

BASIS DATA KELOMPOK 4

NAYLA PUTRI CAHYA RAMADANI - 2024071020

KEREN SANDRA SUDARTA – 2024071031

Analisis Normalisasi Database toko_kue (Sweet Delight)

1NF (First Normal Form):

- Semua tabel sudah memiliki primary key.
- Semua nilai atribut sudah atomic (tidak ada yang multivalued atau composite).
- Tidak ada pengulangan grup data.

2NF (Second Normal Form):

- Semua tabel sudah memenuhi 1NF.
- Semua atribut non-kunci bergantung penuh pada primary key.
- Contoh: Pada **detail_pesanan**, **harga_satuan** dan **subtotal** bergantung penuh pada **id_detail**.

3NF (Third Normal Form):

- Semua tabel sudah memenuhi 2NF.
- Tidak ada ketergantungan transitif.
- Contoh:
- Pada **pesanan**, **total_harga** bergantung pada **id_pesanan** (bukan pada atribut lain yang bukan key)

BCNF (Boyce-Codd Normal Form):

- Semua tabel sudah memenuhi 3NF.
- Semua determinan adalah candidate key.
- Tidak ada pelanggaran BCNF yang terdeteksi.

6 Query Laporan Penjualan

NO	NAMA LAPORAN	QUERY SQL	PENJELASAN
1.	Total Penjualan per Kategori	<pre>-- Query 1: Total Penjualan per Kategori Kue SELECT pk.kategori, COUNT(dp.id_detail) AS total_item_terjual, SUM(dp.subtotal) AS total_pendapatan FROM detail_pesanan dp JOIN produk_kue pk ON dp.id_produk = pk.id_produk JOIN pesanan p ON dp.id_pesanan = p.id_pesanan WHERE p.status_pesanan != 'Dibatalkan' GROUP BY pk.kategori ORDER BY total_pendapatan DESC;</pre>	Menampilkan pendapatan per kategori kue
2.	5 Produk Terlaris	<pre>-- Query 2: 5 Produk Terlaris SELECT pk.nama_kue, pk.kategori, SUM(dp.jumlah) AS total_terjual, SUM(dp.subtotal) AS total_pendapatan_produk FROM detail_pesanan dp JOIN produk_kue pk ON dp.id_produk = pk.id_produk JOIN pesanan p ON dp.id_pesanan = p.id_pesanan WHERE p.status_pesanan != 'Dibatalkan' GROUP BY pk.nama_kue, pk.kategori ORDER BY total_terjual DESC LIMIT 5;</pre>	Menampilkan produk paling laris
3.	Pendapatan Harian	<pre>-- Query 3: Pendapatan Harian SELECT DATE(p.tanggal_pesan) AS</pre>	Pendapatan hari ini

		<pre> tanggal, COUNT(p.id_pesanan) AS total_pesanan, SUM(p.total_harga) AS pendapatan_harian FROM pesanan p WHERE DATE(p.tanggal_pesan) = CURDATE() AND p.status_pesanan != 'Dibatalkan' GROUP BY DATE(p.tanggal_pesan); </pre>	
4.	Top Pelanggan	<pre> -- Query 4: Pelanggan dengan Total SELECT pl.nama_pelanggan, pl.nomor_telepon, COUNT(p.id_pesanan) AS total_pesanan, SUM(p.total_harga) AS total_pembelian FROM pesanan p JOIN pelanggan pl ON p.id_pelanggan = pl.id_pelanggan WHERE p.status_pesanan != 'Dibatalkan' GROUP BY pl.id_pelanggan, pl.nama_pelanggan, pl.nomor_telepon ORDER BY total_pembelian DESC LIMIT 10; </pre>	Pelanggan dengan belanja tertinggi
5.	Rata-rata Pesanan/Bulan	<pre> -- Query 5: Rata-rata Nilai Pesanan SELECT YEAR(tanggal_pesan) AS tahun, MONTH(tanggal_pesan) AS bulan, COUNT(*) AS total_pesanan, AVG(total_harga) AS rata_rata_nilai_pesanan, SUM(total_harga) AS total_pendapatan_bulanan FROM pesanan WHERE status_pesanan != 'Dibatalkan' GROUP BY YEAR(tanggal_pesan), MONTH(tanggal_pesan) ORDER BY tahun, bulan; </pre>	Trend nilai pesanan bulanan

6.	Pesanan Pending	<pre>-- Query 6: Pesanan yang Masih Pending atau Diproses SELECT p.id_pesanan, pl.nama_pelanggan, p.tanggal_pesan, p.total_harga, p.status_pesanan, p.tanggal_pengiriman FROM pesanan p JOIN pelanggan pl ON p.id_pelanggan = pl.id_pelanggan WHERE p.status_pesanan IN ('Pending', 'Diproses') ORDER BY p.tanggal_pesan ASC;</pre>	Pesanan yang butuh follow-up
----	-----------------	---	------------------------------