

**LAPORAN**  
**Praktikum Ujian Akhir Semester**  
**Implementasi Penggunaan *Java Database Conectivity* pada Sistem**  
**Manajemen Hotel**



Disusun Oleh :

Akmal Nur Fauzan	(20220040060)
Annisa Tasya A.	(20220040274)
Selpia Meilani	(20220040148)

**UNIVERSITAS NUSA PUTRA**  
**JURUSAN TEKNIK INFORMATIKA**  
**JULI 2024**

## **Abstrak**

Salah satu industri paling dinamis dan kompetitif di dunia adalah perhotelan. Hotel menawarkan tempat tinggal sementara bagi wisatawan, pelancong bisnis, dan orang lain yang membutuhkan akomodasi. Dalam menjalankan operasinya, hotel harus mengelola berbagai aspek seperti reservasi, pencatatan data tamu, layanan kamar, dan lain-lain. Manajemen yang efektif ini dan efisien dari semua aspek ini sangat penting untuk memberikan pengalaman yang memuaskan bagi tamu dan memastikan operasional yang lancar serta menguntungkan bagi hotel. Dengan perkembangan teknologi informasi, banyak hotel telah beralih dari sistem manajemen manual ke sistem manajemen berbasis komputer yang lebih canggih. Salah satu komponen kunci dari sistem manajemen ini adalah basis data yang terintegrasi dan dirancang dengan baik untuk membantu hotel dalam mengelola data secara efisien, mengurangi kesalahan, dan meningkatkan produktivitas. Pada laporan ini, didapatkan hasil berupa sebuah *database* manajemen hotel yang berhasil kami rancang. *Database* manajemen hotel ini dibuat untuk memberikan solusi terintegrasi untuk semua aspek operasional hotel, dan diharapkan dapat meningkatkan efisiensi operasional, memberikan layanan yang lebih baik kepada tamu, dan mengelola sumber daya lebih efektif.

**Kata kunci :** Database, Manajemen Hotel

## ***Abstract***

One of the most dynamic and competitive industries in the world is hospitality. The hotel offers temporary accommodation for tourists, business travelers, and others who need accommodation. In carrying out its operations, the hotel has to manage various aspects such as reservations, recording of guest data, room service, and so on. This efficient and efficient management of all these aspects is crucial to providing a satisfactory experience for guests and ensuring smooth and profitable operations for the hotel. With the advancement of information technology, many hotels have shifted from manual management systems to more sophisticated computer-based management systems. One key component of this management system is an integrated and well-designed database to help hotels manage data efficiently, reduce errors, and increase productivity. In this report, we have the results of a database of hotel management that we have successfully designed. The hotel management database is designed to provide integrated solutions for all aspects of hotel operations, and is expected to improve operational efficiency, provide better service to guests, and manage resources more effectively.

***Keywords :*** *Database, Hotel Management*

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Dalam era globalisasi dan digitalisasi saat ini, industri perhotelan menghadapi tantangan dan peluang yang signifikan. Kemajuan teknologi informasi telah mengubah cara hotel mengelola operasional dan melayani tamu. Hotel-hotel modern kini beralih dari metode manajemen tradisional yang manual dan memakan waktu ke sistem manajemen berbasis komputer yang lebih efisien dan akurat.

Manajemen hotel yang efektif memerlukan pengelolaan berbagai aspek operasional, termasuk reservasi kamar, pencatatan data tamu, pengelolaan staf, inventaris, layanan kamar, dan pemrosesan pembayaran. Dengan volume data yang besar dan kebutuhan untuk mengakses informasi secara real-time, sistem manajemen manual menjadi kurang efisien dan rentan terhadap kesalahan. Kesalahan dalam pengelolaan data dapat berdampak negatif pada pengalaman tamu, efisiensi operasional, dan reputasi hotel.

Seiring dengan meningkatnya persaingan di industri perhotelan, kebutuhan akan sistem manajemen yang efisien dan andal semakin mendesak. Sistem manajemen database terintegrasi muncul sebagai solusi yang ideal untuk mengatasi tantangan ini. Sistem ini tidak hanya membantu dalam pengelolaan data yang lebih baik tetapi juga memungkinkan hotel untuk memberikan layanan yang lebih responsif dan personal kepada tamu mereka.

### **1.2. Tujuan**

- 1) Memahami apa itu *Database* dan *Java Database Connectivity*
- 2) Memahami cara membuat *Database* Manajemen Hotel menggunakan *Java Database Connectivity*

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Database**

Basis data atau *database* adalah kumpulan terorganisir dari data yang disimpan dan dikelola secara elektronik. *Database* dirancang untuk memungkinkan penyimpanan, pengambilan, dan pengelolaan data yang efisien dan terstruktur. Dalam dunia yang semakin digital ini, basis data memainkan peran penting dalam berbagai bidang, termasuk bisnis, pendidikan, pemerintahan, industri, dan teknologi. *Database* digunakan dalam berbagai aplikasi untuk mengelola data dengan cara yang efisien dan terorganisir. Mereka memungkinkan perusahaan dan organisasi untuk menyimpan data pelanggan, produk, transaksi, dan berbagai jenis data lainnya secara aman dan terstruktur.

Dalam konteks industri perhotelan, *database* memainkan peran penting dalam mengelola berbagai aspek operasional, seperti reservasi, data tamu, dan layanan kamar. Dengan menggunakan sistem manajemen *database* yang efisien, hotel dapat meningkatkan efisiensi operasional, mengurangi kesalahan, dan memberikan layanan yang lebih baik kepada tamu.

#### **2.2. MySQL**

MySQL merupakan software yang tergolong database server dan bersifat *Open Source*. *Open Source* menyatakan bahwa *software* ini dilengkapi dengan *source code* (kode yang dipakai untuk membuat MySQL), selain tentu saja bentuk *executable*-nya atau kode yang dapat dijalankan secara langsung dalam sistem operasi, dan bisa diperoleh dengan cara mengunduh di internet secara gratis.

#### **2.3. Java Database Connectivity**

*Java Database Connectivity* (JDBC) adalah API (*Application Programming Interface*) yang menyediakan cara standar untuk berinteraksi dengan *database* dari program *java* dan membantu aplikasi *java* untuk mengeksekusi *SQL statement*. JDBC merupakan interface pemrograman aplikasi yang mendefinisikan bagaimana pemrograman *java* dapat mengakses *database* dalam format tabular dari kode-kode *java* menggunakan sekumpulan *interface standard* dan *clas-class* yang tertulis dalam bahasa *java*. JDBC memungkinkan aplikasi *java* untuk melakukan operasi seperti menyimpan, mengambil, memperbarui, dan menghapus data dari *database* yang

terhubung. JDBC merupakan bagian penting dari pengembangan aplikasi *java* yang berhubungan dengan *database*, dan memahami cara kerjanya sangatlah penting dalam membangun aplikasi yang terhubung dengan database secara efisien dan aman.

## **2.4. Manajemen Hotel**

Manajemen hotel mencakup berbagai aktivitas dan tanggungjawab yang diperlukan untuk mengelola operasi sehari-hari sebuah properti perhotelan. Ini meliputi pengawasan atas berbagai aspek seperti pelayanan tamu, operasional kamar, keuangan, pemasaran, sumber daya manusia, dan pemeliharaan fasilitas. Tujuan utama manajemen hotel adalah untuk menyediakan pengalaman menginap yang menyenangkan dan memuaskan bagi tamu, sambil menjaga keberlanjutan dan profitabilitas bisnis.

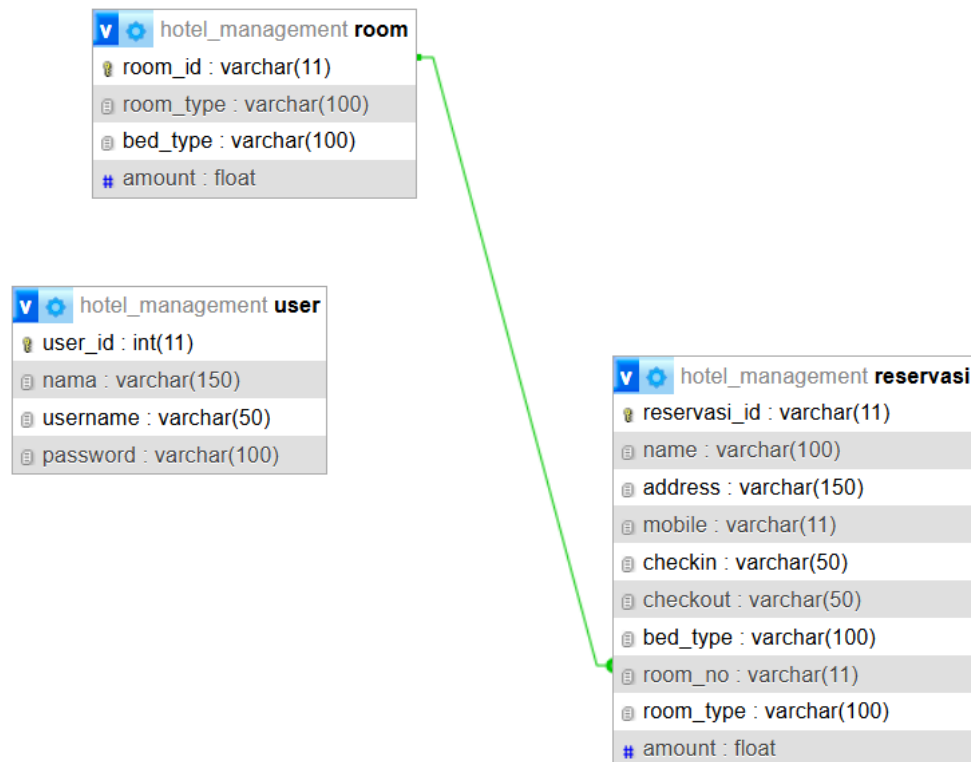
Dalam manajemen hotel, JDBC berperan sebagai jembatan antara program java yang mengelola hotel dan database yang menyimpan informasi hotel. Dengan JDBC, developer dapat membangun sistem manajemen hotel yang efisien dan terintegrasi. Dan hotel dapat mengelola data secara efektif, meningkatkan akurasi, dan membuat proses bisnis lebih efisien.

### **BAB III**

### **PEMBAHASAN**

Dalam industri perhotelan yang semakin berkembang, manajemen reservasi yang efisien menjadi faktor kunci untuk memastikan operasi hotel berjalan lancar dan pelanggan mendapatkan pengalaman yang memuaskan. Salah satu alat yang sangat penting untuk mencapai efisiensi ini adalah penggunaan *database* dalam manajemen reservasi hotel. *Database* memungkinkan penyimpanan, pengelolaan, dan pemrosesan informasi reservasi dengan cara yang terstruktur dan sistematis. Dengan teknologi *database*, hotel dapat mengoptimalkan operasional, mengurangi kesalahan, dan meningkatkan layanan pelanggan. Implementasi JDBC dalam manajemen reservasi hotel dapat menunjukkan berbagai hasil positif dalam meningkatkan efisiensi operasional dan kualitas layanan hotel. Berikut adalah beberapa hasil dari proyek yang kami rancang :

#### **1) Class Diagram**



Gambar 3.1 Class Diagram

Gambar di atas adalah desain *class diagram* yang kami rancang untuk database manajemen hotel. *Class diagram* ini menggambarkan entitas inti dan hubungannya dalam sistem manajemen hotel. Hal ini memungkinkan pengelolaan kamar hotel, *user*, dan reservasi. Sistem dapat melacak ketersediaan kamar, informasi pengguna, detail reservasi, dan menetapkan kamar ke reservasi. Berikut hasil rancangan tabel *database* manajemen hotel, dan *source code* yang digunakan untuk mengkoneksikan java dengan MySQL *database*.

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> reservasi		2	InnoDB	utf8mb4_general_ci	32.0 KiB	-
<input type="checkbox"/> room		2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> user		1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
3 tables	Sum	5	InnoDB	utf8mb4_general_ci	64.0 KiB	0 B

Gambar 3.2 Database Manajemen Hotel



```

public void Connect() {
    try {
        Class.forName("com.mysql.jdbc.Driver");
        con = DriverManager.getConnection("jdbc:mysql://localhost/hotel_management", "root", "");
        if (con != null) {
            System.out.println("Connected to the database!");
        } else {
            System.out.println("Failed to connect to the database!");
        }
    } catch (ClassNotFoundException | SQLException ex) {
        Logger.getLogger(room.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

Gambar 3.3 Source Code Koneksi MySQL Database

## 2) Tampilan Login

*Table* login memiliki peran yang sangat penting untuk mengelola akses pengguna ke sistem, dan digunakan untuk menyimpan informasi kredensial *user*. *Table* ini memastikan bahwa hanya pengguna sah yang dapat mengakses dan mengelola data dalam sistem manajemen hotel.



The image shows a graphical user interface for a login system. It consists of a light blue rectangular window with a title bar at the top containing a small icon and standard window controls (minimize, maximize, close). The main area of the window has a light gray background. At the top center, the word "LOGIN" is displayed in a large, bold, black sans-serif font. Below this, there is a white rectangular box with a thin gray border. Inside this box, the label "Username" is positioned to the left of a white text input field with a blue border. Below the "Username" field, the label "Password" is positioned to the left of another white text input field with a blue border. At the bottom of the white box, there is a wide, light gray button with rounded corners and a subtle gradient, containing the word "Login" in a bold, black sans-serif font.

Gambar 3.4 Tampilan Login

Source code yang digunakan adalah sebagai berikut :

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String username = txt_username.getText().trim();
    String password = txt_pass.getText().trim();
    try{
        String sql = "select * from user where username=? and password =? ";
        Statement s = con.createStatement();
        pst = con.prepareStatement(sql);
        pst.setString(1, username);
        pst.setString(2, password);
        ResultSet rs = pst.executeQuery();

        if (username.isEmpty() || password.isEmpty()) {
            JOptionPane.showMessageDialog(this, "Username or password cannot be empty", "Error", JOptionPane.ERROR_MESSAGE);
            return;
        }

        if(rs.next()){
            if(txt_username.getText().equals(rs.getString("username")) && txt_pass.getText().equals(rs.getString("password"))){
                JOptionPane.showMessageDialog(this, "Berhasil Login");
                this.setVisible(false);
                Main m = new Main();
                m.setVisible(true);
            }else{
                JOptionPane.showMessageDialog(this, "Invalid username or password", "Error", JOptionPane.ERROR_MESSAGE);
            }
        }
    }catch (Exception e){
        e.printStackTrace();
        JOptionPane.showMessageDialog(this, "An error occurred while trying to log in", "Error", JOptionPane.ERROR_MESSAGE);
    }
}

```

Gambar 3.5 Source Code Login

### 3) Table Reservasi

Halaman ini berisi data pelanggan yang telah melakukan registrasi yang dilihat dari sisi admin.

RID	Nama	Alamat	Telepon	CheckIn	CheckOut	Tipe Kamar	No. Kamar	Tipe Ranjang	Jumlah
RE002	bambang	Cisarua	225598	2024-06-04	2024-06-06	Non AC	K0007	Double	120000
RE003	Akmal	Sukabumi	0895	2024-06-04	2024-06-06	AC	K0006	Single	150000

Gambar 3.6 Menu Reservasi

Jika admin menambahkan data baru, maka hasilnya seperti berikut :

**Reservasi**

No. Reservasi: **RE003**

Nama: Akmal

Alamat: Sukabumi

Telepon: 0895

Check In: Jun 4, 2024

Check Out: Jun 6, 2024

Tipe Kamar: AC

No. Kamar: K0006

Tipe Tempat Tidur: Single

Jumlah: 150000

RID	Nama	Alamat	Telepon	CheckIn	CheckOut	Tipe Kamar	No. Kamar	Tipe Ranjang	Jumlah
RE002	bambang	Cisarua	1241415	2024-08-05	2024-08-14	Non AC	K0003	Single	210000

Message: Reservasi Telah Ditambahkan!

Buttons: Save, Edit, Delete, Clear, Close

Gambar 3.6 Menambahkan Data Reservasi

Jika admin mengedit data reservasi, maka hasilnya seperti berikut :

**Reservasi**

No. Reservasi: **RE002**

Nama: bambang

Alamat: Cisarua

Telepon: 225598

Check In: Jun 4, 2024

Check Out: Jun 6, 2024

Tipe Kamar: Non AC

No. Kamar: K0007

Tipe Tempat Tidur: Double

Jumlah: 120000

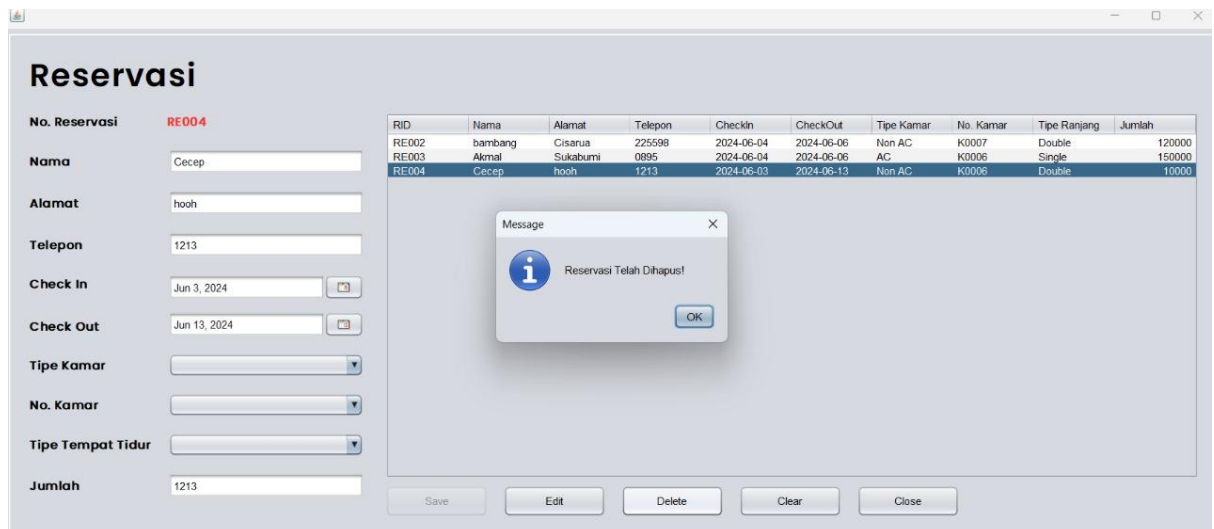
RID	Nama	Alamat	Telepon	CheckIn	CheckOut	Tipe Kamar	No. Kamar	Tipe Ranjang	Jumlah
RE002	bambang	Cisarua	1241415	2024-08-05	2024-08-14	Non AC	K0003	Single	210000
RE003	Akmal	Sukabumi	0895	2024-06-04	2024-06-06	AC	K0006	Single	150000

Message: Reservasi Telah Diubah!

Buttons: Save, Edit, Delete, Clear, Close

Gambar 3.7 Mengedit Data Reservasi

Jika admin menghapus data reservasi, maka hasilnya seperti berikut :



Gambar 3.8 Menghapus Data Reservasi

*Source code* yang digunakan adalah *class* “*Load\_reservasi*”, yang digunakan untuk memperlihatkan seluruh isi *table* reservasi melalui *database* sehingga bisa dilihat melalui *table* yang ada pada aplikasi.

```
public void Load_reservasi() {
    int c;

    try {
        pst = con.prepareStatement("select * from reservasi");
        ResultSet rs = pst.executeQuery();

        ResultSetMetaData rsd = rs.getMetaData();
        c = rsd.getColumnCount();

        d = (DefaultTableModel)jTable1.getModel();
        d.setRowCount(0);

        while(rs.next()){
            Vector v2 = new Vector();

            for(int i = 1; i<=c; i++){
                v2.add(rs.getString("reservasi_id"));
                v2.add(rs.getString("name"));
                v2.add(rs.getString("address"));
                v2.add(rs.getString("mobile"));
                v2.add(rs.getString("checkin"));
                v2.add(rs.getString("checkout"));
                v2.add(rs.getString("bed_type"));
                v2.add(rs.getString("room_no"));
                v2.add(rs.getString("room_type"));
                v2.add(rs.getString("amount"));
            }
            d.addRow(v2);
        }

    } catch (SQLException ex) {
        Logger.getLogger(room.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

Gambar 3.9 Source Code Reservasi

#### 4) *Table Room*

Halaman ini berisi data kamar yang tersedia dengan berbagai fasilitas yang ditawarkan kepada pelanggan.

No. Kamar	Tipe Kamar	Tipe Tempat Tidur	Jumlah
K0006	Non AC	Double	200000
K0007	AC	Single	450000

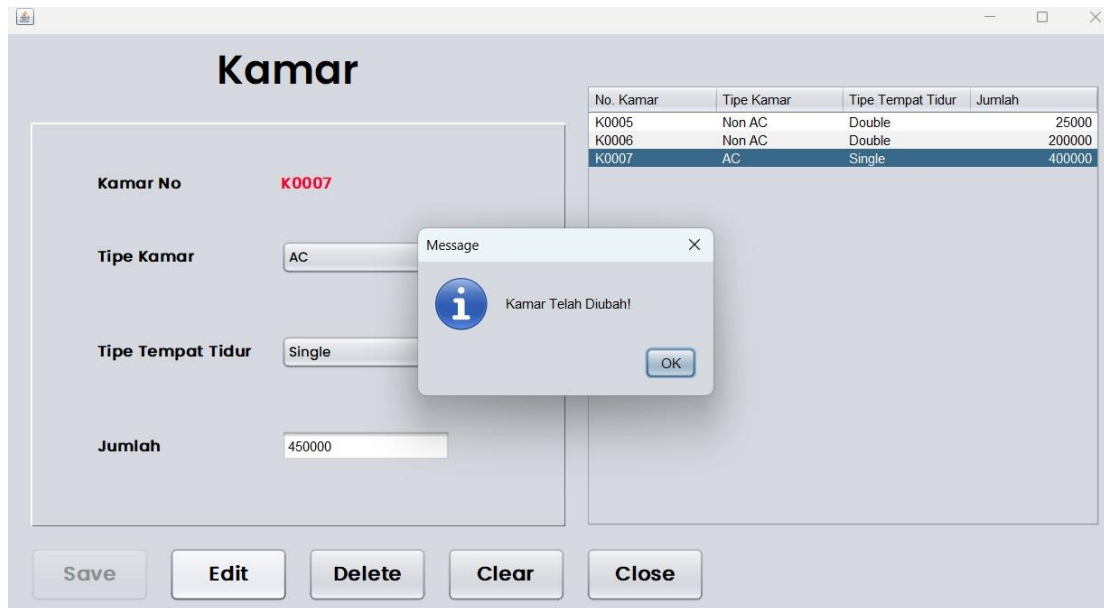
Gambar 3.10 Menu *Room* atau Kamar

Jika admin menambahkan data baru, maka hasilnya seperti berikut :

No. Kamar	Tipe Kamar	Tipe Tempat Tidur	Jumlah
K0005	Non AC	Double	25000
K0006	Non AC	Double	200000

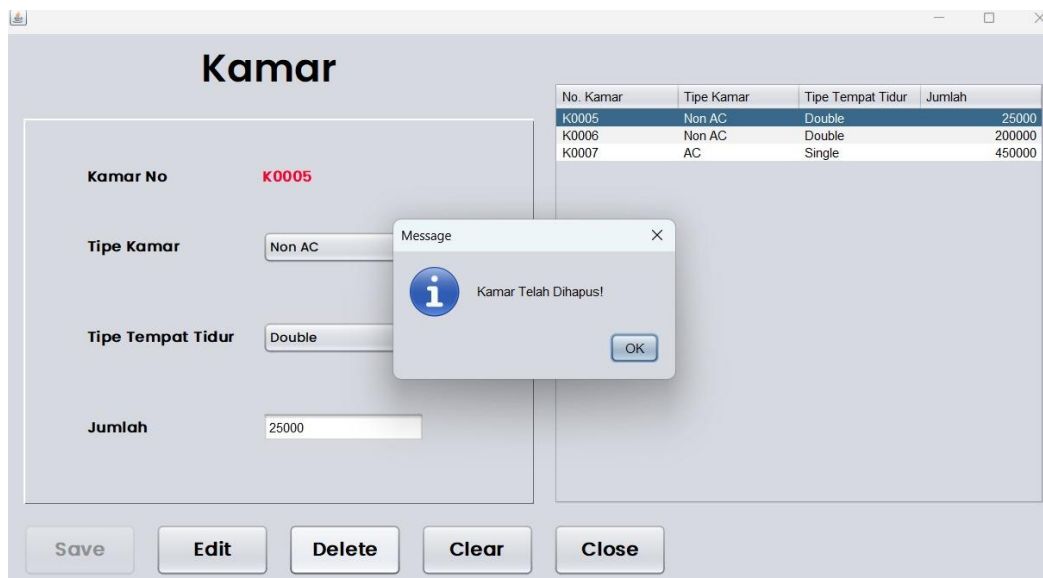
Gambar 3.11 Menambahkan Data Kamar

Jika admin mengedit data baru kamar, maka hasilnya seperti berikut :



Gambar 3.12 Mengedit Data Kamar

Jika admin menghapus data kamar, maka hasilnya seperti berikut :



Gambar 3.13 Menghapus Data Kamar

Untuk mengambil data dari sebuah *table room* ke *table reservasi*, menggunakan *source code* berikut :

```

public void RoomTypeL() {
    try {
        pst = con.prepareStatement("select Distinct room_type from room");
        ResultSet rs = pst.executeQuery();
        txt_rtype.removeAllItems();

        while(rs.next()){
            txt_rtype.addItem(rs.getString("room_type"));
        }
    } catch (SQLException ex) {
        Logger.getLogger(reservation.class.getName()).log(Level.SEVERE, null, ex);
    }
}

public void RoomNo() {
    try {
        pst = con.prepareStatement("select Distinct room_id from room");
        ResultSet rs = pst.executeQuery();
        txt_ro.removeAllItems();

        while(rs.next()){
            txt_ro.addItem(rs.getString("room_id"));
        }
    } catch (SQLException ex) {
        Logger.getLogger(reservation.class.getName()).log(Level.SEVERE, null, ex);
    }
}

public void BedTypeL() {
    try {
        pst = con.prepareStatement("select Distinct bed_type from room");
        ResultSet rs = pst.executeQuery();
        txt_btype.removeAllItems();

        while(rs.next()){
            txt_btype.addItem(rs.getString("bed_type"));
        }
    } catch (SQLException ex) {
        Logger.getLogger(reservation.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

Gambar 3.14 *Table Room* ke *Table Reservasi*

## BAB V

### KESIMPULAN

Database memungkinkan penyimpanan data reservasi, informasi tamu, dan detail kamar secara terstruktur dan sistematis. Dengan database, hotel dapat mengelola operasi sehari-hari dengan lebih efisien, mengurangi kesalahan manual, dan meningkatkan kecepatan layanan. Penggunaan JDBC memungkinkan aplikasi Java berinteraksi dengan database, melakukan operasi seperti menyimpan, mengambil, memperbarui, dan menghapus data. Dengan JDBC, sistem manajemen hotel dapat terintegrasi dengan baik, memungkinkan berbagai operasi dilakukan secara otomatis dan efisien.

Secara keseluruhan, penggunaan database dalam manajemen reservasi hotel memberikan banyak manfaat, mulai dari efisiensi operasional hingga peningkatan kualitas layanan kepada pelanggan. Implementasi JDBC sebagai penghubung antara aplikasi Java dan database memperkuat kemampuan sistem dalam mengelola data secara efektif, menjadikan proses bisnis lebih lancar dan terorganisir.

## **DAFTAR PUSTAKA**

Agustin, Maria. (2012). “Analisis dan Perancangan Sistem Informasi Reservasi Hotel”, Skripsi, Palembang : Politeknik Negeri Sriwijaya.

Abdul Kadir (2009). “Membuat Aplikasi Web dengan PHP + Database MySQL”, Yogyakarta: Andi Offset

Suartana, ”Sistem informasi perhotelan”, Andi, Yogyakarta, (2004).

<https://www.scribd.com/document/692276825/HASIL-LAPORAN-RESERVASI-HOTEL>

<https://www.scribd.com/document/694355605/Laporan-Tubes-PBO-Kelompok-Ardi>