LAPORAN UJIAN AKHIR SEMESTER

PENGEMBANGAN BACKEND

Dosen Pengampu :

M. Taufiq M.Kom

Disusun Oleh :

Amanda Melda Savitri / 3124101291

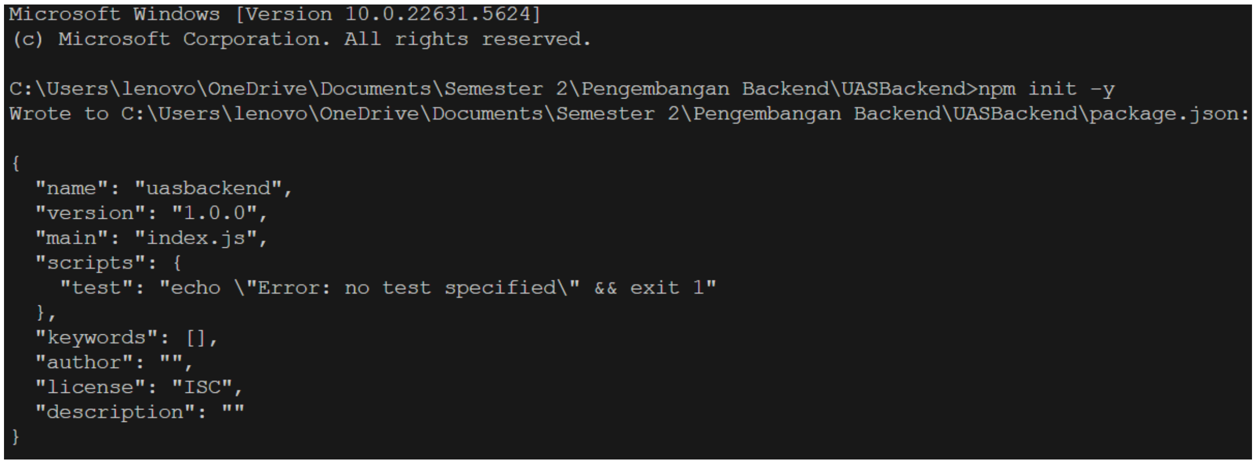
Putri Sahma Dewi / 3124101293

PRODI D3 MANAJEMEN INFORMATIKA

STIKOM PGRI BANYUWANGI

TAHUN 2025

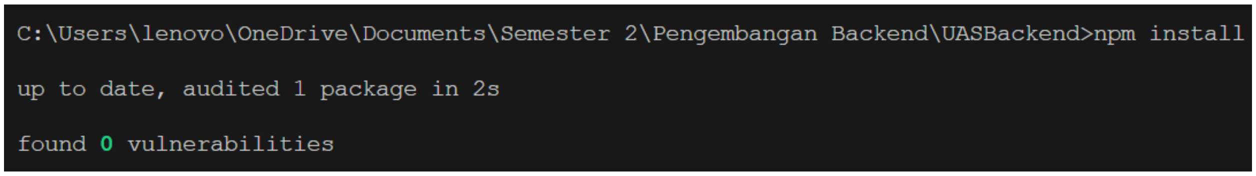
1. npm init -y



```
Microsoft Windows [Version 10.0.22631.5624]
(c) Microsoft Corporation. All rights reserved.

C:\Users\lenovo\OneDrive\Documents\Semester 2\Pengembangan Backend\UASBackend>npm init -y
Wrote to C:\Users\lenovo\OneDrive\Documents\Semester 2\Pengembangan Backend\UASBackend\package.json:

{
  "name": "uasbackend",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": ""
}
```

Penjelasan : Perintah ini digunakan untuk inisialisasi sebuah proyek Node.js dengan membuat file package.json secara otomatis.

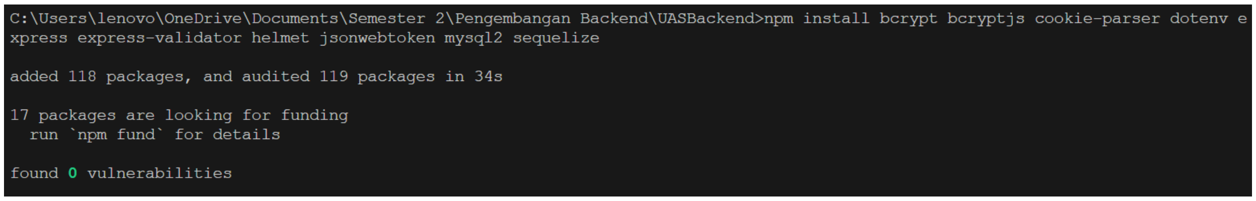
2. npm install



```
C:\Users\lenovo\OneDrive\Documents\Semester 2\Pengembangan Backend\UASBackend>npm install

up to date, audited 1 package in 2s

found 0 vulnerabilities
```

Penjelasan : Perintah ini digunakan untuk menginstal semua library atau package yang tercantum dalam file package.json.

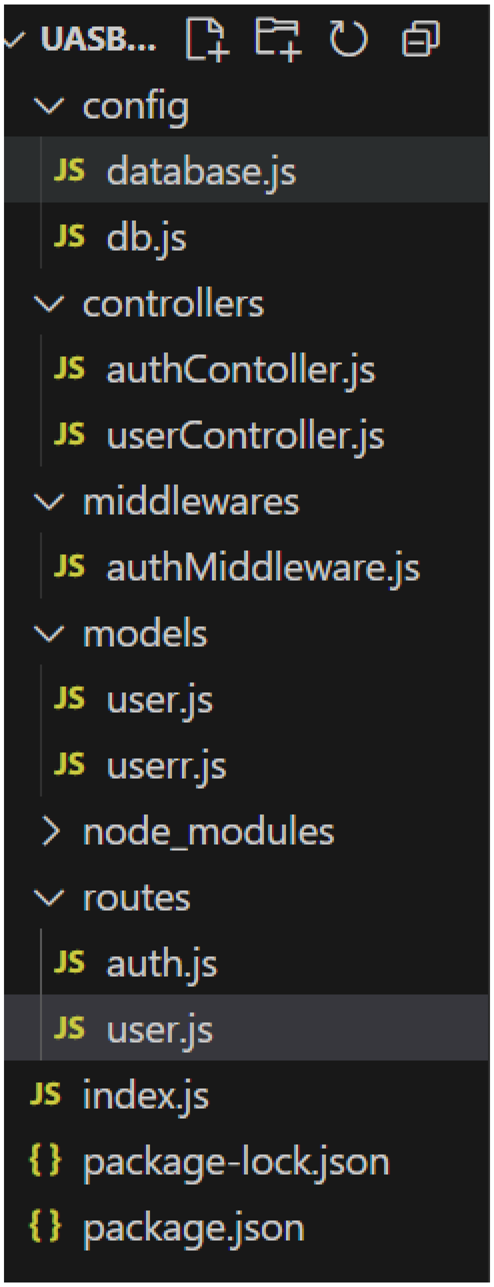
3. npm install bcrypt bcryptjs cookie-parser dotenv express express-validator helmet jsonwebtoken mysql2 sequelize



```
C:\Users\lenovo\OneDrive\Documents\Semester 2\Pengembangan Backend\UASBackend>npm install bcrypt bcryptjs cookie-parser dotenv e
xpress express-validator helmet jsonwebtoken mysql2 sequelize

added 118 packages, and audited 119 packages in 34s

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

found 0 vulnerabilities
```

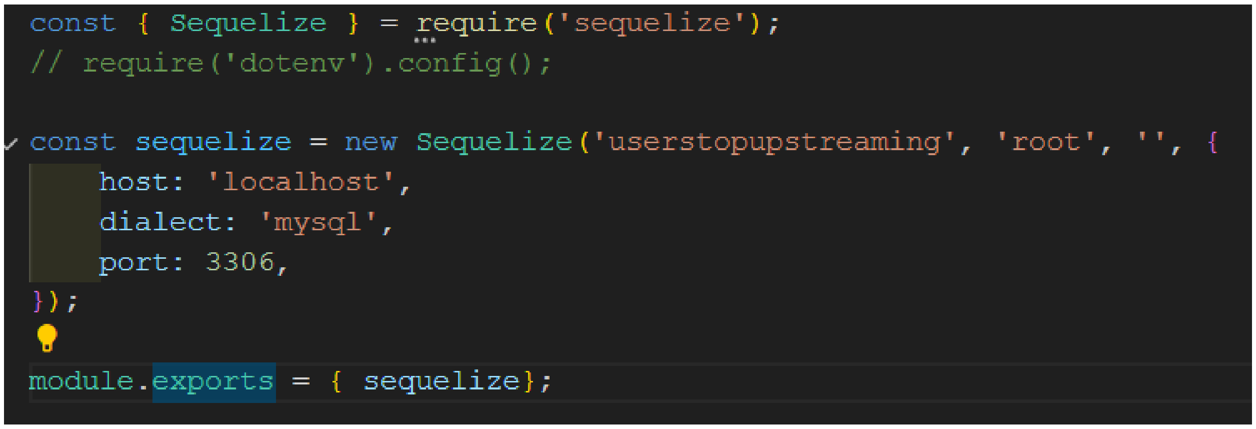
Perintah ini digunakan untuk menginstal package yang dibutuhkan dalam pengembangan backend

4. Membuat folder dan isi masing – masing folder



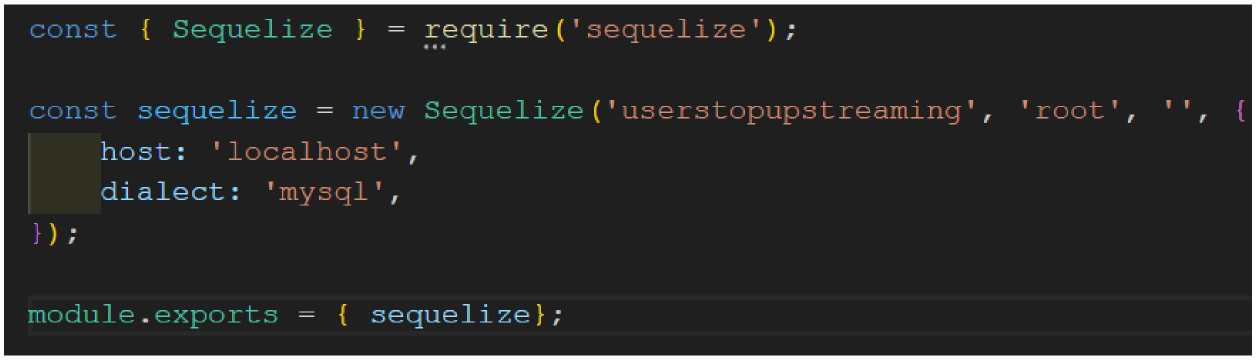
```
UASB...
config
  database.js
  db.js
controllers
  authContoller.js
  userController.js
middlewares
  authMiddleware.js
models
  user.js
  userr.js
node_modules
routes
  auth.js
  user.js
index.js
package-lock.json
package.json
```

5. Membuat isi file **database.js** pada folder config



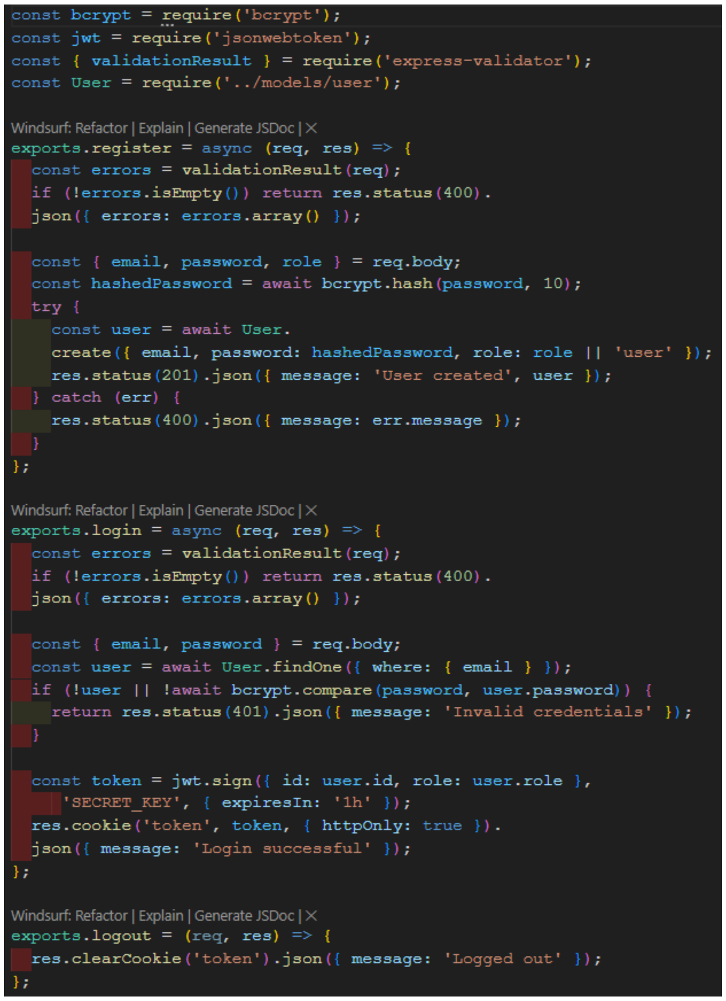
```
const { Sequelize } = require('sequelize');
// require('dotenv').config();

const sequelize = new Sequelize('userstopupstreaming', 'root', '', {
    host: 'localhost',
    dialect: 'mysql',
    port: 3306,
});

module.exports = { sequelize};
```

6. Membuat isi file **db.js** pada folder config



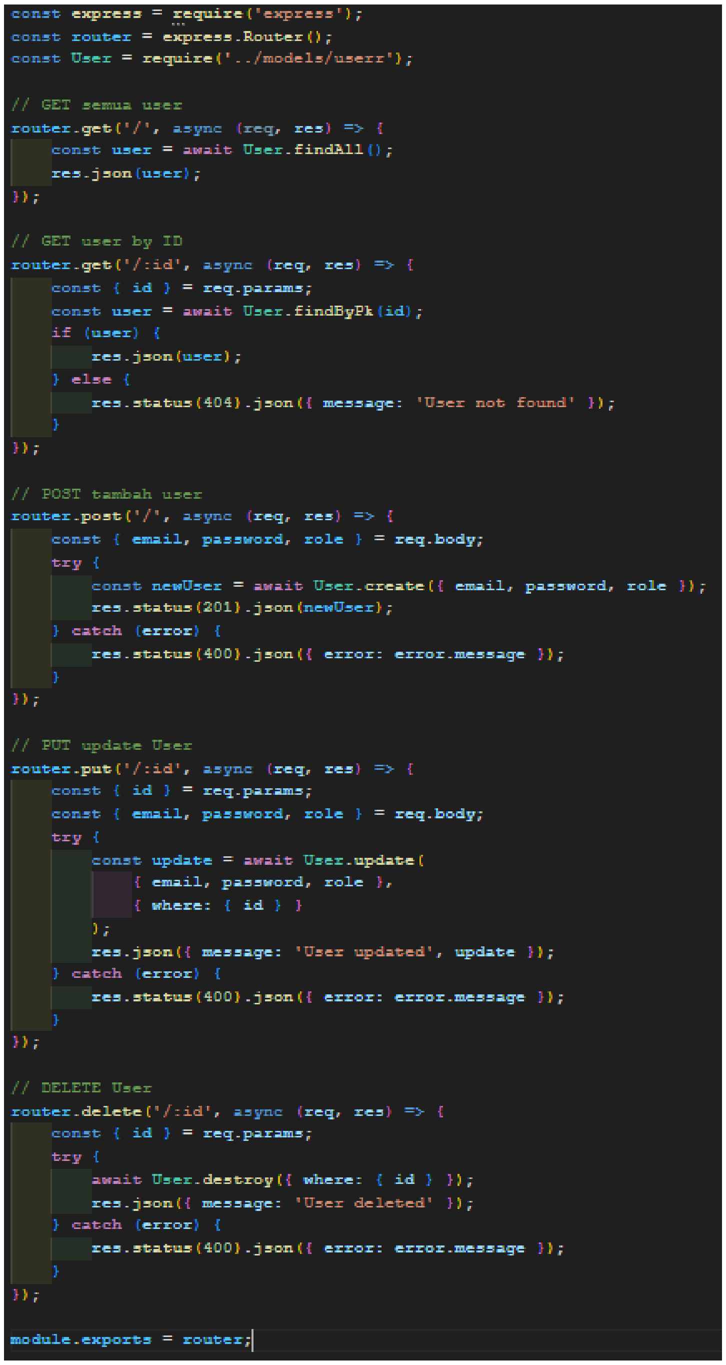
```
const { Sequelize } = require('sequelize');

const sequelize = new Sequelize('userstopupstreaming', 'root', '', {
    host: 'localhost',
    dialect: 'mysql',
});

module.exports = { sequelize};
```

7. Membuat isi file **authController.js** pada folder controllers



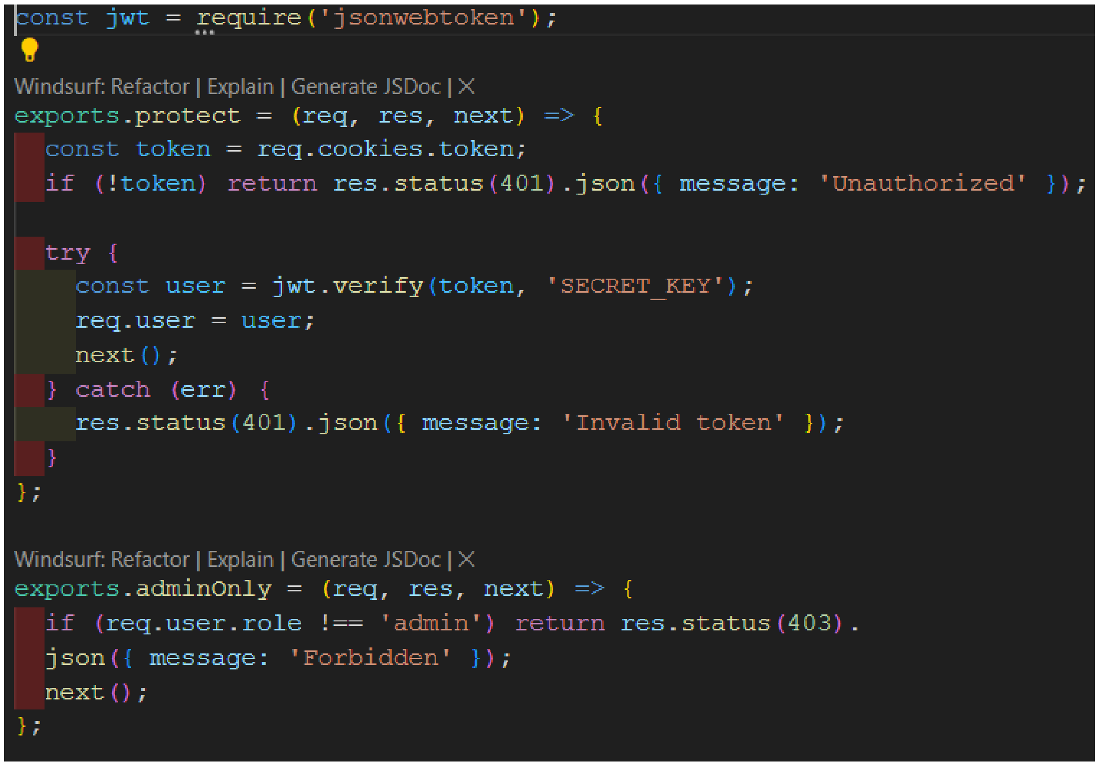
```
const bcrypt = require('bcrypt');
const jwt = require('jsonwebtoken');
const { validationResult } = require('express-validator');
const User = require('../models/user');

Windsurf: Refactor | Explain | Generate JSDoc | ✕
exports.register = async (req, res) => {
  const errors = validationResult(req);
  if (!errors.isEmpty()) return res.status(400).
  json({ errors: errors.array() });

  const { email, password, role } = req.body;
  const hashedPassword = await bcrypt.hash(password, 10);
  try {
    const user = await User.
    create({ email, password: hashedPassword, role: role || 'user' });
    res.status(201).json({ message: 'User created', user });
  } catch (err) {
    res.status(400).json({ message: err.message });
  }
};

Windsurf: Refactor | Explain | Generate JSDoc | ✕
exports.login = async (req, res) => {
  const errors = validationResult(req);
  if (!errors.isEmpty()) return res.status(400).
  json({ errors: errors.array() });

  const { email, password } = req.body;
  const user = await User.findOne({ where: { email } });
  if (!user || !await bcrypt.compare(password, user.password)) {
    return res.status(401).json({ message: 'Invalid credentials' });
  }

  const token = jwt.sign({ id: user.id, role: user.role },
     'SECRET_KEY', { expiresIn: '1h' });
  res.cookie('token', token, { httpOnly: true }).
  json({ message: 'Login successful' });
};

Windsurf: Refactor | Explain | Generate JSDoc | ✕
exports.logout = (req, res) => {
  res.clearCookie('token').json({ message: 'Logged out' });
};
```

8. Membuat isi file **userController.js** pada folder controller



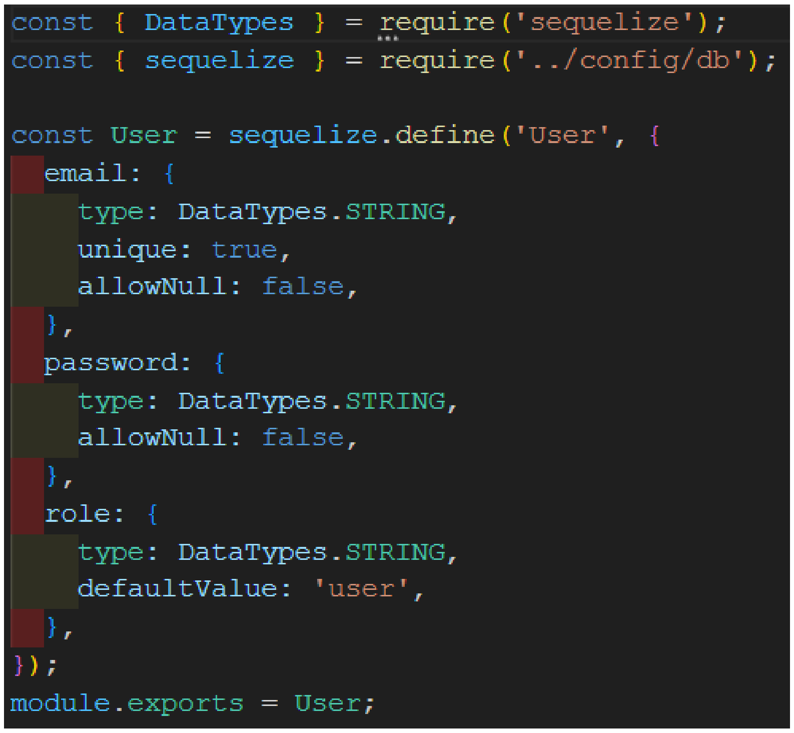
```javascript
const express = require('express');
const router = express.Router();
const User = require('../models/userr');

// GET semua user
router.get('/', async (req, res) => {
    const user = await User.findAll();
    res.json(user);
});

// GET user by ID
router.get('/:id', async (req, res) => {
    const { id } = req.params;
    const user = await User.findByPk(id);
    if (user) {
        res.json(user);
    } else {
        res.status(404).json({ message: 'User not found' });
    }
});

// POST tambah user
router.post('/', async (req, res) => {
    const { email, password, role } = req.body;
    try {
        const newUser = await User.create({ email, password, role });
        res.status(201).json(newUser);
    } catch (error) {
        res.status(400).json({ error: error.message });
    }
});

// PUT update User
router.put('/:id', async (req, res) => {
    const { id } = req.params;
    const { email, password, role } = req.body;
    try {
        const update = await User.update(
            { email, password, role },
            { where: { id } }
        );
        res.json({ message: 'User updated', update });
    } catch (error) {
        res.status(400).json({ error: error.message });
    }
});

// DELETE User
router.delete('/:id', async (req, res) => {
    const { id } = req.params;
    try {
        await User.destroy({ where: { id } });
        res.json({ message: 'User deleted' });
    } catch (error) {
        res.status(400).json({ error: error.message });
    }
});

module.exports = router;
```

9. Membuat isi file **authMiddleware.js** pada folder middlewares



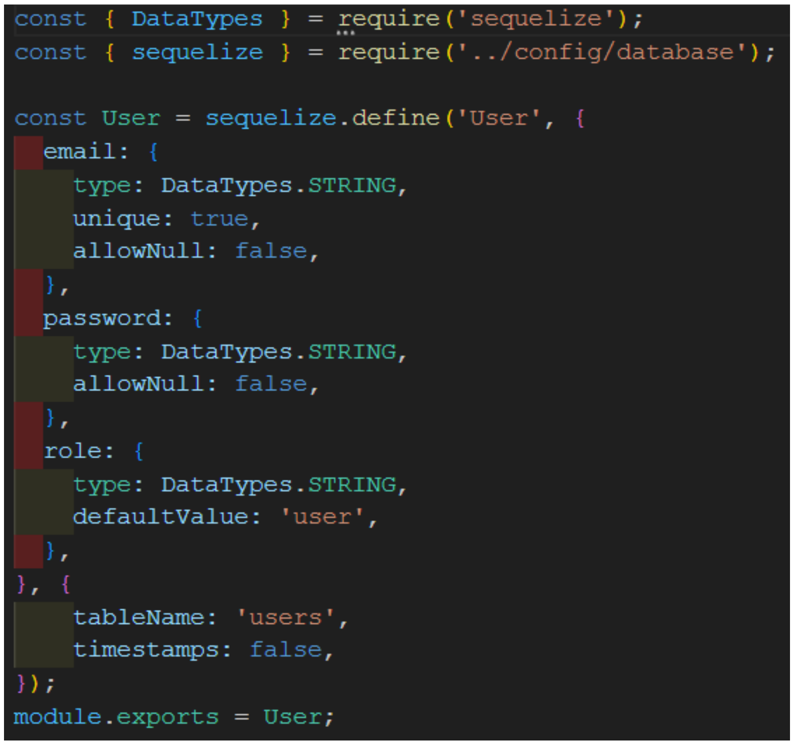
```
const jwt = require('jsonwebtoken');

Windsurf: Refactor | Explain | Generate JSDoc | X
exports.protect = (req, res, next) => {
  const token = req.cookies.token;
  if (!token) return res.status(401).json({ message: 'Unauthorized' });

  try {
    const user = jwt.verify(token, 'SECRET_KEY');
    req.user = user;
    next();
  } catch (err) {
    res.status(401).json({ message: 'Invalid token' });
  }
};

Windsurf: Refactor | Explain | Generate JSDoc | X
exports.adminOnly = (req, res, next) => {
  if (req.user.role !== 'admin') return res.status(403).
  json({ message: 'Forbidden' });
  next();
};
```

10. Membuat isi file **user.js** pada folder models



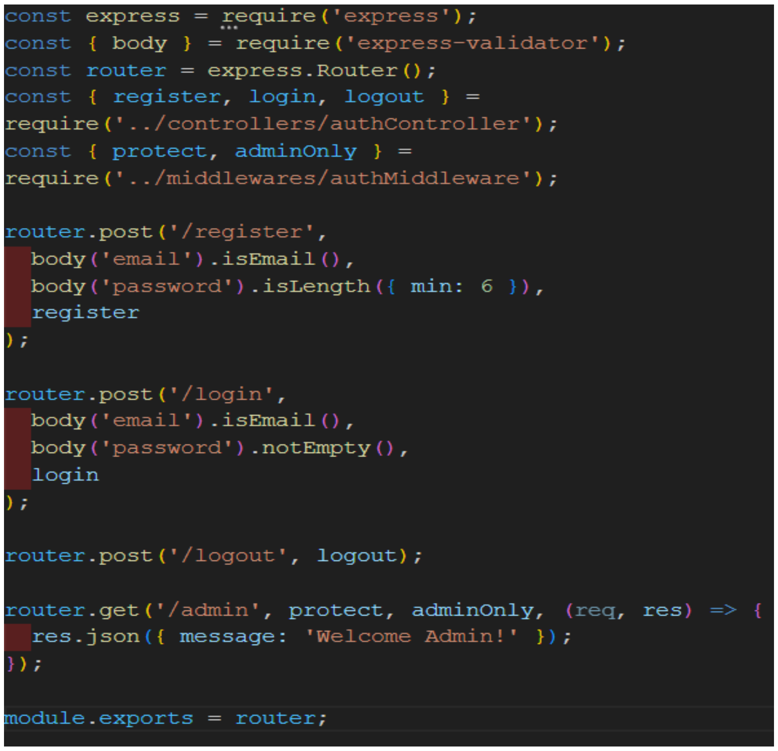
```javascript
const { DataTypes } = require('sequelize');
const { sequelize } = require('../config/db');

const User = sequelize.define('User', {
  email: {
    type: DataTypes.STRING,
    unique: true,
    allowNull: false,
  },
  password: {
    type: DataTypes.STRING,
    allowNull: false,
  },
  role: {
    type: DataTypes.STRING,
    defaultValue: 'user',
  },
});
module.exports = User;
```

11. Membuat isi file **userr.js** pada folder models

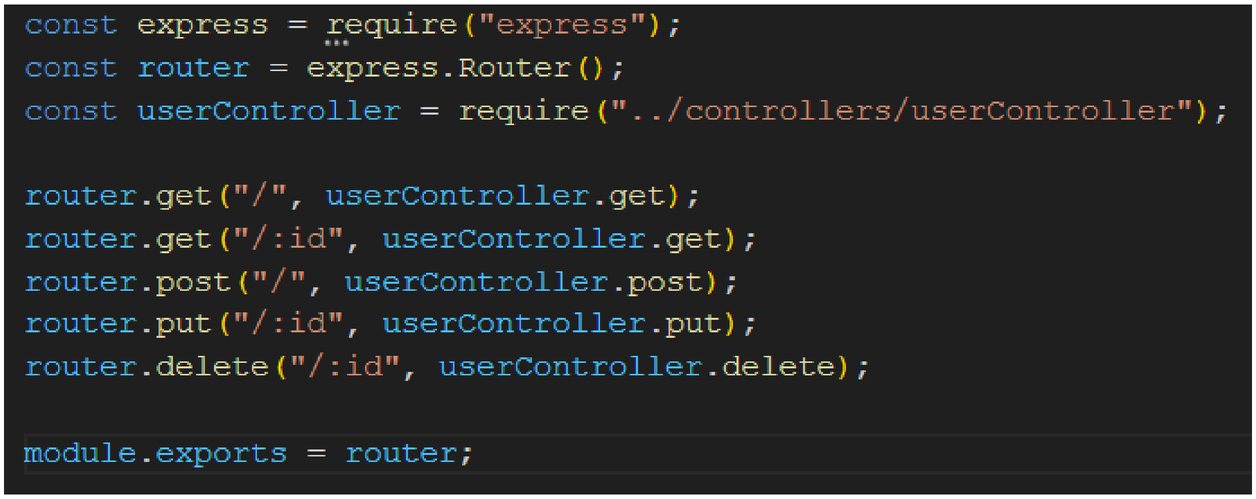


```javascript
const { DataTypes } = require('sequelize');
const { sequelize } = require('../config/database');

const User = sequelize.define('User', {
  email: {
    type: DataTypes.STRING,
    unique: true,
    allowNull: false,
  },
  password: {
    type: DataTypes.STRING,
    allowNull: false,
  },
  role: {
    type: DataTypes.STRING,
    defaultValue: 'user',
  },
}, {
    tableName: 'users',
    timestamps: false,
});
module.exports = User;
```

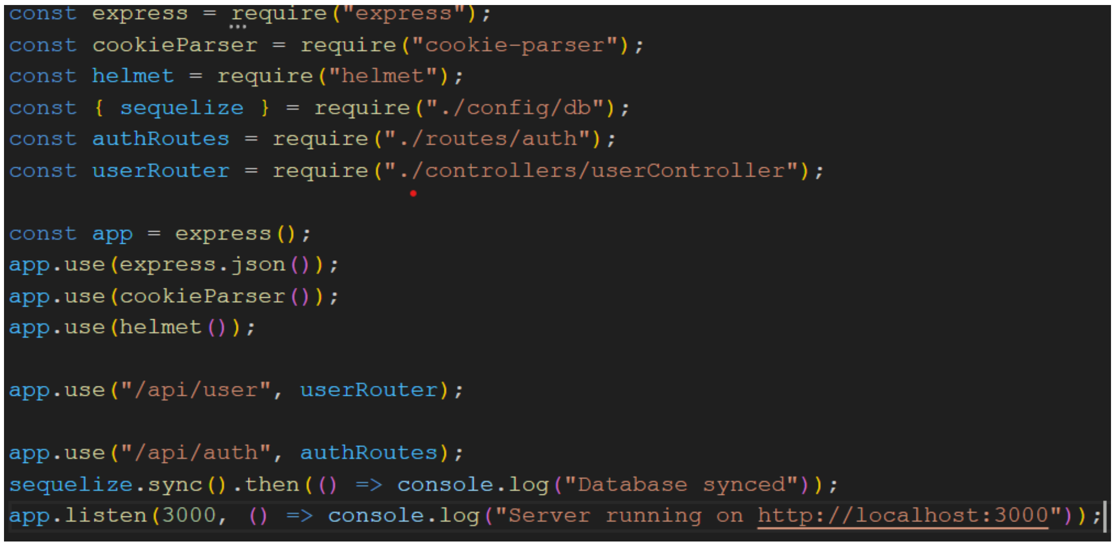
12. Membuat isi file **auth.js** pada folder routes


```
const express = require('express');
const { body } = require('express-validator');
const router = express.Router();
const { register, login, logout } =
require('../controllers/authController');
const { protect, adminOnly } =
require('../middlewares/authMiddleware');

router.post('/register',
  body('email').isEmail(),
  body('password').isLength({ min: 6 }),
  register
);

router.post('/login',
  body('email').isEmail(),
  body('password').notEmpty(),
  login
);

router.post('/logout', logout);

router.get('/admin', protect, adminOnly, (req, res) => {
  res.json({ message: 'Welcome Admin!' });
});

module.exports = router;
```

13. Membuat isi file **user.js** pada folder routes

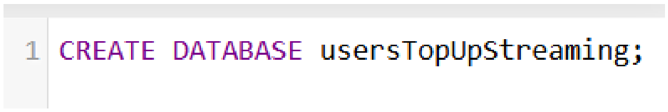

```
const express = require("express");
const router = express.Router();
const userController = require("../controllers/userController");

router.get("/", userController.get);
router.get("/:id", userController.get);
router.post("/", userController.post);
router.put("/:id", userController.put);
router.delete("/:id", userController.delete);

module.exports = router;
```

14. Membuat isi file **index.js**



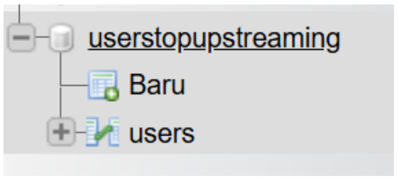
```
const express = require("express");
const cookieParser = require("cookie-parser");
const helmet = require("helmet");
const { sequelize } = require("./config/db");
const authRoutes = require("./routes/auth");
const userRouter = require("./controllers/userController");

const app = express();
app.use(express.json());
app.use(cookieParser());
app.use(helmet());

app.use("/api/user", userRouter);

app.use("/api/auth", authRoutes);
sequelize.sync().then(() => console.log("Database synced"));
app.listen(3000, () => console.log("Server running on http://localhost:3000"));
```

15. Membuat database di **MySQL**

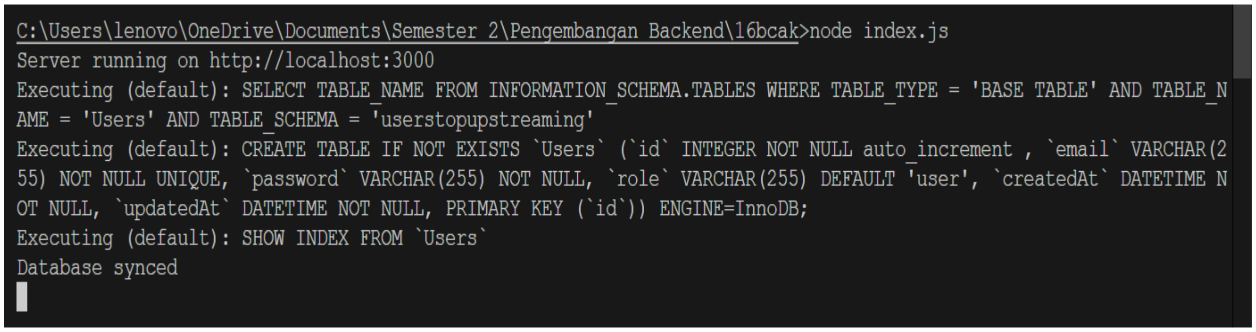


```
CREATE DATABASE usersTopUpStreaming;
```

Tampilan :



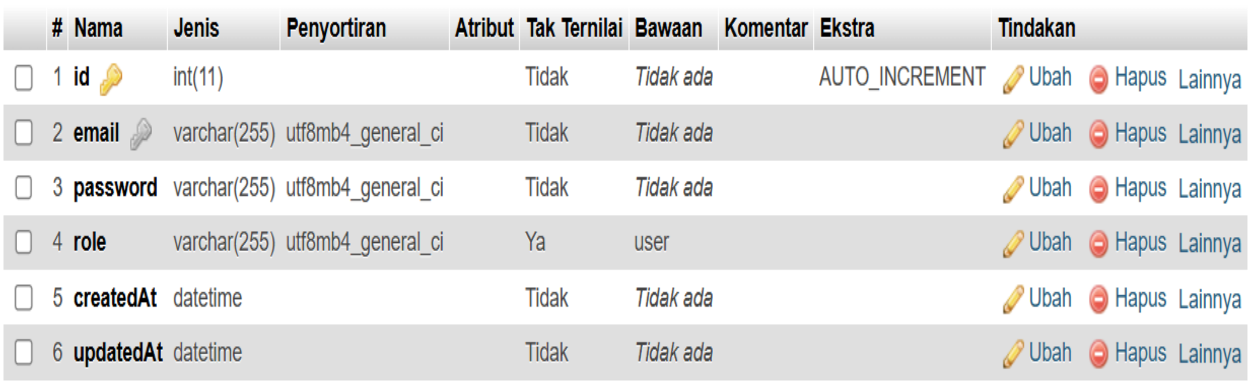
16. Node index.js



```
C:\Users\lenovo\OneDrive\Documents\Semester 2\Pengembangan Backend\16bcak>node index.js
Server running on http://localhost:3000
Executing (default): SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_TYPE = 'BASE TABLE' AND TABLE_N
AME = 'Users' AND TABLE_SCHEMA = 'userstopupstreaming'
Executing (default): CREATE TABLE IF NOT EXISTS `Users` (`id` INTEGER NOT NULL auto_increment , `email` VARCHAR(2
55) NOT NULL UNIQUE, `password` VARCHAR(255) NOT NULL, `role` VARCHAR(255) DEFAULT 'user', `createdAt` DATETIME N
OT NULL, `updatedAt` DATETIME NOT NULL, PRIMARY KEY (`id`)) ENGINE=InnoDB;
Executing (default): SHOW INDEX FROM `Users`
Database synced
```

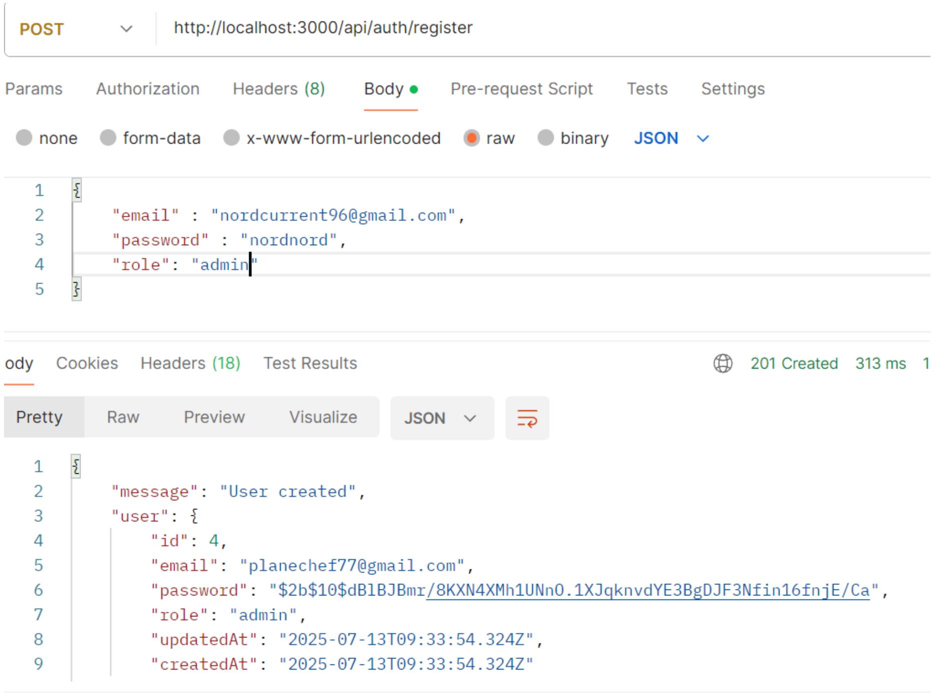
Penjelasan : Perintah ini digunakan untuk menjalankan file utama (index.js) dari server Express. File ini berfungsi sebagai entry point yang menginisialisasi server, middleware, dan koneksi ke database. Setelah dijalankan, server akan aktif dan siap menerima permintaan dari client.

17. Tampilan di localhost setelah terhubung dengan project



	#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan		
☐	1	**id**	int(11)			Tidak	*Tidak ada*		AUTO_INCREMENT	Ubah	Hapus	Lainnya
☐	2	**email**	varchar(255)	utf8mb4_general_ci		Tidak	*Tidak ada*			Ubah	Hapus	Lainnya
☐	3	**password**	varchar(255)	utf8mb4_general_ci		Tidak	*Tidak ada*			Ubah	Hapus	Lainnya
☐	4	**role**	varchar(255)	utf8mb4_general_ci		Ya	user			Ubah	Hapus	Lainnya
☐	5	**createdAt**	datetime			Tidak	*Tidak ada*			Ubah	Hapus	Lainnya
☐	6	**updatedAt**	datetime			Tidak	*Tidak ada*			Ubah	Hapus	Lainnya

18. **Ujicoba di postman** : Registrasi admin



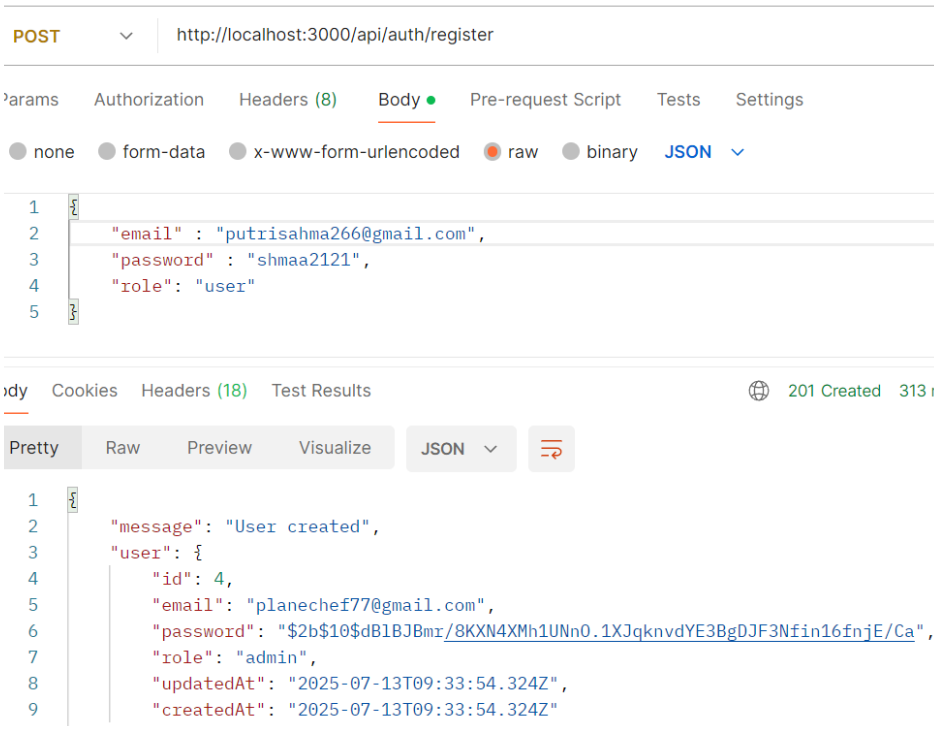
```
http://localhost:3000/api/auth/register
```

```
{
    "email" : "nordcurrent96@gmail.com",
    "password" : "nordnord",
    "role": "admin"
}
```

201 Created 313 ms 1

```
{
    "message": "User created",
    "user": {
        "id": 4,
        "email": "planechef77@gmail.com",
        "password": "$2b$10$dBlBJBmr/8KXN4XMh1UNn0.1XJqknvdYE3BgDJF3Nfin16fnjE/Ca",
        "role": "admin",
        "updatedAt": "2025-07-13T09:33:54.324Z",
        "createdAt": "2025-07-13T09:33:54.324Z"
```

19. **Ujicoba di postman** : Registrasi user



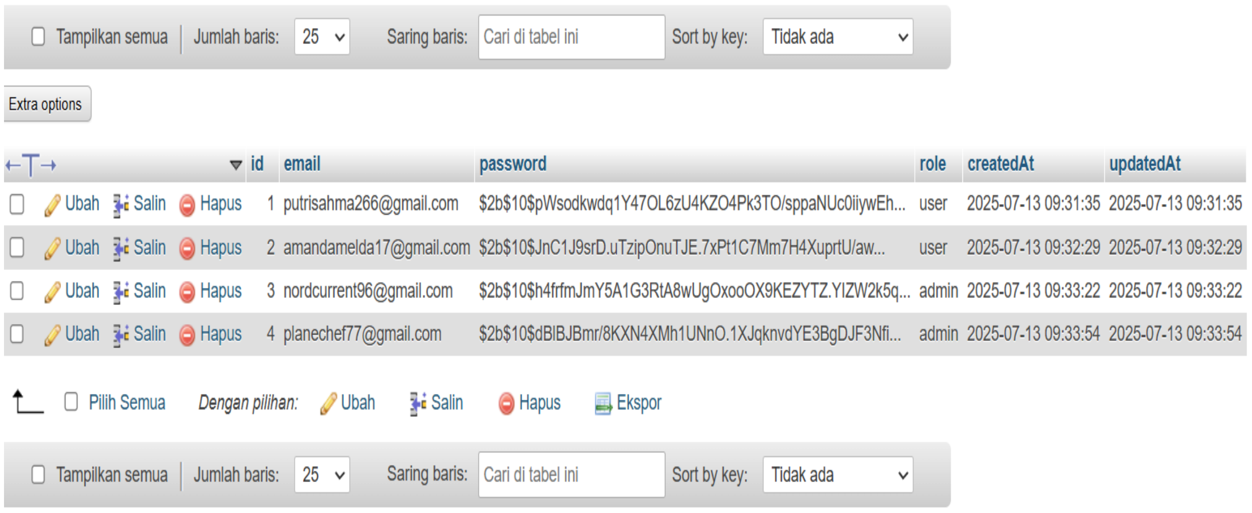
POST http://localhost:3000/api/auth/register

```
{
    "email" : "putrisahma266@gmail.com",
    "password" : "shmaa2121",
    "role": "user"
}
```

201 Created

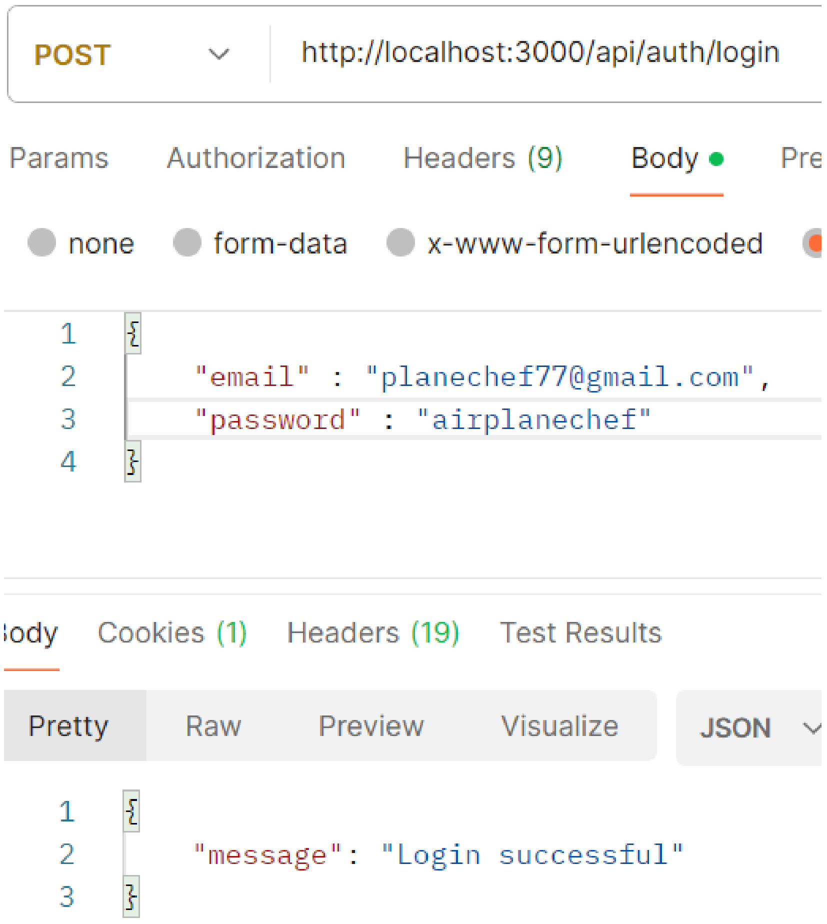
```
{
    "message": "User created",
    "user": {
        "id": 4,
        "email": "planechef77@gmail.com",
        "password": "$2b$10$dBlBJBmr/8KXN4XMh1UNnO.1XJqknvdYE3BgDJF3Nfin16fnjE/Ca",
        "role": "admin",
        "updatedAt": "2025-07-13T09:33:54.324Z",
        "createdAt": "2025-07-13T09:33:54.324Z"
```

20. Tampilan di localhost :



	id	email	password	role	createdAt	updatedAt
Ubah Salin Hapus	1	putrisahma266@gmail.com	$2b$10$pWsodkwdq1Y47OL6zU4KZO4Pk3TO/sppaNUc0iiywEh...	user	2025-07-13 09:31:35	2025-07-13 09:31:35
Ubah Salin Hapus	2	amandamelda17@gmail.com	$2b$10$JnC1J9srD.uTzipOnuTJE.7xPt1C7Mm7H4XuprtU/aw...	user	2025-07-13 09:32:29	2025-07-13 09:32:29
Ubah Salin Hapus	3	nordcurrent96@gmail.com	$2b$10$h4frfmJmY5A1G3RtA8wUgOxooOX9KEZYTZ.YIZW2k5q...	admin	2025-07-13 09:33:22	2025-07-13 09:33:22
Ubah Salin Hapus	4	planechef77@gmail.com	$2b$10$dBIBJBmr/8KXN4XMh1UNnO.1XJqknvdYE3BgDJF3Nfi...	admin	2025-07-13 09:33:54	2025-07-13 09:33:54

21. **Ujicoba di postman** : Admin login “Login juga berlaku bagi user”

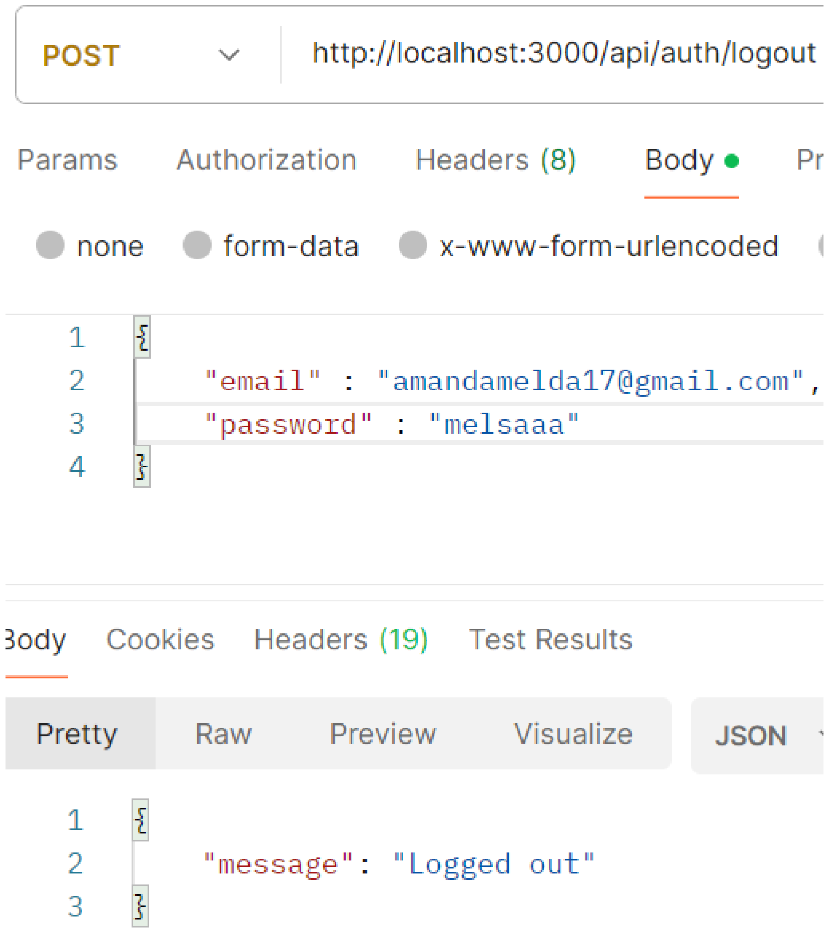


```
http://localhost:3000/api/auth/login
```

```
{
    "email" : "planechef77@gmail.com",
    "password" : "airplanechef"
}
```

```
{
    "message": "Login successful"
}
```

22. **Ujicoba di postman** : Logout

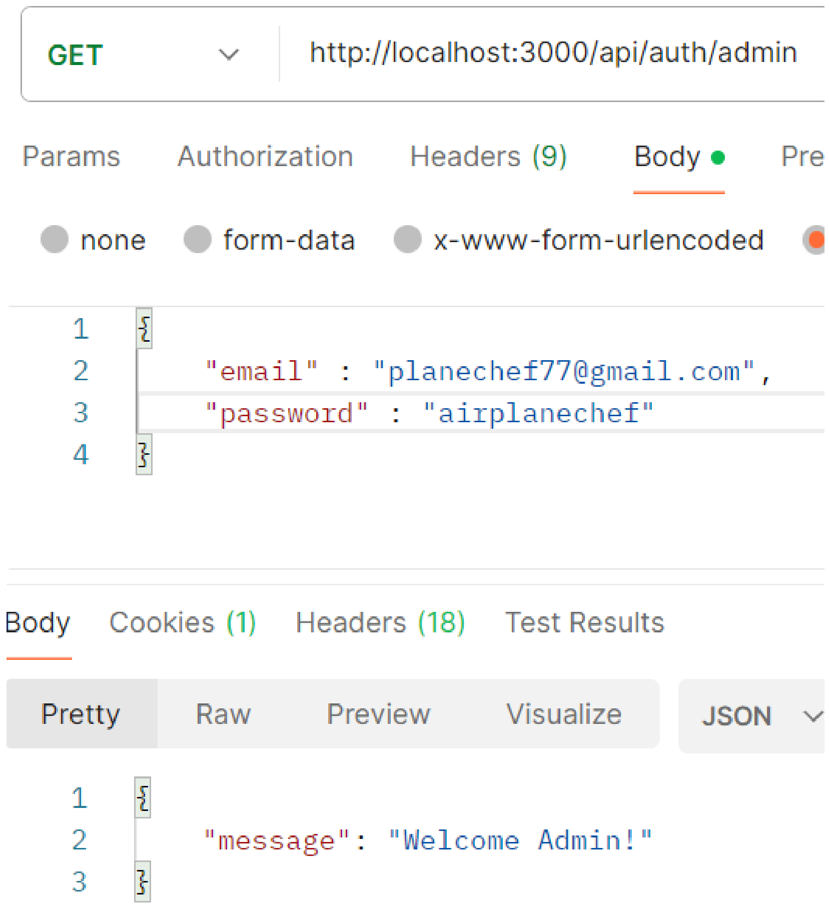


```
http://localhost:3000/api/auth/logout
```

```
{
    "email" : "amandamelda17@gmail.com",
    "password" : "melsaaa"
}
```

```
{
    "message": "Logged out"
}
```

23. Tempat akses khusus admin


GET http://localhost:3000/api/auth/admin

Params Authorization Headers (9) Body ● Pre

none form-data x-www-form-urlencoded

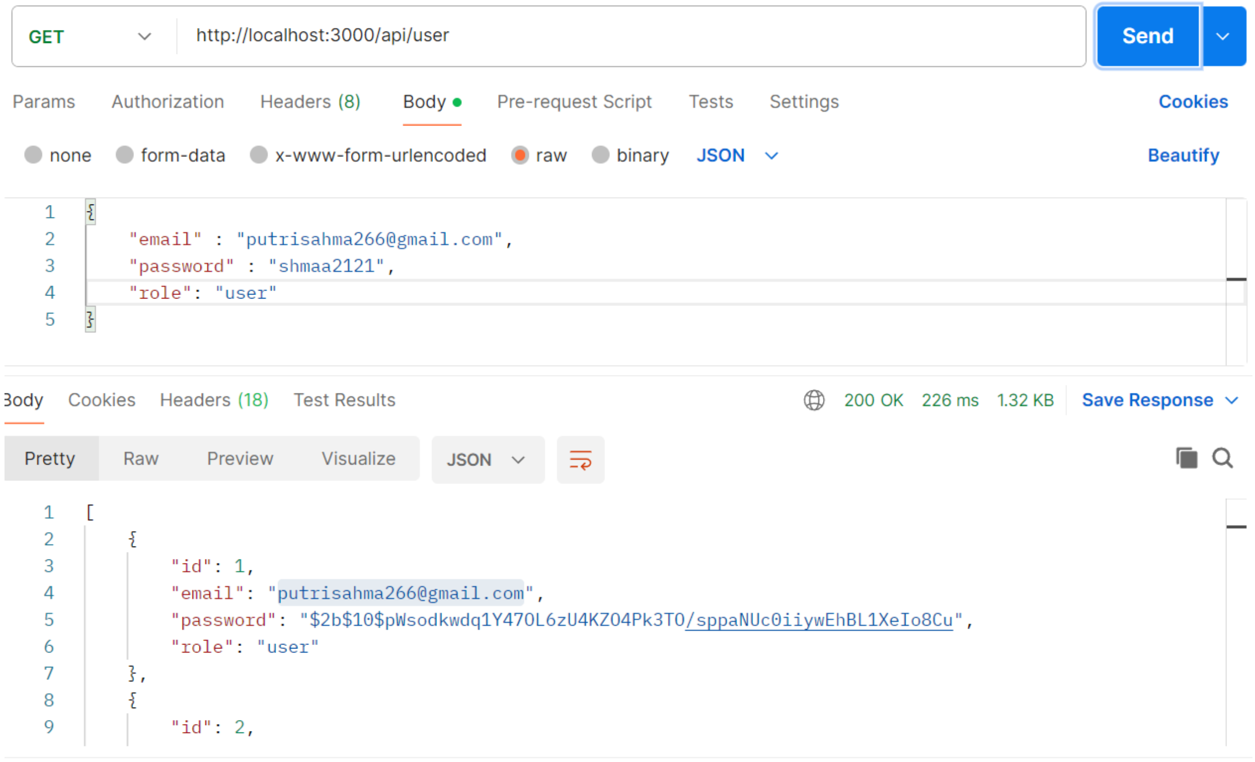
```
{
    "email" : "planechef77@gmail.com",
    "password" : "airplanechef"
}
```

Body Cookies (1) Headers (18) Test Results

Pretty Raw Preview Visualize JSON

```
{
    "message": "Welcome Admin!"
}
```

24. Proses **CRUD** : GET



GET http://localhost:3000/api/user

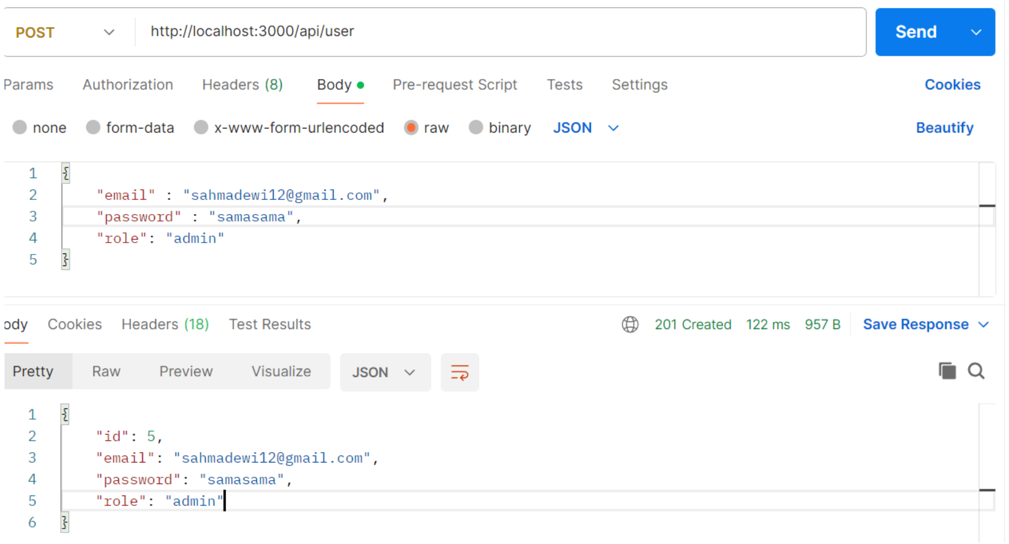
Body: raw JSON

```
{
    "email" : "putrisahma266@gmail.com",
    "password" : "shmaa2121",
    "role": "user"
}
```

200 OK 226 ms 1.32 KB

```
[
    {
        "id": 1,
        "email": "putrisahma266@gmail.com",
        "password": "$2b$10$pWsodkwdq1Y470L6zU4KZ04Pk3TO/sppaNUc0iiywEhBL1XeIo8Cu",
        "role": "user"
    },
    {
        "id": 2,
```

25. Proses **CRUD** : POST



POST http://localhost:3000/api/user

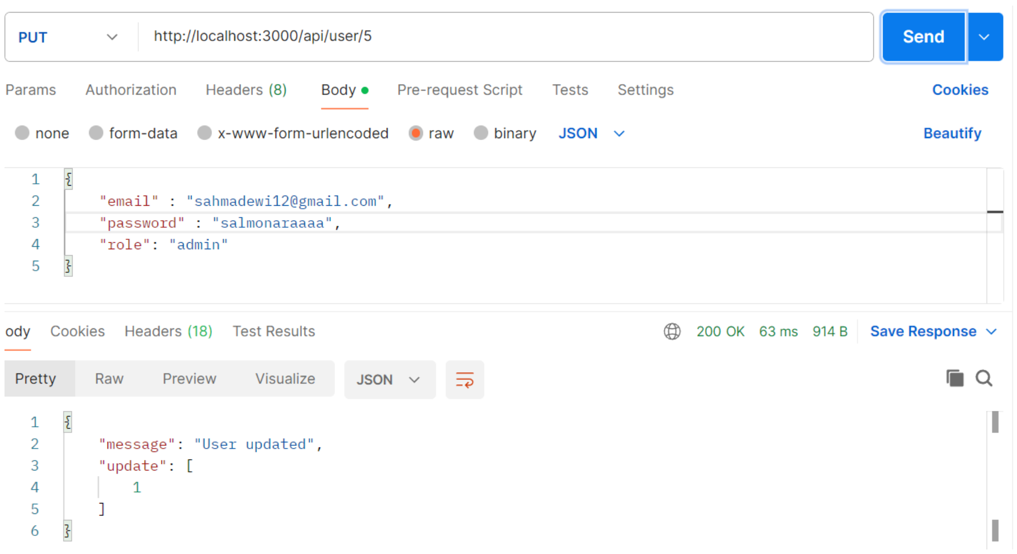
Body: raw JSON

```
{
    "email" : "sahmadewi12@gmail.com",
    "password" : "samasama",
    "role": "admin"
}
```

201 Created 122 ms 957 B

```
{
    "id": 5,
    "email": "sahmadewi12@gmail.com",
    "password": "samasama",
    "role": "admin"
}
```

26. Proses **CRUD** : PUT



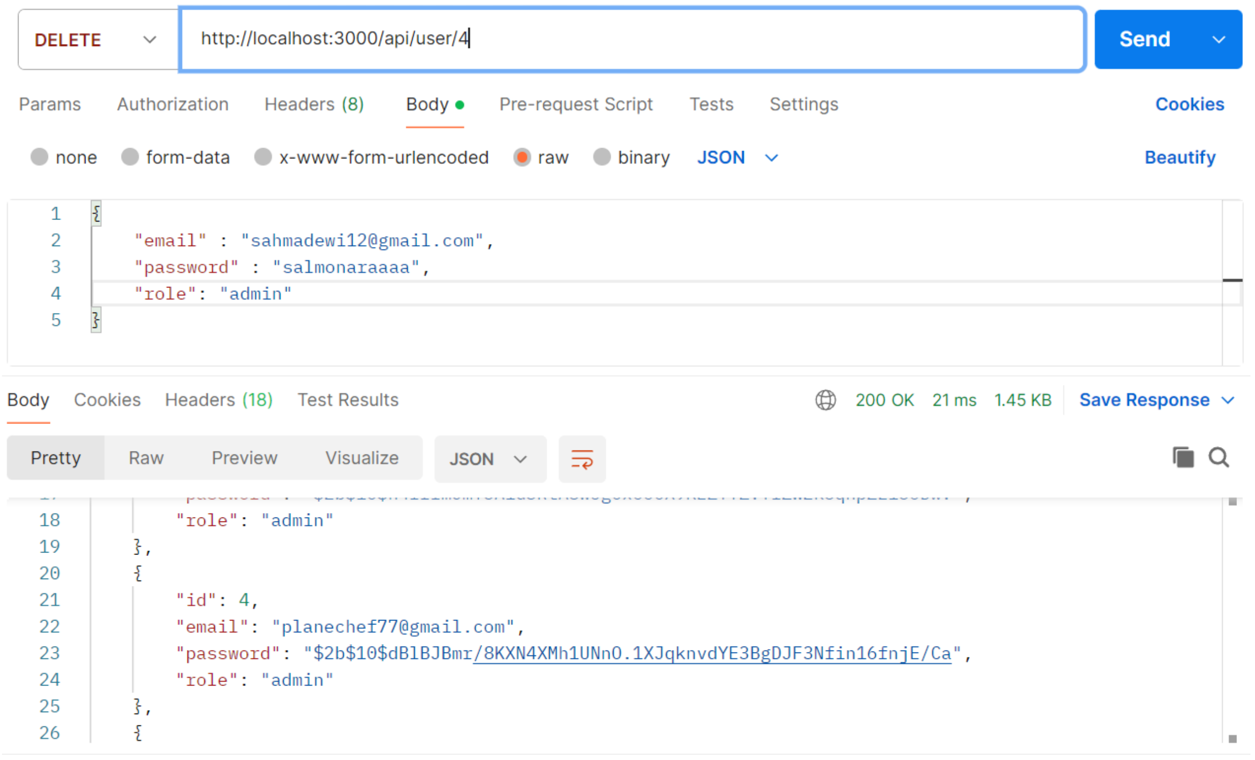
http://localhost:3000/api/user/5

```
{
    "email" : "sahmadewi12@gmail.com",
    "password" : "salmonaraaaa",
    "role": "admin"
}
```

200 OK 63 ms 914 B

```
{
    "message": "User updated",
    "update": [
        1
    ]
}
```

27. Proses **CRUD** : DELETE



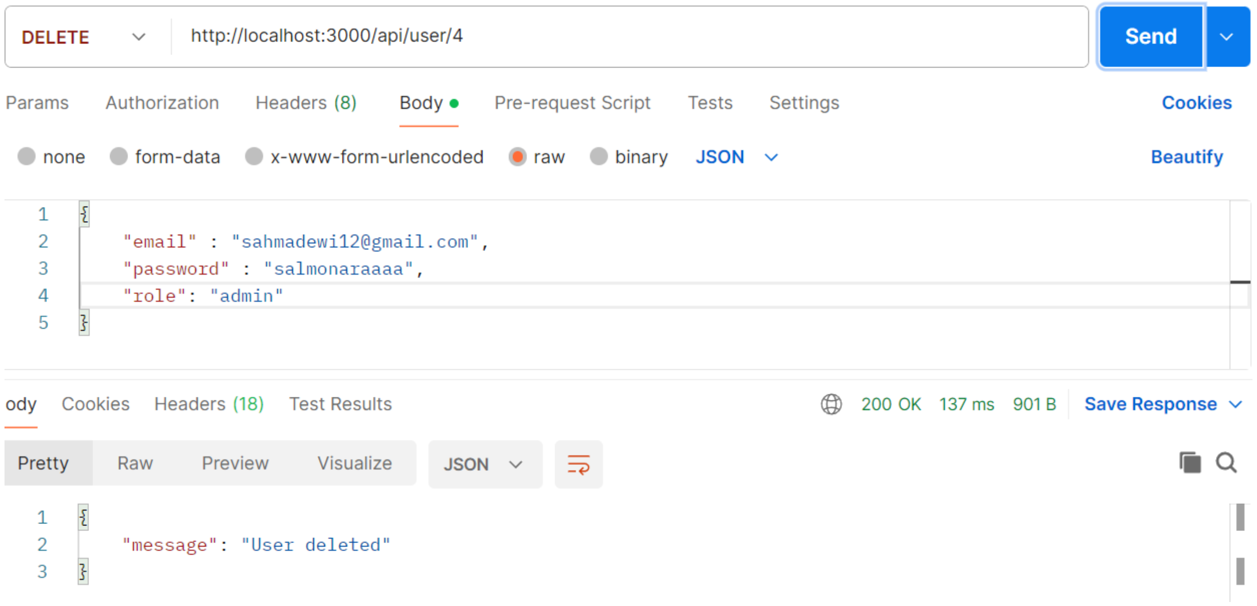
http://localhost:3000/api/user/4

```
{
    "email" : "sahmadewi12@gmail.com",
    "password" : "salmonaraaaa",
    "role": "admin"
}
```

200 OK 21 ms 1.45 KB

```
        "password": [illegible]
        "role": "admin"
    },
    {
        "id": 4,
        "email": "planechef77@gmail.com",
        "password": "$2b$10$dBlBJBmr/8KXN4XMh1UNn0.1XJqknvdYE3BgDJF3Nfin16fnjE/Ca",
        "role": "admin"
    },
    {
```

Tampilan setelah di delete :



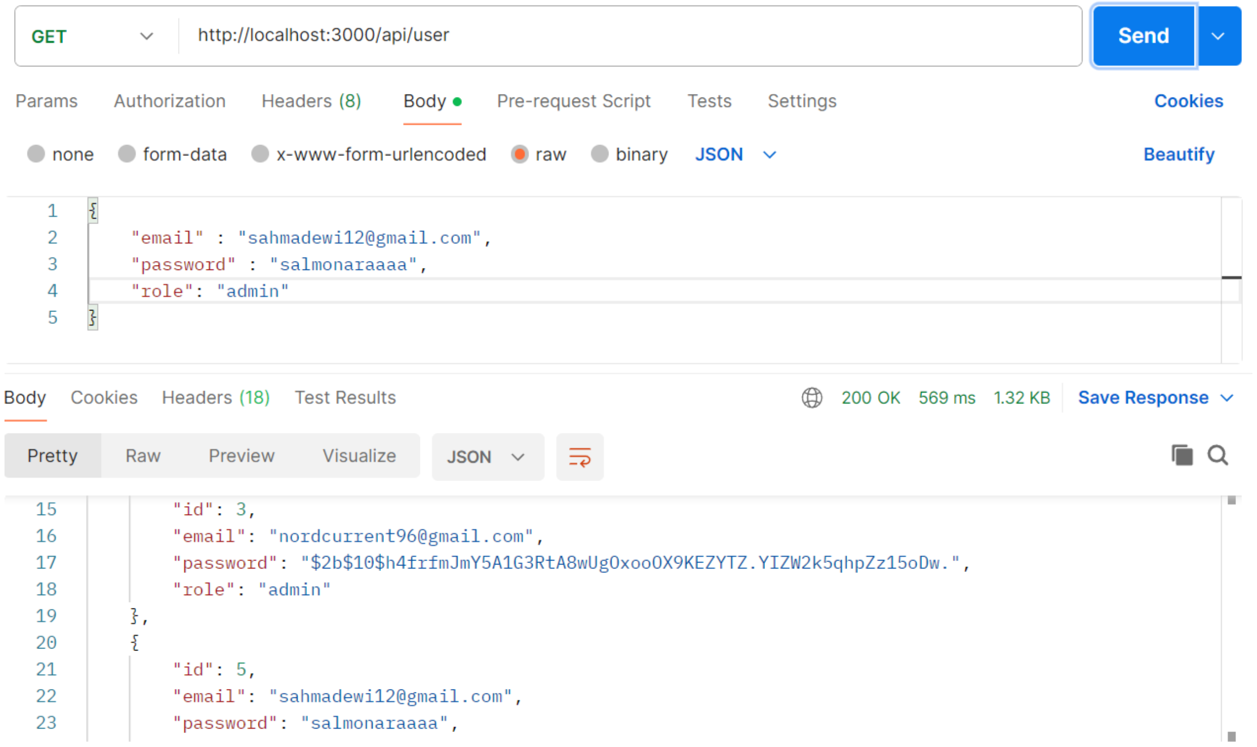
DELETE http://localhost:3000/api/user/4

Body: raw, JSON

```
{
    "email" : "sahmadewi12@gmail.com",
    "password" : "salmonaraaaa",
    "role": "admin"
}
```

200 OK 137 ms 901 B

```
{
    "message": "User deleted"
}
```



GET http://localhost:3000/api/user

Body: raw, JSON

```
{
    "email" : "sahmadewi12@gmail.com",
    "password" : "salmonaraaaa",
    "role": "admin"
}
```

200 OK 569 ms 1.32 KB

```
        "id": 3,
        "email": "nordcurrent96@gmail.com",
        "password": "$2b$10$h4frfmJmY5A1G3RtA8wUg0xooOX9KEZYTZ.YIZW2k5qhpZz15oDw.",
        "role": "admin"
    },
    {
        "id": 5,
        "email": "sahmadewi12@gmail.com",
        "password": "salmonaraaaa",
```

Link github : https://github.com/putrisyk888/pengembangan-backend/tree/main/Pertemuan%2016