

PRAKTIKUM 11

KONEKSI PHP DENGAN MONGODB

POKOK BAHASAN

1. Proses instal driver MongoDB pada PHP.
2. Koneksi antara PHP dan MongoDB.
3. Operasi dasar CRUD di MongoDB menggunakan PHP.

TUJUAN BELAJAR:

Setelah mempelajari materi dalam bab ini, mahasiswa diharapkan mampu:

1. Mahasiswa mampu menginstal driver MongoDB pada PHP.
2. Mahasiswa mampu membuat koneksi antara PHP dan MongoDB.
3. Mahasiswa mampu melakukan operasi dasar CRUD di MongoDB menggunakan PHP.

PENDAHULUAN

Untuk menghubungkan PHP dengan MongoDB, memerlukan driver MongoDB untuk PHP.

1. Pastikan PHP sudah terinstal di sistem.
2. Pastikan MongoDB sudah berjalan di server atau komputer lokal.
3. Disarankan untuk menggunakan **Composer** sebagai manajer paket PHP untuk menginstal MongoDB PHP Library.

1. Instalasi MongoDB dan PHP:

Pastikan Anda sudah memiliki instalasi **MongoDB** dan **PHP** di sistem Anda. Jika belum, ikuti langkah berikut:

- **Install MongoDB:**
 - Di Windows: Ikuti panduan di situs resmi [MongoDB](#).
 - Di Linux: Gunakan package manager seperti apt atau yum untuk menginstal MongoDB.
- **Install PHP:**
 - Di Linux: `sudo apt install php`
 - Di Windows: Download dari [php.net](#).

2. Instalasi MongoDB PHP Driver:

Untuk menghubungkan PHP dengan MongoDB, Anda perlu menginstal **MongoDB PHP Driver**.

Jika Anda belum menginstal MongoDB PHP Driver, Anda bisa menginstalnya dengan Composer:

1. Buka terminal atau command prompt.

2. Arahkan ke direktori proyek PHP Anda.

- Jalankan perintah berikut untuk menginstal driver:

```
bash

sudo pecl install mongodb
```

- Tambahkan ekstensi MongoDB ke file `php.ini`:

```
ini

extension=mongodb.so
```

- Verifikasi instalasi dengan menjalankan:

```
bash

php -m | grep mongodb
```

3. Instalasi Composer:

- Download dan install Composer dengan perintah berikut:

```
bash

curl -sS https://getcomposer.org/installer | php
sudo mv composer.phar /usr/local/bin/composer
```

- Setelah itu, instal library MongoDB untuk PHP:

```
bash

composer require mongodb/mongodb
```

PERCOBAAN

A. Membuat Koneksi PHP dengan MongoDB

Langkah 1: Setup Koneksi

Buat file PHP baru bernama `mongodb_connect.php` dan tambahkan kode berikut:

```
<?php
require 'vendor/autoload.php'; // Autoload MongoDB library

// Buat instance client MongoDB
$client = new MongoClient("mongodb://localhost:27017");

// Pilih database
$db = $client->selectDatabase('praktikum_db');

// Pilih koleksi
$collection = $db->selectCollection('users');

// Tampilkan pesan berhasil
echo "Koneksi ke MongoDB berhasil!";
?>
```

- Buat file PHP dan jalankan pada server lokal untuk memastikan koneksi berhasil.
- Jika pesan "Koneksi ke MongoDB berhasil!" muncul, maka koneksi berhasil.

B. Operasi CRUD (Create, Read, Update, Delete)

1. Create (Menambah Data ke MongoDB)

Buat file baru `insert.php` untuk menambahkan data ke MongoDB:

```
<?php
require 'vendor/autoload.php';

$client = new MongoClient("mongodb://localhost:27017");
$collection = $client->praktikum_db->users;

$insertOneResult = $collection->insertOne([
    'name' => 'John Doe',
    'email' => 'johndoe@example.com',
    'age' => 30
]);

printf("Inserted %d document(s)\n", $insertOneResult->getInsertedCount());
printf("Inserted document ID: %s\n", $insertOneResult->getInsertedId());
?>
```

- Jalankan file ini dan pastikan data berhasil ditambahkan ke MongoDB.
- Verifikasi dengan melihat data di MongoDB menggunakan MongoDB shell atau Compass.

2. Read (Menampilkan Data dari MongoDB)

Buat file `read.php` untuk membaca data dari MongoDB:

```
<?php
require 'vendor/autoload.php';

$client = new MongoClient("mongodb://localhost:27017");
$collection = $client->praktikum_db->users;

$cursor = $collection->find();

foreach ($cursor as $document) {
    echo $document['name'] . " - " . $document['email'] . "<br>";
}
?>
```

- Jalankan file ini untuk menampilkan semua data dari koleksi `users`.

3. Update (Memperbarui Data di MongoDB)

Buat file `update.php` untuk memperbarui data:

```
<?php
require 'vendor/autoload.php';

$client = new MongoClient("mongodb://localhost:27017");
$collection = $client->praktikum_db->users;

$updateResult = $collection->updateOne(
    ['name' => 'John Doe'],
    ['$set' => ['age' => 35]]
);

printf("Matched %d document(s)\n", $updateResult->getMatchedCount());
printf("Modified %d document(s)\n", $updateResult->getModifiedCount());
?>
```

- Jalankan file ini untuk memperbarui umur John Doe menjadi 35.
- Verifikasi hasilnya di MongoDB.

4. Delete (Menghapus Data dari MongoDB)

Buat file `delete.php` untuk menghapus data:

```
<?php
require 'vendor/autoload.php';

$client = new MongoClient("mongodb://localhost:27017");
$collection = $client->praktikum_db->users;

$result = $collection->deleteOne(['name' => 'John Doe']);

printf("Deleted %d document(s)\n", $result->getDeletedCount());
?>
```

- Jalankan file ini untuk menghapus data John Doe dari koleksi `users`.
- Verifikasi di MongoDB apakah data telah terhapus.

Latihan Praktikum

1. Tambahkan data mahasiswa (nama, email, umur) ke MongoDB menggunakan PHP.
2. Buat halaman yang menampilkan semua data mahasiswa yang tersimpan.
3. Buat form untuk memperbarui data mahasiswa berdasarkan email.
4. Buat tombol hapus untuk setiap data mahasiswa, yang ketika diklik akan menghapus data tersebut dari MongoDB.
5. Buatlah database baru bernama `Datadiri`. Di dalam database tersebut, buatlah sebuah koleksi bernama `biodata` yang menyimpan informasi masing-masing data mahasiswa dan keluarga masing-masing.
6. Gunakan operasi CRUD (Create, Read, Update, Delete) terhadap database `'Datadiri'`.
7. Tampilkan Biodata tersebut menggunakan PHP.