

PRAKTIKUM SISTEM BASIS DATA

| | | | |
|-------------|--|------------------|-----------|
| Nama | Wiellona Darlene Oderia Saragih | No. Modul | 6 |
| NPM | 2306264396 | Tipe | CS |

1. Screenshot hasil

Kode POST (user/register)

```
const emailRegex = /^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9-]+\.[a-zA-Z]{2,}$/;
const passwordRegex = /^(?=.*\d)(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&]{8,}$/;

if (!email || !password || !name) {
  return baseResponse(
    res,
    false,
    400,
    "Email, password, and name are required",
    null
  );
}

if (!emailRegex.test(email)) {
  return baseResponse(res, false, 400, "Invalid email format", null);
}

if (!passwordRegex.test(password)) {
  return baseResponse(
    res,
    false,
    400,
    "Password must be at least 8 characters long, contain at least one number and one special character",
    null
  );
}
```

Kode PUT (user/)

```
// Regex patterns
const emailRegex = /^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9-]+\.[a-zA-Z]{2,}$/;
const passwordRegex = /^(?=.*\d)(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&]{8,}$/;
const nameRegex = /^[A-Za-z\s]{3,}$/;

// Validasi input
if (!id || !name || !email || !password) {
  return baseResponse(
    res,
    false,
    400,
    "ID, name, email, and password are required",
    null
  );
}

if (!emailRegex.test(email)) {
  return baseResponse(res, false, 400, "Invalid email format", null);
}

if (!passwordRegex.test(password)) {
  return baseResponse(
    res,
    false,
    400,
    "Password must be at least 8 characters long, contain at least one number and one special character",
    null
  );
}

if (!nameRegex.test(name)) {
  return baseResponse(
    res,
    false,
    400,
    "Name must be at least 3 characters and contain only letters and spaces",
    null
  );
}
```

| | Keterangan | Screenshot |
|----------------------|------------------------------------|---|
| POST (user/register) | Berhasil |  |
| | Invalid email format |  |
| | Password tidak sesuai requirements |  |
| PUT (user/) | Berhasil |  |
| | Invalid email error |  |
| | Password tidak sesuai requirements |  |

2. Screenshot

Kode POST (user/register)

```
try {
  const existingUser = await userRepository.getUserByEmail(email);
  if (existingUser) {
    return baseResponse(res, false, 400, "Email already used", null);
  }

  const hashedPassword = await bcrypt.hash(password, saltRounds);

  const user = await userRepository.createUser({
    email,
    password: hashedPassword,
    name,
  });
}
```

Kode PUT (user/)

//user.controller.js

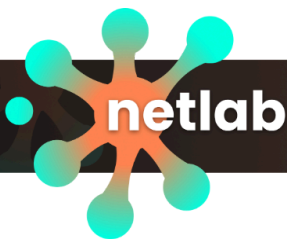
```
// Hash password sebelum update
const hashedPassword = await bcrypt.hash(password, saltRounds);

// Update user dengan password yang sudah di-hash
const updatedUser = await userRepository.updateUser({
  id,
  name,
  email,
  password: hashedPassword,
});
```

// user.repository.js (buat update hash password ke database)

```
exports.getUserById = async (id) => {
  try {
    const result = await db.query("SELECT * FROM users WHERE id = $1", [id]);
    return result.rows[0] || null;
  } catch (error) {
    console.error("Error getting user by ID:", error);
    return null;
  }
};
```

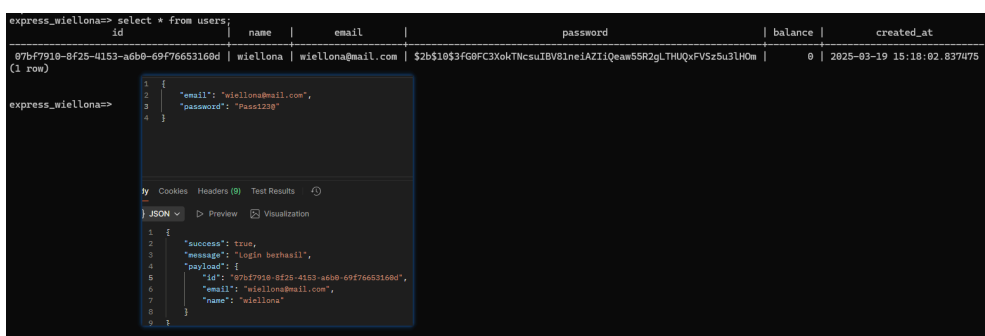
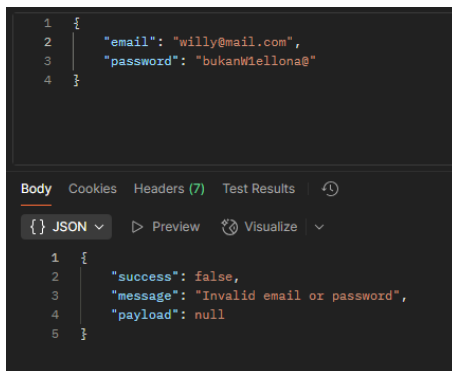
Hasil Database



```
express_wiellona=> select * from users;
id | name | email | password | balance | created_at
-----+-----+-----+-----+-----+-----
9758e642-f722-487a-879a-aa96e6fa5c7 | Edgrant | env@jangan.public.ya | $2b$10$1MSfySGXLU9r79c0GpNCLuUgXx90rQ3cdJi4AgzpzMMY5Vks0LLxu | 0 | 2025-03-13 08:53:40.338952
c6ad9678-c460-49f7-88aa-8baef1dc34f6 | wiellona | wielly@mail.com | $2b$10$sCBKKhBNmEcze6/Vvjtxke.g5I4gHAI/seqkb0TWMNDNQ/R0PsFa | 0 | 2025-03-19 04:30:23.634852
57cbbf9e-c832-42f3-8b2a-8d44ee3456ca | bukan wiellona | wiellonaupdate@mail.com | $2b$10$2ZMJJCZ0jTYoEeTG31ys75es0rwV0AvHses16EgoipWmv81C9zyRUe | 0 | 2025-03-19 04:11:36.265659
(3 rows)

express_wiellona=> select * from users;
SSL SYSCALL error: EOF detected
The connection to the server was lost. Attempting reset: Succeeded.
psql (17.2, server 17.4)
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off, ALPN: postgresql)
express_wiellona=> select * from users;
id | name | email | password | balance | created_at
-----+-----+-----+-----+-----+-----
9758e642-f722-487a-879a-aa96e6fa5c7 | Edgrant | env@jangan.public.ya | $2b$10$1MSfySGXLU9r79c0GpNCLuUgXx90rQ3cdJi4AgzpzMMY5Vks0LLxu | 0 | 2025-03-13 08:53:40.338952
57cbbf9e-c832-42f3-8b2a-8d44ee3456ca | bukan wiellona | wiellonaupdate@mail.com | $2b$10$2ZMJJCZ0jTYoEeTG31ys75es0rwV0AvHses16EgoipWmv81C9zyRUe | 0 | 2025-03-19 04:11:36.265659
c6ad9678-c460-49f7-88aa-8baef1dc34f6 | willy wonka | wonka123@mail.com | $2b$10$V0PQyR83oNSjCkrypiKGf.MSOoEjBaw7nH0B4dcHppx64PNMqOdSW | 0 | 2025-03-19 04:30:23.634852
(3 rows)
```

3. Screenshot



4. Screenshot

```
const express = require("express");
require("dotenv").config(); 6.3k (gzipped: 2.8k)

const app = express();
const port = process.env.PORT || 3000;
const cors = require("cors"); 4.5k (gzipped: 1.9k)

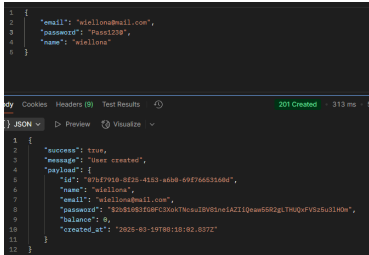
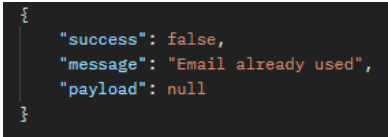
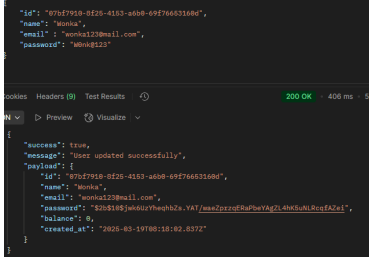
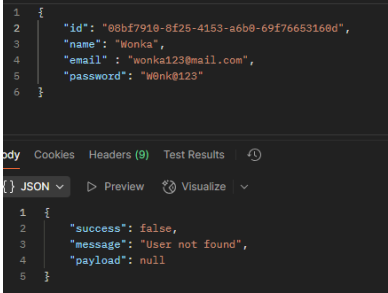
const corsOptions = {
  origin: "https://os.netlabdte.com",
  methods: ["GET", "POST", "PUT", "DELETE"],
};

app.use(cors(corsOptions));
```

5. Screenshot

```
express_wiellona=> CREATE type transaction_status AS ENUM ('pending', 'paid');
CREATE TYPE
express_wiellona=> CREATE TABLE IF NOT EXISTS transactions(
express_wiellona(> id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
express_wiellona(> user_id UUID NOT NULL,
express_wiellona(> item_id UUID NOT NULL,
express_wiellona(> quantity INT NOT NULL,
express_wiellona(> total INT NOT NULL,
express_wiellona(> status transaction_status DEFAULT 'pending',
express_wiellona(> created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
express_wiellona(> FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE,
express_wiellona(> FOREIGN KEY (item_id) REFERENCES items(id) ON DELETE CASCADE
express_wiellona(> );
CREATE TABLE
```

6. Screenshot response

| Method | Endpoint | Success | Failed |
|--------|----------------|--|---|
| POST | /user/register |  |  |
| PUT | /user |  |  |

| | | | |
|--------|---------------------|---|--|
| POST | /user/login | <pre> 1 { 2 "email": "wiellona@mail.com", 3 "password": "Pass123B" 4 } </pre> <p>Body Cookies Headers (9) Test Results</p> <p>JSON Preview Visualization</p> <pre> 1 { 2 "success": true, 3 "message": "Login berhasil", 4 "payload": { 5 "id": "07b7918-8f25-4153-a6b8-69f76653168d", 6 "email": "wiellona@mail.com", 7 "name": "wiellona" 8 } 9 } </pre> | <pre> { "success": false, "message": "Email atau password salah", "payload": null } </pre> |
| POST | /user/topUp | | |
| POST | /transaction/create | <pre> 1 { 2 "item_id": "3b2fe83d-ab88-4f65-b4b3-78985793e9af", 3 "quantity": 1, 4 "user_id": "07b7918-8f25-4153-a6b8-69f76653168d" 5 } </pre> <p>Body Cookies Headers (9) Test Results</p> <p>JSON Preview Visualize</p> <pre> 1 { 2 "success": true, 3 "message": "Transaction created", 4 "payload": { 5 "id": "049561e9-df97-4eef-a36a-3864ac898037", 6 "user_id": "07b7918-8f25-4153-a6b8-69f76653168d", 7 "item_id": "3b2fe83d-ab88-4f65-b4b3-78985793e9af", 8 "quantity": 1, 9 "total": 104000, 10 "status": "pending", 11 "created_at": "2025-03-19T09:03:48.922Z" 12 } 13 } </pre> | |
| POST | /transaction/pay | | |
| DELETE | /transaction/:id | | |