Praktikum
Basis Data Lanjut

UAS

Saiq Syahru Qadri - 3122600029

01

Persiapkan Aplikasi

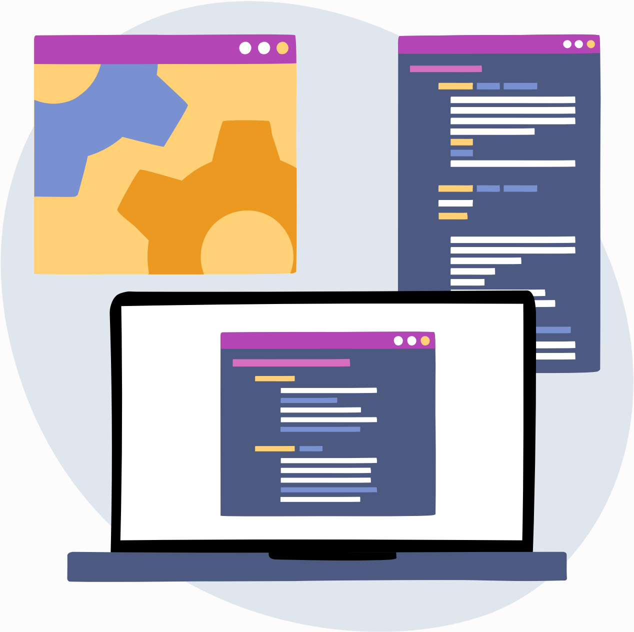
Persiapkan Aplikasi

Aplikasi yang digunakan ada tiga, yaitu:

- Virtual Studio Code
- MongoDB
- Node.js

02

Membuat Folder dan File



Membuat Folder dan File

1. Buat folder ‘mahasiswa’
2. Jalankan ‘npm init’
3. Jalankan ‘install express mongoose cors’
4. Buat folder ‘app’
5. Buat file ‘server.js’

1. Buat folder ‘mahasiswa’

mahasiswa

2. Jalankan ‘npm init’

package.json

```
package name: (mahasiswa)
version: (1.0.0)
description: Mahasiswa
entry point: (index.js) server.js
test command:
git repository: github.com/xsaiq/mahasiswa
keywords:
author: Saiq
license: (ISC)
```

3. Jalankan ‘install express mongoose cors’

Node_module – package-lock.json

```
npm install express mongoose cors -save
```

yang berisi folder config, controller, models, routes

4. Buat folder ‘app’

app

```
Buat 4 folder dalam folder app
```

config – controllers – models – routes

4. Buat file ‘server.js’

server.js

```
Code berada di slide selanjutnya
```

03

Source Code

1. server.js

```
const express = require("express")
const cors = require("cors")
const db = require("./app/models")
const app = express();

const corsOption = {
    origin: "*"
};
```

1. server.js

//register cors middleware

```
app.use(cors(corsOption));
app.use(express.json());
```

1. server.js

//konek ke database

```
const mongooseConfig = {
    useNewUrlParser: true,
    useUnifiedTopology: true,
}

db.mongoose.connect(db.url, mongooseConfig)
    .then(() => console.log("database connected"))
    .catch(enn => {
        console.log(`gagal konek ${enn.message}`);
        process.exit();
    });
```

1. server.js

//memanggil routes mahasiswa

```
require("./app/routes/mahasiswa.routes")(app);

const PORT = process.env.PORT || 3000;
app.listen(PORT, () => console.log(`server started on port ${PORT}`));
```

2. package.json

//tambahkan

```
"dev": "nodemon server.js",
```

3. config/database.js

```
module.exports = {
    url: "mongodb://localhost:27017/mahasiswa"
};
```

4. controllers/mahasiswa.controller.js

```
const db = require("../models")
const Mahasiswa = db.mahasiswa
```

4. controllers/mahasiswa.controller.js

//create

```
exports.create = (req, res) => {

    req.body.tanggal_lahir = new Date(req.body.tanggal_lahir);

    Mahasiswa.create(req.body)
        .then(() => res.send({message: "Data berhasil disimpan"}))
        .catch(err => res.status(500).send({message: err.message}));
}
```

4. controllers/mahasiswa.controller.js

//findAll

```
exports.findAll = (req, res) => {
    Mahasiswa.find()
        .then(data => res.send(data))
        .catch(err => res.status(500).send({message: err.message}))
}
```

4. controllers/mahasiswa.controller.js

//show

```
exports.show = (req, res) => {
    const id = req.params.id;

    Mahasiswa.findById(id)
        .then(data => res.send(data))
        .catch(err => res.status(500).send({message: err.message}))
}
```

4. controllers/mahasiswa.controller.js

//update

```
exports.update = (req, res) => {
    const id = req.params.id;

    req.body.tanggal_lahir = new Date(req.body.tanggal_lahir)

    Mahasiswa.findByIdAndUpdate(id, req.body, {useFindAndModify: false})
        .then(data => {
            if (!data) {
                res.status(404).send({message: "Tidak dapat mengupdate data"})
            }
            res.send({message: "Data berhasil diupdate"})
        })
        .catch(err => res.status(500).send({message: err.message}))
}
```

4. controllers/mahasiswa.controller.js

//delete

```
exports.delete = (req, res) => {
    const id = req.params.id;

    Mahasiswa.findByIdAndDelete(id)
        .then(data => {
            if (!data) {
                res.status(404).send({message: "Tidak dapat menghapus data"})
            }

            res.send({message: "Data berhasil dihapus"})
        })
        .catch(err => res.status(500).send({message: err.message}))
}
```

5. models/index.js

```
const dbconfig = require("../config/database");
const mongoose = require("mongoose");

module.exports = {
    mongoose,
    url: dbconfig.url,
    mahasiswa: require('./mahasiswa.model.js')(mongoose)
}
```

6. models/mahasiswa.model.js

```
module.exports = mongoose => {

    const schema = mongoose.Schema(
        {
            nama: String,
            nrp: String,
            tanggal_lahir: Date,

        }, {
            timestamps: true
        }
    );

    schema.method("toJSON", function() {
        const {__v, _id, ...object} = this.toObject();
        object.id = _id;

        return object;
    })

    return mongoose.model("mahasiswa", schema);

}
```

7. routes/mahasiswa.routes.js

```
module.exports = app => {
    const mahasiswa = require("../controllers/mahasiswa.controller")
    const r = require("express").Router();

    r.get("/", mahasiswa.findAll);
    r.get("/:id", mahasiswa.show);
    r.post("/", mahasiswa.create);
    r.put("/:id", mahasiswa.update);
    r.delete("/:id", mahasiswa.delete);

    app.use("/mahasiswa", r);
}
```

THANK YOU!