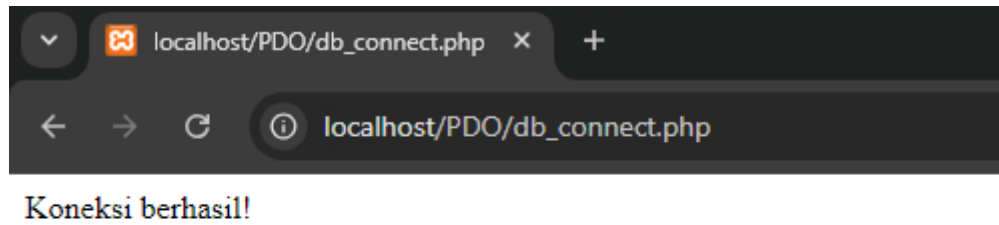


Nama : Vanzha Feby Daniar

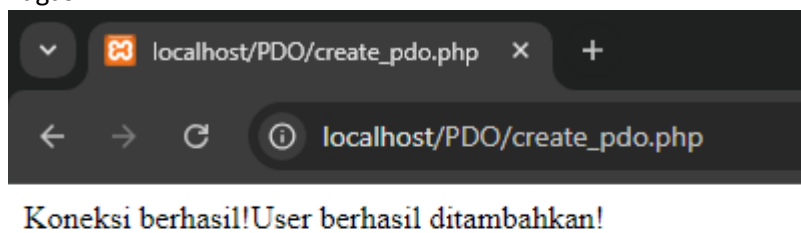
Kelas Web A

Tugas HandOut

1. Tugas 1



2. Tugas 2



Hasil data yang telah terupdate

Showing rows 0 - 0 (1 total, Query took 0.0003 seconds.)

```
SELECT * FROM users;
```

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain SQL](#) ] [ [Create PHP code](#) ] [ [Refresh](#) ]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

	id	username	email	created_at
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	1	user123	user123@example.com	2024-08-02 17:35:23

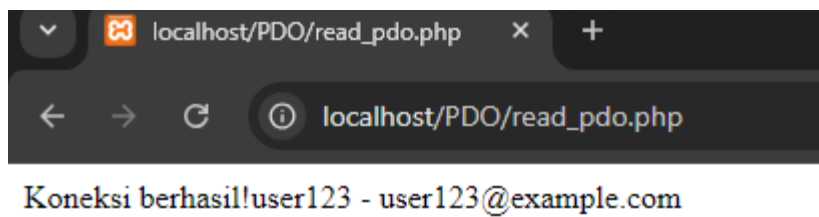
[Check all](#) | With selected: [Edit](#) [Copy](#) [Delete](#) [Export](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table

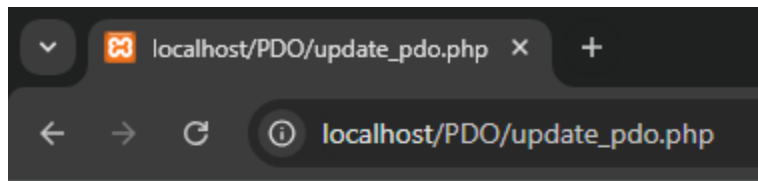
Query results operations

[Print](#) [Copy to clipboard](#) [Export](#) [Display chart](#) [Create view](#)

### 3. Tugas 3



#### 4. Tugas 4



Koneksi berhasil!Email user berhasil diperbarui!

Database management interface showing a successful query execution.

Showing rows 0 - 0 (1 total, Query took 0.0003 seconds.)

`SELECT * FROM `users``

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 | Filter rows: Search this table

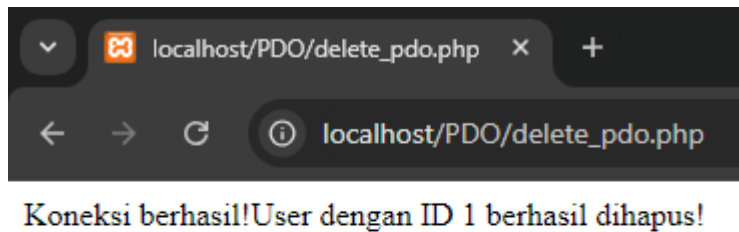
Extra options

	id	username	email	created_at
<input type="checkbox"/> Edit Copy Delete	1	user123	new_email@example.com	2024-08-02 17:35:23

Check all | With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table

## 5. Tugas 5



## 6. Tugas 6

Setelah memodifikasi insert\_user.php, kesimpulannya adalah:

- **Simplicity:** Kode kini hanya menyimpan username dan email, mengurangi kompleksitas.
- **Keamanan:** Password dihilangkan, mengurangi risiko terkait penyimpanan password.
- **Validasi:** Validasi input masih ada untuk memastikan data yang dimasukkan valid.
- **Kinerja:** Query SQL lebih sederhana dan efisien tanpa kolom password.

### PERTANYAAN

#### 1. Perbedaan Penggunaan MySQLi dan PDO

**MySQLi (MySQL Improved):**

- **Spesifik untuk MySQL:** Hanya mendukung database MySQL.
- **Dukungan Fitur:** Menyediakan fitur-fitur seperti prepared statements, transaksi, dan prosedur tersimpan.
- **API:** Tersedia dalam dua bentuk: Object-oriented dan Procedural.
- **Metode:**
  - `mysqli_connect()`, `mysqli_query()`, `mysqli_prepare()`, dll.
  - Object-oriented: `$mysqli->query()`, `$stmt->execute()`, dll.

**PDO (PHP Data Objects):**

- **Multi-Database:** Mendukung berbagai jenis database (MySQL, PostgreSQL, SQLite, dll.).
- **Dukungan Fitur:** Juga mendukung prepared statements, transaksi, dan prosedur tersimpan.
- **API:** Hanya tersedia dalam bentuk Object-oriented.
- **Metode:**
  - `new PDO()`, `$pdo->prepare()`, `$stmt->execute()`, dll.

#### 2. Pilihan Antara MySQLi dan PDO

Saya akan memilih PDO. Berikut alasan-alasannya:

##### 1. Dukungan Multi-Database:

- PDO mendukung berbagai jenis database, memungkinkan fleksibilitas jika Anda ingin berpindah atau menggunakan beberapa jenis database dalam aplikasi.
2. **Keamanan dan Fitur Prepared Statements:**
    - Kedua PDO dan MySQLi mendukung prepared statements, tetapi PDO memiliki API yang lebih konsisten dan lebih bersih dalam penanganan prepared statements.
  3. **Object-Oriented API:**
    - PDO menggunakan hanya pendekatan object-oriented yang lebih modern dan terstruktur, sedangkan MySQLi menyediakan opsi prosedural yang bisa membuat kode kurang konsisten.
  4. **Transaksi dan Error Handling:**
    - PDO menyediakan penanganan kesalahan yang lebih baik dengan mode exception handling (PDO::ERRMODE\_EXCEPTION), membuatnya lebih mudah untuk menangani dan mendebug kesalahan.
  5. **Konsistensi API:**
    - PDO memiliki API yang lebih konsisten dan bersih, sementara MySQLi dapat menggunakan gaya prosedural yang mungkin tidak seragam dengan gaya kode berbasis objek.