

Nama : Kailana Al Kais

NIM : 21.01.55.0009

Jawaban UTS

Phpmyadmin gym equipments

The screenshot shows the phpMyAdmin web interface. The top navigation bar includes links for 'Jelajahi', 'Struktur', 'SQL', 'Cari', 'Tambahkan', 'Ekspor', 'Impor', 'Hak Akses', 'Operasi', 'Pelacakan', and 'Trigger'. The left sidebar shows a tree view of databases, with 'gym' selected. The main area displays the 'gym' database and the 'gym' table. A SQL query 'SELECT * FROM `gym`' is entered in the query box. Below the query box, there are options to 'Profil', 'Edit kotak', 'Ubah', 'Jelaskan SQL', 'Buat kode PHP', and 'Segarkan'. A table of results is shown with columns 'id', 'name', 'type', 'kondisi', and 'quantity'. The table contains two rows: one with '1 kailana pemula pemula baik' and another with '2 tegar pemula pemula baik'. Below the table, there are options to 'Pilih Semua', 'Dengan pilihan', 'Ubah', 'Salin', 'Hapus', and 'Ekspor'. At the bottom, there are links for 'Cetak', 'Salin ke clipboard', 'Ekspor', 'Tampilkan bagan', and 'Buat tampilan'.

Program pada gym_equitment.php

```
<?php
header("Content-Type: application/json; charset=UTF-8");
header("Access-Control-Allow-Origin: *");
header("Access-Control-Allow-Methods: GET, POST, PUT, DELETE");
header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, type,kondisiization, X-Requested-With");

$method = $_SERVER['REQUEST_METHOD'];
$request = [];

if (isset($_SERVER['PATH_INFO'])) {
    $request = explode('/', trim($_SERVER['PATH_INFO'], '/'));
}

function getConnection() {
    $host = 'localhost';
    $db = 'gym';
```

```

$user = 'root';
$pass = ''; // Ganti dengan password MySQL Anda jika ada
$charset = 'utf8mb4';

$dsn = "mysql:host=$host;dbname=$db;charset=$charset";
$options = [
    PDO::ATTR_ERRMODE            => PDO::ERRMODE_EXCEPTION,
    PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
    PDO::ATTR_EMULATE_PREPARES  => false,
];
try {
    return new PDO($dsn, $user, $pass, $options);
} catch (\PDOException $e) {
    throw new \PDOException($e->getMessage(), (int)$e->getCode());
}
}

function response($status, $data = NULL) {
    header("HTTP/1.1 " . $status);
    if ($data) {
        echo json_encode($data);
    }
    exit();
}

$db = getConnection();

switch ($method) {
    case 'GET':
        if (!empty($request) && isset($request[0])) {
            $id = $request[0];
            $stmt = $db->prepare("SELECT * FROM gym WHERE id = ?");
            $stmt->execute([$id]);
            $gym = $stmt->fetch();
            if ($gym) {
                response(200, $gym);
            } else {
                response(404, ["message" => "gym not found"]);
            }
        } else {
            $stmt = $db->query("SELECT * FROM gym");
            $gym = $stmt->fetchAll();
            response(200, $gym);
        }
        break;

    case 'POST':

```

```

        $data = json_decode(file_get_contents("php://input"));
        if (!isset($data->name) || !isset($data->type) || !isset($data->kondisi) || !isset($data->quantity)) {
            response(400, ["message" => "Missing required fields"]);
        }
        $sql = "INSERT INTO gym (name , type, kondisi, quantity) VALUES (?, ?, ?, ?)";
        $stmt = $db->prepare($sql);
        if ($stmt->execute([$data->name , $data->type, $data->kondisi, $data->quantity])) {
            response(201, ["message" => "gym created", "id" => $db->lastInsertId()]);
        } else {
            response(500, ["message" => "Failed to create gym"]);
        }
        break;

    case 'PUT':
        if (empty($request) || !isset($request[0])) {
            response(400, ["message" => "gym ID is required"]);
        }
        $id = $request[0];
        $data = json_decode(file_get_contents("php://input"));
        if (!isset($data->name ) || !isset($data->type,) || !isset($data->kondisi,) || !isset($data->quantity)) {
            response(400, ["message" => "Missing required fields"]);
        }
        $sql = "UPDATE gym SET name = ?, type = ?,kondisi = ?, quantity = ? WHERE id = ?";
        $stmt = $db->prepare($sql);
        if ($stmt->execute([$data->name , $data->type,$data->kondisi, $data->quantity, $id])) {
            response(200, ["message" => "gym updated"]);
        } else {
            response(500, ["message" => "Failed to update gym"]);
        }
        break;

    case 'DELETE':
        if (empty($request) || !isset($request[0])) {
            response(400, ["message" => "gym ID is required"]);
        }
        $id = $request[0];
        $sql = "DELETE FROM gym WHERE id = ?";
        $stmt = $db->prepare($sql);
        if ($stmt->execute([$id])) {
            response(200, ["message" => "gym deleted"]);
        }
    }
}

```

```

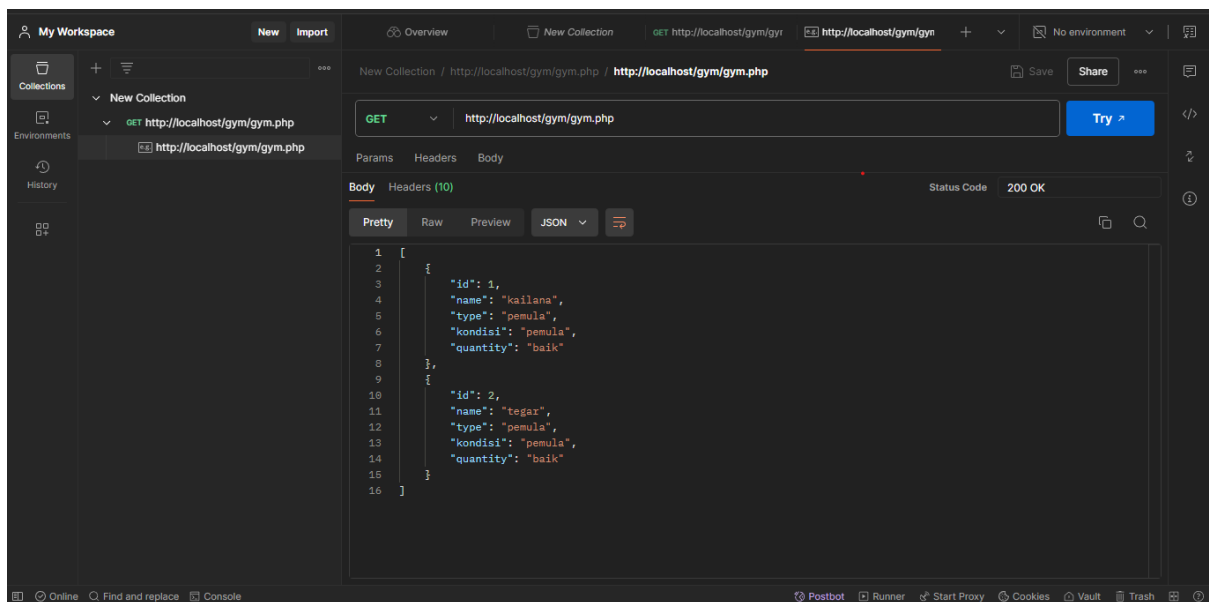
    } else {
        response(500, ["message" => "Failed to delete gym"]);
    }
    break;

default:
    response(405, ["message" => "Method not allowed"]);
    break;
}
?>

```

Hasil seluruh data

GET <http://localhost/gym/gym.php>



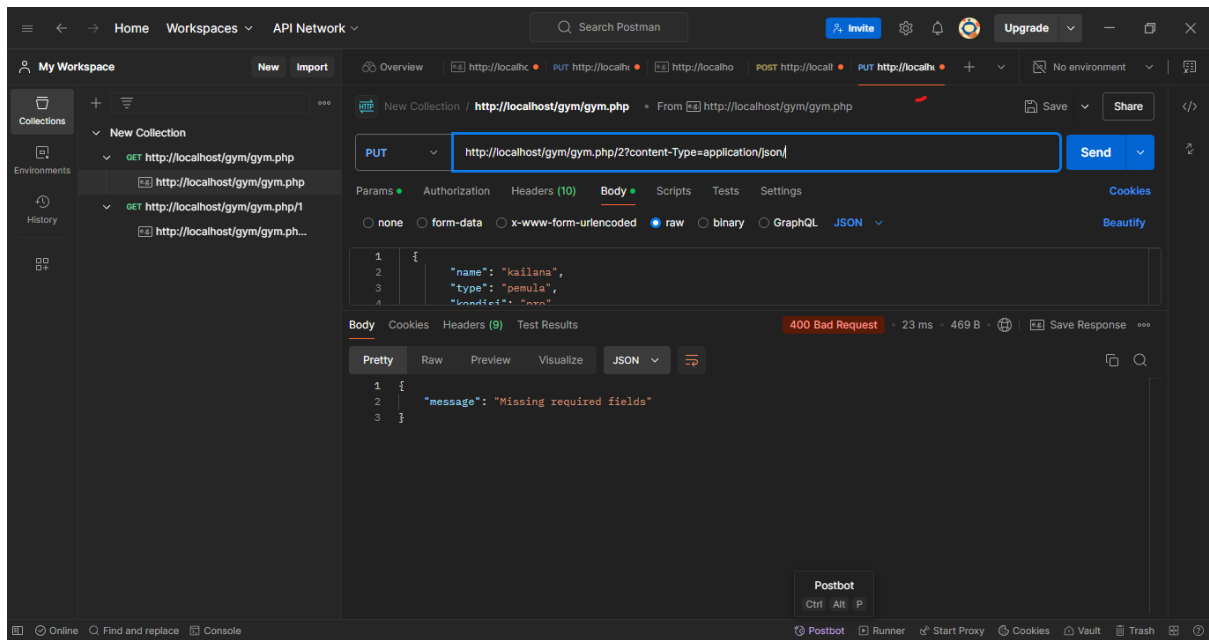
The screenshot shows the Postman interface with a new collection named 'http://localhost/gym/gym.php'. A GET request to 'http://localhost/gym/gym.php' has been executed successfully, returning a 200 OK status. The response body is displayed in JSON format, showing an array of two gym entries.

```

1  [
2    {
3      "id": 1,
4      "name": "kailana",
5      "type": "pemula",
6      "kondisi": "pemula",
7      "quantity": "baik"
8    },
9    {
10     "id": 2,
11     "name": "tegar",
12     "type": "pemula",
13     "kondisi": "pemula",
14     "quantity": "baik"
15   }
16 ]

```

put



delete

