

# PRAKTIK PENGEMBANGAN BACK END

19 November 2024

## FUNGSI READ DI LARAVEL

---

### a. Apa itu CRUD

CRUD adalah singkatan dari Create, Read, Update, dan Delete, yang merupakan empat fungsi dasar yang digunakan untuk mengelola data dalam basis data dan penting untuk sistem manajemen basis data. Fungsi-fungsi ini sering digunakan sebagai dasar untuk membangun aplikasi web.

Pertimbangkan panel admin atau backend apa pun, CRUD akan menjadi fungsi pentingnya untuk mengelola data.

### Membuat / create

Operasi create digunakan untuk **menambahkan data baru ke dalam basis data**. Misalnya, jika Anda memiliki basis data pelanggan dan ingin menambahkan pelanggan baru ke dalam basis data, Anda dapat menggunakan operasi create. Hal ini dapat melibatkan penambahan rekaman baru ke tabel pelanggan dengan nama pelanggan, alamat, dan informasi lainnya.

### Membaca / read

Operasi baca digunakan untuk **mengambil data dari basis data**. Misalnya, jika Anda ingin mengambil daftar semua pelanggan dalam basis data, Anda akan menggunakan operasi baca. Ini dapat melibatkan permintaan ke tabel pelanggan untuk mengambil semua rekaman dan menampilkannya di situs web atau aplikasi.

### Memperbarui / update

Operasi pembaruan digunakan untuk **mengubah data yang ada dalam basis data**. Misalnya, jika Anda ingin memperbarui alamat pelanggan dalam basis data, Anda akan menggunakan operasi pembaruan. Ini dapat melibatkan perubahan catatan dalam tabel pelanggan dengan informasi alamat baru.

## Menghapus / delete

Operasi hapus digunakan untuk **menghapus data dari basis data** . Misalnya, jika Anda ingin menghapus pelanggan dari basis data, Anda akan menggunakan operasi hapus. Ini dapat melibatkan penghapusan rekaman dari tabel pelanggan.

### b. Apa Itu Model

Model menjadi salah satu bagian penting dari konsep [MVC](#) (Model-View-Controller) pada framework Laravel yang bertanggung jawab untuk mengatur interaksi antara aplikasi dengan database. Dalam konsep MVC, Model dipadukan dengan Controller dan View untuk membentuk sebuah fitur atau halaman pada aplikasi Laravel kita. Model bertanggung jawab untuk memproses data dari database dan mengembalikan data tersebut ke Controller.

### c. Apa itu Controller

Dalam konteks Laravel, controller adalah kelas PHP yang bertanggung jawab untuk memproses permintaan HTTP dari pengguna dan mengembalikan respons yang sesuai. Controller menyatukan logika aplikasi dan memastikan bahwa permintaan dari pengguna diarahkan dengan benar ke model yang sesuai atau ke tampilan yang tepat.

### d. Apa itu Routes / routing

Routing adalah salah satu komponen inti dalam aplikasi web yang memungkinkan Anda mendefinisikan alamat URL yang akan dipetakan ke controller tertentu dalam aplikasi Anda. Dalam Laravel, routing digunakan untuk mengarahkan semua request HTTP ke handler / method yang tepat.

Routing di Laravel sangat fleksibel dan memberi Anda kontrol penuh atas setiap request ditangani oleh aplikasi Anda. Ini

memungkinkan struktur yang rapi dan mudah dikelola untuk aplikasi yang kompleks.

e. Konsep MVC (Model, View, Controller)

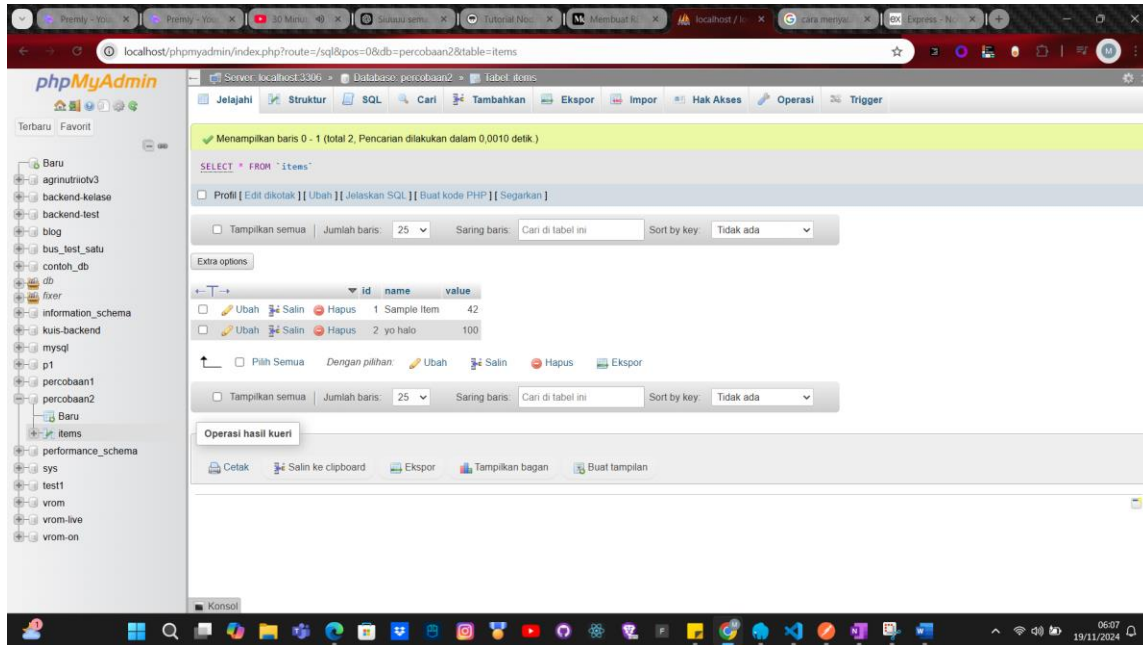
MVC adalah sebuah pendekatan perangkat lunak yang memisahkan aplikasi logika dari presentasi. MVC memisahkan aplikasi berdasarkan komponen- komponen aplikasi, seperti : manipulasi data, controller, dan user interface.

1. Model, Model mewakili struktur data. Biasanya model berisi fungsi-fungsi yang membantu seseorang dalam pengelolaan basis data seperti memasukkan data ke basis data, pembaruan data dan lain-lain.
2. View, View adalah bagian yang mengatur tampilan ke pengguna. Bisa dikatakan berupa halaman web.
3. Controller, Controller merupakan bagian yang menjembatani model dan view.

## e. Proses Pembuatan Fungsi/Function read/Read/Show di Laravel

1. Pertama, cek terlebih dahulu apakah table/data yang ingin ditunjukkan pastikan sudah ada :

Contoh :



2. Kemudian buka Kembali express.js nya, kemudian pergi ke server.js, dan tambahkanlah function read ini :

- Function read/show semua data table (untuk disini Namanya index)

```
// GET route untuk mengambil semua data
app.get('/items', (req, res) => {
  const query = 'SELECT * FROM items';
  db.query(query, (err, results) => {
    if (err) {
      console.error('Error fetching data: ', err);
      return res.status(500).json({ message: 'Database error' });
    }

    res.status(200).json(results);
  });
});
```

- Function read /show data berdasarkan id (disini nama functionnya show(\$id), "\$id" = digunakan khusus untuk dilakukan oleh id yang dipanggil

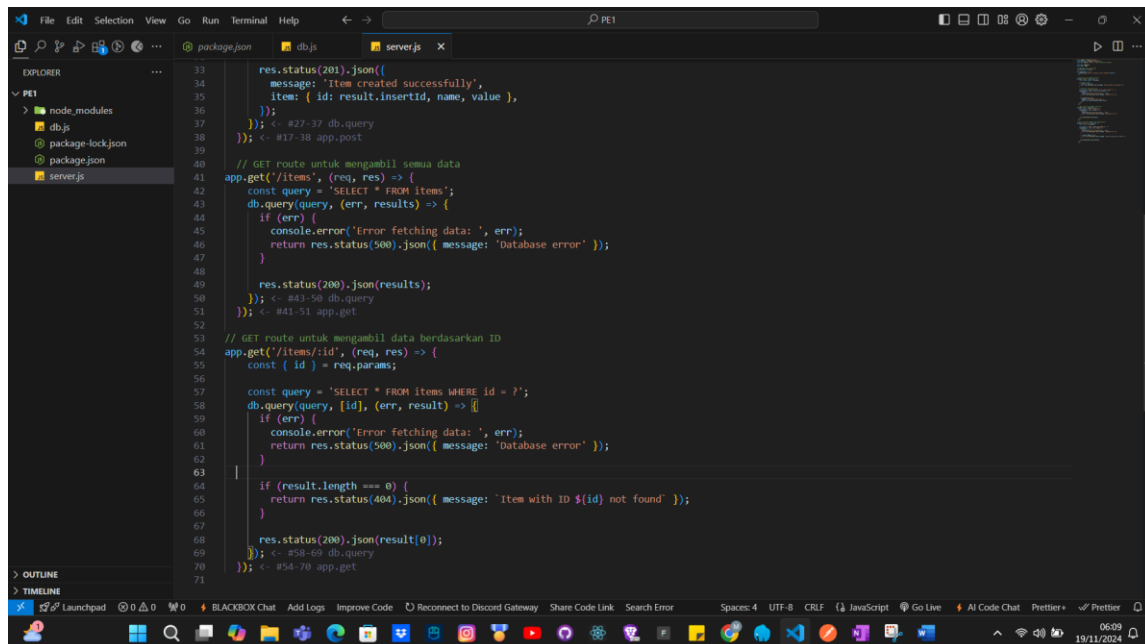
```
// GET route untuk mengambil data berdasarkan ID
app.get('/items/:id', (req, res) => {
  const { id } = req.params;

  const query = 'SELECT * FROM items WHERE id = ?';
  db.query(query, [id], (err, result) => {
    if (err) {
      console.error('Error fetching data: ', err);
      return res.status(500).json({ message: 'Database error' });
    }

    if (result.length === 0) {
      return res.status(404).json({ message: `Item with ID ${id} not found` });
    }

    res.status(200).json(result[0]);
  });
});
```

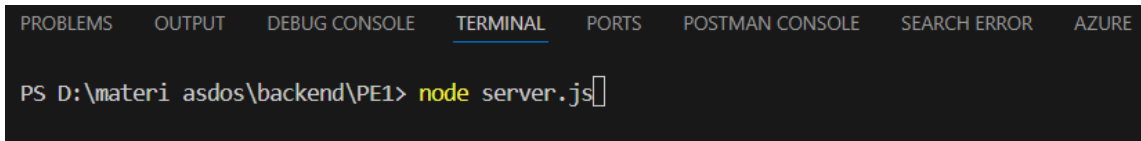
- Contoh penerapan :



```
33 res.status(201).json({
34   message: 'Item created successfully',
35   item: { id: result.insertId, name, value },
36 });
37 // < #27-37 db.query
38 // < #17-38 app.post
39
40 // GET route untuk mengambil semua data
41 app.get('/items', (req, res) => {
42   const query = 'SELECT * FROM items';
43   db.query(query, (err, results) => {
44     if (err) {
45       console.error('Error fetching data: ', err);
46       return res.status(500).json({ message: 'Database error' });
47     }
48
49     res.status(200).json(results);
50   });
51 // < #41-54 db.query
52 // < #41-51 app.get
53
54 // GET route untuk mengambil data berdasarkan ID
55 app.get('/items/:id', (req, res) => {
56   const { id } = req.params;
57
58   const query = 'SELECT * FROM items WHERE id = ?';
59   db.query(query, [id], (err, result) => {
60     if (err) {
61       console.error('Error fetching data: ', err);
62       return res.status(500).json({ message: 'Database error' });
63     }
64
65     if (result.length === 0) {
66       return res.status(404).json({ message: `Item with ID ${id} not found` });
67     }
68
69     res.status(200).json(result[0]);
70   });
71 // < #54-69 db.query
72 // < #54-70 app.get
```

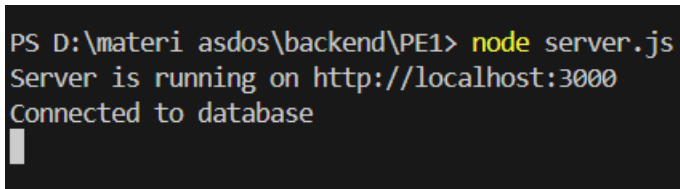
3. Kemudian buka kembali terminal dan nyalakan projectnya dengan command :

“ node server.js “



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE SEARCH ERROR AZURE

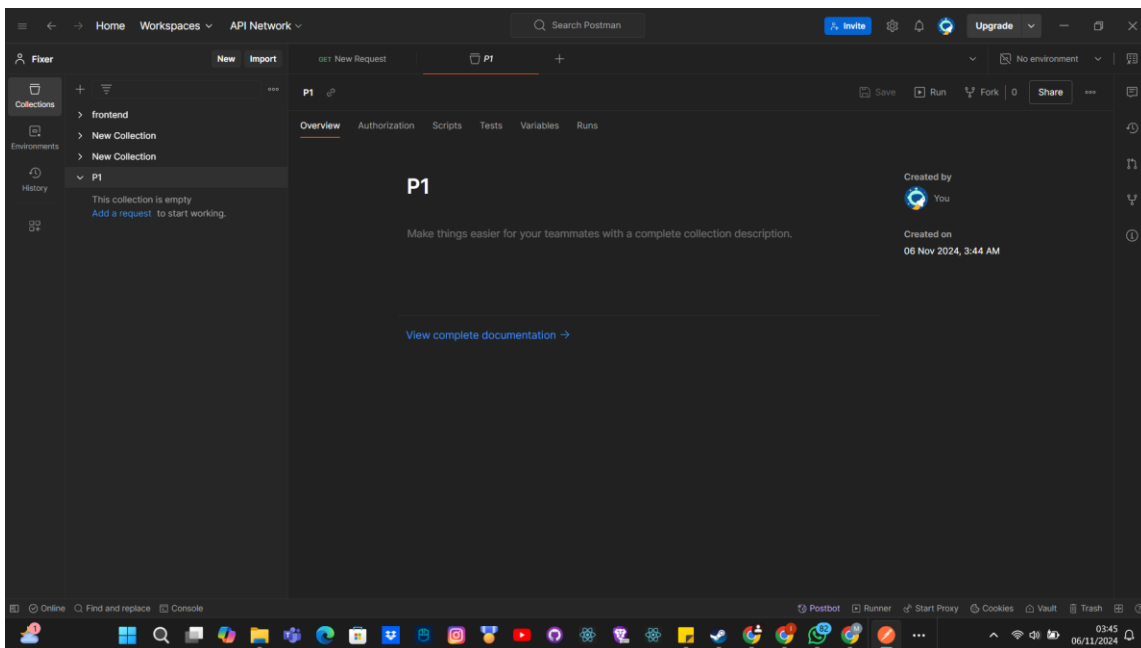
PS D:\materi asdos\backend\PE1> node server.js
```



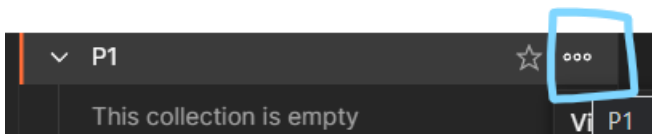
```
PS D:\materi asdos\backend\PE1> node server.js
Server is running on http://localhost:3000
Connected to database
```

4. Kemudian buka postman, setelah itu bukalah collection yang kalian sudah buat sebelumnya

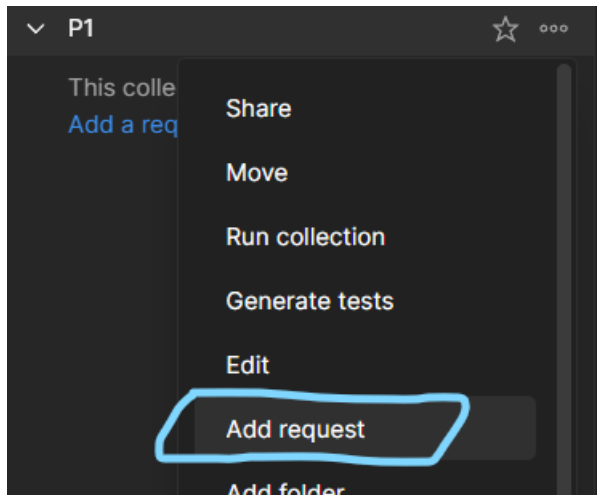
Contoh :



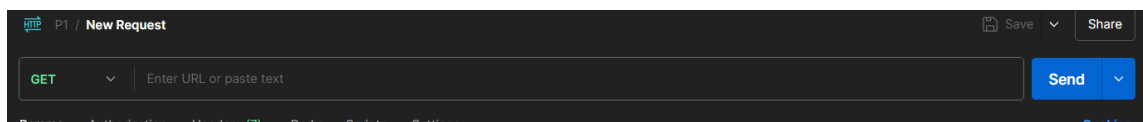
5. Setelah terbuat, klik titik tiga yang berada di sebelah nama project



6. Kemudian klik add request



7. Setelah itu biarkanlah requestnya tetap Get

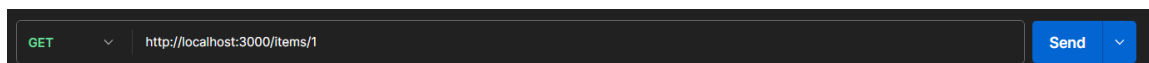


8. Setelah itu isilah url atasnya sesuai dengan routes yang dibuat :

- Function read/show semua data table  
**URL:** `http://localhost:3000/items`

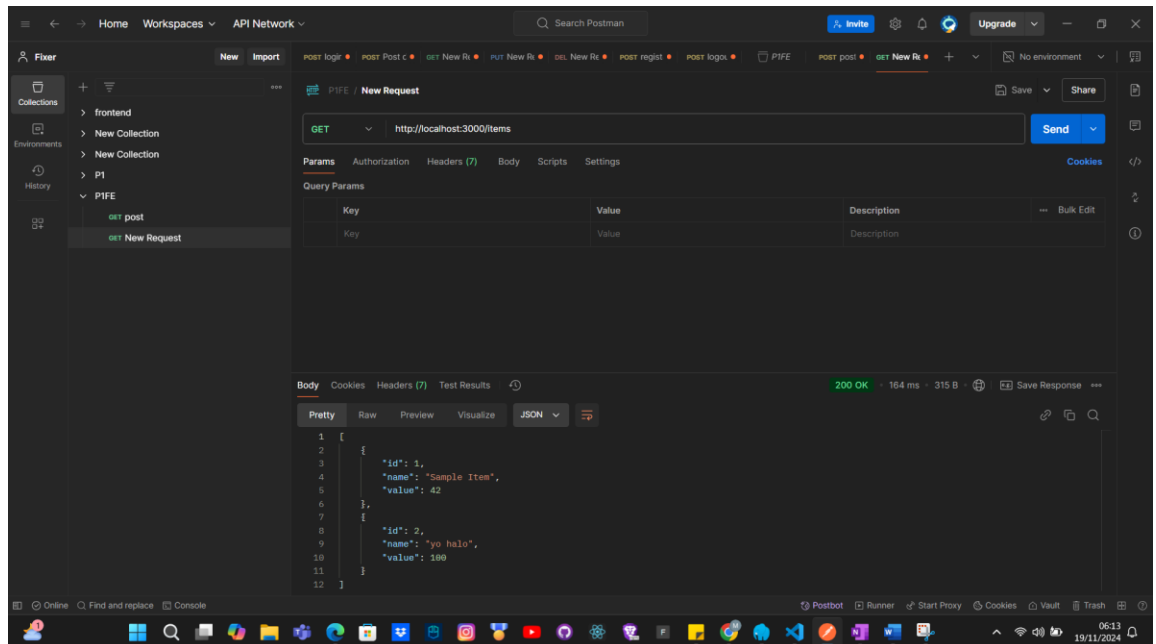


- Funtion read /show data berdasarkan id  
**URL:** `http://localhost:3000/items/{id}`

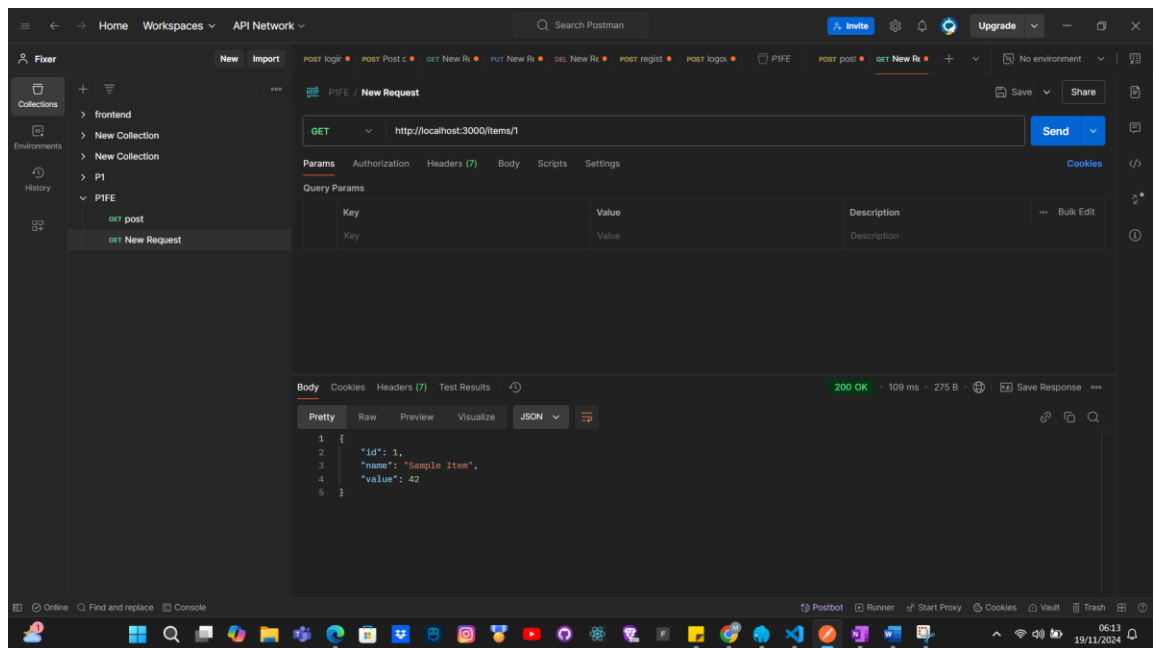


9. Kemudian klik send dan hasilnya seperti ini

- Function read/show semua data table



- Function read /show data berdasarkan id





## 2. TUGAS

Jawablah Pertanyaan dari Soal ini |

1. Cobalah untuk melakukan semua percobaan diatas
2. Kemudian buatlah versi project kalian sendiri
3. minimal sudah 3 table yang bisa melakukan proses read/show semua dan berdasarkan dengan id

### PENGUMPULAN TUGAS DAN DEADLINE

- a. Pengumpulan Tugas di Google Drive
- b. Deadline = 19 November 2024 | 23.00 WIB

## DAFTAR PUSTAKA

<https://medium.com/@kevinffa0107/pengertian-mvc-model-view-controller-pada-framework-laravel-20f261ccf233>

<https://medium.com/chevalier-lab/membuat-restful-api-node-js-express-mysql-crud-c4a1512600b6>

<https://santrikoding.com/tutorial-set/tutorial-nodejs-express-mysql>

<https://expressjs.com/>