

# **Laporan Rancangan Project Tugas Besar**

## **Pemograman Web 2**

Dosen pengajar:

RUDHI WAHYUDI FEBRIANTO



Disusun Oleh:

Nama : Ade Hikmat

Kelas : 222K

NPM : 22552011130

**PROGRAM STUDI TEKNIK INFORMATIKA**

**UNIVERSITAS TEKNOLOGI BANDUNG**

**2024**

## Table of Contents

<b><i>BAB I PENDAHULUAN .....</i></b>	<b><i>2</i></b>
1.1 Latar Belakang .....	2
1.2 Rumusan Masalah .....	3
1.3 Batasan Masalah .....	3
1.4 Landasan Teori .....	4
1.5 Tujuan .....	5
<b><i>BAB II METODOLOGI PENELITIAN.....</i></b>	<b><i>6</i></b>
2.1 Metodologi .....	6
2.2 Sumber Data .....	6
2.3 Teknik Analisa .....	7
<b><i>BAB III PEMBAHASAN .....</i></b>	<b><i>8</i></b>
3.1 Pembahasan .....	8
3.2 Rancangan Relasi Database .....	9
3.3 Rancangan User Interface.....	10
<b><i>BAB IV PENUTUP.....</i></b>	<b><i>16</i></b>
4.1 Kesimpulan.....	16
4.2 Saran .....	16
<b><i>Daftar Pustaka .....</i></b>	<b><i>17</i></b>

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Perkembangan teknologi informasi yang pesat telah mengubah banyak aspek kehidupan, termasuk dalam pengelolaan data. Dalam era digital ini, pengelolaan data secara manual kini dianggap kurang efisien dan rentan terhadap kesalahan. Oleh karena itu, solusi yang lebih canggih dan otomatis menjadi kebutuhan mendesak. Aplikasi web berbasis PHP muncul sebagai solusi unggul dalam pengelolaan data karena kemudahan dalam penggunaannya dan dukungan komunitas yang luas.

Salah satu aspek penting yang memerlukan sistem pengelolaan data yang efisien adalah pengelolaan data feedback pelanggan. Feedback pelanggan sangat berharga bagi perusahaan karena memberikan wawasan langsung tentang kualitas produk dan layanan yang ditawarkan. Dengan adanya sistem yang efisien, pengumpulan, analisis, dan pengambilan keputusan berdasarkan feedback pelanggan dapat dilakukan dengan lebih cepat dan akurat.

Dalam konteks ini, proyek pengembangan aplikasi web untuk mengelola data feedback pelanggan menggunakan Laravel, sebuah framework PHP yang populer dan kuat, menjadi sangat relevan. Laravel menyediakan berbagai fitur yang mendukung pengembangan aplikasi web yang handal dan efisien, seperti ORM (Eloquent), routing yang mudah, dan alat bantu pengembangan lainnya. Dengan menggunakan basis data relasional, proyek ini menawarkan solusi sederhana namun efektif untuk kebutuhan pengelolaan data feedback pelanggan.

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 Laravel. 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.

Sejalan dengan tujuan tersebut, proyek ini diharapkan dapat menjadi referensi bagi pengembangan aplikasi serupa di masa depan dan memberikan manfaat nyata dalam peningkatan efisiensi dan akurasi pengelolaan data feedback pelanggan.

## **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 Laravel.
- 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.

### **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.

## 1.4 Landasan Teori

### 1. PHP (Hypertext Preprocessor)

**Definisi:** 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.

**Relevansi:** 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)

**Definisi:** MySQL adalah sistem manajemen basis data relasional yang menggunakan SQL (Structured Query Language) untuk mengelola dan mengakses data yang disimpan dalam basis data.

**Relevansi:** 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)

**Definisi:** HTML adalah bahasa markup standar untuk membuat halaman web. HTML digunakan untuk membangun struktur dasar konten di web.

**Relevansi:** 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)

**Definisi:** CSS adalah bahasa stylesheet yang digunakan untuk mendesain dan memperindah tampilan halaman web. CSS memungkinkan pemisahan antara konten dan presentasi.

**Relevansi:** CSS digunakan dalam proyek ini untuk mendesain antarmuka pengguna yang responsif dan menarik, seperti formulir feedback dan dasbor admin.

### 5. JavaScript

**Definisi:** 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.

**Relevansi:** JavaScript digunakan untuk menambah interaktivitas dalam aplikasi, seperti validasi formulir feedback dan fitur-fitur dinamis pada dasbor admin.

## 6. **Laravel**

**Definisi:** Laravel adalah framework PHP yang menyediakan struktur dan alat untuk membangun aplikasi web yang skalabel dan terorganisir. Laravel mengikuti pola MVC (Model-View-Controller) untuk memisahkan logika aplikasi, presentasi, dan data.

**Relevansi:** Laravel digunakan dalam proyek ini untuk mempercepat dan mempermudah pengembangan aplikasi web, menyediakan fitur-fitur seperti Eloquent ORM untuk interaksi dengan database, sistem routing yang mudah, dan alat bantu pengembangan lainnya.

## 7. **SMTP (Simple Mail Transfer Protocol)**

**Definisi:** 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.

**Relevansi:** 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.

## 1.5 Tujuan

1. **Mengembangkan Aplikasi Web untuk Mengelola Data Feedback Pelanggan**  
Membangun aplikasi web yang mampu mengelola data feedback pelanggan secara efisien, dari pengumpulan hingga analisis data, untuk meningkatkan kualitas layanan dan produk yang ditawarkan.
2. **Mempermudah Proses Pengelolaan Data Feedback Pelanggan melalui Antarmuka Web**  
Menciptakan antarmuka pengguna yang intuitif dan mudah digunakan sehingga mempermudah administrator dalam mengelola data feedback pelanggan. Ini termasuk penambahan, pengeditan, penghapusan, dan tampilan data feedback.
3. **Menerapkan Konsep MVC (Model-View-Controller) dalam Pengembangan Aplikasi Web**  
Mengimplementasikan arsitektur MVC untuk memisahkan logika aplikasi, presentasi, dan data. Hal ini akan meningkatkan modularitas dan maintainability aplikasi, serta mempermudah pengembangan dan perawatan aplikasi di masa depan.
4. **Menyediakan Fitur Notifikasi Email untuk Feedback Baru**  
Mengembangkan fitur yang mengirim notifikasi email kepada administrator setiap kali ada feedback baru yang masuk, sehingga memungkinkan respons yang cepat terhadap umpan balik pelanggan.
5. **Meningkatkan Keterampilan dalam Pengembangan Aplikasi Web Menggunakan Laravel**  
Menggunakan Laravel sebagai framework PHP untuk mengembangkan aplikasi web, dengan tujuan meningkatkan keterampilan dalam menggunakan framework ini, termasuk fitur-fitur seperti Eloquent ORM, sistem routing, dan alat bantu pengembangan lainnya.
6. **Menyimpan Data Feedback Pelanggan dengan Format yang Sederhana namun Efisien**  
Merancang dan mengimplementasikan struktur database yang efisien untuk menyimpan data feedback pelanggan, memastikan data mudah diakses dan diolah tanpa mengorbankan performa dan efisiensi ruang penyimpanan.

## 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 menggunakan Laravel.
- 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:

- o **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:

- o **Basis Data Relasional:** 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:

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

```
4.  
5. CREATE TABLE feedback (  
6.   id INT AUTO_INCREMENT PRIMARY KEY,  
7.   customer_name VARCHAR(100) NOT NULL,  
8.   feedback_date DATE NOT NULL,  
9.   feedback_text TEXT NOT NULL,  
10.  rating INT NOT NULL,
```

```
11.     created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
12. );
13.
```

#### 14. **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.

#### 15. **Notifikasi Email:**

- **Pemberitahuan:** Setiap kali ada feedback baru yang masuk, notifikasi email akan dikirim kepada administrator. Hal ini memastikan bahwa umpan balik pelanggan dapat segera ditanggapi.

### 2.3 **Teknik Analisa**

Analisis dilakukan dengan mengevaluasi fungsionalitas aplikasi dan kinerja sistem dalam mengelola data kendaraan. Pengujian dilakukan dengan skenario uji untuk memastikan aplikasi bekerja sesuai dengan yang diharapkan.



## BAB III PEMBAHASAN

### 3.1 Pembahasan

#### 1. Pengumpulan Data Feedback Pelanggan

**Formulir Online:** 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

**Basis Data Relasional:** 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.

**Eloquent ORM:** 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)

**Penambahan Data:** Pelanggan dapat menambah feedback mereka melalui formulir yang tersedia di halaman web. Data yang dimasukkan akan disimpan langsung ke dalam basis data.

**Tampilan Data:** Data feedback yang telah dikumpulkan dapat ditampilkan dalam bentuk tabel di dasbor admin. Administrator dapat melihat dan menilai feedback yang diberikan oleh pelanggan.

**Pengeditan Data:** Administrator dapat mengedit data feedback yang telah disimpan jika terdapat kesalahan atau perlu perbaikan.

**Penghapusan Data:** Administrator juga memiliki kemampuan untuk menghapus data feedback yang tidak lagi relevan atau dibutuhkan.

#### 4. Notifikasi Email

**Pemberitahuan Real-Time:** 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

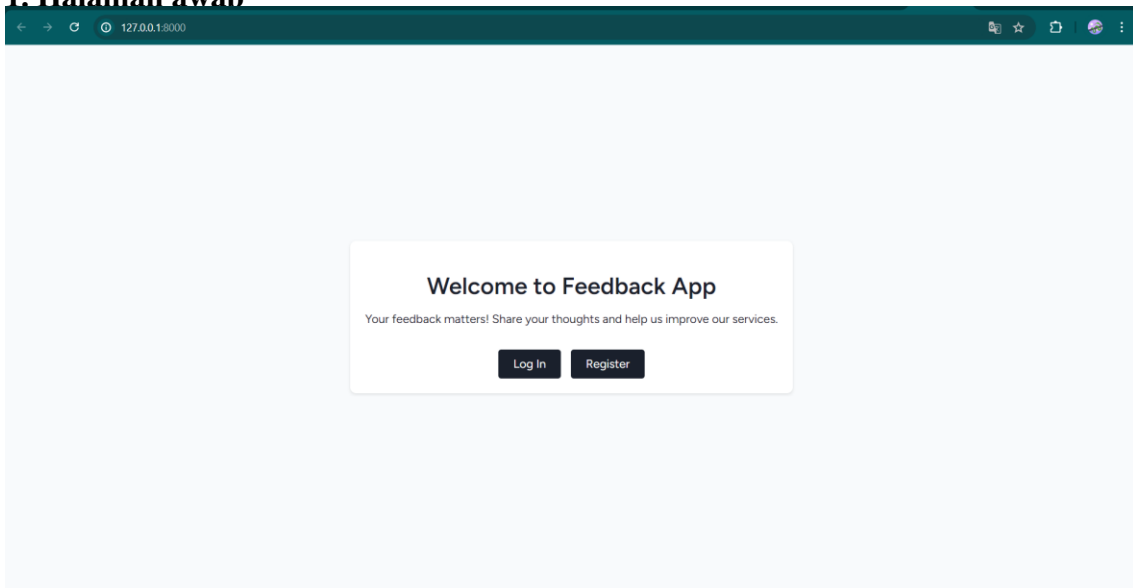
**Nama Tabel:** feedback

**Kolom-kolom:**

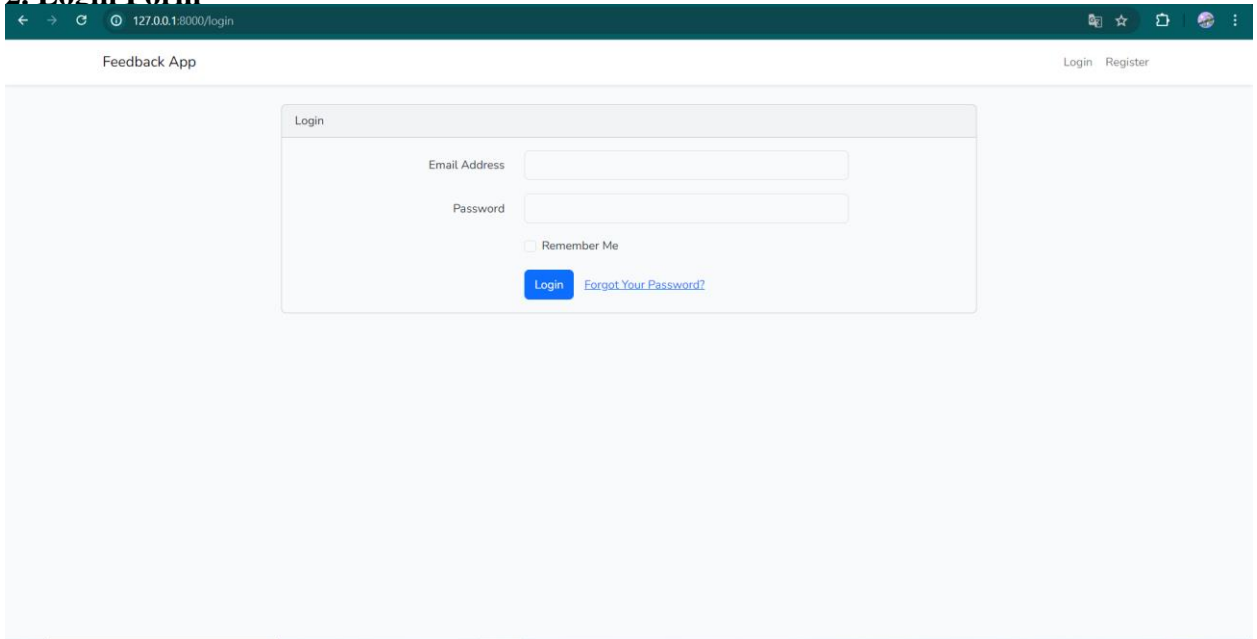
1. **id**
  - **Tipe:** INT
  - **Atribut:** AUTO\_INCREMENT, PRIMARY KEY
  - **Deskripsi:** Identifikasi unik untuk setiap entri feedback.
2. **customer\_name**
  - **Tipe:** VARCHAR(100)
  - **Deskripsi:** Nama pelanggan yang memberikan feedback.
3. **feedback\_date**
  - **Tipe:** DATE
  - **Deskripsi:** 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).
6. **created\_at**
  - **Tipe:** TIMESTAMP
  - **Atribut:** DEFAULT CURRENT\_TIMESTAMP
  - **Deskripsi:** Waktu ketika entri feedback dibuat.

### **3.3 Rancangan User Interface**

## 1. Halaman awab



## 2. Login Form



### 3. Register Form

Feedback App

Login Register

Register

Name

Email Address

Password

Confirm Password

Register

### 4. Dashboard

Feedback App

ade hikmat ▾

Dasbor Admin

Total Feedback: 5

Rata-rata Rating: 4.0

Feedback Terbaru:

1. Rese (1) - Buruk

2. Zask (4) - oke lah

3. Liu Bi (5) - Good service

4. Zuan (5) - enak

5. Yance (5) - good job

Lihat Semua Feedback

Submit Feedback

Click the link below to submit feedback:

<http://127.0.0.1:8000/feedback/create/1>

Copy Feedback Link

Go to Feedback Form

## 5. List Feedback and Print Pdf

Feedback App

ade hikmat ▾

Feedback List

Create New Feedback

Print PDF

Search

Search

No	Customer Name	Feedback Date	Feedback Text	Rating	Action
1	Yance	2024-07-15	good job	5	<div>ShowEditDelete</div>
2	Zuan	2024-07-14	enak	5	<div>ShowEditDelete</div>
3	Liu Bi	2024-07-08	Good service	5	<div>ShowEditDelete</div>
4	Zask	2024-07-14	oke Lah	4	<div>ShowEditDelete</div>
5	Rese	2024-07-13	Buruk	1	<div>ShowEditDelete</div>

## 6. Search

Feedback App

ade hikmat ▾

Feedback List

Create New Feedback

Print PDF

zuan

Search

No	Customer Name	Feedback Date	Feedback Text	Rating	Action
1	Zuan	2024-07-14	enak	5	<div>ShowEditDelete</div>

## 7. Pdf Format

Feedback List PDF

1 / 1 | 100% + | [ ] [ ]

1

### Feedback List

No	Customer Name	Feedback Date	Feedback Text	Rating
1	Yance	2024-07-15	good job	5
2	Zuan	2024-07-14	enak	5
3	Liu Bi	2024-07-08	Good service	5
4	Zask	2024-07-14	oke lah	4
5	Rese	2024-07-13	Buruk	1

## 8. Feedback Form external

Feedback App

Login Register

127.0.0.1:8000/feedback/create/1

### Create Feedback

Customer Name

Feedback Text

Rating

[Submit Feedback](#)

9. Notification Email

New Feedback Created

From: Feedback App <hello@example.com>  
To: <ade@getnada.com>

    
2024-07-06 16:07, 698 Bytes

[Show Headers](#)

- HTML
- HTML Source
- Text
- Raw
- Spam Analysis
- HTML Check 1
- Tech Info



New Feedback Created

A new feedback has been created with the following details:

- Customer Name: ade
- Feedback Date: 2024-07-05
- Feedback Text: ademaksnaj
- Rating: 2



## **BAB IV PENUTUP**

### **4.1 Kesimpulan**

Pengembangan aplikasi web berbasis PHP untuk pengelolaan data feedback pelanggan menawarkan solusi yang efisien dan efektif untuk mengumpulkan, menyimpan, dan menganalisis feedback yang diberikan oleh pelanggan. Melalui implementasi fitur CRUD (Create, Read, Update, Delete), antarmuka pengguna yang intuitif, dan notifikasi email real-time, aplikasi ini mampu memberikan dukungan yang signifikan dalam pengambilan keputusan bisnis yang lebih baik.

Pemanfaatan framework Laravel dalam proyek ini tidak hanya mempercepat proses pengembangan tetapi juga meningkatkan struktur dan kualitas kode, sehingga aplikasi menjadi lebih mudah untuk dipelihara dan diperluas di masa depan. Dengan menggunakan basis data relasional, data feedback disimpan secara terstruktur dan efisien, memungkinkan akses yang cepat dan analisis yang akurat.

### **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.

## Daftar Pustaka

SOURCE CODE: <https://github.com/adehikmatfr/utb-uas-s4-feedback-app>

- <https://www.php.net/docs.php>
- <https://dev.mysql.com/doc/>
- <https://laravel.com/docs>
- <https://developer.mozilla.org/en-US/docs/Web/HTML>
- <https://developer.mozilla.org/en-US/docs/Web/CSS>
- <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- <https://getbootstrap.com/docs/5.0/getting-started/introduction/>
- <https://www.w3schools.com/sql/>
- <https://documentation.mailgun.com/>