**Laporan Pengembangan Fitur: Penambahan Halaman Input Limbah pada Aplikasi General Application Control (GAC)**

**Tanggal:** 03/07/2024

**Penulis :** Deden

**Daftar Isi**

1. **Latar Belakang**
2. **Tujuan Pengembangan**
3. **Spesifikasi Fitur**
   1. Halaman Input Limbah
   2. Keamanan dan Privasi
4. **Desain Antarmuka Pengguna (UI)**
   1. Mockup dan Alur Pengguna
5. **Teknologi yang Digunakan**
   1. Framework Backend
   2. Frontend
   3. Database
   4. Keamanan
6. **Implementasi**
   1. Langkah-langkah Pengembangan
7. **Manfaat bagi Pengguna**
8. **Kesimpulan**

**1. Latar Belakang**

Saat ini, proses input data limbah dilakukan secara manual menggunakan Google Sheets. Metode ini memiliki beberapa kelemahan:

* Kapasitas Penyimpanan: Data limbah yang terus bertambah membuat kapasitas penyimpanan di Google Sheets menjadi penuh, menyebabkan perlambatan dalam proses penyimpanan dan pengambilan data.
* Pengelolaan Data yang Rumit: Dengan volume data yang besar, pengelolaan dan analisis data menjadi semakin sulit dan tidak efisien.
* Keterbatasan Fitur: Google Sheets, meskipun berguna, memiliki keterbatasan dalam hal fitur dan fungsionalitas yang diperlukan untuk pengelolaan data limbah yang lebih kompleks dan dinamis.

**2. Tujuan Pengembangan**

Pengembangan halaman input limbah pada aplikasi GAC bertujuan untuk:

* Mempermudah Input Data: Memfasilitasi proses input data limbah sehingga lebih cepat dan efisien.
* Meningkatkan Kapasitas Penyimpanan: Mengurangi ketergantungan pada Google Sheets dan menghindari masalah kapasitas penyimpanan.
* Memperbaiki Pengelolaan Data: Menyediakan alat dan antarmuka yang lebih baik untuk mengelola, mengolah, dan menganalisis data limbah.
* User-Friendly: Menciptakan pengalaman pengguna yang lebih intuitif dan nyaman dalam mengelola data limbah.

**3. Spesifikasi Fitur**

**Halaman Input Limbah:**

* Formulir Input Data: Pengguna dapat memasukkan data limbah dengan berbagai atribut seperti jenis limbah, jumlah, tanggal, lokasi, dan catatan tambahan.
* Validasi Data: Sistem akan melakukan validasi terhadap input data untuk memastikan konsistensi dan akurasi.
* Upload File: Opsi untuk mengunggah dokumen atau foto terkait dengan data limbah.
* Pencarian dan Filter: Pengguna dapat mencari dan memfilter data limbah yang telah dimasukkan untuk analisis dan pelaporan yang lebih mudah.

**Keamanan dan Privasi:**

* Autentikasi Pengguna: Sistem memerlukan autentikasi untuk mengakses halaman input limbah, memastikan hanya pengguna yang berwenang yang dapat memasukkan atau mengedit data.
* Pengelolaan Hak Akses: Pengaturan hak akses untuk berbagai tingkat pengguna (admin, user biasa, dll.).

**4. Desain Antarmuka Pengguna (UI)**

**Mockup dan Alur Pengguna**

1. **Halaman Login:**
   * Pengguna memasukkan kredensial mereka (nama pengguna dan kata sandi) untuk masuk ke dalam sistem.
   * Setelah login berhasil, pengguna akan diarahkan ke halaman utama.
2. **Halaman Utama:**
   * Menampilkan menu atau navigasi yang berisi berbagai opsi, termasuk tombol untuk menuju ke halaman input limbah.
   * Menyediakan akses cepat ke halaman-halaman lain yang relevan, seperti statistik dan ringkasan data limbah terbaru.
   * Pengguna biasa akan melihat menu input limbah, sementara admin akan memiliki akses ke menu pengelolaan data.
3. **Halaman Input Limbah:**
   * Pengguna dapat memasukkan data limbah dengan berbagai atribut seperti jenis limbah, jumlah, tanggal, lokasi, dan catatan tambahan.
   * Sistem menyediakan area untuk mengunggah file terkait seperti dokumen atau foto.
   * Setelah data diinput, pengguna dapat menyimpan atau memperbarui informasi limbah.
   * Tombol tindakan untuk menyimpan, menghapus, atau memperbarui data disediakan di halaman ini.
4. **Halaman Pengelolaan Data (Admin):**
   * Admin login dan mengakses menu "Data Limbah" untuk melihat daftar data limbah yang telah diinput oleh pengguna.
   * Admin dapat memeriksa apakah ada input limbah baru yang perlu diproses.
   * Sistem menyediakan opsi untuk mencari, memfilter, dan mengelompokkan data limbah yang telah diinput.
   * Admin dapat melakukan tindakan seperti memvalidasi, mengedit, atau menghapus entri data limbah.
   * Opsi untuk mengekspor data ke berbagai format seperti CSV atau Excel tersedia untuk keperluan pelaporan dan analisis lebih lanjut.

**5. Teknologi yang Digunakan**

**Framework Backend:**

* CodeIgniter 3: Digunakan untuk pengembangan backend dan pengelolaan logika aplikasi. CodeIgniter 3 dipilih karena kesederhanaan dan performanya yang baik.

**Frontend:**

* HTML dan CSS: Untuk membangun struktur dan gaya halaman.
* Bootstrap: Digunakan untuk membuat antarmuka yang responsif dan modern dengan komponen yang siap pakai seperti form, tabel, dan layout grid.
* JavaScript: Untuk interaktivitas tambahan, seperti validasi form di sisi klien.

**Database:**

* MySQL: Digunakan untuk menyimpan data limbah secara efisien dan mendukung query yang kompleks untuk pengelolaan data.

**Keamanan:**

* Autentikasi Pengguna: Menggunakan mekanisme autentikasi bawaan CodeIgniter dan session management untuk memastikan hanya pengguna yang sah yang dapat mengakses fitur tertentu.

**6. Implementasi**

**Langkah-langkah Pengembangan:**

1. Analisis Kebutuhan: Mengumpulkan detail kebutuhan dari pengguna dan pemangku kepentingan.
2. Desain UI/UX: Membuat desain antarmuka pengguna yang ramah dan mudah digunakan dengan Bootstrap.
3. Pengembangan Frontend: Membangun halaman input limbah menggunakan HTML, CSS, dan Bootstrap untuk tampilan dan JavaScript untuk interaktivitas.
4. Pengembangan Backend: Membangun logika bisnis menggunakan CodeIgniter 3. Mengatur koneksi ke database MySQL dan mengimplementasikan logika penyimpanan dan pengambilan data.
5. Integrasi: Menggabungkan frontend dan backend sehingga halaman input limbah dapat berfungsi penuh.
6. Pengujian: Melakukan pengujian untuk memastikan semua fitur berjalan dengan baik dan sesuai kebutuhan.
7. Peluncuran dan Pelatihan: Meluncurkan fitur baru dan memberikan pelatihan kepada pengguna tentang cara menggunakan halaman input limbah.

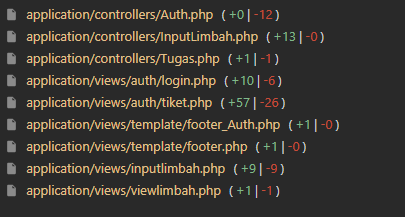
**7. Manfaat bagi Pengguna**

* Kemudahan dan Kecepatan: Proses input data limbah menjadi lebih cepat dan mudah, mengurangi beban kerja manual.
* Pengelolaan yang Lebih Baik: Memungkinkan pengelolaan dan analisis data limbah yang lebih baik dan efisien.
* Penghematan Ruang Penyimpanan: Mengurangi penggunaan ruang penyimpanan di Google Sheets.
* Fitur yang Lebih Kaya: Memberikan lebih banyak fitur dan fungsionalitas dibandingkan dengan Google Sheets.

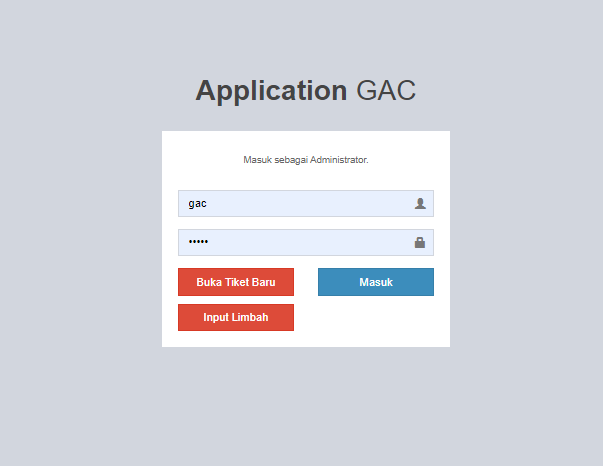
**8. Kesimpulan**

Penambahan halaman input limbah pada aplikasi GAC menggunakan CodeIgniter 3 dan Bootstrap adalah langkah strategis untuk meningkatkan efisiensi dan efektivitas dalam pengelolaan data limbah. Dengan mengadopsi solusi ini, pengguna akan merasakan manfaat dari proses yang lebih terorganisir, mudah diakses, dan dapat diskalakan.

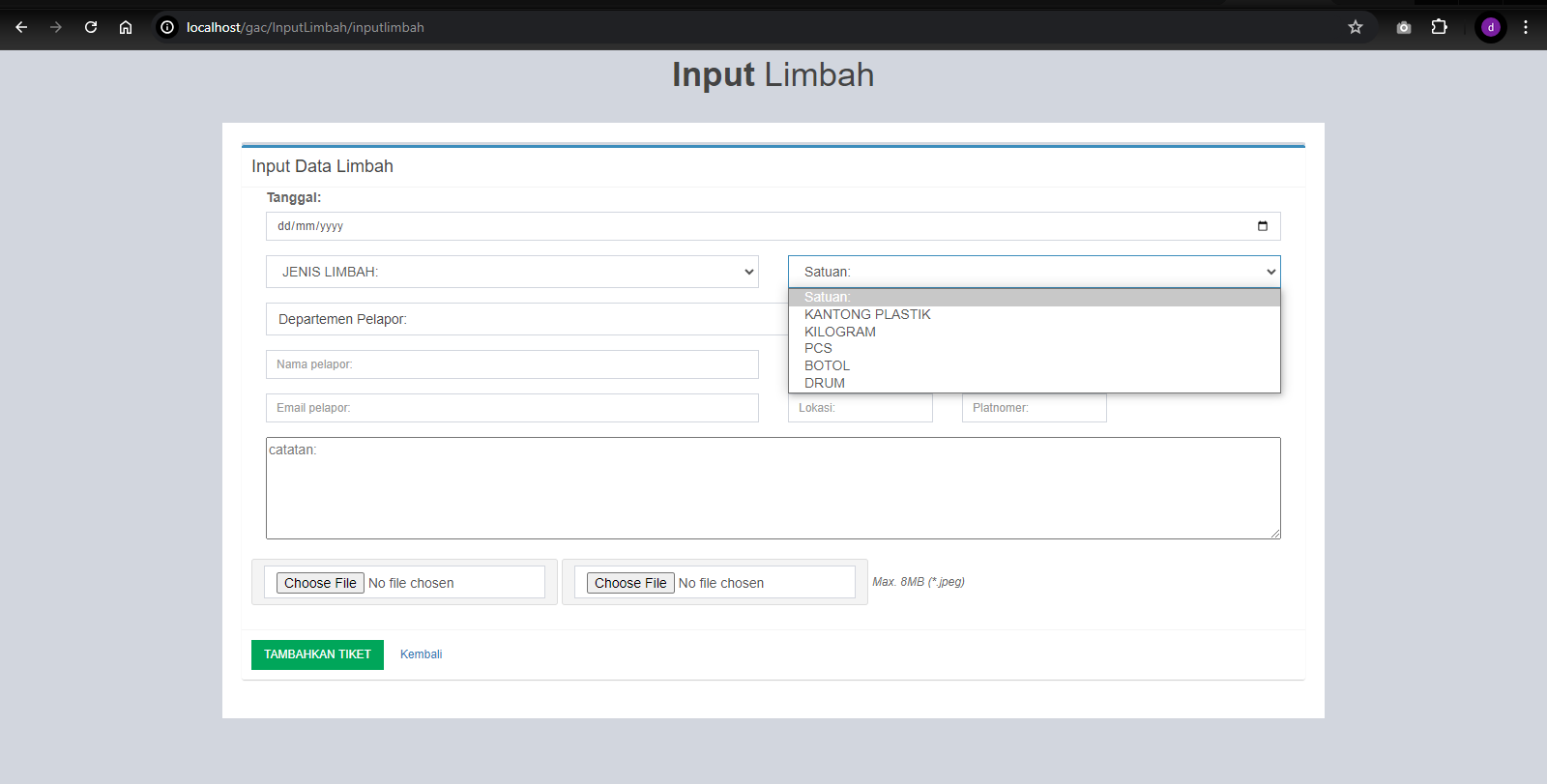
TERIMAKASIH

File yang saya ubah   


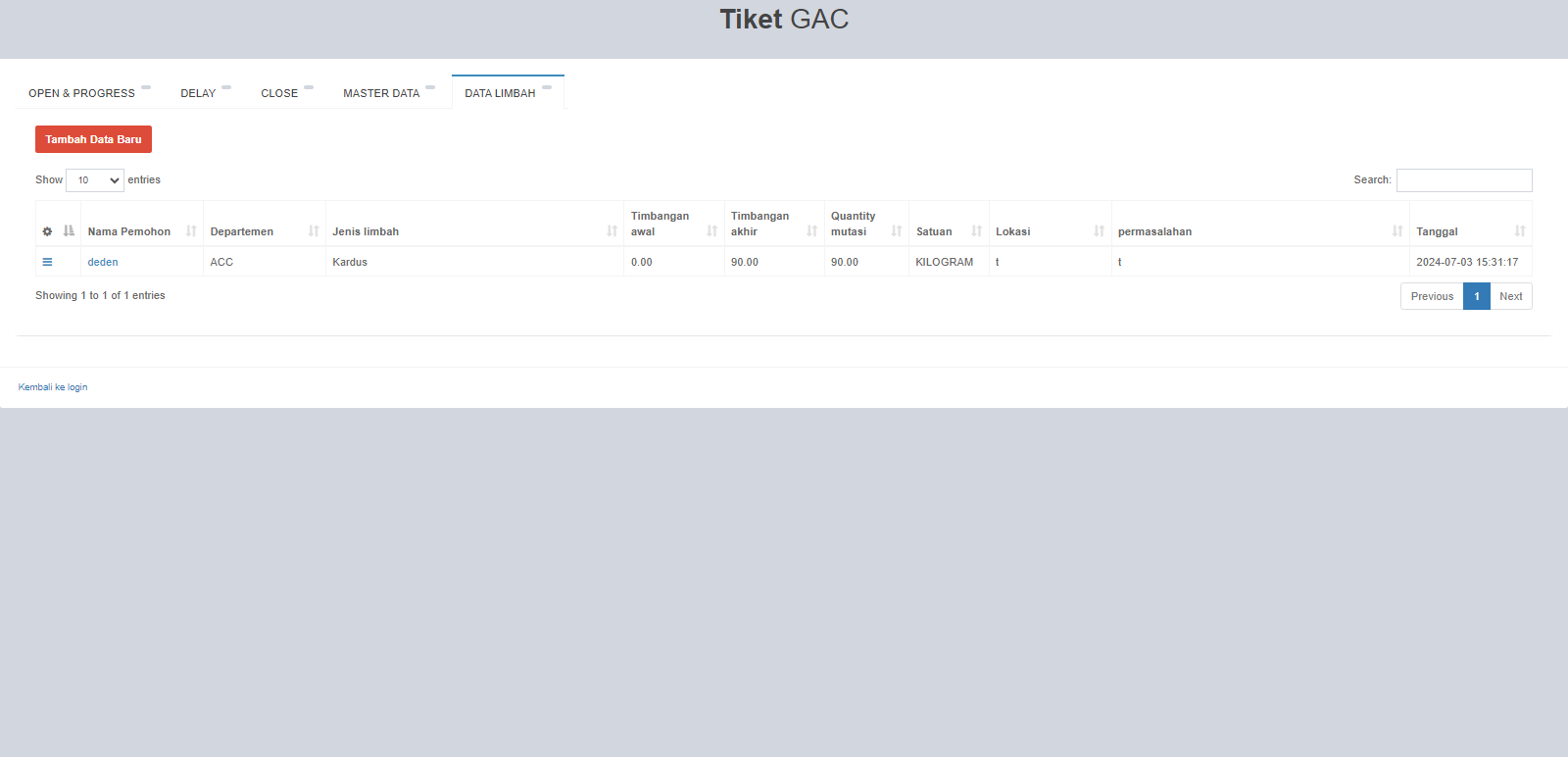
HALAMAN LOGIN DAN BUTTON INPUT LIMBAH



HALAMAN INPUT LIMBAH



HALAMAN DATA LIMBAH YANG TERINPUT



HALAMAN EDIT DATA LIMBAH

