**Memproses Data dari Kotor ke Bersih**

**Memverifikasi dan melaporkan hasil**

Verifikasi adalah proses untuk mengkonfirmasi bahwa upaya pembersihan data telah dilakukan dengan baik sehingga data yang dihasilkan akurat dan dapat diandalkan. Langkah ini terdiri dari pengecekan ulang terhadap dataset yang sudah bersih, pembersihan manual jika diperlukan, dan meluangkan waktu untuk benar-benar memahami tujuan awal proyek.

Bagian besar lainnya dari proses verifikasi adalah melaporkan upaya Anda. Komunikasi yang terbuka ibarat nyawa untuk setiap proyek analitis data. Pelaporan adalah cara yang sangat efektif untuk menunjukkan kepada seluruh tim bahwa Anda bersikap 100 persen transparan dalam pembersihan data. Pelaporan juga merupakan kesempatan besar untuk menunjukkan kepada pemangku kepentingan bahwa Anda bertanggung jawab, membangun kepercayaan dengan anggota tim, dan memastikan bahwa semua orang memiliki pemahaman yang sama mengenai detail proyek. Terdapat berbagai strategi yang berbeda dalam pelaporan, seperti membuat laporan pembersihan data, mendokumentasikan proses pembersihan, dan menggunakan changelog. Changelog adalah file yang berisi daftar kronologis tentang modifikasi atau perubahan yang dilakukan terhadap proyek. Changelog biasanya terorganisir berdasarkan versi dan tanggal, beserta daftar fitur yang ditambahkan, diperbaiki, atau dihapus. Changelog sangat berguna untuk melacak bagaimana dataset berevolusi selama proyek berlangsung. Membuat Changelog juga merupakan cara yang bagus untuk mengomunikasikan dan melaporkan data kepada orang lain.

Bagian penting lain dari verifikasi adalah melihat gambaran besar dari proyek Anda. Ini adalah kesempatan untuk mengonfirmasi bahwa Anda benar-benar fokus pada masalah bisnis yang perlu Anda selesaikan dan tujuan proyek secara keseluruhan serta untuk memastikan bahwa data Anda benar-benar mampu memecahkan masalah dan mencapai tujuan. Melihat gambaran besar proyek, terdiri dari tiga hal.

* Pertama, pertimbangkan masalah bisnis yang sedang Anda coba selesaikan dengan data.
* Kedua, Anda perlu mempertimbangkan tujuan proyek.
* Ketiga, Anda perlu mempertimbangkan apakah data Anda mampu memecahkan masalah dan memenuhi tujuan proyek.

**Memperbaiki masalah yang paling umum**

Pastikan bahwa Anda mengidentifikasi masalah yang paling umum dan memperbaikinya, termasuk:

* Sumber kesalahan: Apakah Anda sudah menggunakan alat dan fungsi yang tepat untuk menemukan sumber kesalahan dalam dataset Anda?
* Data null: Apakah Anda sudah mencari NULL menggunakan pemformatan dan filter bersyarat?
* Kata-kata yang salah eja: Apakah Anda sudah menemukan semua kesalahan eja?
* Nomor yang salah ketik: Apakah Anda sudah memastikan bahwa data numerik telah dimasukkan dengan benar?
* Spasi dan karakter tambahan: Apakah Anda sudah menghapus spasi atau karakter tambahan menggunakan fungsi TRIM?
* Duplikat: Apakah Anda sudah menghapus duplikat dalam spreadsheet menggunakan fungsi Remove Duplicates atau DISTINCT dalam SQL?
* Jenis data yang tidak cocok: Apakah Anda sudah memeriksa bahwa data numerik, tanggal, dan string diketik dengan benar?
* String yang berantakan (tidak konsisten): Apakah Anda sudah memastikan bahwa semua string konsisten dan bermakna?
* Format tanggal yang berantakan (tidak konsisten): Apakah Anda sudah memformat tanggal secara konsisten di seluruh kumpulan dataset?
* Label variabel yang menyesatkan (kolom): Apakah Anda sudah memberi nama kolom dengan cara yang bermakna?
* Data terpotong: Apakah Anda sudah memeriksa data yang terpotong atau hilang yang perlu diperbaiki?
* Logika bisnis: Apakah Anda sudah memeriksa bahwa data masuk akal, berkaitan dengan pengetahuan Anda tentang bisnis?

**Meninjau tujuan proyek**

Setelah Anda menyelesaikan tugas pembersihan data ini, sebaiknya tinjau tujuan proyek Anda dan konfirmasikan bahwa data Anda masih selaras dengan tujuan tersebut. Ini adalah proses berkelanjutan yang akan Anda lakukan di sepanjang proyek-- tetapi berikut adalah tiga langkah yang dapat Anda ingat saat memikirkan hal ini:

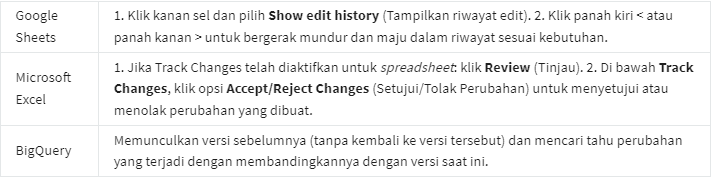
* Konfirmasikan masalah bisnis
* Konfirmasikan tujuan proyek
* Verifikasi bahwa data dapat memecahkan masalah dan selaras dengan tujuan

Apa kesamaan antara data engineer, penulis data, dan analis data? Change atau perubahan.

Data engineer menggunakan perintah perubahan rekayasa (ECO/engineering change orders) untuk melacak detail dari desain produk baru dan perubahan yang diusulkan pada produk yang sudah ada. Penulis data menggunakan riwayat revisi dokumen untuk melacak perubahan aliran dan pengeditan dokumen. Dan analis data menggunakan changelog untuk melacak transformasi dan pembersihan data.

**Kontrol versi otomatis dapat sangat membantu Anda**

Sebagian besar aplikasi peranti lunak memiliki semacam pelacakan riwayat bawaan. Misalnya, di Google Sheets, Anda dapat memeriksa riwayat versi seluruh sheet atau masing-masing sel dan kembali ke versi sebelumnya. Di Microsoft Excel, Anda dapat menggunakan fitur yang disebut Track Changes (Lacak Perubahan). Dan di BigQuery, Anda dapat melihat riwayat untuk memeriksa hal-hal yang telah berubah.



**Changelog membantu Anda menyelesaikan proyek**

Changelog adalah dokumen yang digunakan untuk merekam perubahan penting yang dibuat untuk sebuah proyek selama masa implementasinya di semua tugas. Hal ini biasanya dikuratori sehingga perubahan yang dicatat akan terdaftar secara kronologis di semua versi proyek.

Manfaat utama penggunaan changelog adalah bahwa kontributor dan pengguna yang terhubung dengan proyek mendapatkan daftar spesifik tentang perubahan penting apa yang telah dibuat, kapan mereka dibuat, dan kadang-kadang, versi apa yang mereka rilis. Ini adalah alat yang tak ternilai untuk mengomunikasikan bagaimana proyek telah berkembang dari waktu ke waktu kepada rekan kerja, manajemen, dan pemangku kepentingan.

Sebuah changelog dapat dibuat pada riwayat versi otomatis Anda dengan memberi catatan yang lebih detail tentang pekerjaan Anda. Di sinilah analis data mencatat semua perubahan yang mereka buat terhadap data. Berikut adalah cara lain untuk memahaminya. Riwayat versi mencatat apa yang telah dilakukan dalam perubahan data untuk suatu proyek, tetapi tidak memberi tahu alasannya. Changelog sangat berguna untuk membantu memahami alasan perubahan. Changelog tidak memiliki format yang ditetapkan dan Anda bahkan dapat membuat entri Anda dalam dokumen kosong. Tetapi jika Anda menggunakan changelog bersama, sebaiknya Anda menyepakati format entri log yang akan digunakan dengan para analis data lainnya.

Biasanya, changelog mencatat jenis informasi berikut:

* Data, file, rumus, kueri, atau komponen lain yang berubah
* Deskripsi mengenai perubahan yang dilakukan
* Tanggal perubahan
* Orang yang melakukan perubahan
* Orang yang menyetujui perubahan
* Nomor versi
* Alasan perubahan

Berikut adalah bagaimana sistem pengontrol versi memengaruhi perubahan pada kueri:

* Perusahaan memiliki versi resmi dari kueri dalam sistem pengontrol versi mereka.
* Analis memastikan bahwa versi kueri terbaru adalah versi yang akan mereka ubah. Langkah ini disebut sinkronisasi.
* Analis membuat perubahan pada kueri.
* Analis akan meminta seseorang untuk meninjau perubahan ini. Langkah ini disebut pemeriksaan kode atau code review dan dapat dilakukan secara informal atau pun formal. Tinjauan informal bisa dilakukan dengan meminta analis senior untuk melihat perubahan terkait.
* Setelah peninjau menyetujui perubahan, analis mengirimkan versi kueri yang diperbarui ke repositori di sistem pengontrol versi (version control system) perusahaan. Langkah ini disebut komitmen kode atau code commit. Praktik terbaiknya adalah mendokumentasikan perubahan yang dilakukan dan menjelaskan alasannya di kolom komentar. Kembali ke contoh kueri untuk menarik data pendapatan harian, alasannya mungkin: Memperbarui data pendapatan untuk memasukkan pendapatan yang berasal dari produk baru, Calypso.
* Setelah perubahan dikirimkan, semua orang di perusahaan akan dapat mengakses dan menggunakan kueri baru ini saat mereka menyinkronkan ke kueri terbaru yang disimpan dalam sistem pengontrol versi (version control system).
* Jika terdapat masalah dalam kueri atau terdapat perubahan dalam kebutuhan bisnis, analis dapat membatalkan perubahan pada kueri dengan menggunakan sistem pengontrol versi. Analis dapat melihat daftar kronologis dari semua perubahan yang dibuat pada kueri dan siapa yang membuat setiap perubahan. Kemudian, setelah menemukan perubahan yang mereka lakukan, analis dapat mengembalikannya ke versi sebelumnya.
* Kueri akan kembali seperti semula sebelum analis melakukan perubahan. Dan semua orang di perusahaan juga akan melihat kueri awal yang sudah dikembalikan ini.

Prinsip-prinsip berikut membantu membuat changelog yang dapat diakses oleh orang lain:

* Changelog adalah untuk manusia, bukan mesin, jadi tulislah dengan jelas.
* Setiap versi harus memiliki entri tersendiri.
* Setiap perubahan harus memiliki garisnya sendiri.
* Kelompokkan jenis perubahan yang sama. Misalnya, perubahan Perbaikan harus dikelompokkan secara terpisah dari perubahan Penambahan.
* Versi harus disusun secara kronologis dimulai dari yang terbaru.
* Tanggal rilis setiap versi harus dicatat.

Semua perubahan untuk setiap kategori harus dikelompokkan bersama. Jenis perubahan biasanya termasuk dalam salah satu kategori berikut:

* Added/Ditambahkan: fitur baru yang diperkenalkan
* Changed/Diubah: perubahan fungsi yang ada
* Deprecated/Usang: fitur yang akan dihapus
* Removed/Dihapus: fitur yang telah dihapus
* Fixed/Diperbaiki: perbaikan bug
* Security/Keamanan: mengurangi kerentanan

