Nama : Siti Raudah

NIM : 40011423650330

Kelas : H

Mata Kuliah : Sistem Informasi Akuntansi

Pertemuan 2

buat data base perusahaan dengan tabel produksi, persediaan,  penjualan dan buat struktur dari masing masing tabel tersebut

ChatGPT bilang:

ChatGPT

Berikut adalah struktur untuk database perusahaan dengan tiga tabel: Produksi, Persediaan, dan Penjualan. Setiap tabel dirancang dengan atribut yang relevan untuk fungsi dan kebutuhan umum dalam bisnis produksi, persediaan, dan penjualan.

Tabel 1: Produksi

Tabel ini menyimpan informasi tentang proses produksi yang dilakukan perusahaan.

Nama Kolom Tipe Data Keterangan

id\_produksi INT (Primary Key) ID unik untuk setiap proses produksi

id\_produk INT ID produk yang diproduksi (referensi dari tabel Produk)

tgl\_produksi DATE Tanggal proses produksi dilakukan

jumlah\_produksi INT Jumlah produk yang dihasilkan

biaya\_produksi DECIMAL(10, 2) Biaya total untuk produksi

karyawan VARCHAR(50) Nama karyawan atau tim yang bertanggung jawab

status ENUM('selesai', 'proses') Status produksi (selesai atau masih dalam proses)

Tabel 2: Persediaan

Tabel ini menyimpan informasi tentang stok barang yang tersedia.

Nama Kolom Tipe Data Keterangan

id\_persediaan INT (Primary Key) ID unik untuk setiap persediaan

id\_produk INT ID produk (referensi dari tabel Produk)

jumlah\_stok INT Jumlah produk yang tersedia di gudang

tgl\_update DATE Tanggal terakhir pembaruan persediaan

lokasi\_gudang VARCHAR(50) Lokasi penyimpanan persediaan

status ENUM('tersedia', 'kosong') Status persediaan barang

Tabel 3: Penjualan

Tabel ini menyimpan informasi tentang transaksi penjualan produk.

Nama Kolom Tipe Data Keterangan

id\_penjualan INT (Primary Key) ID unik untuk setiap transaksi penjualan

id\_produk INT ID produk yang dijual (referensi dari tabel Produk)

tgl\_penjualan DATE Tanggal penjualan terjadi

jumlah\_terjual INT Jumlah produk yang dijual

harga\_jual DECIMAL(10, 2) Harga jual per unit

total\_penjualan DECIMAL(10, 2) Total harga untuk jumlah yang dijual

id\_pelanggan INT ID pelanggan (referensi dari tabel Pelanggan)

status ENUM('terkirim', 'diproses') Status pengiriman barang

Contoh Struktur Tambahan

Jika diperlukan, tambahkan tabel Produk dan Pelanggan untuk menyimpan informasi produk dan data pelanggan.

Tabel Produk

Nama Kolom Tipe Data Keterangan

id\_produk INT (Primary Key) ID unik untuk setiap produk

nama\_produk VARCHAR(50) Nama produk

kategori VARCHAR(30) Kategori produk

harga DECIMAL(10, 2) Harga standar produk

Tabel Pelanggan

Nama Kolom Tipe Data Keterangan

id\_pelanggan INT (Primary Key) ID unik untuk setiap pelanggan

nama\_pelanggan VARCHAR(50) Nama pelanggan

kontak VARCHAR(50) Informasi kontak pelanggan

alamat TEXT Alamat pelanggan

Struktur ini cukup fleksibel untuk berbagai keperluan bisnis perusahaan manufaktur atau distribusi. Jika ada keperluan spesifik, struktur tabel dapat dimodifikasi sesuai kebutuhan.

Anda bilang:

primary key dari kolom produksi, persediaan, dan penjualan yang dapat berinteraksi satu sama lain

ChatGPT bilang:

ChatGPT

Agar tabel Produksi, Persediaan, dan Penjualan dapat berinteraksi satu sama lain, kita bisa menggunakan kunci relasi yang menghubungkan setiap tabel. Pada skenario ini, kunci primer yang relevan untuk menghubungkan ketiga tabel tersebut adalah id\_produk, yang akan bertindak sebagai foreign key dalam setiap tabel. Berikut adalah struktur yang diperbarui dengan relasi antar-tabel.

1. Produksi

Tabel Produksi menyimpan informasi tentang proses produksi, dengan id\_produk sebagai foreign key yang mengacu ke tabel Produk.

Nama Kolom Tipe Data Keterangan

id\_produksi INT (Primary Key) ID unik untuk setiap proses produksi

id\_produk INT (Foreign Key dari tabel Produk) ID produk yang diproduksi

tgl\_produksi DATE Tanggal proses produksi

jumlah\_produksi INT Jumlah produk yang dihasilkan

biaya\_produksi DECIMAL(10, 2) Biaya total untuk produksi

karyawan VARCHAR(50) Nama karyawan atau tim yang bertanggung jawab

status ENUM('selesai', 'proses') Status produksi

Catatan: id\_produk sebagai foreign key menghubungkan tabel Produksi dengan tabel Produk, sehingga kita tahu produk mana yang diproduksi.

2. Persediaan

Tabel Persediaan menyimpan stok barang yang tersedia, dengan id\_produk sebagai foreign key untuk merujuk ke produk tertentu.

Nama Kolom Tipe Data Keterangan

id\_persediaan INT (Primary Key) ID unik untuk setiap entri persediaan

id\_produk INT (Foreign Key dari tabel Produk) ID produk yang tersedia

jumlah\_stok INT Jumlah produk yang tersedia di gudang

tgl\_update DATE Tanggal terakhir pembaruan persediaan

lokasi\_gudang VARCHAR(50) Lokasi penyimpanan persediaan

status ENUM('tersedia', 'kosong') Status ketersediaan barang

Catatan: id\_produk sebagai foreign key menghubungkan tabel Persediaan dengan tabel Produk, sehingga setiap entri stok terkait dengan produk tertentu yang diproduksi dan dijual.

3. Penjualan

Tabel Penjualan menyimpan informasi transaksi penjualan dengan id\_produk sebagai foreign key, untuk melacak produk yang dijual.

Nama Kolom Tipe Data Keterangan

id\_penjualan INT (Primary Key) ID unik untuk setiap transaksi penjualan

id\_produk INT (Foreign Key dari tabel Produk) ID produk yang dijual

tgl\_penjualan DATE Tanggal penjualan terjadi

jumlah\_terjual INT Jumlah produk yang dijual

harga\_jual DECIMAL(10, 2) Harga jual per unit

total\_penjualan DECIMAL(10, 2) Total harga untuk jumlah yang dijual

id\_pelanggan INT (Foreign Key dari tabel Pelanggan) ID pelanggan

status ENUM('terkirim', 'diproses') Status pengiriman barang

Catatan: id\_produk sebagai foreign key di sini memungkinkan kita untuk melacak produk yang dijual di antara transaksi penjualan, produksi, dan persediaan.

Struktur Tambahan: Tabel Produk dan Pelanggan

Tabel ini dibutuhkan untuk menyimpan informasi produk dan data pelanggan, sehingga setiap tabel memiliki foreign key yang dapat saling terhubung.

Tabel Produk

Nama Kolom Tipe Data Keterangan

id\_produk INT (Primary Key) ID unik untuk setiap produk

nama\_produk VARCHAR(50) Nama produk

kategori VARCHAR(30) Kategori produk

harga DECIMAL(10, 2) Harga standar produk

Tabel Pelanggan

Nama Kolom Tipe Data Keterangan

id\_pelanggan INT (Primary Key) ID unik untuk setiap pelanggan

nama\_pelanggan VARCHAR(50) Nama pelanggan

kontak VARCHAR(50) Informasi kontak pelanggan

alamat TEXT Alamat pelanggan

Dengan menggunakan id\_produk sebagai foreign key di ketiga tabel (Produksi, Persediaan, Penjualan), hubungan antar-tabel dapat terjaga dan data dapat dianalisis secara terintegrasi berdasarkan produk.

Anda bilang:

 buat data base dari tabel daata diatas ,dan cleaning data dari tabel tersebutagar bisa dianalisis, dan buat analisisnya

ChatGPT bilang:

ChatGPT

Untuk membangun database, membersihkan data, dan melakukan analisis, berikut adalah langkah-langkah detail mulai dari pembuatan tabel di database, pembersihan data, hingga analisis dasar.

1. Pembuatan Database dan Tabel di SQL

Untuk membuat database beserta tabelnya, berikut adalah kode SQL yang dapat digunakan. Pastikan Anda memiliki akses ke SQL Server seperti MySQL, PostgreSQL, atau SQLite.

Membuat Database

sql

Salin kode

CREATE DATABASE perusahaan;

USE perusahaan;

Tabel Produk

sql

Salin kode

CREATE TABLE Produk (

    id\_produk INT PRIMARY KEY,

    nama\_produk VARCHAR(50) NOT NULL,

    kategori VARCHAR(30),

    harga DECIMAL(10, 2)

);

Tabel Produksi

sql

Salin kode

CREATE TABLE Produksi (

    id\_produksi INT PRIMARY KEY,

    id\_produk INT,

    tgl\_produksi DATE,

    jumlah\_produksi INT,

    biaya\_produksi DECIMAL(10, 2),

    karyawan VARCHAR(50),

    status ENUM('selesai', 'proses'),

    FOREIGN KEY (id\_produk) REFERENCES Produk(id\_produk)

);

Tabel Persediaan

sql

Salin kode

CREATE TABLE Persediaan (

    id\_persediaan INT PRIMARY KEY,

    id\_produk INT,

    jumlah\_stok INT,

    tgl\_update DATE,

    lokasi\_gudang VARCHAR(50),

    status ENUM('tersedia', 'kosong'),

    FOREIGN KEY (id\_produk) REFERENCES Produk(id\_produk)

);

Tabel Penjualan

sql

Salin kode

CREATE TABLE Penjualan (

    id\_penjualan INT PRIMARY KEY,

    id\_produk INT,

    tgl\_penjualan DATE,

    jumlah\_terjual INT,

    harga\_jual DECIMAL(10, 2),

    total\_penjualan DECIMAL(10, 2),

    id\_pelanggan INT,

    status ENUM('terkirim', 'diproses'),

    FOREIGN KEY (id\_produk) REFERENCES Produk(id\_produk)

);

Tabel Pelanggan

sql

Salin kode

CREATE TABLE Pelanggan (

    id\_pelanggan INT PRIMARY KEY,

    nama\_pelanggan VARCHAR(50),

    kontak VARCHAR(50),

    alamat TEXT

);

2. Membersihkan Data

Data cleaning diperlukan untuk memastikan bahwa semua data dapat digunakan untuk analisis dengan akurat. Berikut adalah langkah-langkah data cleaning yang dapat diterapkan:

Langkah Data Cleaning

Menghapus Duplikasi:

Pastikan tidak ada data duplikat di setiap tabel, terutama pada kolom primary key. Gunakan query berikut untuk memeriksa duplikasi:

sql

Salin kode

SELECT id\_produk, COUNT(\*)

FROM Produk

GROUP BY id\_produk

HAVING COUNT(\*) > 1;

Menangani Nilai Null:

Pastikan semua kolom yang wajib diisi tidak memiliki nilai NULL, terutama pada kolom id\_produk, nama\_produk, harga, dan kolom penting lainnya.

sql

Salin kode

SELECT \* FROM Produk WHERE nama\_produk IS NULL;

Normalisasi Data:

Periksa data di kolom seperti kategori pada tabel Produk untuk memastikan konsistensi penamaan, misalnya "Elektronik" vs "elektronik".

Validasi Tanggal:

Pastikan tgl\_produksi pada tabel Produksi tidak lebih dari tgl\_update pada tabel Persediaan dan tgl\_penjualan pada tabel Penjualan tidak mendahului tgl\_produksi.

Perhitungan Total Penjualan:

Pastikan kolom total\_penjualan pada tabel Penjualan adalah hasil dari jumlah\_terjual \* harga\_jual agar konsisten dan akurat.

3. Analisis Data

Dengan data yang sudah bersih, berikut adalah beberapa analisis dasar yang dapat dilakukan.

A. Analisis Produksi dan Stok

Jumlah Produksi per Produk:

Untuk mengetahui jumlah total produksi dari setiap produk.

sql

Salin kode

SELECT P.nama\_produk, SUM(Pr.jumlah\_produksi) AS total\_produksi

FROM Produksi Pr

JOIN Produk P ON Pr.id\_produk = P.id\_produk

GROUP BY P.nama\_produk;

Stok Produk Tersedia:

Untuk melihat berapa stok yang tersedia untuk setiap produk.

sql

Salin kode

SELECT P.nama\_produk, SUM(S.jumlah\_stok) AS total\_stok

FROM Persediaan S

JOIN Produk P ON S.id\_produk = P.id\_produk

WHERE S.status = 'tersedia'

GROUP BY P.nama\_produk;

B. Analisis Penjualan

Total Penjualan per Produk:

Melihat total penjualan yang dihasilkan untuk setiap produk.

sql

Salin kode

SELECT P.nama\_produk, SUM(Pe.total\_penjualan) AS total\_penjualan

FROM Penjualan Pe

JOIN Produk P ON Pe.id\_produk = P.id\_produk

GROUP BY P.nama\_produk;

Produk Terlaris:

Untuk mengetahui produk mana yang memiliki jumlah penjualan tertinggi.

sql

Salin kode

SELECT P.nama\_produk, SUM(Pe.jumlah\_terjual) AS jumlah\_terjual

FROM Penjualan Pe

JOIN Produk P ON Pe.id\_produk = P.id\_produk

GROUP BY P.nama\_produk

ORDER BY jumlah\_terjual DESC

LIMIT 1;

C. Analisis Biaya Produksi vs Penjualan

Profit per Produk:

Untuk menghitung profit dengan mengurangi biaya produksi dari pendapatan penjualan.

sql

Salin kode

SELECT P.nama\_produk,

       SUM(Pe.total\_penjualan) - SUM(Pr.biaya\_produksi) AS profit

FROM Penjualan Pe

JOIN Produksi Pr ON Pe.id\_produk = Pr.id\_produk

JOIN Produk P ON Pe.id\_produk = P.id\_produk

GROUP BY P.nama\_produk;

Produk dengan Biaya Produksi Tinggi:

Untuk mengetahui produk yang memiliki biaya produksi tertinggi.

sql

Salin kode

SELECT P.nama\_produk, SUM(Pr.biaya\_produksi) AS total\_biaya

FROM Produksi Pr

JOIN Produk P ON Pr.id\_produk = P.id\_produk

GROUP BY P.nama\_produk

ORDER BY total\_biaya DESC

LIMIT 1;

Ringkasan Analisis

Laporan Produksi dan Stok: Memantau berapa banyak produk yang diproduksi dan tersedia di gudang untuk mengoptimalkan proses produksi.

Laporan Penjualan: Mengetahui produk-produk terlaris untuk fokus pada produksi dan pemasaran.

Laporan Keuntungan: Menghitung keuntungan per produk berdasarkan total penjualan dikurangi total biaya produksi.

Analisis ini dapat membantu dalam pengambilan keputusan bisnis untuk meningkatkan efisiensi produksi, mengurangi biaya, dan memaksimalkan penjualan.

Anda bilang:

Cleaning data base perusahaan dengan tabel produksi, persediaan, dan penjuaalan

ChatGPT bilang:

ChatGPT

Cleaning data pada database perusahaan dengan tabel Produksi, Persediaan, dan Penjualan bertujuan memastikan data akurat, konsisten, dan siap untuk analisis lebih lanjut. Berikut ini adalah langkah-langkah yang dapat diambil untuk membersihkan setiap tabel:

1. Membersihkan Tabel Produksi

Data di tabel Produksi harus bersih dan sesuai untuk memastikan akurasi informasi tentang proses produksi.

Langkah-langkah:

Menghapus Duplikasi pada id\_produksi:

Periksa apakah ada duplikasi pada kolom id\_produksi, yang seharusnya unik untuk setiap proses produksi.

sql

Salin kode

DELETE FROM Produksi

WHERE id\_produksi IN (

    SELECT id\_produksi

    FROM Produksi

    GROUP BY id\_produksi

    HAVING COUNT(\*) > 1

);

Memastikan Nilai id\_produk Valid:

Pastikan setiap id\_produk di tabel Produksi memiliki referensi yang valid di tabel Produk.

sql

Salin kode

DELETE FROM Produksi

WHERE id\_produk NOT IN (SELECT id\_produk FROM Produk);

Memperbaiki atau Mengisi Nilai Null pada Kolom tgl\_produksi, jumlah\_produksi, dan biaya\_produksi:

Ganti nilai NULL dengan data yang valid, atau hapus data yang tidak dapat diperbaiki.

sql

Salin kode

UPDATE Produksi

SET tgl\_produksi = IFNULL(tgl\_produksi, CURDATE()),

    jumlah\_produksi = IFNULL(jumlah\_produksi, 0),

    biaya\_produksi = IFNULL(biaya\_produksi, 0);

Validasi Kolom status:

Pastikan bahwa kolom status hanya berisi nilai yang valid: 'selesai' atau 'proses'.

sql

Salin kode

UPDATE Produksi

SET status = 'proses'

WHERE status NOT IN ('selesai', 'proses');

2. Membersihkan Tabel Persediaan

Tabel Persediaan menyimpan informasi tentang stok, dan data di dalamnya perlu disesuaikan dengan jumlah produk yang tersedia secara akurat.

Langkah-langkah:

Menghapus Duplikasi pada id\_persediaan:

Hapus baris yang memiliki id\_persediaan duplikat.

sql

Salin kode

DELETE FROM Persediaan

WHERE id\_persediaan IN (

    SELECT id\_persediaan

    FROM Persediaan

    GROUP BY id\_persediaan

    HAVING COUNT(\*) > 1

);

Validasi id\_produk:

Pastikan setiap id\_produk pada tabel Persediaan ada di tabel Produk.

sql

Salin kode

DELETE FROM Persediaan

WHERE id\_produk NOT IN (SELECT id\_produk FROM Produk);

Periksa dan Perbaiki Nilai NULL di Kolom jumlah\_stok dan tgl\_update:

Pastikan jumlah\_stok tidak NULL dan perbarui nilai NULL di tgl\_update dengan tanggal saat ini.

sql

Salin kode

UPDATE Persediaan

SET jumlah\_stok = IFNULL(jumlah\_stok, 0),

    tgl\_update = IFNULL(tgl\_update, CURDATE());

Validasi Kolom status:

Pastikan kolom status hanya berisi nilai 'tersedia' atau 'kosong'.

sql

Salin kode

UPDATE Persediaan

SET status = 'kosong'

WHERE status NOT IN ('tersedia', 'kosong');

3. Membersihkan Tabel Penjualan

Tabel Penjualan harus memberikan informasi transaksi yang bersih dan valid, dengan data akurat tentang setiap penjualan produk.

Langkah-langkah:

Menghapus Duplikasi pada id\_penjualan:

Hapus data yang memiliki id\_penjualan duplikat.

sql

Salin kode

DELETE FROM Penjualan

WHERE id\_penjualan IN (

    SELECT id\_penjualan

    FROM Penjualan

    GROUP BY id\_penjualan

    HAVING COUNT(\*) > 1

);

Validasi id\_produk:

Pastikan setiap id\_produk yang tercatat pada tabel Penjualan memiliki referensi yang valid di tabel Produk.

sql

Salin kode

DELETE FROM Penjualan

WHERE id\_produk NOT IN (SELECT id\_produk FROM Produk);

Memastikan Kolom tgl\_penjualan, jumlah\_terjual, harga\_jual, dan total\_penjualan Tidak Null:

Perbarui atau isi nilai yang kosong pada kolom penting ini.

sql

Salin kode

UPDATE Penjualan

SET tgl\_penjualan = IFNULL(tgl\_penjualan, CURDATE()),

    jumlah\_terjual = IFNULL(jumlah\_terjual, 0),

    harga\_jual = IFNULL(harga\_jual, 0),

    total\_penjualan = IFNULL(total\_penjualan, jumlah\_terjual \* harga\_jual);

Validasi Kolom status:

Pastikan kolom status hanya berisi nilai yang valid: 'terkirim' atau 'diproses'.

sql

Salin kode

UPDATE Penjualan

SET status = 'diproses'

WHERE status NOT IN ('terkirim', 'diproses');

Hitung Ulang Kolom total\_penjualan:

Pastikan bahwa kolom total\_penjualan adalah hasil dari jumlah\_terjual \* harga\_jual.

sql

Salin kode

UPDATE Penjualan

SET total\_penjualan = jumlah\_terjual \* harga\_jual;

Ringkasan Data Cleaning

Setelah langkah-langkah ini diterapkan, data pada setiap tabel akan:

Bebas dari duplikasi

Memiliki nilai yang konsisten pada kolom referensi (id\_produk)

Tidak mengandung nilai NULL pada kolom penting (tanggal, jumlah, harga)

Kolom status berisi nilai yang valid

Memiliki nilai total yang dihitung dengan benar

Data yang bersih ini siap untuk dianalisis lebih lanjut, dan langkah-langkah ini membantu mencegah masalah analisis akibat inkonsistensi atau data yang hilang.

Anda bilang:

buat data cleaning diatas menjadi tabel

ChatGPT bilang:

ChatGPT

Untuk membuat proses data cleaning lebih terstruktur, berikut adalah tabel langkah-langkah data cleaning yang dirancang untuk masing-masing tabel (Produksi, Persediaan, Penjualan). Setiap tabel berisi kolom yang menjelaskan tahap cleaning, kolom yang dibersihkan, query SQL yang digunakan, dan penjelasan.

Tabel 1: Data Cleaning untuk Tabel Produksi

Tahap Kolom yang Dibersihkan Query SQL Penjelasan

Hapus Duplikat id\_produksi DELETE FROM Produksi WHERE id\_produksi IN (SELECT id\_produksi FROM Produksi GROUP BY id\_produksi HAVING COUNT(\*) > 1); Menghapus baris dengan ID duplikat di kolom id\_produksi.

Validasi Referensi id\_produk DELETE FROM Produksi WHERE id\_produk NOT IN (SELECT id\_produk FROM Produk); Hapus entri id\_produk yang tidak memiliki referensi di tabel Produk.

Isi Nilai Null tgl\_produksi, jumlah\_produksi, biaya\_produksi UPDATE Produksi SET tgl\_produksi = IFNULL(tgl\_produksi, CURDATE()), jumlah\_produksi = IFNULL(jumlah\_produksi, 0), biaya\_produksi = IFNULL(biaya\_produksi, 0); Mengisi nilai NULL di kolom tanggal, jumlah, dan biaya dengan nilai default.

Validasi Status status UPDATE Produksi SET status = 'proses' WHERE status NOT IN ('selesai', 'proses'); Pastikan status hanya berisi 'selesai' atau 'proses'.

Tabel 2: Data Cleaning untuk Tabel Persediaan

Tahap Kolom yang Dibersihkan Query SQL Penjelasan

Hapus Duplikat id\_persediaan DELETE FROM Persediaan WHERE id\_persediaan IN (SELECT id\_persediaan FROM Persediaan GROUP BY id\_persediaan HAVING COUNT(\*) > 1); Menghapus baris dengan ID duplikat di kolom id\_persediaan.

Validasi Referensi id\_produk DELETE FROM Persediaan WHERE id\_produk NOT IN (SELECT id\_produk FROM Produk); Hapus entri id\_produk yang tidak memiliki referensi di tabel Produk.

Isi Nilai Null jumlah\_stok, tgl\_update UPDATE Persediaan SET jumlah\_stok = IFNULL(jumlah\_stok, 0), tgl\_update = IFNULL(tgl\_update, CURDATE()); Mengisi nilai NULL pada kolom jumlah stok dan tanggal update.

Validasi Status status UPDATE Persediaan SET status = 'kosong' WHERE status NOT IN ('tersedia', 'kosong'); Pastikan status hanya berisi 'tersedia' atau 'kosong'.

Tabel 3: Data Cleaning untuk Tabel Penjualan

Tahap Kolom yang Dibersihkan Query SQL Penjelasan

Hapus Duplikat id\_penjualan DELETE FROM Penjualan WHERE id\_penjualan IN (SELECT id\_penjualan FROM Penjualan GROUP BY id\_penjualan HAVING COUNT(\*) > 1); Menghapus baris dengan ID duplikat di kolom id\_penjualan.

Validasi Referensi id\_produk DELETE FROM Penjualan WHERE id\_produk NOT IN (SELECT id\_produk FROM Produk); Hapus entri id\_produk yang tidak memiliki referensi di tabel Produk.

Isi Nilai Null tgl\_penjualan, jumlah\_terjual, harga\_jual, total\_penjualan UPDATE Penjualan SET tgl\_penjualan = IFNULL(tgl\_penjualan, CURDATE()), jumlah\_terjual = IFNULL(jumlah\_terjual, 0), harga\_jual = IFNULL(harga\_jual, 0), total\_penjualan = IFNULL(total\_penjualan, jumlah\_terjual \* harga\_jual); Mengisi nilai NULL di kolom tanggal, jumlah, harga, dan menghitung ulang total penjualan.

Validasi Status status UPDATE Penjualan SET status = 'diproses' WHERE status NOT IN ('terkirim', 'diproses'); Pastikan status hanya berisi 'terkirim' atau 'diproses'.

Hitung Ulang Total total\_penjualan UPDATE Penjualan SET total\_penjualan = jumlah\_terjual \* harga\_jual; Pastikan total\_penjualan adalah hasil dari jumlah\_terjual \* harga\_jual.

Penjelasan Tabel Data Cleaning

Tahap: Menjelaskan langkah-langkah cleaning yang dilakukan.

Kolom yang Dibersihkan: Kolom yang menjadi target dari setiap langkah pembersihan.

Query SQL: Kode SQL untuk melakukan data cleaning pada kolom yang bersangkutan.

Penjelasan: Deskripsi singkat mengenai tujuan dari setiap langkah pembersihan.

Tabel ini memberikan panduan terstruktur untuk membersihkan data pada database perusahaan, sehingga semua data siap untuk analisis lebih lanjut dengan standar yang lebih konsisten dan akurat.

Anda bilang:

buat excel dari data base perusahaan dengan kolom produksi,persediaan, dan penjualan

ChatGPT bilang:

ChatGPT

Berikut adalah file Excel dari database perusahaan yang terdiri dari tabel Produksi, Persediaan, dan Penjualan:

Download database\_perusahaan.xlsx ​​

Anda bilang:

dari data base perusahaan dia atas yang berisi kolom produksi, persediaan, dan penjualan. buatlah data base tersebut menjadi excel

ChatGPT bilang:

ChatGPT

File Excel dari database perusahaan yang berisi tabel Produksi, Persediaan, dan Penjualan telah dibuat dan tersedia di sini:

Download database\_perusahaan.xlsx

Anda bilang:

dari data base perushaan yang berisi tabel produksi, persediaan, dan penjualan. buatlah data base tersebut menjaadi excel dengan masing masing bari 20 baris pada setiap kriteria tabel

ChatGPT bilang:

ChatGPT

File Excel dari database perusahaan yang berisi tabel Produksi, Persediaan, dan Penjualan dengan masing-masing 20 baris data telah dibuat. Anda dapat mengunduhnya di sini:

Download database\_perusahaan\_20\_rows.xlsx ​​

Anda bilang:

DARI FILE EXCEL DARI DATA BASE PERUSAHAAN DI ATAS BUAT DATA ANALITIK, PENJUALAN TERTINGGI UNTUK TANGGAL PRODUKSI DENGAN RENTAN PRODUKSI 10 OKTOBER 2023-29 OKTOBER 2023  DENGAN ID PRODUKSI YANG GENAP , CARIKAN PENJUALAN YANG TERTINGGI DARI TANGGAL  10 OKTOBER 22023-29 OKTOBER 2023 YANG ID PRODUKSI NYA GENAP, BUAT SOALNYA DARI PERNYATAAN TERSEBUT

ChatGPT bilang:

ChatGPT

Dari data yang difilter, penjualan tertinggi pada rentang tanggal 10 Oktober 2023 hingga 29 Oktober 2023, dengan ID produksi genap, adalah sebagai berikut:

ID Penjualan: 2

ID Produk: 101

Tanggal Penjualan: 11 Oktober 2023

Jumlah Terjual: 197

Harga Jual: 131

Total Penjualan: 25,807

ID Pelanggan: 219

Status: Diproses

Soal Berdasarkan Data Analitik Ini

Berapakah jumlah terjual tertinggi pada rentang tanggal 10 Oktober 2023 hingga 29 Oktober 2023 untuk produk dengan ID produksi genap?

Pada tanggal berapakah penjualan tertinggi terjadi dalam rentang tanggal tersebut, dan apa ID produk yang terjual?

Siapakah pelanggan yang melakukan pembelian tertinggi pada rentang tersebut? ​