

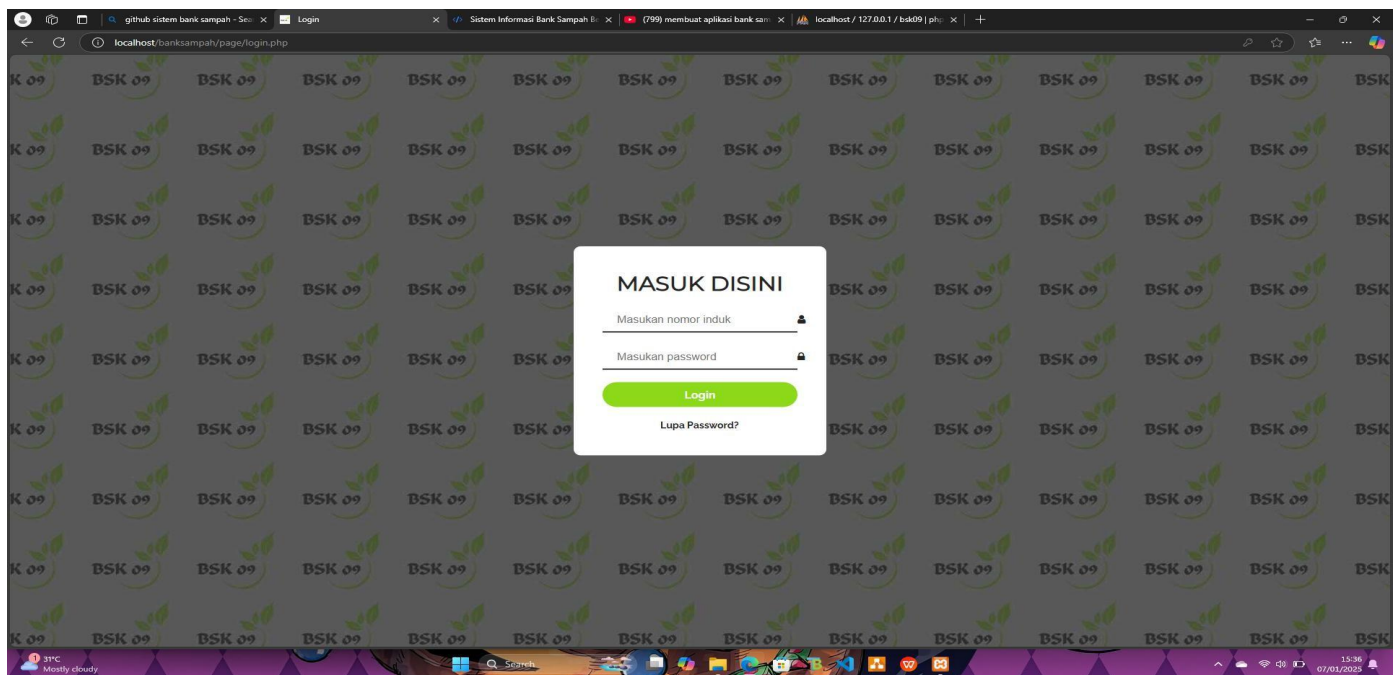
Nama : Djastiano Rivaldi

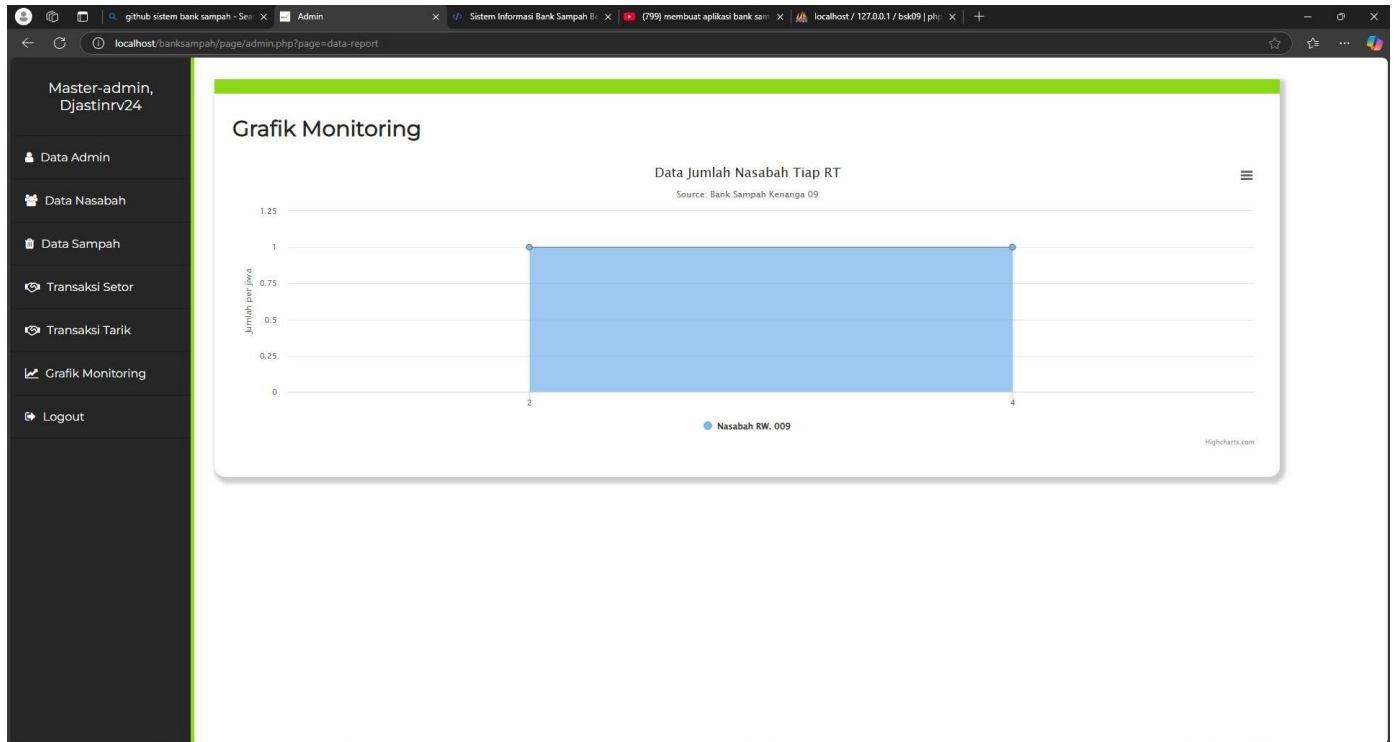
NIM 312210366

Kelas : TI.22.C5

Mata Kuliah : Pemrograman orientasi objek

## Tampilan Sistem





Master-admin, Djastinrv24

- Data Admin
- Data Nasabah
- Data Sampah
- Transaksi Setor
- Transaksi Tarik
- Grafik Monitoring
- Logout

### Transaksi Tarik Saldo

Show 10 entries

Search:

No	ID	Tanggal	NIN	Saldo	Jumlah Tarik	NIA	Aksi
1	11	2025-01-07	NSB1712002	Rp. 108.000,00	Rp. 10.000,00	admin123	<button>Edit</button> <button>hapus</button>
No	ID	Tanggal	NIN	Saldo	Jumlah Tarik	NIA	Aksi

Showing 1 to 1 of 1 entries

Tambah reset hapus

Previous 1 Next

Master-admin,  
Djastinrv24

Data Admin

Data Nasabah

Data Sampah

Transaksi Setor

Transaksi Tarik

Grafik Monitoring

Logout

## Transaksi Setor Sampah

Show  entries

ID	Tanggal	NIN	Jenis Sampah	Berat	Harga	Total	NIA	Aksi
26	0000-00-00	pnin	HVS	0 Kg	Rp. 9.000,00	Rp. 0,00	admin123	<a href="#">edit</a> <a href="#">hapus</a>
27	2025-01-07	NSB1712002	HVS	12 Kg	Rp. 9.000,00	Rp. 108.000,00	admin123	<a href="#">edit</a> <a href="#">hapus</a>

Showing 1 to 2 of 2 entries

[Tambah](#) [Excel](#) [Cetak](#)

Search:

Previous

1

Next

Master-admin,  
Djastinrv24

Data Admin

Data Nasabah

Data Sampah

Transaksi Setor




Transaksi Tarik

Grafik Monitoring

Logout

## Data Sampah

Show  entries

No	Jenis Sampah	Satuan	Harga	Gambar	Deskripsi	Aksi
1	HVS	KG	Rp. 9.000,00		Semua Jenis HVS	<a href="#">C:\xampp\htdocs\banksampah\system/function/view-sampah.php on line 57</a> <a href="#">edit</a> <a href="#">hapus</a>
2	kaleng	KG	Rp. 3.000,00		semua jenis kaleng	<a href="#">C:\xampp\htdocs\banksampah\system/function/view-sampah.php on line 57</a> <a href="#">edit</a> <a href="#">hapus</a>
3	Kardus	KG	Rp. 8.000,00		Semua Jenis Kardus	<a href="#">C:\xampp\htdocs\banksampah\system/function/view-sampah.php on line 57</a> <a href="#">edit</a> <a href="#">hapus</a>

Showing 1 to 3 of 3 entries

[Tambah](#) [Excel](#) [Cetak](#)

Search:

Previous

1

Next

Master-admin,  
Djastinrv24

Data Admin

Data Nasabah

Data Sampah

Transaksi Setor

Transaksi Tarik

Grafik Monitoring

Logout

Data Nasabah

Show  entries

Search:

NIN	Nama	RT	Alamat	Telepon	E-mail	Saldo	Sampah	Aksi
NSB1712001	Ihsmi	2	Jl. murai 8, C.45/4	085617287718	ihsmiica@gmail.com	Rp. 0,00	1 Kg	<a href="#">Edit</a> <a href="#">Hapus</a>
NSB1712002	Sabrina	4	Jl. murai 8, C.45/4	085617287718	sabrina123@gmail.com	Rp. 0,00	0 Kg	<a href="#">Edit</a> <a href="#">Hapus</a>

Showing 1 to 2 of 2 entries

[Tambah](#) [Excel](#) [Cetak](#)

Master-admin,  
Djastinrv24

Data Admin

Data Nasabah

Data Sampah

Transaksi Setor

Transaksi Tarik

Grafik Monitoring

Logout

Data Administrator

Nomor Induk Admin

admin123

Nama Admin

Djastinrv24

Nomor Telepon

085697107576

E-mail

djastinrv@gmail.com

Password

.....

Level Admin

Master-admin

Edit Data

## 1. Persiapan Lingkungan Implementasi

Persiapan lingkungan implementasi melibatkan penyiapan perangkat keras (hardware), perangkat lunak (software), serta infrastruktur jaringan yang diperlukan untuk pengembangan dan implementasi sistem. Berikut langkah-langkahnya:

- **Perangkat Keras:** Menentukan spesifikasi server, komputer, atau perangkat lainnya yang mendukung sistem.
  - **Perangkat Lunak:** Menginstal software pendukung seperti database (MySQL, PostgreSQL), bahasa pemrograman (PHP, Python, atau JavaScript), framework (Laravel, Django, Node.js), dan tools pengembangan.
  - **Infrastruktur Jaringan:** Menyiapkan koneksi internet yang stabil untuk akses sistem.
  - **Lingkungan Pengembangan:** Mengatur repository (GitHub/GitLab), IDE (VS Code, IntelliJ), dan containerization (Docker jika diperlukan).
  - **Akses dan Perizinan:** Menentukan pengguna yang memiliki hak akses selama pengembangan sistem.
- 

## 2. Penerjemahan Desain ke Kode (Coding)

Tahap ini mencakup implementasi kode berdasarkan desain sistem dan arsitektur yang telah dirancang:

- **Front-End Development:** Membuat antarmuka pengguna (User Interface) dengan HTML, CSS, dan JavaScript atau framework seperti React atau Vue.js.
- **Back-End Development:** Mengimplementasikan logika bisnis, pengolahan data, dan komunikasi dengan database menggunakan bahasa pemrograman seperti Python (Django/Flask), PHP (Laravel), atau Node.js.
- **Database Development:** Membuat skema database sesuai dengan Entity Relationship Diagram (ERD) yang telah dirancang, mengelola tabel, relasi, dan prosedur penyimpanan (stored procedure).
- **API Integration:** Membuat dan menghubungkan API untuk integrasi antara front-end, back-end,

atau layanan eksternal seperti pembayaran digital.

- **Keamanan Sistem:** Menerapkan enkripsi data (misalnya dengan Hill Cipher atau AES) pada data sensitif seperti transaksi dan informasi nasabah.

### 3. Integrasi Komponen

Pada tahap ini, semua komponen yang dikembangkan secara terpisah diintegrasikan menjadi satu kesatuan sistem:

- **Integrasi Front-End dan Back-End:** Menghubungkan antarmuka pengguna dengan logika bisnis melalui API atau framework.
- **Pengintegrasian Database:** Memastikan data yang dikirimkan oleh aplikasi tersimpan dan diproses dengan benar di database.
- **Pengujian Fungsionalitas:** Melakukan uji coba untuk memastikan setiap modul bekerja sesuai dengan fungsinya setelah diintegrasikan.
- **Pengelolaan Dependencies:** Mengelola dependensi library atau framework untuk memastikan kompatibilitas antar komponen.

### 4. Pengujian Sistem (Testing)

Pengujian sistem dilakukan untuk memastikan bahwa sistem berjalan dengan baik tanpa error dan sesuai kebutuhan pengguna:

- **Unit Testing:** Menguji setiap komponen individu untuk memastikan tidak ada kesalahan pada fungsi tertentu.
- **Integration Testing:** Menguji modul yang telah diintegrasikan untuk memastikan antar komponen bekerja dengan baik.
- **System Testing:** Menguji keseluruhan sistem untuk melihat apakah sistem berfungsi sesuai spesifikasi.
- **User Acceptance Testing (UAT):** Melibatkan pengguna akhir untuk memastikan sistem memenuhi kebutuhan mereka.
- **Keamanan Sistem:** Menguji keamanan sistem, seperti perlindungan data pengguna, manajemen otentikasi, dan enkripsi.

### 5. Deployment ke Lingkungan Produksi

Setelah sistem lulus pengujian, langkah berikutnya adalah menerapkan sistem ke lingkungan produksi:

- **Penyediaan Server Produksi:** Menyiapkan server produksi yang sesuai spesifikasi (cloud server seperti AWS, GCP, atau server lokal).
- **Migrasi Data:** Memindahkan data dari server pengembangan ke server produksi.



- **Pengaturan DNS:** Menyediakan domain dan SSL untuk keamanan.
- **Konfigurasi Sistem:** Mengatur aplikasi agar siap berjalan di server produksi, termasuk mengonfigurasi load balancing atau backup.
- **Monitoring Awal:** Memantau sistem setelah di-deploy untuk memastikan tidak ada gangguan atau bug yang signifikan.

## 6. Evaluasi dan Pemeliharaan

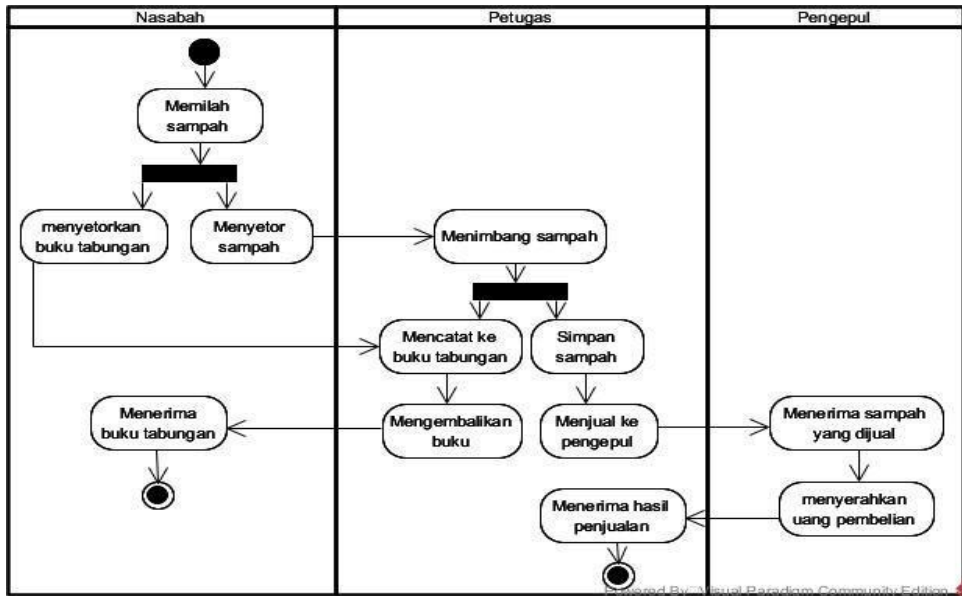
Setelah sistem berjalan di lingkungan produksi, evaluasi dan pemeliharaan dilakukan untuk menjaga stabilitas dan meningkatkan sistem:

- **Evaluasi Sistem:** Mengumpulkan feedback dari pengguna untuk mengetahui kekurangan sistem.
- **Perbaikan Bug:** Memperbaiki error atau bug yang ditemukan selama sistem digunakan.
- **Peningkatan Fitur:** Menambahkan fitur baru sesuai kebutuhan yang muncul.
- **Pemeliharaan Rutin:** Melakukan backup data, pembaruan keamanan, dan pembaruan perangkat lunak.
- **Monitoring Performa:** Memantau penggunaan sistem secara berkala, termasuk kecepatan respon dan kapasitas server.



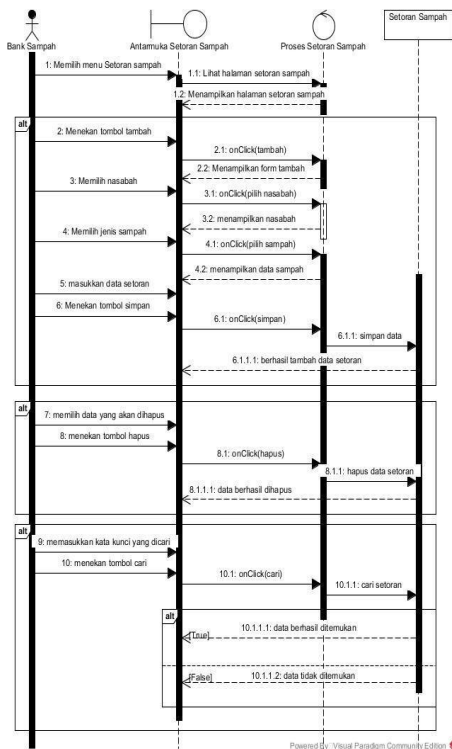
- Activity Diagram Penyetoran Sampah

- Activity Diagram Transaks

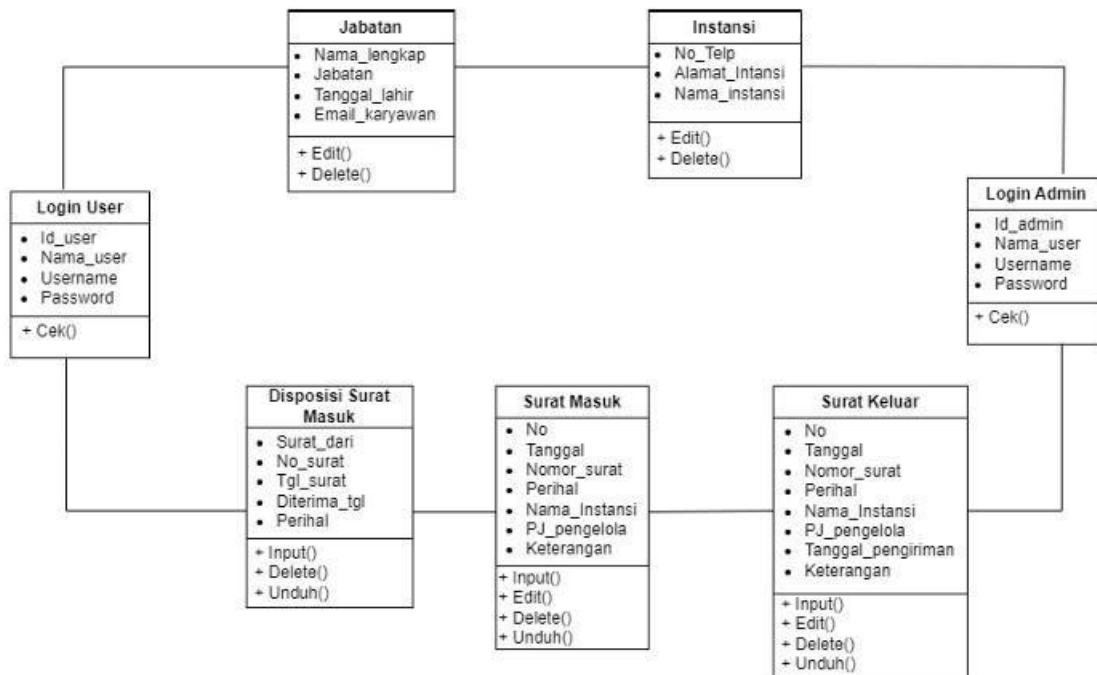


## #Sequence Diagram

- ◆ Sequence Diagram



## # Class Diagram



## 1. Evaluasi Performa Sistem

Evaluasi performa sistem dilakukan untuk menilai apakah sistem bekerja sesuai harapan. Fokus pada:

- **Kinerja Sistem:** Mengukur kecepatan respon aplikasi, waktu pemrosesan data, dan kemampuan menangani beban kerja (load).
- **Stabilitas Sistem:** Mengevaluasi frekuensi downtime dan error yang terjadi.
- **Keandalan:** Memastikan sistem dapat beroperasi secara konsisten dalam berbagai kondisi.
- **Kepuasan Pengguna:** Mengumpulkan umpan balik dari pengguna terkait kemudahan navigasi, kecepatan akses, dan kelengkapan fitur.
- **Analisis Data Penggunaan:** Menganalisis log penggunaan untuk melihat pola trafik, pengguna aktif, dan interaksi dalam sistem.

## 2. Identifikasi Masalah dan Pemeliharaan Korektif

Pemeliharaan korektif dilakukan untuk memperbaiki masalah atau bug yang ditemukan setelah sistem diterapkan.

- **Pengumpulan Data Masalah:** Melalui laporan pengguna, log error, atau hasil monitoring sistem.
- **Analisis Akar Masalah:** Mengidentifikasi penyebab utama dari masalah, seperti kesalahan kode, konfigurasi yang salah, atau batasan infrastruktur.
- **Perbaikan:** Mengupdate kode atau konfigurasi yang bermasalah dan menguji ulang hasil perbaikan.
- **Dokumentasi:** Mendokumentasikan setiap masalah dan solusi yang diterapkan untuk referensi di masa depan.

## 3. Pemeliharaan Adaptif

Pemeliharaan adaptif bertujuan untuk menyesuaikan sistem dengan perubahan kebutuhan pengguna, teknologi, atau regulasi.

- **Pembaruan Teknologi:** Mengadopsi teknologi terbaru seperti pembaruan framework, library, atau database.
- **Penyesuaian Regulasi:** Menyesuaikan sistem dengan peraturan baru, misalnya kebijakan perlindungan data atau pajak.

- **Integrasi Fitur Baru:** Menambahkan kemampuan baru yang dibutuhkan pengguna, seperti metode pembayaran baru atau dashboard pelaporan.
- **Penyesuaian Infrastruktur:** Memperbarui kapasitas server atau jaringan untuk menangani peningkatan pengguna.

#### 4. Pemeliharaan Perfektif

Pemeliharaan perfektif dilakukan untuk meningkatkan performa dan efisiensi sistem tanpa adanya masalah spesifik.

- **Optimasi Kode:** Menyederhanakan atau mempercepat eksekusi kode tanpa mengubah fungsi sistem.
- **Peningkatan UI/UX:** Mengubah antarmuka agar lebih intuitif dan menarik.
- **Efisiensi Database:** Mengoptimalkan query database atau menerapkan indexing untuk mempercepat akses data.
- **Peningkatan Dokumentasi:** Memperbaiki dokumentasi teknis agar lebih mudah dipahami untuk tim pengembang.



## 5. Pemeliharaan Preventif

Pemeliharaan preventif bertujuan untuk mencegah masalah yang mungkin terjadi di masa depan.

- **Pembaruan Keamanan:** Mengaplikasikan patch keamanan untuk melindungi sistem dari serangan cyber.
- **Backup Data:** Melakukan backup data secara berkala untuk mencegah kehilangan data akibat kegagalan sistem.
- **Pemeriksaan Infrastruktur:** Mengevaluasi kapasitas server, bandwidth jaringan, dan penggunaan sumber daya lainnya.
- **Simulasi Kegagalan:** Melakukan pengujian seperti stress test untuk memastikan sistem mampu bertahan dalam kondisi ekstrim.

## 6. Pengujian Kembali

Setelah dilakukan perbaikan atau pemeliharaan, pengujian ulang perlu dilakukan untuk memastikan sistem tetap berfungsi dengan baik.

- **Regression Testing:** Menguji apakah fitur yang sudah ada sebelumnya tidak terpengaruh oleh perubahan yang dilakukan.
- **Performance Testing:** Memastikan sistem tetap cepat dan responsif setelah perubahan.
- **Security Testing:** Memeriksa kembali aspek keamanan setelah pembaruan atau perbaikan.
- **Acceptance Testing:** Meminta pengguna untuk mengevaluasi kembali sistem dan memberikan umpan balik.

## 7. Monitoring dan Evaluasi Berkala

Monitoring dan evaluasi sistem dilakukan secara rutin untuk memastikan sistem tetap berjalan optimal.

- **Monitoring Performa:** Menggunakan tools seperti Google Analytics, Grafana, atau New Relic untuk memantau performa sistem secara real-time.
- **Log Error:** Mengumpulkan log error untuk mendeteksi masalah yang terjadi secara diam-diam.

- **Evaluasi Berkala:** Mengadakan pertemuan bulanan atau triwulanan dengan tim pengembang dan pengguna untuk mengevaluasi sistem.
- **Laporan Kinerja:** Membuat laporan berkala tentang penggunaan sistem, performa, dan masalah yang ditemukan.



