

PRAKTIK PENGEMBANGAN BACK END

20 November 2024

FUNGSI UPDATE DI EXPRESS.JS

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)

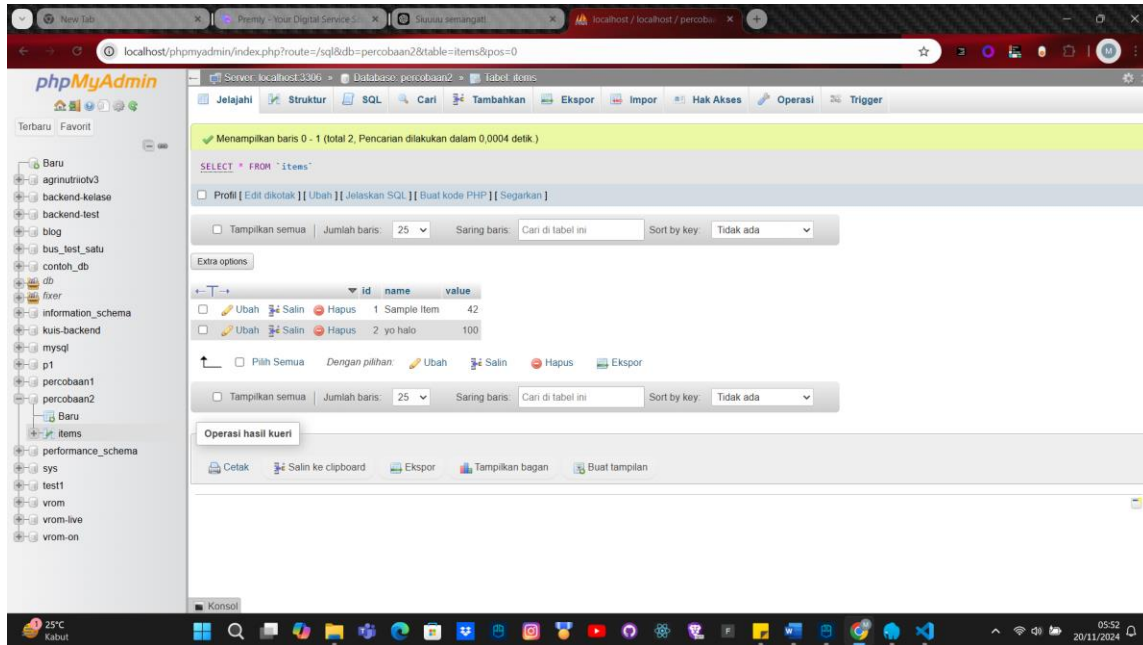
MVC adalah sebuah pendekatan perangkat lunak yang memisahkan aplikasi logika dari presentasi. MVC memisahkan aplikasi berdasarkan komponen- komponen aplikasi, seperti : manipulasi data, controller, dan user interface.

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 Update/Edit Express.js

1. Pertama, cek terlebih dahulu apakah table/data yang ingin ditunjukkan pastikan sudah ada, dan di table tersebut terdapat variabelnya :

Contoh :



2. Kemudian pergi Kembali ke server.js di express.jsnya , dan buatlah function untuk update, dengan kode ini :

- Function update berdasarkan id yang dipilih

```
app.put('/items/:id', (req, res) => {  
  const { id } = req.params; // Ambil ID dari parameter URL  
  const { name, value } = req.body; // Ambil data dari body request  
  
  if (!name || !value) {  
    return res.status(400).json({ message: 'Name and Value are required' });  
  }  
  
  // Query SQL untuk memperbarui data  
  const query = 'UPDATE items SET name = ?, value = ? WHERE id = ?';
```

```

db.query(query, [name, value, id], (err, result) => {

  if (err) {

    console.error('Error updating data: ', err);

    return res.status(500).json({ message: 'Database error' });

  }

  if (result.affectedRows === 0) {

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

  }

  res.status(200).json({ message: `Item with ID ${id} updated successfully` });

});
});

```

- Contoh penerapan :

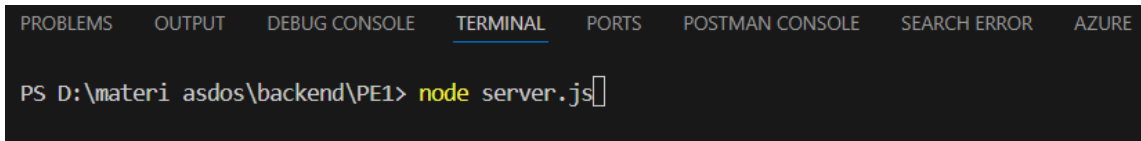
```

71 // PUT route untuk memperbarui data berdasarkan ID
72 app.put('/items/:id', (req, res) => {
73   const { id } = req.params; // Ambil ID dari parameter URL
74   const { name, value } = req.body; // Ambil data dari body request
75
76   if (!name || !value) {
77     return res.status(400).json({ message: 'Name and Value are required' });
78   }
79
80   // Query SQL untuk memperbarui data
81   const query = 'UPDATE items SET name = ?, value = ? WHERE id = ?';
82   db.query(query, [name, value, id], (err, result) => {
83     if (err) {
84       console.error('Error updating data: ', err);
85       return res.status(500).json({ message: 'Database error' });
86     }
87
88     if (result.affectedRows === 0) {
89       return res.status(404).json({ message: `Item with ID ${id} not found` });
90     }
91
92     res.status(200).json({ message: `Item with ID ${id} updated successfully` });
93   });
94   // #83-94 db.query
95   // #73-95 app.put
96

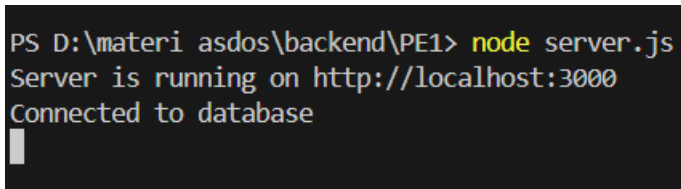
```

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

“ node server.js “



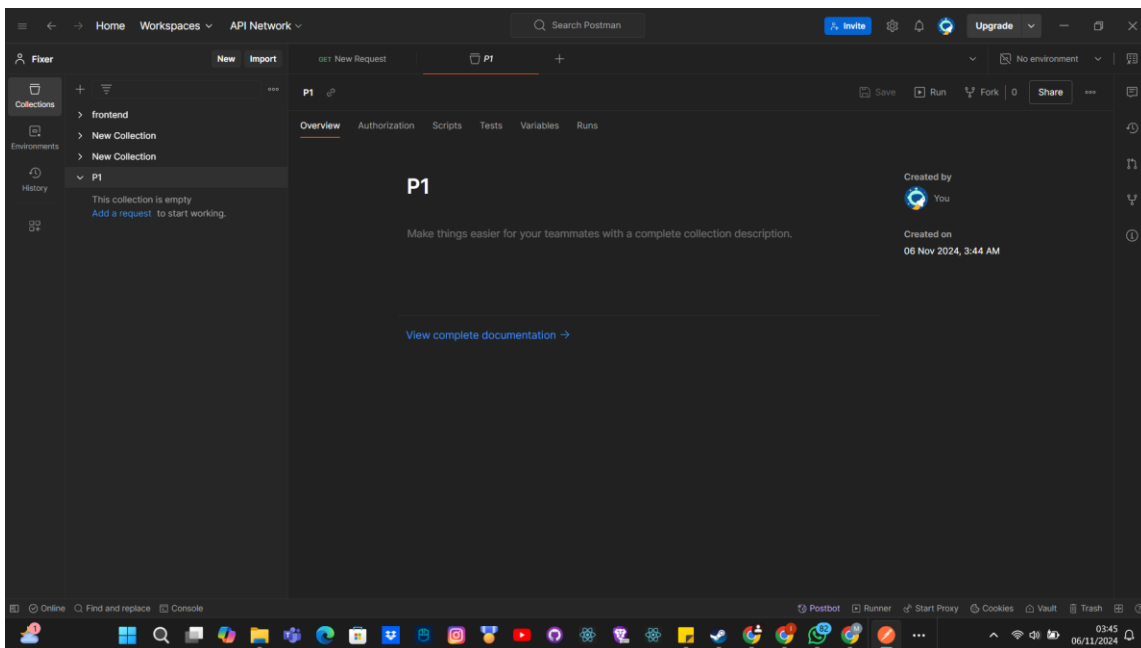
```
PS D:\materi asdos\backend\PE1> node server.js
```



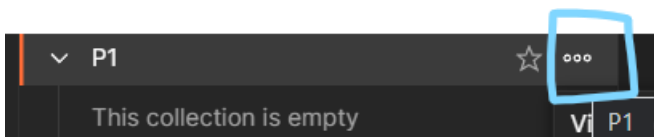
```
PS D:\materi asdos\backend\PE1> node server.js
Server is running on http://localhost:3000
Connected to database
```

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

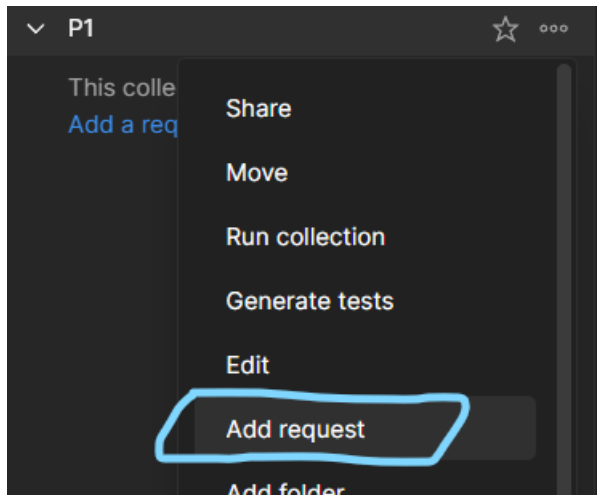
Contoh :



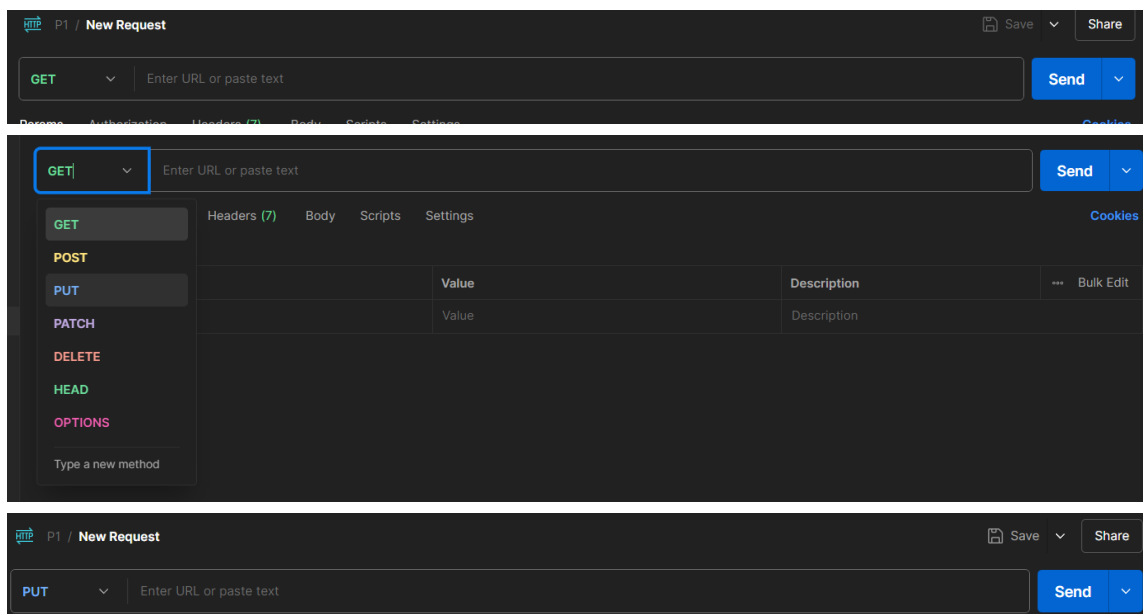
5. Setelah terbuat, klik titik tiga yang berada di sebelah nama project



6. Kemudian klik add request

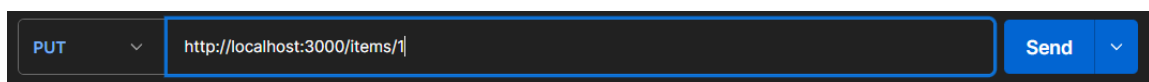


7. Setelah itu gantilah requestnya menjadi PUT



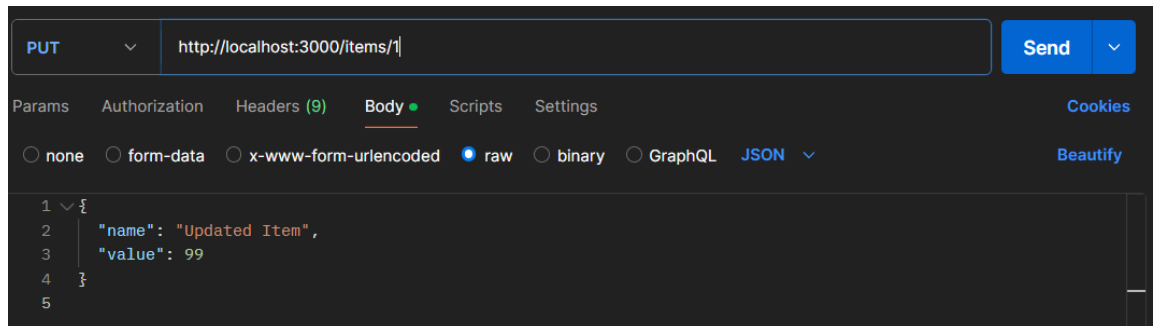
8. Setelah itu isilah url atasnya sesuai dengan routes yang dibuat :

- Function update berdasarkan id yang dipilih
URL: `http://localhost:3000/items/:id`

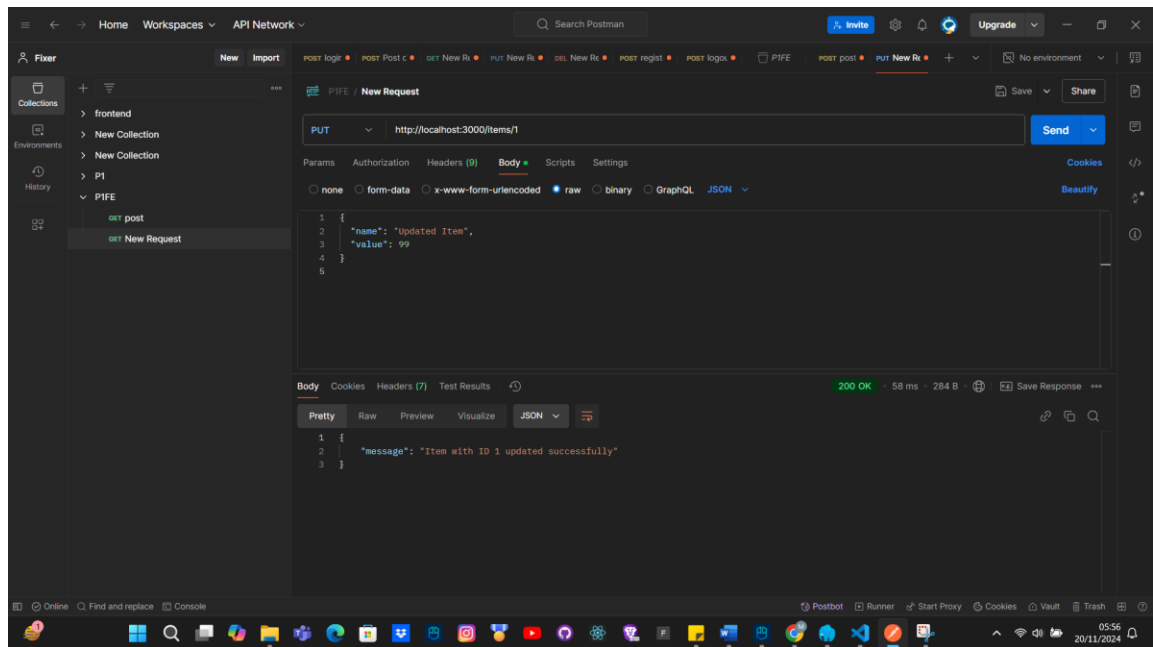


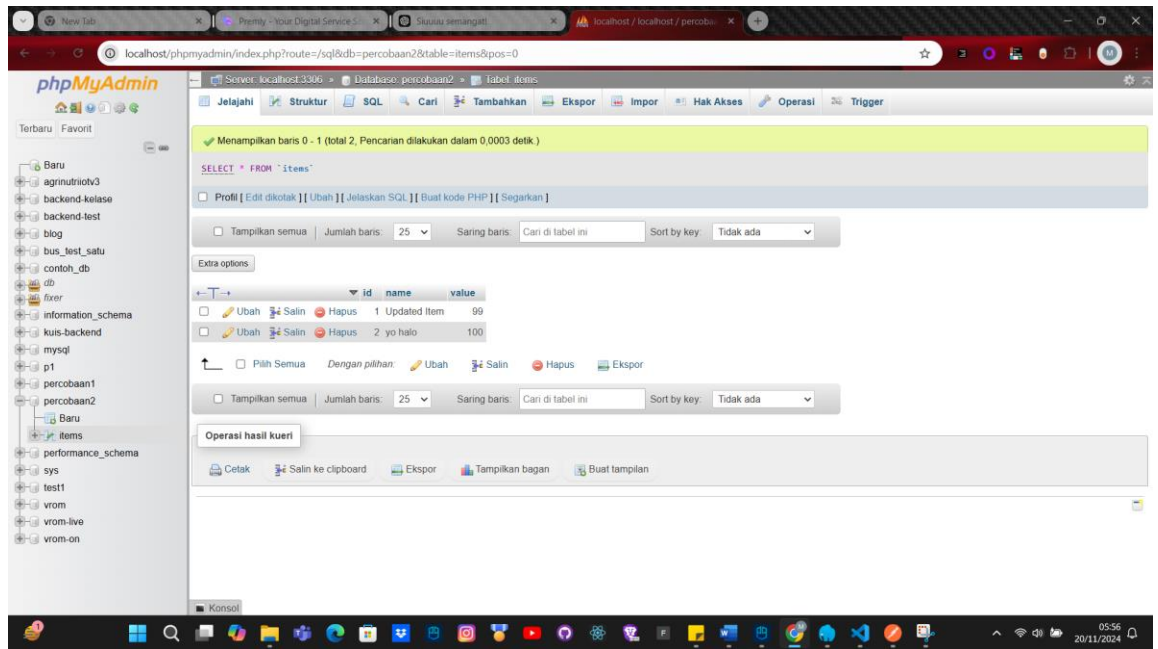
9. Kemudian klik isilah raw datanya, dengan raw data yang yang ingin diubah

- Function update berdasarkan id yang dipilih (contoh disini mau edit data dari id 1), dan klik send



Hasil :





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 update berdasarkan dengan id(primary key dari table tersebut)

PENGUMPULAN TUGAS DAN DEADLINE

- a. Pengumpulan Tugas di Google Drive
- b. Deadline = 20 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/>