

BE NodeJS

Tugas: MongoDB

Leonardho R. Sitanggang

Soal :

1. Buatlah fitur CRUD seperti Product untuk data Category
2. Pastikan data Product mempunyai relasi dengan Category, tutorial tentang relasi mongoose [bisa kamu baca di sini](#)
3. Pastikan semua endpoint CRUD data Category bisa digunakan

Jawab :

1. Source Code

```
20 router.route("/categories")
21   .get(categoriesController.findAll)
22   .post(categoriesController.create);
23
24 router.route("/categories/:id")
25   .get(categoriesController.findOne)
26   .put(categoriesController.update)
27   .delete(categoriesController.delete);
28
```

Gambar 1.1 Route Categories

```

export default {
  async create(req: Request, res: Response) {
    try {
      const result = await CategoriesModel.create(req.body);
      res.status(201).json({
        data: result,
        message: "Success create category",
      });
    } catch (error) {
      const err = error as Error;
      res.status(500).json({
        data: err.message,
        message: "Failed create category",
      });
    }
  },
  async findAll(req: Request, res: Response) {
    try {
      const result = await CategoriesModel.find().populate("products");
      res.status(200).json({
        data: result,
        message: "Success get all categorys",
      });
    } catch (error) {
      const err = error as Error;
      res.status(500).json({
        data: err.message,
        message: "Failed get all categorys",
      });
    }
  },
  async findOne(req: Request, res: Response) {
    try {
      const result = await CategoriesModel.findOne({
        _id: req.params.id,
      });
      res.status(200).json({
        data: result,
        message: "Success get one category",
      });
    } catch (error) {
      const err = error as Error;
      res.status(500).json({
        data: err.message,
        message: "Failed get one category",
      });
    }
  },
}

```

Gambar 1.2 Controller Create, Read

```

async update(req: Request, res: Response) {
  try {
    const result = await CategoriesModel.findOneAndUpdate(
      { _id: req.params.id },
      req.body,
      {
        new: true,
      }
    );

    res.status(200).json({
      data: result,
      message: "Success update category",
    });
  } catch (error) {
    const err = error as Error;
    res.status(500).json({
      data: err.message,
      message: "Failed update category",
    });
  }
},

async delete(req: Request, res: Response) {
  try {
    const result = await CategoriesModel.findOneAndDelete({
      _id: req.params.id,
    });

    res.status(200).json({
      data: result,
      message: "Success delete category",
    });
  } catch (error) {
    const err = error as Error;
    res.status(500).json({
      data: err.message,
      message: "Failed delete category",
    });
  }
},
};

```

Gambar 1.3 Controller Update, Delete

2. Source Code & DB

sanber-be-59.categories

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 203B TOTAL DOCUMENTS: 2 INDEXES TOTAL SIZE: 36KB

Find

Indexes

Schema Anti-Patterns 0

Aggregation

Search Indexes

Generate queries from natural language in Compass

Filter

Type a query: { field: 'value' }

QUERY RESULTS: 1-2 OF 2

_id: ObjectId('672e27a3c26057c576eab6ac')

name: "Plastic"

products: Array (empty)

createdAt: 2024-11-08T15:00:51.443+00:00

updatedAt: 2024-11-08T15:00:51.443+00:00

__v: 0

_id: ObjectId('672e28a6c26057c576eab6b1')

name: "Rubber"

products: Array (empty)

createdAt: 2024-11-08T15:05:10.790+00:00

updatedAt: 2024-11-08T15:05:10.790+00:00

__v: 0

sanber-be-59.products

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 180B TOTAL DOCUMENTS: 1 INDEXES TOTAL SIZE: 36KB

Find

Indexes

Schema Anti-Patterns 0

Aggregation

Search Indexes

Generate queries from natural language in Compass

Filter

Type a query: { field: 'value' }

QUERY RESULTS: 1-1 OF 1

_id: ObjectId('672e28a7c26057c576eab6b7')

name: "Car"

description: "AGP application AI"

images: Array (empty)

price: 779.97

qty: 228

categoryId: ObjectId('672e27a3c26057c576eab6ac')

createdAt: 2024-11-08T15:05:11.128+00:00

updatedAt: 2024-11-08T15:05:11.128+00:00

__v: 0

Gambar 2.1 Evidence Database MongoDB

```

const CategoriesSchema = new Schema(
  {
    name: {
      type: String,
      required: true,
    },
    products: [{
      type: mongoose.Schema.Types.ObjectId,
      ref: "Products",
    }],
  },
  {
    timestamps: true,
  }
);

const CategoriesModel = mongoose.model("Categories", CategoriesSchema);

export default CategoriesModel;

```

```

const ProductsSchema = new Schema(
  {
    name: {
      type: String,
      required: true,
    },
    description: {
      type: String,
      required: true,
    },
    images: {
      type: [String],
      required: true,
    },
    price: {
      type: Number,
      required: true,
    },
    qty: {
      type: Number,
      required: true,
    },
    categoryId: {
      type: mongoose.Schema.Types.ObjectId,
      ref: "Categories",
    },
  },
  {
    timestamps: true,
  }
);

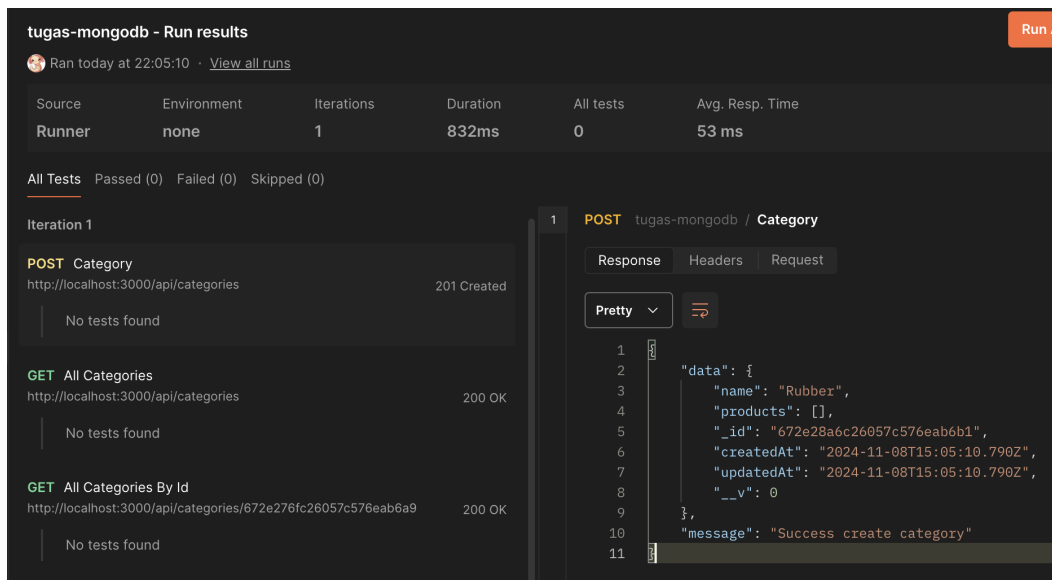
const ProductsModel = mongoose.model("Products", ProductsSchema);

export default ProductsModel;

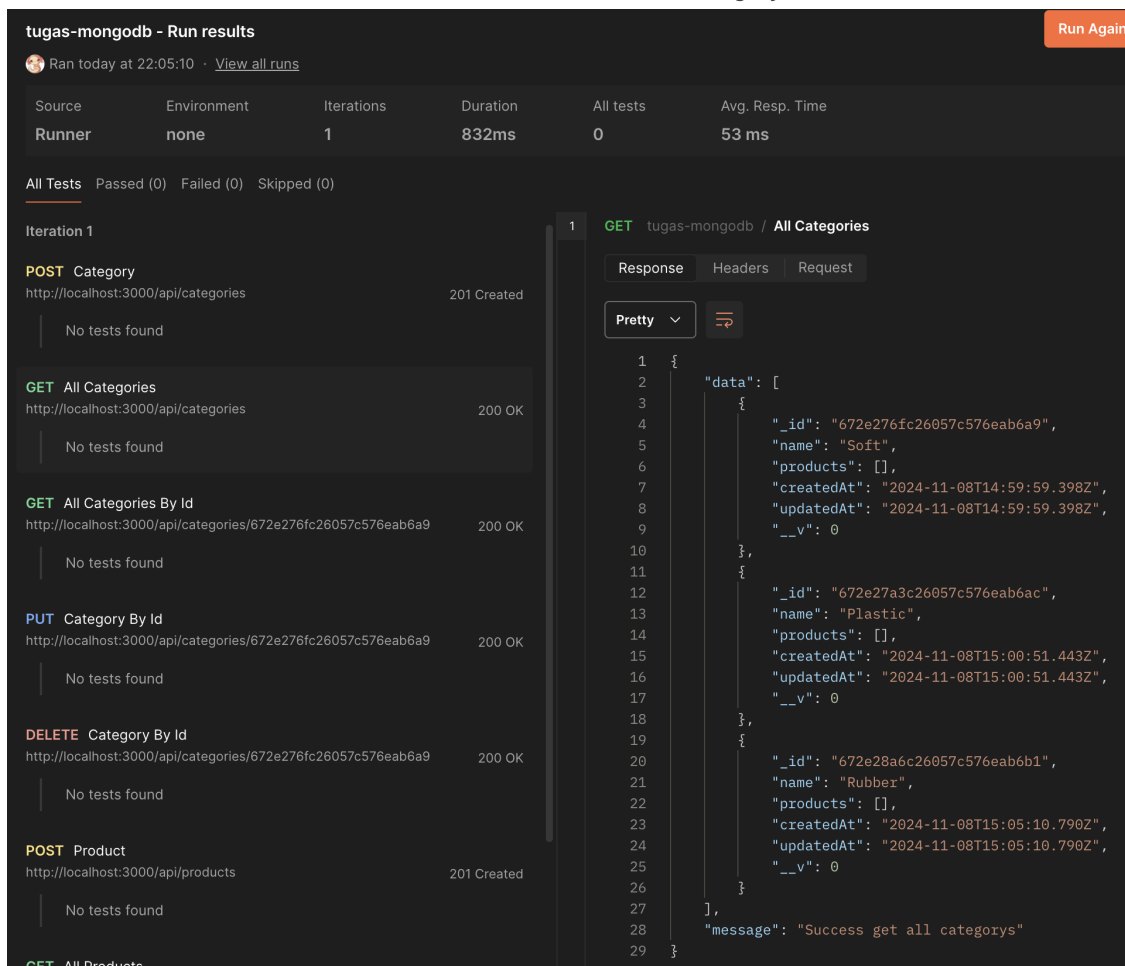
```

Gambar 2.2 Model Products & Categories

3. Test Endpoint Runner Postman



Gambar 3.1 Test Result Post Category



Gambar 3.2 Test Result Get All Categories

tugas-mongodb - Run results

Run today at 22:05:10 · [View all runs](#)

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	none	1	832ms	0	53 ms

All Tests Passed (0) Failed (0) Skipped (0)

Iteration 1

POST Category

http://localhost:3000/api/categories

201 Created

No tests found

GET All Categories

http://localhost:3000/api/categories

200 OK

No tests found

GET All Categories By Id

http://localhost:3000/api/categories/672e276fc26057c576eab6a9

200 OK

No tests found

1

GET tugas-mongodb / All Categories By Id

ResponseHeadersRequest

Pretty

```
1 {
2   "data": {
3     "_id": "672e276fc26057c576eab6a9",
4     "name": "Soft",
5     "products": [],
6     "createdAt": "2024-11-08T14:59:59.398Z",
7     "updatedAt": "2024-11-08T14:59:59.398Z",
8     "__v": 0
9   },
10  "message": "Success get one category"
11 }
```

Gambar 3.3 Test Result Get Categories By Id

tugas-mongodb - Run results

Run today at 22:05:10 · [View all runs](#)

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	none	1	832ms	0	53 ms

All Tests Passed (0) Failed (0) Skipped (0)

Iteration 1

POST Category

http://localhost:3000/api/categories

201 Created

No tests found

GET All Categories

http://localhost:3000/api/categories

200 OK

No tests found

GET All Categories By Id

http://localhost:3000/api/categories/672e276fc26057c576eab6a9

200 OK

No tests found

PUT Category By Id

http://localhost:3000/api/categories/672e276fc26057c576eab6a9

200 OK

No tests found

1

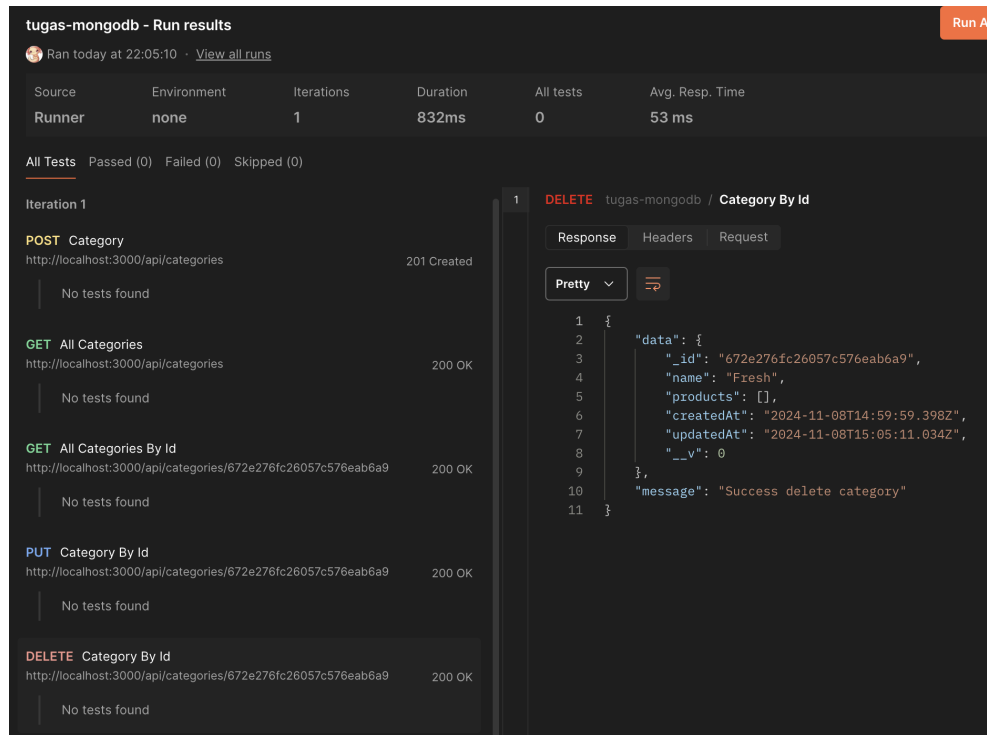
PUT tugas-mongodb / Category By Id

ResponseHeadersRequest

Pretty

```
1 {
2   "data": {
3     "_id": "672e276fc26057c576eab6a9",
4     "name": "Fresh",
5     "products": [],
6     "createdAt": "2024-11-08T14:59:59.398Z",
7     "updatedAt": "2024-11-08T15:05:11.034Z",
8     "__v": 0
9   },
10  "message": "Success update category"
11 }
```

Gambar 3.4 Test Result Put Categories By Id



Gambar 3.5 Test Result Get Categories By Id

Repository :

- Github <https://github.com/FlazeFy/sanbercode-nodejs-xpress/tree/main/tugas-mongodb>
- Postman Documentation <https://documenter.getpostman.com/view/18882129/2sAY52beJ8>

Refrensi :

- Materi Pekan 2.1 Framework Express JS
<https://sanbercode.com/bootcamp/class/684/materi/2-1-framework-expressjs/dc0618b7-fd52-4eec-a578-5ed88c4d5b3c>
- Materi Pekan 2.2 Routing Rest API
<https://sanbercode.com/bootcamp/class/684/materi/routing-rest-api/de70a7be-9056-40e0-bed0-c280c6bc1de0>
- Materi Pekan 2.3 Middleware Express
<https://sanbercode.com/bootcamp/class/684/materi/2-3-middleware-expressjs/c33672de-a33d-4916-b07c-9221b9e836b5>
- Materi Pekan 2.4 Database MongoDB
<https://sanbercode.com/bootcamp/class/684/materi/2-4-database-mongodb/ae319daf-6cf4-4282-8fe2-0c21c4da0d13>