

LAPORAN AKHIR PRAKTIKUM

Mata Praktikum : Pemrograman Web
Kelas : 3IA11
Praktikum ke- : 4
Tanggal : 22/11/2024
Materi : PHP DAN MYSQL
NPM : 51422161
Nama : MUHAMMAD TARMIDZI BARIQ
Ketua Asisten : Fadhil Muhammad
Jumlah Lembar : 12



LABORATORIUM TEKNIK INFORMATIKA

UNIVERSITAS GUNADARMA

2024

Jelaskan code pada ACT

```
1 <?php
2 function getConnection()
3 {
4     $host = "localhost"; //local server
5     $db_name = "pert4_51422161_db"; //nama database
6     $username = "root"; //username
7     $password = ""; //password
8
9     $conn = new mysqli($host, $username, $password, $db_name);
10
11     // jika koneksi gagal
12     if ($conn->connect_error) {
13         die("Connection failed: " . $conn->connect_error);
14     }
15
16     return $conn;
17 }
```

db.php

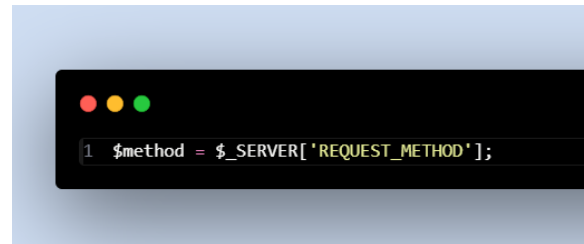
melakukan koneksi ke database pert4__51422161_db

```
1 <?php
2
3 header('Access-Control-Allow-Origin: *');
4 header('Content-Type: application/json');
5 header('Access-Control-Allow-Methods: POST, PUT, DELETE, GET');
6 header('Access-Control-Allow-Headers: Content-Type');
7
```

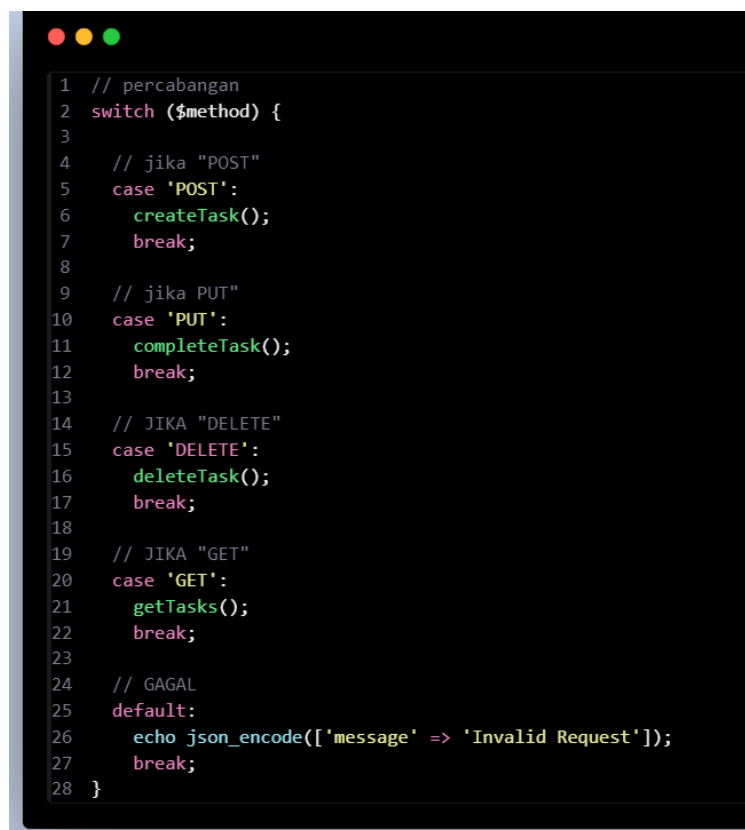
1. Mengizinkan semua domain (*) untuk mengakses sumber daya pada server
2. konten yang dikembalikan oleh server adalah dalam format JSON.
3. Metode HTTP yang diizinkan untuk mengakses sumber daya. Dalam hal ini, metode POST, PUT, DELETE, dan GET diizinkan.
4. Untuk dikirim oleh klien saat membuat permintaan ke server.

```
include_once 'db.php';
```

Import file db.php



Kode ini mendapatkan metode HTTP yang digunakan dalam permintaan saat ini (seperti GET, POST, PUT, DELETE) dari variabel global \$_SERVER.



digunakan untuk menangani berbagai jenis permintaan HTTP (HTTP request methods) dan melakukan tindakan yang sesuai berdasarkan jenis permintaan tersebut.

```

1 function createTask()
2 {
3     $data = json_decode(file_get_contents("php://input"));
4     if (!empty($data->task)) {
5         $conn = getConnection();
6         $stmt = $conn->prepare("INSERT INTO todos (task) VALUES (?)");
7         $stmt->bind_param('s', $data->task);
8
9         if ($stmt->execute()) {
10             echo json_encode(['message' => 'Task Created']);
11         } else {
12             echo json_encode(['message' => 'Task Not Created']);
13         }
14
15         $stmt->close();
16         $conn->close();
17     } else {
18         echo json_encode(['message' => 'Incomplete Data']);
19     }
20 }

```

Melakukan insert tabel

```
$data = json_decode(file_get_contents("php://input"));
```

Mengambil data masukan yang dikirim melalui permintaan HTTP dan mendekodnya dari format JSON menjadi objek PHP.

```
if (!empty($data->task)) {
```

Kondisi ini memeriksa apakah data tugas yang diterima tidak kosong

```
$conn = getConnection();
```

fungsi yang mengembalikan objek koneksi ke basis data.

```
$stmt = $conn->prepare("INSERT INTO todos (task) VALUES (?)");
```

```
$stmt->bind_param('s', $data->task);
```

untuk memasukkan data tugas ke dalam tabel todos.

```
if ($stmt->execute()) {  
  
    echo json_encode(['message' => 'Task Created']);  
  
} else {  
  
    echo json_encode(['message' => 'Task Not Created']);  
  
}
```

Jika eksekusi berhasil, pesan JSON 'Task Created' dikirimkan sebagai respons. Jika tidak, pesan 'Task Not Created' dikirimkan.

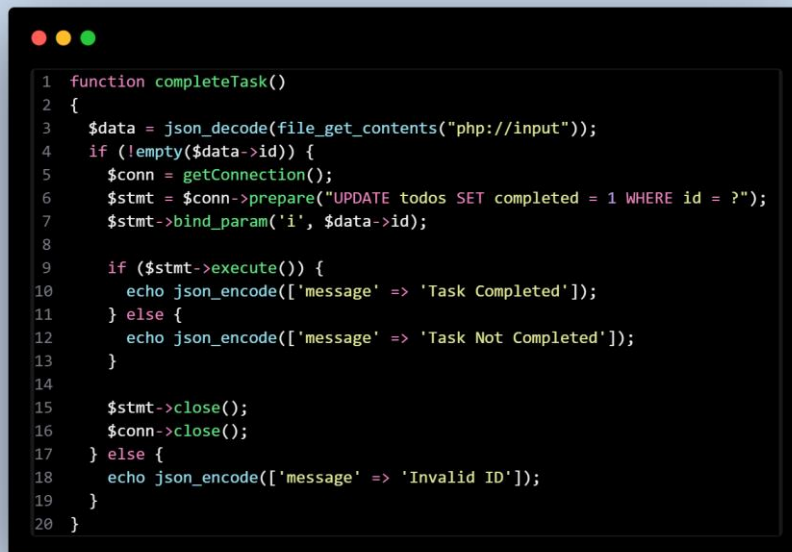
```
$stmt->close();
```

```
$conn->close();
```

Setelah pernyataan dieksekusi, pernyataan dan koneksi ke basis data ditutup untuk membebaskan sumber daya.

```
} else {  
  
    echo json_encode(['message' => 'Incomplete Data']);  
  
}
```

Jika data tugas yang diterima kosong, pesan JSON 'Incomplete Data' dikirimkan sebagai respons.



```

1 function completeTask()
2 {
3     $data = json_decode(file_get_contents("php://input"));
4     if (!empty($data->id)) {
5         $conn = getConnection();
6         $stmt = $conn->prepare("UPDATE todos SET completed = 1 WHERE id = ?");
7         $stmt->bind_param('i', $data->id);
8
9         if ($stmt->execute()) {
10             echo json_encode(['message' => 'Task Completed']);
11         } else {
12             echo json_encode(['message' => 'Task Not Completed']);
13         }
14
15         $stmt->close();
16         $conn->close();
17     } else {
18         echo json_encode(['message' => 'Invalid ID']);
19     }
20 }

```

`$data = json_decode(file_get_contents("php://input"));`

Baris ini mengambil data masukan yang dikirim melalui permintaan HTTP dan mendekodenya dari format JSON menjadi objek PHP.

`if (!empty($data->id)) {`

Kondisi ini memeriksa apakah data ID yang diterima tidak kosong. Jika data ID kosong, proses akan langsung mengirimkan pesan bahwa ID tidak valid.

`$conn = getConnection();`

Fungsi `getConnection()` dipanggil untuk membuat koneksi ke basis data. Diasumsikan bahwa `getConnection()` adalah fungsi yang mengembalikan objek koneksi ke basis data.

`$stmt = $conn->prepare("UPDATE todos SET completed = 1 WHERE id = ?");`

`$stmt->bind_param('i', $data->id);`

Baris ini menyiapkan pernyataan SQL UPDATE untuk memperbarui status tugas menjadi selesai (completed = 1) di tabel todos. Metode bind_param() digunakan untuk mengikat nilai data ID (\$data->id) ke pernyataan SQL.

```
if ($stmt->execute()) {  
  
    echo json_encode(['message' => 'Task Completed']);  
  
} else {  
  
    echo json_encode(['message' => 'Task Not Completed']);  
  
}
```

Pernyataan SQL dieksekusi menggunakan metode execute(). Jika eksekusi berhasil, pesan JSON 'Task Completed' dikirimkan sebagai respons. Jika tidak, pesan 'Task Not Completed' dikirimkan.

```
$stmt->close();  
  
$conn->close();
```

Setelah pernyataan dieksekusi, pernyataan dan koneksi ke basis data ditutup untuk membebaskan sumber daya.

```
} else {  
  
    echo json_encode(['message' => 'Invalid ID']);  
  
}
```

Jika data ID yang diterima kosong, pesan JSON 'Invalid ID' dikirimkan sebagai respons.

```

1 function deleteTask()
2 {
3     $data = json_decode(file_get_contents("php://input"));
4     if (!empty($data->id)) {
5         $conn = getConnection();
6         $stmt = $conn->prepare("DELETE FROM todos WHERE id = ?");
7         $stmt->bind_param('i', $data->id);
8
9         if ($stmt->execute()) {
10             echo json_encode(['message' => 'Task Deleted']);
11         } else {
12             echo json_encode(['message' => 'Task Not Deleted']);
13         }
14
15         $stmt->close();
16         $conn->close();
17     } else {
18         echo json_encode(['message' => 'Invalid ID']);
19     }
20 }

```

Fungsi deleteTask

```
$data = json_decode(file_get_contents("php://input"));
```

Baris ini mengambil data masukan yang dikirim melalui permintaan HTTP dan mendekodnya dari format JSON menjadi objek PHP.

```
if (!empty($data->id)) {
```

Kondisi ini memeriksa apakah data ID yang diterima tidak kosong. Jika data ID kosong, proses akan langsung mengirimkan pesan bahwa ID tidak valid.

```
$conn = getConnection();
```

Fungsi getConnection() dipanggil untuk membuat koneksi ke basis data. Diasumsikan bahwa getConnection() adalah fungsi yang mengembalikan objek koneksi ke basis data.

```
$stmt = $conn->prepare("DELETE FROM todos WHERE id = ?");
```

```
$stmt->bind_param('i', $data->id);
```


Baris ini menyiapkan pernyataan SQL DELETE untuk menghapus data tugas dari tabel todos berdasarkan ID. Metode bind_param() digunakan untuk mengikat nilai data ID (\$data->id) ke pernyataan SQL.

```
if ($stmt->execute()) {  
  
    echo json_encode(['message' => 'Task Deleted']);  
  
} else {  
  
    echo json_encode(['message' => 'Task Not Deleted']);  
  
}
```

Pernyataan SQL dieksekusi menggunakan metode execute(). Jika eksekusi berhasil, pesan JSON 'Task Deleted' dikirimkan sebagai respons. Jika tidak, pesan 'Task Not Deleted' dikirimkan.

```
$stmt->close();  
  
$conn->close();
```

Setelah pernyataan dieksekusi, pernyataan dan koneksi ke basis data ditutup untuk membebaskan sumber daya.

```
} else {  
  
    echo json_encode(['message' => 'Invalid ID']);  
  
}
```

Jika data ID yang diterima kosong, pesan JSON 'Invalid ID' dikirimkan sebagai respons.

```

1 function getTasks()
2 {
3     $conn = getConnection();
4
5     // Check if task ID is provided in the query string
6     if (isset($_GET['id'])) {
7         $id = $_GET['id'];
8         $stmt = $conn->prepare("SELECT * FROM todos WHERE id = ?");
9         $stmt->bind_param('i', $id);
10    } else {
11        $stmt = $conn->prepare("SELECT * FROM todos");
12    }
13
14    $stmt->execute();
15    $result = $stmt->get_result();
16
17    // If there are results, fetch them and return as JSON
18    if ($result->num_rows > 0) {
19        $tasks = [];
20        while ($row = $result->fetch_assoc()) {
21            $tasks[] = $row;
22        }
23        echo json_encode($tasks);
24    } else {
25        echo json_encode(['message' => 'No Tasks Found']);
26    }
27
28    $stmt->close();
29    $conn->close();
30 }

```

Funksi getTasks

```
$conn = getConnection();
```

Funksi getConnection() dipanggil untuk membuat koneksi ke basis data.

```
if (isset($_GET['id'])) {
```

```
    $id = $_GET['id'];
```

```
    $stmt = $conn->prepare("SELECT * FROM todos WHERE id = ?");
```

```
    $stmt->bind_param('i', $id);
```

```
} else {
```

```
    $stmt = $conn->prepare("SELECT * FROM todos");
```

```
}
```

Jika terdapat parameter id dalam query string URL, maka disiapkan pernyataan SQL untuk memilih tugas dengan ID tertentu. Jika tidak ada parameter id, maka pernyataan SQL disiapkan untuk memilih semua tugas dari tabel todos.

```
$stmt->execute();
```

```
$result = $stmt->get_result();
```

Pernyataan SQL dieksekusi menggunakan metode execute() dan hasilnya diperoleh menggunakan metode get_result().

```
if ($result->num_rows > 0) {
```

```
    $tasks = [];
```

```
    while ($row = $result->fetch_assoc()) {
```

```
        $tasks[] = $row;
```

```
    }
```

```
    echo json_encode($tasks);
```

```
} else {
```

```
    echo json_encode(['message' => 'No Tasks Found']);
```

```
}
```

Jika ada baris yang dikembalikan oleh query, maka hasilnya diambil dan disimpan dalam array \$tasks. Setiap baris hasil diambil menggunakan fetch_assoc() dan ditambahkan ke array \$tasks. Setelah semua hasil diambil, array tersebut dikonversi ke format JSON dan dikirimkan sebagai respons. Jika tidak ada hasil yang ditemukan, pesan JSON No Tasks Found dikirimkan sebagai respons.

```
$stmt->close();
```

```
$conn->close();
```

Setelah pernyataan dieksekusi dan hasilnya diproses, pernyataan dan koneksi ke basis data ditutup untuk membebaskan sumber daya.