



Wildan Ardana

1 / D4 TEKNIK INFORMATIKA A
3122600028

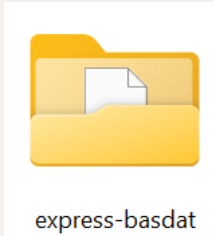


Project UAS

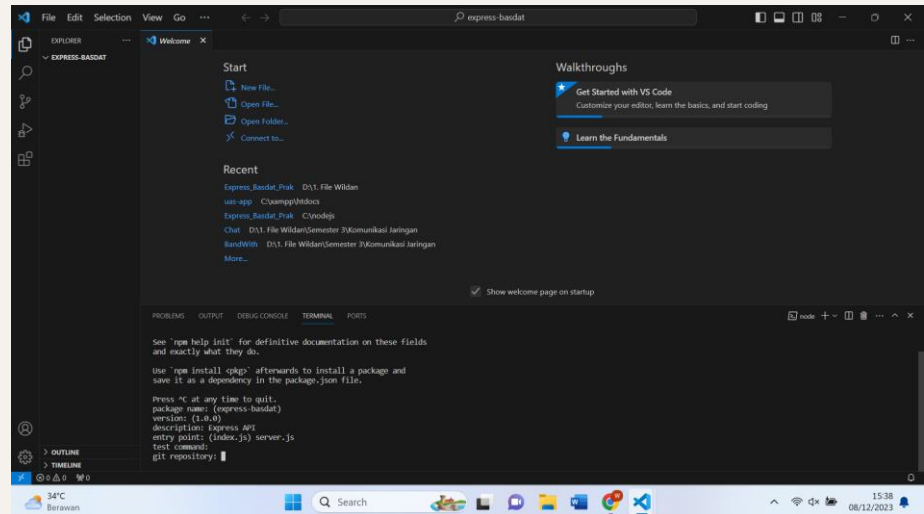
membuat API menggunakan node.js & mongoDB



- Buat Folder



- Buka VSCode Kemudian Buka Folder yang telah dibuat



Buka Terminal dan inputkan **npm init**

```
PS D:\express-basdat> npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.
```

```
See `npm help init` for definitive documentation on these fields
and exactly what they do.
```

```
Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.
```

Kemudian Isi

```
Press ^C at any time to quit.
package name: (express-basdat)
version: (1.0.0)
description: Express API
entry point: (index.js) server.js
test command:
git repository: github.com/Wildan85/Express_Basdat
keywords:
author: wildan
license: (ISC)
```

Kemudian Cek Apakah Sesuai, jika iya Ketik Yes

```
About to write to D:\express-basdat\package.json:
```

```
{
  "name": "express-basdat",
  "version": "1.0.0",
  "description": "Express API",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "repository": {
    "type": "git",
    "url": "github.com/Wildan85/Express_Basdat"
  },
  "author": "wildan",
  "license": "ISC"
}
```

```
Is this OK? (yes) yes
```

Lalu Ketik Perintah Berikut

```
PS D:\express-basdat> npm install express mongoose cors --save
added 86 packages, and audited 87 packages in 8s

12 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

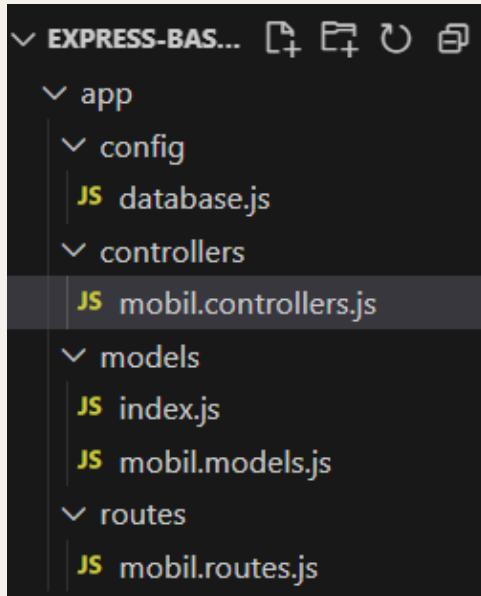
Jika muncul seperti dibawah ini
maka berhasil instalasi

```
added 86 packages, and audited 87 packages in 8s

12 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

Buat folder Bernama app beserta isinya seperti dibawah ini



Buat file baru Bernama server.js

JS server.js

Source Code database.js

```
app > config > JS database.js > [?] <unknown> > 🔑 url
```

```
1  module.exports = {  
2    url: "mongodb://localhost:27017/express_basdat"  
3  };
```

Source Code mobil.controllers.js Create dan findAll

```
app > controllers > JS mobil.controllers.js > update > update
1  const db = require('../models');
2  const Mobil = db.mobil
3
4  exports.create = (req, res) => {
5    req.body.tahun_mobil = new Number(req.body.tahun_mobil);
6
7    Mobil.create(req.body)
8      .then(() => {
9        res.send({message: "Data Mobil Berhasil Disimpan"});
10     }).catch((err) => {
11       res.status(500).send({
12         message: err.message || "Error Menginputkan Data Mobil."
13       });
14     });
15  }
16
17  exports.findAll = (req, res) => {
18    Mobil.find()
19      .then((result) => {
20        res.send(result);
21      }).catch((err) => {
22        res.status(500).send({
23          message: err.message || "Error Mendapatkan Data Mobil."
24        });
25      });
26  }
27
```


Source Code mobil.controllers.js show

```
app > controllers > JS mobil.controllers.js > update > update
28 exports.show = (req, res) => {
29   const id = req.params.id;
30
31   Mobil.findById(id)
32     .then((result) => {
33       if (!result) {
34         res.status(404).send({
35           message: "Mobil Tidak Ditemukan dengan Id " + id
36         });
37       }
38       res.send(result);
39     }).catch((err) => {
40       res.status(500).send({
41         message: err.message || "Error Mendapatkan Mobil dengan Id " + id
42       });
43     });
44 }
45
```

Source Code mobil.controllers.js Update & Delete

```
app > controllers > JS mobil.controllers.js > update > update
46 exports.update = (req, res) => {
47   const id = req.params.id;
48
49   req.body.tahun_mobil = new Number(req.body.tahun_mobil);
50
51   Mobil.findByIdAndUpdate(id, req.body, {useFindAndModify: false })
52   .then(data => {
53     if (!data) {
54       res.status(404).send({message: "Tidak dapat Mengupdate data"})
55     }
56     res.send({ message: "Data Mobil Berhasil Diupdate" });
57   })
58 }
59
60 exports.delete = (req, res) => {
61   const id = req.params.id;
62
63   Mobil.findByIdAndDelete(id)
64   .then((result) => {
65     if (!result) {
66       res.status(404).send({
67         message: "Mobil Tidak Ditemukan Dengan Id " + id
68       });
69     }
70     res.send({ message: "Mobil Berhasil Dihapus" });
71   }).catch((err) => {
72     res.status(500).send({
73       message: err.message || "Gagal Menghapus Mobil dengan Id " + id
74     });
75   });
76 }
77 }
```

Source Code mobil.models.js

```
app > models > JS mobil.models.js > <unknown> > exports
1  module.exports = mongoose => {
2  ...
3
4      const schema = mongoose.Schema(
5          {
6              nama_mobil: String,
7              brand_mobil: String,
8              tahun_mobil: Number,
9              tipe_mobil: String,
10             asalnegara_mobil: String,
11         }, {
12             timestamp: true
13         }
14     );
15
16     schema.method("toJSON", function () {
17         const {__v, _id, ...object } = this.toObject();
18         object.id_mobil = _id;
19         return object;
20     });
21
22     return mongoose.model("mobil", schema);
23 }
```

Source Code index.js

```
app > models > JS index.js > ...
1  const dbConfig = require("../config/database");
2  const mongoose = require("mongoose");
3
4  module.exports = {
5      mongoose,
6      url: dbConfig.url,
7      mobil: require('./mobil.models.js')(mongoose)
8  }
9
```

Source Code mobil.routes.js

app > routes > JS mobil.routes.js > <unknown> > exports > mobil

```
1  module.exports = app => {  
2    const mobil = require("../controllers/mobil.controllers")  
3    const router = require("express").Router();  
4  
5    router.get("/", mobil.findAll);  
6    router.get("/:id", mobil.show);  
7    router.post("/", mobil.create);  
8    router.put("/:id", mobil.update);  
9    router.delete("/:id", mobil.delete);  
10  
11    app.use('/mobil', router);  
12  }
```

Source Code server.js

```
JS server.js > ...
1  const express = require("express");
2  const cors = require("cors");
3  const db = require("../app/models");
4  const app = express();
5
6  const corsOptions = {
7    origin: "*"
8  };
9
10 // Register cors middleware
11 app.use(cors(corsOptions));
12 app.use(express.json());
13
14 // Koneksi ke database
15
16 db.mongoose.connect(db.url)
17   .then(() => {
18     console.log("Connected to the database");
19   })
20   .catch((err) => {
21     console.error("Error connecting to the database:", err);
22   });
23
24 // Memanggil routes mobil
25 require("../app/routes/mobil.routes")(app);
26
27
28 const PORT = process.env.PORT || 8000;
29 app.listen(PORT, () => console.log(`Server started on port ${PORT}`));
30
```

Create Data

HTTP <http://localhost:8000/mobil>

POST <http://localhost:8000/mobil>

Params Authorization Headers (8) **Body** Pre-request Script

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary

```
1 {
2   "nama_mobil": "Hellcat",
3   "brand_mobil": "Dodge",
4   "tahun_mobil": "2016",
5   "tipe_mobil": "Sport Sedan",
6   "asalnegara_mobil": "Germany"
7 }
```

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Data Mobil Berhasil Disimpan"
3 }
```

express_basdat.mobil

Documents Aggregations Schema Indexes Validations

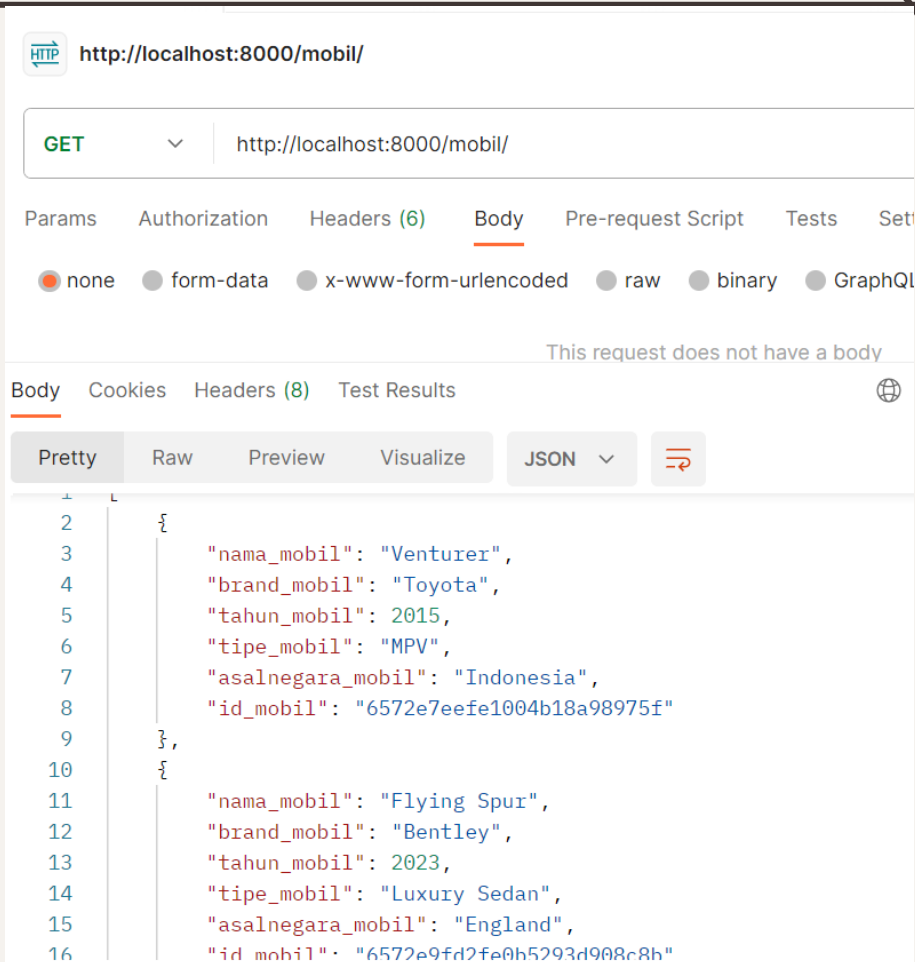
Filter Type a query: { field: 'value' } or [Get](#)

[+ ADD DATA](#) [EXPORT DATA](#)

```
brand_mobil: "Mercedes"
tahun_mobil: 2022
tipe_mobil: "Luxury Sedan"
asalnegara_mobil: "Germany"
__v: 0
```

```
_id: ObjectId('6572f38aa7ebdde1c236b1bd')
nama_mobil: "Hellcat"
brand_mobil: "Dodge"
tahun_mobil: 2016
tipe_mobil: "Sport Sedan"
asalnegara_mobil: "Germany"
__v: 0
```

findAll Data



HTTP **http://localhost:8000/mobil/**


GET **http://localhost:8000/mobil/**

Params Authorization Headers (6) **Body** Pre-request Script Tests Set

☒ none ☐ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

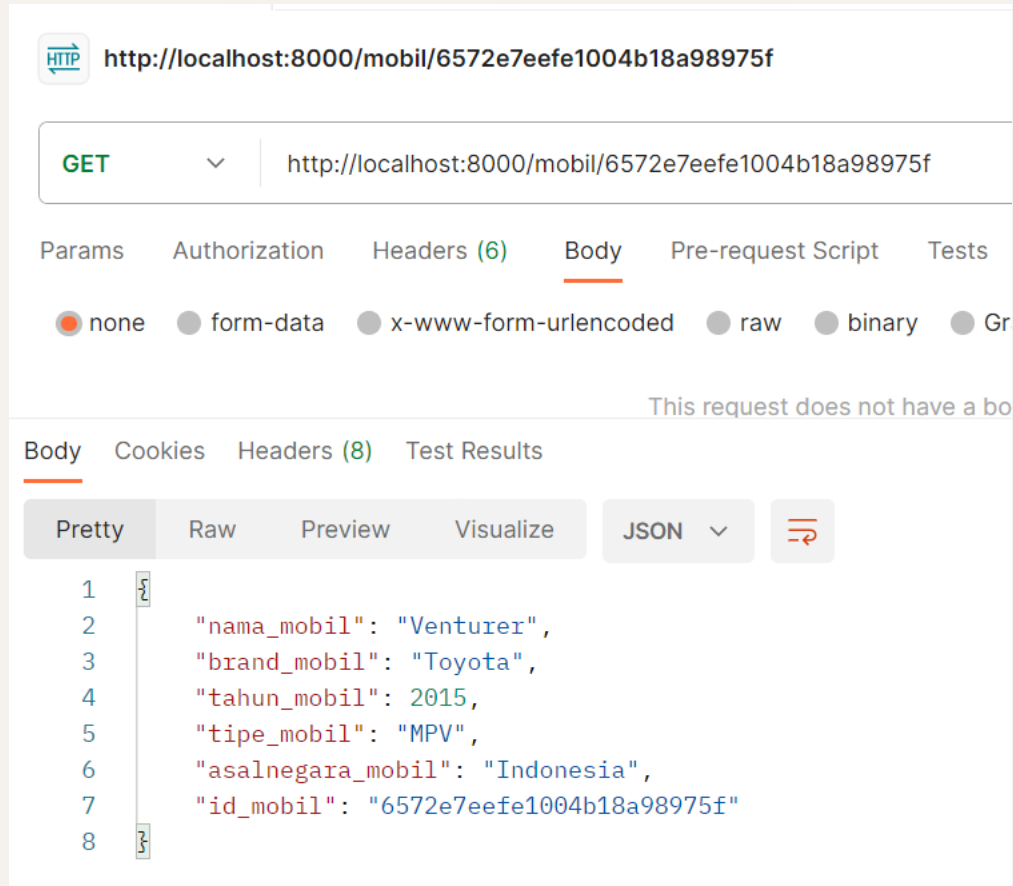
This request does not have a body

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON 

```
1 {
2   {
3     "nama_mobil": "Venturer",
4     "brand_mobil": "Toyota",
5     "tahun_mobil": 2015,
6     "tipe_mobil": "MPV",
7     "asalnegara_mobil": "Indonesia",
8     "id_mobil": "6572e7ee1004b18a98975f"
9   },
10  {
11    "nama_mobil": "Flying Spur",
12    "brand_mobil": "Bentley",
13    "tahun_mobil": 2023,
14    "tipe_mobil": "Luxury Sedan",
15    "asalnegara_mobil": "England",
16    "id_mobil": "6572e9fd2fe0b5293d908c8b"
```


Show Data by ID



The screenshot shows a web browser's developer tools interface. At the top, the URL bar displays `http://localhost:8000/mobil/6572e7eefe1004b18a98975f`. Below the URL bar, the **GET** method is selected. The **Body** tab is active, showing the response data in JSON format. The JSON data is as follows:

```
{
  "nama_mobil": "Venturer",
  "brand_mobil": "Toyota",
  "tahun_mobil": 2015,
  "tipe_mobil": "MPV",
  "asalnegara_mobil": "Indonesia",
  "id_mobil": "6572e7eefe1004b18a98975f"
}
```

Update Data

HTTP <http://localhost:8000/mobil/6572e7eefe1004b18a98975f>

PUT <http://localhost:8000/mobil/6572e7eefe1004b18a98975f>

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL

```
1 {
2   ... "nama_mobil": "Venturer Diesel",
3   ... "brand_mobil": "Toyota",
4   ... "tahun_mobil": 2017,
5   ... "tipe_mobil": "MPV",
6   ... "asalnegara_mobil": "Indonesia",
7   ... "id_mobil": "6572e7eefe1004b18a98975f"
8 }
```

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Data Mobil Berhasil Diupdate"
3 }
```

```
_id: ObjectId('6572e7eefe1004b18a98975f')
nama_mobil: "Venturer Diesel"
brand_mobil: "Toyota"
tahun_mobil: 2017
tipe_mobil: "MPV"
asalnegara_mobil: "Indonesia"
__v: 0
```

Delete Data

Data Yang Dihapus

```
_id: ObjectId('6572f38aa7ebdde1c236b1bd')
nama_mobil: "Hellcat"
brand_mobil: "Dodge"
tahun_mobil: 2016
tipe_mobil: "Sport Sedan"
asalnegara_mobil: "Germany"
__v: 0
```

The screenshot shows a REST client interface with a DELETE request to `http://localhost:8000/mobil/6572f38aa7ebdde1c236b1bd`. The response is a 200 OK status with a JSON body: `{"message": "Mobil Berhasil Dihapus"}`.

DELETE `http://localhost:8000/mobil/6572f38aa7ebdde1c236b1bd`

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies Headers (8) Test Results 200 OK

Pretty Raw Preview Visualize JSON `↻`

```
1 {
2   "message": "Mobil Berhasil Dihapus"
3 }
```

**SEKIAN,
TERIMA KASIH**

Dosen Pembimbing
Tessy Badriyah S.Kom.,M.T., Ph.D.