**PROJECT TOKO KOMPUTER**

**(CRUD ADMIN)**

## Persiapan Library

1. Install library yang dibutuhkan yaitu express, body-parser, cors dan md5 dengan command npm i --save express body-parser cors md5
2. Cek pada Dependencies di file package.json, jika sudah ada pada list artinya sudah berhasil diinstall

"dependencies": {

"body-parser": "^1.19.1",

"cors": "^2.8.5",

"express": "^4.17.2",

"md5": "^2.3.0",

"mysql2": "^2.3.3",

"nodemon": "^2.0.15",

"sequelize": "^6.12.5"

}

1. Buat File index.js, dan tambahkan kode seperti berikut

| //import  const express = require('express');  const cors = require('cors');  //implementasi  const app = express();  app.use(cors());  //endpoint nanti ditambahkan di sini  //run server  app.listen(8080, () => {  console.log('server run on port 8080')  }) |
| --- |

1. Kemudian, buat folder bernama **routes**, dan tambahkan file **admin.js**
2. Tambahkan kode untuk mengimport library dan model seperti berikut

| //import library  const express = require('express');  const bodyParser = require('body-parser');  const md5 = require('md5');  //implementasi library  const app = express();  app.use(bodyParser.json());  app.use(bodyParser.urlencoded({extended: true}));  //import model  const model = require('../models/index');  const admin = model.admin |
| --- |

1. Tambahkan endpoint untuk menampilkan data admin

| //endpoint menampilkan semua data admin, method: GET, function: findAll()  app.get("/", (req,res) => {  admin.findAll()  .then(result => {  res.json({  admin : result  })  })  .catch(error => {  res.json({  message: error.message  })  })  }) |
| --- |

1. Tambahkan endpoint untuk menyimpan data admin baru

| //endpoint untuk menyimpan data admin, METHOD: POST, function: create  app.post("/", (req,res) => {  let data = {  name : req.body.name,  username : req.body.username,  password : md5(req.body.password)  }  admin.create(data)  .then(result => {  res.json({  message: "data has been inserted"  })  })  .catch(error => {  res.json({  message: error.message  })  })  }) |
| --- |

1. Tambahkan endpoint untuk mengubah data admin yang sudah ada sesuai id

| //endpoint mengupdate data admin, METHOD: PUT, function:update  app.put("/:id", (req,res) => {  let param = {  admin\_id : req.params.id  }  let data = {  name : req.body.name,  username : req.body.username,  password : md5(req.body.password)  }  admin.update(data, {where: param})  .then(result => {  res.json({  message: "data has been updated"  })  })  .catch(error => {  res.json({  message: error.message  })  })  }) |
| --- |

1. Tambahkan endpoint untuk menghapus data admin berdasarkan id

| //endpoint menghapus data admin, METHOD: DELETE, function: destroy  app.delete("/:id", (req,res) => {  let param = {  admin\_id : req.params.id  }  admin.destroy({where: param})  .then(result => {  res.json({  message: "data has been deleted"  })  })  .catch(error => {  res.json({  message: error.message  })  })  }) |
| --- |

1. Tambahkan export agar file dapat diimport ke file lain

| module.exports = app |
| --- |

1. Import file admin.js di index.js

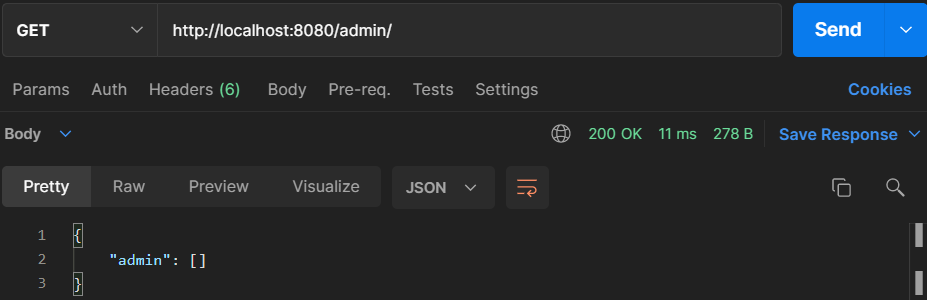
| //import  const express = require('express');  const cors = require('cors');  //implementasi  const app = express();  app.use(cors());    //endpoint admin  const admin = require('./routes/admin');  app.use("/admin", admin)  //run server  app.listen(8080, () => {  console.log('server run on port 8080')  }) |
| --- |

1. Jalankan perintah berikut pada terminal/cmd

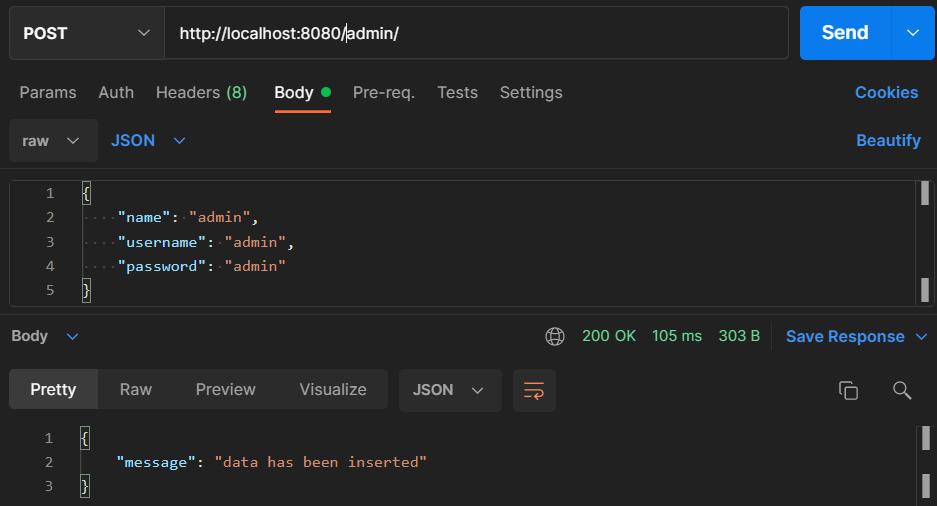
**npm start**

1. Cobalah hasil API pada postman

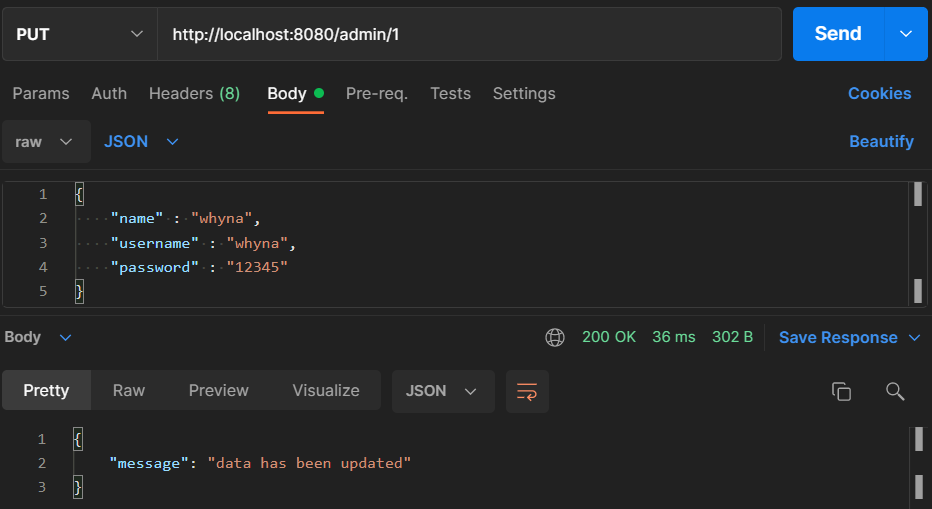
GET ALL ADMIN



POST ADMIN



PUT ADMIN



DELETE ADMIN

