data(){
    DefaultTableModel model = new DefaultTableModel();
    model.addColumn("No Transaksi");
    model.addColumn("No KTP");
    model.addColumn("No Kamar");
    model.addColumn("Jumlah Kamar");
    model.addColumn("Jumlah Hari");
    model.addColumn("Tanggal");
    model.addColumn("Total Biaya");

    try{
        String sql = "SELECT * FROM transaksi";
        java.sql.Connection conn = (Connection)Konfig.configDB();
        java.sql.Statement stm = conn.createStatement();
        java.sql.ResultSet res = stm.executeQuery(sql);

        while(res.next()){
            model.addRow(new Object[]{res.getString(1),res.getString(2),res.getString(3),res.getString(4),res.getString(5),res
            });
        }
        tabelTransaksi.setModel(model);
    }catch (SQLException e){
        System.out.println("Error : " + e.getMessage());
    }
}

```

- **(INSERT)**

```

private void btn_simpanActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{
        String sql = "INSERT INTO transaksi VALUES ('"+txt_transaksi.getText()+"','"+txt_noktp.getText()+"','"+txt_nokamar.get
        java.sql.Connection conn = (Connection)Konfig.configDB();
        java.sql.PreparedStatement pstmt = conn.prepareStatement(sql);
        pstmt.execute();
        JOptionPane.showMessageDialog(null, "Proses Simpan Data Berhasil..");
        tampilkan_data();
        kosongkan_form();
    }catch(HeadlessException | SQLException e){
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
}

```

- **(UPDATE)**

```

private void btn_editActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{
        String sql = "UPDATE transaksi SET no_kamarFK='"+txt_nokamar.getText()+"',no_ktpFK='"+txt_noktp.getText()+"',jumlah_kar
        java.sql.Connection conn = (Connection)Konfig.configDB();
        java.sql.PreparedStatement pstmt = conn.prepareStatement(sql);
        pstmt.execute();
        JOptionPane.showMessageDialog(null, "Edit Data Berhasil...");
    }catch(HeadlessException | SQLException e){
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
    tampilkan_data();
    kosongkan_form();
}

```

- **(DELETE)**

```

private void btn_deleteActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{
        String sql = "DELETE FROM transaksi WHERE no_ktpFK='"+txt_noktp.getText()+"'";
        java.sql.Connection conn = (Connection)Konfig.configDB();
        java.sql.PreparedStatement pstmt = conn.prepareStatement(sql);
        pstmt.execute();
        JOptionPane.showMessageDialog(null, "Hapus Data Berhasil...");
    }catch(HeadlessException | SQLException e){
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
    tampilkan_data();
    kosongkan_form();
}

```