**UJI KOMPETENSI KEAHLIAN DOKUMENTASI MEMBUAT APLIKASI**

**TODO LIST**

**TAHUN PELAJARAN 2024/2025**



Disusun:

Nama : Faizah Nahdah

Kelas : Xll PRL 2

NIS/NISN : 192070077/0072934982

KOMPETENSI KEAHLIAN

**REKAYASA PERANGKAT LUNAK DAN GIM**

# SMKN 6 KOTA TANGERANG SELATAN

1. **Cara Kerja**

Aplikasi ini adalah sistem **Todo List** berbasis PHP dengan database MySQL. Pengguna dapat menambahkan, memperbarui, dan menghapus tugas. Setiap tugas memiliki prioritas (**High, Medium, Low**) serta tanggal jatuh tempo.

1. **Menghubungkan ke Database**

* Menggunakan **PDO** untuk koneksi database.
* Jika koneksi gagal, aplikasi akan menampilkan pesan error.

1. **CRUD Todo List**
   * **Create**: Menambahkan tugas baru dengan task, due\_date, dan status (High, Medium,Low).
   * **Read**: Menampilkan semua tugas dari database.
   * **Update**: Mengedit tugas yang sudah ada.
   * **Delete**: Menghapus tugas dari database.
2. **Tampilan dan Interaksi**
   * + Menggunakan HTML dan CSS untuk tampilan.
     + JavaScript digunakan untuk interaksi UI seperti modal edit dan tombol aksi.
     + Checkbox untuk menandai tugas yang sudah selesai.

## **Fitur Aplikasi**

1. **Menambah tugas** dengan tambah tugas, tanggal jatuh tempo dan prioritas.
2. **Mengedit tugas** yang sudah ada.
3. **Menghapus tugas** dengan konfirmasi.
4. **Menampilkan daftar tugas** dalam tabel yang terurut berdasarkan tanggal dibuat.
5. **Tombol aksi** untuk mengedit atau menghapus tugas.
6. **Modal edit** untuk memperbarui tugas tanpa memuat ulang halaman.
7. **Tampilan responsif** menggunakan CSS dan JavaScript.
8. **Fungsi**

### **Koneksi Database**

try {

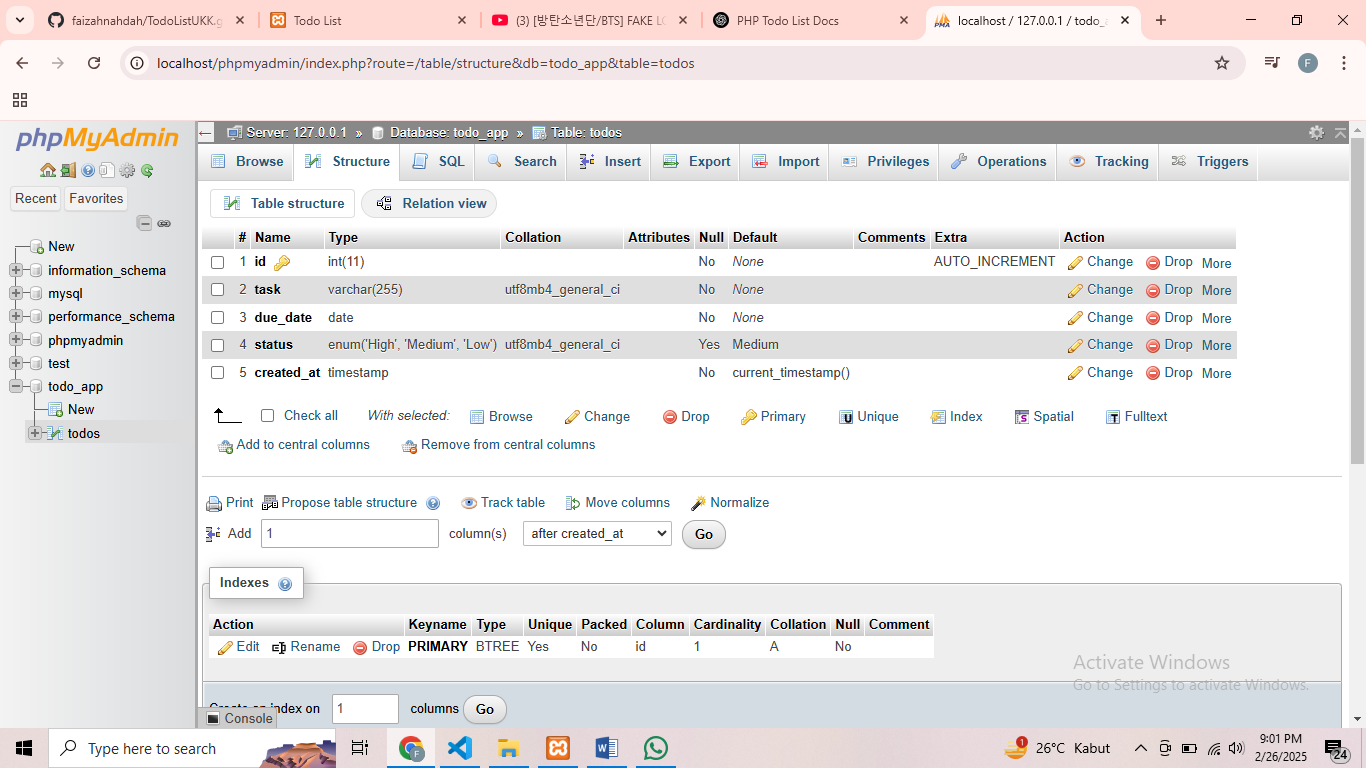
$pdo = new PDO("mysql:host=$host;dbname=$db", $user, $pass);

$pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

} catch (PDOException $e) {

die("Koneksi gagal: " . $e->getMessage());

}



* Menghubungkan PHP ke MySQL menggunakan **PDO**.
* Jika gagal, akan menampilkan pesan error.

### **Menambah Tugas Baru**

if (isset($\_POST['task']) && isset($\_POST['due\_date']) && !isset($\_POST['update\_id'])) {

$task = htmlspecialchars($\_POST['task']);

$due\_date = $\_POST['due\_date'];

$status = $\_POST['status'];

$sql = "INSERT INTO todos (task, due\_date, status) VALUES (:task, :due\_date, :status)";

$stmt = $pdo->prepare($sql);

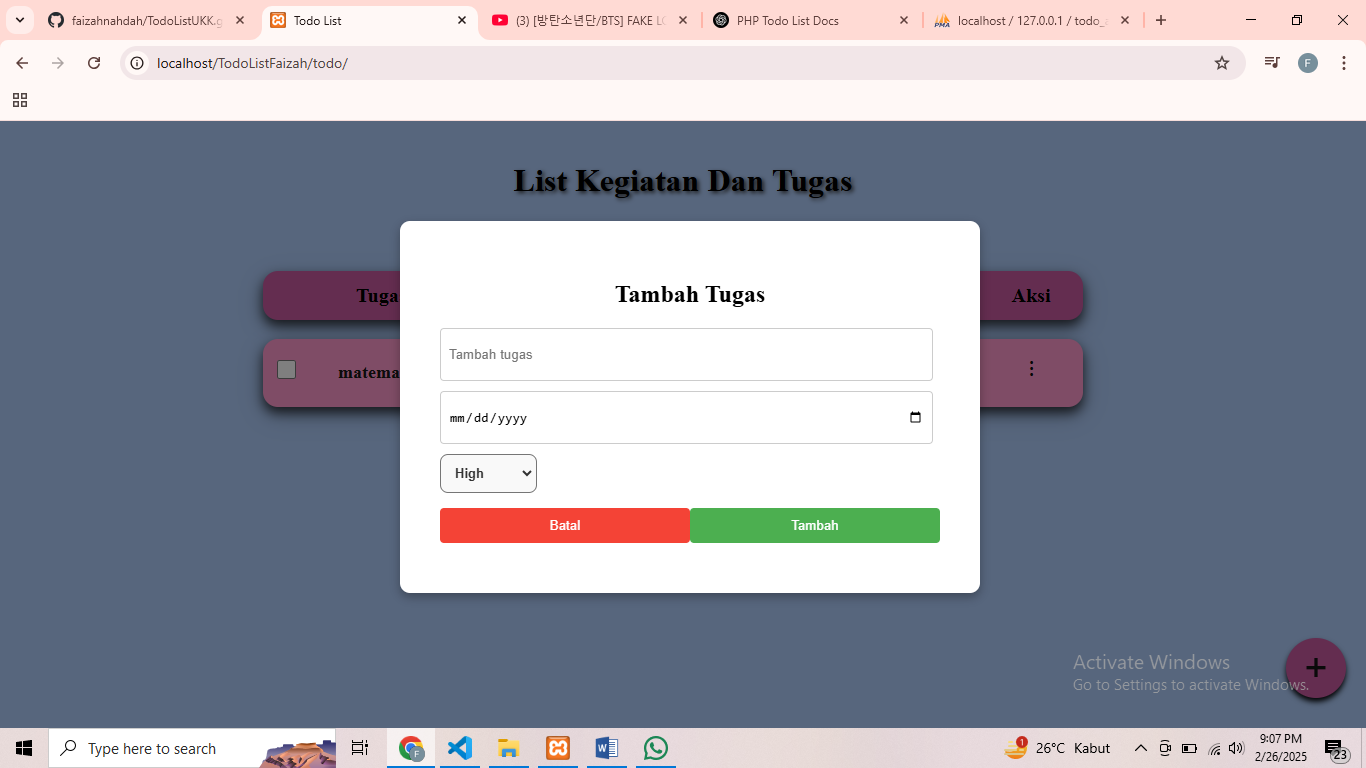
$stmt->bindParam(':task', $task);

$stmt->bindParam(':due\_date', $due\_date);

$stmt->bindParam(':status', $status);

$stmt->execute();

}



* Menerima data dari form.
* Menggunakan htmlspecialchars() untuk mencegah **XSS**.
* Menyimpan data ke database menggunakan **prepared statement**.

### **Memperbarui Tugas**

if (isset($\_POST['update\_id']) && isset($\_POST['task']) && isset($\_POST['due\_date'])) {

$id = $\_POST['update\_id'];

$task = htmlspecialchars($\_POST['task']);

$due\_date = $\_POST['due\_date'];

$status = $\_POST['status'];

$sql = "UPDATE todos SET task = :task, due\_date = :due\_date, status = :status WHERE id = :id";

$stmt = $pdo->prepare($sql);

$stmt->bindParam(':id', $id);

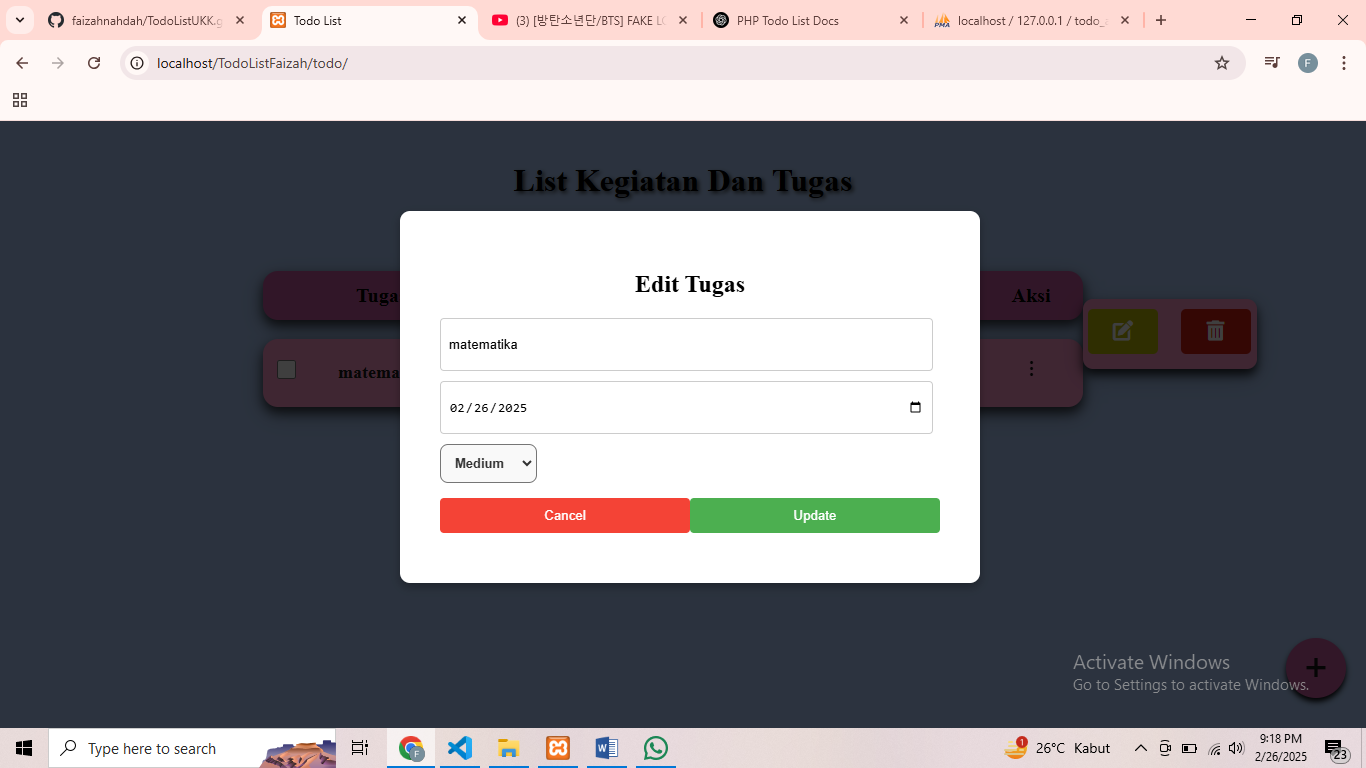
$stmt->bindParam(':task', $task);

$stmt->bindParam(':due\_date', $due\_date);

$stmt->bindParam(':status', $status);

$stmt->execute();

}



* Menerima ID tugas yang ingin diperbarui.
* Memperbarui data tugas di database.

### **Menghapus Tugas**

if (isset($\_POST['delete\_id'])) {

$id = $\_POST['delete\_id'];

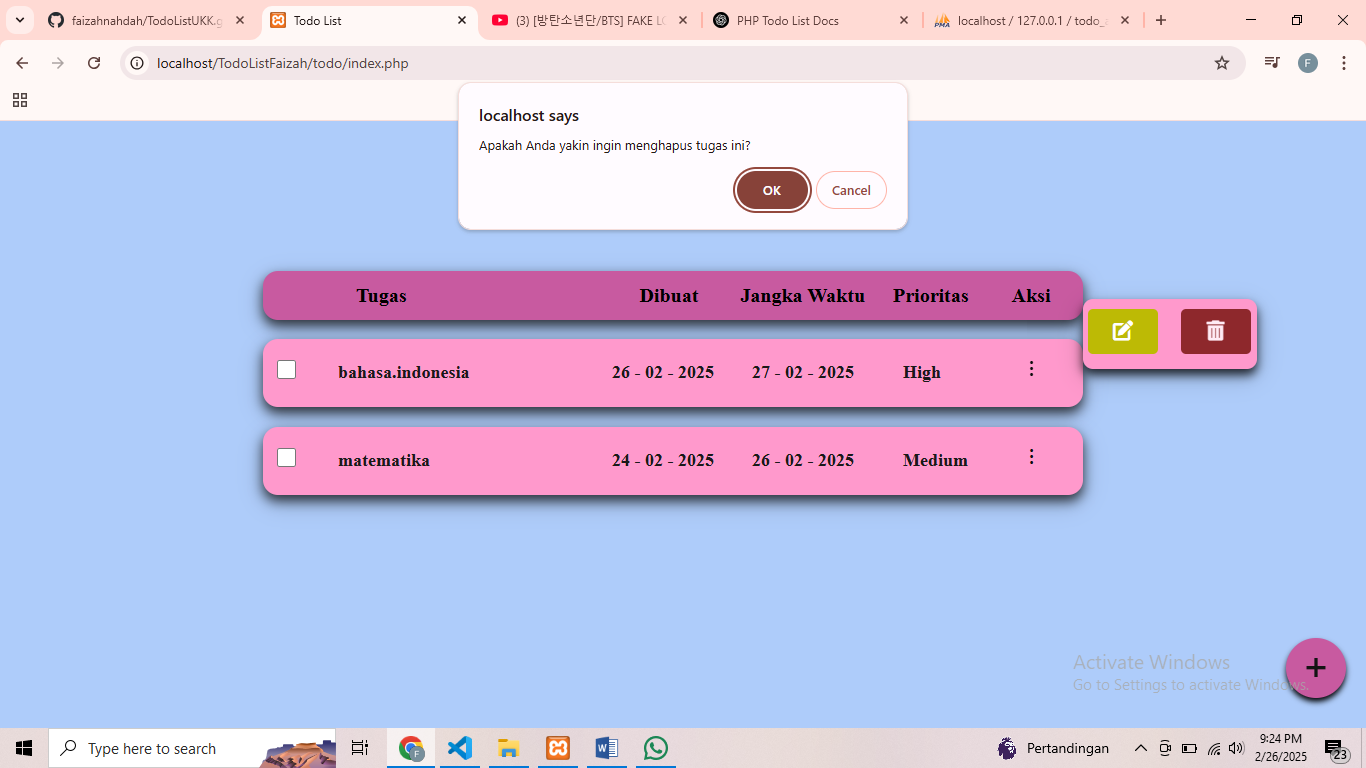
$sql = "DELETE FROM todos WHERE id = :id";

$stmt = $pdo->prepare($sql);

$stmt->bindParam(':id', $id);

$stmt->execute();

}



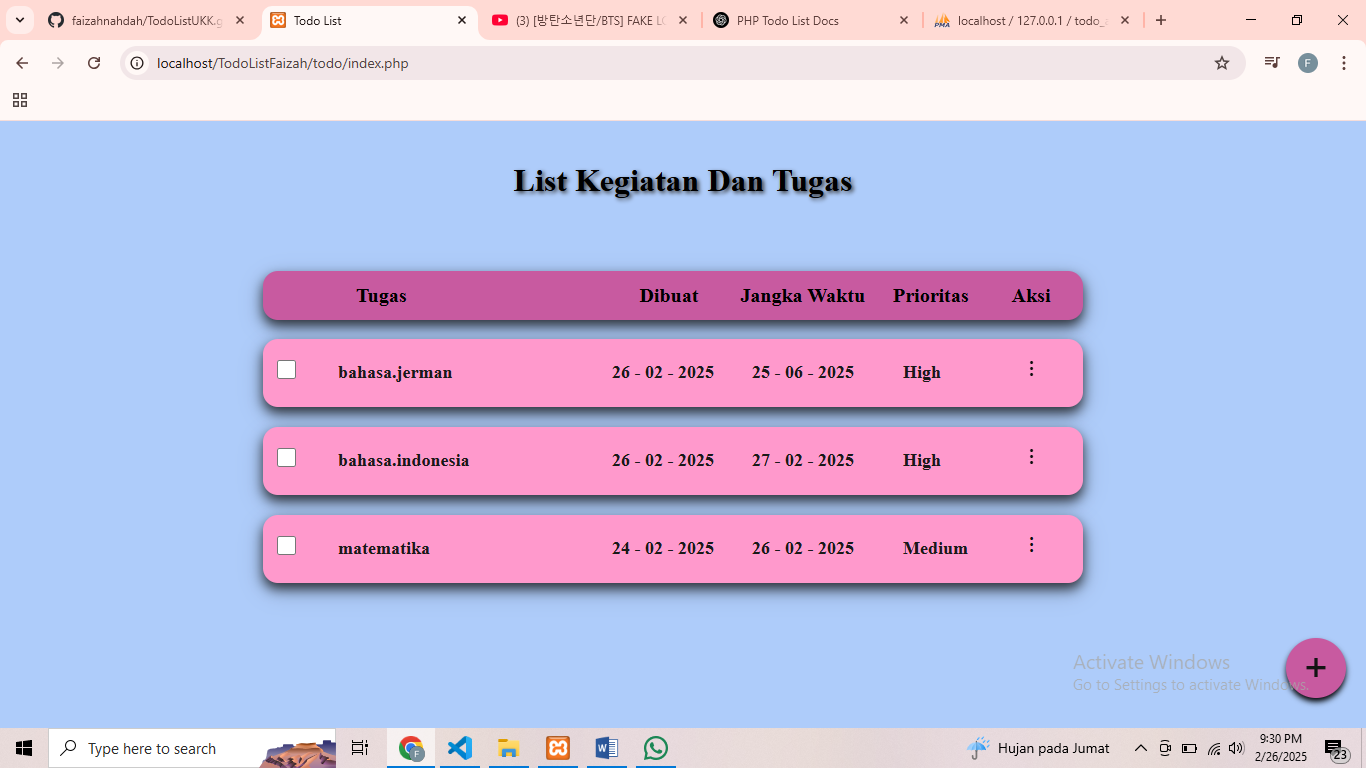
* Menghapus tugas berdasarkan ID yang dikirim dari form.

### **Menampilkan Data dalam Tabel**

$stmt = $pdo->prepare("SELECT \* FROM todos ORDER BY created\_at DESC");

$stmt->execute();

$todos = $stmt->fetchAll();



* Mengambil semua tugas dari database dan menampilkannya dalam tabel.