

# BACKEND CRUD APLIKASI WEBSITE E-COMMERCE

Donny Hamzah P Harahap, Nursina Nasir

[donny.hamzah@upi-yai.ac.id](mailto:donny.hamzah@upi-yai.ac.id), [nursina@upi-yai.ac.id](mailto:nursina@upi-yai.ac.id)

## Abstrac

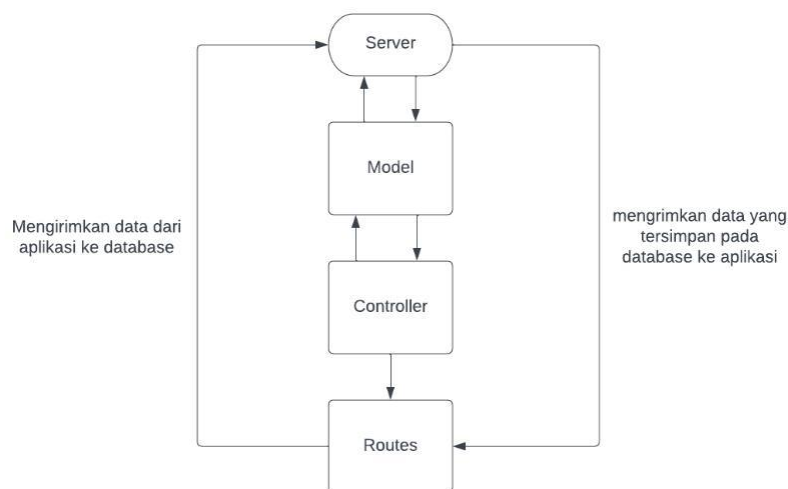
"E-commerce application backend". To create this application, the user interface on the product page is quite important to make it easier for users to search for and buy a product. Therefore, this report will focus on creating E-commerce applications. Based on the information above, in writing this journal the title "Creating a CRUD Backend for E-commerce Website Applications" was chosen, which is the interface system for an E-commerce website application.

**Keyword** "Creating a CRUD Backend for E-commerce Website Applications" was chosen, which is the interface system for an E-commerce website application.

## Perancangan Aplikasi

### Arsitektur Aplikasi

Arsitektur Aplikasi adalah rancangan desain menyeluruh dari sebuah sistem aplikasi beserta infrastruktur pendukungnya untuk dapat melayani kebutuhan bisnis atau pendukung bisnis yang disematkan pada aplikasi tersebut.



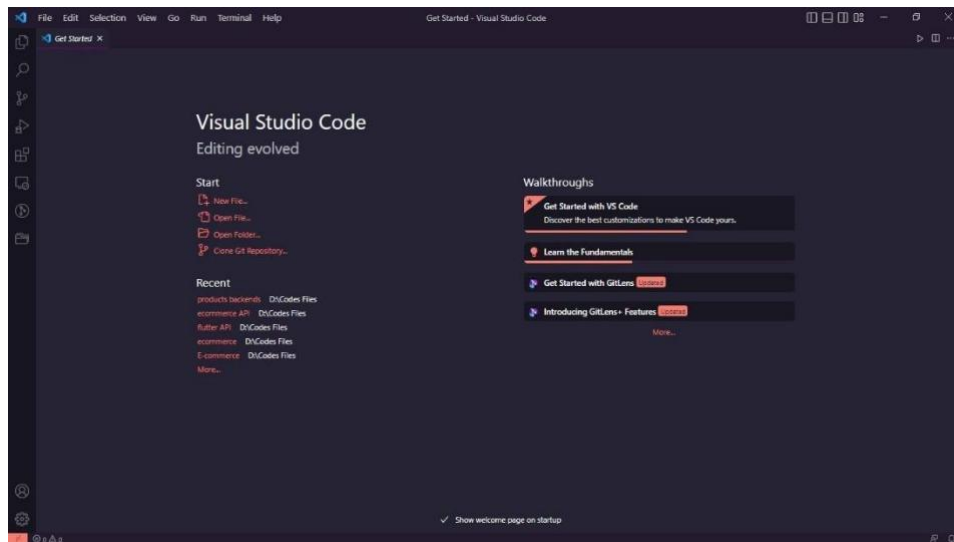
Gambar Arsitektur Aplikasi

## • Tools dan Frameworks

Tools dan aplikasi yang digunakan dalam pembuatan backend dari aplikasi E-commerce, yaitu:

### *Visual Studio Code (VS Code)*

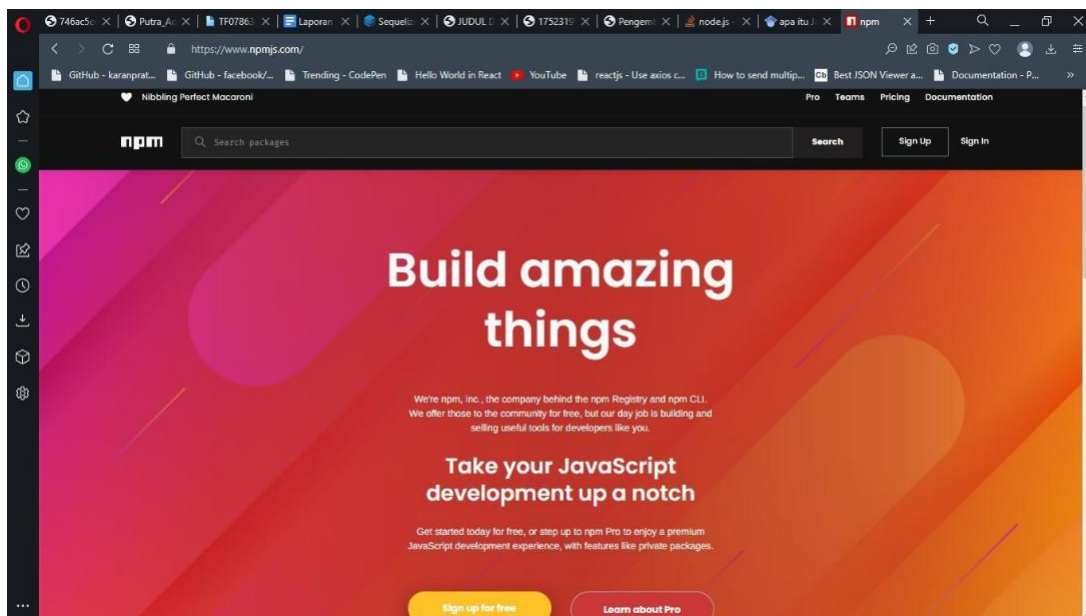
Visual Studio Code atau biasa disebut dengan *VS Code* adalah editor kode sumber yang dikembangkan oleh Microsoft untuk Windows, Linux dan macOS.



Gambar Tampilan Visual Studio Code 2.

### *Node Package Manager (NPM)*

NPM adalah salah satu registri software terbesar saat ini yang digunakan oleh para developer untuk saling berbagi kode package JavaScript. Secara default, NPM sudah disertakan dengan node.js.

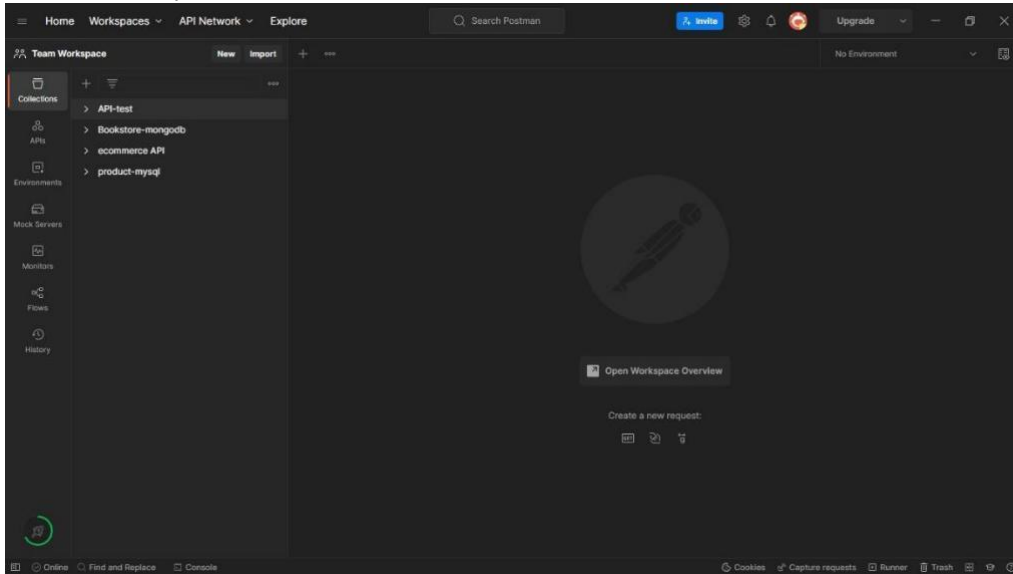


Gambar Tampilan Node Package Manager (NPM)

### Postman

Postman adalah developing tools yang membantu penggunaannya untuk membangun, menguji, dan memodifikasi API. Ketika menjalankan pengujian, Postman mengirim request API ke server web dan kemudian menerima segala jenis respons. Sesuai dengan namanya, ia berfungsi layaknya tukang pos.

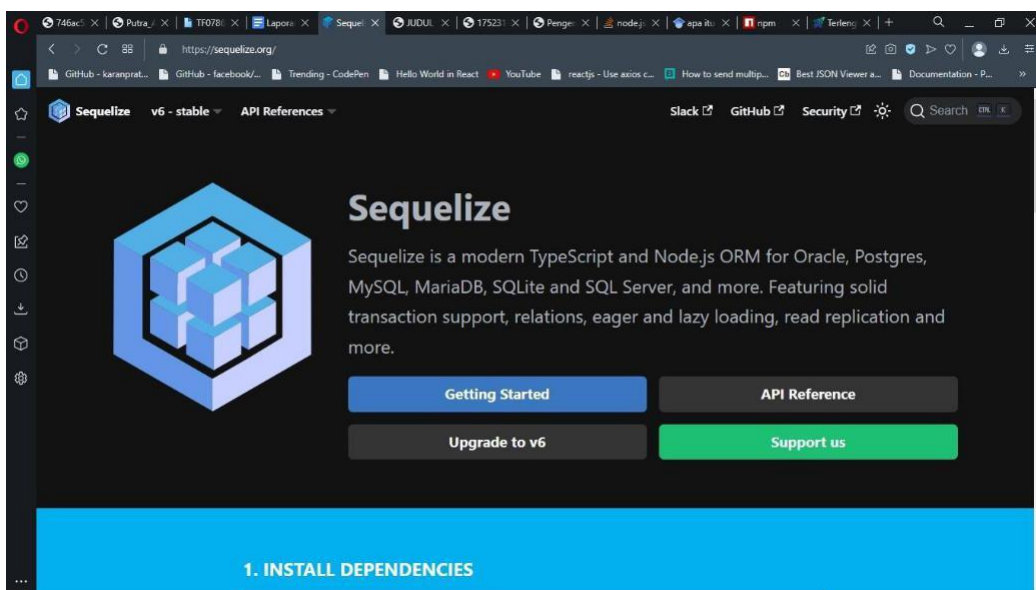
Aplikasi ini dapat mengakomodasi berbagai jenis request HTTP, menyimpan lingkungan untuk penggunaan selanjutnya, hingga mengubah API menjadi kode dengan berbagai bahasa pemrograman, seperti JavaScript dan Python.



Gambar Tampilan Postman

### Sequelize

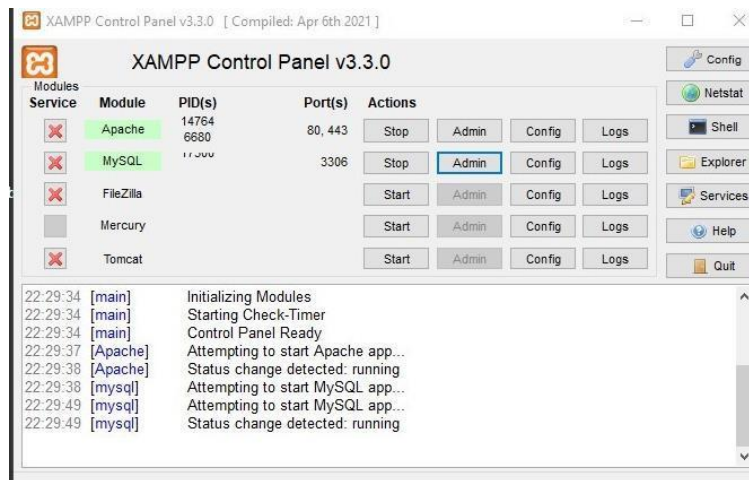
Sequelize adalah Node.js promise-based ORM untuk MySQL, PostgreSQL, SQLite, MSSQL dan database SQL lainnya. Sequelize berfungsi untuk bekerja dengan database dan relasi-relasi di dalamnya.



Gambar Tampilan Sequelize

## XAMPP

XAMPP yaitu program aplikasi pengembang yang berguna untuk pengembangan website berbasis PHP dan MySQL.



Gambar Tampilan XAMPP

### Pembuatan Aplikasi Backend

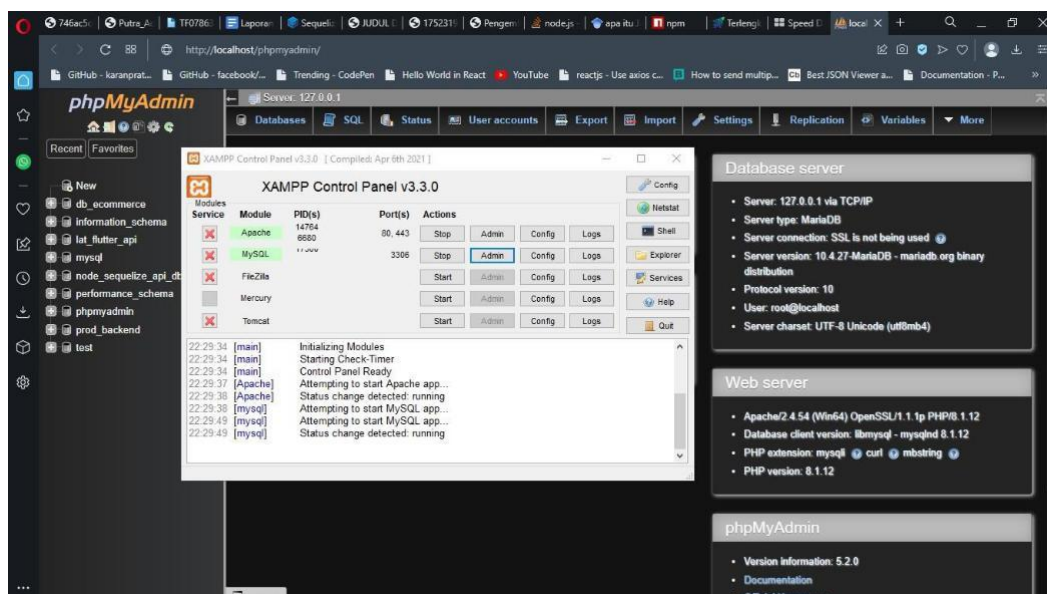
Berikut ini adalah proses atau langkah – langkah dalam pembuatan Aplikasi Backend

#### 1. Download dan Install Tools dan Aplikasi

Beberapa Software yang harus di install sebelum pembuatan backend adalah Visual Studio Code, XAMPP, dan Postman.

#### 2. Aktifkan Apache dan MySQL pada XAMPP

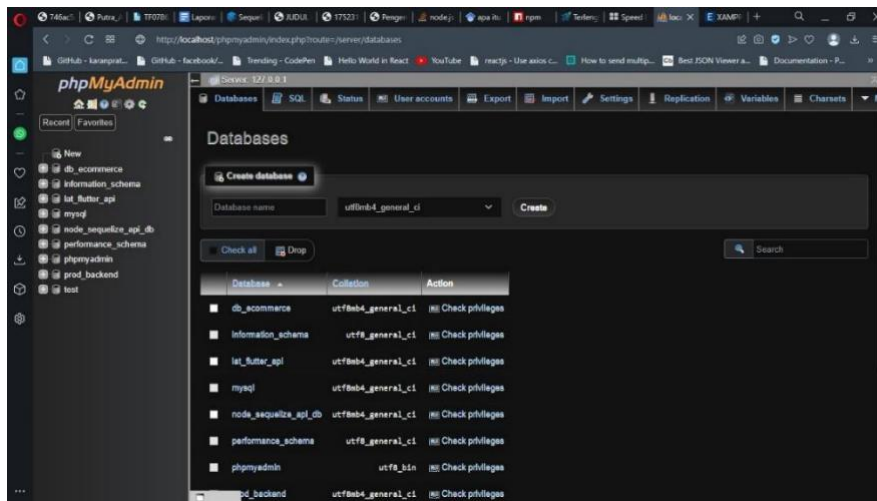
Saat menjalankan aplikasi XAMPP pastikan untuk “Run as Administrator” terlebih dahulu sebelum menjalankan Apache dan MySQL digunakan untuk membuat server Open Source. Setelah Apache dan MySQL berwarna hijau itu artinya server sudah berjalan dan bisa digunakan. Setelah itu tekan tombol admin pada MySQL untuk mengakses phpMyAdmin.



Gambar XAMPP dan phpMyAdmin

### 3. Membuat *database* baru pada *phpMyAdmin*

Tekan tombol new pada menu di sebelah kiri lalu namai database yang akan digunakan. Di proses ini database yang akan digunakan dinamai sebagai `db_ecommerce`.

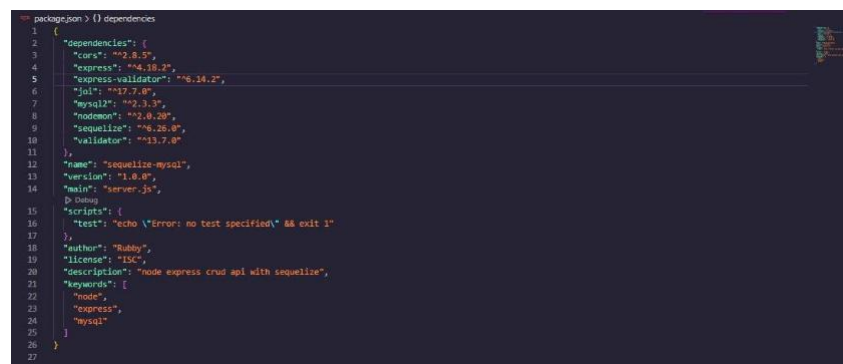


Gambar Membuat database pada *phpMyAdmin*

### 4. Persiapan menginstall *Middlewares* atau *Packages* yang akan digunakan.

Sebelum menginstall *Middlewares* di rekomendasikan untuk membuat *folder* kosong terlebih dahulu untuk menyimpan kodingan untuk backend. Setelah itu buka VS Code dan pilih *folder* yang telah dibuat. untuk backend ini *folder* dinamai ***product backends***. lalu buka terminal dan ketik pada terminal “`npm init`”.

Pada terminal akan muncul beberapa pertanyaan pada langkah ini tekan saja *Enter* karena yang dibutuhkan hanyalah *package* nya saja. jika sudah akan terbuat sebuah *file* yang bernama ***package.json*** masukan *dependencies* (*Packages*) yang digunakan untuk menginstall *Middlewares*.



Gambar 3.12 *Dependencies* yang digunakan pada pembuatan *backend*

### 5. Menginstall *Middlewares*

Untuk menginstall ketikkan “`npm install`” atau “`npm i`” pada terminal dan semua *dependencies* yang telah dimasukan kepada “*package.json*” akan terinstall dengan sendirinya



## 6. Membuat Config

Buat *folder* baru dengan nama **config** lalu buat *file* baru di dalam *folder config* dengan nama **dbConfig**, *file* ini di buat sebagai setting dari *file model/index.js* yang bertujuan untuk menghubungkan server dengan aplikasi. berikut adalah pengaturan pada **dbConfig**. Pastikan nama DB yang dimasukan sama dengan yang ada di *phpMyAdmin*

```
config > JS dbConfig.js > [?] <unknown> > DB
1  module.exports = {
2    // config untuk database
3    HOST: 'localhost',
4    USER: 'root',
5    PASSWORD: '',
6    DB: 'db_ecommerce',
7    dialect: 'mysql',
8
9    pool: {
10     max: 5,
11     min: 0,
12     acquire: 30000,
13     idle: 10000
14   }
15 }
```

Gambar *config/dbConfig*

## 7. Membuat Model

Di proses ini akan dibuat *folder models* beberapa *files* untuk model – model yang berbeda. *Files* tersebut yaitu:

### a. *models/Index.js*

*files* ini berfungsi sebagai penghubung antara database dengan model – model yang akan dibuat. berikut adalah codingan dari *index.js*.

```
models > JS index.js > ...
1  const dbConfig = require('../config/dbConfig.js');
2
3  const { Sequelize, DataTypes } = require('sequelize');
4
5  const sequelize = new Sequelize(
6    dbConfig.DB,
7    dbConfig.USER,
8    dbConfig.PASSWORD, {
9    host: dbConfig.HOST,
10   dialect: dbConfig.dialect,
11   operatorAliases: false,
12
13   pool: {
14     max: dbConfig.pool.max,
15     min: dbConfig.pool.min,
16     acquire: dbConfig.pool.acquire,
17     idle: dbConfig.pool.idle
18   }
19 });
20
21
22
23 sequelize.authenticate()
24 .then(() => {
25   console.log('connecting...');
26 })
27 .catch(err => {
28   console.log('Error' + err);
29 });
30
31 const db = {}
32
33 db.Sequelize = Sequelize;
34 db.sequelize = sequelize;
35
36 db.prods = require('./productModel.js')(sequelize, DataTypes);
37 db.details = require('./detailModel.js')(sequelize, DataTypes);
38 db.reviews = require('./reviewsModel.js')(sequelize, DataTypes);
39
40 db.sequelize.sync({ force: false })
41 .then(() => {
42   console.log('yes, re-sync done');
```

Gambar Codingan *index.js* (part 1)

```
40 db.sequelize.sync({ force: false})
41 .then(() => {
42   console.log('yes, re-sync done');
43 });
44
45
46 // 1. One to Many Relationship // detail to prods
47 db.prods.hasMany(db.details, {
48   foreignKey: 'prod_id',
49   as: 'Product_Details'
50 });
51
52 db.details.belongsTo(db.prods, {
53   foreignKey: 'prod_id',
54   as: 'product'
55 });
56
57 // 2. One to Many Relationship // reviews to prods
58 db.prods.hasMany(db.reviews, {
59   foreignKey: 'prod_id',
60   as: 'Product_Reviews'
61 });
62
63 db.reviews.belongsTo(db.prods, {
64   foreignKey: 'prod_id',
65   as: 'product'
66 });
67
68 module.exports = db;
```

Gambar Codingan *index.js* (part 2)

**b. *model/productModel.js***

Model dari product yang akan di buat oleh user dan datanya akan dikirimkan ke database. Berikut adalah codingan dari *productModel.js*.

```
models > JS productModel.js > <unknown> > exports
1  module.exports = (sequelize, DataTypes) => {
2  ...
3  const Prod = sequelize.define("prod", {
4    title: {
5      type: DataTypes.STRING,
6      allowNull: false,
7      validate: {
8        notEmpty: {
9          args: true,
10         msg: 'please fill the title'
11       }
12     },
13     description: {
14       type: DataTypes.STRING,
15       allowNull: false,
16     },
17   });
18   return Prod;
19 }
20
21
22
23
```

Gambar Codingan *productModel.js*

**c. *model/detailModel.js***

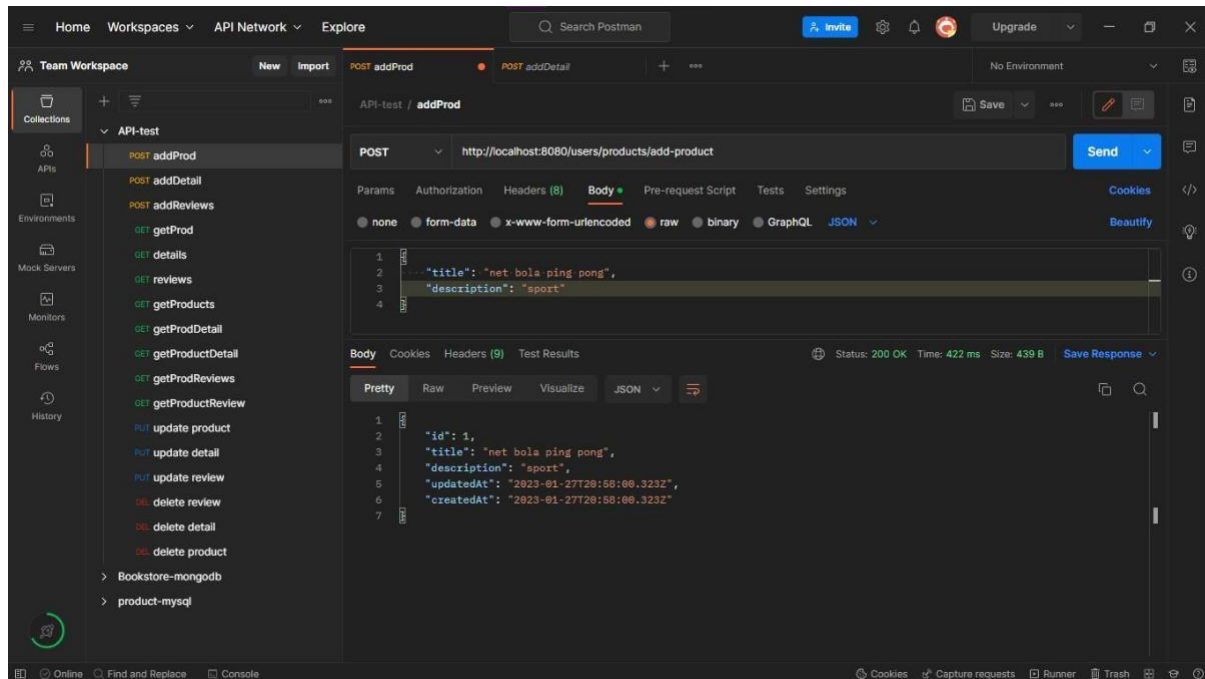
Model dari detail pada database. *File* ini berfungsi untuk menyimpan detail yang di masukan oleh user (penjual) yang ingin menjual product user. *Data* dari detail tersebut akan dihubungkan dengan data product yang telah dikirim ke *database*. Berikut adalah codingan dari *detailModel.js*.

```
models > JS detailModel.js > ...
1  module.exports = (sequelize, DataTypes) => {
2  ...
3  const Detail = sequelize.define("detail", {
4    detail: {
5      type: DataTypes.STRING,
6      allowNull: false
7    },
8    prod_id : {
9      type: DataTypes.INTEGER,
10     allowNull: false
11   }
12 });
13 return Detail;
14 };
```

Gambar Codingan *detailModel.js*

## Hasil Pembuatan Aplikasi Backend

Hasil dari aplikasi *backend* ini :  
*CRUD Product*



Gambar *Create Product* pada *Postman*

## DAFTAR PUSTAKA

- Putra, aditya Nova. 2020. *Laporan Kerja Praktek Pembuatan Website Company Profile Kantor Kepala Desa Keharuman Universitas 17 Agustus 1945 Surabaya*.
- Christian, Angel Eman. 2017. *LAPORAN KERJA PRAKTEK Pembangunan Backend Aplikasi Mobile untuk Registrasi Pelanggan (Backend) PT.Saranainsan Mudaselaras (PT. SIMS) Universitas Atma Jaya Yogyakarta*.
- Muhammad, Alif Nurrahman. 2022. *LAPORAN AKHIR MAGANG & STUDI INDEPENDEN BERSERTIFIKAT MACHINE LEARNING LEARNING PATH di Bangkit Academy 2022 by Google, GoTo, Traveloka PT Presentologics Universitas Persada Indonesia Y.A.I*.
- Selli, Mariko. 2019. *APLIKASI WEBSITE BERBASIS HTML DAN JAVASCRIPT UNTUK MENYELESAIKAN FUNGSI INTEGRAL PADA MATA KULIAH KALKULUS*. Universitas Indraprasta PGRI.
- Nasution. 2021. *IMPLEMENTASI MONGO DB, EXPRESS JS, REACT JS DAN NODE JS (MERN) PADA PENGEMBANGAN APLIKASI FORMULIR, KUIS, DAN SURVEI ONLINE*. Universitas Islam Indonesia.
- Faisal Fajar Nursaid<sup>1</sup>, Adam Hendra Brata<sup>2</sup>, Agi Putra Kharisma<sup>3</sup>. 2020. *Pengembangan Sistem Informasi Pengelolaan Persediaan Barang Dengan ReactJS Dan React Native Menggunakan Prototype (Studi Kasus : Toko Uda Fajri)*. Universitas Brawijaya.
- Wikipedia. 2023. "CodeIgniter". <https://id.wikipedia.org/wiki/CodeIgniter>, diakses pada 12 Januari 2023 pukul 19.32.
- Sequelize. 2023 "Sequelize". <https://sequelize.org/>, diakses pada 13 januari 2023 pukul 23.12.