

**Laporan Rancangan Project Tugas Besar**  
**PEMROGRAMAN BERIORENTASI OBJEK II**  
**FEEDBACK UMKM NASI GORENG**

Dosen pengajar:  
RUDHI WAHYUDI FEBRIANTO



Disusun Oleh:

Ade Hikmat Pauji Ridwan 22552011130

Juwita Kristina T.oinan 21552011209

Parid Hadiansyah 21552011116

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**UNIVERSITAS TEKNOLOGI BANDUNG**

**2024**

<b>BAB I PENDAHULUAN.....</b>	<b>2</b>
<b>1.1 Latar Belakang .....</b>	<b>2</b>
<b>1.2 Rumusan Masalah.....</b>	<b>3</b>
<b>1.3 Batasan Masalah .....</b>	<b>4</b>
<b>1.4 Landasan Teori.....</b>	<b>4</b>
<b>BAB II METODOLOGI PENELITIAN.....</b>	<b>5</b>
<b>2.1 Metodologi.....</b>	<b>5</b>
<b>2.2 Sumber Data .....</b>	<b>5</b>
<b>2.3 Teknik Analisa.....</b>	<b>6</b>
<b>BAB III PEMBAHASAN .....</b>	<b>7</b>
<b>3.1 Pembahasan .....</b>	<b>7</b>
<b>3.2 Rancangan Relasi Database .....</b>	<b>8</b>
<b>3.3 Rancangan User Interface .....</b>	<b>9</b>
<b>3.4 Sourcode.....</b>	<b>11</b>
<b>BAB IV PENUTUP.....</b>	<b>13</b>
<b>4.1 Kesimpulan .....</b>	<b>13</b>
<b>4.2 Saran.....</b>	<b>13</b>

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Nasi goreng adalah salah satu hidangan paling ikonik dalam masakan Asia Tenggara, khususnya di Indonesia. Hidangan ini dikenal luas karena rasa yang kaya, kemudahan dalam penyajian, dan fleksibilitas dalam bahan-bahannya. Nasi goreng secara harfiah berarti “nasi yang digoreng” dan umumnya terdiri dari nasi yang dimasak dengan bumbu-bumbu seperti kecap manis, bawang putih, dan cabai, serta ditambah dengan berbagai bahan seperti daging, sayuran, dan telur..

Dalam era digital yang serba cepat, pelanggan memiliki banyak saluran untuk menyampaikan opini mereka, baik positif maupun negatif. Platform online seperti situs web, media sosial, dan aplikasi ulasan memberikan kesempatan kepada pelanggan untuk berbagi pengalaman mereka dengan cepat dan mudah. Oleh karena itu, mengumpulkan umpan balik secara teratur melalui kanal-kanal ini tidak hanya membantu dalam memahami kepuasan pelanggan, tetapi juga memberikan kesempatan untuk menanggapi dan memperbaiki kekurangan dengan segera. Pentingnya umpan balik pelanggan juga terletak pada kemampuannya untuk memberikan data yang objektif dan berbasis fakta. Ini membantu manajemen dalam mengambil keputusan yang lebih terinformasi dan strategis. Selain itu, umpan balik yang dikumpulkan dapat berfungsi sebagai alat promosi yang efektif jika pelanggan memberikan ulasan positif, yang pada gilirannya dapat menarik lebih banyak pelanggan baru..

Tujuan utama dari proyek ini adalah untuk mengembangkan aplikasi web yang mampu mengelola data feedback pelanggan dengan mudah dan efisien. Selain itu, proyek ini juga bertujuan untuk meningkatkan keterampilan dalam pengembangan aplikasi web menggunakan Java . Aplikasi ini akan melibatkan pembuatan sistem formulir feedback pelanggan, penyimpanan dan manajemen data feedback, dasbor admin untuk analisis data, serta notifikasi email kepada administrator. Dengan adanya sistem ini, diharapkan perusahaan dapat mengumpulkan dan mengelola feedback pelanggan secara lebih terstruktur dan efisien. Hal ini tidak hanya akan meningkatkan kualitas layanan dan produk yang ditawarkan, tetapi juga akan memberikan kontribusi positif dalam pengembangan teknologi informasi, khususnya dalam pengelolaan data berbasis web.

## 1.2 Rumusan Masalah

- Bagaimana merancang dan mengembangkan aplikasi web berbasis PHP yang efisien untuk mengelola data feedback pelanggan?
- Bagaimana cara menampilkan, menambah, mengedit, dan menghapus data feedback pelanggan secara efektif dalam aplikasi web ini?
- Bagaimana cara menyimpan data feedback pelanggan dengan format yang sederhana namun efisien?

## 1.3 Batasan Masalah

### 1. Rancangan dan Pengembangan Aplikasi Web Berbasis PHP:

- o Pengembangan aplikasi dibatasi pada penggunaan PHP dengan framework java .
- o Aplikasi hanya akan dibangun dengan fungsi dasar tanpa integrasi dengan platform eksternal lainnya kecuali disebutkan.
- o Aplikasi dioptimalkan untuk penggunaan pada browser desktop; tampilan dan performa pada perangkat mobile tidak menjadi fokus utama.

### 2. Menampilkan, Menambah, Mengedit, dan Menghapus Data Feedback Pelanggan:

- o Fitur CRUD hanya mencakup operasi dasar: penambahan, tampilan, pengeditan, dan penghapusan data feedback pelanggan.
- o Tampilan data feedback pelanggan akan menggunakan tabel sederhana tanpa visualisasi data yang kompleks seperti grafik atau diagram.
- o Fungsi pengeditan dan penghapusan hanya tersedia untuk pengguna dengan hak akses administrator.
- o Validasi input formulir terbatas pada pemeriksaan kesalahan dasar seperti format teks dan panjang karakter.vv

### 3. Menyimpan Data Feedback Pelanggan dengan Format yang Sederhana namun Efisien:

- o Penyimpanan data menggunakan database relasional seperti MySQL atau PostgreSQL.
- o Data yang disimpan meliputi ID, nama pelanggan, tanggal feedback, teks feedback, dan rating.
- o Tidak ada penyimpanan data multimedia seperti gambar atau video dalam database feedback pelanggan. o Optimalisasi penyimpanan dan akses data dibatasi pada penggunaan indeks dan normalisasi tabel.

## 1.3 Landasan Teori

### 1. PHP (*Hypertext Preprocessor*)

PHP adalah bahasa pemrograman server-side yang digunakan untuk mengembangkan aplikasi web dinamis. PHP memungkinkan pembuatan halaman web yang interaktif dengan mengintegrasikan logika bisnis dan akses basis data. Dalam proyek pengelolaan

data *feedback* pelanggan, PHP digunakan sebagai bahasa pemrograman utama untuk membangun fitur-fitur aplikasi seperti formulir *feedback*, manajemen data, dan dasbor admin. *Framework Laravel* yang berbasis PHP menyediakan alat dan struktur untuk mempermudah pengembangan aplikasi web.

2. MySQL (*My Structured Query Language*)

MySQL adalah sistem manajemen basis data relasional yang menggunakan SQL (*Structured Query Language*) untuk mengelola dan mengakses data yang disimpan dalam basis data. MySQL digunakan dalam proyek ini untuk menyimpan data *feedback* pelanggan secara terstruktur dan efisien. Database relasional ini memungkinkan penyimpanan data yang mudah diakses dan diolah, mendukung fitur CRUD (*Create, Read, Update, Delete*) dalam aplikasi.

3. HTML (*Hypertext Markup Language*)

HTML adalah bahasa markup standar untuk membuat halaman web. HTML digunakan untuk membangun struktur dasar konten di web. Dalam aplikasi pengelolaan data *feedback* pelanggan, HTML digunakan untuk membuat antarmuka pengguna seperti formulir input *feedback* dan tampilan daftar *feedback*.

4. CSS (*Cascading Style Sheets*)

CSS adalah bahasa *stylesheet* yang digunakan untuk mendesain dan memperindah tampilan halaman web. CSS memungkinkan pemisahan antara konten dan presentasi. CSS digunakan dalam proyek ini untuk mendesain antarmuka pengguna yang responsif dan menarik, seperti formulir *feedback* dan dasbor admin.

5. JavaScript

JavaScript adalah bahasa pemrograman yang digunakan untuk membuat halaman web lebih interaktif dan dinamis. JavaScript dapat digunakan untuk memanipulasi elemen HTML dan berinteraksi dengan server tanpa harus memuat ulang halaman. JavaScript digunakan untuk menambah interaktivitas dalam aplikasi, seperti validasi formulir *feedback* dan fitur-fitur dinamis pada dasbor admin.

6. SMTP (*Simple Mail Transfer Protocol*)

SMTP adalah protokol standar untuk pengiriman email melalui internet. SMTP digunakan untuk mengirim pesan email dari klien email ke server email atau antara server email. Dalam aplikasi ini, SMTP digunakan untuk mengirim notifikasi email kepada administrator setiap kali ada *feedback* baru yang masuk, memastikan bahwa administrator dapat segera menanggapi umpan balik pelanggan.

## **BAB II**

### **METODOLOGI PENELITIAN**

#### **2.1 Metodologi**

Metodologi yang digunakan dalam proyek ini adalah pendekatan berbasis proyek dengan langkah-langkah sebagai berikut:

- Analisis Kebutuhan: Mengidentifikasi kebutuhan pengguna dan fungsi yang harus ada dalam aplikasi.
- Perancangan: Membuat desain sistem, termasuk struktur tabel dan antarmuka pengguna.
- Implementasi: Mengkodekan aplikasi
- Pengujian: Menguji aplikasi untuk memastikan semua fitur berfungsi dengan baik.
- Pemeliharaan: Melakukan perbaikan dan pembaruan berdasarkan umpan balik pengguna.

#### **2.2 Sumber Data**

Data yang digunakan dalam aplikasi ini berasal dari feedback pelanggan yang dikumpulkan melalui formulir online. Aplikasi ini mengelola data feedback pelanggan dengan cara berikut:

##### **1. Formulir Feedback Pelanggan:**

- Pengumpulan Data: Data dikumpulkan melalui formulir online yang diisi oleh pelanggan. Formulir ini mencakup beberapa kolom penting, seperti:
  - Nama Pelanggan: Nama pelanggan yang memberikan feedback.
  - Tanggal Feedback: Tanggal ketika feedback diberikan.
  - Feedback: Teks umpan balik yang diberikan oleh pelanggan.
  - Rating: Penilaian yang diberikan oleh pelanggan dalam bentuk angka (misalnya, skala 1-5).

##### **2. Penyimpanan Data:**

Data yang dikumpulkan melalui formulir disimpan dalam basis data relasional seperti MySQL atau PostgreSQL. Basis data ini dirancang untuk menyimpan data secara terstruktur dan efisien, memungkinkan akses dan pengelolaan data yang mudah.

##### **3. Struktur Tabel Basis Data:**

- Berikut adalah contoh struktur tabel untuk menyimpan data feedback pelanggan:

```
4. CREATE TABLE users (  
5.   id SERIAL PRIMARY KEY,  
6.   username VARCHAR(50) NOT NULL UNIQUE,  
7.   password VARCHAR(255) NOT NULL,
```

```
8.     created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
9. );
```

#### 10. Pengelolaan Data:

- **Antarmuka Pengguna:** Aplikasi web menyediakan antarmuka pengguna yang intuitif untuk menampilkan, menambah, mengedit, dan menghapus data feedback pelanggan. Administrator dapat dengan mudah mengelola data feedback melalui antarmuka ini.
- **Analisis Data:** Data yang disimpan dapat dianalisis untuk mendapatkan wawasan berharga mengenai kualitas layanan dan produk. Analisis ini mendukung pengambilan keputusan bisnis yang lebih baik.

### 2.3 Teknik Analisa

Analisis dilakukan dengan mengevaluasi fungsionalitas aplikasi dan kinerja sistem dalam mengelola data kendala . Pengujian dilakukan dengan skenario uji untuk memastikan aplikasi bekerja sesuai dengan yang diharapkan tanpa adanya kendala yang diinginkan sebelum pemakaian.

## **BAB III PEMBAHASAN**

### **3.1 Pembahasan**

#### **1. Pengumpulan Data Feedback Pelanggan**

Data feedback dikumpulkan melalui formulir online yang mudah diisi oleh pelanggan. Formulir ini mencakup informasi seperti nama pelanggan, tanggal feedback, teks feedback, dan rating. Hal ini memastikan bahwa data yang diperoleh lengkap dan relevan untuk analisis lebih lanjut.

#### **2. Penyimpanan dan Manajemen Data**

Data feedback yang dikumpulkan disimpan dalam basis data relasional seperti MySQL atau PostgreSQL. Struktur tabel yang dirancang secara efisien memungkinkan penyimpanan data yang terorganisir dan mudah diakses. Setiap entri feedback dilengkapi dengan informasi penting yang disimpan dalam kolom-kolom tabel basis data. Dalam implementasinya, Laravel Eloquent ORM digunakan untuk memudahkan interaksi dengan basis data, sehingga pengelolaan data menjadi lebih efisien dan kodingan menjadi lebih rapi.

#### **3. Fitur CRUD (Create, Read, Update, Delete)**

Pelanggan dapat menambah feedback mereka melalui formulir yang tersedia di halaman web. Data yang dimasukkan akan disimpan langsung ke dalam basis data. Data feedback yang telah dikumpulkan dapat ditampilkan dalam bentuk tabel di dasbor admin.

Administrator dapat melihat dan menilai feedback yang diberikan oleh pelanggan.

Administrator dapat mengedit data feedback yang telah disimpan jika terdapat kesalahan atau perlu perbaikan. Administrator juga memiliki kemampuan untuk menghapus data feedback yang tidak lagi relevan atau dibutuhkan.

#### **4. Notifikasi Email**

Setiap kali feedback baru masuk, aplikasi akan mengirimkan notifikasi email kepada administrator. Hal ini memastikan bahwa umpan balik pelanggan dapat segera ditanggapi, meningkatkan responsivitas dan kepuasan pelanggan.

### **3.2 Rancangan Relasi Database**

Untuk mengelola data feedback pelanggan dengan efektif, basis data relasional digunakan. Rancangan database mencakup tabel utama yang menyimpan informasi feedback pelanggan. Berikut adalah struktur detail dari tabel yang digunakan dalam aplikasi ini:



## **Tabel Feedback**

```
CREATE TABLE feedbacks (  
    id SERIAL PRIMARY KEY,  
    customer_name VARCHAR(255) NOT NULL,  
    feedback_date TIMESTAMP NOT NULL,  
    feedback_text TEXT,  
    rating INT  
);
```

**Nama Tabel:** feedback

**Kolom-kolom:**

### **1. id**

- Tipe: INT
- Atribut: AUTO\_INCREMENT, PRIMARY KEY
- Deskripsi: Identifikasi unik untuk setiap entri feedback.

### **2. customer\_name**

- Tipe: VARCHAR(225)
- Deskripsi: Nama pelanggan yang memberikan feedback.

### **3. feedback\_date**

- Tipe: DATE
- Deskripsi: Tanggal ketika feedback diberikan.

### **4. feedback\_text**

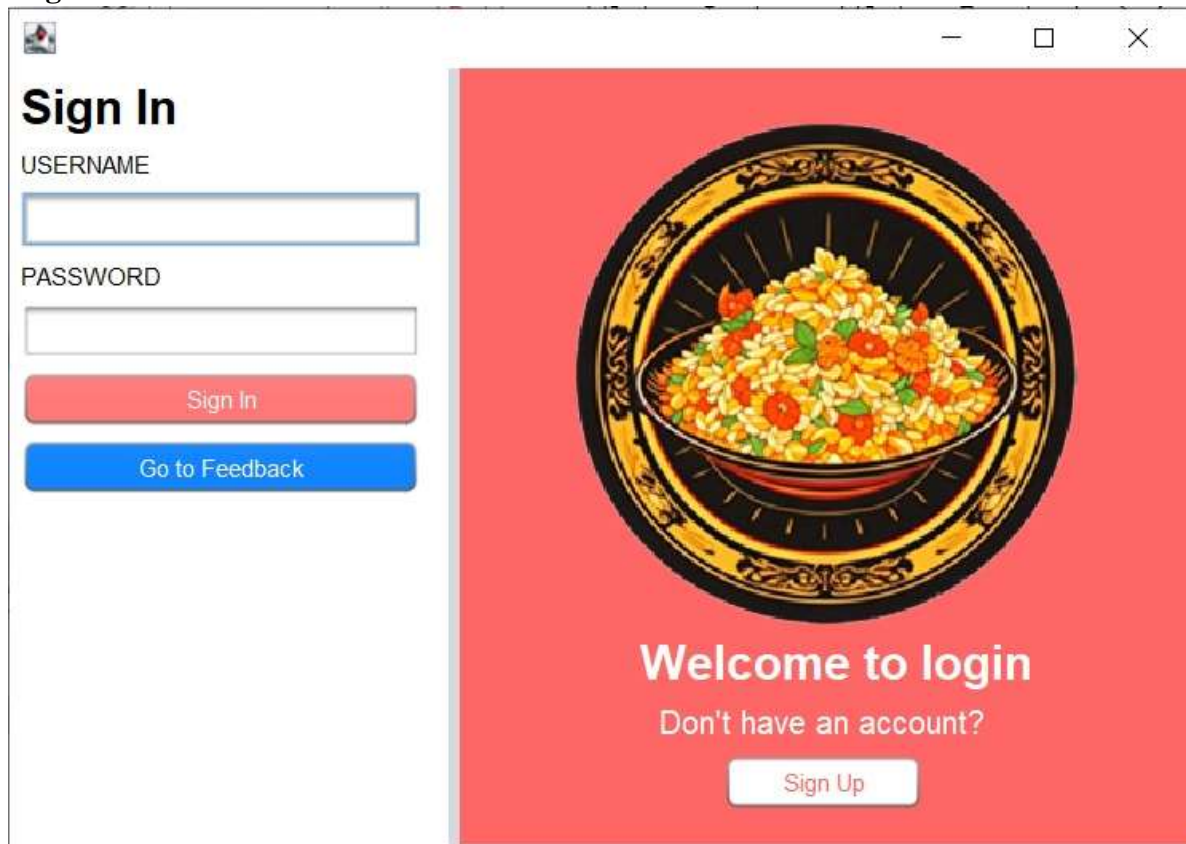
- Tipe: TEXT
- Deskripsi: Teks umpan balik yang diberikan oleh pelanggan.

### **5. rating**

- Tipe: INT
- Deskripsi: Penilaian yang diberikan oleh pelanggan dalam bentuk angka (misalnya, skala 1-5).

## **3.3 Rancangan User Interface**

## 1. Login Form



A screenshot of a web browser window displaying a login form. The form is titled "Sign In" and includes input fields for "USERNAME" and "PASSWORD". Below these fields are two buttons: a red "Sign In" button and a blue "Go to Feedback" button. To the right of the form is a large red banner featuring a bowl of orzo (small rice-like pasta) with tomatoes and herbs. Below the bowl, the text "Welcome to login" is displayed in white, followed by "Don't have an account?" and a white "Sign Up" button.

**Sign In**

USERNAME

PASSWORD

Sign In

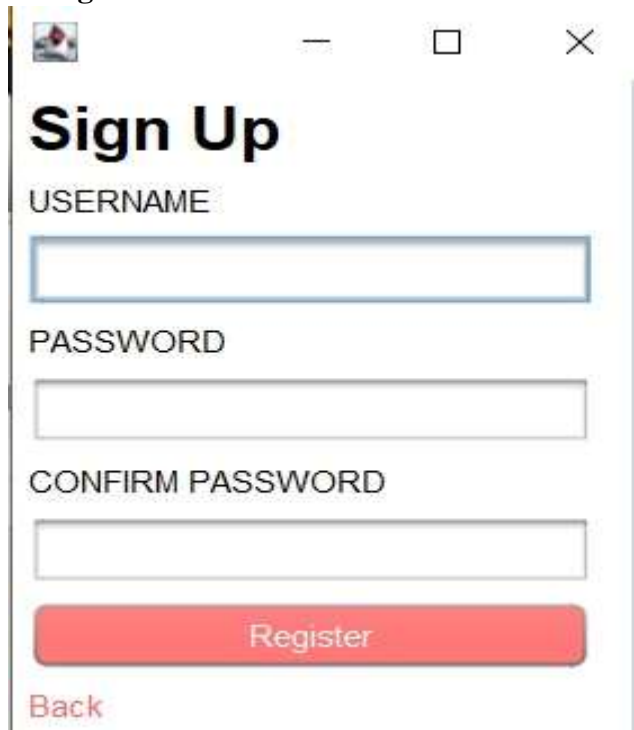
Go to Feedback

**Welcome to login**

Don't have an account?

Sign Up

## 2. Register Form



A screenshot of a web browser window displaying a registration form. The form is titled "Sign Up" and includes input fields for "USERNAME", "PASSWORD", and "CONFIRM PASSWORD". Below these fields is a red "Register" button. At the bottom left of the form, there is a red "Back" link. The browser window has standard window controls (minimize, maximize, close) at the top.

**Sign Up**

USERNAME

PASSWORD

CONFIRM PASSWORD

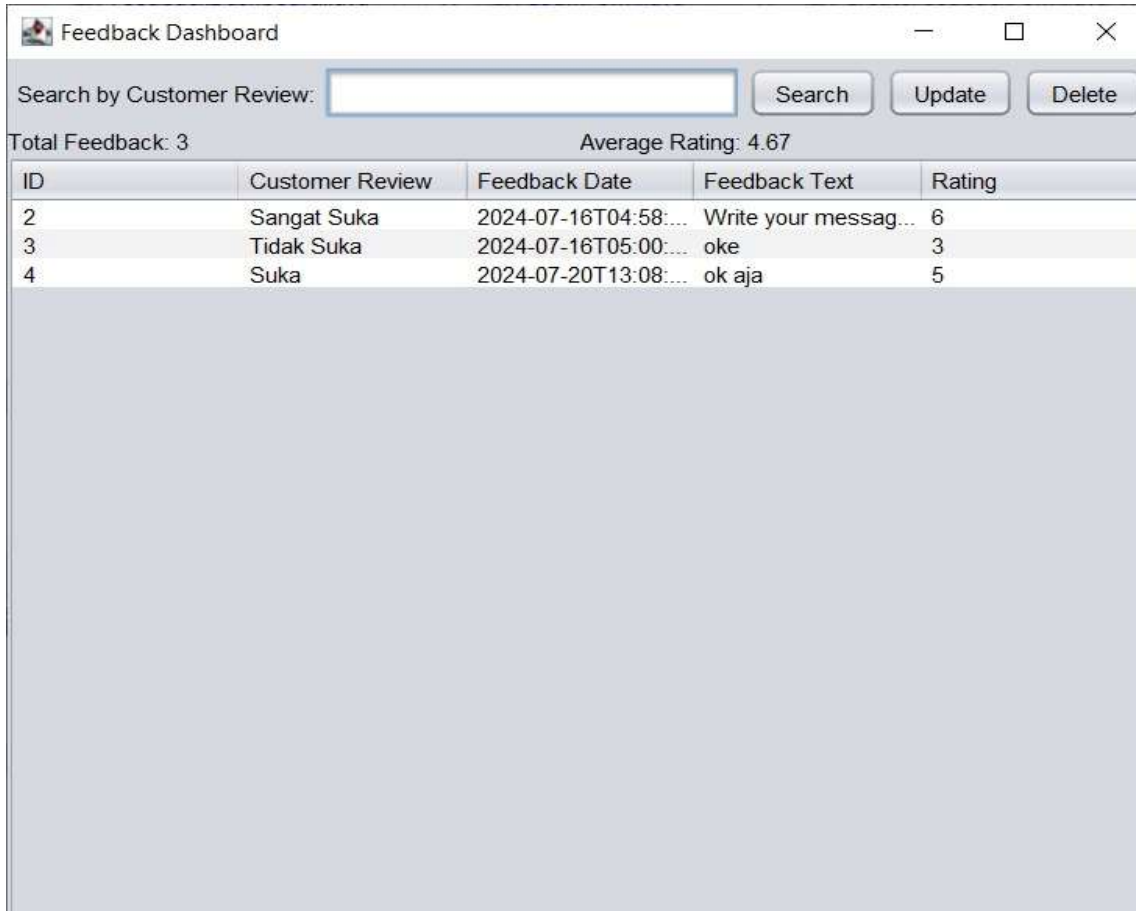
Register

Back

### 3. Feedback Pelanggan

A screenshot of a web browser window titled "Create Fee...". The window contains a question "How likely are you to answer to this question?" followed by four emoji-based rating options: a crying face, a face with a single tear, a smiling face with a tongue sticking out, and a smiling face with heart eyes. Below the emojis is a text input field with the placeholder "Write your message here". At the bottom right of the window is a "Submit" button.

### 4. Dashboard

A screenshot of a web browser window titled "Feedback Dashboard". The window has a search bar labeled "Search by Customer Review:" with a text input field and three buttons: "Search", "Update", and "Delete". Below the search bar, it shows "Total Feedback: 3" and "Average Rating: 4.67". A table displays the feedback data with columns: ID, Customer Review, Feedback Date, Feedback Text, and Rating. The table contains three rows of data. Below the table is a large empty grey area.

ID	Customer Review	Feedback Date	Feedback Text	Rating
2	Sangat Suka	2024-07-16T04:58:...	Write your messag...	6
3	Tidak Suka	2024-07-16T05:00:...	oke	3
4	Suka	2024-07-20T13:08:...	ok aja	5

```
List<Feedback> feedbackList =  
feedbackController.searchFeedbacksByCustomerName(customerName);  
3.4 Sourcecode
```

1. *Create Feedback*

**Input:**

```
String feedbackText = feedbackTextArea.getText();  
if (feedbackText.equals(placeholderText)) {  
    feedbackText = ""; // Reset placeholder text to empty  
}  
Feedback feedback = new Feedback();  
feedback.setCustomerName(customerName);  
feedback.setFeedbackText(feedbackText);  
feedback.setFeedbackDate(LocalDateTime.now());  
feedback.setRating(loveButton.isSelected() ? 6 : happyButton.isSelected() ? 5 :  
neutralButton.isSelected() ? 3 : 1); // Rating for love button  
  
feedbackController.saveFeedback(feedback);
```

**Output:**

```
JOptionPane.showMessageDialog(this, "Feedback submitted successfully!", "Success",  
JOptionPane.INFORMATION_MESSAGE);
```

2. *Play audio*

**Input:**

```
try (InputStream is = getClass().getResourceAsStream(filePath);  
    BufferedInputStream bis = new BufferedInputStream(is)) {  
    Player player = new Player(bis);  
    player.play();  
} catch (JavaLayerException | IOException e) {  
    e.printStackTrace();  
}
```

**Output:**

audio playing

3. *Login*

**Input:**

```
String username = usernameField.getText();  
String password = new String(passwordField.getPassword());  
User user = userController.authenticate(username, password);
```

**output:**

```
if (user != null) {  
    JOptionPane.showMessageDialog(LoginForm.this, "Login successful!");  
}
```

```

        // Open the dashboard and close the login form
        new FeedbackDashboard().setVisible(true);
        dispose();
    } else {
        JOptionPane.showMessageDialog(LoginForm.this, "Invalid username or password",
        "Login Error", JOptionPane.ERROR_MESSAGE);
    }
}

```

#### 4. search feedback

**Input:**

*String* customerName = searchField.getText();

**output:**

*List<Feedback>* feedbackList =  
feedbackController.searchFeedbacksByCustomerName(customerName);

Berikut link google drive :

[https://drive.google.com/drive/folders/1X09R\\_E6uFd03vNSFNeZm5pe9A8oriwrB?usp=sharing](https://drive.google.com/drive/folders/1X09R_E6uFd03vNSFNeZm5pe9A8oriwrB?usp=sharing)

## **BAB IV PENUTUP**

### **4.1 Kesimpulan**

Rancangan situs web ini telah berhasil memenuhi sebagian besar kebutuhan dan harapan pelanggan, dengan beberapa area yang masih dapat ditingkatkan. Dengan menerapkan rekomendasi yang diusulkan, diharapkan situs web restoran nasi goreng dapat memberikan pengalaman yang lebih baik bagi pelanggan, meningkatkan kepuasan, dan memperkuat loyalitas mereka. Kami berterima kasih kepada semua pelanggan yang telah memberikan feedback mereka dan berkomitmen untuk terus mengembangkan dan memperbaiki situs web demi memberikan layanan terbaik.

### **4.2 Saran**

Untuk pengembangan lebih lanjut, aplikasi ini dapat ditingkatkan dengan fitur pencarian dan penyortiran data, serta penggunaan database relasional untuk manajemen data yang lebih kompleks.