Pengecekan Data

Panduan Modul: Pengecekan Data

Lokasi Modul
Pengaturan > Peralatan > Pengecekan Data

Tujuan Modul
Modul **Pengecekan Data** adalah sebuah konsol *database query* yang sangat kuat dan berisiko tinggi. Fitur ini memberikan akses langsung ke *database* yang mendasari aplikasi, memungkinkan pengguna teknis untuk menjalankan perintah SQL (*Structured Query Language*) secara langsung. Modul ini bukanlah untuk pengguna bisnis, melainkan sebuah peralatan khusus untuk Administrator *Database* (DBA), pengembang, atau tim dukungan teknis tingkat lanjut untuk tujuan diagnosis masalah data yang kompleks, perbaikan data, dan pembuatan laporan *ad-hoc*.

1. Tampilan Utama (Konsol Query)

Halaman utama adalah sebuah antarmuka yang terbagi menjadi beberapa bagian untuk menulis, menjalankan, dan mengelola *query* database.

Penjelasan Tampilan

- **Query Untuk Database**:
    - Ini adalah area utama di mana pengguna menulis perintah SQL.

    - **Dropdown Database**: Memungkinkan pengguna memilih *database* target di mana *query* akan dijalankan (contoh: kokola).
    - **Tombol Aksi Query**:
        - `Jalankan`: Mengeksekusi perintah SQL yang telah ditulis.
        - `Bersihkan`: Menghapus teks dari area *query*.
        - `Simpan Sebagai`: Menyimpan *query* yang sering digunakan sebagai templat untuk penggunaan di masa depan.
- **Query Templates**:
    - Area di sebelah kanan yang menampilkan daftar *query* yang telah disimpan. Ini memungkinkan pengguna untuk dengan cepat memuat ulang dan menjalankan kembali *query* yang kompleks tanpa harus mengetiknya ulang.
- **Area Hasil**:
    - Bagian bawah layar (saat ini kosong) adalah tempat di mana hasil dari eksekusi *query* (terutama perintah `SELECT`) akan ditampilkan dalam bentuk tabel.

2. Langkah-langkah Penggunaan

1. **Pilih Database**: Pilih *database* target dari *dropdown* di bagian atas.

2. **Tulis Query**: Ketikkan perintah SQL langsung di area teks "Query Untuk Database".

    - **Untuk Melihat Data**: Gunakan perintah `SELECT`. Contoh: `SELECT * FROM m_customer WHERE customer_city = 'Jakarta';`

    - **Untuk Mengubah Data (Sangat Berbahaya!)**: Gunakan perintah `UPDATE`. Contoh: `UPDATE m_item SET is_active = 0 WHERE item_code = 'XYZ';`

3. **Eksekusi Query**: Klik tombol **Jalankan**.

4. **Lihat Hasil**: Jika menggunakan `SELECT`, hasilnya akan muncul di area bawah. Jika menggunakan `UPDATE` atau `DELETE`, sistem akan memberikan pesan konfirmasi jumlah baris yang terpengaruh.

5. **Simpan Template (Opsional)**: Jika *query* tersebut akan sering digunakan, klik **Simpan Sebagai** untuk menyimpannya di daftar "Query Templates".

3. Alur Kerja & Proses Bisnis Terintegrasi

- **Alat Diagnostik**: Ketika terjadi anomali data yang tidak dapat dijelaskan melalui antarmuka pengguna standar (misalnya, saldo yang tidak cocok, transaksi yang hilang dari laporan), tim teknis akan menggunakan modul ini untuk "melihat langsung" ke dalam tabel *database* dan mendiagnosis akar masalahnya.

- **Perbaikan Data Darurat**: Dalam kasus di mana terjadi kerusakan data (*data corruption*) atau kesalahan input massal, modul ini adalah satu-satunya cara untuk melakukan perbaikan data secara langsung dan efisien.

- **Tidak untuk Pengguna Bisnis**: Modul ini secara desain tidak terintegrasi dengan alur kerja bisnis sehari-hari. Ini adalah peralatan "di belakang layar" yang digunakan untuk memelihara dan memperbaiki sistem, bukan untuk menjalankan bisnis.

4. Peringatan Kritis & Tips Keamanan

- **RISIKO SANGAT TINGGI**: Ini adalah modul paling berbahaya di seluruh sistem. Satu perintah `UPDATE` atau `DELETE` yang salah tanpa klausa `WHERE` yang tepat dapat **menghapus atau merusak seluruh data perusahaan secara permanen dan tidak dapat diperbaiki**.

- **AKSES SUPER TERBATAS**: Hak akses ke modul ini harus dibatasi secara sangat ketat dan hanya diberikan kepada personel teknis yang sangat terlatih dan tepercaya, seperti Administrator *Database* (DBA) atau Pengembang Senior. **Modul ini sama sekali tidak boleh diakses oleh pengguna biasa, manajer, atau bahkan staf IT junior.**

- **Selalu Lakukan Backup**: Sebelum menjalankan perintah modifikasi data (`UPDATE`, `DELETE`, `INSERT`), sangat disarankan untuk melakukan *backup database* terlebih dahulu.

- **Uji Coba Terlebih Dahulu**: Setiap *query* modifikasi harus selalu diuji coba di lingkungan *staging* atau *testing* (non-

produksi) sebelum dijalankan di *database* produksi.

Revision #2
Created 19 October 2025 10:24:36 by Sayu
Updated 23 October 2025 08:23:50 by Sayu