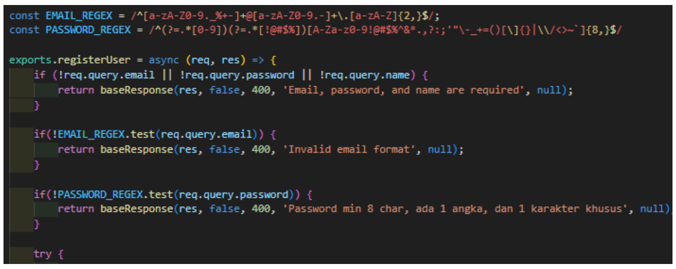
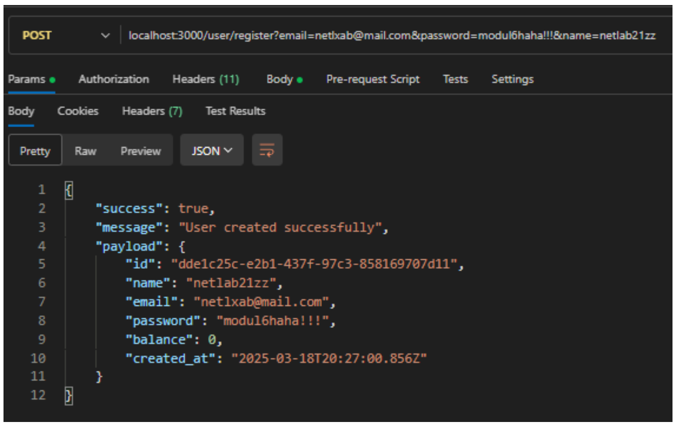
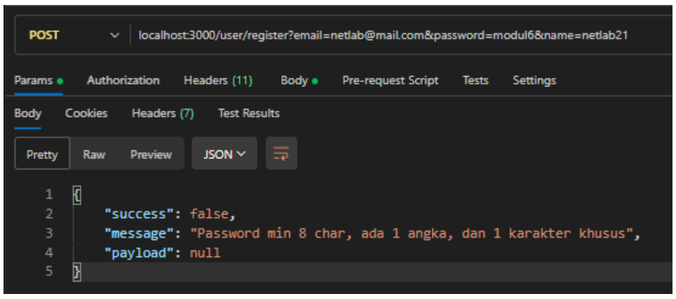
PRAKTIKUM
SISTEM BASIS DATA

Nama	**Jesaya David Gamalael N P**	**No. Modul**	**6**
NPM	**2306161965**	**Tipe**	**Case Study**

1. Regex

Regex	Screenshot
/user/register	  

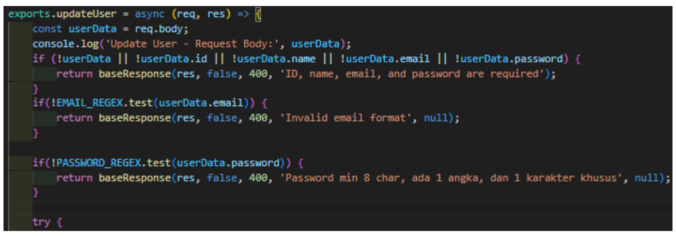
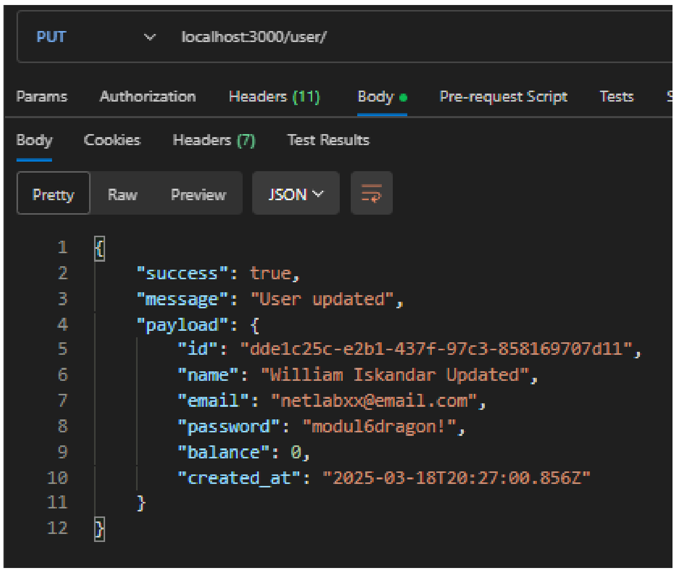
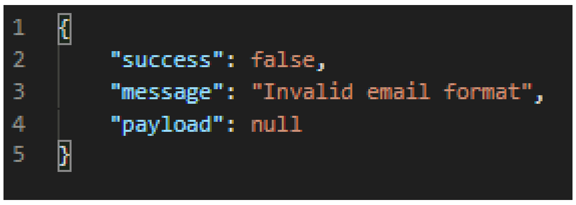
```
const EMAIL_REGEX = /^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/;
const PASSWORD_REGEX = /^(?=.*[0-9])(?=.*[!@#$%])[A-Za-z0-9!@#$%^&*.,?:;'"\-_+=()[\]{}|\\/<>~`]{8,}$/

exports.registerUser = async (req, res) => {
    if (!req.query.email || !req.query.password || !req.query.name) {
        return baseResponse(res, false, 400, 'Email, password, and name are required', null);
    }

    if(!EMAIL_REGEX.test(req.query.email)) {
        return baseResponse(res, false, 400, 'Invalid email format', null);
    }

    if(!PASSWORD_REGEX.test(req.query.password)) {
        return baseResponse(res, false, 400, 'Password min 8 char, ada 1 angka, dan 1 karakter khusus', null)
    }

    try {
```

POST localhost:3000/user/register?email=netlxab@mail.com&password=modul6haha!!!&name=netlab21zz

```
{
    "success": true,
    "message": "User created successfully",
    "payload": {
        "id": "dde1c25c-e2b1-437f-97c3-858169707d11",
        "name": "netlab21zz",
        "email": "netlxab@mail.com",
        "password": "modul6haha!!!",
        "balance": 0,
        "created_at": "2025-03-18T20:27:00.856Z"
    }
}
```

POST localhost:3000/user/register?email=netlab@mail.com&password=modul6&name=netlab21

```
{
    "success": false,
    "message": "Password min 8 char, ada 1 angka, dan 1 karakter khusus",
    "payload": null
}
```

/user/ (PUT)	  

```
exports.updateUser = async (req, res) => {
    const userData = req.body;
    console.log('Update User - Request Body:', userData);
    if (!userData || !userData.id || !userData.name || !userData.email || !userData.password) {
        return baseResponse(res, false, 400, 'ID, name, email, and password are required');
    }
    if(!EMAIL_REGEX.test(userData.email)) {
        return baseResponse(res, false, 400, 'Invalid email format', null);
    }

    if(!PASSWORD_REGEX.test(userData.password)) {
        return baseResponse(res, false, 400, 'Password min 8 char, ada 1 angka, dan 1 karakter khusus', null);
    }

    try {
```

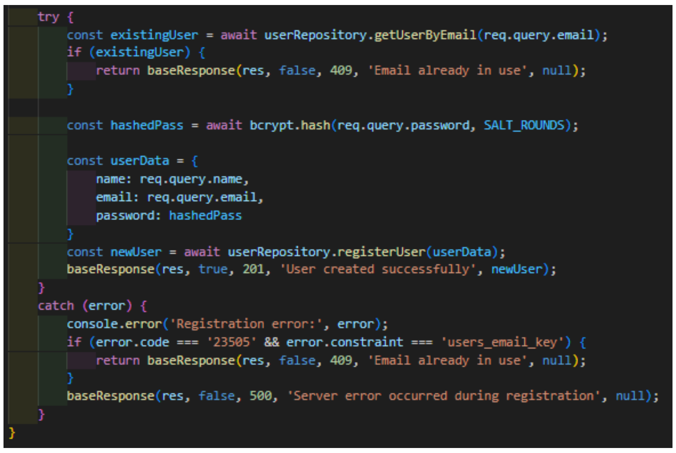
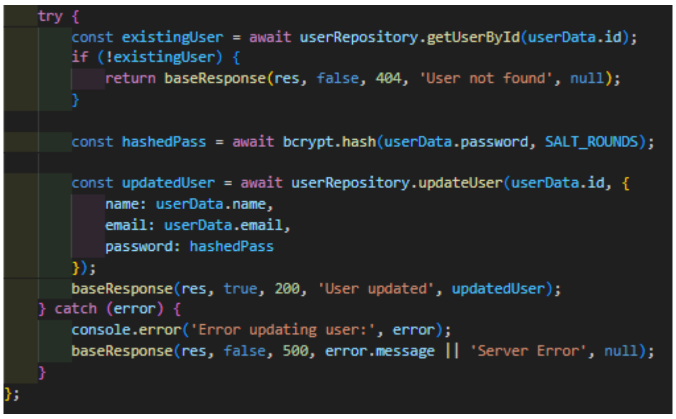
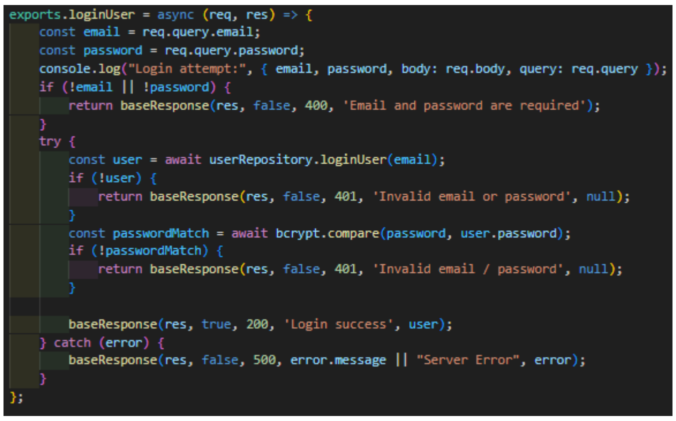
PUT localhost:3000/user/

```
{
    "success": true,
    "message": "User updated",
    "payload": {
        "id": "dde1c25c-e2b1-437f-97c3-858169707d11",
        "name": "William Iskandar Updated",
        "email": "netlabxx@email.com",
        "password": "modul6dragon!",
        "balance": 0,
        "created_at": "2025-03-18T20:27:00.856Z"
    }
}
```

```
{
    "success": false,
    "message": "Invalid email format",
    "payload": null
}
```

2. Hashing Bcrypt di PUT dan POST
3. Cocokkan dengan LOGIN

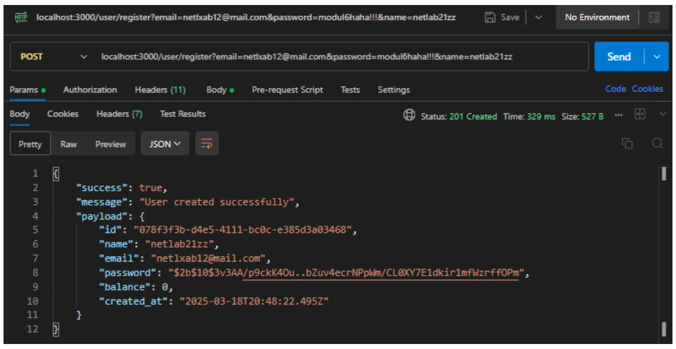
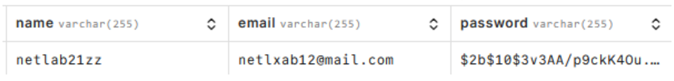
Regex	Screenshot

Hash Password Register	
Hash Password PUT	
Cocokkan di Endpoint LOGIN	

```
    try {
        const existingUser = await userRepository.getUserByEmail(req.query.email);
        if (existingUser) {
            return baseResponse(res, false, 409, 'Email already in use', null);
        }

        const hashedPass = await bcrypt.hash(req.query.password, SALT_ROUNDS);

        const userData = {
            name: req.query.name,
            email: req.query.email,
            password: hashedPass
        }
        const newUser = await userRepository.registerUser(userData);
        baseResponse(res, true, 201, 'User created successfully', newUser);
    }
    catch (error) {
        console.error('Registration error:', error);
        if (error.code === '23505' && error.constraint === 'users_email_key') {
            return baseResponse(res, false, 409, 'Email already in use', null);
        }
        baseResponse(res, false, 500, 'Server error occurred during registration', null);
    }
}
```

```
    try {
        const existingUser = await userRepository.getUserById(userData.id);
        if (!existingUser) {
            return baseResponse(res, false, 404, 'User not found', null);
        }

        const hashedPass = await bcrypt.hash(userData.password, SALT_ROUNDS);

        const updatedUser = await userRepository.updateUser(userData.id, {
            name: userData.name,
            email: userData.email,
            password: hashedPass
        });
        baseResponse(res, true, 200, 'User updated', updatedUser);
    } catch (error) {
        console.error('Error updating user:', error);
        baseResponse(res, false, 500, error.message || 'Server Error', null);
    }
};
```

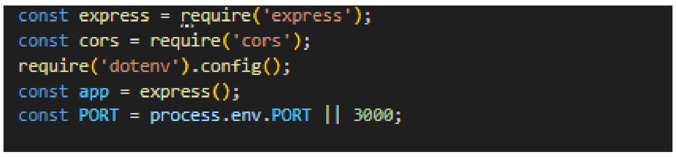
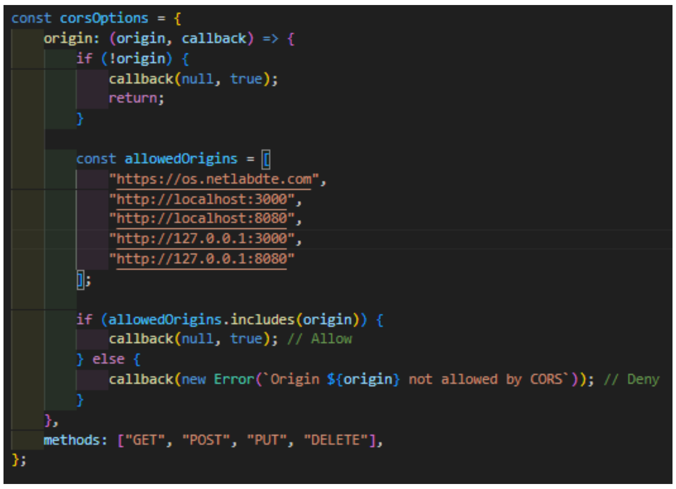
```
exports.loginUser = async (req, res) => {
    const email = req.query.email;
    const password = req.query.password;
    console.log("Login attempt:", { email, password, body: req.body, query: req.query });
    if (!email || !password) {
        return baseResponse(res, false, 400, 'Email and password are required');
    }
    try {
        const user = await userRepository.loginUser(email);
        if (!user) {
            return baseResponse(res, false, 401, 'Invalid email or password', null);
        }
        const passwordMatch = await bcrypt.compare(password, user.password);
        if (!passwordMatch) {
            return baseResponse(res, false, 401, 'Invalid email / password', null);
        }

        baseResponse(res, true, 200, 'Login success', user);
    } catch (error) {
        baseResponse(res, false, 500, error.message || "Server Error", error);
    }
};
```

Uji (Hash Sukses)	 Pass Input : modul6haha!! Pass Stored : Hash Bcrypt 

```
POST localhost:3000/user/register?email=netlxab12@mail.com&password=modul6haha!!!&name=netlab21zz
Status: 201 Created  Time: 329 ms  Size: 527 B
{
    "success": true,
    "message": "User created successfully",
    "payload": {
        "id": "078f3f3b-d4e5-4111-bc0c-e385d3a03468",
        "name": "netlab21zz",
        "email": "netlxab12@mail.com",
        "password": "$2b$10$3v3AA/p9ckK4Ou..bZuv4ecrNPpWm/CL0XY7E1dkir1mfWzrffOPm",
        "balance": 0,
        "created_at": "2025-03-18T20:48:22.495Z"
    }
}
```

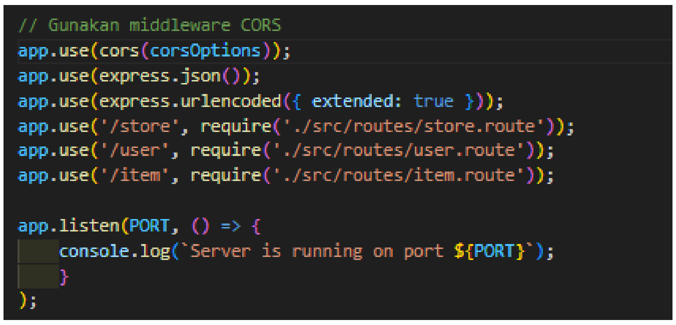
name varchar(255)	email varchar(255)	password varchar(255)
netlab21zz	netlxab12@mail.com	$2b$10$3v3AA/p9ckK4Ou.…

4. CORS Middleware

Step	Screenshot
Index.js	 

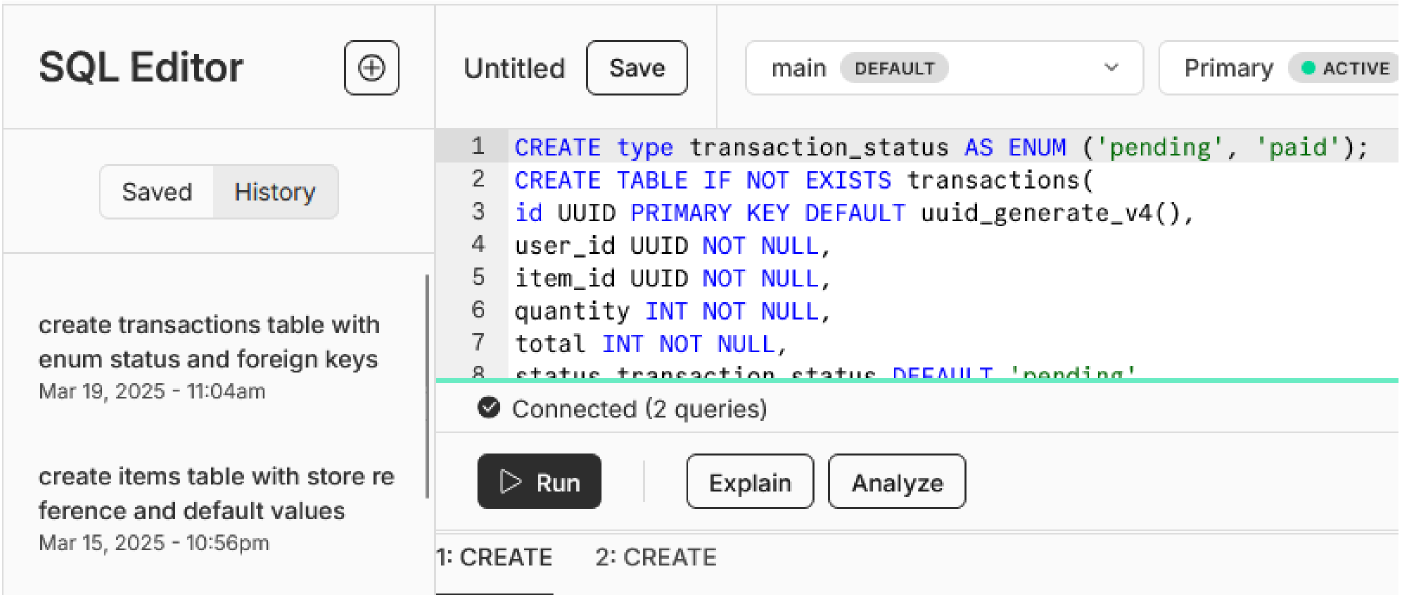
```
const express = require('express');
const cors = require('cors');
require('dotenv').config();
const app = express();
const PORT = process.env.PORT || 3000;
```

```
const corsOptions = {
    origin: (origin, callback) => {
        if (!origin) {
            callback(null, true);
            return;
        }

        const allowedOrigins = [
            "https://os.netlabdte.com",
            "http://localhost:3000",
            "http://localhost:8080",
            "http://127.0.0.1:3000",
            "http://127.0.0.1:8080"
        ];

        if (allowedOrigins.includes(origin)) {
            callback(null, true); // Allow
        } else {
            callback(new Error(`Origin ${origin} not allowed by CORS`)); // Deny
        }
    },
    methods: ["GET", "POST", "PUT", "DELETE"],
};
```

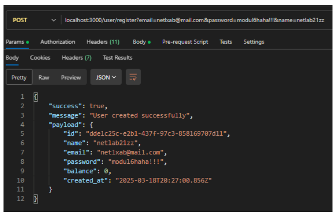
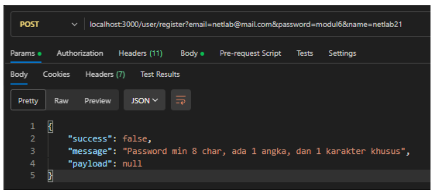
```
// Gunakan middleware CORS
app.use(cors(corsOptions));
app.use(express.json());
app.use(express.urlencoded({ extended: true }));
app.use('/store', require('./src/routes/store.route'));
app.use('/user', require('./src/routes/user.route'));
app.use('/item', require('./src/routes/item.route'));

app.listen(PORT, () => {
    console.log(`Server is running on port ${PORT}`);
    }
);
```

5. Query Neon Console



```
CREATE type transaction_status AS ENUM ('pending', 'paid');
CREATE TABLE IF NOT EXISTS transactions(
id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
user_id UUID NOT NULL,
item_id UUID NOT NULL,
quantity INT NOT NULL,
total INT NOT NULL,
status transaction_status DEFAULT 'pending'
```

6. Tabel API Transactions

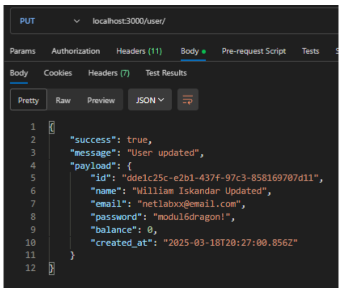
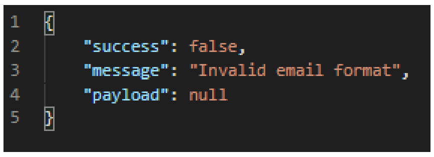
Method	Endpoint	Success	Failed
POST	/user/register		

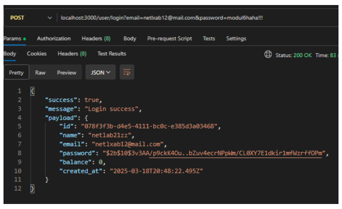
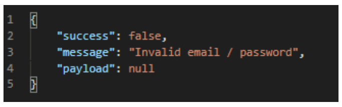
Success:

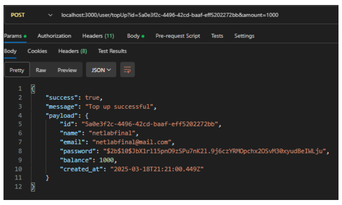
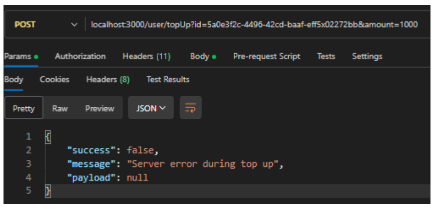
```
POST localhost:3000/user/register?email=netlxab@mail.com&password=modul6haha!!!&name=netlab21zz
{
    "success": true,
    "message": "User created successfully",
    "payload": {
        "id": "dde1c25c-e2b1-437f-97c3-858169707d11",
        "name": "netlab21zz",
        "email": "netlxab@mail.com",
        "password": "modul6haha!!!",
        "balance": 0,
        "created_at": "2025-03-18T20:27:00.856Z"
    }
}
```

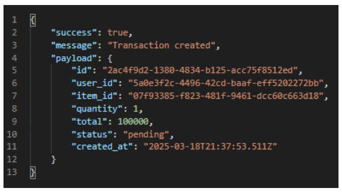
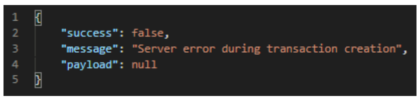
Failed:

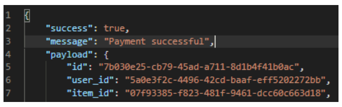
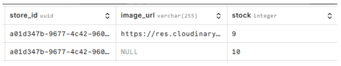
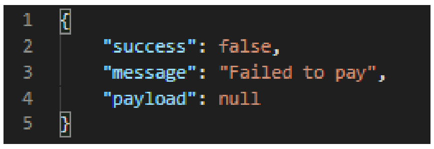
```
POST localhost:3000/user/register?email=netlab@mail.com&password=modul6&name=netlab21
{
    "success": false,
    "message": "Password min 8 char, ada 1 angka, dan 1 karakter khusus",
    "payload": null
}
```

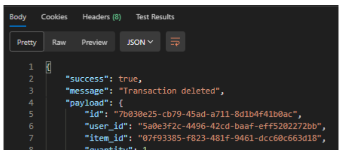
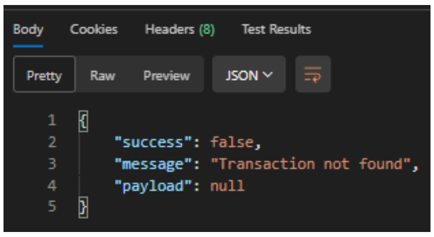
PUT	/user	 PUT localhost:3000/user/
		{
		"success": true,
		"message": "User updated",
		"payload": {
		"id": "dde1c25c-e2b1-437f-97c3-858169707d11",
		"name": "William Iskandar Updated",
		"email": "netlabxx@email.com",
		"password": "modul6dragon!",
		"balance": 0,
		"created_at": "2025-03-18T20:27:00.856Z"
		}
		}
			 {
			"success": false,
			"message": "Invalid email format",
			"payload": null
			}

POST	/user/login	 POST localhost:3000/user/login?email=netlxab12@mail.com&password=modul6haha!!!
		Status: 200 OK Time: 83
		{
		"success": true,
		"message": "Login success",
		"payload": {
		"id": "078f3f3b-d4e5-4111-bc0c-e385d3a03468",
		"name": "netlab21zz",
		"email": "netlxab12@mail.com",
		"password": "$2b$10$3v3AA/p9ckK4Ou..bZuv4ecrNPpWm/CL0XY7E1dkir1mfWzrffOPm",
		"balance": 0,
		"created_at": "2025-03-18T20:48:22.495Z"
		}
		}
			 {
			"success": false,
			"message": "Invalid email / password",
			"payload": null
			}

POST	/user/topUp	 POST localhost:3000/user/topUp?id=5a0e3f2c-4496-42cd-baaf-eff5202272bb&amount=1000
		{
		"success": true,
		"message": "Top up successful",
		"payload": {
		"id": "5a0e3f2c-4496-42cd-baaf-eff5202272bb",
		"name": "netlabfinal",
		"email": "netlabfinal@mail.com",
		"password": "$2b$10$JbX1rl15pnO9zSPu7nK21.9j6czYRMOpchx2OSvM30xyudBeIWLju",
		"balance": 1000,
		"created_at": "2025-03-18T21:21:00.449Z"
		}
		}
			 POST localhost:3000/user/topUp?id=5a0e3f2c-4496-42cd-baaf-eff5x02272bb&amount=1000
			{
			"success": false,
			"message": "Server error during top up",
			"payload": null
			}

POST	/transaction/create	 {
		"success": true,
		"message": "Transaction created",
		"payload": {
		"id": "2ac4f9d2-1380-4834-b125-acc75f8512ed",
		"user_id": "5a0e3f2c-4496-42cd-baaf-eff5202272bb",
		"item_id": "07f93385-f823-481f-9461-dcc60c663d18",
		"quantity": 1,
		"total": 100000,
		"status": "pending",
		"created_at": "2025-03-18T21:37:53.511Z"
		}
		}
			 {
			"success": false,
			"message": "Server error during transaction creation",
			"payload": null
			}

POST	/transaction/pay	 {
		"success": true,
		"message": "Payment successful",
		"payload": {
		"id": "7b030e25-cb79-45ad-a711-8d1b4f41b0ac",
		"user_id": "5a0e3f2c-4496-42cd-baaf-eff5202272bb",
		"item_id": "07f93385-f823-481f-9461-dcc60c663d18",
		 store_id uuid | image_url varchar(255) | stock integer
		a01d347b-9677-4c42-960… | https://res.cloudinary… | 9
		a01d347b-9677-4c42-960… | NULL | 10
		(Stok berkurang)
			 {
			"success": false,
			"message": "Failed to pay",
			"payload": null
			}

DELETE	/transaction/:id	 {
		"success": true,
		"message": "Transaction deleted",
		"payload": {
		"id": "7b030e25-cb79-45ad-a711-8d1b4f41b0ac",
		"user_id": "5a0e3f2c-4496-42cd-baaf-eff5202272bb",
		"item_id": "07f93385-f823-481f-9461-dcc60c663d18",
		"quantity": 1
			 {
			"success": false,
			"message": "Transaction not found",
			"payload": null
			}