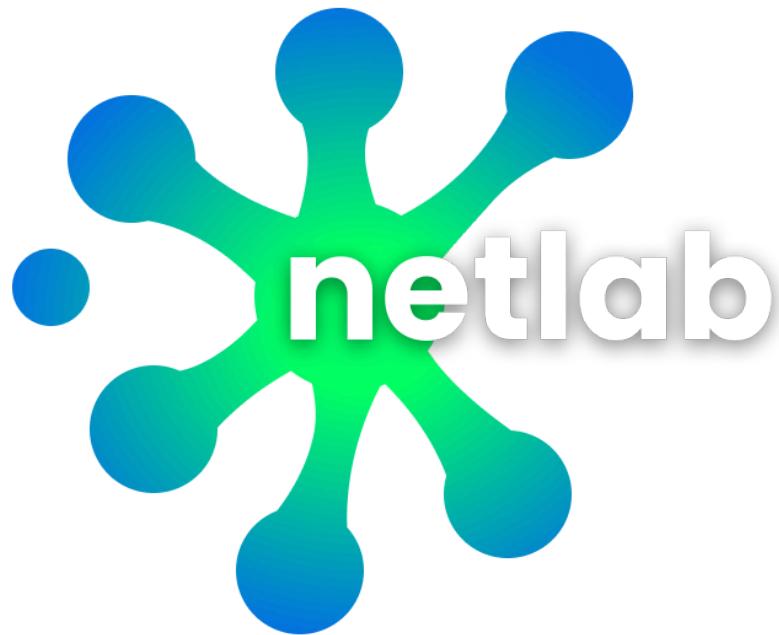


**PROYEK AKHIR  
KEAMANAN JARINGAN 2025**



**Ryan Adidaru 2306266994  
Fathan Yazid Satriani 2306250560**

# 1 Pendahuluan

Aplikasi web sederhana berbasis Node.js/Express yang menjadi objek Proyek Akhir ini menyediakan fitur autentikasi dasar dan pengelolaan file melalui endpoint upload. Tanpa kontrol keamanan yang memadai, fitur-fitur tersebut berpotensi dieksloitasi oleh pihak tidak berwenang. Laporan ini mendokumentasikan hasil pengujian keamanan yang dilakukan terhadap aplikasi tersebut, mencakup analisis risiko, pembuktian kerentanan, serta usulan mitigasi untuk menjaga keamanan layanan.

## 1.1 Tujuan

Mengidentifikasi dua kerentanan utama pada aplikasi, yaitu:

- Insecure File Upload
- Insecure Change Password (ganti password tanpa verifikasi password lama)

Menilai dampak terhadap kerahasiaan, integritas, dan keamanan data pengguna.

Menyusun rekomendasi perbaikan yang aplikatif untuk menutup kerentanan tersebut.

## 1.2 Ruang Lingkup

Ruang lingkup pengujian dalam proyek ini meliputi:

- Backend Node.js/Express yang dijalankan secara lokal.
- Fitur upload melalui endpoint /upload.
- Fitur autentikasi dan ganti password melalui endpoint /change-password.
- Pengujian hanya mencakup serangan berbasis logic flaw dan improper access control.
- Serangan DoS, social engineering, dan brute force tidak termasuk ruang lingkup.

## 1.3 Metodologi

Metodologi yang digunakan mengikuti pendekatan industri keamanan aplikasi web:

- Black-box testing dikombinasikan dengan review kode statis untuk memvalidasi logika otorisasi.
- Referensi utama berasal dari OWASP Web Security Testing Guide (WSTG).
- Pembuktian dilakukan melalui pengujian manual dan observasi perilaku server terhadap input berbahaya.

## 2 Ringkasan Eksekutif

Berdasarkan pengujian terhadap aplikasi, ditemukan dua kerentanan dengan risiko tinggi:

### 1. Insecure File Upload

Endpoint upload memperbolehkan semua jenis file diunggah tanpa validasi. Hal ini memungkinkan penyerang mengunggah file berbahaya seperti .exe, .js, atau script lainnya yang dapat digunakan untuk eskalasi serangan.

### 2. Insecure Change Password Feature

Fitur ganti password tidak mewajibkan pengguna memasukkan password lama. Akibatnya, siapa pun yang memiliki akses session (misal: melalui cookie yang dicuri) dapat mengganti password pemilik akun tanpa konfirmasi.

Kedua celah ini memungkinkan penyerang untuk menanam file berbahaya di server, melakukan kompromi akun pengguna, dan menciptakan risiko keamanan serius bagi sistem. Mitigasi yang diterapkan kemudian berhasil mencegah eksploitasi ulang.

## 3 Detail Kerentanan

### 3.1 Insecure File Upload

#### Deskripsi:

Endpoint **/upload** menerima file dari klien tanpa:

- Validasi ekstensi
- Validasi MIME type
- Pembatasan jenis file yang diperbolehkan
- Sanitasi terhadap file yang berpotensi berbahaya

Server menyimpan file apa pun yang dikirimkan, termasuk executable dan script.

#### Langkah Eksloitasi:

1. Login sebagai user.
2. Upload file dengan ekstensi .exe atau .js.
3. File diterima dan disimpan di server.

#### Dampak:

Penyerang dapat mengunggah file berbahaya yang berpotensi untuk eksloitasi seperti mengeksekusi kode di server, menjadi backdoor, menyimpan payload malware, dan menjadi pintu masuk untuk serangan lanjutan.

#### Proof Of Concept:

1. Upload file payload.js, akan diterima tanpa penolakan.
2. File muncul di direktori uploads/ meski bukan file gambar.

### 3.2 Insecure Change Password Feature

#### Deskripsi:

Fitur ganti password tidak meminta password lama sebagai syarat perubahan. Sebagai contoh, endpoint hanya menerima:

```
{ "newPassword": "<password baru>" }
```

tanpa memvalidasi bahwa pengguna benar-benar mengetahui password sebelumnya.

**Langkah Eksplorasi:**

1. Login sebagai user.
2. Akses halaman ganti password.
3. Masukkan password baru tanpa verifikasi password lama.

**Dampak:**

- Akun dapat diambil alih permanen.
- Integritas kredensial pengguna hilang.
- Risiko tinggi jika session dicuri melalui XSS, shoulder surfing, atau session hijacking.

**Proof Of Concept:**

- Ganti password tanpa field password lama pada page /change-password.
- Password berhasil diubah tanpa verifikasi.

## 4 Mitigasi dan Remediasi

### 4.1 Mitigasi Insecure File Upload

Untuk menutup celah insecure file upload yang memungkinkan berkas berbahaya diunggah ke dalam sistem, beberapa langkah mitigasi diterapkan pada sisi backend:

- Server kini menerapkan validasi tipe berkas secara ketat dengan memeriksa MIME type dan ekstensi file, sehingga hanya berkas gambar seperti .jpg, .jpeg, dan .png yang dapat diterima.
- Validasi dilakukan sepenuhnya di sisi server untuk mencegah manipulasi header pada request. Selain itu, server juga melakukan sanitasi nama file untuk menghindari karakter atau pola yang dapat digunakan untuk melakukan path traversal ataupun penyisipan perintah berbahaya.
- Selanjutnya, direktori penyimpanan file dibatasi pada folder non-eksekusi sehingga berkas yang berhasil diunggah tidak dapat dijalankan sebagai kode, mengurangi risiko remote code execution.
- Pembatasan ukuran file juga diterapkan untuk mencegah penyalahgunaan kapasitas penyimpanan atau upaya denial of service melalui unggahan file berukuran besar. Implementasi ini memastikan bahwa fitur unggah file hanya digunakan sesuai tujuan awalnya dan tidak dapat dimanfaatkan untuk menanamkan payload berbahaya ke dalam sistem.

### 4.2 Mitigasi Change Password Feature

Untuk mencegah penyalahgunaan fitur ganti kata sandi yang sebelumnya tidak mewajibkan verifikasi kata sandi lama, server kini mewajibkan pengguna memasukkan current password sebelum melakukan perubahan. Validasi dilakukan dengan membaca hash kata sandi dari basis data dan membandingkannya dengan input pengguna, sehingga perubahan hanya dapat dilakukan oleh pemilik akun yang memiliki kredensial yang sah. Pendekatan ini menghilangkan risiko pengambilalihan akun yang terjadi apabila penyerang memperoleh token sesi tanpa mengetahui kata sandi utama.

Selain itu, pembatasan hak akses juga diterapkan pada proses penggantian kata sandi. Pengguna non-admin hanya diperbolehkan mengganti kata sandi miliknya sendiri, terlepas dari parameter atau user\_id apa pun yang dikirimkan dari sisi klien. Server sepenuhnya mengabaikan parameter yang dapat dimanipulasi oleh klien dan hanya menggunakan identitas pengguna berdasarkan token autentikasi yang telah diverifikasi. Dengan penerapan mitigasi tersebut, fitur perubahan kata sandi kini lebih aman dan tidak lagi menjadi titik masuk yang mudah untuk eskalasi serangan ataupun pengambilalihan akun secara tidak sah.

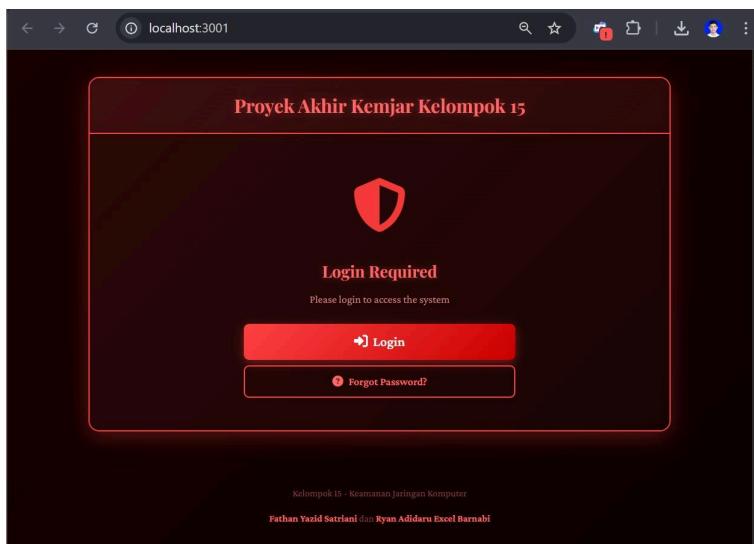
## Kesimpulan

Aplikasi web yang diuji terbukti memiliki dua celah keamanan kritis, insecure file upload dan insecure change password. Kedua *vulnerability* tersebut memungkinkan penyerang mengunggah file berbahaya serta mengambil alih akun tanpa perlu mengetahui kata sandi lama. Melalui pengujian berbasis black-box dan review logika backend, risiko terhadap kerahasiaan, integritas, dan kontrol akses pengguna berhasil dipetakan dengan jelas. Setelah diterapkan mitigasi seperti validasi tipe file, sanitasi nama berkas, pembatasan direktori penyimpanan, serta verifikasi kata sandi lama pada proses perubahan password, aplikasi kini memiliki perlindungan yang jauh lebih kuat dan tidak lagi rentan dieksloitasi melalui kedua celah tersebut.

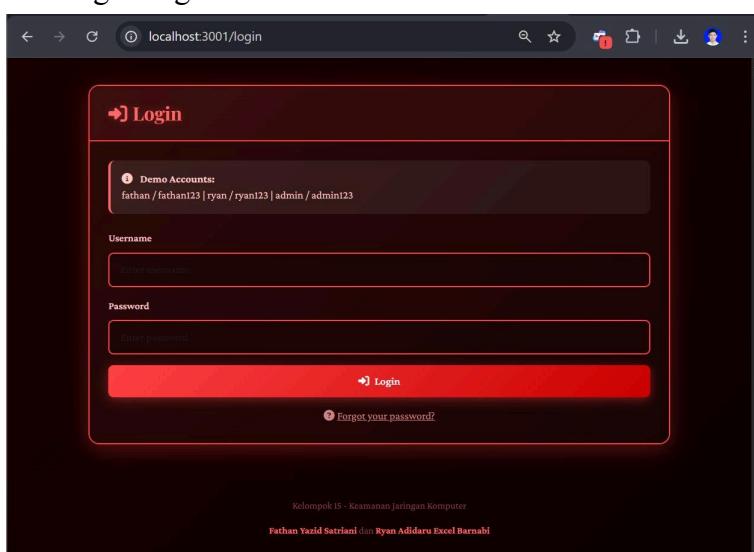
## 6 Lampiran

Walkthrough vulnerable website

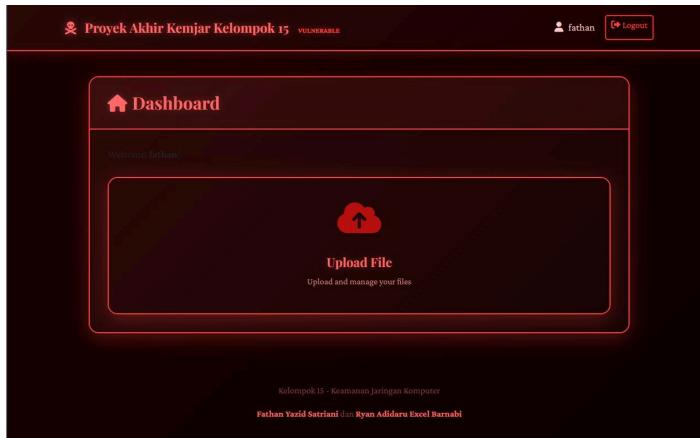
### 6.1. Landing Page



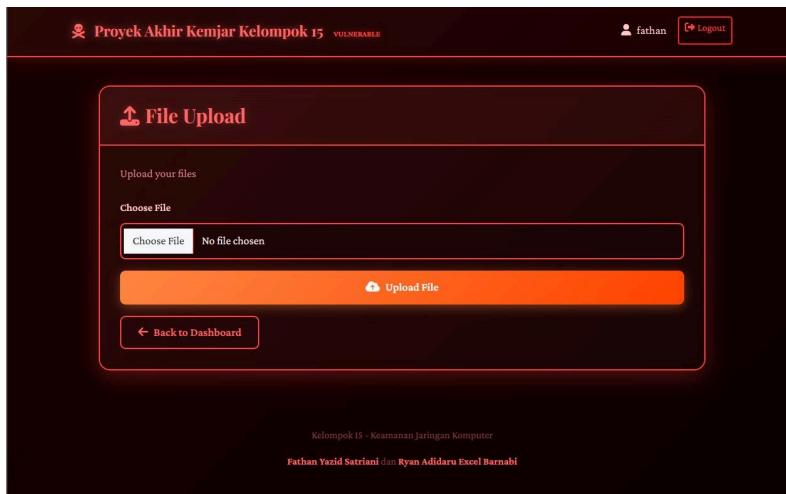
### 6.2. Login Page



### 6.3. File Upload Page

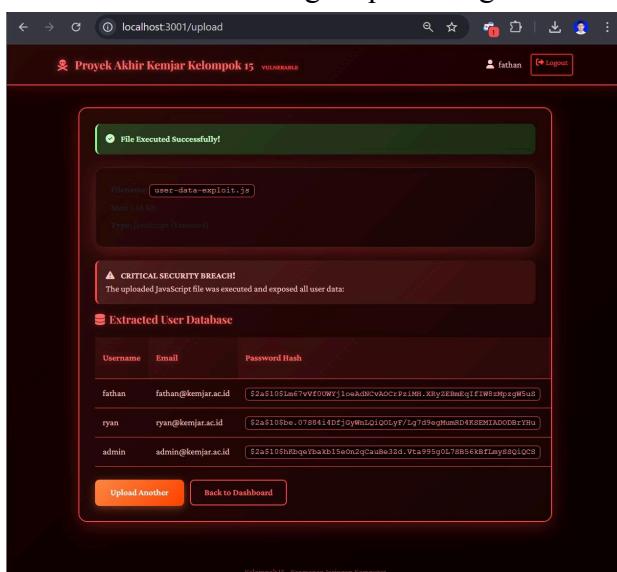


The screenshot shows a dark-themed web application interface. At the top, it displays "Proyek Akhir Kemjar Kelompok 15" and "VULNERABLE". A user profile for "fathan" is shown with a "Logout" button. The main area is titled "Dashboard" and features a large "Upload File" button with a cloud icon. Below the button, the text "Upload and manage your files" is visible. At the bottom of the dashboard, there is copyright information: "Kelompok 15 - Keamanan Jaringan Komputer" and "Fathan Yazid Satriani dan Ryan Adidara Excel Barnabi".

This screenshot shows the same application after navigating to the "File Upload" section. The title bar remains the same. The main content area has a "Choose File" input field containing "No file chosen" and an orange "Upload File" button with a cloud icon. Below these buttons is a "Back to Dashboard" link. The footer information at the bottom is identical to the previous screenshot.

### 6.4. File Execute Through Upload Page



This screenshot shows the result of executing a file through the upload feature. A green success message box at the top says "File Executed Successfully!". Below it, a terminal-like window shows the command "node user-data-exploit.js" and its output, which includes a warning about a critical security breach. A red warning box states: "The uploaded JavaScript file was executed and exposed all user data:". Below this, a section titled "Extracted User Database" lists user information in a table:

Username	Email	Password Hash
fathan	fathan@kemjar.ac.id	\$2a\$10\$Slm67vvf0UWY1owAHCvAOCrPzIMH.XxyxxBmrgfI1W8zMpzgk5ub
ryan	ryan@kemjar.ac.id	\$2a\$10\$be..07B8414DfjGy6nLQ1Q5yF/5q7dSeqMm8D4XSEMIAD0S8rYRq
admin	admin@kemjar.ac.id	\$2a\$10\$ShRbqeabkb15eOn2gCaBe33d.Vta995gOL78B56kfIay8SQLQCB

At the bottom of the page are "Upload Another" and "Back to Dashboard" buttons. The footer information is present at the very bottom.



## 6.5. Forgot Password Page and Exploitation with otp-bruteforce.js

The screenshot shows a web browser window with the URL `localhost:3001/forgot-password`. The page has a dark red theme. At the top left is a feathered pen icon followed by the text "Forgot Password". Below this is a form with a placeholder "Enter your username to reset your password". A large red rectangular button at the bottom contains the text "Resend Password Reset" in white. Below this button is a link "← Back to Login".

The screenshot shows a web browser window with the URL `localhost:3001/forgot-password`. The main content is a form titled "Verification Code Sent". A green success message box contains the text: "A 4-digit verification code has been sent to [fahim@kempic.ac.id](#)". Below this, a red error message box contains the text: "For Demo: Check the console for OTP code: `2142`". The form itself has a yellow header bar with the text "Verify & Reset Password". There are two input fields: one for "Enter the 4-digit code sent to your email!" and another for "New Password". At the bottom right of the form is a link labeled "Forgot Code".

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Yaftha\OneDrive\Documents\TugasKuliah\5Semester5\Kemjar\ProjekAkhir_Kelompok15_Kemjar> node exploit-examples/otp-bruteforce.js fathan Hacked@password123

[+] ATTACK SUCCESSFUL! []

④ Correct OTP Found: 6108
④ Time Elapsed: 3.23 seconds
④ Total Attempts: 1000
④ Attack Rate: 1894 attempts/second
④ Success Rate: 61.09%
[+] IMPACT:
    ✓ Password successfully changed without knowing the GIP
    ✓ Account successfully retrieved through brute force
    ✓ Failed rate limiting prevented the attack
    ✓ No account lockout mechanism exists
    ✓ Attacker can now login as: fathan
    ✓ New password: Hacked@password123

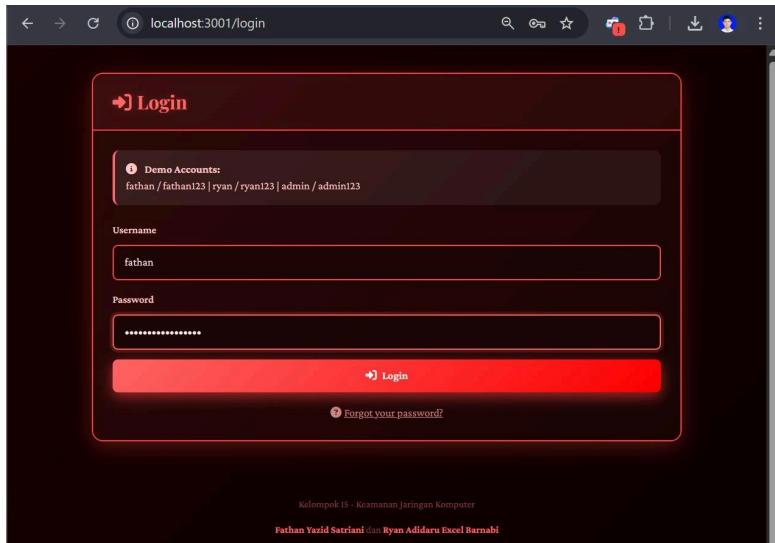
[+] VULNERABILITY DETAILS:
1. GIP is only 4 digits (10,000 possibilities)
2. No rate limiting on verification attempts
3. No account lockout after failed attempts
4. No CAPTCHAs to prevent automation
5. No GIP expiration time

[+] RECOMMENDATION:
1. Implement rate limiting (max 3-5 attempts)
2. Add account lockout after failed attempts
3. Use longer GIP codes (6-8 digits)
4. Implement CAPTCHA for verification attempts
5. Add GIP expiration (5-10 minutes)
6. Log and alert on suspicious activity
7. Add delay between verification attempts

[+] NEXT STEPS:
1. Login To https://localhost:3001/login
2. Enter: fathan
3. Password: Hacked@password123
4. Access granted! Account compromised.

% PS C:\Users\Yaftha\OneDrive\Documents\TugasKuliah\5Semester5\Kemjar\ProjekAkhir_Kelompok15_Kemjar>
```

## 6.6. Exploitation Successfully Attempted



localhost:3001/login

**Login**

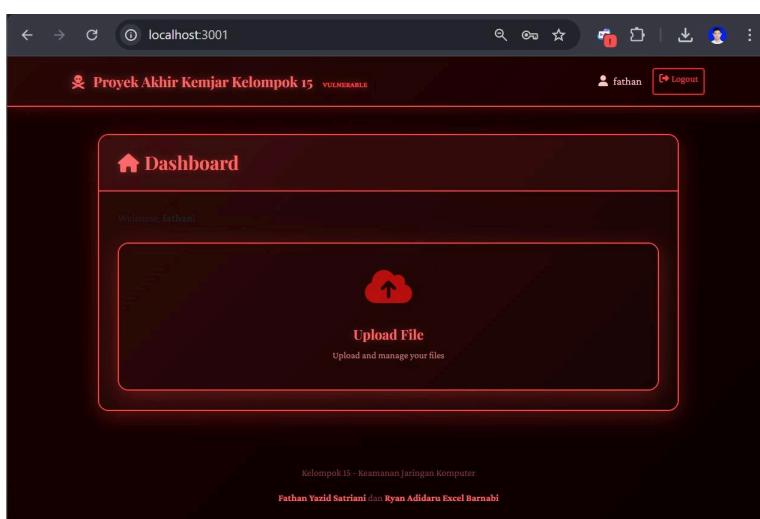
Demo Accounts:  
fathan / fathan123 | ryan / ryan123 | admin / admin123

Username  
fathan

Password  
\*\*\*\*\*

**Login**

[Forgot your password?](#)



localhost:3001

**Dashboard**

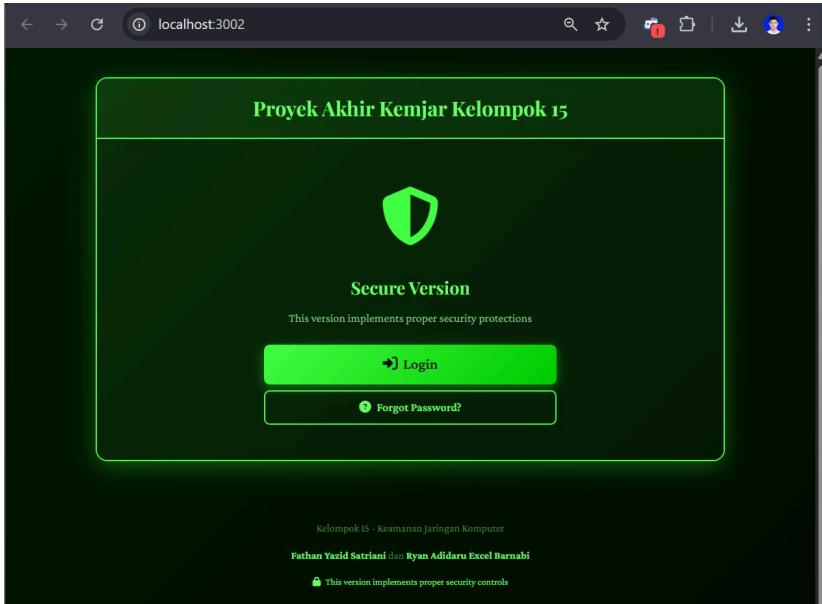
Welcome, fathan!

**Upload File**  
Upload and manage your files

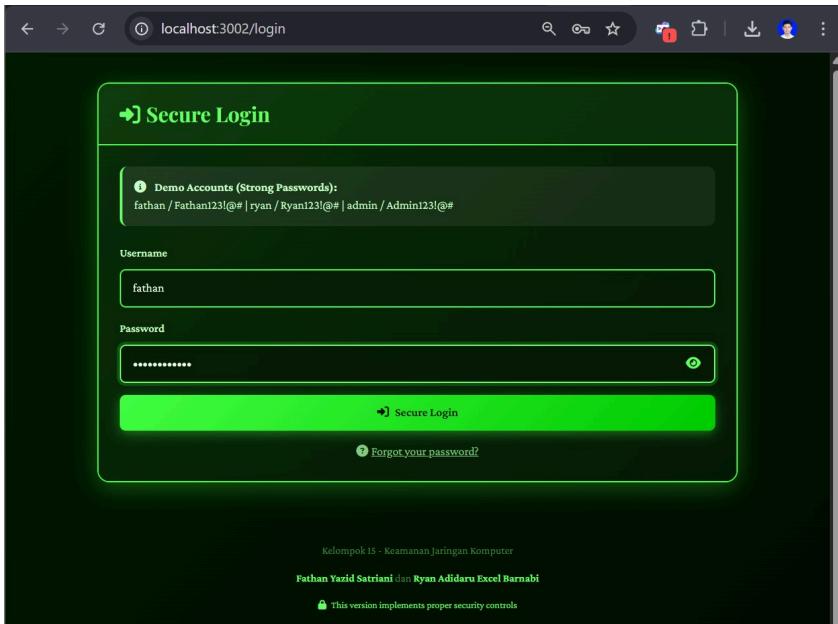
Proyek Akhir Kemjar Kelompok 15 VULNERABLE

fathan Logout

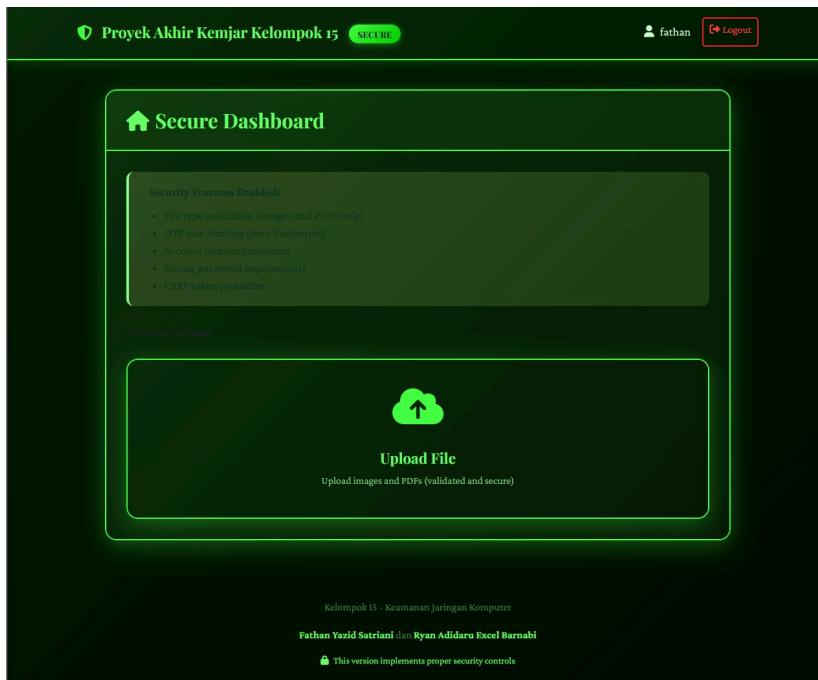
## 6.7 Secured Landing Page



## 6.8 Secured Login Page



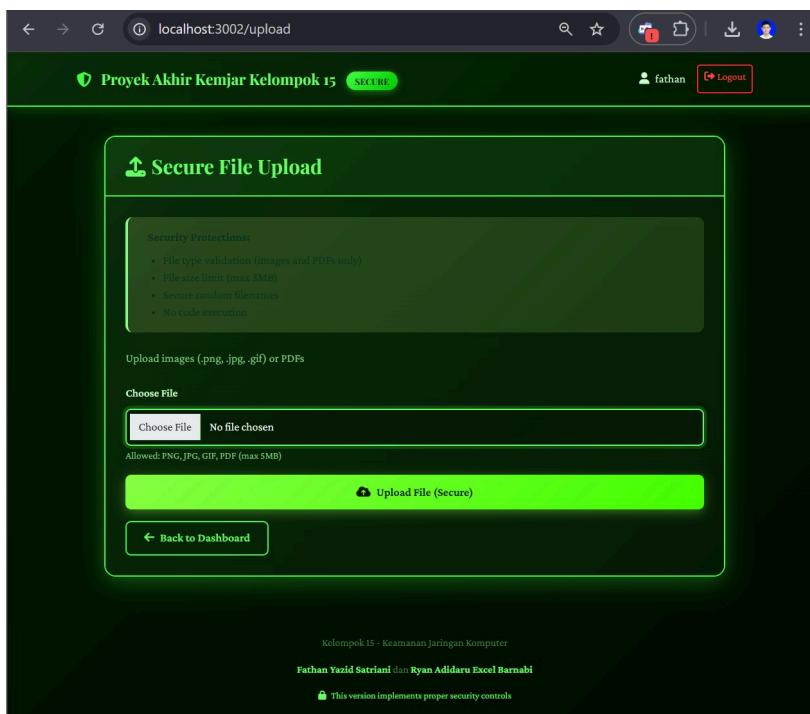
## 6.9 Secured Dashboard and File Upload



The screenshot shows a secure dashboard interface. At the top, there's a header bar with the project name "Proyek Akhir Kemarj Kelompok 15" and a "SECURE" badge. On the right, it shows a user profile for "fathan" and a "Logout" button. Below the header is a section titled "Secure Dashboard" with a house icon. It displays a list of "Security Features Enabled:":

- File type validation (images and PDFs only)
- OTP rate limiting (max 5 attempts)
- Account lockout protection
- Strong password requirements
- CSRF token protection

A welcome message "Welcome fathan!" is shown below. The main area features a large cloud icon with an upward arrow, labeled "Upload File". A sub-instruction "Upload images and PDFs (validated and secure)" is provided. At the bottom, there's footer information: "Kelompok 15 - Keamanan Jaringan Komputer", names "Fathan Yazid Satriani dan Ryan Adidaru Excel Barnabi", and a note "This version implements proper security controls".

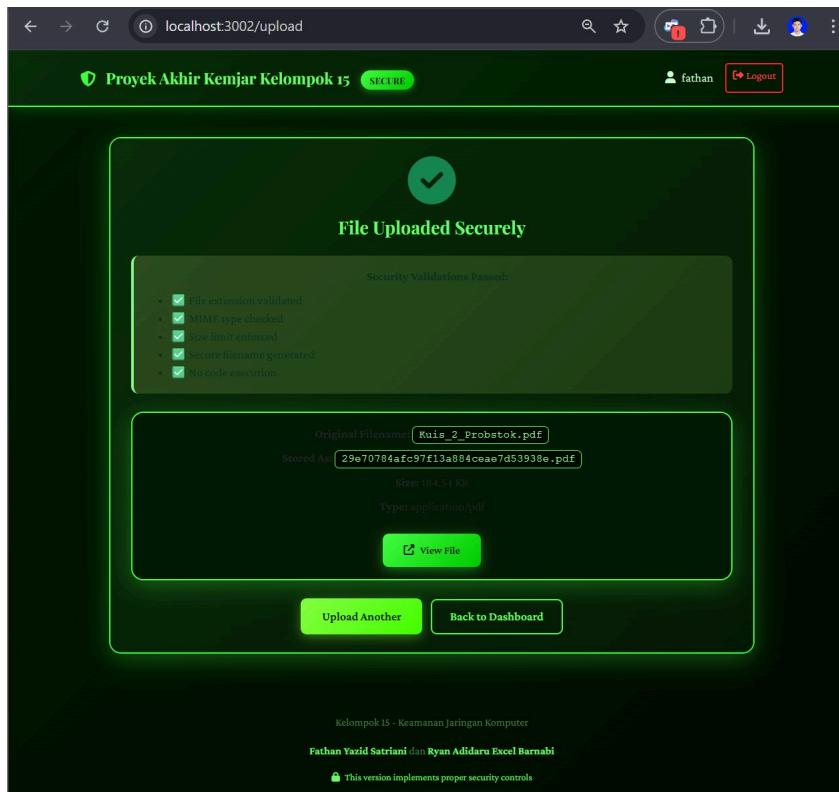


The screenshot shows a secure file upload page. The browser address bar indicates the URL is "localhost:3002/upload". The page has a header with the project name and a "SECURE" badge, along with a user profile for "fathan" and a "Logout" button. The main content area is titled "Secure File Upload" with an upward arrow icon. It contains a "Security Protections:" section listing:

- File type validation (images and PDFs only)
- File size limit (max 5MB)
- Secure random filenames
- No code execution

Below this, there's a "Choose File" input field showing "No file chosen" and a note "Allowed: PNG, JPG, GIF, PDF (max 5MB)". A prominent green "Upload File (Secure)" button is at the bottom. A "Back to Dashboard" link is also present. The footer is identical to the dashboard, mentioning "Kelompok 15 - Keamanan Jaringan Komputer", "Fathan Yazid Satriani dan Ryan Adidaru Excel Barnabi", and "This version implements proper security controls".

## 6.10 Proof of Secured Mechanisms (Secure File Upload and OTP Rate Limit Mechanism)



The screenshot shows a web application interface for a file upload. At the top, there's a navigation bar with icons for back, forward, search, and user profile. The URL is `localhost:3002/upload`. Below the header, a banner says "Proyek Akhir Kemjari Kelompok 15" and has a green "SECURE" button. On the right, it shows a user profile for "fathan" and a "Logout" button.

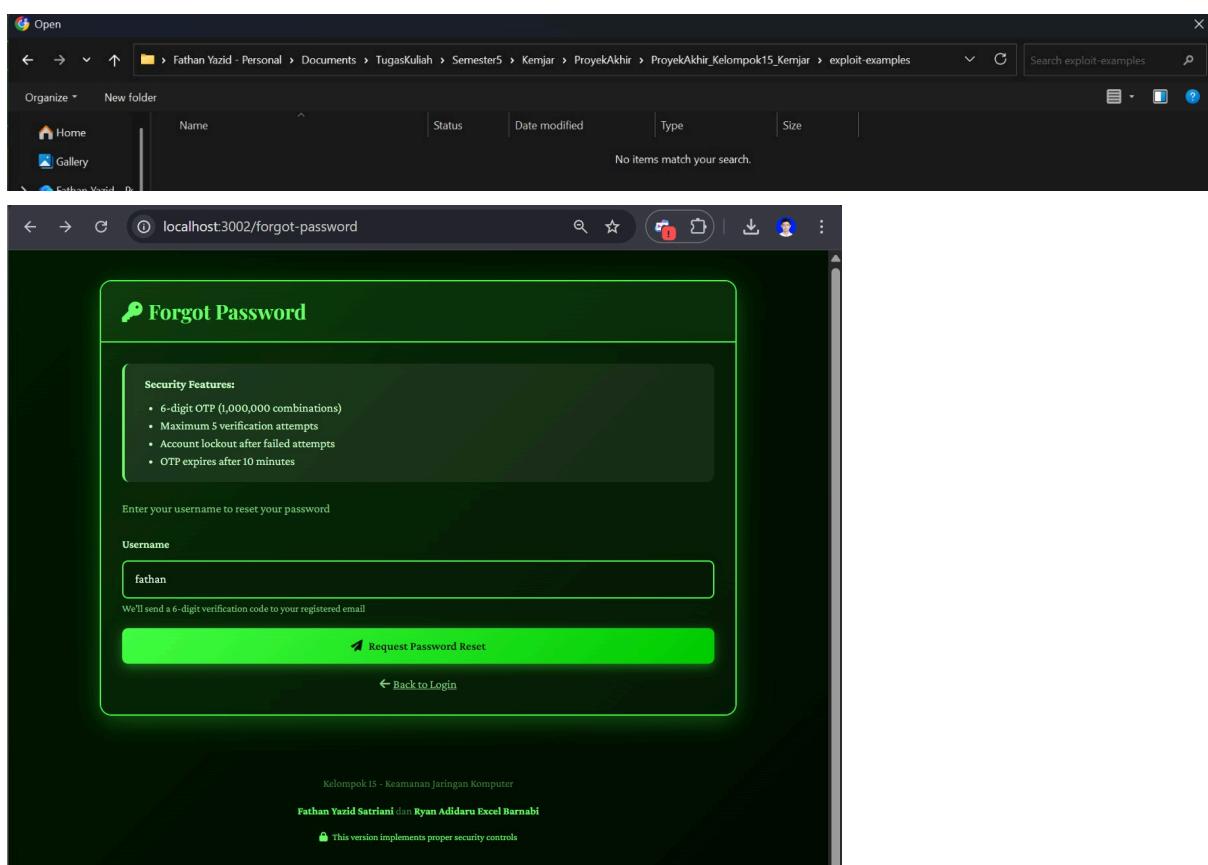
The main content area has a green background with white text. It displays a large green checkmark icon and the message "File Uploaded Securely". Below this, a section titled "Security Validations Passed:" lists several items with green checkmarks:

- File extension validated
- MIME type checked
- Size limit enforced
- Secure filename generated
- No code execution

Below the validation section, there's a preview of the uploaded file: "Original Filename: Kuis\_2\_Probstok.pdf", "Stored As: 29e70784afc97f13a884ceae7d5393e.pdf", "Size: 184.51 KB", and "Type: application/pdf". A "View File" button is present.

At the bottom of the main content area are two buttons: "Upload Another" and "Back to Dashboard".

At the very bottom of the page, there's footer text: "Kelompok 15 - Keamanan Jaringan Komputer", "Fathan Yazid Satriani dan Ryan Adidaru Excel Barnabi", and a note about security controls: "This version implements proper security controls".



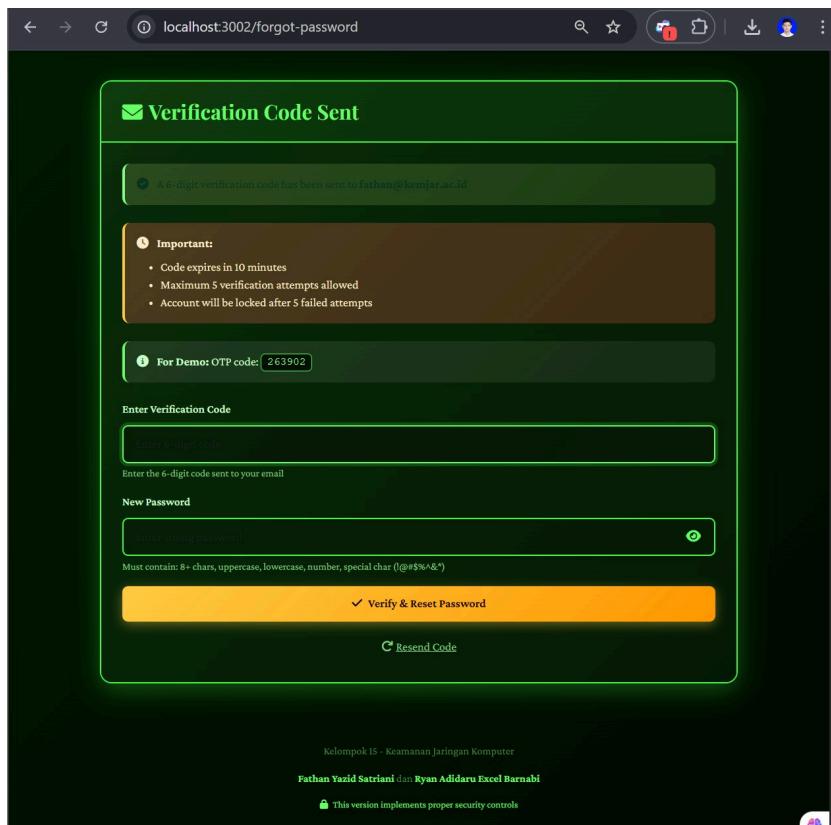
The screenshot shows a password reset form. At the top, there's a navigation bar with icons for back, forward, search, and user profile. The URL is `localhost:3002/forgot-password`. Below the header, there's a banner with a key icon and the text "Forgot Password".

The main content area has a green background with white text. It displays a section titled "Security Features:" with the following list:

- 6-digit OTP (1,000,000 combinations)
- Maximum 5 verification attempts
- Account lockout after failed attempts
- OTP expires after 10 minutes

Below this, there's a text input field labeled "Enter your username to reset your password" and a placeholder "Username" with the value "fathan". A note below the input field says "We'll send a 6-digit verification code to your registered email". A large green "Request Password Reset" button is at the bottom of the form. To its left is a "Back to Login" link.

At the very bottom of the page, there's footer text: "Kelompok 15 - Keamanan Jaringan Komputer", "Fathan Yazid Satriani dan Ryan Adidaru Excel Barnabi", and a note about security controls: "This version implements proper security controls".



A 6-digit verification code has been sent to farhan@kemjar.ac.id

**Important:**

- Code expires in 10 minutes
- Maximum 5 verification attempts allowed
- Account will be locked after 5 failed attempts

**For Demo: OTP code: 263902**

Enter Verification Code

Enter 6-digit code

Enter the 6-digit code sent to your email

New Password

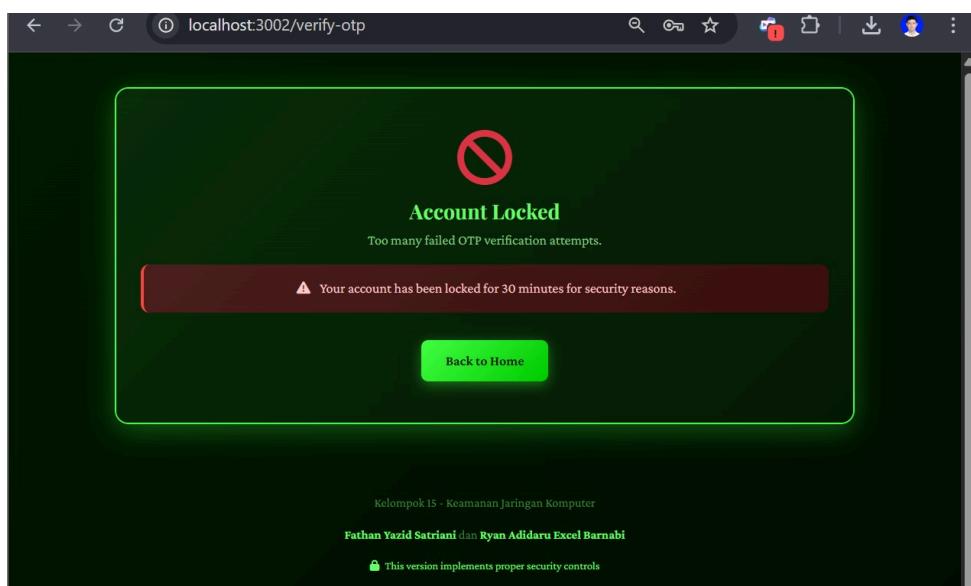
Enter strong password

Must contain: 8+ chars, uppercase, lowercase, number, special char (@#\$%^&\*)

✓ Verify & Reset Password

↻ Resend Code

```
PS C:\Users\fatha\OneDrive\Documents\TugasKuliah\Semester5\Kemjar\ProyekAkhir\ProyekAkhir_Kelompok15_Kemjar>
PS C:\Users\fatha\OneDrive\Documents\TugasKuliah\Semester5\Kemjar\ProyekAkhir\ProyekAkhir_Kelompok15_Kemjar> curl http://localhost:3002/api/users
curl : Cannot GET /api/users
At line:1 char:1
+ curl http://localhost:3002/api/users
+ ~~~~~
+ CategoryInfo          : InvalidOperation: (System.Net.HttpWebRequest:HttpWebRequest) [Invoke-WebRequest], WebException
+ FullyQualifiedErrorMessage : WebCmdletWebResponseException,Microsoft.PowerShell.Commands.InvokeWebRequestCommand
✖ PS C:\Users\fatha\OneDrive\Documents\TugasKuliah\Semester5\Kemjar\ProyekAkhir\ProyekAkhir_Kelompok15_Kemjar>
```



Account Locked

Too many failed OTP verification attempts.

⚠ Your account has been locked for 30 minutes for security reasons.

Back to Home

Kelompok 15 - Keamanan Jaringan Komputer

Fathan Yazid Satriani dan Ryan Adidaru Excel Barnabi

🔒 This version implements proper security controls

6.11 Link Video

[https://drive.google.com/drive/folders/12YplhYWc\\_Lk6BYItYMYrwENcAkFTk1tp?usp=sharing](https://drive.google.com/drive/folders/12YplhYWc_Lk6BYItYMYrwENcAkFTk1tp?usp=sharing)

6.12 Link Github

[https://github.com/coolcmyk/ProyekAkhir\\_Kelompok15\\_Kemjar](https://github.com/coolcmyk/ProyekAkhir_Kelompok15_Kemjar)