

# Database Penjualan & Pemesanan

Proyek Akhir Mata Kuliah Database Semester Genap  
2024/2025

Kelompok 4 Database (A)  
Dosen Pengampu : Devvi Sarwinda, M.Si.



# Anggota Kelompok



Siti Nur Salamah  
2206048833



Raissa Anggia Maharani  
2206048581



Widya Siti Ropiah  
2206048745



Bryan Jonathan  
2206052780



Hasthabrata Christopher Liatna  
2206824741

# Latar Belakang

## Hal Utama

Masih banyak pelaku UMKM yang menjalankan proses pencatatan transaksi penjualan dan pemesanan secara manual.

### Tentu menimbulkan masalah karena



#### Tingginya risiko kehilangan data.

Nota transaksi yang berbentuk fisik sangat rentan terhadap kerusakan, kehilangan, atau kesalahan penyimpanan.



#### Pencatatan stok barang secara manual seringkali tidak akurat.

Ketidaksesuaian antara stok fisik dan catatan yang ada, yang pada akhirnya berisiko menyebabkan kekurangan atau kelebihan persediaan.



#### Pelayanan kepada pelanggan dapat terganggu.

Keterlambatan dalam memberikan informasi kepada pelanggan dapat berdampak pada menurunnya kepuasan dan kepercayaan pelanggan terhadap layanan yang diberikan.

**Diperlukan sebuah sistem database yang dapat menyimpan, mengelola, dan memproses data transaksi penjualan dan pemesanan secara terstruktur dan efisien.**

### Keuntungan

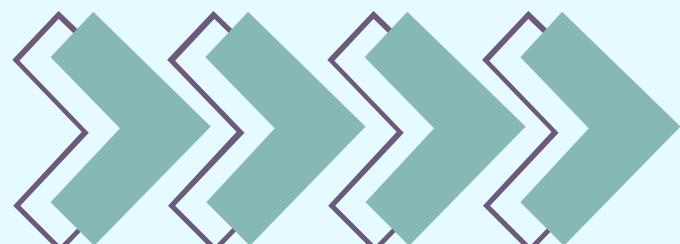
- 1 Pelaku UMKM dapat mencatat data pelanggan, produk, pemesanan, pembayaran, dan stok secara lebih akurat dan *real-time*.
- 2 Proses pencarian dan pelaporan data menjadi lebih cepat sehingga mempercepat pengambilan keputusan dan meningkatkan efisiensi operasional.
- 3 Sistem database dapat meminimalisir *human error* karena proses pencatatan, perhitungan, dan pengolahan data dilakukan secara otomatis dan terstruktur.
- 4 Meningkatkan kualitas pelayanan pelanggan sehingga meningkatkan kepuasan, kepercayaan, dan loyalitas pelanggan terhadap usaha yang dijalankan.

Dari latar belakang, didapat rumusan masalahnya sebagai berikut



**Bagaimana merancang sistem database yang efektif untuk mempercepat proses pencarian, pencatatan, pengelolaan, dan pelaporan data transaksi secara cepat, akurat, *real-time*, aman, serta mendukung pengambilan keputusan dan peningkatan kualitas layanan bagi pelaku usaha?**

Didapatkan tujuan dari perancangan database ini, yaitu...



1

Mempercepat proses pencarian dan pelaporan data melalui penyediaan akses informasi yang cepat, akurat, dan *real-time* bagi pelaku usaha.

2

Meminimalisasi *human error* karena proses pencatatan, perhitungan, dan pengolahan data dilakukan secara otomatis dan terstruktur.

3

Mendukung pengambilan keputusan bisnis yang lebih tepat dan strategis melalui ketersediaan data transaksi yang lengkap, terorganisir, dan aman.

# Aluran Penelitian



# Rancangan Database



**Customer:** Untuk menyimpan informasi tentang pelanggan yang memesan pada toko.



**Product:** Untuk menyimpan informasi tentang produk yang tersedia pada toko.



**Order:** Untuk menyimpan informasi tentang status pesanan oleh customer.



**Order\_Detail:** Untuk menyimpan informasi tentang product yang dipesan oleh customer.



**Order\_Payment:** Untuk menyimpan informasi tentang pembayaran yang perlu dilakukan oleh customer.

# Rancangan Database

Tabel: Entitas, Atribut, Key, Domain

Entitas	Atribut	Key	Domain
customers	customer_id	Primary Key	Kumpulan karakter unik sebagai ID pelanggan (misal: C001, C002, ...)
	name_cust		Kumpulan karakter string yang merepresentasikan nama pelanggan
	email		Kumpulan karakter string yang merepresentasikan email pelanggan
	phone_number		Kumpulan angka (10-13 digit) yang merepresentasikan nomor telepon pelanggan
	address		Kumpulan karakter string yang merepresentasikan alamat pelanggan
	password		kumpulan karakter string yang merepresentasikan password akun customer

# Rancangan Database

Tabel: Entitas, Atribut, Key, Domain

Entitas	Atribut	Key	Domain
Product	product_id	Primary Key	Kumpulan karakter unik yang merepresentasikan identitas unik setiap produk
	produt_name		kumpulan karakter string yang merepresentasikan nama produk
	product_description		kumpulan karakter string yang merepresentasikan detail produk (bahan, ukuran, dll)
	price		kumpulan digit angka yang merepresentasikan harga setiap produk
	category		kumpulan string yang merepresentasikan kategori produk (elektronik, makanan, dll)
	stock		kumpulan digit angka yang merepresentasikan jumlah stok produk yang tersedia
	rating_product		kumpulan digit angka yang merepresentasikan penilaian pelanggan terhadap produk, biasanya dalam skala 1–5.

# Rancangan Database

Tabel: Entitas, Atribut, Key, Domain

Entitas	Atribut	Key	Domain
order_details	order_detail_id	Primary Key	Kumpulan karakter unik sebagai ID detail pesanan
	order_id	Foreign Key	ID pemesanan terkait
	product_id	Foreign Key	ID produk yang dipesan
	quantity		Kumpulan angka yang menunjukkan jumlah produk yang dipesan
	price_total		Total harga per produk (quantity × price)

# Rancangan Database

Tabel: Entitas, Atribut, Key, Domain

Entitas	Atribut	Key	Domain
orders	order_id	Primary Key	ID pemesanan terkait
	customer_id	Foreign Key	Kumpulan karakter unik sebagai ID pelanggan (misal: C001, C002, ...)
	order_date		tanggal pemesanan
	status		status pemesanan ( pending, diproses, dikirim, selesai )

# Rancangan Database

Tabel: Entitas, Atribut, Key, Domain

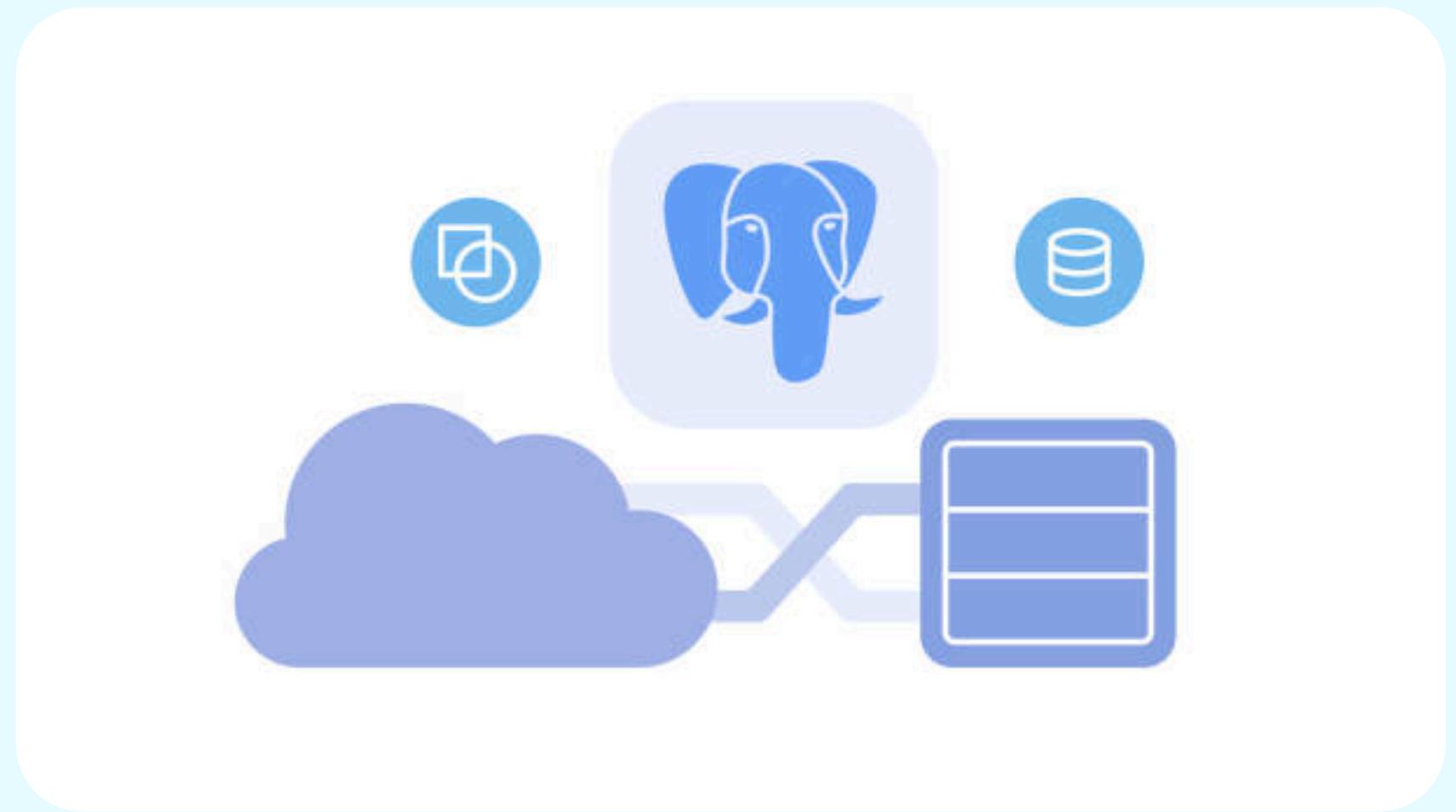
Entitas	Atribut	Key	Domain
order_payments	payment_id	Primary Key	Kumpulan karakter unik sebagai ID detail pembayaran
	order_id	Foreign Key	ID pemesanan terkait
	payment_type		tipe pembayaran yang digunakan
	payment_date		tanggal pembayaran
	payment_status		Status pembayaran (Paid atau Unpaid)
	amount_paid		Total harga yang dibayar (termasuk pajak)

# Entity Relationship Diagram

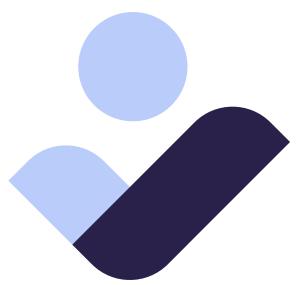


# Entity Relationship Diagram

Entitas Terkait	Jenis Kardinalitas	Relasi	Penjelasan Relasi
customers - orders	1:M	Memiliki	<p>Satu customer dapat memiliki banyak order.</p> <p>Setiap order harus milik satu customer.</p> <p>Tidak semua customer harus membuat order.</p>
orders - order_payments	1:1	Memiliki	<p>Satu order hanya memiliki satu order payment.</p> <p>Setiap order harus memiliki pembayaran dan sebaliknya.</p>
orders - order_details	1:M	Terdiri	<p>Satu order dapat terdiri dari banyak order detail.</p> <p>Setiap order harus memiliki order detail</p>
product - order_detail	M:1	Termuat	Satu produk dapat muncul di banyak order detail



# IMPLEMENTASI SQL



# OVERVIEW DATABASE

- Rancangan database sebelumnya kemudian diimport menjadi sebuah database dengan software DB Browser. Gambar di bawah adalah cuplikan informasi dari database.
- Sebelumnya pada rancangan hanya ada 6 tabel, akan tetapi pada database ini terdiri dari 7 tabel dimana “table\_sequence” adalah tabel yang memberi informasi jumlah observasi setiap tabel.
- Data yang digunakan dalam database ini bersifat dummy.

Name	Type	Schema
Tables (7)		
> admin		CREATE TABLE "admin" ( "email" TEXT UNIQUE, "password" TEXT )
> customers		CREATE TABLE "customers" ( "customer_id" TEXT NOT NULL UNIQUE, "customer_name" TEXT, "customer_email" TEXT, "customer_password" TEXT )
> order_details		CREATE TABLE "order_details" ( "order_detail_id" TEXT UNIQUE, "order_id" TEXT, "product_id" TEXT, "quantity" INTEGER )
> order_payments		CREATE TABLE "order_payments" ( "payment_id" TEXT UNIQUE, "order_id" TEXT, "amount" REAL )
> orders		CREATE TABLE "orders" ( "order_id" TEXT UNIQUE, "customer_id" TEXT, "order_date" TEXT, "order_status" TEXT )
> product		CREATE TABLE "product" ( "product_id" TEXT UNIQUE, "product_name" TEXT, "product_description" TEXT, "product_price" REAL )
> table_sequence		CREATE TABLE table_sequence(name,seq)
Indices (0)		
Views (0)		
Triggers (0)		

# CREATE TABLE

## 1 Tabel Admin

Name	Type	NN	PK	AI	U
email	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
password	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

```
CREATE TABLE "admin" (
    "email" TEXT NOT NULL UNIQUE,
    "password" TEXT NOT NULL
);
```

## 2 Tabel Customers

Name	Type	NN	PK	AI	U
customer_id	TEXT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
name_cust	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
email	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
address	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
phone_number	INTEGER	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
password	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

```
CREATE TABLE "customers" (
    "customer_id" TEXT NOT NULL UNIQUE,
    "name_cust" TEXT NOT NULL,
    "email" TEXT NOT NULL,
    "address" TEXT NOT NULL,
    "phone_number" INTEGER NOT NULL,
    "password" TEXT NOT NULL,
    PRIMARY KEY("customer_id")
);
```

## 3 Tabel Order\_Details

Name	Type	NN	PK	AI	U
order_detail_id	TEXT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
order_id	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
product_id	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
quantity	INTEGER	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
price_total	REAL	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

```
CREATE TABLE "order_details" (
    "order_detail_id" TEXT NOT NULL UNIQUE,
    "order_id" TEXT NOT NULL,
    "product_id" TEXT NOT NULL,
    "quantity" INTEGER NOT NULL,
    "price_total" REAL NOT NULL,
    PRIMARY KEY("order_detail_id"),
    FOREIGN KEY("order_id") REFERENCES "orders"("order_id"),
    FOREIGN KEY("product_id") REFERENCES "products"("product_id")
);
```

## 4 Tabel Order\_Payments

Name	Type	NN	PK	AI	U
payment_id	TEXT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
payment_date	TEXT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
payment_type	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
amount_paid	INTEGER	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
payment_status	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
order_id	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

```
CREATE TABLE "order_payments" (
    "payment_id" TEXT NOT NULL UNIQUE,
    "payment_date" TEXT NOT NULL,
    "payment_type" TEXT NOT NULL,
    "amount_paid" INTEGER NOT NULL,
    "payment_status" TEXT NOT NULL,
    "order_id" TEXT NOT NULL,
    PRIMARY KEY("payment_id"),
    FOREIGN KEY("order_id") REFERENCES "orders"("order_id")
);
```

# CREATE TABLE

## 5 Tabel Products

Name	Type	NN	PK	AI	U
product_id	TEXT	✓	✓	□	✓
product_name	TEXT	✓	□	□	□
category	TEXT	✓	□	□	□
price	INTEGER	✓	□	□	□
product_description	TEXT	✓	□	□	□
stock	INTEGER	✓	□	□	□
rating_product	INTEGER	✓	□	□	□

```
CREATE TABLE "products" (
    "product_id"    TEXT NOT NULL UNIQUE,
    "product_name"  TEXT NOT NULL,
    "category"      TEXT NOT NULL,
    "price"         INTEGER NOT NULL,
    "product_description" TEXT NOT NULL,
    "stock"         INTEGER NOT NULL,
    "rating_product"  INTEGER NOT NULL,
    PRIMARY KEY("product_id")
);
```

## 6

### Tabel Orders

Name	Type	NN	PK	AI	U
order_id	TEXT	✓	✓	□	✓
customer_id	TEXT	✓	□	□	□
order_date	TEXT	✓	□	□	□
status	TEXT	✓	□	□	□

```
CREATE TABLE "orders" (
    "order_id"    TEXT NOT NULL UNIQUE,
    "customer_id" TEXT NOT NULL,
    "order_date"  TEXT NOT NULL,
    "status"      TEXT NOT NULL,
    PRIMARY KEY("order_id"),
    FOREIGN KEY("customer_id") REFERENCES "customers"("customer_id")
);
```

## Banyak Observasi Setiap Tabel

	name	seq
1	admin	5
2	customers	80
3	order_details	476
4	order_payments	150
5	orders	150
6	product	30

# INSERT INTO

Memasukkan data kedalam suatu tabel

```
INSERT INTO order_details(order_detail_id, order_id, product_id,  
quantity, price_total) VALUES ('OD2001', 'ORD0001', 'P001', 2, 39.98);
```

472	OD00472	ORD0149	PROD002	1	496.25
473	OD00473	ORD0149	PROD015	3	691.92
474	OD00474	ORD0149	PROD003	3	1204.95
475	OD00475	ORD0149	PROD019	2	372.38
476	OD00476	ORD0150	PROD024	3	1453.62
477	OD2001	ORD0001	P001	2	39.98

Data berhasil di tambahkan pada observasi 477.

# UPDATE

Memperbarui data dalam suatu tabel

```
UPDATE order_payments SET payment_status = "paid" WHERE payment_id = 'PAY0003'
```

Sebelum

	payment_id	payment_date	payment_type	amount_paid	payment_status	order_id
1	PAY0001	2025-02-18 04:50:31	debit_card	345.68	refunded	ORD0001
2	PAY0002	2025-04-20 17:23:40	credit_card	872.11	refunded	ORD0002
3	PAY0003	2025-03-27 19:12:47	credit_card	329.4	unpaid	ORD0003
4	PAY0004	2025-01-13 13:50:05	credit_card	1068.02	refunded	ORD0004

Sesudah

	payment_id	payment_date	payment_type	amount_paid	payment_status	order_id
1	PAY0001	2025-02-18 04:50:31	debit_card	345.68	refunded	ORD0001
2	PAY0002	2025-04-20 17:23:40	credit_card	872.11	refunded	ORD0002
3	PAY0003	2025-03-27 19:12:47	credit_card	329.4	paid	ORD0003
4	PAY0004	2025-01-13 13:50:05	credit_card	1068.02	refunded	ORD0004

# SELECT WHERE

Memilih bagian dari tabel yang diinginkan

```
SELECT payment_id, payment_type, payment_status FROM order_payments  
WHERE (payment_status) = 'paid' AND payment_type = 'debit_card'
```

	payment_id	payment_type	payment_status
1	PAY0006	debit_card	paid
2	PAY0016	debit_card	paid
3	PAY0059	debit_card	paid
4	PAY0067	debit_card	paid
5	PAY0070	debit_card	paid
6	PAY0077	debit_card	paid
7	PAY0085	debit_card	paid
8	PAY0088	debit_card	paid
9	PAY0090	debit_card	paid
10	PAY0106	debit_card	paid
11	PAY0138	debit_card	paid

# SELECT ORDER BY

Seleksi data dari tabel dengan urutan kolom tertentu

```
SELECT price, product_name FROM products ORDER BY price
```

	price	product_name
1	45	Mini Screwdriver Set
2	55	Indoor Plant Pot
3	60	Scented Candle Set
4	85	Document Storage Box
5	90	Desktop Whiteboard
6	95	Multipurpose Screwdriver Kit
7	95	Office Desk Organizer
8	110	Collapsible Storage Bin
9	120	Smart LED Bulb
10	135	USB Rechargeable Flashlight
11	145	Ergonomic Footrest
12	170	Adjustable Monitor Stand

# SELECT COUNT

Menghitung jumlah dari suatu record berdasarkan kriteria yang diinginkan pengguna

```
SELECT count(*) FROM order_payments WHERE payment_status = 'unpaid'
```

count(*)
1 50

Banyaknya pelanggan yang belum membayar adalah 50

# JOIN ON

Menggabungkan beberapa tabel berdasarkan tabel yang berkaitan

```
SELECT customers.customer_id, order_payments.payment_status  
FROM customers  
JOIN orders ON customers.customer_id = orders.customer_id  
JOIN order_payments ON orders.order_id = order_payments.order_id  
WHERE order_payments.payment_status = 'paid';
```

	customer_id	payment_status
1	CUST075	paid
2	CUST010	paid
3	CUST003	paid
4	CUST058	paid
5	CUST005	paid
6	CUST062	paid
7	CUST047	paid

# DELETE

Menghapus data pada tabel

```
DELETE FROM products WHERE product_id='P002'
```

Sebelum

Table: products							
	product_id	product_name	category	price	product_description	stock	rating_product
1	P001	Smart LED Bulb	Electronics	120	An energy-efficient smart LED bulb ...	50	4.5
2	P002	Portable Electric Kettle	Electronics	299.99	A compact and portable electric ...	30	4.2
3	P003	Mini Air Purifier	Electronics	349	Mini air purifier to remove dust, ...	20	4
4	P004	Wireless Extension Plug	Electronics	250	A wireless extension plug that ...	28	4.3
5	P005	Smart Light Switch	Electronics	190	Smart light switch that allows app ...	22	4.5

Setelah

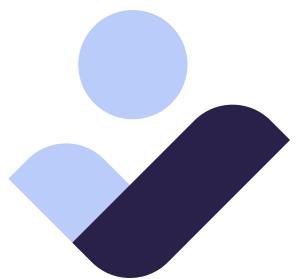
	product_id	product_name	category	price	product_description	stock	rating_product
1	P001	Smart LED Bulb	Electronics	120	An energy-efficient smart LED bulb ...	50	4.5
2	P003	Mini Air Purifier	Electronics	349	Mini air purifier to remove dust, ...	20	4
3	P004	Wireless Extension Plug	Electronics	250	A wireless extension plug that ...	28	4.3
4	P005	Smart Light Switch	Electronics	190	Smart light switch that allows app ...	22	4.5
5	P006	USB Rechargeable Flashlight	Electronics	135	A compact and powerful USB ...	40	4.4

Terlihat data dengan product\_id='P002'  
sudah terhapus

Dapat disimpulkan bahwa implementasi SQL berhasil diterapkan pada database "penjualan\_new.db"



# IMPLEMENTASI GUI DAN TESTING



# Implementasi GUI

**Graphical User Interface (GUI)** merupakan tampilan antarmuka komputer yang memungkinkan pengguna berinteraksi dengan perangkat lunak atau sistem komputer menggunakan elemen grafis seperti ikon, tombol, jendela, dan menu.

Tujuan  
GUI

Membuat pengalaman pengguna lebih intuitif dan mudah digunakan dibandingkan dengan berinteraksi dengan perintah teks atau baris perintah (Xie et al., 2022).

**Pada sistem database penjualan dan pemesanan memiliki peranan penting yaitu...**

-  Mempermudah pengguna untuk melakukan berbagai fungsi seperti pencatatan pesanan, pengecekan status pembayaran, pengelolaan stok, hingga pembuatan laporan penjualan.
-  Mempermudah pengaksesan dan melakukan manipulasi data yang tersimpan dalam database dengan lebih efisien dan minim kesalahan.

# Package yang Digunakan

## 1 **sqlite3**

Untuk membuat dan mengelola database ringan yang memungkinkan operasi CRUD, penyimpanan data tanpa server terpisah, serta eksekusi query SQL untuk pelaporan dan analisis.

## 2 **pandas**

Untuk mengonversi hasil query database ke format DataFrame yang mudah dibaca, kemudian bisa digunakan untuk agregasi seperti total dan rata-rata, pengelompokan berdasarkan kategori, serta pemrosesan data untuk laporan, dashboard, filtering, dan sorting.

## 3 **ipywidgets**

Untuk membuat widget interaktif berbasis HTML seperti form input, slider, checkbox, dropdown, dan tombol yang memungkinkan pencatatan data langsung dari pengguna, penampilan data dalam tabel interaktif.

## 4 **IPython**

Untuk operasi seperti akses database dan manipulasi file dan terintegrasi dengan ipywidgets untuk menciptakan antarmuka interaktif.

## 5 **qrcode**

Untuk menghasilkan QR code dalam bentuk objek PillImage yang dapat disisipkan pada invoice, serta diintegrasikan dengan sistem pembayaran digital.

## 6 **base64**

Untuk mengonversi QR code menjadi representasi teks yang mudah ditransmisikan.

## 7 **PIL**

Untuk pemrosesan gambar QR code yang dihasilkan, seperti menyesuaikan ukuran atau menambahkan elemen visual tambahan.

## 8 **re**

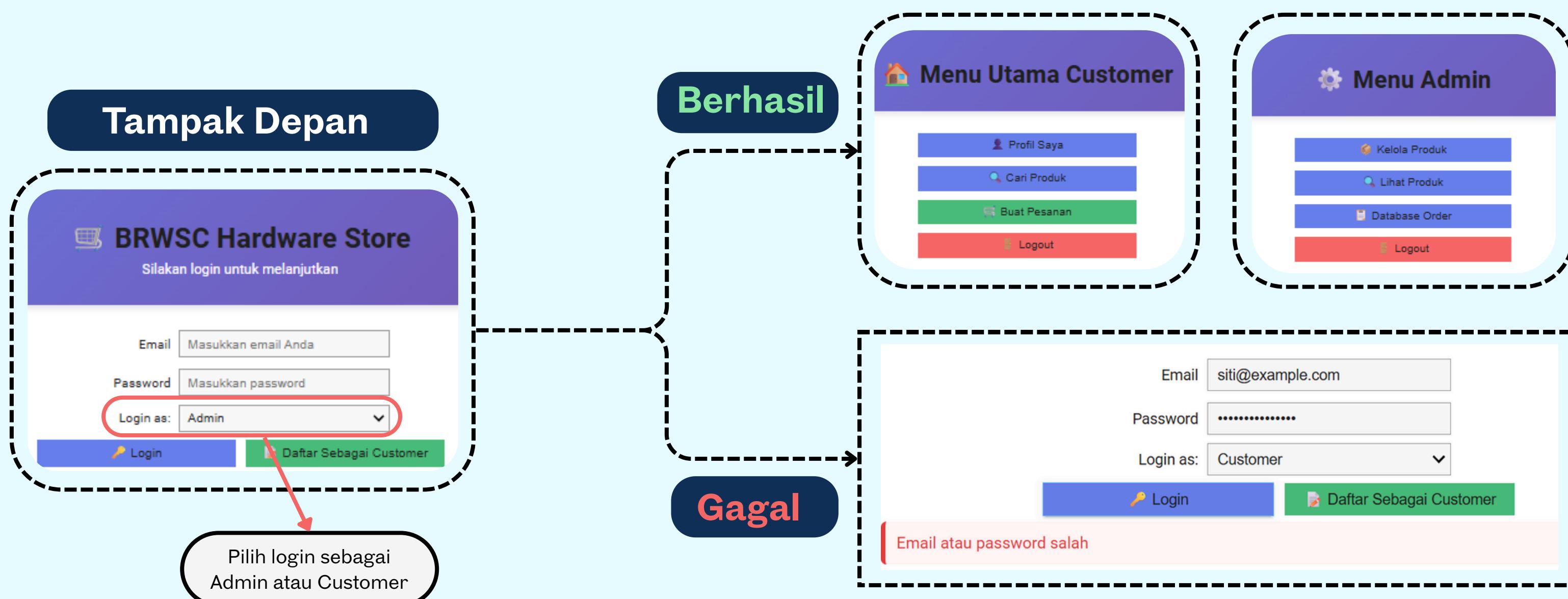
Untuk memvalidasi format input dari pengguna, seperti alamat email, nomor telepon, atau kode produk.

# Fitur Login

## LOGIN

jika pengguna memilih “**Login**”, pengguna akan diminta untuk memasukkan email dan *password*. Kemudian pengguna akan memilih status login sebagai customer atau admin. Sistem kemudian memeriksa apakah email dan password cocok dengan akun yang tersedia pada database.

- Jika **login berhasil**, pengguna akan diarahkan ke menu utama baik sebagai customer maupun admin, tergantung jenis akunnya.
- Jika **login gagal**, pengguna akan diberitahu pesan bahwa email atau *password* yang dimasukkan salah.



# Fitur Registrasi

## Registrasi Customer

Jika belum memiliki akun, pengguna dapat melakukan registrasi terlebih dahulu untuk membuat akun baru

### Melakukan registrasi

Email

Password

Login as:

Pengguna dapat memilih mendaftar sebagai customer jika belum memiliki akun

### Mengisi data

### Daftar Akun Baru

Nama

Email

Alamat

Telepon

Password

Konfirmasi

Pada tahap daftar akun, pengguna akan diarahkan untuk mengisi data berupa nama, **email belum pernah terdaftar**, alamat, telepon, dan **password**

Kondisi **jika email** yang pengguna masukkan **sudah ada dalam data**

Email sudah terdaftar

Jika **berhasil daftar** maka akan ada pemberitahuan bahwa registrasi berhasil dan pengguna dalam kembali ke fitur login

Registrasi berhasil!

← Kembali

# Menu Utama

## Menu Utama

Menu utama akan menampilkan fitur-fitur yang sesuai dengan menu utama customer atau admin

### MAIN MENU CUSTOMER

Menu utama sebagai customer mencakup berbagai fitur, yaitu:

1. Melihat profil
2. Mencari produk
3. Membuat pesanan
4. Logout

### MAIN MENU ADMIN

Menu utama sebagai admin mencakup berbagai fitur, yaitu:

1. Mengelolah produk
2. Melihat produk
3. Melihat database order
4. Logout

### Main Menu Customer

#### Menu Utama Customer

👤 Profil Saya

🔍 Cari Produk

🛒 Buat Pesanan

Logout

### Main Menu Admin

#### Menu Admin

📦 Kelola Produk

🔍 Lihat Produk

📋 Database Order

Logout

# Fitur Customer



Fitur “**Profil Saya**” memungkinkan pengguna untuk melihat informasi pribadi, seperti nama, email, alamat, dan telepon. Pengguna juga dapat melihat statistik pembelian dan detail mengenai order barang.

1

The image shows a screenshot of the 'Informasi Personal' (Personal Information) section. It displays the following details for a user named Carl Hall:

- Nama: Carl Hall
- Email: sharonmurphy@turner.biz
- Alamat: 83188 Baker Station Apt. 507, Hayesmouth, MO 88765
- Telepon: +1-844-517-6840x18594

A red dashed arrow points from the 'Profil Saya' button in the main menu to this information.

**Informasi personal**  
pengguna berupa nama, email, alamat, dan telepon

2

The image shows a screenshot of the 'Statistik Pembelian' (Purchase Statistics) section. It displays the following statistics:

- 2 Total Order
- Rp 2,025,050 Total Belanja

A red dashed arrow points from the 'Profil Saya' button in the main menu to this section.

**Statistik pembelian** yang berisi total order dan total belanja pengguna

3

The image shows a screenshot of the 'Detail order' (Order Details) section. It displays two orders:

- Order ORD0059:**
  - Tanggal: 2025-05-16 06:26:39
  - Status: pending
  - Status Pembayaran: paid
  - Total: Rp 902,480

Produk	Jumlah	Harga Satuan	Total
WiFi Smart Plug	2	Rp175,000	Rp657,000
Desktop Whiteboard	2	Rp90,000	Rp345,520
Laptop Cooling Pad	1	Rp199,000	Rp254,230
- Order ORD0020:**
  - Tanggal: 2025-01-05 18:17:28
  - Status: delivered
  - Status Pembayaran: refunded
  - Total: Rp 1,122,570

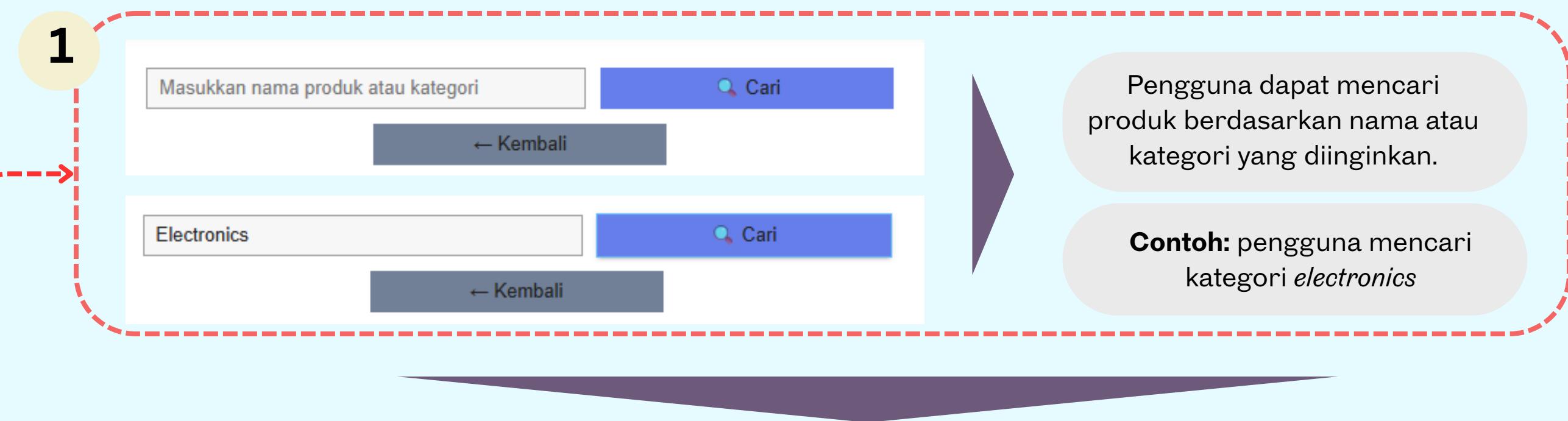
Produk	Jumlah	Harga Satuan	Total
Collapsible Storage Bin	1	Rp110,000	Rp21,450
Wall Mounted Shoe Rack	1	Rp280,000	Rp316,630
Wireless Extension Plug	1	Rp250,000	Rp374,850
Office Desk Organizer	2	Rp95,000	Rp969,080

**Detail order** pengguna yang berisi tanggal pemesanan, status pengiriman (*cancelled, delivered, shipped, pending, proceed*), status pembayaran (*paid, unpaid, refunded*), dan total yang akan dibayarkan.

# Fitur Customer



Fitur “**Cari Produk**” memungkinkan pengguna untuk mencari dan melihat produk-produk yang dijual.



Pengguna dapat mencari produk berdasarkan nama atau kategori yang diinginkan.

**Contoh:** pengguna mencari kategori *electronics*

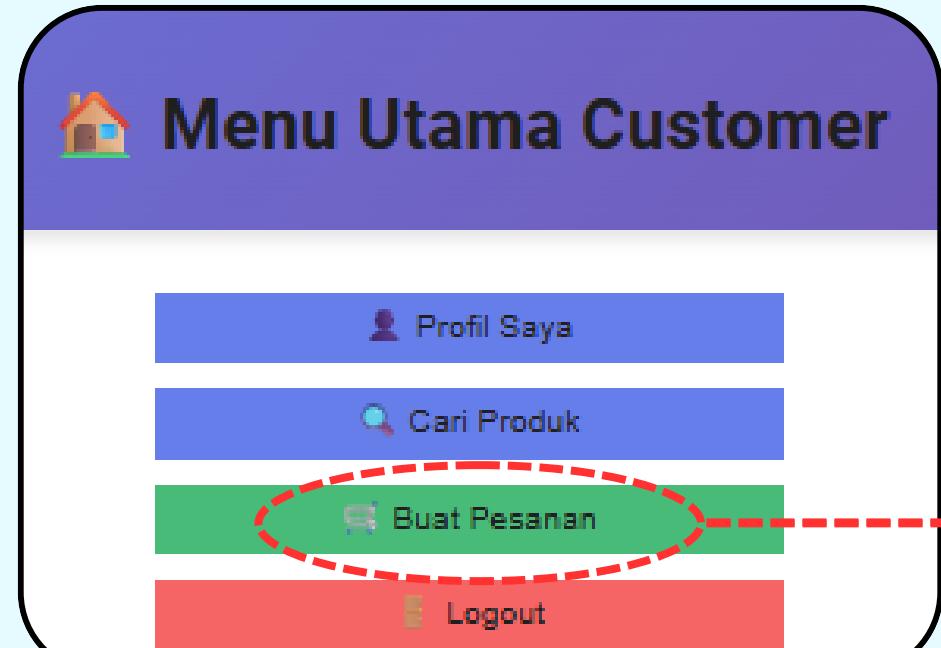
The diagram shows a table of product results. The columns are labeled: ID, Nama, Kategori, Harga, Deskripsi, Stok, and Rating. The data rows are as follows:

ID	Nama	Kategori	Harga	Deskripsi	Stok	Rating
PROD001	Smart LED Bulb	Electronics	Rp 120,000	An energy-efficient smart LED bulb that can be controlled remotely via mobile app.	50	4.5
PROD002	Portable Electric Kettle	Electronics	Rp 299,990	A compact and portable electric kettle ideal for travel and home use.	30	4.2
PROD003	Mini Air Purifier	Electronics	Rp 349,000	Mini air purifier to remove dust, allergens, and odors from small spaces.	20	4.0
PROD004	Wireless Extension Plug	Electronics	Rp 250,000	A wireless extension plug that supports multiple devices and remote control.	28	4.3
PROD005	Smart Light Switch	Electronics	Rp 190,000	Smart light switch that allows app and voice control for home lighting.	22	4.5
PROD006	USB Rechargeable Flashlight	Electronics	Rp 135,000	A compact and powerful USB rechargeable flashlight for emergency and outdoor use.	40	4.4
PROD007	WiFi Smart Plug	Electronics	Rp 175,000	Smart plug with WiFi connectivity to control your devices remotely.	33	4.3

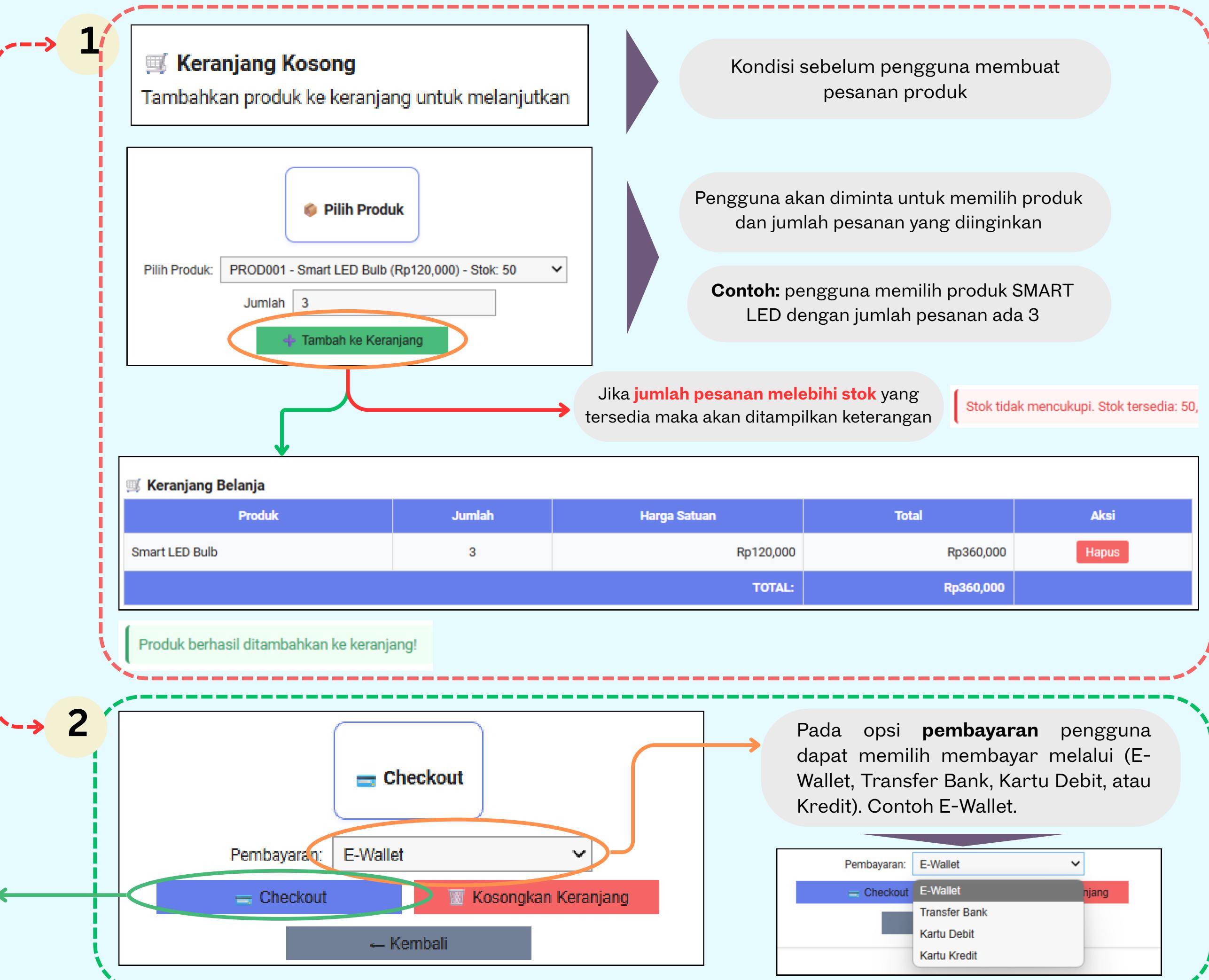
Setelah mencari produk yang diinginkan, akan ditampilkan produk yang tersedia dengan tampilan berupa **ID Produk, Nama Produk, Kategori Produk, Harga, Deksripsi Produk, Stok, dan Rating Produk**.

Hal tersebut tentu akan memudahkan pengguna untuk melihat produk yang diinginkan sebelum membuat pesanan.

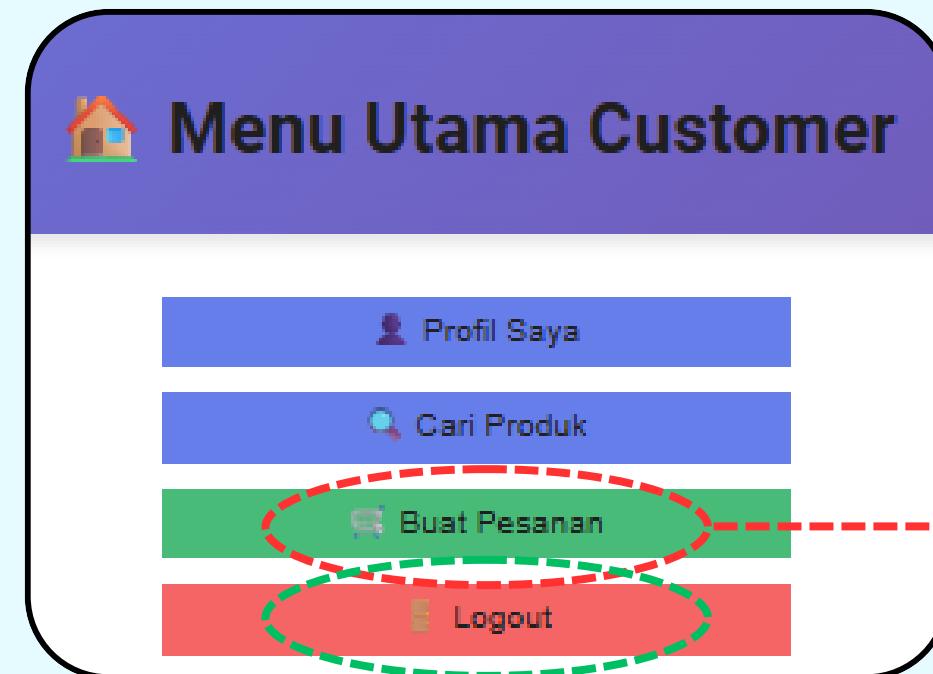
# Fitur Customer



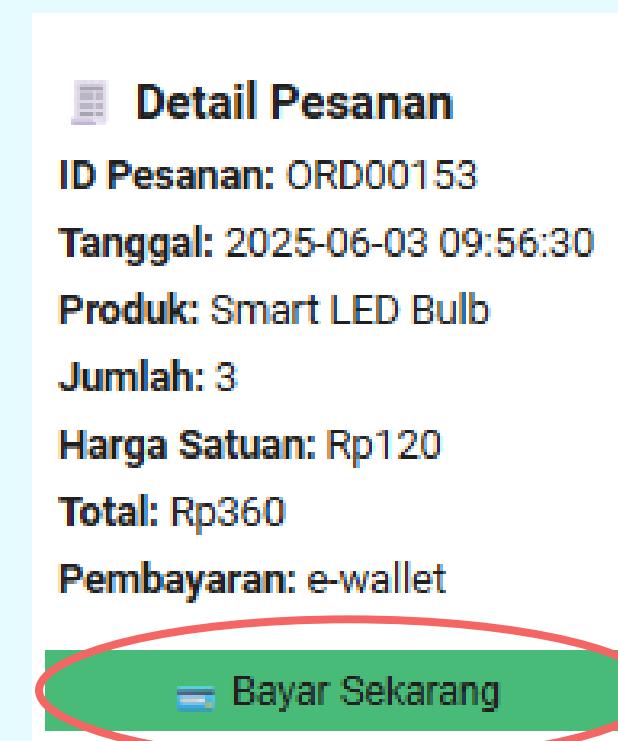
Fitur “**Buat Pesanan**” memungkinkan pengguna untuk membuat pesanan produk yang diinginkan setelah mencari produk.



# Fitur Customer

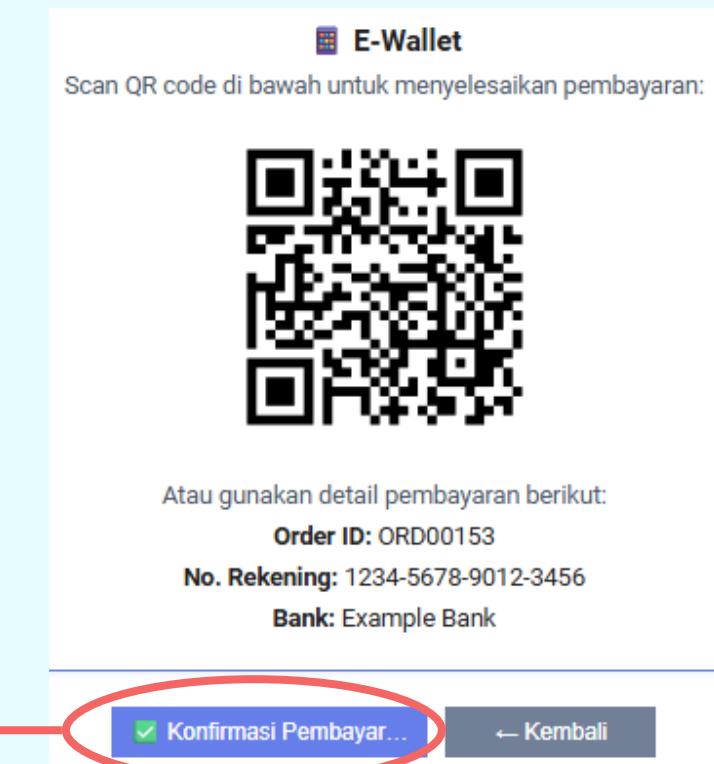


3



Setelah klik checkout pengguna akan diberikan detail pemesanan dan jika sudah benar maka pengguna dapat mengklik **bayar sekarang**

## Contoh tampilan E-Wallet



4

Terakhir pengguna dapat logout untuk kembali ke menu utama

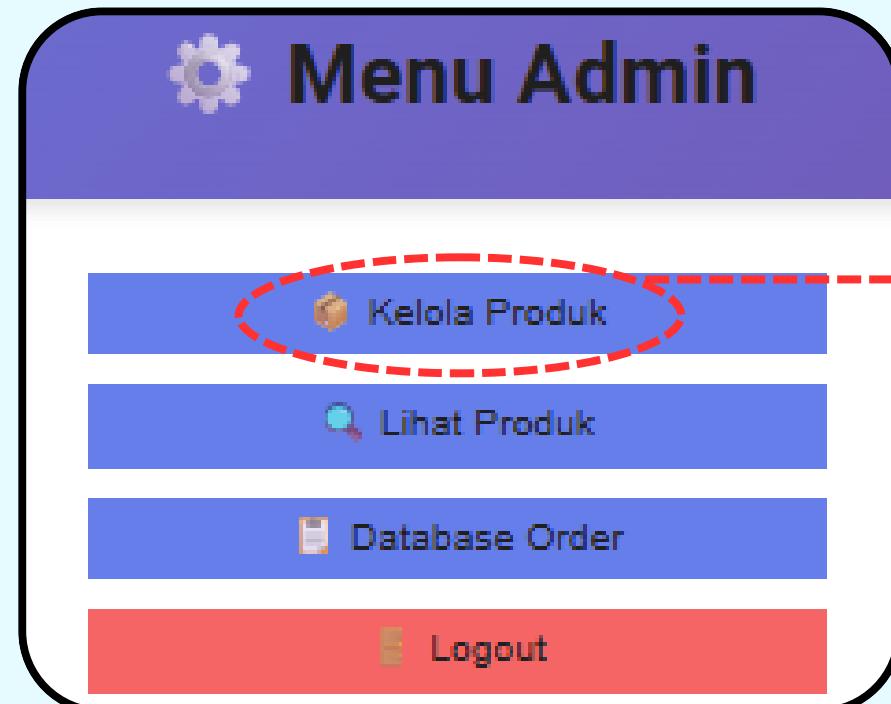
Setelah melakukan scan QR dan melakukan pembayaran, selanjutnya pengguna dapat mengklik **konfirmasi pembayaran**

Pembayaran berhasil!

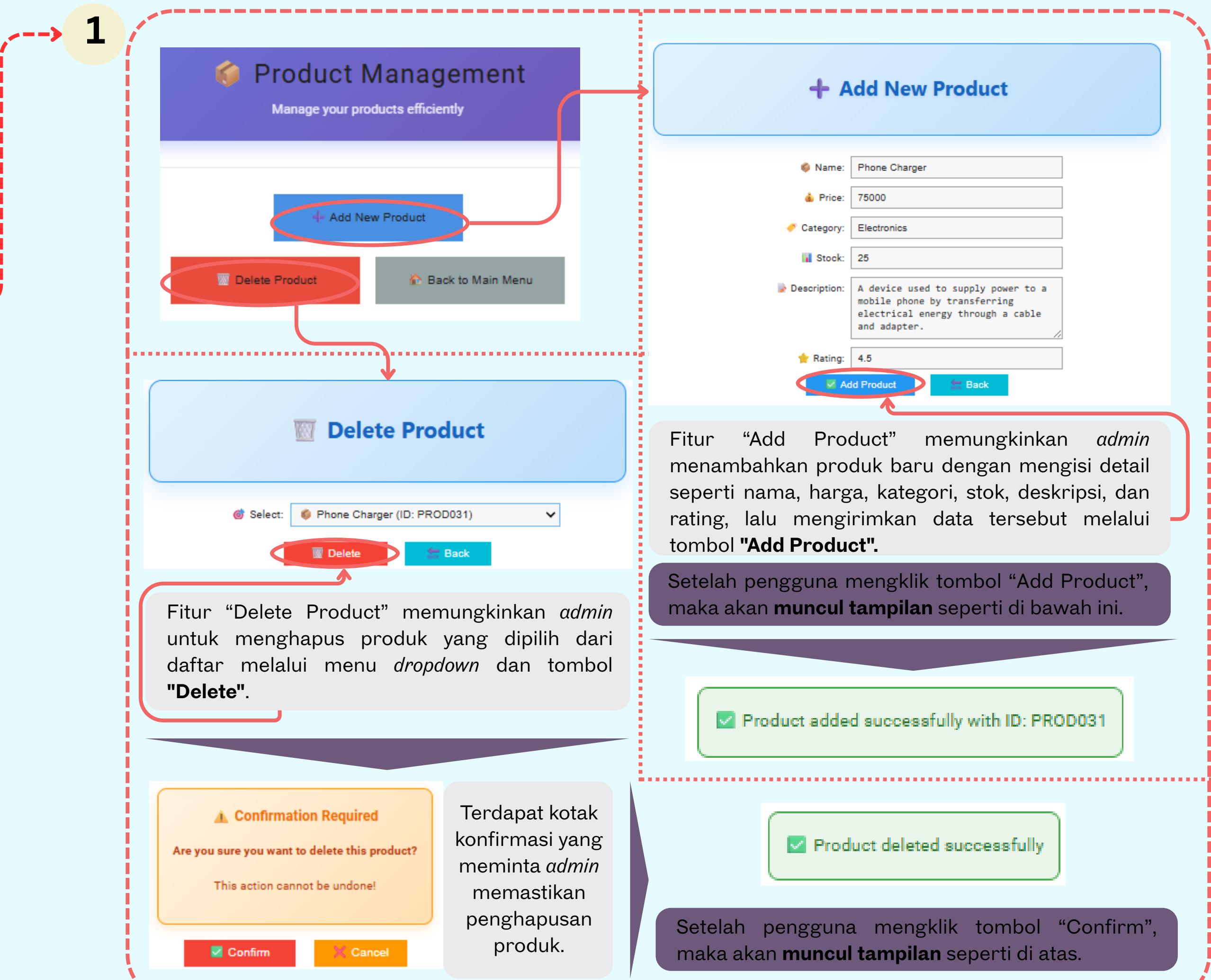
Order ORD00155				
Item dalam Order:				
Produk	Jumlah	Harga Satuan	Total	
Smart LED Bulb	3	Rp120,000	Rp360,000	

Pemesanan akan berhasil masuk ke fitur "Profil Saya" dengan status pembayaran sudah paid.

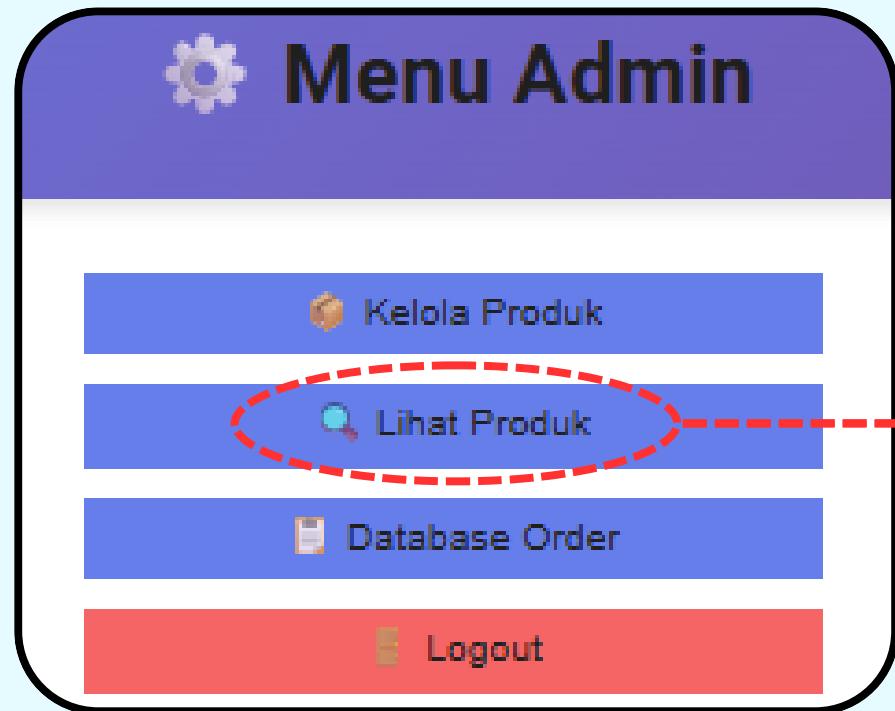
# Fitur Admin



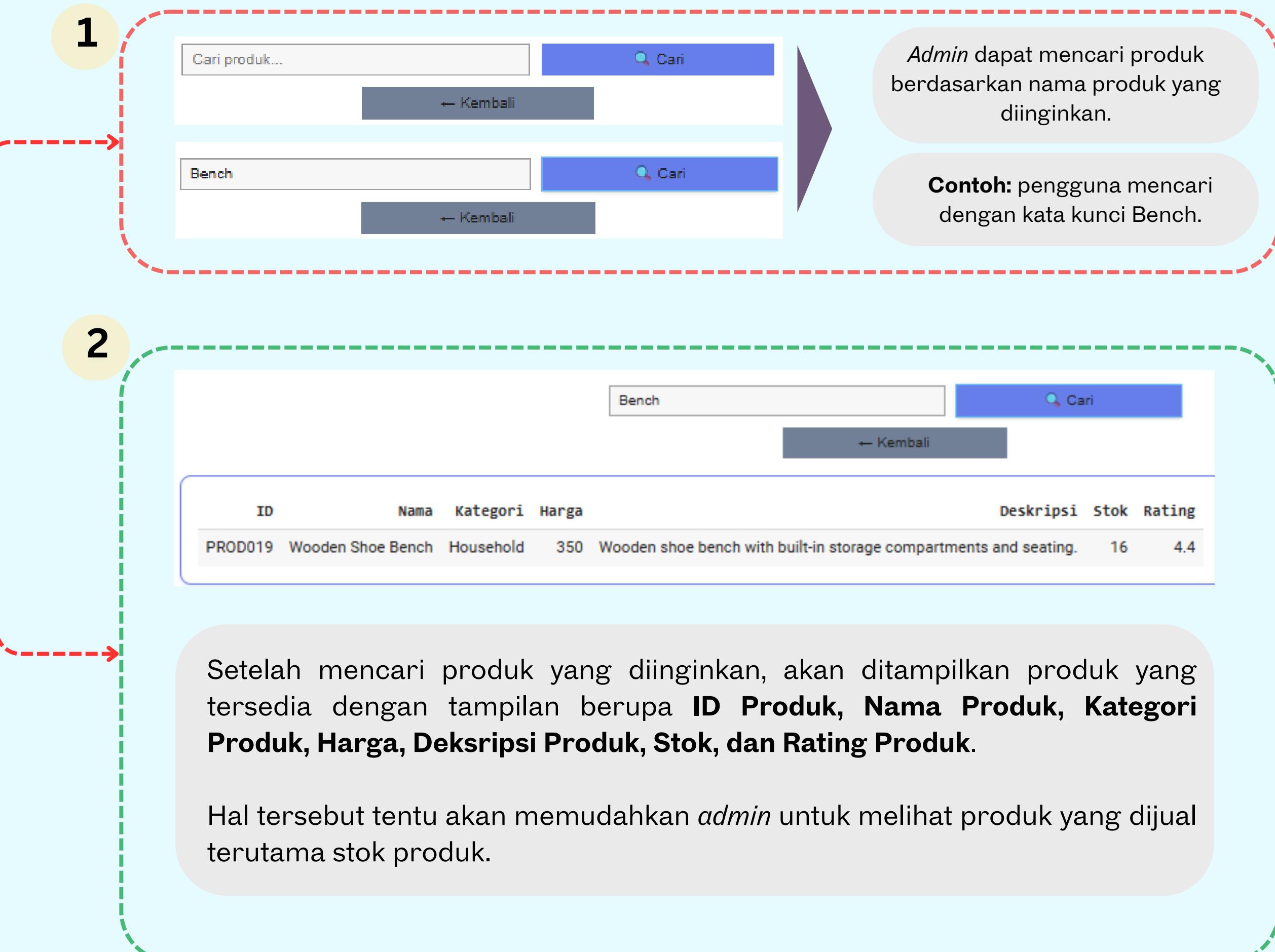
Fitur “Kelola Produk” memungkinkan *admin* untuk manajemen produk, seperti menambahkan produk dan menghapus produk.



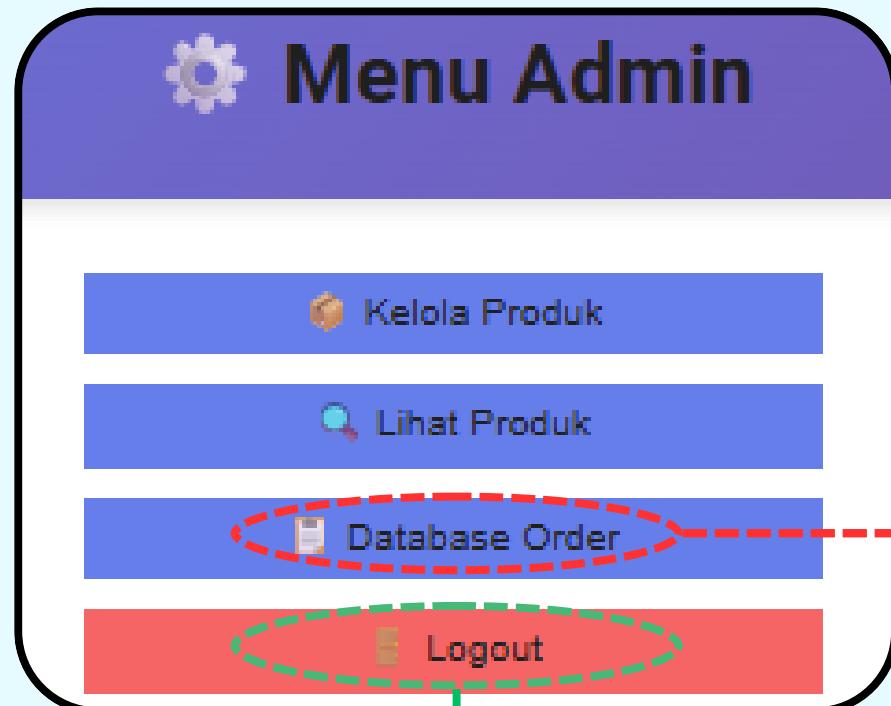
# Fitur Admin



Fitur “Lihat Produk” memungkinkan *admin* untuk mencari dan melihat produk-produk yang dijual.



# Fitur Admin



3  
Terakhir pengguna dapat *logout* untuk kembali ke menu utama

1  
Fitur “Database Order” memungkinkan *admin* untuk melihat serta mencari *order* yang diperoleh dari *customer* berdasarkan ID Order atau Nama Customer.

The image shows a purple header bar with the text "Database Order". Below it is a white content area with a search bar containing the placeholder "Cari berdasarkan ID Order atau Nama Customer...". To the right of the search bar is a teal "Cari" button.

2  
Misalkan *admin* ingin mencari ID Order “ORD00151”. Berikut merupakan tampilan “Database” Order setelah pencarian.

The image shows a search result for "Order ORD00151". The result includes:  
Customer: Carl Hall  
Tanggal: 2025-06-03 13:23:22  
Status Order: proceed  
Status Bayar: paid  
Total: Rp870,000  
Detail Items:  

ID Produk	Nama Produk	Jumlah	Harga Satuan	Total
PROD016	Multi-Layer Bookshelf	1	Rp320,000	Rp320,000
PROD017	Wall Mounted Shoe Rack	1	Rp280,000	Rp280,000
PROD006	USB Rechargeable Flashlight	2	Rp135,000	Rp270,000

[← Kembali](#)

Terdapat detail *order customer* yang berisi nama *customer*, tanggal pemesanan, status pengiriman, status pembayaran, dan total yang dibayar *customer*.

# Kesimpulan

- Hasil penggerjaan Projek Akhir Database menunjukkan bahwa sistem database penjualan dan pemesanan telah berhasil diimplementasikan menggunakan SQL dan GUI.
- Database “penjualan\_new.db” terdiri dari beberapa tabel relasional, yaitu Customer, Product, Order, Order\_Detail, Order\_Payment, dan Admin. Tabel Admin digunakan untuk keperluan kontrol dan manajemen, tetapi tidak memiliki relasi ke entitas lain.
- Implementasi SQL mendukung proses input data, pengubahan data, penghapusan data, serta pengambilan data melalui GUI yang menyediakan fitur login, registrasi, dan menu utama untuk admin serta customer.
- Pengujian dilakukan untuk memastikan semua fitur berjalan dengan baik dan data dapat dikelola dengan benar.
- Secara keseluruhan, database ini telah berhasil diimplementasikan untuk mendukung proses pencatatan, pengelolaan, dan pelaporan data penjualan dan pemesanan.

# Terima Kasih

**Proyek Akhir Mata Kuliah Database Semester Genap  
2024/2025**

Kelompok 4 Database (A)  
Dosen Pengampu : Devvi Sarwinda, M.Si.