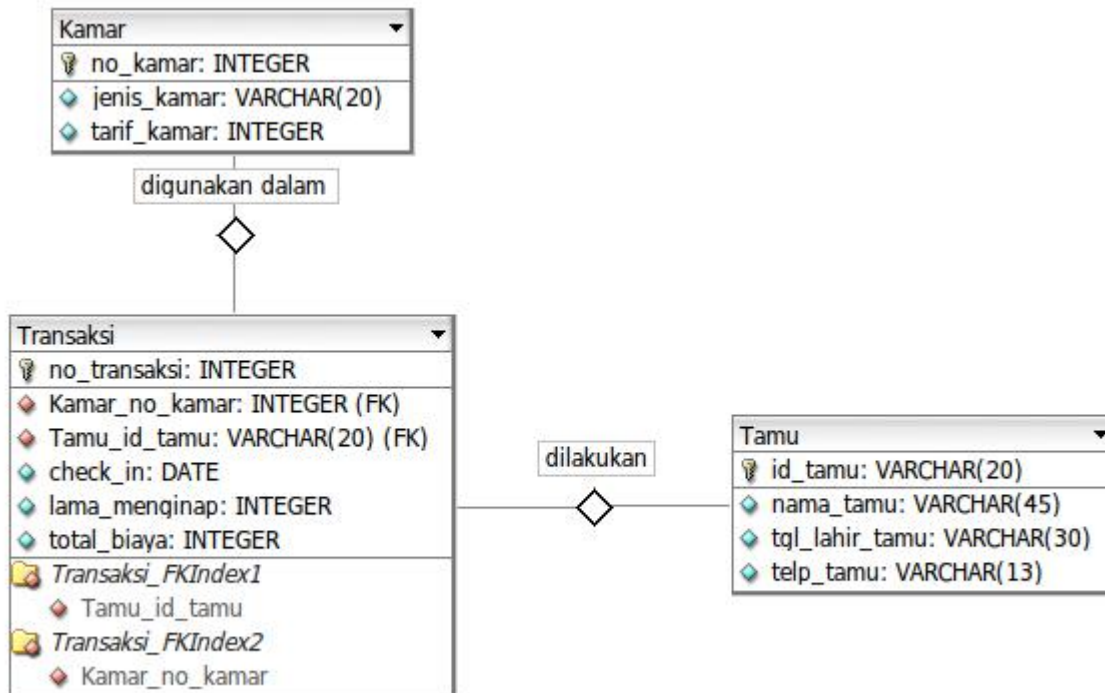


Laporan Praktikum Sistem Basis Data

NIM : L200180014
Nama : Andika Wirapala Fathin Azfar
Modul : 12

TUGAS

ER Diagram Perhotelan



Studi Kasus

Pada Hotel tersedia Kamar. Kamar memiliki nomor kamar dan jenis kamar. Terdapat dua jenis kamar: Standar dan VIP. Untuk kamar Standar tarif per malam senilai 300.000 rupiah, sedangkan VIP 500.000 rupiah.

Tamu yang ingin memesan kamar harus melakukan registrasi Tamu. Data yang diperlukan adalah nomor identitas (ID Tamu), nama, tanggal lahir, dan nomor telepon.

Setelah registrasi, langsung dilanjutkan transaksi. Pada setiap transaksi memiliki nomor transaksi yang nilainya otomatis bertambah. Data untuk transaksi mengambil id tamu dan nomor kamar (FOREIGN KEY). Untuk nomor kamar, tamu dapat terlebih dahulu memilih jenis kamar, setelah itu data nomor kamar akan muncul untuk jenis kamar yang dipilih sebelumnya. Data transaksi lain yang diperlukan adalah tanggal check in, lama menginap, dan total biaya yang otomatis dihitung dari tarif per kamar dikali lama menginap.

Penggunaan

File Project Netbeans ada di **Perhotelan.zip**. Ekstrak file tersebut.

Sebelum menjalankan aplikasi, terlebih dahulu membuat database awal. Nama database `Perhotelan`. Struktur sama dengan ER Diagram, atau langsung import `perhotelan.sql` dari folder Perhotelan.

File java dan form lengkap sebagai file mentah ada di folder *src/perhotelan*. Atau untuk menjalankan aplikasi secara instan dapat run **Perhotelan.jar** dalam folder *dist/*.

Tamu

View (SELECT)

```
32 private void tampilkanDataTamu() {
33     DefaultTableModel model = new DefaultTableModel();
34     model.addColumn("ID");
35     model.addColumn("Nama");
36     model.addColumn("Tgl. Lahir");
37     model.addColumn("No. Telepon");
38
39     try {
40         int no = 1;
41         String sql = "SELECT * FROM Tamu";
42         java.sql.Connection conn;
43         conn = (Connection)Koneksi.configDB();
44         java.sql.Statement stm=conn.createStatement();
45         java.sql.ResultSet res=stm.executeQuery(sql);
46         while(res.next()){
47             model.addRow(new Object[]{res.getString(1),res.getString(2),res.getString(3),res.getString(4)});
48         }
49         tabelTamu.setModel(model);
50     } catch (SQLException e) {
51         System.out.println("Error: " + e.getMessage());
52     }
53 }
```

INSERT

```
325 private void simpanTamuActionPerformed(java.awt.event.ActionEvent evt) {
326     // TODO add your handling code here:
327     try {
328         SimpleDateFormat sdf = new SimpleDateFormat("dd MMMM yyyy");
329         String date = sdf.format(tglLahir.getDate());
330         String sql = "INSERT INTO Tamu VALUES ('" + txtIdTamu.getText()+"', '" + txtNama.getText()+"', '" +
331             date+"', '" + telp.getText()+"'";
332         java.sql.Connection conn = (Connection)Koneksi.configDB();
333         java.sql.PreparedStatement pstmt = conn.prepareStatement(sql);
334         pstmt.execute();
335         JOptionPane.showMessageDialog(null, "Data berhasil disimpan");
336         tampilkanDataTamu();
337         kosongkan_form();
338     } catch (HeadlessException | SQLException e) {
339         JOptionPane.showMessageDialog(this, e.getMessage());
340     }
341 }
```

UPDATE

```
289 private void updateTamuActionPerformed(java.awt.event.ActionEvent evt) {
290     // TODO add your handling code here:
291     try {
292         SimpleDateFormat sdf = new SimpleDateFormat("dd MMMM yyyy");
293         String date = sdf.format(tglLahir.getDate());
294         String sql = "UPDATE Tamu SET id_tamu='"+txtIdTamu.getText()+"',nama_tamu='"+ txtNama.getText()+
295             "',tgl_lahir_tamu='"+date+"',telp_tamu='"+telp.getText()+
296             "'WHERE id_tamu='"+txtIdTamu.getText()+"'";
297         java.sql.Connection conn = (Connection)Koneksi.configDB();
298         java.sql.PreparedStatement pstmt = conn.prepareStatement(sql);
299         pstmt.execute();
300         JOptionPane.showMessageDialog(null, "Data berhasil diupdate");
301     } catch (HeadlessException | SQLException e) {
302         JOptionPane.showMessageDialog(this, e.getMessage());
303     }
304     tampilkanDataTamu();
305     kosongkan_form();
306 }
```


DELETE

```
362 private void hapusTamuActionPerformed(java.awt.event.ActionEvent evt) {  
363     // TODO add your handling code here:  
364     try {  
365         String sql = "DELETE FROM Transaksi WHERE id_tamu='"+ txtIdTamu.getText()+"'";  
366         java.sql.Connection conn = (Connection)Koneksi.configDB();  
367         java.sql.PreparedStatement pstmt = conn.prepareStatement(sql);  
368         pstmt.execute();  
369         JOptionPane.showMessageDialog(null, "Data berhasil dihapus");  
370     } catch (HeadlessException | SQLException e) {  
371         JOptionPane.showMessageDialog(this, e.getMessage());  
372     }  
373     tampilkanDataTamu();  
374     kosongkan_form();  
375 }
```

Kamar

SELECT

```
20 private void tampilkanDataKamar() {  
21     DefaultTableModel model = new DefaultTableModel();  
22     model.addColumn("No. Kamar");  
23     model.addColumn("Jenis");  
24     model.addColumn("Tarif /malam");  
25  
26     try {  
27         String sql = "SELECT * FROM Kamar";  
28         java.sql.Connection conn;  
29         conn = (Connection)Koneksi.configDB();  
30         java.sql.Statement stm=conn.createStatement();  
31         java.sql.ResultSet res=stm.executeQuery(sql);  
32         while(res.next()){  
33             model.addRow(new Object[]{res.getString(1),res.getString(2),res.getString(3)});  
34         }  
35         tabelKamar.setModel(model);  
36     } catch (SQLException e) {  
37         System.out.println("Error: " + e.getMessage());  
38     }  
39 }
```

INSERT

```
258 private void tbSimpanActionPerformed(java.awt.event.ActionEvent evt) {  
259     // TODO add your handling code here:  
260     try {  
261         String sql = "INSERT INTO Kamar VALUES ('" + noKamar.getText()+"', '" +  
262             cbJenisKamar.getSelectedItem()+"', '" +  
263             tarifKamar.getText() +"'";  
264         java.sql.Connection conn = (Connection)Koneksi.configDB();  
265         java.sql.PreparedStatement pstmt = conn.prepareStatement(sql);  
266         pstmt.execute();  
267         JOptionPane.showMessageDialog(null, "Data berhasil disimpan");  
268         tampilkanDataKamar();  
269         kosongkan_form();  
270     } catch (HeadlessException | SQLException e) {  
271         JOptionPane.showMessageDialog(this, e.getMessage());  
272     }  
273 }
```

UPDATE

```
275 private void tbUpdateActionPerformed(java.awt.event.ActionEvent evt) {  
276     // TODO add your handling code here:  
277     try {  
278         String sql = "UPDATE Kamar SET no_kamar='"+noKamar.getText() +  
279             "','jenis_kamar='"+cbJenisKamar.getSelectedItem()+  
280             "','tarif_kamar='"+tarifKamar.getText()+  
281             "'WHERE no_kamar='"+ noKamar.getText()+"'";  
282         java.sql.Connection conn = (Connection)Koneksi.configDB();  
283         java.sql.PreparedStatement pstmt = conn.prepareStatement(sql);  
284         pstmt.execute();  
285         JOptionPane.showMessageDialog(null, "Data berhasil diupdate");  
286     } catch (HeadlessException | SQLException e) {  
287         JOptionPane.showMessageDialog(this, e.getMessage());  
288     }  
289     tampilkanDataKamar();  
290     kosongkan_form();  
291 }
```

DELETE

```
293 private void tbHapusActionPerformed(java.awt.event.ActionEvent evt) {  
294     // TODO add your handling code here:  
295     try {  
296         String sql = "DELETE FROM Kamar WHERE no_kamar='"+ noKamar.getText()+ "'";  
297         java.sql.Connection conn = (Connection)Koneksi.configDB();  
298         java.sql.PreparedStatement pstmt = conn.prepareStatement(sql);  
299         pstmt.execute();  
300         JOptionPane.showMessageDialog(null, "Data berhasil dihapus");  
301     } catch (HeadlessException | SQLException e) {  
302         JOptionPane.showMessageDialog(this, e.getMessage());  
303     }  
304     tampilkanDataKamar();  
305     kosongkan_form();  
306 }
```

Transaksi

SELECT

```
120 private void tampil_tarifKamar() {  
121     String jenisKamar = cbJenisKamar.getSelectedItem().toString();  
122     try {  
123         Connection con = Koneksi.configDB();  
124         java.sql.Statement stt = con.createStatement();  
125         String sql = "SELECT tarif_kamar FROM Kamar WHERE jenis_kamar='"+jenisKamar+"'";  
126         java.sql.ResultSet res = stt.executeQuery(sql);  
127  
128         while(res.next()){  
129             Object[] ob = new Object[3];  
130             ob[0] = res.getString(1);  
131             txtTarif.setText((String) ob[0]);  
132         }  
133  
134         res.close(); stt.close();  
135     } catch (SQLException e) {  
136         System.out.println(e.getMessage());  
137     }  
138 }
```


INSERT

```
516 private void tbSimpanTransaksiActionPerformed(java.awt.event.ActionEvent evt) {  
517     // TODO add your handling code here:  
518     try {  
519         SimpleDateFormat sdf = new SimpleDateFormat("dd MMMM yyyy");  
520         String dateCheckIn = sdf.format(tglCheckIn.getDate());  
521         String sql = "INSERT INTO Transaksi VALUES (" +  
522             "NULL, '"+  
523             idTamu.getText()+"', '" +  
524             cbNoKamar.getSelectedItem().toString()+"', '" +  
525             dateCheckIn+"', '" +  
526             lamaMenginap.getValue()+"', '" +  
527             totalBiaya.getText()+"'";  
528         java.sql.Connection conn = (Connection)Koneksi.configDB();  
529         java.sql.PreparedStatement pstmt = conn.prepareStatement(sql);  
530         pstmt.execute();  
531         JOptionPane.showMessageDialog(null, "Data berhasil disimpan");  
532         tampilkanDataTransaksi();  
533         kosongkan_form();  
534     } catch (HeadlessException | SQLException e) {  
535         JOptionPane.showMessageDialog(this, e.getMessage());  
536     }  
537 }
```

UPDATE

```
539 private void tbUpdateActionPerformed(java.awt.event.ActionEvent evt) {  
540     // TODO add your handling code here:  
541     try {  
542         SimpleDateFormat sdf = new SimpleDateFormat("dd MMMM yyyy");  
543         String dateCheckIn = sdf.format(tglCheckIn.getDate());  
544         String sql = "UPDATE Transaksi SET id_tamu='"+idTamu.getText()+  
545             "', no_kamar='"+cbNoKamar.getSelectedItem().toString()+  
546             "', check_in='"+dateCheckIn+  
547             "', lama_menginap='"+lamaMenginap.getValue()+  
548             "', total_biaya='"+totalBiaya.getText()+  
549             "'WHERE id_tamu='"+idTamu.getText()+"'";  
550         java.sql.Connection conn = (Connection)Koneksi.configDB();  
551         java.sql.PreparedStatement pstmt = conn.prepareStatement(sql);  
552         pstmt.execute();  
553         JOptionPane.showMessageDialog(null, "Data berhasil diupdate");  
554     } catch (HeadlessException | SQLException e) {  
555         JOptionPane.showMessageDialog(this, e.getMessage());  
556     }  
557     tampilkanDataTransaksi();  
558     kosongkan_form();  
559 }
```

DELETE

```
585 private void tbHapusActionPerformed(java.awt.event.ActionEvent evt) {  
586     // TODO add your handling code here:  
587     try {  
588         String sql = "DELETE FROM Transaksi WHERE id_tamu='"+idTamu.getText()+"'";  
589         java.sql.Connection conn = (Connection)Koneksi.configDB();  
590         java.sql.PreparedStatement pstmt = conn.prepareStatement(sql);  
591         pstmt.execute();  
592         JOptionPane.showMessageDialog(null, "Data berhasil dihapus");  
593     } catch (HeadlessException | SQLException e) {  
594         JOptionPane.showMessageDialog(this, e.getMessage());  
595     }  
596     tampilkanDataTransaksi();  
597     kosongkan_form();  
598 }
```

Demo Aplikasi

Registrasi Tamu/INSERT

<2>

DATA TAMU

ID Tamu

12233378

Nama

Kiki Kukoki

Tgl. Lahir

01 June 2000

ID	Nama	Tgl. Lahir	No. Telepon
12345678	Andi	31 January 2000	081234567890
20180014	Dika WFA	30 December 2001	08987654321
98765432	Dodi Setiawan	09 June 1996	085000000000
99900012	Billy Situmeong	01 September 2020	082109876543

June

2000

Sun

Mon

Tue

Wed

Thu

Fri

Sat

23

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

Simpan

Update

Hapus

TRANSAKSI

Data Kamar

<2>

DATA TAMU

ID Tamu

12233378

Nama

Kiki Kukoki

Tgl. Lahir

01 June 2000

No. Telepon

087234990

ID	Nama	Tgl. Lahir	No. Telepon
12345678	Andi	31 January 2000	081234567890
20180014	Dika WFA	30 December 2001	08987654321
98765432	Dodi Setiawan	09 June 1996	085000000000
99900012	Billy Situmeong	01 September 2020	082109876543

Tambah

Simpan

Update

Hapus

TRANSAKSI

Data Kamar

Message

i

Data berhasil disimpan

OK

DATA TAMU

ID Tamu

12233378

Nama

Kiki Kukoki

Tgl. Lahir

01 June 2000

No. Telepon

087234990

ID	Nama	Tgl. Lahir	No. Telepon
12233378	Kiki Kukoki	01 June 2000	087234990
12345678	Andi	31 January 2000	081234567890
20180014	Dika WFA	30 December 2001	08987654321
98765432	Dodi Setiawan	09 June 1996	085000000000
99900012	Billy Situmeong	01 September 2020	082109876543

Tambah

Simpan

Update

Hapus

TRANSAKSI

Data Kamar

Update Tamu (Tgl Lahir Billy Sitimeong)

ID Tamu

99900012

Nama

Billy Situmeong

Tgl. Lahir

01 September 2000

No. Telepon

082109876543

ID	Nama	Tgl. Lahir	No. Telepon
12233378	Kiki Kukoki	01 June 2000	087234990
12345678	Andi	31 January 2000	081234567890
20180014	Dika WFA	30 December 2001	08987654321
98765432	Dodi Setiawan	09 June 1996	085000000000
99900012	Billy Situmeong	01 September 2020	082109876543

Tambah

Simpan

Update

Hapus

TRANSAKSI

Data Kamar

ID Tamu

99900012

Nama

Billy Situmeong

Tgl. Lahir

01 September 2000

No. Telepon

082109876543

ID	Nama	Tgl. Lahir	No. Telepon
12233378	Kiki Kukoki	01 June 2000	087234990
12345678	Andi	31 January 2000	081234567890
20180014	Dika WFA	30 December 2001	08987654321
98765432	Dodi Setiawan	09 June 1996	085000000000
99900012	Billy Situmeong	01 September 2000	082109876543

Tambah

Simpan

Update

Hapus

TRANSAKSI

Data Kamar

Hapus Tamu

ID Tamu

99900012

Nama

Billy Situmeong

Tgl. Lahir

01 September 2000

No. Telepon

082109876543

ID	Nama	Tgl. Lahir	No. Telepon
12233378	Kiki Kukoki	01 June 2000	087234990
12345678	Andi	31 January 20...	081234567890
20180014	Dika WFA	30 December...	08987654321
98765432	Dodi Setiawan	09 June 1996	085000000000
99900012	Billy Situmeong	01 Septembe...	082109876543

Tambah

Simpan

Update

Hapus

TRANSAKSI

Data Kamar

Message

i

Data berhasil dihapus

OK

ID Tamu

Nama

Tgl. Lahir

No. Telepon

ID	Nama	Tgl. Lahir	No. Telepon
12233378	Kiki Kukoki	01 June 2000	087234990
12345678	Andi	31 January 20...	081234567890
20180014	Dika WFA	30 December...	08987654321
98765432	Dodi Setiawan	09 June 1996	085000000000

Tambah

Simpan

Update

Hapus

TRANSAKSI

Data Kamar

Insert Kamar

DATA KAMAR

Nomor Kamar

301

Jenis Kamar

Standar

Tarif Kamar (/malam)

300000

No. Kamar	Jenis	Tarif /malam
101	Standar	300000
102	Standar	300000
103	Standar	300000
104	Standar	300000
111	VIP	500000
113	VIP	500000
201	Standar	300000
202	Standar	300000
203	Standar	300000
211	VIP	500000
212	VIP	500000
213	VIP	500000

Tambah Kamar

Simpan

Update

Hapus

Data Tamu

TRANSAKSI

DATA KAMAR

Nomor Kamar

301

Jenis Kamar

Standar

Tarif Kamar (/malam)

300000

No. Kamar	Jenis	Tarif /malam
101	Standar	300000
102	Standar	300000
103	Standar	300000
104	Standar	300000
111	VIP	500000
113	VIP	500000
201	Standar	300000
202	Standar	300000
203	Standar	300000
211	VIP	500000
212	VIP	500000
213	VIP	500000
301	Standar	300000

Tambah Kamar

Simpan

Update

Hapus

Data Tamu

TRANSAKSI

Update Kamar (301 -> VIP)

DATA KAMAR

Nomor Kamar

301

Jenis Kamar

VIP

Tarif Kamar (/malam)

300000

No. Kamar	Jenis	Tarif /malam
101	Standar	300000
102	Standar	300000
103	Standar	300000
104	Standar	300000
111	VIP	500000
113	VIP	500000
201	Standar	300000
202	Standar	300000
203	Standar	300000
211	VIP	500000
212	VIP	500000
213	VIP	500000
301	VIP	300000

Tambah Kamar

Simpan

Update

Hapus

Data Tamu

TRANSAKSI

Delete (Kamar 301)

DATA KAMAR

Nomor Kamar

Jenis Kamar

Tarif Kamar (/malam)

No. Kamar	Jenis	Tarif /malam
101	Standar	300000
102	Standar	300000
103	Standar	300000
104	Standar	300000
111	VIP	500000
113	VIP	500000
201	Standar	300000
202	Standar	300000
203	Standar	300000
211	VIP	500000
212	VIP	500000
213	VIP	500000

Tambah Kamar

Simpan

Update

Hapus

Data Tamu

TRANSAKSI

INSERT Transaksi

Memilih nama. Pilihan nama diambil otomatis dari database Tamu (ID tamu otomatis terisi)

TRANSAKSI

Nama Tamu

-- Pilih Nama --

-- Pilih Nama --

Kiki Kukoki

Andi

Dika WFA

Dodi Setiawan

Jenis Kamar

--

No. Kamar

--

Check In

Lama menginap

0

malam

Biaya (Rp)

0

Reset

Simpan

Update

Hapus

No. Transaksi	ID Tamu	No Kamar	Check In	Lama Menginap	Biaya
1	20180014	111	15 June 2020	3	1500000

Data Tamu

Data Kamar

Setelah memilih jenis kamar, otomatis pilihan nomor kamar menyesuaikan jenis kamar (Tarif kamar juga otomatis mengambil dari database). Selanjutnya memilih tanggal check in, dan lama menginap. Biaya dihitung otomatis dari lama menginap dan tarif kamar.

TRANSAKSI

Nama Tamu

ID Tamu

Jenis Kamar Tarif /malam

No. Kamar

Check In

Lama menginap malam

Biaya (Rp)

No. Transaksi	ID Tamu	No Kamar	Check In	Lama Menginap	Biaya
1	20180014	111	15 June 2020	3	1500000

No. Transaksi	ID Tamu	No Kamar	Check In	Lama Menginap	Biaya
1	20180014	111	15 June 2020	3	1500000
5	98765432	213	01 June 2020	2	1000000

Hapus Transaksi

TRANSAKSI

Nama Tamu

ID Tamu

Jenis Kamar

Tarif /malam

No. Kamar

Check In

Lama menginap

malam

Biaya (Rp)

Reset

Simpan

Update

Hapus

No. Transaksi	ID Tamu	No Kamar	Check In	Lama Menginap	Biaya
1	20180014	111	15 June 2020	3	1500000
5	98765432	213	01 June 2020	2	1000000

Data Tamu

Data Kamar

No. Transaksi	ID Tamu	No Kamar	Check In	Lama Menginap	Biaya
1	20180014	111	15 June 2020	3	1500000

Data Tamu

Data Kamar