

## **BASIS DATA KELOMPOK 4**

NAYLA PUTRI CAHYA RAMADANI - 2024071020

KEREN SANDRA SUDARTA – 2024071031

---

### **1. STUDI KASUS**

Toko kue "Sweet Delights" adalah usaha kecil menengah yang bergerak di bidang penjualan berbagai macam kue seperti kue ulang tahun, cupcake, brownies, tart, dan kue kering. Toko ini melayani pemesanan kue dengan sistem pengiriman ke alamat pelanggan.

Permasalahan yang dihadapi:

- Pencatatan data pelanggan masih manual
- Kesulitan melacak alamat pengiriman yang berbeda-beda untuk satu pelanggan
- Tidak ada sistem untuk mencatat detail pesanan dan item yang dipesan
- Sulit membuat laporan penjualan dan monitoring stok
- Tidak ada pencatatan nomor telepon pelanggan yang terstruktur

Solusi: Membuat sistem basis data terkomputerisasi yang dapat mengelola data pelanggan, produk, pesanan, dan alamat pengiriman secara terintegrasi.

### **2. TUJUAN**

Tujuan dari pembuatan sistem basis data ini adalah:

1. Merancang struktur database yang efisien untuk toko kue
2. Mengimplementasikan 5 tabel utama yang saling berelasi
3. Menyimpan data pelanggan termasuk nomor telepon
4. Mengelola alamat pengiriman yang lengkap dan detail
5. Mencatat produk kue beserta harga dan kategori
6. Mengelola transaksi pesanan dengan detail item
7. Menghasilkan laporan penjualan dan monitoring stok

### 3. PEMBUATAN DATABASE DAN TABEL

#### Query untuk Membuat Database

```
CREATE DATABASE IF NOT EXISTS toko_kue;  
USE toko_kue;
```

#### Query untuk Membuat Tabel Pelanggan

```
CREATE TABLE pelanggan (  
    id_pelanggan INT PRIMARY KEY AUTO_INCREMENT,  
    nama_pelanggan VARCHAR(100) NOT NULL,  
    nomor_telepon VARCHAR(15) NOT NULL,  
    email VARCHAR(100),  
    tanggal_daftar DATETIME DEFAULT CURRENT_TIMESTAMP  
);
```

#### Query untuk Membuat Alamat Pengiriman

```
CREATE TABLE alamat_pengiriman (  
    id_alamat INT PRIMARY KEY AUTO_INCREMENT,  
    id_pelanggan INT NOT NULL,  
    alamat_lengkap TEXT NOT NULL,  
    kecamatan VARCHAR(50),  
    kota VARCHAR(50) NOT NULL,  
    kode_pos VARCHAR(10),  
    catatan_alamat TEXT,  
    FOREIGN KEY (id_pelanggan) REFERENCES pelanggan(id_pelanggan)  
);
```

#### Query untuk Membuat Tabel Produk Kue

```
CREATE TABLE produk_kue (  
    id_produk INT PRIMARY KEY AUTO_INCREMENT,  
    nama_kue VARCHAR(100) NOT NULL,  
    kategori VARCHAR(50),  
    harga DECIMAL(10,2) NOT NULL,  
    deskripsi TEXT,  
    stok INT DEFAULT 0,  
    status_tersedia BOOLEAN DEFAULT TRUE  
);
```

### Query untuk Membuat Tabel Pesanan

```
CREATE TABLE pesanan (  
    id_pesanan INT PRIMARY KEY AUTO_INCREMENT,  
    id_pelanggan INT NOT NULL,  
    id_alamat INT NOT NULL,  
    tanggal_pesan DATETIME DEFAULT CURRENT_TIMESTAMP,  
    tanggal_pengiriman DATETIME,  
    total_harga DECIMAL(10,2) NOT NULL,  
    status_pesanan VARCHAR(20) DEFAULT 'Pending',  
    metode_pembayaran VARCHAR(30),  
    catatan_pesanan TEXT,  
    FOREIGN KEY (id_pelanggan) REFERENCES pelanggan(id_pelanggan),  
    FOREIGN KEY (id_alamat) REFERENCES alamat_pengiriman(id_alamat)  
);
```

### Query untuk Membuat Tabel Detail Pesanan

```
CREATE TABLE detail_pesanan (  
    id_detail INT PRIMARY KEY AUTO_INCREMENT,  
    id_pesanan INT NOT NULL,  
    id_produk INT NOT NULL,  
    jumlah INT NOT NULL,  
    harga_satuan DECIMAL(10,2) NOT NULL,  
    subtotal DECIMAL(10,2) NOT NULL,  
    catatan_item TEXT,  
    FOREIGN KEY (id_pesanan) REFERENCES pesanan(id_pesanan),  
    FOREIGN KEY (id_produk) REFERENCES produk_kue(id_produk)  
);
```

### Query untuk Menambahkan Data Pelanggan

```
INSERT INTO pelanggan (nama_pelanggan, nomor_telepon, email) VALUES  
('Budi Santoso', '081234567890', 'budi@email.com'),  
('Siti Aminah', '082345678901', 'siti@email.com'),  
('Ahmad Rizki', '083456789012', 'ahmad@email.com'),  
('Dewi Lestari', '084567890123', 'dewi@email.com');
```

### Query untuk Menambahkan Data Produk Kue

```
INSERT INTO produk_kue (nama_kue, kategori, harga, deskripsi, stok)
VALUES
('Kue Ulang Tahun Coklat', 'Birthday Cake', 250000, 'Kue coklat 3
lapis dengan buttercream', 5),
('Kue Ulang Tahun Red Velvet', 'Birthday Cake', 300000, 'Kue red
velvet dengan cream cheese frosting', 3),
('Kue Ulang Tahun Rainbow', 'Birthday Cake', 350000, 'Kue pelangi 3
lapis dengan buttercream warna-warni', 2),
('Kue Ulang Tahun Vanilla', 'Birthday Cake', 240000, 'Kue vanilla
klasik dengan buttercream', 4),
('Cupcake Vanilla', 'Cupcake', 5000, 'Cupcake vanilla dengan topping
cream cheese', 50),
('Cupcake Coklat', 'Cupcake', 5500, 'Cupcake coklat dengan topping
ganache', 45),
('Cupcake Red Velvet', 'Cupcake', 6000, 'Cupcake red velvet dengan
cream cheese', 40),
('Cupcake Matcha', 'Cupcake', 6500, 'Cupcake green tea dengan topping
buttercream', 30),
('Brownies Panggang', 'Brownies', 75000, 'Brownies coklat panggang
ukuran 20x20', 10),
('Brownies Kukus', 'Brownies', 65000, 'Brownies coklat kukus ukuran
20x20', 15),
('Brownies Keju', 'Brownies', 80000, 'Brownies dengan topping keju
premium', 8),
('Tart Buah', 'Tart', 180000, 'Tart dengan topping buah segar', 8),
('Tart Coklat', 'Tart', 200000, 'Tart coklat dengan ganache dan choco
chips', 6),
('Tart Keju', 'Tart', 190000, 'Tart dengan topping keju leleh', 7),
('Kue Lapis Legit', 'Kue Tradisional', 150000, 'Kue lapis legit
premium 18 lapis', 4),
('Nastar Premium', 'Kue Kering', 85000, 'Nastar isi nanas per toples
(500gr)', 20),
('Putri Salju', 'Kue Kering', 80000, 'Kue putri salju per toples
(500gr)', 25),
('Kastengel Keju', 'Kue Kering', 90000, 'Kastengel keju premium per
toples (500gr)', 18),
('Cheese Cake Original', 'Cheese Cake', 120000, 'Cheese cake lembut
ukuran 18cm', 7),
('Tiramisu Cake', 'Cheese Cake', 140000, 'Tiramisu dengan mascarpone
cheese', 5),
('Cheese Cake Blueberry', 'Cheese Cake', 135000, 'Cheese cake dengan
topping blueberry', 6);
```

### Query untuk Menambahkan Data Alamat Pengiriman

```
INSERT INTO alamat_pengiriman (id_pelanggan, alamat_lengkap,
kecamatan, kota, kode_pos, catatan_alamat) VALUES
(1, 'Jl. Merpati No. 15', 'Kebayoran Baru', 'Jakarta Selatan',
'12345', 'Rumah cat hijau, depan minimarket'),
(2, 'Jl. Anggrek Raya No. 88 Blok A5', 'Cengkareng', 'Jakarta Barat',
'11730', 'Komplek Perumahan Anggrek Residence'),
(3, 'Jl. Melati No. 99', 'Tebet', 'Jakarta Selatan', '12820',
'Sebelah kantor pos'),
```

```
(4, 'Jl. Mawar Indah No. 45', 'Kelapa Gading', 'Jakarta Utara',  
'14240', 'Ruko warna putih');
```

### Query untuk Menambahkan Data Pesanan dan Detail

```
-- Pesanan 1  
INSERT INTO pesanan (id_pelanggan, id_alamat, tanggal_pengiriman,  
total_harga, status_pesanan, metode_pembayaran, catatan_pesanan)  
VALUES (1, 1, '2025-10-05 09:00:00', 265000, 'Diproses', 'Transfer  
Bank', 'Mohon dikirim pagi hari');  
  
INSERT INTO detail_pesanan (id_pesanan, id_produk, jumlah,  
harga_satuan, subtotal, catatan_item) VALUES  
(1, 1, 1, 250000, 250000, 'Tulisan: Happy Birthday Mom, warna pink'),  
(1, 5, 3, 5000, 15000, NULL);  
  
-- Pesanan 2  
INSERT INTO pesanan (id_pelanggan, id_alamat, tanggal_pengiriman,  
total_harga, status_pesanan, metode_pembayaran)  
VALUES (2, 2, '2025-10-03 14:00:00', 85000, 'Dikirim', 'COD');  
  
INSERT INTO detail_pesanan (id_pesanan, id_produk, jumlah,  
harga_satuan, subtotal) VALUES  
(2, 16, 1, 85000, 85000);  
  
-- Pesanan 3  
INSERT INTO pesanan (id_pelanggan, id_alamat, tanggal_pengiriman,  
total_harga, status_pesanan, metode_pembayaran, catatan_pesanan)  
VALUES (3, 3, '2025-10-04 16:30:00', 111000, 'Pending', 'E-wallet',  
'Tolong dipacking rapi');
```

### Query untuk Membuat View Pesanan Lengkap

```
CREATE OR REPLACE VIEW view_pesanan_lengkap AS  
SELECT  
    p.id_pesanan,  
    p.tanggal_pesanan,  
    p.tanggal_pengiriman,  
    pel.nama_pelanggan,  
    pel.nomor_telepon,  
    a.alamat_lengkap,  
    a.kota,  
    p.total_harga,  
    p.status_pesanan,  
    p.metode_pembayaran  
FROM pesanan p  
JOIN pelanggan pel ON p.id_pelanggan = pel.id_pelanggan  
JOIN alamat_pengiriman a ON p.id_alamat = a.id_alamat;
```

### Query untuk membuat View Detail Pesanan Lengkap

```
CREATE OR REPLACE VIEW view_detail_pesanan_lengkap AS
SELECT
    d.id_pesanan,
    k.nama_kue,
    k.kategori,
    d.jumlah,
    d.harga_satuan,
    d.subtotal,
    d.catatan_item
FROM detail_pesanan d
JOIN produk_kue k ON d.id_produk = k.id_produk;
```

### Query untuk Menampilkan Data Pelanggan

```
SELECT * FROM pelanggan;
```

### Query untuk Membuat View Pesanan

```
SELECT id_produk, nama_kue, kategori, harga, stok
FROM produk_kue
ORDER BY kategori, harga;
```

### Query untuk Menampilkan Data Alamat Pengiriman

```
SELECT
    a.id_alamat,
    p.nama_pelanggan,
    a.alamat_lengkap,
    a.kecamatan,
    a.kota,
    a.kode_pos,
    a.catatan_alamat
FROM alamat_pengiriman a
JOIN pelanggan p ON a.id_pelanggan = p.id_pelanggan;
```

### Query Data Pesanan Lengkap

```
SELECT * FROM view_pesanan_lengkap ORDER BY tanggal_pesan DESC;
```

### Query untuk Melihat Detail Pesanan Lengkap

```
SELECT * FROM view_detail_pesanan_lengkap;
```

## Query Laporan Penjualan Per Kategori

```
SELECT
    k.kategori,
    COUNT(d.id_detail) as total_item_terjual,
    SUM(d.subtotal) as total_pendapatan
FROM detail_pesanan d
JOIN produk_kue k ON d.id_produk = k.id_produk
JOIN pesanan p ON d.id_pesanan = p.id_pesanan
WHERE p.status_pesanan != 'Dibatalkan'
GROUP BY k.kategori
ORDER BY total_pendapatan DESC;
```

## OUTPUT

Menampilkan tabel Data Pelanggan

	id_pelanggan	nama_pelanggan	nomor_telepon	email	tanggal_daftar
▶	1	Budi Santoso	081234567890	budi@email.com	2025-10-01 13:02:53
	2	Siti Aminah	082345678901	siti@email.com	2025-10-01 13:02:53
	3	Ahmad Rizki	083456789012	ahmad@email.com	2025-10-01 13:02:53
	4	Dewi Lestari	084567890123	dewi@email.com	2025-10-01 13:02:53
✱	NULL	NULL	NULL	NULL	NULL

Penjelasan: Terdapat 4 pelanggan yang terdaftar dalam sistem dengan nomor telepon masing-masing untuk keperluan komunikasi dan pengiriman.

Menampilkan Data Produk Kue

	id_produk	nama_kue	kategori	harga	stok	status_tersedia
▶	4	Kue Ulang Tahun Vanilla	Birthday Cake	240000.00	4	1
	1	Kue Ulang Tahun Coklat	Birthday Cake	250000.00	5	1
	2	Kue Ulang Tahun Red Velvet	Birthday Cake	300000.00	3	1
	3	Kue Ulang Tahun Rainbow	Birthday Cake	350000.00	2	1
	10	Brownies Kukus	Brownies	65000.00	15	1
	11	Brownies Keju	Brownies	80000.00	8	1
	19	Cheese Cake Original	Cheese Cake	120000.00	7	1
	21	Cheese Cake Blueberry	Cheese Cake	135000.00	6	1
	20	Tiramisu Cake	Cheese Cake	140000.00	5	1
	5	Cupcake Vanilla	Cupcake	5000.00	50	1
	6	Cupcake Coklat	Cupcake	5500.00	45	1
	7	Cupcake Red Velvet	Cupcake	6000.00	40	1
	8	Cupcake Matcha	Cupcake	6500.00	30	1
	17	Putri Salju	Kue Kering	80000.00	25	1
	16	Nastar Premium	Kue Kering	85000.00	20	1

18	Kastengel Keju	Kue Kering	90000.00	18	1
15	Kue Lapis Legit	Kue Tradisional	150000.00	4	1
12	Tart Buah	Tart	180000.00	8	1
14	Tart Keju	Tart	190000.00	7	1
13	Tart Coklat	Tart	200000.00	6	1
NULL	NULL	NULL	NULL	NULL	NULL

Penjelasan: Toko memiliki 21 jenis produk kue dengan 8 kategori berbeda. Harga berkisar dari Rp 5.000 (Cupcake) hingga Rp 350.000 (Kue Ulang Tahun Rainbow).

Query Data Alamat Pengiriman

Menampilkan Data Alamat Pengiriman

	nama_pelanggan	alamat_lengkap	kecamatan	kota	kode_pos	catatan_alamat
▶	Budi Santoso	Jl. Merpati No. 15	Kebayoran Baru	Jakarta Selatan	12345	Rumah cat hijau, depan minimarket
	Siti Aminah	Jl. Anggrek Raya No. 88 Blok A5	Cengkareng	Jakarta Barat	11730	Komplek Perumahan Anggrek Residence
	Ahmad Rizki	Jl. Melati No. 99	Tebet	Jakarta Selatan	12820	Sebelah kantor pos
	Dewi Lestari	Jl. Mawar Indah No. 45	Kelapa Gading	Jakarta Utara	14240	Ruko warna putih

Penjelasan: Setiap pelanggan memiliki alamat pengiriman yang lengkap dengan catatan patokan untuk memudahkan kurir dalam pengiriman kue.

Melihat Pesanan Lengkap

id_pesanan	tanggal_pesanan	tanggal_pengiriman	nama_pelanggan	nomor_telepon	alamat_lengkap	kota	total_harga	status_pesanan	metode_p
3	2025-10-01 13:03:52	2025-10-04 16:30:00	Ahmad Rizki	083456789012	Jl. Melati No. 99	Jakarta Selatan	111000.00	Pending	E-wallet
2	2025-10-01 13:03:42	2025-10-03 14:00:00	Siti Aminah	082345678901	Jl. Anggrek Raya No. 88 Blok A5	Jakarta Barat	85000.00	Dikirim	COD
1	2025-10-01 13:03:33	2025-10-05 09:00:00	Budi Santoso	081234567890	Jl. Merpati No. 15	Jakarta Selatan	265000.00	Diproses	Transfer B

Penjelasan: Terdapat 3 pesanan aktif dengan total nilai Rp 461.000. Pesanan tersebar di Jakarta Selatan (2), Jakarta Barat (1) dengan berbagai metode pembayaran.

Melihat Detail Pesanan

id_pesanan	nama_kue	kategori	jumlah	harga_satuan	subtotal	catatan_item
1	Kue Ulang Tahun Coklat	Birthday Cake	1	250000.00	250000.00	Tulisan: Happy Birthday Mom, war
1	Cupcake Vanilla	Cupcake	3	5000.00	15000.00	NULL
2	Nastar Premium	Kue Kering	1	85000.00	85000.00	NULL
3	Brownies Panggang	Brownies	1	75000.00	75000.00	NULL

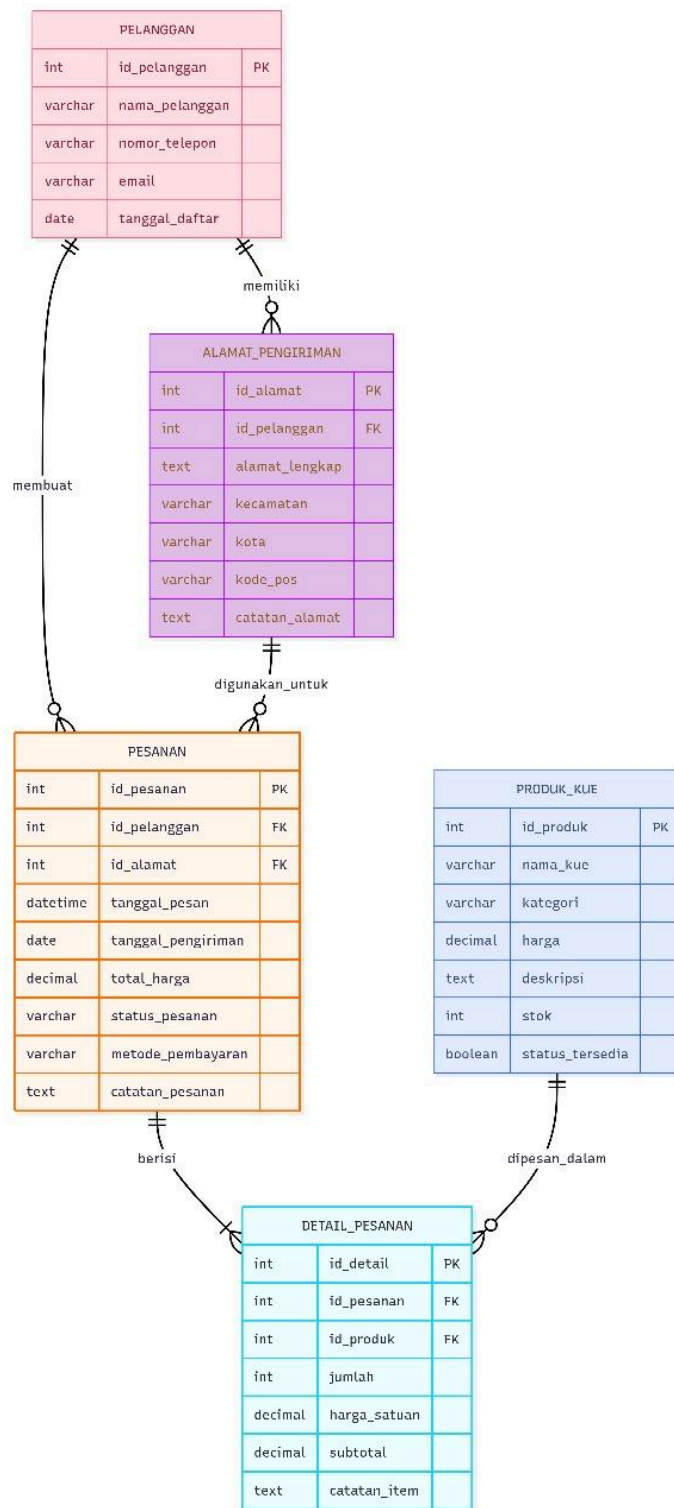
Penjelasan: Total terdapat 5 item berbeda yang dipesan dengan berbagai catatan khusus seperti tulisan di kue dan varian yang diinginkan. Ini menunjukkan fleksibilitas sistem dalam menangani permintaan khusus pelanggan.

Melihat Laporan Penjualan Per Kategori

kategori	total_item_terjual	total_pendapatan
Birthday Cake	1	250000.00
Kue Kering	2	170000.00
Brownies	1	75000.00
Cupcake	2	51000.00



#### 4. Entity Relationship Diagram (ERD)



#### Penjelasan Relasi dalam Diagram:

##### A. PELANGGAN ↔ ALAMAT\_PENGIRIMAN (1:N)

- Arti: Satu pelanggan dapat memiliki banyak alamat pengiriman
- Cara Membaca: "Pelanggan MEMILIKI alamat pengiriman"

- Implementasi: id\_pelanggan di ALAMAT\_PENGIRIMAN adalah FK
- Simbol: Garis dengan label "memiliki (1:N)"
- Contoh Real: Budi Santoso punya alamat rumah dan alamat kantor

#### B. PELANGGAN ↔ PESANAN (1:N)

- Arti: Satu pelanggan dapat melakukan banyak pesanan
- Cara Membaca: "Pelanggan MEMBUAT pesanan"
- Implementasi: id\_pelanggan di PESANAN adalah FK
- Simbol: Garis dengan label "membuat (1:N)"
- Contoh Real: Siti Aminah sudah pesan 5 kali di bulan ini

#### C. ALAMAT\_PENGIRIMAN ↔ PESANAN (1:N)

- Arti: Satu alamat dapat digunakan untuk banyak pesanan
- Cara Membaca: "Alamat DIGUNAKAN\_UNTUK pesanan"
- Implementasi: id\_alamat di PESANAN adalah FK
- Simbol: Garis dengan label "digunakan\_untuk (1:N)"
- Contoh Real: Alamat rumah Budi dipakai untuk 10 pesanan berbeda

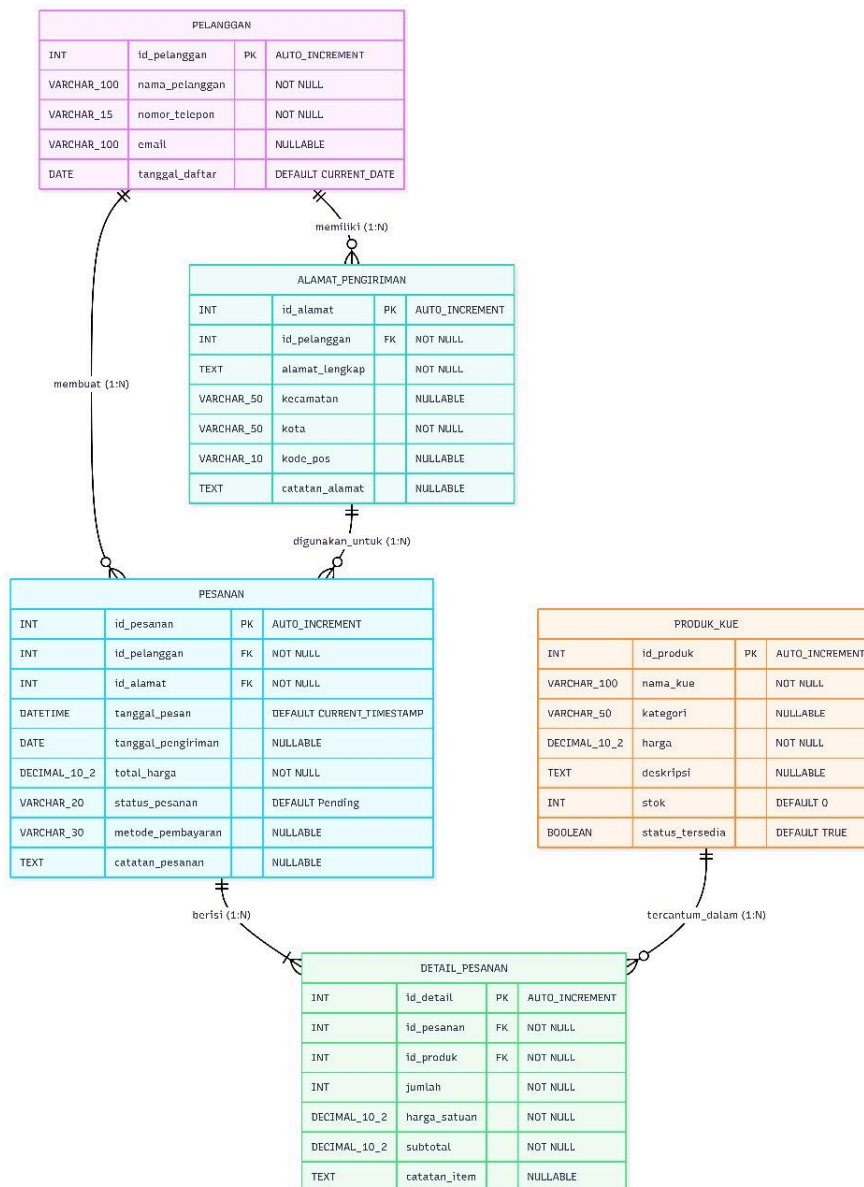
#### D. PESANAN ↔ DETAIL\_PESANAN (1:N)

- Arti: Satu pesanan dapat memiliki banyak item detail
- Cara Membaca: "Pesanan BERISI detail\_pesanan"
- Implementasi: id\_pesanan di DETAIL\_PESANAN adalah FK
- Simbol: Garis dengan label "berisi (1:N)"
- Contoh Real: Pesanan #1 berisi: 1 kue ultah + 3 cupcake + 1 brownies

#### E. PRODUK\_KUE ↔ DETAIL\_PESANAN (1:N)

- Arti: Satu produk dapat muncul di banyak detail pesanan
- Cara Membaca: "Produk TERCANTUM\_DALAM detail\_pesanan"
- Implementasi: id\_produk di DETAIL\_PESANAN adalah FK
- Simbol: Garis dengan label "tercantum\_dalam (1:N)"
- Contoh Real: Cupcake Vanilla terjual di 50 pesanan berbeda

## 5. Physical Data Model (PDM)



### Penjelasan Relasi dalam Diagram:

#### A. PELANGGAN ↔ ALAMAT\_PENGIRIMAN (1:N)

- Arti: Satu pelanggan dapat memiliki banyak alamat pengiriman
- Cara Membaca: "Pelanggan MEMILIKI alamat pengiriman"
- Implementasi: id\_pelanggan di ALAMAT\_PENGIRIMAN adalah FK
- Simbol: Garis dengan label "memiliki (1:N)"
- Contoh Real: Budi Santoso punya alamat rumah dan alamat kantor

#### B. PELANGGAN ↔ PESANAN (1:N)

- Arti: Satu pelanggan dapat melakukan banyak pesanan

- Cara Membaca: "Pelanggan MEMBUAT pesanan"
- Implementasi: id\_pelanggan di PESANAN adalah FK
- Simbol: Garis dengan label "membuat (1:N)"
- Contoh Real: Siti Aminah sudah pesan 5 kali di bulan ini

#### C. ALAMAT\_PENGIRIMAN ↔ PESANAN (1:N)

- Arti: Satu alamat dapat digunakan untuk banyak pesanan
- Cara Membaca: "Alamat DIGUNAKAN\_UNTUK pesanan"
- Implementasi: id\_alamat di PESANAN adalah FK
- Simbol: Garis dengan label "digunakan\_untuk (1:N)"
- Contoh Real: Alamat rumah Budi dipakai untuk 10 pesanan berbeda

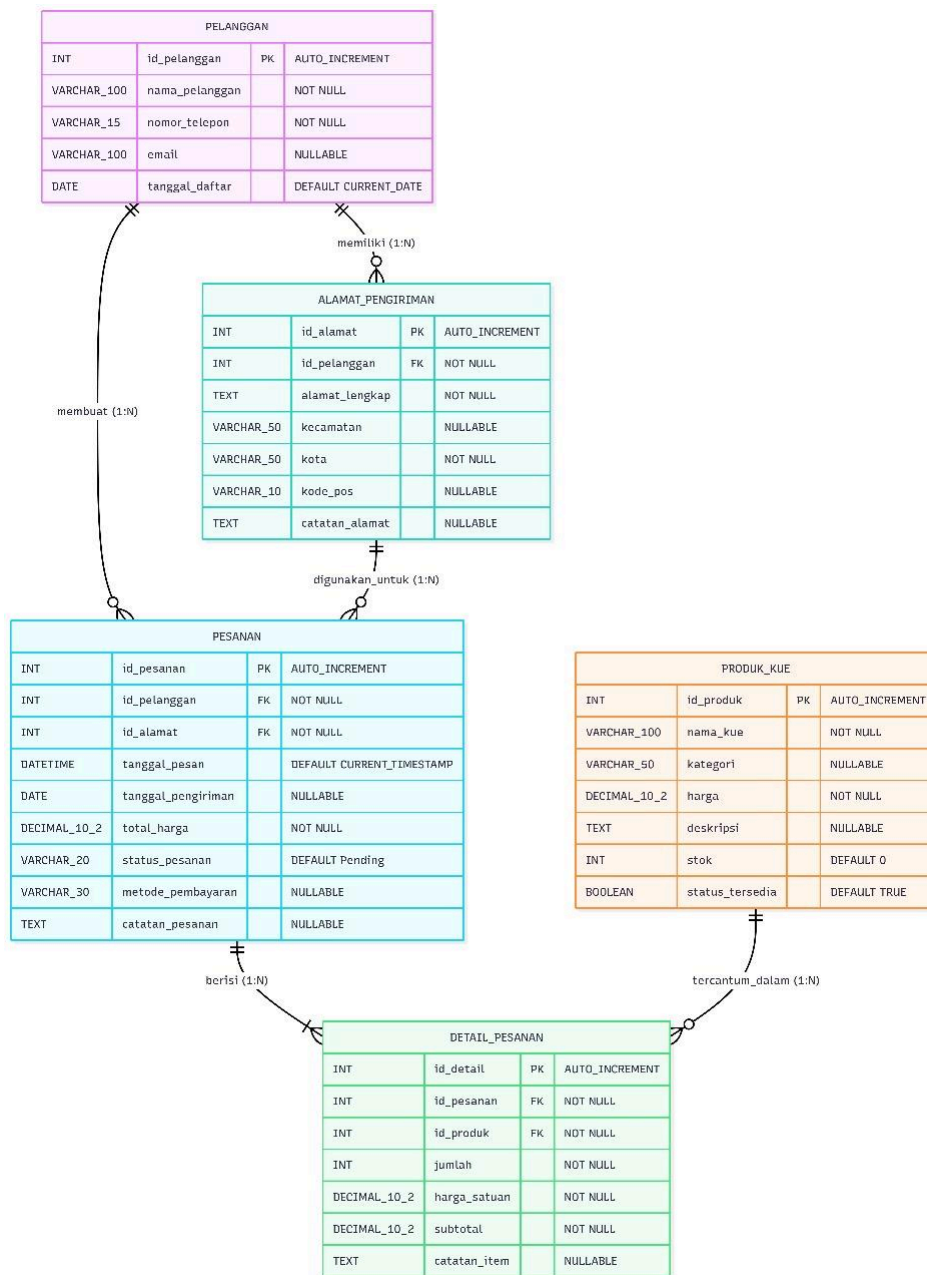
#### D. PESANAN ↔ DETAIL\_PESANAN (1:N)

- Arti: Satu pesanan dapat memiliki banyak item detail
- Cara Membaca: "Pesanan BERISI detail\_pesanan"
- Implementasi: id\_pesanan di DETAIL\_PESANAN adalah FK
- Simbol: Garis dengan label "berisi (1:N)"
- Contoh Real: Pesanan #1 berisi: 1 kue ultah + 3 cupcake + 1 brownies

#### E. PRODUK\_KUE ↔ DETAIL\_PESANAN (1:N)

- Arti: Satu produk dapat muncul di banyak detail pesanan
- Cara Membaca: "Produk TERCANTUM\_DALAM detail\_pesanan"
- Implementasi: id\_produk di DETAIL\_PESANAN adalah FK
- Simbol: Garis dengan label "tercantum\_dalam (1:N)"
- Contoh Real: Cupcake Vanilla terjual di 50 pesanan berbeda

## 6. Conceptual Data Model (CDM)



### Penjelasan Relasi dalam Diagram CDM

#### A. PELANGGAN ↔ ALAMAT\_PENGIRIMAN (1:N)

- Arti: Satu pelanggan dapat memiliki banyak alamat pengiriman
- Cara Membaca: "Pelanggan MEMILIKI alamat pengiriman"
- Konsep Bisnis: Pelanggan dapat menyimpan beberapa alamat tujuan untuk pengiriman (rumah, kantor, alamat kerabat)
- Aturan: Setiap pelanggan minimal harus memiliki satu alamat untuk dapat menerima pesanan
- Contoh Real: Budi Santoso punya alamat rumah dan alamat kantor

## B. PELANGGAN ↔ PESANAN (1:N)

- Arti: Satu pelanggan dapat melakukan banyak pesanan
- Cara Membaca: "Pelanggan MEMBUAT pesanan"
- Konsep Bisnis: Pelanggan dapat berbelanja berkali-kali, sistem menyimpan riwayat semua transaksi pelanggan
- Aturan: Setiap pesanan harus dibuat oleh pelanggan terdaftar, pelanggan baru boleh belum punya pesanan
- Contoh Real: Siti Aminah sudah pesan 5 kali di bulan ini

## C. ALAMAT\_PENGIRIMAN ↔ PESANAN (1:N)

- Arti: Satu alamat dapat digunakan untuk banyak pesanan
- Cara Membaca: "Alamat DIGUNAKAN\_UNTUK pesanan"
- Konsep Bisnis: Alamat yang sudah tersimpan dapat digunakan berulang kali tanpa perlu input ulang
- Aturan: Setiap pesanan harus memiliki alamat tujuan pengiriman, alamat boleh belum pernah dipakai
- Contoh Real: Alamat rumah Budi dipakai untuk 10 pesanan berbeda

## D. PESANAN ↔ DETAIL\_PESANAN (1:N)

- Arti: Satu pesanan dapat memiliki banyak item detail
- Cara Membaca: "Pesanan BERISI detail\_pesanan"
- Konsep Bisnis: Dalam satu transaksi, pelanggan dapat membeli beberapa jenis produk sekaligus
- Aturan: Setiap pesanan minimal berisi 1 item produk, detail tidak bisa ada tanpa pesanan induk
- Contoh Real: Pesanan #1 berisi: 1 kue ultah + 3 cupcake + 1 brownies

## E. PRODUK\_KUE ↔ DETAIL\_PESANAN (1:N)

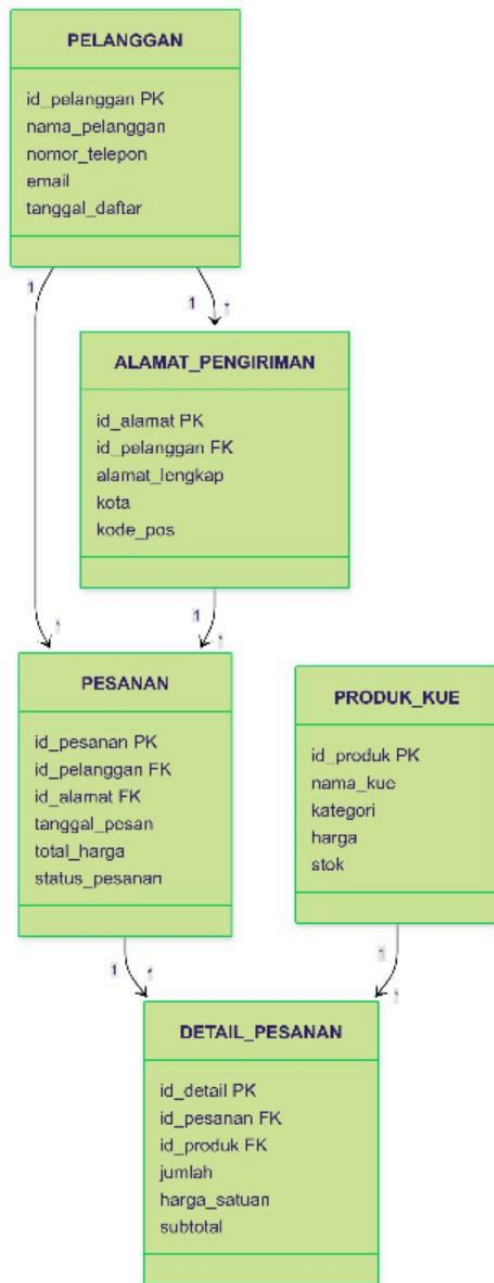
- Arti: Satu produk dapat muncul di banyak detail pesanan
- Cara Membaca: "Produk TERCANTUM\_DALAM detail\_pesanan"
- Konsep Bisnis: Produk yang sama dapat dibeli oleh banyak pelanggan di waktu berbeda
- Aturan: Setiap detail harus merujuk ke produk yang ada, produk boleh belum pernah dipesan
- Contoh Real: Cupcake Vanilla terjual di 50 pesanan berbeda

## F. Relasi Many-to-Many (Implicit): PESANAN ↔ PRODUK\_KUE

- Arti: Satu pesanan dapat memiliki banyak produk, satu produk dapat ada di banyak pesanan
- Cara Membaca: "Pesanan dan Produk memiliki relasi banyak-ke-banyak melalui Detail Pesanan"

- Konsep Bisnis: Hubungan fleksibel antara pesanan dan produk dimediasi oleh entitas penghubung
- Aturan: Relasi N:M dipecah menjadi dua relasi 1:N untuk menghindari kompleksitas
- Contoh Real: Pesanan #1 punya 3 produk, Cupcake Vanilla ada di 50 pesanan (subtotal, catatan\_item)
- Fleksibel untuk query dan analisis
- Normalisasi database terjaga

## **7. Struktur Tabel Database**



## Penjelasan Relasi dalam Tabel

### A. PELANGGAN ↔ ALAMAT\_PENGIRIMAN (1:\*)

- Arti: Satu pelanggan dapat memiliki banyak alamat pengiriman
- Cara Membaca: "Pelanggan MEMILIKI alamat pengiriman"
- Kardinalitas: 1 (di sisi PELANGGAN) ke \* (di sisi ALAMAT\_PENGIRIMAN)
- Koneksi: id\_pelanggan FK di tabel ALAMAT\_PENGIRIMAN
- Simbol: Garis dari PELANGGAN ke ALAMAT\_PENGIRIMAN dengan notasi "1" dan "\*"



- Arti: Satu produk dapat muncul di banyak detail pesanan
- Cara Membaca: "Produk TERCANTUM\_DALAM detail\_pesanan"
- Kardinalitas: 1 (di sisi PRODUK\_KUE) ke \* (di sisi DETAIL\_PESANAN)
- Koneksi: id\_produk FK di tabel DETAIL\_PESANAN
- Simbol: Garis dari PRODUK\_KUE ke DETAIL\_PESANAN dengan notasi "1" dan "\*"
- Contoh Real: Cupcake Vanilla terjual di 50 pesanan berbeda

