

DOKUMEN SKENARIO (USE CASE DOCUMENT)

Sistem Pengaduan Masyarakat (CodeIgniter)

Detail Proyek

Detail	Keterangan
Proyek	Pengaduan Masyarakat
Aktor Utama	Masyarakat, Petugas dan Admin
Platform	Web (CodeIgniter)

1. Use Case: Ajukan Pengaduan

Use Case Name	Ajukan Pengaduan
Scenario	Masyarakat mengisi laporan keluhan atau aspirasi dengan menyertakan bukti foto untuk disimpan dalam sistem.
Triggering Event	Masyarakat menekan tombol "Ajukan Laporan".
Brief Description	Sistem menampilkan formulir input. Masyarakat mengisi laporan dan wajib mengunggah foto bukti. Setelah disimpan, laporan masuk ke database dengan status "Pending".
Actors	Masyarakat
Related Use Case	Include: Upload Foto Bukti
Precondition	Masyarakat sudah melakukan Login dan berada di halaman beranda/pengaduan.
Postconditions	Laporan tersimpan di database dan muncul di daftar riwayat pengaduan milik masyarakat.

Flow Of Activities	<p>Actor</p> <ol style="list-style-type: none"> 1. Masyarakat menekan tombol "Ajukan Pengaduan". 3. Masyarakat mengisi semua <i>field</i> (Judul Laporan, Isi Laporan, dan Kategori) 4. Masyarakat mengunggah foto sebagai bukti laporan (Fungsi <i>Include</i>). 5. Masyarakat menekan tombol "Simpan". <p>System</p> <ol style="list-style-type: none"> 2. Sistem menampilkan <i>Form Tambah Pengaduan</i>. 6. Sistem menerima input, menyimpan data ke database, dan menampilkan <i>dashboard</i> utama dengan notifikasi sukses.
Exception Conditions	<p>5.1 Gagal Simpan (Validasi Kosong): Jika terdapat <i>field</i> (seperti Isi Laporan atau Kategori) yang kosong atau foto bukti belum diunggah, sistem akan menampilkan notifikasi kesalahan: "Laporan dan Foto Bukti wajib diisi sebelum mengirim" dan pengguna tetap berada di halaman form tersebut.</p>

2. Use Case: Beri Tanggapan

Use Case Name	Beri Tanggapan
Scenario	Petugas menanggapi laporan yang masuk dan secara otomatis merubah status pengaduan tersebut.
Triggering Event	Petugas menekan tombol "Tanggapi" pada baris data pengaduan yang dipilih.
Brief Description	Petugas menuliskan respon atau solusi atas aduan masyarakat. Saat tanggapan disimpan, sistem otomatis menjalankan fungsi update status.
Actors	Petugas
Related Use Case	Include: Update Status Pengaduan

Precondition	Petugas sudah Login dan terdapat data pengaduan yang masuk.
Postconditions	Tanggapan tersimpan dan status pengaduan berubah (misal: dari 'Proses' menjadi 'Selesai').
Flow Of Activities	<p>Actor</p> <ol style="list-style-type: none"> 1. Petugas menekan tombol "Tanggapi" pada baris data pengaduan yang dipilih. 3. Petugas mengisi <i>field</i> (Isi Tanggapan/Respon). 4. Petugas menekan tombol "Kirim Tanggapan". <p>System</p> <ol style="list-style-type: none"> 2. Sistem menampilkan <i>Form Tanggapan</i> beserta detail aduan masyarakat. 5. Sistem menerima input, menyimpan tanggapan ke database. 6. (Include) Sistem secara otomatis mengubah Status Pengaduan menjadi "Selesai" atau "Terverifikasi". 7. Sistem kembali ke halaman daftar aduan dan menampilkan notifikasi sukses.
Exception Conditions	<p>4.1 Gagal Validasi (Isi Kosong): Jika petugas menekan tombol "Kirim Tanggapan" namun kolom teks tanggapan masih kosong, sistem akan menampilkan notifikasi kesalahan: "Isi tanggapan tidak boleh kosong sebelum menyelesaikan aduan" dan tetap berada di halaman form tersebut.</p> <p>4.2 Gagal Update Status: Jika koneksi database terputus saat proses perubahan status (poin 6), sistem akan membatalkan penyimpanan tanggapan dan menampilkan pesan "Gagal memperbarui status pengaduan, silakan coba lagi".</p>

3. Use Case: Kelola Kategori Pengaduan

Use Case Name	Kelola Kategori Pengaduan
Scenario	Admin menambah, mengubah, atau menghapus kategori (seperti: Kebersihan, Infrastruktur, dll) yang tersedia di sistem.
Triggering Event	Admin memilih menu "Kategori Pengaduan" di dashboard.

Brief Description	Admin melakukan manajemen data kategori agar masyarakat bisa memilih kategori yang sesuai saat mengadu.
Actors	Admin
Related Use Case	Include: Login.
Precondition	Admin sudah Login ke Dashboard Admin.
Postconditions	Perubahan data kategori (Tambah/Edit/Hapus) tersimpan di database.

Flow Of Activities	<p>Actor</p> <ol style="list-style-type: none"> 1. Admin menekan tombol "Tambah Kategori" atau "Edit" pada daftar kategori. 3. Admin mengisi atau mengubah nama kategori (misal: Infrastruktur, Kebersihan) 4. Admin menekan tombol "Simpan" atau "Update Data". Admin menekan tombol "Update Data". <p>System</p> <ol style="list-style-type: none"> 2. Sistem menampilkan <i>Form Kategori</i> (berisi Nama Kategori dan Deskripsi). 5. Sistem menerima input, menyimpan/memperbarui data ke database. 6. Sistem kembali ke halaman utama kategori dan menampilkan notifikasi sukses.
Exception Conditions	<p>4.1 Gagal Validasi (Isi Kosong): Jika Admin mengosongkan <i>field</i> Nama Kategori, sistem akan menampilkan notifikasi kesalahan "Nama kategori wajib diisi" dan tetap berada di halaman form.</p> <p>4.2 Data Duplikat: Jika Nama Kategori yang dimasukkan sudah ada di database, sistem akan menampilkan notifikasi "Kategori sudah tersedia" agar tidak terjadi penggandaan data.</p>