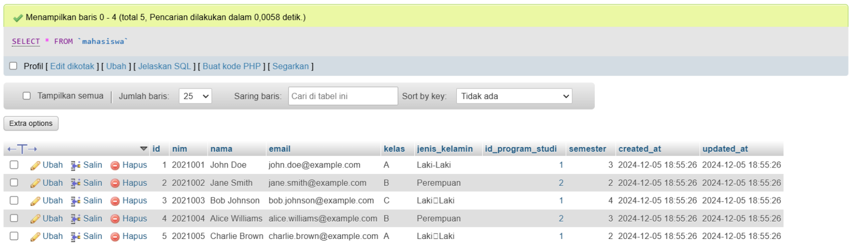
Aktivitas 4 - m11

DataBase

1. Mahasiswa



✓ Menampilkan baris 0 - 4 (total 5, Pencarian dilakukan dalam 0,0058 detik.)

```
SELECT * FROM `mahasiswa`
```

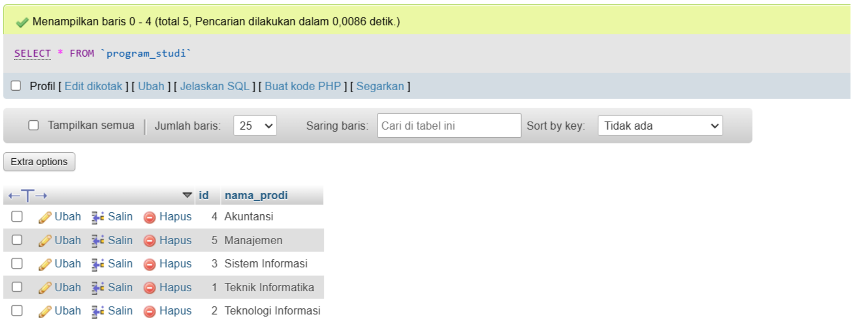
□ Profil [Edit dikotak] [Ubah] [Jelaskan SQL] [Buat kode PHP] [Segarkan]

□ Tampilkan semua | Jumlah baris: 25 Saring baris: Cari di tabel ini Sort by key: Tidak ada

Extra options

←T→			id	nim	nama	email	kelas	jenis_kelamin	id_program_studi	semester	created_at	updated_at
□	Ubah	Salin	Hapus	1	2021001	John Doe	john.doe@example.com	A	Laki-Laki	1	3	2024-12-05 18:55:26	2024-12-05 18:55:26
□	Ubah	Salin	Hapus	2	2021002	Jane Smith	jane.smith@example.com	B	Perempuan	2	2	2024-12-05 18:55:26	2024-12-05 18:55:26
□	Ubah	Salin	Hapus	3	2021003	Bob Johnson	bob.johnson@example.com	C	Laki□Laki	1	4	2024-12-05 18:55:26	2024-12-05 18:55:26
□	Ubah	Salin	Hapus	4	2021004	Alice Williams	alice.williams@example.com	B	Perempuan	2	3	2024-12-05 18:55:26	2024-12-05 18:55:26
□	Ubah	Salin	Hapus	5	2021005	Charlie Brown	charlie.brown@example.com	A	Laki□Laki	1	2	2024-12-05 18:55:26	2024-12-05 18:55:26

2. Program studi



✓ Menampilkan baris 0 - 4 (total 5, Pencarian dilakukan dalam 0,0086 detik.)

```
SELECT * FROM `program_studi`
```

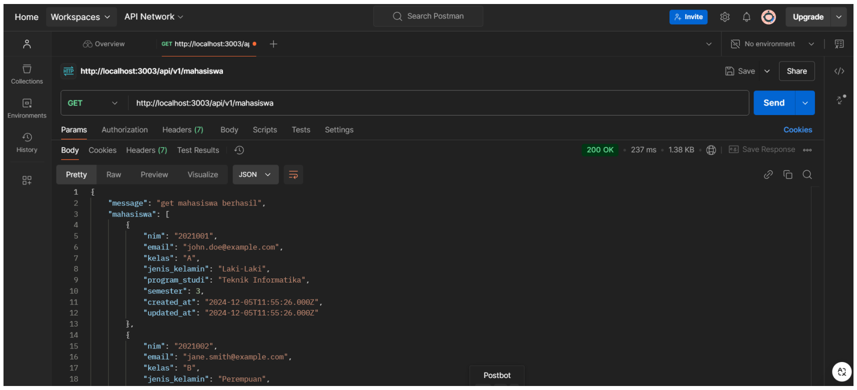
□ Profil [Edit dikotak] [Ubah] [Jelaskan SQL] [Buat kode PHP] [Segarkan]

□ Tampilkan semua | Jumlah baris: 25 Saring baris: Cari di tabel ini Sort by key: Tidak ada

Extra options

←T→				id	nama_prodi
□	Ubah	Salin	Hapus	4	Akuntansi
□	Ubah	Salin	Hapus	5	Manajemen
□	Ubah	Salin	Hapus	3	Sistem Informasi
□	Ubah	Salin	Hapus	1	Teknik Informatika
□	Ubah	Salin	Hapus	2	Teknologi Informasi

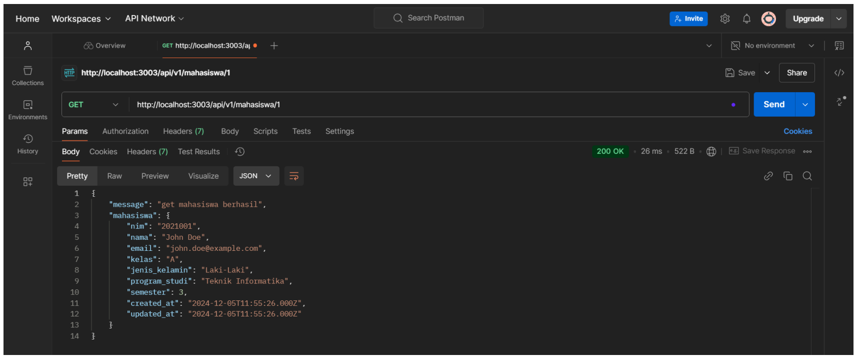
Hasil Code



GET http://localhost:3003/api/v1/mahasiswa

Body — 200 OK • 237 ms • 1.38 KB

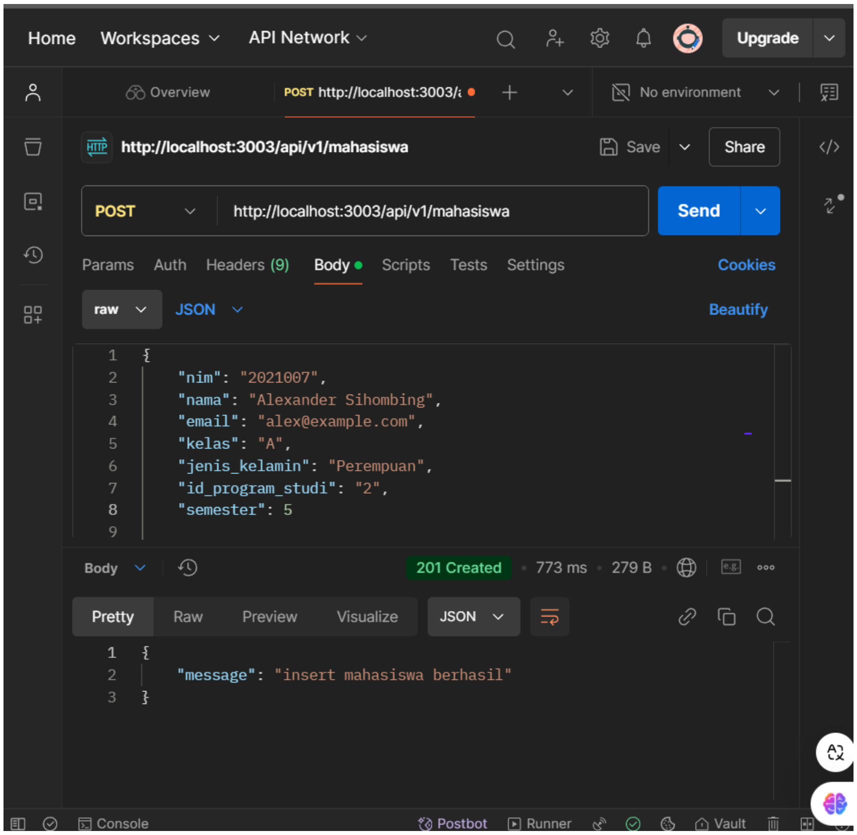
```
{
    "message": "get mahasiswa berhasil",
    "mahasiswa": [
        {
            "nim": "2021001",
            "email": "john.doe@example.com",
            "kelas": "A",
            "jenis_kelamin": "Laki-Laki",
            "program_studi": "Teknik Informatika",
            "semester": 3,
            "created_at": "2024-12-05T11:55:26.000Z",
            "updated_at": "2024-12-05T11:55:26.000Z"
        },
        {
            "nim": "2021002",
            "email": "jane.smith@example.com",
            "kelas": "B",
            "jenis_kelamin": "Perempuan",
```



GET http://localhost:3003/api/v1/mahasiswa/1

200 OK • 26 ms • 522 B

```
{
    "message": "get mahasiswa berhasil",
    "mahasiswa": {
        "nim": "2021001",
        "nama": "John Doe",
        "email": "john.doe@example.com",
        "kelas": "A",
        "jenis_kelamin": "Laki-Laki",
        "program_studi": "Teknik Informatika",
        "semester": 3,
        "created_at": "2024-12-05T11:55:26.000Z",
        "updated_at": "2024-12-05T11:55:26.000Z"
    }
}
```



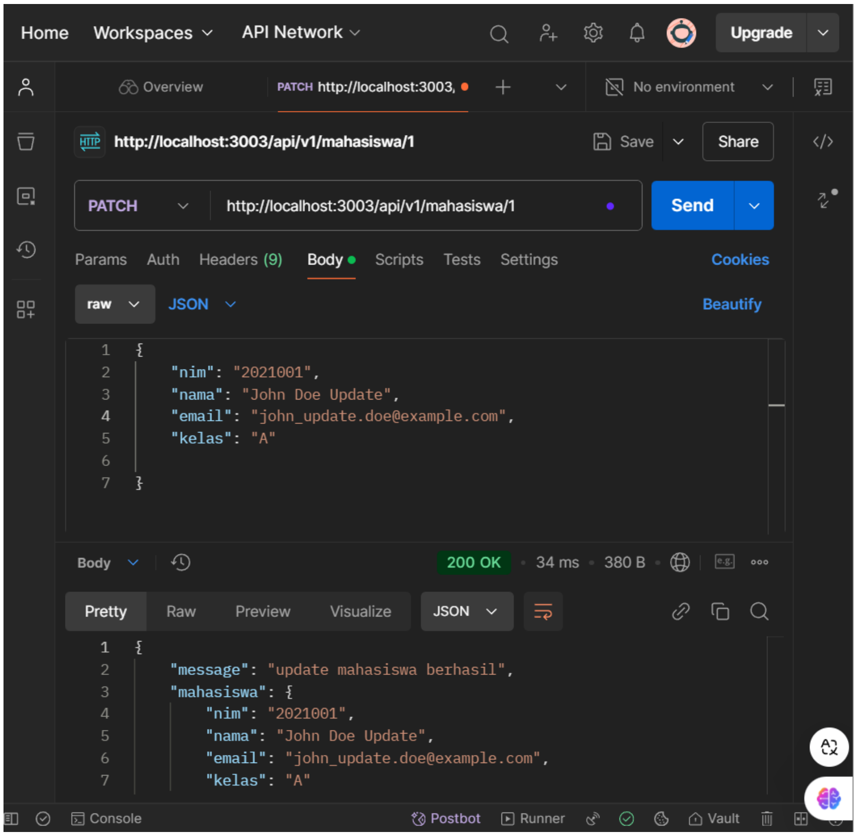
POST http://localhost:3003/api/v1/mahasiswa

```
{
    "nim": "2021007",
    "nama": "Alexander Sihombing",
    "email": "alex@example.com",
    "kelas": "A",
    "jenis_kelamin": "Perempuan",
    "id_program_studi": "2",
    "semester": 5

```

201 Created • 773 ms • 279 B

```
{
    "message": "insert mahasiswa berhasil"
}
```



http://localhost:3003/api/v1/mahasiswa/1

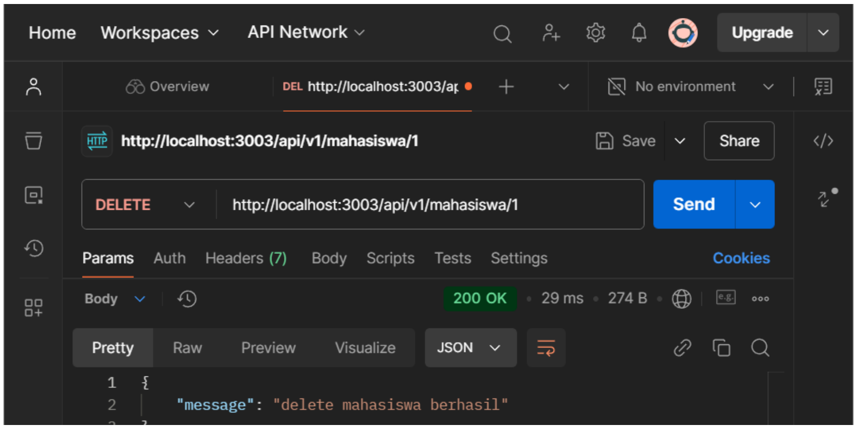
PATCH http://localhost:3003/api/v1/mahasiswa/1

```
{
    "nim": "2021001",
    "nama": "John Doe Update",
    "email": "john_update.doe@example.com",
    "kelas": "A"

}
```

200 OK • 34 ms • 380 B

```
{
    "message": "update mahasiswa berhasil",
    "mahasiswa": {
        "nim": "2021001",
        "nama": "John Doe Update",
        "email": "john_update.doe@example.com",
        "kelas": "A"
```



http://localhost:3003/api/v1/mahasiswa/1

DELETE http://localhost:3003/api/v1/mahasiswa/1

200 OK • 29 ms • 274 B

```
{
    "message": "delete mahasiswa berhasil"
```

Tugas

1. Silahkan tambahkan operasi crud pada entitas program_studi

prodiConstrollers.js

```
const db = require("./database")

const getAll = (req, res) => {
    const query = "SELECT * FROM `program_studi`"
    db.query(query, [], (err, result) => {
        if (err) {
            res.status(400).json({
                massage: "prodi gagal get",
                serverMassage: err
            })
        } else {
            res.status(200).json({
                massage: "prodi berhasil get",
                ProgramStudi: result
            })
        }
    })
}

const getOne = (req, res) => {
    const id = req.params.id
    const query = "SELECT * FROM `program_studi` WHERE id = ?;"
    db.query(query, [id], (err, result) => {
        if (err) {
            res.status(400).json({
                massage: "salah satu get prodi gagal",
                serverMassage: err
            })
        } else {
            res.status(200).json({
                massage: "salah satu get prodi berhasil",
                ProgramStudi: result[0]
            })
        }
    })
}

const insert = (req, res) => {
    const prodi = req.body
    const query = "INSERT INTO program_studi SET ?"
    db.query(query, prodi, (err, result) => {
        if (err) {
            res.status(400).json({
                massage: "insert prodi gagal",
                servisMassage: err
            })
        } else {
            res.status(200).json({
                massage: "insert prodi berhasil",
            })
        }
    })
}

const update = (req, res) => {
    const id = req.params.id
    const prodi = req.body
    const query = `UPDATE mahasiswa SET ? WHERE id = ${id}`
    db.query(query, prodi, (err, result) => {
        if(err) {
            res.status(400).json({
                massage: "update prodi gagal",
                serverMassage: err
            })
        } else {
            res.status(200).json({
                massage: "update prodi berhasil",
                ProgramStudi: prodi
            })
        }
    })
}

const remove = (req, res) => {
    const id = req.params.id
    const query = "DELETE FROM mahasiswa WHERE id =?"
    db.query(query, [id], (err, result) => {
        if(err) {
            res.status(400).json({
                massage: "delete gagal",
                serverMassage: err
            })
        } else {
            res.status(200).json({
                massage: "delete berhasil"
            })
        }
    })
}

module.exports = {getAll, insert, update, remove, getOne}
```

program_studi.js

```
const express = require("express");
const route = express.Router();

const prodiControllers = require("./prodiControllers")

route.get("/", prodiControllers.getAll);

route.get("/:id", prodiControllers.getOne);

route.post("/", prodiControllers.insert);

route.patch("/:id", prodiControllers.update);

route.delete("/:id", prodiControllers.remove);

module.exports = route;
```

app.js

```
const express = require("express");
const app = express();

const mahasiswa = require("./mahasiswa");
const prodi = require("./program_studi")

app.use("/mahasiswa", mahasiswa);
app.use("/prodi", prodi)
```

```
module.exports = app;
```

2. SILAHKAN TAMBAHKAN ROUTE UNTUK MENAMPILKAN MAHASISWA BERDASARKAN:
a. Kelas
b. Semester
c. Program Studi

mahasiswaControllers.js

```
const getKelas = (req, res) => {
    const kelas = req.params.kelas
    const query = "SELECT * FROM mahasiswa INNER JOIN program_studi ON id_program_studi = program_studi.id
    db.query(query, [kelas], (err, result) => {
        if (err) {
            res.status(400).json({
                message: "get kelas gagal",
                serverMessage: err,
            });
        } else {
            res.status(200).json({
                message: "get kelas berhasil",
                kelas: result[0]
            });
        }
    })
}

const getSemester = (req, res) => {
    const semester = "SELECT * FROM mahasiswa INNER JOIN program_studi ON id_program_studi = program_studi.
    db.query(query, [semester], (err, result) => {
        if (err) {
            res.status(400).json({
                message: "get semester gagal",
                serverMessage: err,
            });
        } else {
            res.status(200).json({
                message: "get semester berhasil",
                semester: result[0]
            });
        }
    })
}

const getProgram = (req, res) => {
    const program = req.params.id_program_studi
    const query = "SELECT * FROM mahasiswa INNER JOIN program_studi ON id_program_studi = program_studi.id
db.query(query, [program], (err, result) => {
    if (err) {
        res.status(400).json({
            message: "get program studi gagal",
            serverMessage: err,
        });
    } else {
        res.status(200).json({
            message: "get program studi berhasil",
            program: result[0]
        });
    }
})
}
```

app.js

```
route.get("/:kelas", mahasiswaControllers.getKelas);
route.get("/:semester", mahasiswaControllers.getSemester);
route.get("/:program", mahasiswaControllers.getProgram);
```