

**LAPORAN Pengerjaan
Ujian Akhir Semester (UAS)
Basis Data Lanjut**

Dosen Pengampu : Dr. Raswa, M.Pd



Disusun oleh:

REVANA SANREIGO PUTRA (2303101)

**JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI INDRAMAYU
2024**

1. Buatlah sebuah sistem sederhana untuk mengelola data mahasiswa pada sebuah perguruan tinggi. Saudara diminta membuat satu tabel utama bernama mahasiswa, lalu melakukan operasi CRUD terhadap tabel tersebut menggunakan perintah SQL di Oracle Database, 5 mahasiswa dengan atribut bebas. Nama tabel: mahasiswa. Struktur Tabel:

Kolom	Tipe Data	Keterangan
Nim	VARCHAR2(10)	Primary Key
Nama	VARCHAR2(30)	Tidak Boleh Kosong
Jurusan	VARCHAR2(50)	Tidak Boleh Kosong
Angkatan	NUMBER(4)	Contoh: 2021
ipk	NUMBER(3,2)	Range: 0.00 – 4.00

Pengerjaan:

1. Langkah pertama yaitu membuka Oracle Database menggunakan SQLplus
2. Setelah itu diminta login untuk menginputkan username dan password nya
3. Buat table mahasiswa dengan ketentuan di soal seperti gambar di bawah ini.

```
SQL*Plus: Release 21.0.0.0.0 - Production on Tue May 27 20:00:51 2025
Version 21.3.0.0.0

Copyright (c) 1982, 2021, Oracle. All rights reserved.

Enter user-name: system
Enter password:
Last Successful login time: Tue May 20 2025 10:59:04 +07:00

Connected to:
Oracle Database 21c Enterprise Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL> CREATE TABLE mahasiswa (
  2   nim VARCHAR2(10) PRIMARY KEY,
  3   nama VARCHAR2(50) NOT NULL,
  4   jurusan VARCHAR2(30) NOT NULL,
  5   angkatan NUMBER(4),
  6   ipk NUMBER(3,2)
  7 );

Table created.
```

Penjelasan:

```
CREATE TABLE mahasiswa (
    nim VARCHAR2(10) PRIMARY KEY,
    nama VARCHAR2(50) NOT NULL,
    jurusan VARCHAR2(30) NOT NULL,
    angkatan NUMBER(4),
    ipk NUMBER(3,2)
);
```

- nim: Tipe data **VARCHAR2(10)** dan dijadikan **PRIMARY KEY** untuk memastikan nilai unik dan sebagai identitas utama mahasiswa.

- nama: VARCHAR2(50) dan diberi constraint **NOT NULL** yang berarti tidak boleh kosong.
- jurusan: VARCHAR2(30) juga diberi constraint **NOT NULL**.
- angkatan: NUMBER(4) untuk mencatat tahun angkatan.
- ipk: NUMBER(3,2) untuk menyimpan nilai IPK, dengan format misalnya 3.75.

4. Untuk perintah *Create* dari operasi CRUD seperti gambar di bawah ini.

```
SQL> INSERT INTO mahasiswa VALUES ('2303101', 'Revana Sanreigo Putra', 'Teknik Informatika', 2023, 3.92);
1 row created.

SQL> INSERT INTO mahasiswa VALUES ('2303094', 'Irgi Nazwa Mustofa', 'Teknik Informatika', 2023, 3.88);
1 row created.

SQL> INSERT INTO mahasiswa VALUES ('2303090', 'Dwi Rizki Fatur Rahman', 'Teknik Informatika', 2023, 3.65);
1 row created.

SQL> INSERT INTO mahasiswa VALUES ('2303089', 'Dinar Setiawan Firdaus', 'Teknik Informatika', 2023, 3.65);
1 row created.

SQL> INSERT INTO mahasiswa VALUES ('2303086', 'Anissa Fatimatus Zahra', 'Teknik Informatika', 2023, 4.00);
1 row created.
```

Penjelasan:

```
INSERT INTO mahasiswa VALUES (...);
```

Sebanyak 5 mahasiswa dimasukkan ke dalam tabel mahasiswa, sesuai dengan permintaan soal.

5. Untuk perintah *Read* dari operasi CRUD seperti gambar di bawah ini.

```
SQL> SELECT * FROM mahasiswa;
```

NIM	NAMA	ANGKATAN	IPK
2303101	Revana Sanreigo Putra	2023	3.92
2303094	Irgi Nazwa Mustofa	2023	3.88
2303090	Dwi Rizki Fatur Rahman	2023	3.65
2303089	Dinar Setiawan Firdaus	2023	3.65
2303086	Anissa Fatimatus Zahra	2023	4

Penjelasan:

```
SELECT * FROM mahasiswa;
```

Perintah diatas untuk membaca seluruh data yang ada dalam tabel mahasiswa, yang merupakan bagian dari operasi R (Read) dalam sistem CRUD.

```
SQL> SELECT * FROM mahasiswa WHERE nama LIKE '%Putra%';
```

NIM	NAMA	JURUSAN	ANGKATAN	IPK
2303101	Revana Sanreigo Putra	Teknik Informatika	2023	3.92

Penjelasan:

```
SELECT * FROM mahasiswa WHERE nama LIKE '%Putra%';
```

Perintah diatas untuk mencari data *mahasiswa* berdasarkan nama yang mengandung kata "**Putra**",

```
SQL> SELECT nama, ipk FROM mahasiswa WHERE nama LIKE '%e%';
```

NAMA	IPK
Revana Sanreigo Putra	3.92
Dinar Setiawan Firdaus	3.65

Penjelasan:

```
SELECT nama, ipk FROM mahasiswa WHERE nama LIKE '%e%';
```

Perintah tersebut adalah bagian dari operasi **READ** dalam sistem CRUD, yang digunakan untuk:

- Menampilkan hanya kolom nama dan ipk dari tabel mahasiswa.
- Memfilter data berdasarkan kolom nama yang mengandung "e".

6. Untuk perintah *Update* dari operasi CRUD seperti gambar di bawah ini.

```
SQL> UPDATE mahasiswa SET ipk = 4.00 WHERE nim = '2303101';
```

1 row updated.

```
SQL> SELECT * FROM mahasiswa WHERE nama LIKE '%Putra%';
```

NIM	NAMA	JURUSAN	ANGKATAN	IPK
2303101	Revana Sanreigo Putra	Teknik Informatika	2023	4

Penjelasan:

```
UPDATE mahasiswa SET ipk = 4.00 WHERE nim = '2303101';
```

Perintah ini digunakan untuk mengubah nilai IPK mahasiswa dengan NIM 2303101 menjadi 4.00. Artinya, data mahasiswa atas nama **Revana Sanreigo Putra** diperbarui nilai IPK-nya.

7. Untuk perintah *Delete* dari operasi CRUD seperti gambar di bawah ini.

```
SQL> DELETE FROM mahasiswa WHERE nim = '2303086';
```

```
1 row deleted.
```

```
SQL> SELECT * FROM mahasiswa;
```

NIM	NAMA	JURUSAN	ANGKATAN	IPK
2303101	Revana Sanreigo Putra	Teknik Informatika	2023	4
2303094	Irgi Nazwa Mustofa	Teknik Informatika	2023	3.88
2303090	Dwi Rizki Faturrohman	Teknik Informatika	2023	3.65

NIM	NAMA	JURUSAN	ANGKATAN	IPK
2303089	Dinar Setiawan Firdaus	Teknik Informatika	2023	3.65

Penjelasan:

```
DELETE FROM mahasiswa WHERE nim = '2308086';
```

- Menghapus satu baris data mahasiswa dengan NIM 2308086.
- Mahasiswa yang dihapus adalah: Anissa Fatimatuz Zahra.

```
SELECT * FROM mahasiswa;
```

Menampilkan seluruh data yang tersisa di tabel mahasiswa setelah penghapusan.

3. Studi kasus melakukan operasi CRUD dan perintah SQL menggunakan dua tabel: CUSTOMERS dan ORDERS, dengan hubungan satu ke banyak (one-to-many), di mana satu customer dapat memiliki banyak order.

- a. Buatlah dua tabel berikut:

Tabel CUSTOMERS

- customer_id (Primary Key)
- name
- email

Tabel ORDERS

- order_id (Primary Key)
- order_date
- amount
- customer_id (Foreign Key ke tabel CUSTOMERS)

b. Kemudian lakukan operasi CRUD berikut (data bebas):

- 1) Tambahkan 2 data customer dan masing-masing memiliki 2 order?
- 2) Tampilkan semua order beserta nama customernya?
- 3) Ubah email dari salah satu customer?
- 4) Hapus salah satu customer beserta semua order-nya?
- 5) Buat koneksi PHP ke database Oracle?
- 6) Tampilkan semua order beserta nama customernya?
- 7) Tambahkan satu customer dan satu order terkait?
- 8) Update email salah satu customer?
- 9) Hapus salah satu customer dan semua order-nya?

Pengerjaan:

a. Pembuatan table CUSTOMERS dan ORDERS:

```
SQL> CREATE TABLE CUSTOMERS (  
2   customer_id VARCHAR2(10) PRIMARY KEY,  
3   name VARCHAR2(50),  
4   email VARCHAR2(100)  
5 );  
  
Table created.  
  
SQL> CREATE TABLE ORDERS (  
2   order_id VARCHAR2(10) PRIMARY KEY,  
3   order_date DATE,  
4   amount NUMBER(10,2),  
5   customer_id VARCHAR2(10),  
6   CONSTRAINT fk_customer  
7   FOREIGN KEY (customer_id)  
8   REFERENCES CUSTOMERS(customer_id)  
9   ON DELETE CASCADE  
10 );  
  
Table created.
```

Penjelasan:

```
CREATE TABLE CUSTOMERS (  
    customer_id VARCHAR2(10) PRIMARY KEY,  
    name VARCHAR2(100),  
    email VARCHAR2(100)  
);
```

- **customer_id**: Merupakan *Primary Key* atau kunci utama yang bersifat unik dan menjadi identitas setiap pelanggan.
- **name**: Menyimpan nama pelanggan.
- **email**: Menyimpan alamat email pelanggan.

```

CREATE TABLE ORDERS (

    order_id VARCHAR2(10) PRIMARY KEY,

    order_date DATE,

    amount NUMBER(8,2),

    customer_id VARCHAR2(10),

    CONSTRAINT fk_customer

        FOREIGN KEY (customer_id)

        REFERENCES CUSTOMERS(customer_id)

        ON DELETE CASCADE

);

```

- **order_id**: Merupakan *Primary Key* untuk setiap pesanan.
- **order_date**: Menyimpan tanggal pesanan dilakukan.
- **amount**: Menyimpan jumlah total pesanan dalam format angka desimal (maksimal 8 digit, dengan 2 digit di belakang koma).
- **customer_id**: Merupakan *Foreign Key* yang merujuk pada *customer_id* di tabel *CUSTOMERS*.

Relasi One-to-Many:

- Satu pelanggan (*CUSTOMERS*) dapat memiliki banyak pesanan (*ORDERS*), tapi satu pesanan hanya dimiliki oleh satu pelanggan.
- Penggunaan **FOREIGN KEY** memastikan keterkaitan data antar tabel.
- **ON DELETE CASCADE**: Jika data pelanggan dihapus, maka semua pesanan terkait di tabel *ORDERS* juga otomatis dihapus.

b. Kemudian operasi CRUD:

- 1) Menambahkan 2 data customer yang masing-masing memiliki 2 orders

```

SQL> INSERT INTO CUSTOMERS VALUES ('C001', 'Revana Sanreigo Putra', 'revanasanreigo@mail.com');
1 row created.

SQL> INSERT INTO CUSTOMERS VALUES ('C002', 'Andre Revana', 'andrerevana@mail.com');
1 row created.

SQL> INSERT INTO ORDERS VALUES ('0001', TO_DATE('2025-05-01', 'YYYY-MM-DD'), 100000, 'C001');
1 row created.

SQL> INSERT INTO ORDERS VALUES ('0002', TO_DATE('2025-05-02', 'YYYY-MM-DD'), 150000, 'C001');
1 row created.

SQL> INSERT INTO ORDERS VALUES ('0003', TO_DATE('2025-05-03', 'YYYY-MM-DD'), 200000, 'C002');
1 row created.

SQL> INSERT INTO ORDERS VALUES ('0004', TO_DATE('2025-05-04', 'YYYY-MM-DD'), 120000, 'C002');
1 row created.

```

Penjelasan:

```

INSERT INTO CUSTOMERS VALUES ('C001', 'Revana Sanreigo Putra',
'revanasanreigo@mail.com');
INSERT INTO CUSTOMERS VALUES ('C002', 'Andre Revana',
'andrerevana@mail.com');

```

Menambahkan 2 data customer ke tabel CUSTOMERS

```

INSERT INTO ORDERS VALUES ('0001', TO_DATE('2025-05-01',
'YYYY-MM-DD'), 100000, 'C001');
INSERT INTO ORDERS VALUES ('0002', TO_DATE('2025-05-02',
'YYYY-MM-DD'), 150000, 'C001');
INSERT INTO ORDERS VALUES ('0003', TO_DATE('2025-05-03',
'YYYY-MM-DD'), 200000, 'C002');
INSERT INTO ORDERS VALUES ('0004', TO_DATE('2025-05-04',
'YYYY-MM-DD'), 120000, 'C002');

```

- Memberikan masing-masing customer 2 data order ke tabel ORDERS.
- Format tanggal diubah menggunakan fungsi TO_DATE agar sesuai dengan format Oracle.
- Hubungan one-to-many (1 pelanggan – banyak pesanan).

2) Menampilkan semua order beserta nama customernya


```
SQL> SELECT
2   o.order_id,
3   o.order_date,
4   o.amount,
5   c.name AS customer_name
6 FROM ORDERS o
7 JOIN CUSTOMERS c ON o.customer_id = c.customer_id;
```

ORDER_ID	ORDER_DAT	AMOUNT	CUSTOMER_NAME
O001	01-MAY-25	100000	Revana Sanreigo Putra
O002	02-MAY-25	150000	Revana Sanreigo Putra
O003	03-MAY-25	200000	Andre Revana
O004	04-MAY-25	120000	Andre Revana

Penjelasan:

```
SELECT
    o.order_id,
    o.order_date,
    o.amount,
    c.name AS customer_name
FROM ORDERS o
JOIN CUSTOMERS c ON o.customer_id = c.customer_id;
```

- **SELECT:** Menentukan kolom-kolom yang akan ditampilkan: *order_id*, *order_date*, *amount*, dan *customer_name*.
- **FROM ORDERS o:** Mengambil data dari tabel *ORDERS* dengan alias *o*.
- **JOIN CUSTOMERS c:** Melakukan *INNER JOIN* dengan tabel *CUSTOMERS* dengan alias *c*.
- **ON o.customer_id = c.customer_id:** Syarat penggabungan berdasarkan *customer_id* yang menjadi *foreign key* di tabel *ORDERS*.

3) Mengubah email customer

```
SQL> UPDATE CUSTOMERS SET email = 'dregot@mail.com' WHERE customer_id = 'C001';
1 row updated.
SQL> select * from customers;
```

CUSTOMER_I	NAME	EMAIL
C001	Revana Sanreigo Putra	dregot@mail.com
C002	Andre Revana	andrerevana@mail.com

Penjelasan:

```
UPDATE CUSTOMERS SET email = 'dregot@mail.com' WHERE
customer_id = 'C001';
```

- **UPDATE CUSTOMERS:** Menunjukkan bahwa data yang akan diubah berada di tabel *CUSTOMERS*.
- **SET email = 'revanasanreigo@mail.com':** Mengatur kolom email menjadi alamat email baru yaitu dregot@mail.com.
- **WHERE customer_id = 'C001':** Menentukan baris data yang akan diubah, yaitu customer dengan customer_id = 'C001'.

4) Menghapus salah satu customer dan seluruh ordernya

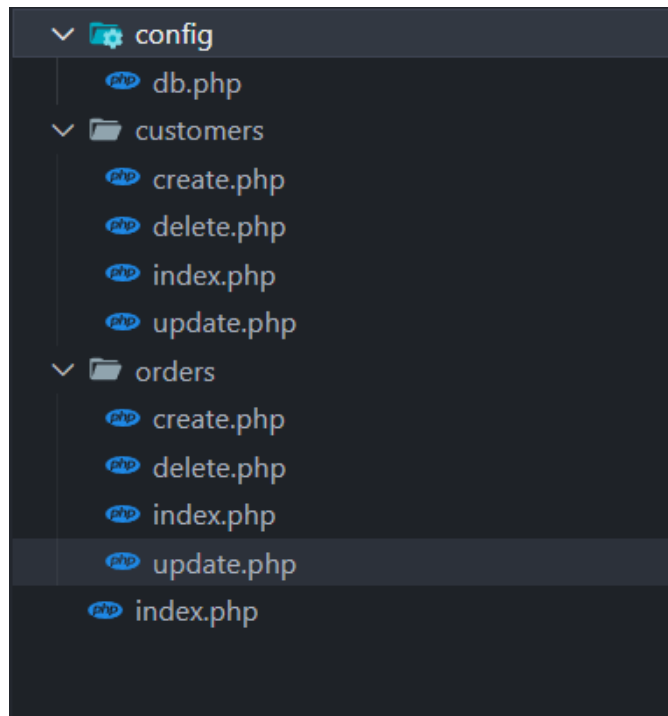
```
SQL> DELETE FROM CUSTOMERS where customer_id = 'C002';
1 row deleted.
SQL> select * from customers;
CUSTOMER_I NAME
EMAIL
-----
C001      Revana Sanreigo Putra
dregot@mail.com
```

Penjelasan:

- Customer dengan customer_id = 'C002' (Andre Revana) dihapus.
- Karena ada relasi **ON DELETE CASCADE**, maka semua order milik Andre Revana juga ikut terhapus.

5) Membuat koneksi php ke database Oracle

- Buat file di dalam *C:\xampp\htdocs* karena saya menggunakan xampp
- Berikut struktur file phpnya



c. Isi folder config di file *db.php*

```
config> db.php > a.btn
1 <?php
2 $conn = oci_connect('system', '112N4ZMu$', '//localhost/orcl');
3 if (!$conn) {
4     $e = oci_error();
5     die("Koneksi gagal: " . $e['message']);
6 }
7 ?>
```

Kode di atas yaitu untuk **menghubungkan PHP ke Oracle Database**. Jika koneksi gagal, maka akan muncul pesan error yang menjelaskan penyebab kegagalan.

d. Isi folder customers di file *create.php*

```
customers > create.php > a.btn
1 <?php include '../config/db.php';
2 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
3     $id = $_POST['customer_id'];
4     $name = $_POST['name'];
5     $email = $_POST['email'];
6     $sql = "INSERT INTO CUSTOMERS VALUES (:id, :name, :email)";
7     $stid = oci_parse($conn, $sql);
8     oci_bind_by_name($stid, ':id', $id);
9     oci_bind_by_name($stid, ':name', $name);
10    oci_bind_by_name($stid, ':email', $email);
11    oci_execute($stid);
12    header('Location: index.php');
13 }
14 ?>
15 <h2>Tambah Customer</h2>
16 <form method="POST">
17     ID: <input type="text" name="customer_id"><br>
18     Nama: <input type="text" name="name"><br>
19     Email: <input type="email" name="email"><br>
20     <button type="submit">Simpan</button>
21 </form>
22 <a href="index.php" class="btn">Kembali</a>
```

Kode ini adalah **fitur tambah customer**:

- Terkoneksi ke database Oracle.
- Menyimpan data dari form menggunakan query **INSERT**.
- Melakukan redirect ke halaman utama setelah data ditambahkan.

e. Isi folder customers di file *delete.php*

```
customers > delete.php > ...
1
2 <?php
3 include '../config/db.php';
4
5 if (!isset($_GET['id']) || !is_numeric($_GET['id'])) {
6     die("ID tidak valid.");
7 }
8
9 $id = $_GET['id'];
10
11 $sql = "DELETE FROM CUSTOMERS WHERE customer_id = :id";
12 $stmt = oci_parse($conn, $sql);
13 oci_bind_by_name($stmt, ':id', $id);
14
15 if (!oci_execute($stmt)) {
16     $e = oci_error($stmt);
17     echo "Gagal menghapus data: " . htmlentities($e['message']);
18     exit;
19 }
20
21 header('Location: index.php');
22 exit;
23 ?>
24
```

f. Isi folder customers di file *update.php*

```
customers > update.php > ...
1 <?php include '../templates/header.php'; include '../config/db.php';
2 $id = $_GET['id'];
3 $query = "SELECT * FROM CUSTOMERS WHERE customer_id = :id";
4 $stmt = oci_parse($conn, $query);
5 oci_bind_by_name($stmt, ':id', $id);
6 oci_execute($stmt);
7 $row = oci_fetch_assoc($stmt);
8 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
9     $name = $_POST['name'];
10    $email = $_POST['email'];
11    $update = "UPDATE CUSTOMERS SET name = :name, email = :email WHERE customer_id = :id";
12    $stmt = oci_parse($conn, $update);
13    oci_bind_by_name($stmt, ':name', $name);
14    oci_bind_by_name($stmt, ':email', $email);
15    oci_bind_by_name($stmt, ':id', $id);
16    oci_execute($stmt);
17    header("Location: index.php");
18 }
19 ?>
20 <h2>Edit Customer</h2>
21 <form method="POST">
22     Nama: <input type="text" name="name" value="= $row['NAME'] ?"><br>
23     Email: <input type="email" name="email" value="= $row['EMAIL'] ?"><br>
24     <button type="submit">Update</button>
25 </form>
26 <a href="index.php" class="btn">Kembali</a>
27 <?php include '../templates/footer.php'; ?>
```

g. Isi folder customers di file *index.php*

```

customers > index.php > ...
1  <?php include '../templates/header.php'; include '../config/db.php'; ?>
2  <h2>Data Customers</h2>
3  <a class="btn" href="create.php">+ Tambah Customer</a>
4
5  <table>
6  <tr><th>ID</th><th>Nama</th><th>Email</th><th>Aksi</th></tr>
7  <?php
8  $query = "SELECT * FROM CUSTOMERS";
9  $stid = oci_parse($conn, $query);
10 oci_execute($stid);
11 while ($row = oci_fetch_assoc($stid)) {
12     echo "<tr>
13         <td>{$row['CUSTOMER_ID']}</td>
14         <td>{$row['NAME']}</td>
15         <td>{$row['EMAIL']}</td>
16         <td>
17             <a class='btn btn-warning' href='update.php?id={$row['CUSTOMER_ID']}'>Edit</a>
18             <a class='btn btn-danger' href='delete.php?id={$row['CUSTOMER_ID']}'>Hapus</a>
19         </td>
20     </tr>";
21 }
22 oci_free_statement($stid);
23 oci_close($conn);
24 ?>
25 </table>
26 <a class="btn" href="..orders/index.php">Kelola Orders</a>
27 <?php include '../templates/footer.php'; ?>

```

- h. Untuk di folder orders sama saja seperti di folder customers hanya perlu penyesuaian karena ada beberapa yang berbeda
- i. Isi file index.php

```

1  <?php include 'templates/header.php'; ?>
2  <h2>Selamat Datang</h2>
3  <p>Silakan pilih menu:</p>
4  <a class="btn" href="customers/index.php">Kelola Customers</a>
5  <a class="btn" href="orders/index.php">Kelola Orders</a>
6  <?php include 'templates/footer.php'; ?>

```

- 6) Menampilkan semua order beserta nama customernya

Data Orders

[+ Tambah Order](#)

ID	Tanggal	Jumlah	Customer	Aksi
2	07-MAY-25	3	REVANA SANREIGO PUTRA	Edit Hapus
1	31-MAY-25	89	Irgi	Edit Hapus

[Kelola Customers](#)

Gambar diatas merupakan tampilan **halaman web** yang menampilkan **data semua order** beserta **nama customer** yang melakukan order.

- 7) Menambahkan satu customer dan satu order terkait
- Data tabel customers awal

Data Customers

[+ Tambah Customer](#)

ID	Nama	Email	Aksi
1	REVANA SANREIGO PUTRA	revanasanreigo1@gmail.com	Edit Hapus
2	Irgi	irgi@gmail.com	Edit Hapus

[Kelola Orders](#)

Proses menambahkan Customers

Tambah Customer

ID:

Nama:

Email:

[Kembali](#)

Data table customers yang telah di tambah

Data Customers

[+ Tambah Customer](#)


ID	Nama	Email	Aksi
1	REVANA SANREIGO PUTRA	revanasanreigo1@gmail.com	Edit Hapus
2	Irgi	irgi@gmail.com	Edit Hapus
3	andre	andre@gmail.com	Edit Hapus

[Kelola Orders](#)

Proses menambahkan orders dengan cutomers *andre* yang baru saja di tambahkan

Tambah Order

ID Order:

Tanggal Order: 

Jumlah:

Customer:

[Kembali](#)

Table orders yang sudah di tambah satu order terkait

Data Orders

[+ Tambah Order](#)

ID	Tanggal	Jumlah	Customer	Aksi
4	09-MAY-25	6	andre	Edit Hapus
2	07-MAY-25	3	REVANA SANREIGO PUTRA	Edit Hapus

[Kelola Customers](#)

8) Update email salah satu customer

Email di data table customers dengan id 3 adalah andre@gmail.com

Data Customers

[+ Tambah Customer](#)

ID	Nama	Email	Aksi
1	REVANA SANREIGO PUTRA	dregot@mail.com	Edit Hapus
3	andre	andre@gmail.com	Edit Hapus

[Kelola Orders](#)

Saya akan ubah menjadi revan@gmail.com

Edit Customer

Nama:

Email:

[Kembali](#)

Isi data table yang sudah di update

Data Customers

[+ Tambah Customer](#)

ID	Nama	Email	Aksi
1	REVANA SANREIGO PUTRA	dregot@mail.com	Edit Hapus
3	andre	revan@gmail.com	Edit Hapus

[Kelola Orders](#)

9) Menghapus salah satu customer dan semua order-nya

Data customers **sebelum di hapus**

Data Customers

[+ Tambah Customer](#)

ID	Nama	Email	Aksi
1	REVANA SANREIGO PUTRA	dregot@mail.com	Edit Hapus
3	andre	revan@gmail.com	Edit Hapus

[Kelola Orders](#)

Data customers atas nama *andre* **telah di hapus**

Data Customers

[+ Tambah Customer](#)

ID	Nama	Email	Aksi
1	REVANA SANREIGO PUTRA	dregot@mail.com	Edit Hapus

[Kelola Orders](#)

Data orders sebelum di hapus

Data Orders

[+ Tambah Order](#)

ID	Tanggal	Jumlah	Customer	Aksi
4	09-MAY-25	6	andre	Edit Hapus
2	07-MAY-25	3	REVANA SANREIGO PUTRA	Edit Hapus

[Kelola Customers](#)

Data orders atas nama *andre* telah di hapus

Data Orders

[+ Tambah Order](#)

ID	Tanggal	Jumlah	Customer	Aksi
2	07-MAY-25	3	REVANA SANREIGO PUTRA	Edit Hapus

[Kelola Customers](#)

Demikian laporan ini saya buat berdasarkan pemahaman saya terhadap materi yang telah dipelajari.