



## PERANCANGAN PROGRAM PADA TOKO ROTI *PURRY'S CAKE AND BAKERY* BERBASIS WEB MENGGUNAKAN *FRAMEWORK* LARAVEL

Fakhri Hamdani<sup>1</sup>, Roland Kenneth<sup>2</sup>, Bryan Mathew Napitupulu<sup>3</sup>, Muhammad Ari Shidqi<sup>4</sup>

<sup>1</sup>19230654, Program Studi Sistem Informasi, Fakultas Teknik Dan Informatika,  
Universitas Bina Sarana Informatika

<sup>2</sup>19230772, Program Studi Sistem Informasi, Fakultas Teknik Dan Informatika,  
Universitas Bina Sarana Informatika

<sup>3</sup>19230824, Program Studi Sistem Informasi, Fakultas Teknik Dan Informatika,  
Universitas Bina Sarana Informatika

<sup>4</sup>19230724, Program Studi Sistem Informasi, Fakultas Teknik Dan Informatika,  
Universitas Bina Sarana Informatika

<sup>1</sup>[fakhrihamdani30@gmail.com](mailto:fakhrihamdani30@gmail.com), <sup>2</sup>[rolandkennethcen@gmail.com](mailto:rolandkennethcen@gmail.com),

<sup>3</sup>[bryannapitupulu70@gmail.com](mailto:bryannapitupulu70@gmail.com), <sup>4</sup>[muhammadarishidqi@gmail.com](mailto:muhammadarishidqi@gmail.com)

### Abstrak

Penerapan sistem informasi berbasis web telah menjadi kebutuhan penting dalam mendukung operasional bisnis, termasuk pada sektor UKM seperti Toko Roti *Purry's Cake and Bakery*. Laporan ini membahas perancangan dan pengembangan sistem informasi penjualan dengan pendekatan spiral, baik sebelum maupun sesudah menggunakan *framework* Laravel. Pada tahap awal, sistem dirancang secara manual menggunakan HTML, CSS, PHP, dan MySQL tanpa *framework* untuk memberikan pemahaman dasar mengenai alur sistem. Selanjutnya, sistem yang sama dikembangkan menggunakan Laravel untuk meningkatkan modularitas, efisiensi, dan keamanan. Hasil penelitian menunjukkan bahwa Laravel memberikan keunggulan signifikan dalam pengelolaan kode, performa, dan fitur keamanan. Selain itu, sistem dengan Laravel lebih efisien dalam menangani data yang besar dan menyediakan fitur tambahan seperti autentikasi pengguna. Pengujian dilakukan menggunakan metode *black-box* dan *white-box* testing untuk memastikan fungsi sistem berjalan dengan baik dan struktur kode optimal. Perbandingan hasil menunjukkan bahwa pengembangan dengan Laravel memberikan dampak positif pada kualitas sistem secara keseluruhan. Penelitian ini diharapkan dapat menjadi referensi bagi pengembang sistem informasi penjualan di masa depan.

**Kata Kunci:** Sistem Informasi Penjualan, Laravel, *Framework*, *Black-Box Testing*, *White-Box Testing*.

### Article History

Received: January 2025  
Reviewed: January 2025  
Published: January 2025

Plagirism Checker No 234  
Prefix DOI :  
10.8734/Kohesi.v1i2.365  
Copyright : Author  
Publish by : Kohesi



This work is licensed under  
a [Creative Commons  
Attribution-NonCommercial 4.0  
International License](https://creativecommons.org/licenses/by-nc/4.0/)



## PENDAHULUAN

### A. Latar Belakang

Toko Roti Purry's *Cake and Bakery* merupakan salah satu usaha kecil menengah (UKM) yang berfokus pada penjualan berbagai jenis roti dan kue. Sebagai usaha yang terus berkembang, efisiensi dalam pengelolaan transaksi dan pencatatan penjualan menjadi kebutuhan utama. Namun, hingga saat ini, sistem pencatatan transaksi di toko ini masih dilakukan secara manual. Hal ini tidak hanya memperlambat proses operasional, tetapi juga berpotensi menyebabkan kesalahan pencatatan yang dapat memengaruhi pengambilan keputusan.

Seiring dengan kemajuan teknologi, penerapan sistem informasi berbasis web menjadi solusi yang relevan untuk mengatasi permasalahan tersebut. Sistem informasi penjualan berbasis web tidak hanya memungkinkan pengelolaan data yang lebih terorganisasi, tetapi juga memberikan akses cepat dan real-time terhadap data transaksi. Dengan teknologi ini, pengambilan keputusan berbasis data dapat dilakukan dengan lebih akurat.

Dalam laporan ini, sistem informasi penjualan dirancang menggunakan metode pengembangan spiral yang memberikan fleksibilitas pada proses iteratif. Desain awal sistem dibuat tanpa menggunakan *framework* untuk memberikan gambaran dasar perancangan. Selanjutnya, sistem dikembangkan menggunakan *framework* Laravel yang mendukung modularitas kode, keamanan, dan efisiensi performa. Perbandingan antara kedua pendekatan ini akan dianalisis untuk menilai keunggulan dan kekurangan masing-masing.

Dengan pengembangan ini, diharapkan Toko Roti Purry's *Cake and Bakery* dapat meningkatkan efisiensi operasional dan memanfaatkan teknologi untuk mendukung pertumbuhan bisnis. Studi ini juga memberikan wawasan praktis bagi pengembang mengenai manfaat penggunaan *framework* Laravel dalam pengembangan sistem informasi penjualan.

### B. Rumusan Masalah

1. Bagaimana perancangan sistem informasi penjualan sebelum menggunakan *framework* Laravel?
2. Bagaimana pengembangan sistem informasi penjualan menggunakan *framework* Laravel?
3. Apa perbedaan performa dan efisiensi antara sistem sebelum dan sesudah menggunakan *framework* Laravel?

### C. Tujuan Penelitian

1. Merancang sistem informasi penjualan berbasis web tanpa menggunakan *framework* untuk memahami dasar pengembangan.
2. Mengembangkan sistem informasi penjualan dengan *framework* Laravel untuk meningkatkan performa dan struktur.
3. Membandingkan sistem sebelum dan sesudah menggunakan *framework* Laravel.

### D. Manfaat penelitian

Meningkatkan efisiensi dalam pengelolaan transaksi penjualan dan pembuatan laporan.



## LANDASAN TEORI

### 1. Pengertian Sistem Informasi

Sistem informasi merupakan sebuah sistem dalam suatu organisasi yang menghubungkan kebutuhan pengolahan transaksi harian dengan fungsi operasional dan manajerial organisasi, serta mendukung kegiatan strategis. Tujuannya adalah untuk menghasilkan laporan-laporan yang dibutuhkan oleh pihak-pihak terkait (Firmansyah et al., 2022).

### 2. Sistem Informasi Penjualan

Sistem informasi penjualan adalah sistem yang dirancang untuk membantu pengelolaan transaksi penjualan, mulai dari pencatatan data pelanggan, produk, hingga laporan penjualan. Sistem ini bertujuan meningkatkan efisiensi, mengurangi risiko kesalahan, dan mempermudah akses informasi bagi pengelola toko. (Nugroho & Nugroho, 2021).

### 3. Framework Laravel

Laravel adalah *framework* berbasis PHP yang mendukung pengembangan aplikasi *website* dengan pendekatan *Model-View-Controller* (MVC). *Framework* ini menawarkan efisiensi dalam pengelolaan *database*, keamanan bawaan, dan kemudahan integrasi dengan layanan pihak ketiga (Nugroho & Nugroho, 2021). Laravel telah digunakan dalam berbagai proyek *e-commerce* karena kemampuannya yang *powerful*, aman, dan cepat (Nugroho & Nugroho, 2021).

### 4. E-commerce dan Manfaatnya

*E-commerce* memungkinkan transaksi *online* secara efisien dengan menyediakan informasi produk secara lengkap dan fitur integrasi pembayaran. Keunggulan lainnya adalah peningkatan pengalaman pelanggan melalui kemudahan akses dan transparansi data (Nugroho & Nugroho, 2021).

### 5. Keamanan Sistem Informasi Berbasis Website

Teknik-teknik seperti enkripsi data, autentikasi pengguna, dan manajemen izin akses diperlukan untuk memastikan data tetap aman dari ancaman eksternal. Laravel memiliki fitur keamanan bawaan, seperti *CSRF* (*CrossSite Request Forgery*) *protection* dan *hashing password*, yang membantu melindungi aplikasi dari serangan siber (Nugroho & Nugroho, 2021).

## METODE PENELITIAN

### A. Perancangan Sistem

Pada tahap ini, perancangan sistem berfokus pada desain alur kerja sistem yang akan memecahkan masalah yang ditemukan pada tahap pengumpulan data. Beberapa langkah yang dilakukan dalam perancangan sistem adalah sebagai berikut:

#### 1. Desain Antarmuka Pengguna (UI)

Merancang antarmuka pengguna (UI) yang mudah digunakan oleh pelanggan dan admin. Halaman untuk pemesanan produk dirancang dengan menampilkan gambar produk, harga, dan tombol untuk melakukan pemesanan. Untuk admin, antarmuka menampilkan *dashboard* untuk memantau status pesanan, mengelola produk, dan menghasilkan laporan.



## 2. Desain Database

Pada tahap ini, perancangan sistem dilakukan dengan pendekatan berbasis diagram, yaitu *Entity Relationship Diagram* (ERD) dan *Use Case Diagram* untuk menggambarkan struktur dan alur kerja sistem yang akan memecahkan masalah yang ditemukan pada tahap pengumpulan data.

### B. Pengembangan sistem

Metode yang digunakan dalam pembuatan sistem informasi ini adalah metode spiral. Metode spiral merupakan model pengembangan perangkat lunak yang bersifat evolusioner, menggabungkan pendekatan iteratif dari prototipe dengan pengendalian dan struktur sistematis dari model linier. Model ini memungkinkan pengembangan perangkat lunak dalam berbagai versi secara cepat. Dalam pendekatan spiral, pengembangan perangkat lunak dilakukan secara bertahap. Pada tahap awal iterasi, rilis awal dapat berupa rancangan di atas kertas atau prototipe sederhana. Pada iterasi selanjutnya, dihasilkan versi sistem teknik yang lebih lengkap (Puspita et al., 2021).



Gambar 1. Metode Spiral

#### 1. Identifikasi dan Perencanaan

Mengidentifikasi kebutuhan sistem berdasarkan pengumpulan data dan perencanaan pengembangan modul-modul sistem.

- a. Aktivitas: Wawancara dengan pemilik toko dan observasi proses bisnis untuk memahami kebutuhan pelanggan dan admin.
- b. *Output*: Daftar kebutuhan sistem, seperti fitur pemesanan *online*, pengelolaan produk, dan laporan penjualan.

#### 2. Analisis Risiko dan Solusi Alternatif

Melakukan analisis risiko pada setiap tahap pengembangan untuk mengidentifikasi potensi masalah dan menentukan solusi alternatif.

- Risiko:
  - a. Ketidaksesuaian desain antarmuka dengan kebutuhan pengguna.
  - b. Kemungkinan kesalahan data pada laporan.
- Solusi:
  - a. Melibatkan pengguna untuk memberikan umpan balik pada prototipe.
  - b. Menambahkan validasi pada *input* data.



### 3. Pengembangan dan Pengujian

Pada tahap ini, prototipe dikembangkan berdasarkan kebutuhan yang telah diidentifikasi, kemudian diuji oleh pengguna untuk mendapatkan umpan balik.

- Aktivitas:

- a. Membuat modul pemesanan produk dan pengelolaan stok.
- b. Menguji prototipe awal untuk memastikan fungsionalitasnya.

### 4. Evaluasi dan Perbaikan

Setelah prototipe diuji, evaluasi dilakukan untuk menentukan perubahan yang diperlukan pada iterasi berikutnya.

- Aktivitas:

- a. Memperbaiki antarmuka berdasarkan umpan balik pelanggan.
- b. Menyempurnakan modul laporan untuk admin.

## C. Tahapan Iterasi Pengembangan Spiral

Setiap iterasi dalam metode spiral menghasilkan prototipe yang semakin mendekati sistem final. Berikut adalah tahapan iterasi yang dilakukan:

#### 1. Iterasi Pertama:

- Tujuan: Membuat modul pemesanan produk untuk pelanggan.

- Aktivitas:

- a. Mendesain antarmuka katalog produk.
- b. Menghubungkan *form* pemesanan dengan basis data.

- *Output*: Prototipe pemesanan *online*.

#### 2. Iterasi Kedua:

- Tujuan: Membuat *dashboard* untuk admin.

- Aktivitas:

- a. Mengembangkan fitur untuk menambah, mengedit, dan menghapus produk.
- b. Mengelola stok produk secara otomatis.

- *Output*: *Dashboard* admin untuk manajemen produk.

#### 3. Iterasi Ketiga:

- Tujuan: Menambahkan fitur laporan penjualan.

- Aktivitas:

- a. Membuat modul untuk menghasilkan laporan harian, mingguan, dan bulanan.
- b. Menguji laporan dengan data simulasi.

- *Output*: Laporan penjualan otomatis.

#### 4. Iterasi Keempat:

- Tujuan: Finalisasi dan integrasi sistem.

- Aktivitas:

- a. Mengintegrasikan semua modul.
- b. Melakukan pengujian akhir dengan data operasional nyata.

- *Output*: Sistem final yang siap digunakan.



## D. Pengujian sistem

Pengujian sistem dilakukan setelah tahap pengembangan untuk memastikan bahwa sistem berfungsi sesuai dengan yang diharapkan. Pengujian ini dilakukan menggunakan *Black Box Testing* dan *White Box Testing*.

- *Black Box Testing*: Menguji fungsionalitas sistem dari sudut pandang pengguna tanpa melihat kode sumber.
- *White Box Testing*: Menguji struktur internal kode untuk memastikan tidak ada *bug* atau kesalahan logika dalam sistem.

## HASIL DAN PEMBAHASAN

### A. Implementasi Sistem Sebelum Menggunakan *Framework Laravel*

Pada tahap ini, sistem dirancang dengan struktur dasar tanpa *framework*. Sistem dikembangkan menggunakan PHP dan MySQL dengan antarmuka dasar berbasis HTML dan CSS.

#### 1. Contoh Implementasi Pada *Header.php*

```
<main>
  <!-- Bagian Konten Utama -->
  <section id="hero">
    <h1>Selamat Datang di Toko Roti Kami</h1>
    <p>Temukan roti terbaik yang dibuat dengan cinta dan bahan berkualitas.</p>
    <a href="produk.php" class="btn">Jelajahi Produk</a>
  </section>

  <section id="about">
    <h2>Tentang Kami</h2>
    <p>Toko Roti adalah penyedia roti dan kue yang segar setiap hari. Kami berdedikasi untuk memberikan rasa terbaik untuk pelanggan kami.</p>
  </section>

  <section id="featured-products">
    <h2>Produk Unggulan</h2>
    <div class="product-list">
      <article class="product-item">
        
        <h3>Roti Sourdough</h3>
        <p>Roti tradisional dengan rasa yang khas.</p>
      </article>
      <article class="product-item">
        
        <h3>Croissant</h3>
        <p>Pastry lembut dan renyah untuk sarapan Anda.</p>
      </article>
      <article class="product-item">
```



```

<h3>Kue Cokelat</h3>
<p>Manisnya coklat terbaik di setiap gigitan.</p>
</article>
</div>
</section>
</main>

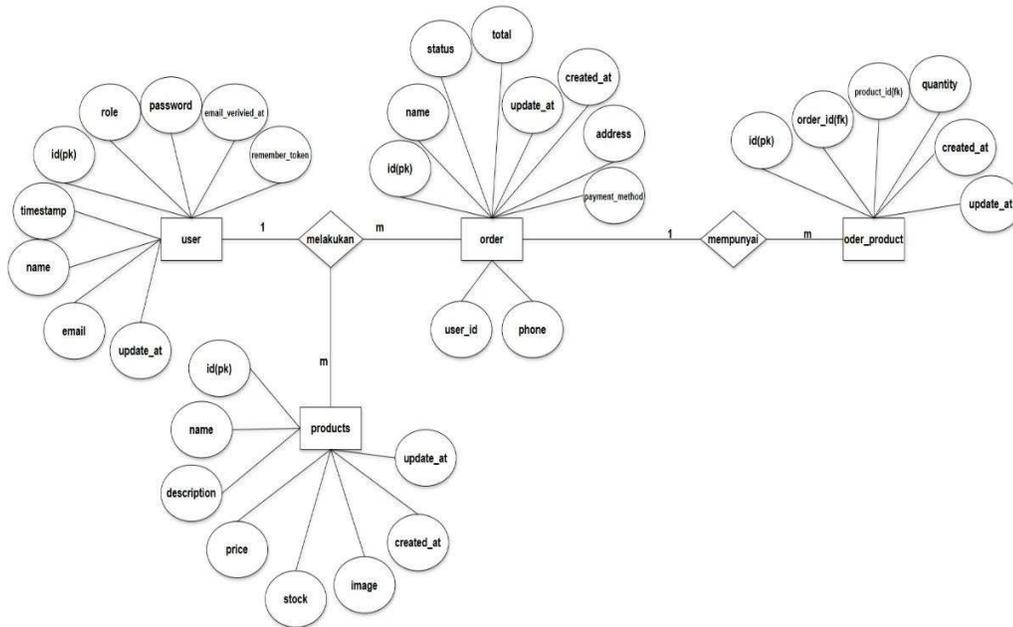
<footer>
<p>&copy; 2025 Toko Roti. Semua Hak Dilindungi.</p>
<ul>
<li><a href="https://facebook.com">Facebook</a></li>
<li><a href="https://instagram.com">Instagram</a></li>
<li><a href="https://twitter.com">Twitter</a></li>
</ul>
</footer>

</body>
</html>
```

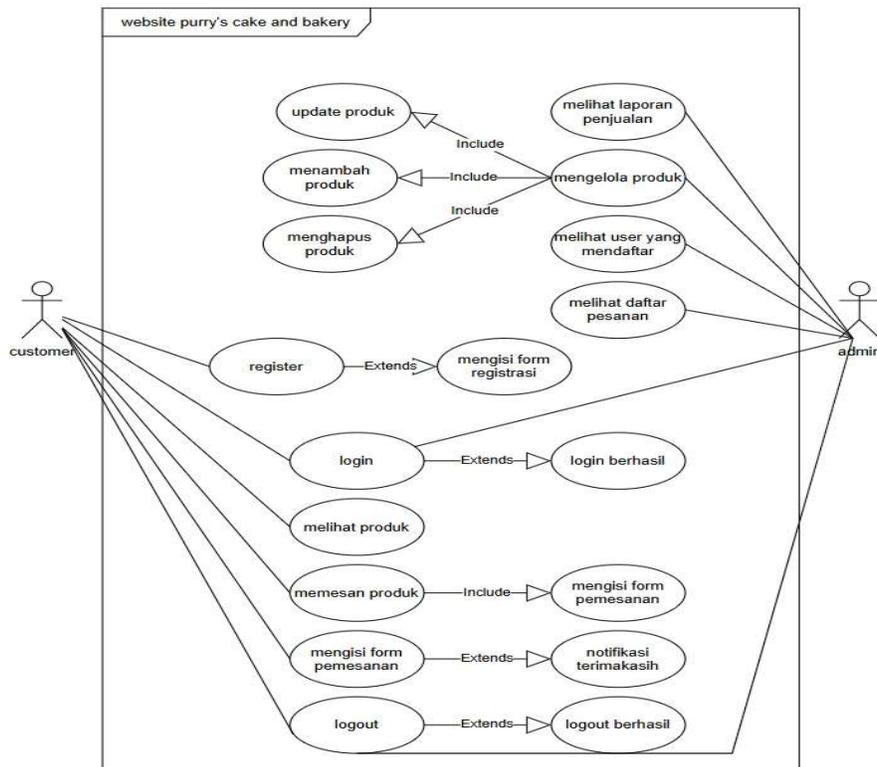
## 2. ERD (*Entity Relationship Diagram*)

Untuk memodelkan struktur basis data, digunakan *Entity Relationship Diagram* (ERD). ERD ini menggambarkan hubungan antara entitas yang ada dalam sistem, serta atribut yang ada pada masing-masing entitas. Adapun entitas utama dalam sistem informasi ini meliputi:

- a. Tabel *Users*: Menyimpan data pengguna sistem, termasuk pelanggan dan admin. Atribut yang disimpan meliputi *id*, *name*, *email*, *password*, *role*, dan tanggal pembuatan serta pembaruan akun (*created\_at*, *updated\_at*).
- b. Tabel *Orders*: Menyimpan informasi mengenai pesanan yang dilakukan oleh pelanggan. Atribut yang disimpan meliputi *id*, *user\_id* (relasi dengan pengguna), *name*, *address*, *phone*, *payment\_method*, *total*, dan status pesanan.
- c. Tabel *Products*: Menyimpan data produk yang dijual, seperti *id*, *name*, *description*, *price*, *stock*, *image*, dan tanggal pembuatan serta pembaruan produk.
- d. Tabel *Order\_Product*: Merupakan tabel penghubung antara tabel *Orders* dan *Products*, yang menyimpan data produk yang dipesan dalam setiap transaksi. Atribut yang disimpan adalah *order\_id*, *product\_id*, *quantity*, serta tanggal pembuatan dan pembaruan.
- e. Tabel *Reports*: Tabel ini digunakan untuk menyimpan data laporan yang dihasilkan sistem, seperti laporan penjualan dan laporan stok produk. Atribut yang disimpan meliputi *id*, *type*, *data*, *generated\_at*, *created\_at*, dan *updated\_at*.



### Use Case Diagram





*Website Purry's Cake and Bakery*. Diagram ini menunjukkan berbagai *use case* (fungsi atau fitur) yang tersedia di sistem, serta bagaimana aktor dapat mengaksesnya. Berikut adalah penjelasan elemen-elemen diagram:

a. Aktor *Customer*:

Representasi pelanggan yang menggunakan *website* untuk melakukan kegiatan seperti registrasi, *login*, melihat produk, memesan produk, dan *logout*.

b. *Use Case* untuk *Customer*:

1. *Register*: Pelanggan mendaftar ke sistem proses ini melibatkan mengisi *form* registrasi.
2. *Login*: Pelanggan masuk ke sistem, proses ini melibatkan *login* berhasil sebagai hasil *login* yang sukses.
3. *Melihat Produk*: Pelanggan dapat melihat produk yang tersedia.
4. *Memesan Produk*: Pelanggan dapat memesan produk.
5. *Include*: Proses ini melibatkan mengisi *form* pemesanan.
6. *Mengisi Form Pemesanan*: Bagian dari proses pemesanan produk, proses ini menghasilkan notifikasi terima kasih setelah selesai.
7. *Logout*: Pelanggan dapat keluar dari sistem, proses ini melibatkan *logout* berhasil.

## B. Pengembangan Sistem Menggunakan *Framework Laravel*

Dengan *Laravel*, sistem diperbarui untuk mendukung modularitas kode melalui penggunaan *MVC (Model-View-Controller)*. Fitur tambahan seperti autentikasi pengguna dan pengelolaan data dengan *Eloquent ORM* diimplementasikan untuk meningkatkan efisiensi.

### 1. Model

Menggunakan *Eloquent ORM* untuk mengelola database.

```
<?php
```

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
class Product extends Model
```

```
{
```

```
    use HasFactory;
```

```
    protected $fillable = [
```

```
        'name',
```

```
        'description',
```

```
        'price',
```

```
        'image',
```

```
        'stock',
```

```
        // Tambahkan atribut lain sesuai kebutuhan
```

```
    ];
```

```
}
```

Contoh: Model Produk mengelola tabel produk dengan relasi *hasMany* ke tabel transaksi.



## 2. View

*Blade template* digunakan untuk membuat antarmuka dinamis.

```
@extends('layouts.app')

@section('title', 'Beranda')

@section('content')

<main class="main">

<!-- Hero Section -->
<section id="hero" class="hero section light-background">

<div class="container">
<div class="row gy-4 justify-content-center justify-content-lg-between">
<div class="col-lg-5 order-2 order-lg-1 d-flex flex-column justify-content-center">
<h1 data-aos="fade-up">Selamat datang di <br>Pury's Cake & Bakery</h1>
<p data-aos="fade-up" data-aos-delay="100">Temukan berbagai macam roti yang
lezat dan berkualitas.</p>
<div class="d-flex" data-aos="fade-up" data-aos-delay="200">
</div>
</div>
<div class="col-lg-5 order-1 order-lg-2 hero-img" data-aos="zoom-out">

</div>
</div>
</div>

</section><!-- /Hero Section -->
@endsection
```

## 3. Controller

Mengatur logika aplikasi, seperti validasi *input* dan pengelolaan *request* pengguna.

```
<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use App\Models\admin\User;
use App\Models\admin\Product;
use App\Models\admin\Order;

class DashboardController extends Controller
{
    public function index()
    {
```



```
$user = auth()->user(); // Ambil data pengguna yang sedang login
$productCount = Product::count(); // Hitung jumlah produk
$orderCount = Order::count(); // Hitung jumlah pesanan
$userCount = User::count(); // Hitung jumlah pengguna

return view('admin.dashboard', compact('user', 'productCount', 'orderCount',
'userCount'));
}
}
```

### C. Pengujian Sistem

#### 1. Black Box Testing

Semua fitur diuji berdasarkan skenario pengguna. Hasil pengujian menunjukkan bahwa sistem berfungsi sesuai spesifikasi, tanpa kesalahan fungsional.

- Pengujian pada *customer*

Pengujian ini bertujuan untuk memastikan bahwa semua fitur yang diakses oleh pelanggan berfungsi sebagaimana mestinya.

Hasil pengujian dirangkum dalam tabel berikut:

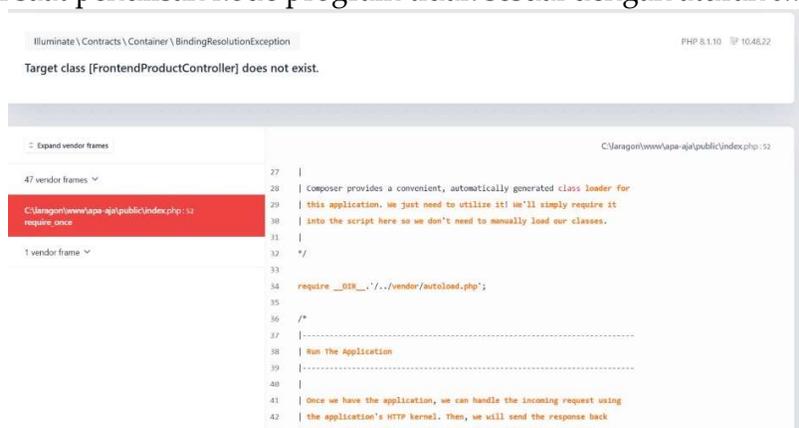
Table Black-Box Testing pengguna

No	Pengujian	Proses	Hasil
1	Registrasi <i>customer</i>	Frontend registrasi	✓
2	Login <i>customer</i>	Frontend Login	✓
3	Beranda <i>customer</i>	Frontend home	✓
4	Produk <i>customer</i>	Frontend produk index	✓
5	Pemesanan <i>customer</i>	Ordercreate	✓
6	Pesanan berhasil	Order <i>thankyou</i>	✓
7	Tentang kami <i>customer</i>	Frontend <i>about</i>	✓

#### 2. White Box Testing

- Kesalahan *sintaks*

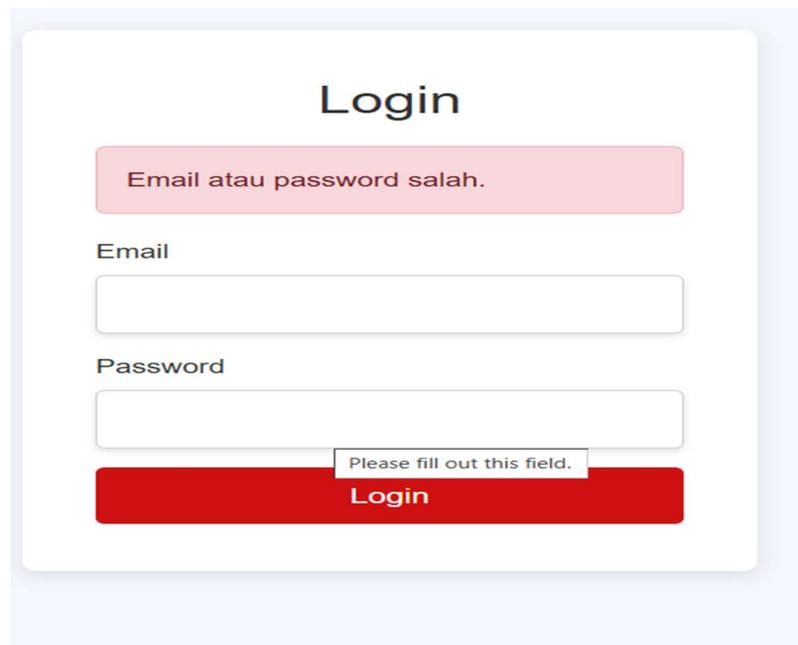
Kesalahan yang umum terjadi selama proses pengembangan sistem biasanya muncul saat penulisan kode program tidak sesuai dengan aturan *sintaks* yang berlaku.





- Kesalahan logika

Kesalahan logika terjadi ketika program berjalan namun tidak menghasilkan *output* sesuai dengan yang diharapkan. Gambar di bawah ini menunjukkan contoh kesalahan logika pada *form login* yang ditemukan pada *website Purry's cake and bakery* di mana pengguna melakukan *login* menggunakan 3 karakter yang tidak sesuai dengan logika program.



#### D. Perbandingan Sebelum dan Sesudah Menggunakan Laravel

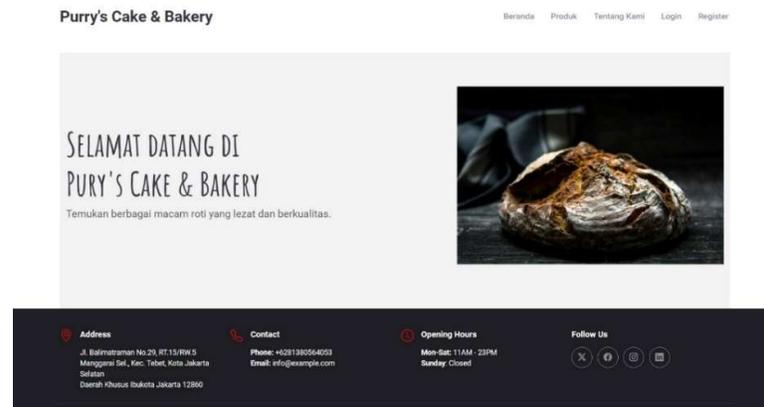
Aspek	Sebelum Laravel	Setelah Laravel
<b>Struktur Kode</b>	Tidak terorganisasi, sulit dipelihara.	Terorganisasi dengan MVC.
<b>Keamanan</b>	Tidak ada proteksi CSRF atau validasi.	CSRF <i>protection</i> dan validasi bawaan.
<b>Pengembangan</b>	Memakan waktu karena banyak kode manual.	Lebih cepat berkat <i>scaffolding</i> .
<b>Performa</b>	<i>Query</i> database kurang efisien.	<i>Query</i> lebih optimal dengan <i>Eloquent</i> .
<b>UI/UX</b>	Sederhana, kurang interaktif.	Lebih dinamis dengan <i>Blade Template</i> .



- Sebelum Menggunakan Laravel



- Sesudah Menggunakan Laravel



## KESIMPULAN DAN SARAN

### 1. Kesimpulan

Kesimpulannya, proses pengembangan sistem dilakukan melalui empat iterasi utama untuk memastikan fungsionalitas dan integrasi sistem berjalan optimal. Pada iterasi pertama, fokus diberikan pada pembuatan modul pemesanan produk untuk pelanggan dengan menghasilkan prototipe pemesanan *online*. Iterasi kedua mengembangkan *dashboard* admin untuk manajemen produk, termasuk fitur pengelolaan stok. Iterasi ketiga menambahkan fitur laporan penjualan otomatis untuk kebutuhan analisis. Terakhir, iterasi keempat memfinalisasi dan mengintegrasikan semua modul sekaligus melakukan pengujian akhir, menghasilkan sistem siap pakai yang mendukung operasional secara keseluruhan.

### 2. Saran

Untuk pengembang selanjutnya, disarankan untuk mengeksplorasi fitur-fitur Laravel yang lebih kompleks, seperti *API development* dan integrasi dengan teknologi *frontend* modern. Toko Roti Purry's Cake and Bakery diharapkan dapat terus memanfaatkan teknologi untuk mendukung pertumbuhan bisnis.



## REFERENSI

- Amelia, D., & Fajar, N. (2020). Studi Penggunaan *Framework* PHP pada Sistem Informasi. *Jurnal Teknologi dan Sistem Informasi*, 5(3), 89-95.
- Boehm, B. W. (1988). *A Spiral Model of Software Development and Enhancement*. ACM SIGSOFT Software Engineering Notes, 11(4), 14-24.
- Hakim, S., & Yulianto, D. (2022). Analisis Keamanan pada *Framework* Laravel dalam Pengembangan Aplikasi Web. *Jurnal Keamanan Siber dan Teknologi Informasi*, 3(2), 65-72.
- Kurniawan, R., Santoso, B., & Pratama, A. (2022). Implementasi Sistem Informasi Penjualan Berbasis Web untuk UKM. *Jurnal Sistem Informasi dan Teknologi*, 8(1), 23-34.
- Laudon, K. C., & Laudon, J. P. (2020). *Management Information Systems: Managing the Digital Firm*. Pearson Education.
- Nugroho, A. (2019). Analisis Perbandingan Kinerja *Framework* PHP dalam Pengembangan Web. *Jurnal Teknologi Komputer*, 15(2), 102-108.
- Rahman, M. (2021). Laravel: A Modern PHP *Framework* for Efficient Web Development. *International Journal of Computer Science and Applications*, 18(2), 45-56.
- Setiawan, A., & Harahap, S. (2021). Manfaat Teknologi Informasi untuk Efisiensi Usaha Mikro. *Jurnal Inovasi Bisnis dan Teknologi*, 12(1), 55-61.
- Suryadi, T. (2021). Model Pengembangan Aplikasi Web Menggunakan Metode Agile dan Spiral. *Jurnal Rekayasa Perangkat Lunak*, 10(4), 38-46.
- Widodo, T. (2020). Penerapan Teknologi Informasi dalam Pengelolaan Bisnis UMKM. *Jurnal Teknologi Informasi dan Bisnis*, 6(3), 125-132.