**PROTOKOL OTORISASI DAN AUTENTIKASI PADA WEB STATIS**

**TUGAS AKHIR**

**Diajukan Sebagai Salah Satu Syarat untuk Memperoleh Gelar Sarjana Terapan Teknik Informatika (S.Tr. Kom.) pada Program Pendidikan D4 Teknik Informatika**



**Oleh:**

**MUHAMMAD RIFQI DAFFA ULHAQ**

**1204045**

**PROGRAM DIPLOMA IV TEKNIK INFORMATIKA SEKOLAH VOKASI**

**UNIVERSITAS LOGISTIK DAN BISNIS INTERNASIONAL BANDUNG**

**2024**

**PROTOKOL OTORISASI DAN AUTENTIKASI PADA WEB STATIS**

**TUGAS AKHIR**

**Diajukan Sebagai Salah Satu Syarat untuk Memperoleh Gelar Sarjana Terapan Teknik Informatika (S.Tr. Kom.) pada Program Pendidikan D4 Teknik Informatika**



**Oleh:**

**MUHAMMAD RIFQI DAFFA ULHAQ**

**1204045**

**PROGRAM DIPLOMA IV TEKNIK INFORMATIKA SEKOLAH VOKASI**

**UNIVERSITAS LOGISTIK DAN BISNIS INTERNASIONAL BANDUNG**

**2024**

# HALAMAN PERNYATAAN ORISINALITAS

Yang bertanda tangan dibawah ini:

Nama : Muhammad Rifqi Daffa Ulhaq  
NPM : 1204045  
Program Studi : D4 Teknik Informatika  
Perguruan Tinggi : Universitas Logistik dan Bisnis Internasional

Dengan ini menyatakan bahwa tugas akhir yang telah saya buat dengan judul: “**Protokol Otorisasi Dan Autentikasi Pada Web Statis”** adalah asli (orisinil) dan belum pernah diterbitkan/dipublikasikan dimanapun dan dalam bentuk apapun. Demikianlah surat pernyataan ini saya buat dengan sebenar-benarnya tanpa ada paksaan dari pihak manapun juga. Apabila dikemudian hari ternyata saya memberikan keterangan palsu dan atau ada pihak lain yang mengklaim bahwa tugas akhir yang telah saya buat adalah hasil karya milik seseorang atau badan tertentu, saya bersedia diproses baik secara pidana maupun perdata dan kelulusan saya dari Universitas Logistik dan Bisnis Internasional dicabutkan/dibatalkan.

|  |
| --- |
| Dibuat di : Bandung |
| Pada tanggal. : Juli 2024 |
| Yang menyatakan, |
|  |
| Materai Rp. 10000 |
| Muhammad Rifqi Daffa Ulhaq |

# LEMBAR PENGESAHAN

**PROTOKOL OTORISASI DAN AUTENTIKASI PADA WEB STATIS**

NPM(1.20.4.045) MUHAMMAD RIFQI DAFFA ULHAQ

Laporan Program Tugas Akhir ini telah diperiksa, disetujui dan disidangkan di Bandung, (tanggal sidang)

Oleh :

|  |  |
| --- | --- |
| Penguji Utama | Penguji Pendamping |
|  |  |
| Rolly Maulana Awangga, S.T., MT., CAIP, SFPC  NIK. 117.86.219 | Nama Dosen Penguji Pendamping  NIK. |
|  | |
| Menyetujui,  Koordinator Tugas Akhir | |
|  | |
| M. Yusril Helmi Setyawan, S. Kom., M. Kom., SFPC.  NIK. 113.74.163 | |

# LEMBAR PENGESAHAN

**PROTOKOL OTORISASI DAN AUTENTIKASI PADA WEB STATIS**

NPM(1.20.4.045) MUHAMMAD RIFQI DAFFA ULHAQ

Laporan Program Tugas Akhir ini telah diperiksa, disetujui dan disidangkan di Bandung, (tanggal sidang)

Oleh :

|  |  |
| --- | --- |
| Penguji Utama | Penguji Pendamping |
|  |  |
| Rolly Maulana Awangga, S.T., MT., CAIP, SFPC  NIK. 117.86.219 | Nama Dosen Penguji Pendamping  NIK. |
|  | |
| Menyetujui,  Ketua Program Studi D-IV Teknik Informatika | |
|  | |
| Roni Andarsyah, S.T., M. Kom., SFPC.  NIK. 115.88.193 | |

# KATA PENGANTAR

Penulis panjatkan segala puji terhadap Tuhan Yang Maha Esa atas berkat-Nya, yang memungkinkan penyelesaian laporan tugas akhir tanpa kendala yang berarti. Keberhasilan dalam penulisan laporan ini merupakan hasil dari dukungan dari berbagai pihak yang memberikan masukan untuk meningkatkannya. Kemudian, penulis mengucapkan banyak terima kasih terhadap semua pihak yang telah memberikan kontribusi:

1. Allah Yang Maha Esa dengan limpahan berkat-Nya, penulis berhasil membuat laporan ini dengan baik.
2. Orang tua dan rekan-rekan yang senantiasa memberikan doa dan dukungan.
3. Bapak Rolly Maulana Awangga, S.T., MT., CAIP, SFPC selaku pembimbing internal.
4. Bapak M. Yusril Helmi Setyawan, S. Kom., M. Kom., SFPC selaku pembimbing internal.
5. Bapak Roni Andarsyah, S.T., M. Kom., SFPC selaku Kepala Program Studi DIV Teknik Informatika Universitas Logistik Bisnis Internasional.
6. Bapak M. Yusril Helmi Setyawan, S. Kom., M. Kom., SFPC selaku Koordinator tugas akhir tahun 2024/ 2025.

Penulis menyadari bahwa penyusunan laporan tugas akhir ini masih memiliki ruang untuk peningkatan. Kritik dan saran sangat diharapkan dan dapat disampaikan melalui penulis.

# HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Logistik dan Bisnis Internasional (ULBI), saya yang bertanda tangan di bawah ini:

|  |  |
| --- | --- |
| Nama : | Muhammad Rifqi Daffa Ulhaq |
| Npm : | 1204045 |
| Program Studi : | D4 Teknik Informatika |
| Jenis karya : | Tugas Akhir |

Demi pengembangan ilmu pengetahuan, memberikan kepada ULBI Hak Bebas Royalti Noneksklusif (Non-exclusive Royalty- Free Right) atas karya ilmiah saya yang berjudul:

**“Protokol Otorisasi Dan Autentikasi Pada Web Statis”**

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non eksklusifini Universitas Logistik dan Bisnis Internasional berhak menyimpan, mengalihmedia/format- kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan Tugas Akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

|  |
| --- |
| Dibuat di : Bandung |
| Pada tanggal. : Juli 2024 |
| Yang menyatakan, |
|  |
| Materai Rp. 10000 |
|  |
| Muhammad Rifqi Daffa Ulhaq |

# ABSTRAK

Keamanan dalam autentikasi pengguna adalah aspek krusial dalam pengembangan sistem *web*, terutama *web* statis yang sering kali kurang aman dibandingkan *web* dinamis. Penelitian ini mengembangkan sistem autentikasi yang menggabungkan QR *code*, OAuth2 (Google), dan *secure* *temporary* *password* (STP) untuk meningkatkan keamanan *web* statis. Kombinasi ini diharapkan memberikan solusi autentikasi berlapis yang lebih kuat dalam mencegah akses tidak sah, dengan QR *code* memanfaatkan perangkat *mobile*, OAuth2 menawarkan autentikasi token tanpa menyimpan kata sandi, dan STP memberikan keamanan tambahan dengan kata sandi sementara. Selain itu, teknik CAPTCHA dan *rate* *limiting* diterapkan untuk mencegah serangan otomatis dan memantau permintaan mencurigakan. Evaluasi sistem melibatkan pengujian *brute* *force* dan *dictionary* *attack*, menunjukkan ketidakberhasilan serangan ini dalam menebak kata sandi. Implementasi dan evaluasi menunjukkan peningkatan signifikan dalam keamanan autentikasi *web* statis, mengatasi kelemahan pendekatan konvensional dan menawarkan pendekatan yang dapat diadopsi pengembang *web* statis untuk meningkatkan perlindungan data dan pengguna.

Keywords: Keamanan Web, Autentikasi, Otorisasi, Web Statis.

# ABSTRACT

*Security in user authentication is a crucial aspect in web system development, especially static web which is often less secure than dynamic web. This research develops an authentication system that combines QR code, OAuth2 (Google), and secure temporary password (STP) to improve static web security. This combination is expected to provide a layered authentication solution that is stronger in preventing unauthorized access, with QR codes utilizing mobile devices, OAuth2 offering token authentication without storing passwords, and STP providing additional security with temporary passwords. In addition, CAPTCHA and rate limiting techniques are implemented to prevent automated attacks and monitor suspicious requests. System evaluation involved brute force and dictionary attack, showing the unsuccessfulness of these attacks in guessing passwords. The implementation and evaluation show significant improvements in static web authentication security, addressing the weaknesses of conventional approaches and offering an approach that static web developers can adopt to improve data and user protection.*

*Keywords: Web Security, Authentication, Authorization, Static Web.*

# DAFTAR ISI

[HALAMAN PERNYATAAN ORISINALITAS i](#_Toc173379972)

[LEMBAR PENGESAHAN ii](#_Toc173379973)

[LEMBAR PENGESAHAN iii](#_Toc173379974)

[KATA PENGANTAR iv](#_Toc173379975)

[HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS v](#_Toc173379976)

[ABSTRAK vi](#_Toc173379977)

[ABSTRACT vii](#_Toc173379978)

[DAFTAR ISI viii](#_Toc173379979)

[DAFTAR TABEL x](#_Toc173379980)

[DAFTAR GAMBAR xi](#_Toc173379981)

[DAFTAR LAMPIRAN xii](#_Toc173379982)

[BAB 1 1](#_Toc173379983)

[PENDAHULUAN 1](#_Toc173379984)

[1. 1 Latar Belakang 1](#_Toc173379985)

[1. 2 Identifikasi Masalah 2](#_Toc173379986)

[1. 3 Tujuan dan Manfaat 2](#_Toc173379987)

[1. 4 Ruang Lingkup 3](#_Toc173379988)

[BAB II 4](#_Toc173379989)

[LANDASAN TEORI 4](#_Toc173379990)

[2. 1 Deskripsi Topik Yang Sama 4](#_Toc173379991)

[2. 2 Deskripsi Metode Yang Sama 4](#_Toc173379992)

[BAB III 6](#_Toc173379993)

[METODOLOGI PENELITIAN 6](#_Toc173379994)

[3. 1 Metodologi Penelitian 6](#_Toc173379995)

[3. 2 Tahapan Tahapan Diagram Metodologi Penelitian 9](#_Toc173380010)

[BAB IV 11](#_Toc173380011)

[EKSPERIMEN DAN HASIL 11](#_Toc173380012)

[4. 1 Eksperimen 11](#_Toc173380013)

[4. 2 Hasil 15](#_Toc173380014)

[BAB V 33](#_Toc173380015)

[KESIMPULAN 33](#_Toc173380016)

[5. 1 Kesimpulan Masalah 33](#_Toc173380017)

[5. 2 Kesimpulan Metode 34](#_Toc173380018)

[5. 3 Kesimpulan Eksperimen 34](#_Toc173380019)

[BAB VI 36](#_Toc173380020)

[SARAN 36](#_Toc173380021)

[DAFTAR PUSTAKA 38](#_Toc173380022)

# DAFTAR TABEL

# DAFTAR GAMBAR

[Gambar 3. 1 Autentikasi QR Code 7](#_Toc173488851)

[Gambar 3. 2 Autentikasi OAuth2 (Google) 7](#_Toc173488852)

[Gambar 3. 3 Autentikasi STP 9](#_Toc173488853)

[Gambar 3. 4 Diagram Alur Penelitian 10](#_Toc173488854)

[Gambar 4. 1 Logika Penetration Test 14](#_Toc173488871)

[Gambar 4. 2 Hasil Autentikasi QR 16](#_Toc173488872)

[Gambar 4. 3 Hasil Autentikasi Google 17](#_Toc173488873)

[Gambar 4. 4 Skenario Autentikasi STP: Berhasil 18](#_Toc173488874)

[Gambar 4. 5 Skenario Autentikasi STP: Password kadaluarsa 19](#_Toc173488875)

[Gambar 4. 6 Skenario Autentikasi STP: Kirim ulang password 20](#_Toc173488876)

[Gambar 4. 7 Hasil Pengujian Brute Force 21](#_Toc173488877)

[Gambar 4. 8 Hasil Pengujian Serangan Dictionary 22](#_Toc173488878)

[Gambar 4. 9 Pseudocode GeneratePasswordHandler 24](#_Toc173488879)

[Gambar 4. 10 Pseudocode VerifyPasswordHandler 25](#_Toc173488880)

[Gambar 4. 11 Pseudocode ResendPasswordHandler 26](#_Toc173488881)

[Gambar 4. 12 Ilustrasi program lemah dalam menangani password 27](#_Toc173488882)

[Gambar 4. 13 Program yang digunakan dalam penelitian untuk menangani password 28](#_Toc173488883)

[Gambar 4. 14 Ilustrasi program yang tidak memiliki validasi CAPTCHA 29](#_Toc173488884)

[Gambar 4. 15 Program yang digunakan penelitian untuk validasi CAPTCHA 29](#_Toc173488885)

[Gambar 4. 16 Ilustrasi program yang tidak memiliki rate limiter 30](#_Toc173488886)

[Gambar 4. 17 Program yang menerapkan mekanisme rate limiter 30](#_Toc173488887)

[Gambar 4. 18 Konfigurasi indeks MongoDb 31](#_Toc173488888)

[Gambar 4. 19 Database ketika ada proses autentikasi 31](#_Toc173488889)

[Gambar 4. 20 Database setelah 5 menit dari proses autentikasi 32](#_Toc173488890)

# DAFTAR LAMPIRAN

# BAB 1

# PENDAHULUAN

## Latar Belakang

Keamanan merupakan aspek fundamental dalam pengembangan sistem *web*, khususnya pada era digital saat ini di mana ancaman terhadap data dan informasi semakin meningkat[1]. Menurut laporan dari *Cybersecurity* *Ventures*, kerugian global akibat kejahatan siber diperkirakan akan mencapai $10,5 triliun per tahun pada 2025[2]. Peningkatan signifikan ini menekankan urgensi untuk mengembangkan sistem keamanan yang lebih efektif dan efisien dalam melindungi data pengguna.

Autentikasi pengguna adalah salah satu elemen kunci dalam menjaga keamanan sistem, dengan tujuan untuk memastikan bahwa hanya pengguna yang sah yang dapat mengakses sumber daya yang dilindungi[3]. *Web* statis, yang sering kali dianggap lebih sederhana dan kurang dinamis dibandingkan dengan web dinamis, tetap menghadapi tantangan keamanan yang signifikan[4]. Oleh karena itu, pengembangan metode autentikasi yang efektif dan aman menjadi sangat penting.

Berbagai metode autentikasi telah dikembangkan dan diterapkan dalam upaya meningkatkan keamanan sistem *web*. Salah satu metode yang populer adalah penggunaan QR *code*, yang menawarkan kemudahan dan kenyamanan bagi pengguna dengan memanfaatkan perangkat mobile untuk proses autentikasi. Penelitian[5] menunjukkan bahwa penggunaan QR *code* dapat menggantikan kebutuhan akan tabel verifikasi kata sandi dan memanfaatkan ponsel sebagai token yang praktis dan ekonomis. Selain itu, metode *Time*-*Based* *One*-*Time* *Password* (TOTP) yang dikombinasikan dengan QR *code* telah terbukti meningkatkan keamanan sistem gateway[6].

Metode autentikasi lain yang semakin populer adalah OAuth2, yang menyediakan mekanisme autentikasi berbasis token yang aman tanpa perlu menyimpan kata sandi. OAuth2 telah diadopsi secara luas oleh berbagai layanan *web* untuk memastikan keamanan akses data dan layanan. Penelitian[7], [8] membahas mekanisme dan keamanan OAuth2 dalam konteks layanan web dan aplikasi *mobile*.

Dalam konteks *short* *temporary* *password* (STP), metode ini menawarkan keamanan tambahan dengan memberikan kata sandi sementara yang hanya berlaku untuk satu sesi autentikasi. Penelitian[9] mengusulkan sistem autentikasi yang menggunakan QR *code* untuk lingkungan komputasi awan mobile dengan pendekatan STP, sementara[10] membahas sistem autentikasi berbasis QR *code* kontekstual yang menyimpan informasi berubah-ubah untuk setiap sesi autentikasi. Dalam penelitian ini, STP dikirimkan kepada pengguna melalui WhatsApp agar mudah diadopsi, sehingga dapat meningkatkan pengalaman pengguna karena WhatsApp sangat populer dan digandrungi belakangan ini[11].

Meskipun berbagai metode autentikasi telah dikembangkan, penelitian yang menggabungkan ketiga metode ini QR *code*, OAuth2, dan STP untuk *web* statis masih belum banyak ditemukan. Oleh karena itu, penelitian ini bertujuan untuk mengisi kekosongan tersebut dengan mengembangkan sistem autentikasi yang menggabungkan ketiga metode tersebut, memberikan solusi autentikasi yang lebih kuat dan berlapis untuk *web* statis. Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi signifikan dalam meningkatkan keamanan *web* statis dan memberikan pendekatan autentikasi yang inovatif dan efektif bagi pengembang *web*.

## Identifikasi Masalah

Penelitian ini didasarkan pada beberapa permasalahan utama yang sering dihadapi dalam sistem autentikasi konvensional. Masalah-masalah ini perlu diidentifikasi dan dianalisis untuk menemukan solusi yang lebih aman dan efisien.

1. Bagaimana mengamankan autentikasi dan otorisasi *user* pada *website* statis?
2. Bagaimana menghitung resiko keamanan yang terjadi dari setiap skema autentikasi dan otorisasi?

## Tujuan dan Manfaat

Tujuan dari penelitian ini adalah untuk mengembangkan sistem autentikasi yang lebih aman dan efisien menggunakan STP yang dikirim melalui *WhatsApp*. Manfaat yang diharapkan dari penelitian ini adalah peningkatan keamanan dan kemudahan penggunaan sistem autentikasi.

1. Membuat skema otorisasi dan authentikasi pada *website* statis
2. Menghitung resiko pembobolan akun

## Ruang Lingkup

Untuk mencapai tujuan penelitian, ruang lingkup penelitian ini mencakup beberapa aspek teknis dan fungsional yang harus dibatasi, guna menjaga fokus dan esensi dari penelitian.

1. Pengembangan sistem otorisasi dan autentikasi untuk *website* statis.
2. Implementasi protokol OAuth2 yang terintegrasi dengan *Google*.
3. Pemanfaatan teknologi *bcrypt* untuk *hashing* *password* dan MongoDB untuk manajemen data.
4. Implementasi *client* menggunakan *React* untuk membangun aplikasi *web* statis dan SPA (*Single* *Page* *Application*).
5. Penggunaan *WhatsApp* sebagai media penyampaian *password* kepada pengguna.
6. Pengujian akan menggunakan metode *penetrasion* *test*, dengan Teknik *brute* *force* dan *dictionary* *attack*. Pengujian ini berfokus pada satu metode autentikasi yaitu pada metode *short temporary password* (STP).

# BAB II

# LANDASAN TEORI

## Deskripsi Topik Yang Sama

Bagian ini akan membahas tentang berbagai penelitian dan implementasi terkait yang memiliki tujuan dan pendekatan serupa dengan penelitian ini, untuk memberikan konteks dan mendemonstrasikan relevansi penelitian yang sedang dilakukan.

Penelitian tentang sistem autentikasi dan otorisasi telah banyak dilakukan mengingat pentingnya keamanan data di era digital. Beberapa topik yang relevan dengan penelitian ini mencakup:

1. Keamanan *Web* dan Autentikasi

Penelitian[12] menekankan pentingnya protokol keamanan pada aplikasi web, namun belum fokus pada web statis dan modifikasi OAuth2. Penelitian ini mengkaji berbagai protokol keamanan dan menyoroti kebutuhan akan metode autentikasi yang lebih aman dan efisien.

1. Implementasi OAuth2 dalam Autentikasi

Penelitian[13] menjelaskan implementasi OAuth2 dalam autentikasi. Penelitian ini menunjukkan bagaimana OAuth2 dapat digunakan untuk mengamankan akses ke aplikasi web dengan memberikan hak akses terbatas tanpa mengungkapkan kredensial pengguna.

## Deskripsi Metode Yang Sama

Bagian ini akan menjelaskan berbagai metode yang digunakan dalam penelitian sebelumnya yang serupa dengan metode yang digunakan dalam penelitian ini, untuk menunjukkan bagaimana metode tersebut telah berhasil diterapkan dan mendukung relevansi serta validitas metode yang dipilih.

1. *QR Code-Based Authentication*

Penelitian"*A Novel User Authentication Scheme Based on QR-Code*"[5]. Penelitian ini mengusulkan skema autentikasi berbasis QR *code* untuk menggantikan tabel verifikasi kata sandi, dengan memanfaatkan ponsel sebagai token praktis dan ekonomis.

Penelitian"TOTP *Based* *Authentication* *Using* QR *Code* *for* *Gateway* *Entry* *System*"[6]. Penelitian ini mengkombinasikan QR *code* dan *Time*-*Based* *One*-*Time* *Password* (TOTP) untuk meningkatkan keamanan sistem masuk *gateway*.

Penelitian "*A One-Time Password Scheme with* QR-*Code* *Based* *on* *Mobile* *Phone*"[14]. Skema OTP berbasis QR *code* yang menghilangkan penggunaan tabel verifikasi kata sandi, menggunakan ponsel untuk autentikasi.

1. OAuth2 *Authentication*

Penelitian "*Authentication for Web Services via OAuth*"[8]. Penelitian ini mendeskripsikan mekanisme OAuth dan cara penggunaannya untuk autentikasi layanan *web*.

Penelitian"*Breaking and Fixing Mobile App Authentication with* OAuth2.0-*based* *Protocols*"[7]. Analisis terhadap keamanan autentikasi aplikasi mobile menggunakan protokol OAuth2.0 dan usulan solusi untuk memperbaiki kerentanannya.

1. *Short* *Temporary* *Password* (STP)

Penelitian "*A Study on Authentication System Using QR Code for Mobile Cloud Computing Environment*"[9]. Penelitian ini mengusulkan sistem autentikasi yang menggunakan QR code untuk lingkungan komputasi awan mobile dengan pendekatan STP.

Penelitian "*Secure Authentication Using One Time Contextual QR Code*"[10]. Sistem autentikasi berbasis QR *code* kontekstual yang menyimpan informasi berubah-ubah untuk setiap sesi autentikasi.

# BAB III

# METODOLOGI PENELITIAN

1. Metodologi Penelitian

Penelitian ini menggunakan metode eksperimental dengan tahapan yang sistematis untuk mengembangkan dan menguji autentikasi. Metodologi ini dirancang untuk mengoptimalkan keamanan dan kenyamanan pengguna dalam proses autentikasi di aplikasi web statis dan Single Page Application (SPA). Bagian ini menjelaskan secara rinci tahapan-tahapan yang dilakukan dalam pengembangan dan implementasi sistem tersebut.

1. Desain Sistem

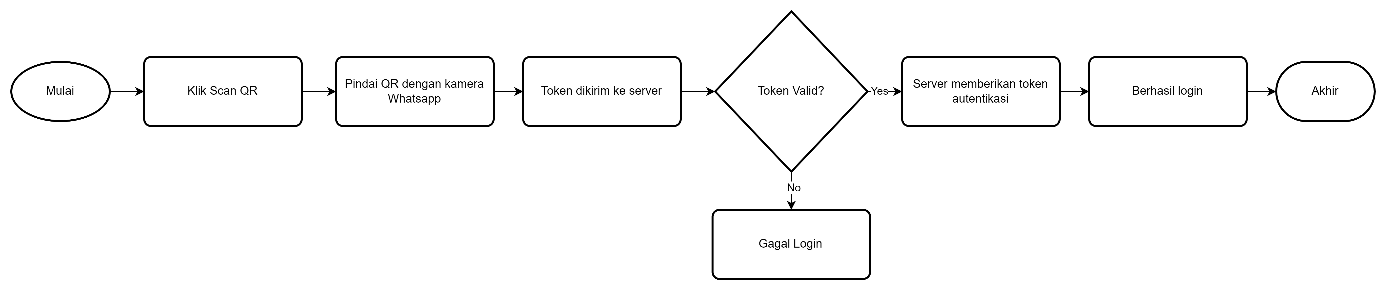
Dengan mengembangkan sistem autentikasi untuk web statis yang menggabungkan tiga metode autentikasi yang berbeda, yaitu QR code, OAuth2 (Google), dan short temporary password (STP). Sistem autentikasi yang diusulkan terdiri dari tiga komponen utama:

* Autentikasi menggunakan kode QR;
* Autentikasi menggunakan google;
* Autentikasi menggunakan STP.

1. Implementasi Sistem
2. Autentikasi QR Code

Pengguna akan memindai QR code yang ditampilkan pada halaman login web menggunakan perangkat mobile mereka. Untuk merinci proses autentikasi ini, Berikut adalah alur dari proses autentikasi menggunakan QR code.

1. Pengguna memilih untuk autentikasi menggunakan kode QR.
2. Aplikasi menampilkan kode QR untuk dipindai oleh pengguna.
3. Pengguna memindai kode QR menggunakan kamera dari aplikasi Whatsapp.
4. Whatsapp akan mengirimkan token ke server.
5. Server memeriksa validitas token.
6. Jika token valid, maka server mengirim token autentikasi kepada pengguna. Dan pengguna berhasil login.

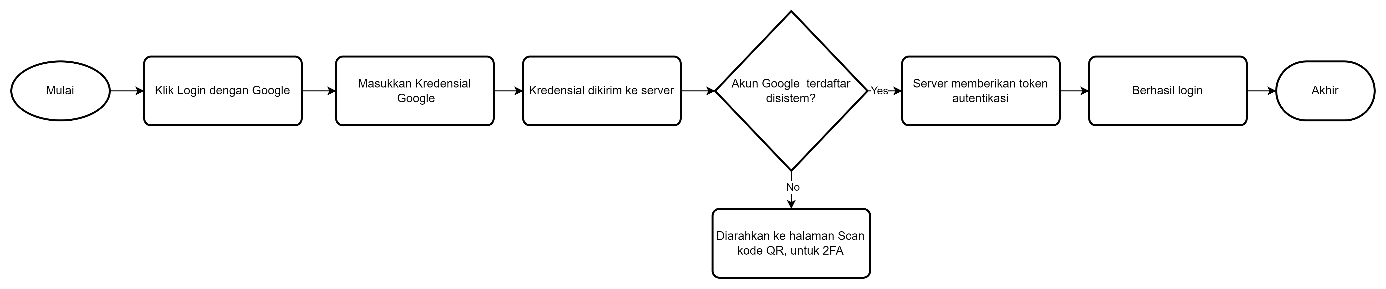


Gambar 3. 1 Autentikasi QR Code

1. OAuth2 (Google)

Pengguna akan dapat login menggunakan akun Google mereka, yang menggunakan protokol OAuth2 untuk memberikan token autentikasi yang aman. Untuk merinci proses autentikasi ini, Berikut adalah alur dari proses autentikasi menggunakan google.

1. Pengguna memilih untuk autentikasi menggunakan Google.
2. Pengguna mengklik tombol "Login with Google".
3. Pengguna memasukkan kredensial Google mereka.
4. Google mengirimkan kredensial ke server.
5. Server memverifikasi kredensial Google yang diterima.
6. Server memeriksa validitas kredensial dari akun Google.
7. Jika kredensial valid, server mengirim token autentikasi kepada pengguna dan pengguna berhasil login.
8. Jika kredensial tidak valid, login gagal.

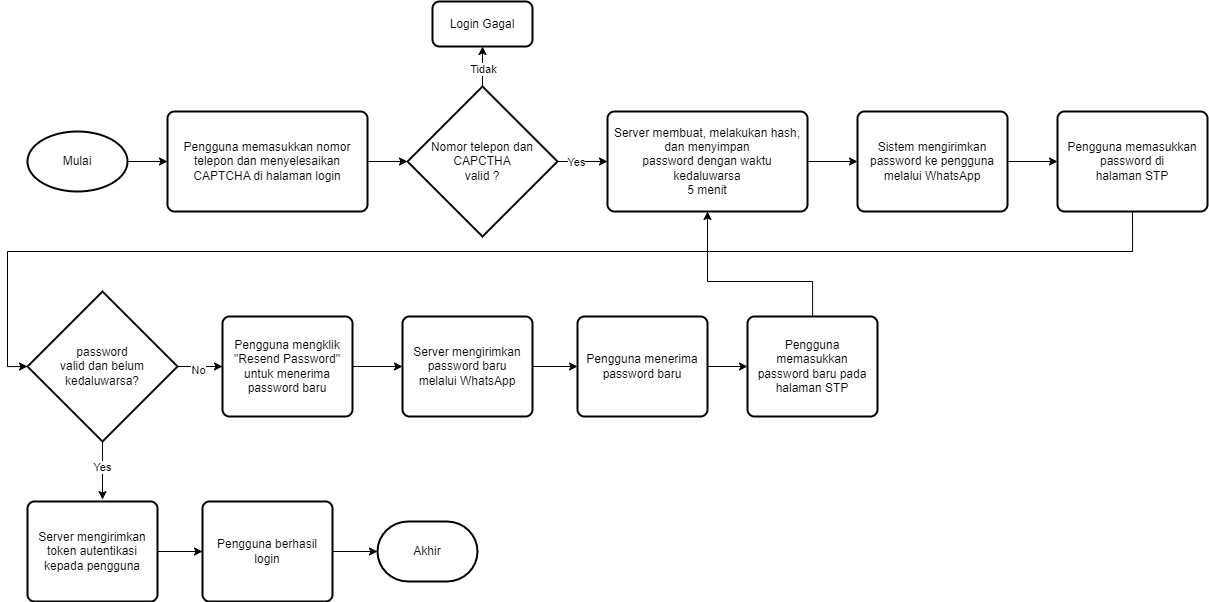


Gambar 3. 2 Autentikasi OAuth2 (Google)

1. Secure Temporary Password (STP)

Sistem akan mengirimkan STP ke pengguna melalui WhatsApp untuk memastikan keamanan tambahan. STP ini hanya berlaku untuk satu sesi dan memiliki masa berlaku yang singkat. Untuk merinci proses autentikasi ini, Berikut adalah alur dari proses autentikasi menggunakan STP.

1. Pengguna memasukkan nomor telepon dan menyelesaikan CAPTCHA di halaman login.
2. Sistem memeriksa validitas nomor telepon dan CAPTCHA.
3. Jika nomor telepon dan CAPTCHA valid, server membuat, melakukan hash, dan menyimpan password dengan waktu kedaluwarsa 5 menit.
4. Sistem mengirimkan password ke pengguna melalui WhatsApp.
5. Pengguna menerima password melalui WhatsApp.
6. Pengguna memasukkan password di halaman STP.
7. Sistem memverifikasi password yang dimasukkan dan memeriksa apakah sudah kedaluwarsa.
8. Jika password valid dan belum kedaluwarsa, server mengirimkan token autentikasi kepada pengguna dan pengguna berhasil login.
9. Jika password tidak valid atau sudah kedaluwarsa, pengguna mengklik "Resend Password" untuk menerima password baru.
10. Server mengirimkan password baru melalui WhatsApp.
11. Pengguna menerima password baru.
12. Pengguna memasukkan password baru pada halaman STP.
13. Sistem memverifikasi password baru yang dimasukkan dan memeriksa apakah sudah kedaluwarsa.
14. Jika password valid dan belum kedaluwarsa, server mengirimkan token autentikasi kepada pengguna dan pengguna berhasil login.
15. Jika password tidak valid, login gagal.



Gambar 3. 3 Autentikasi STP

1. Pengujian dan Evaluasi

Melakukan pengujian fungsionalitas dan keamanan sistem autentikasi yang telah diimplementasikan. Pengujian meliputi:

* Pengujian autentikasi menggunakan kode QR;
* Pengujian autentifikasi menggunakan google;
* Pengujian autentifikasi menggunakan STP;
* Evaluasi kinerja dan keamanan sistem.

1. Dokumentasi dan Pelaporan

Menyusun laporan penelitian yang mencakup latar belakang, metodologi, hasil, dan kesimpulan dari penelitian ini. Laporan ini juga mencakup rekomendasi untuk pengembangan lebih lanjut dan aplikasi praktis dari hasil penelitian.

## Tahapan Tahapan Diagram Metodologi Penelitian

Bagian ini akan memvisualisasikan tahapan-tahapan metodologi penelitian dalam bentuk diagram alur untuk memberikan gambaran yang jelas tentang proses penelitian. Berikut adalah diagram alur metodologi penelitian:

1. Desain Sistem

* Merancang skema autentikasi;
* Menentukan teknologi yang digunakan (Golang, MongoDB, React, WhatsApp).

1. Implementasi Sistem

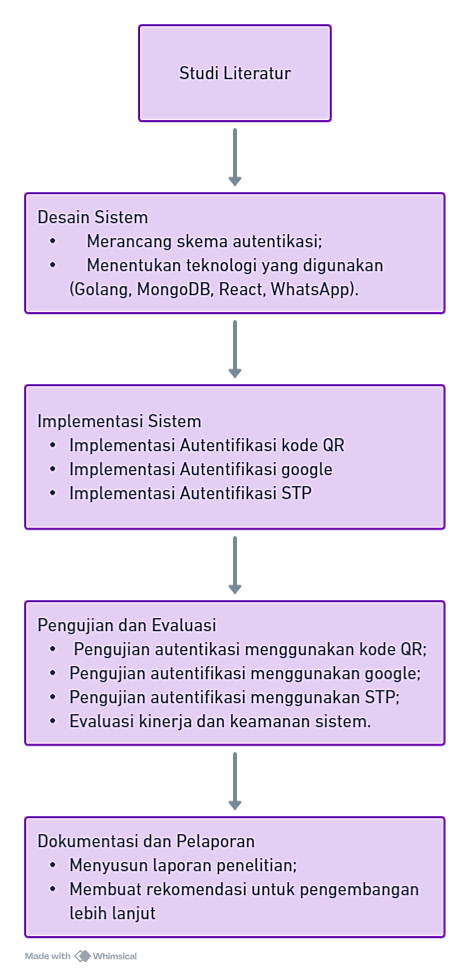
* Implementasi Autentifikasi kode QR
* Implementasi Autentifikasi google
* Implementasi Autentifikasi STP.

1. Pengujian dan Evaluasi

* Pengujian autentikasi menggunakan kode QR;
* Pengujian autentifikasi menggunakan google;
* Pengujian autentifikasi menggunakan STP;
* Evaluasi kinerja dan keamanan sistem.

1. Dokumentasi dan Pelaporan

* Menyusun laporan penelitian;
* Membuat rekomendasi untuk pengembangan lebih lanjut.



Gambar 3. 4 Diagram Alur Penelitian

# BAB IV

# EKSPERIMEN DAN HASIL

## Eksperimen

Bagian ini menjelaskan langkah-langkah dan prosedur yang diambil untuk menguji sistem autentikasi dan otorisasi yang diimplementasikan pada website statis. Pengujian dilakukan dalam tiga jenis autentikasi utama, yaitu autentikasi QR, autentikasi Google, dan autentikasi STP (*Secure Temporary Password*). Setiap jenis autentikasi diuji berdasarkan beberapa skenario untuk memastikan keamanan dan keefektifan sistem.

1. Autentikasi QR

Autentikasi QR adalah metode yang memungkinkan pengguna untuk login dengan memindai kode QR menggunakan perangkat mobile mereka. Metode ini bertujuan untuk mengurangi risiko pencurian kredensial karena tidak melibatkan input manual dari pengguna. Langkah-langkah Pengujian:

* Pengguna mengakses halaman login dan memilih opsi autentikasi QR.
* Sistem menampilkan kode QR yang unik untuk setiap sesi.
* Pengguna memindai kode QR menggunakan aplikasi autentikasi di ponsel mereka.
* Sistem memverifikasi hasil pemindaian dan mengizinkan akses jika berhasil.

1. Autentikasi Google

Autentikasi Google memanfaatkan OAuth2 untuk mengizinkan pengguna login menggunakan akun Google mereka. Metode ini memanfaatkan infrastruktur keamanan Google, sehingga meningkatkan kepercayaan dan keamanan. Langkah-langkah Pengujian:

* Pengguna mengakses halaman login dan memilih opsi autentikasi Google.
* Sistem mengarahkan pengguna ke halaman login Google.
* Pengguna memasukkan kredensial Google mereka dan memberikan izin akses.
* Sistem menerima token dari Google dan memverifikasi identitas pengguna.
* Jika verifikasi berhasil, sistem mengizinkan akses.

1. Autentikasi STP (*Secure* *Temporary* *Password*)

Autentikasi STP menggunakan *password* sementara yang dikirimkan melalui WhatsApp. Metode ini dirancang untuk mengurangi risiko pencurian *password* dengan menggunakan *password* yang berumur pendek dan dikirimkan melalui saluran yang aman. Skenario Pengujian:

1. Autentikasi Berhasil

* Pengguna mengakses halaman login dan memasukkan nomor telepon.
* Sistem mengirimkan *password* sementara ke nomor WhatsApp pengguna.
* Pengguna memasukkan *password* sementara pada halaman STP.
* Sistem memverifikasi *password* dan mengizinkan akses jika berhasil.

1. *Password* Expired

* Pengguna mengakses halaman login dan memasukkan nomor telepon.
* Sistem mengirimkan *password* sementara ke nomor WhatsApp pengguna.
* Pengguna mencoba login setelah 5 menit (masa berlaku *password* berakhir).
* Sistem menolak akses dan menampilkan pesan bahwa *password* telah kedaluwarsa.

1. Resend Password

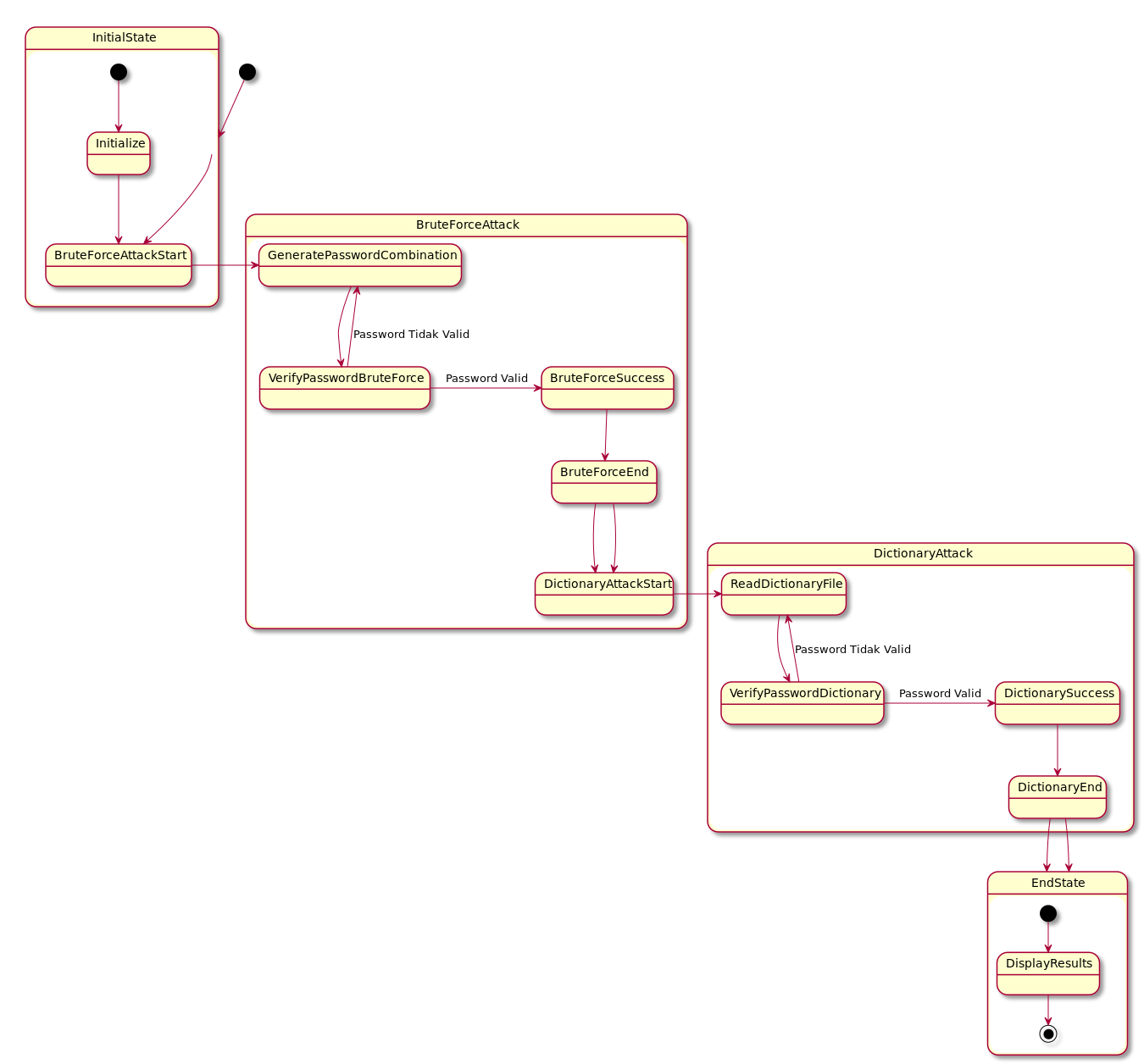
* Pengguna mengakses halaman login dan memasukkan nomor telepon.
* Sistem mengirimkan *password* sementara ke nomor WhatsApp pengguna.
* Pengguna memilih opsi untuk mengirim ulang *password* setelah waktu tunggu habis.
* Sistem mengirimkan *password* baru dan pengguna dapat login menggunakan *password* baru tersebut.

1. Pengujian Penetrasi

Pada bagian ini, keamanan sistem akan diuji dengan menggunakan metode penetration test. Teknik brute force akan digunakan sebagai salah satu alat utama untuk menguji ketahanan sistem terhadap upaya pembobolan yang agresif. Penetration test ini dirancang untuk mensimulasikan serangan yang mungkin dilakukan oleh pihak tidak bertanggung jawab, dengan tujuan untuk menemukan dan memperbaiki kerentanan yang ada.

Teknik *brute* *force* adalah metode serangan di mana penyerang mencoba semua kemungkinan kombinasi untuk menebak kata sandi pengguna[15]. Selain itu, teknik *dictionary* *attack* juga digunakan, di mana penyerang menggunakan daftar kata sandi umum untuk menebak kata sandi pengguna[16].

Dalam penelitian ini, kedua teknik ini digunakan untuk menguji keamanan sistem STP (*Secure* *Temporary* *Password*) yang telah dibuat. Logika dari program pengujian penetrasi yang digunakan dalam penelitian ini dapat dijelaskan sebagai berikut:



Gambar 4. 1 Logika Penetration Test

Diagram *State* (Logika Penetration Test) diatas menggambarkan alur proses dari program pengujian *brute force* dan *dictionary attack* yang dirancang untuk menguji keamanan sistem autentikasi STP. Berikut adalah rincian dari setiap *state* nya.

* 1. InitialState

Proses dimulai dari InitialState, di mana program kemudian bergerak menuju BruteForceAttackStart. Pada tahap ini, serangan brute force dimulai dengan mencoba setiap kombinasi *password* yang mungkin hingga panjang maksimum yang telah ditentukan.

* 1. BruteForceAttack

Dalam *state* BruteForceAttack, proses dimulai dengan BruteForceGeneratePassword, di mana program menghasilkan kombinasi *password*. *Password* yang dihasilkan kemudian diverifikasi pada *state* BruteForceVerifyPassword. Jika *password* tidak valid, proses kembali ke BruteForceGeneratePassword untuk mencoba kombinasi lainnya. Jika *password* valid, proses bergerak ke state BruteForceSuccess, menandakan bahwa *password* telah ditemukan. Proses *brute* *force* berakhir pada state BruteForceEnd, yang kemudian berlanjut ke state DictionaryAttackStart untuk memulai serangan *dictionary*.

* 1. DictionaryAttack

Pada state DictionaryAttack, proses dimulai dengan DictionaryAttackStart. Program kemudian membaca file dictionary pada state DictionaryReadFile dan memverifikasi *password* pada *state* DictionaryVerifyPassword. Jika *password* yang diverifikasi tidak valid, proses kembali ke DictionaryReadFile untuk mencoba password berikutnya dalam daftar. Jika password valid, proses bergerak ke *state* DictionarySuccess, menandakan bahwa *password* telah ditemukan. Proses serangan dictionary berakhir pada state DictionaryEnd, yang kemudian berlanjut ke EndState.

* 1. EndState

Dalam EndState, hasil dari serangan *brute* *force* dan *dictionary* *attack* ditampilkan pada *state* DisplayResults, dan proses berakhir.

Diagram ini mengilustrasikan bagaimana program pengujian *brute* *force* dan *dictionary* *attack* bekerja secara sistematis untuk menguji keamanan sistem autentikasi. Melalui serangkaian langkah yang melibatkan penghasilan dan verifikasi kombinasi *password*, program ini bertujuan untuk menemukan *password* yang valid, baik melalui *brute* *force* maupun *dictionary* *attack*. Implementasi ini membantu mengidentifikasi potensi kelemahan dalam sistem autentikasi dan meningkatkan keamanannya.

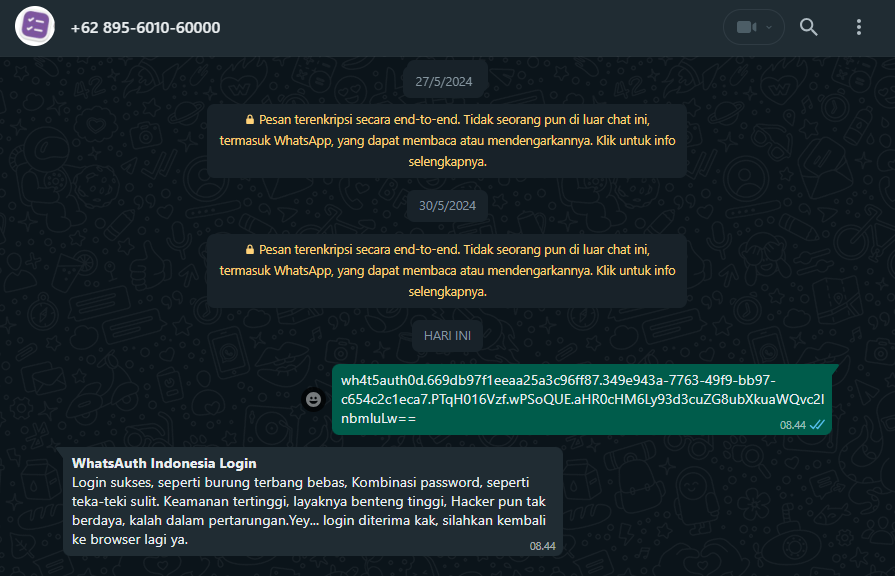
## Hasil

Bagian ini menyajikan hasil dari setiap skenario pengujian yang telah dilakukan. Hasil tersebut dianalisis untuk mengevaluasi keamanan dari setiap metode autentikasi.

1. Hasil Autentikasi QR

Pengujian autentikasi QR menunjukkan bahwa metode ini dapat mengurangi risiko pencurian kredensial dengan menghindari input manual dari pengguna.

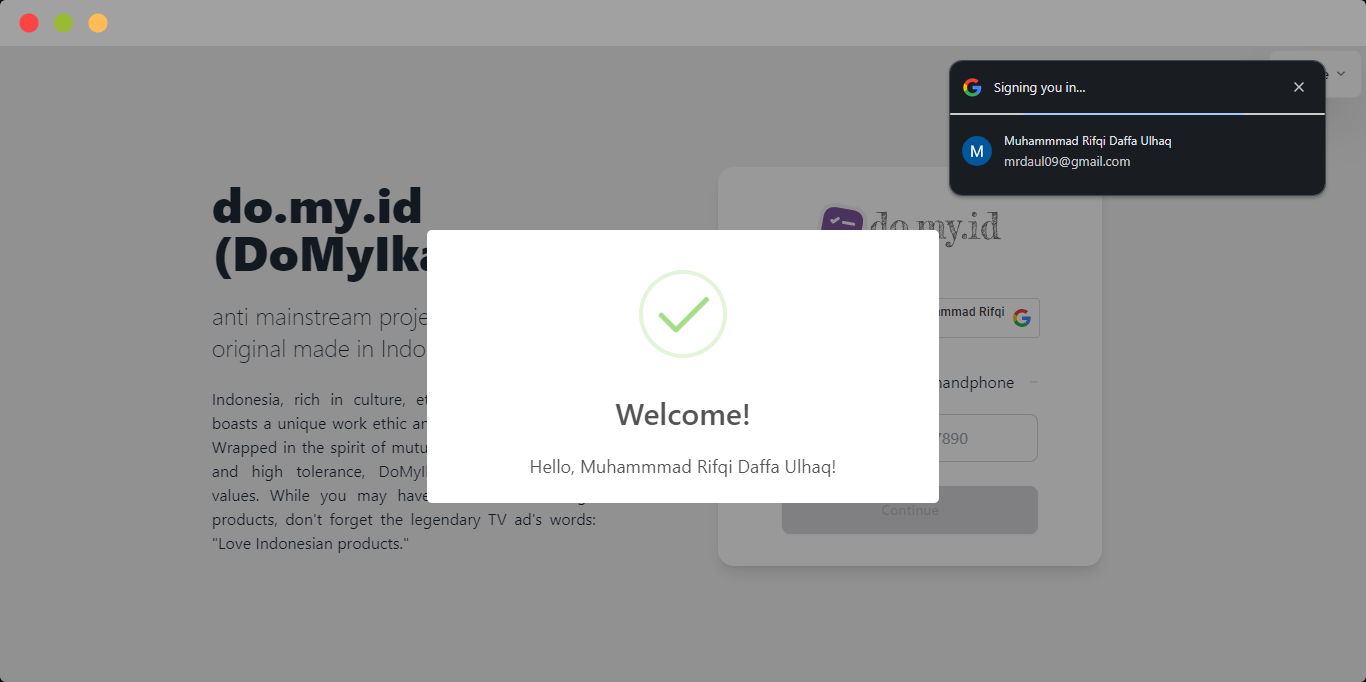




Gambar 4. 2 Hasil Autentikasi QR

1. Hasil Autentikasi Google

Autentikasi Google berhasil memanfaatkan infrastruktur keamanan Google untuk meningkatkan kepercayaan pengguna. Namun, risiko tetap ada jika token akses berhasil dicuri oleh pihak yang tidak berwenang.

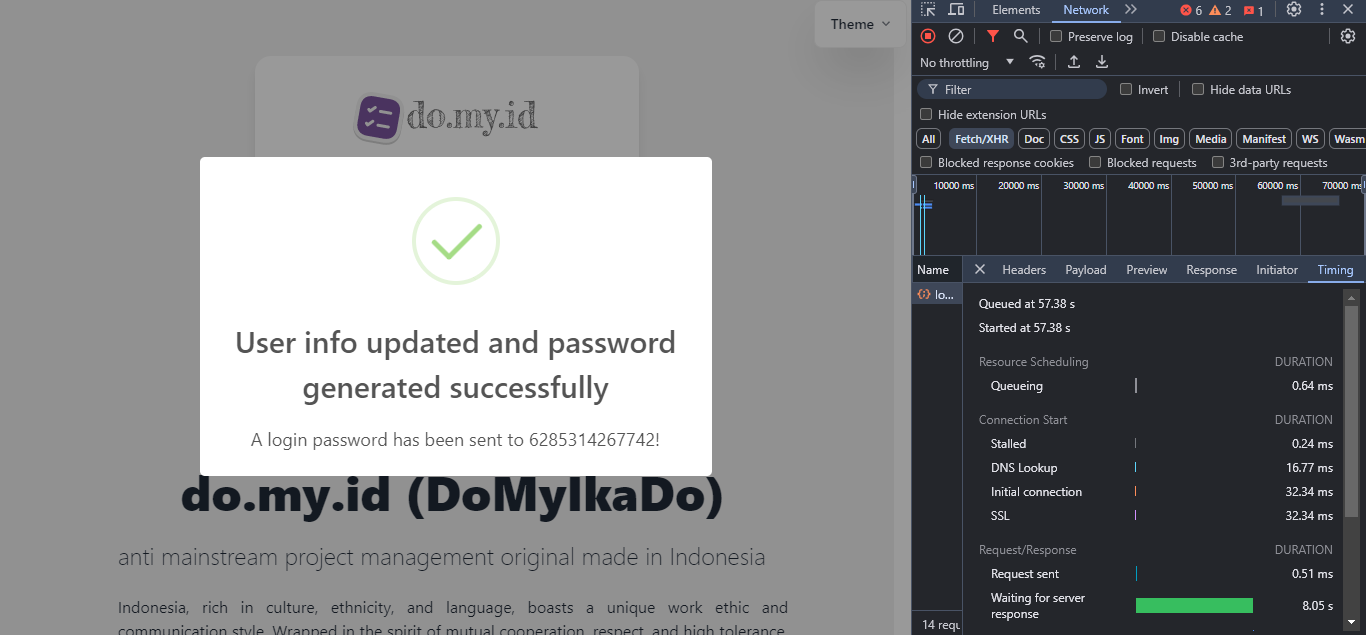


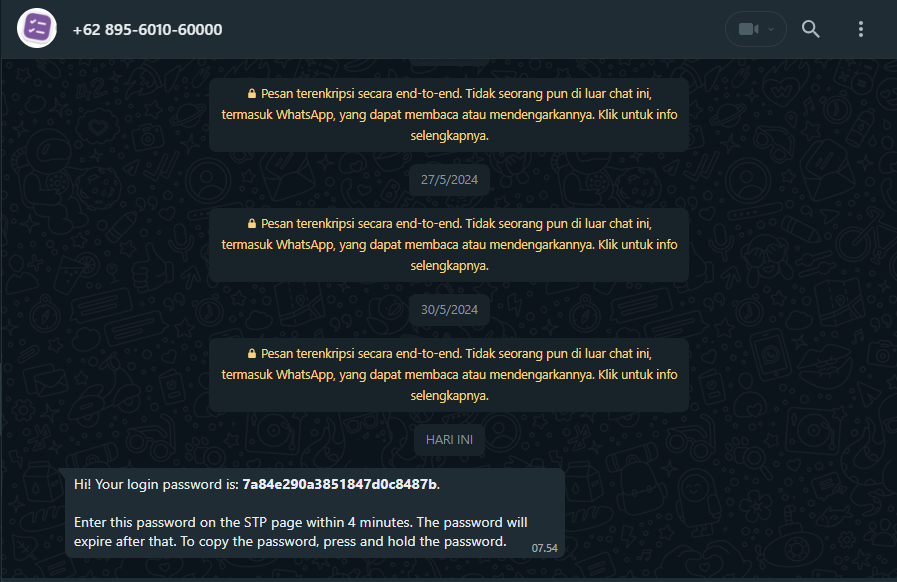
Gambar 4. 3 Hasil Autentikasi Google

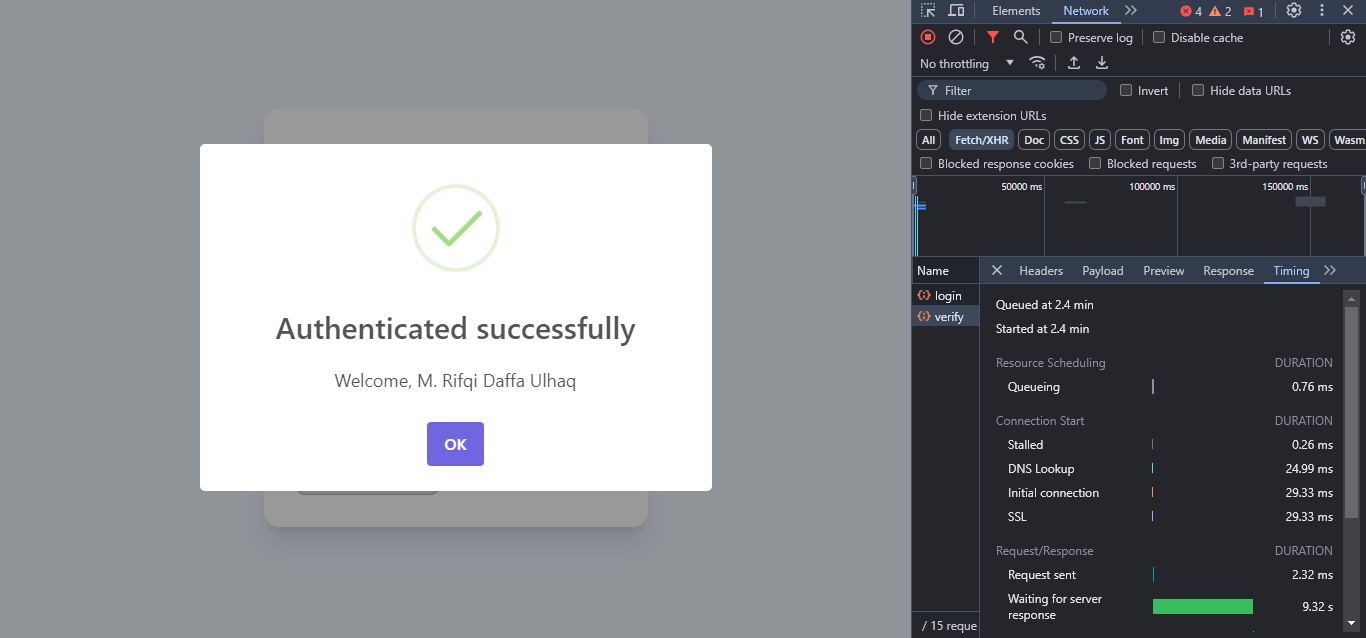
1. Hasil Autentikasi STP

* Autentikasi Berhasil

Pengujian menunjukkan bahwa pengguna dapat login dengan sukses menggunakan password sementara yang dikirimkan melalui WhatsApp. Metode ini meningkatkan keamanan karena menggunakan durasi password valid yang singkat.



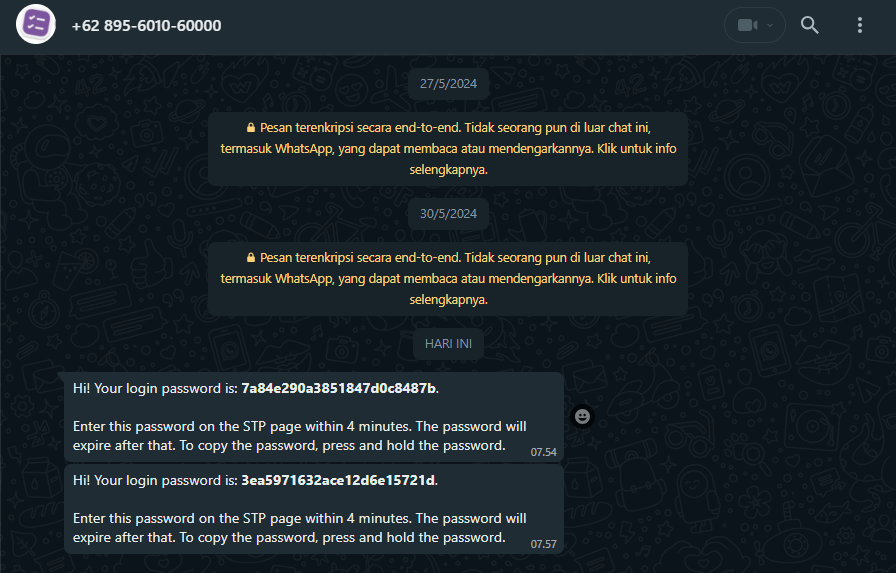


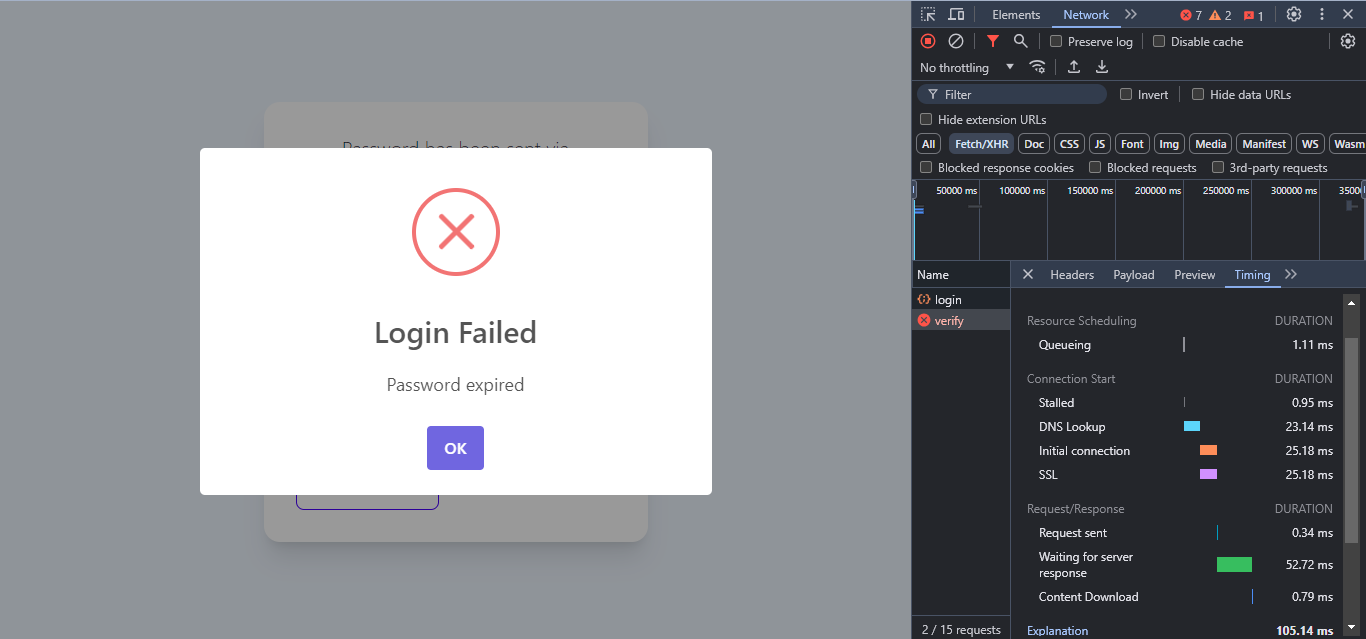


Gambar 4. 4 Skenario Autentikasi STP: Berhasil

* *Password* *Expired*

Pengujian menunjukkan bahwa sistem berhasil menolak akses jika pengguna mencoba login setelah masa berlaku *password* berakhir. Hal ini menambah lapisan keamanan tambahan.

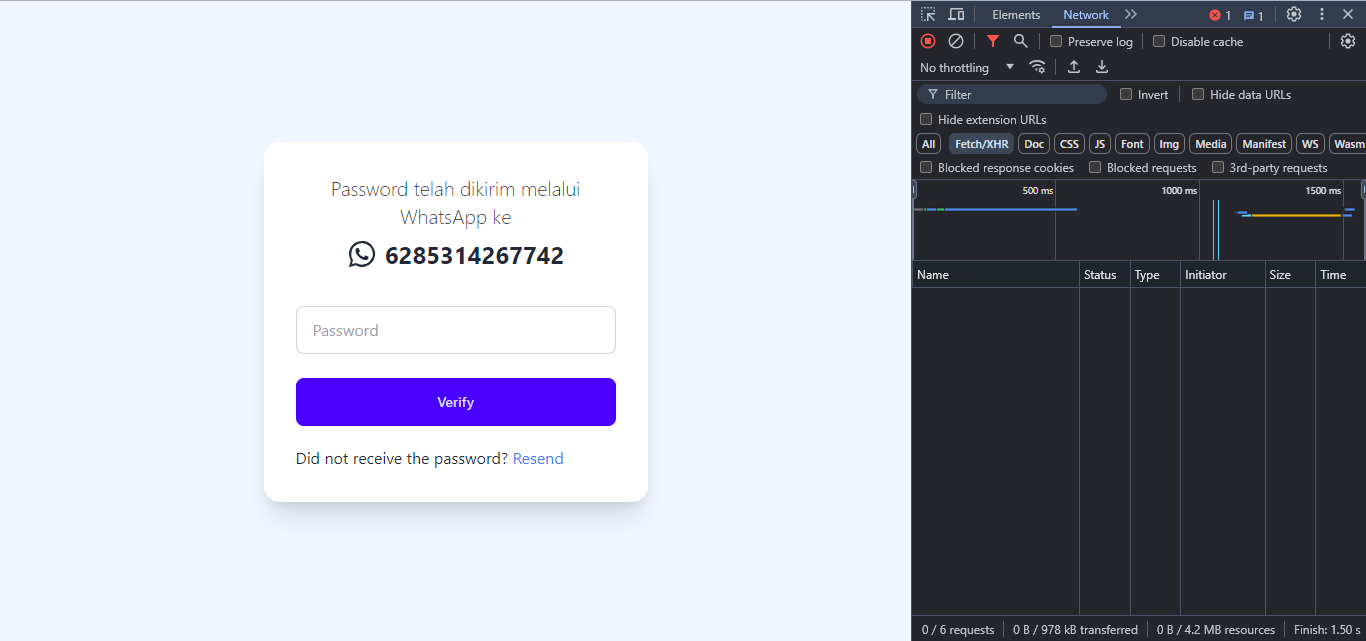


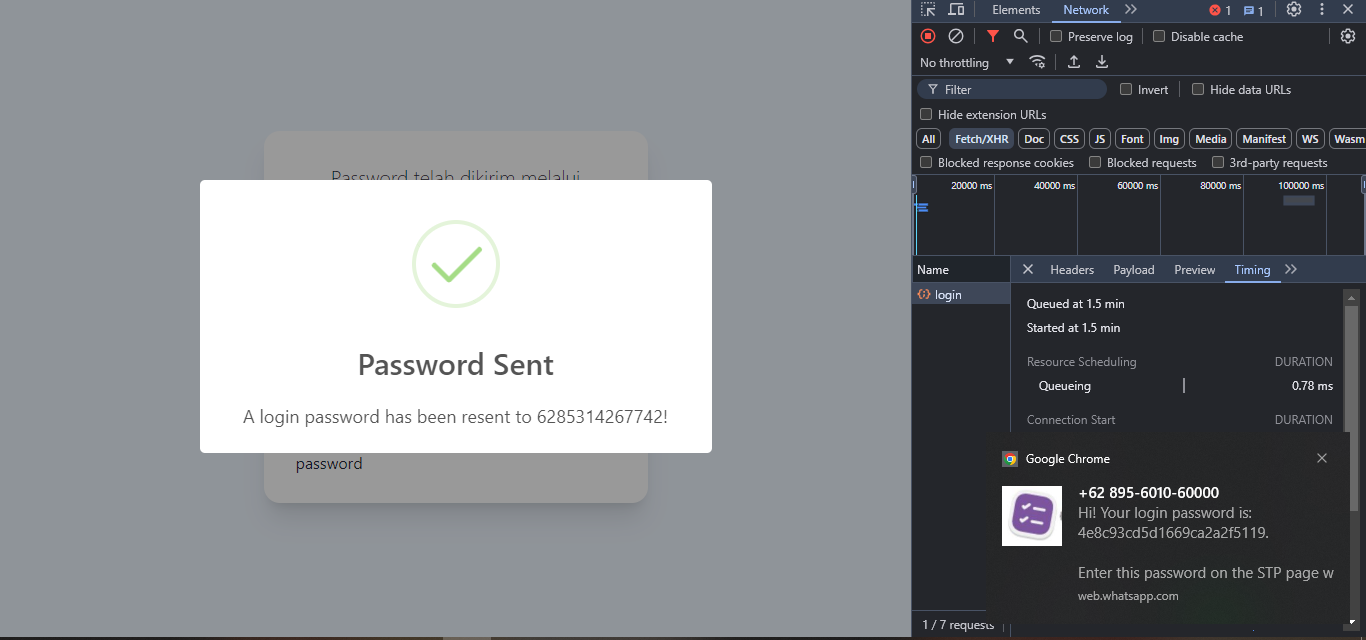


Gambar 4. 5 Skenario Autentikasi STP: Password kadaluarsa

* Resend *Password*

Pengujian menunjukkan bahwa pengguna dapat meminta pengiriman ulang *password* dan berhasil login menggunakan *password* baru. Ini menunjukkan fleksibilitas sistem dalam menangani skenario dimana pengguna tidak menerima atau lupa *password* sementara.





Gambar 4. 6 Skenario Autentikasi STP: Kirim ulang password

1. Evaluasi Keamanan

Evaluasi keamanan merupakan langkah krusial dalam memastikan bahwa sistem autentikasi yang dikembangkan mampu melindungi pengguna dari berbagai ancaman. Pada bagian ini, akan berfokus pada evaluasi keamanan sistem dengan menggunakan metode *penetration* *test*. Hasil dari pengujian ini akan memberikan gambaran yang jelas mengenai efektivitas dan kekuatan sistem autentikasi dalam menghadapi ancaman keamanan yang nyata.

1. Hasil Pengujian *Brute* *Force*

Pengujian penetrasi menggunakan Teknik *brute* *force* dilakukan dalam kurun waktu 5 menit. Karena pada tenggat waktu itulah satu *password* akan berlaku untuk sebuah akun yang tengah melakukan autentikasi. Dalam kurun waktu tersebut akan ditebak berbagai kemungkinan *password* yang digunakan pengguna untuk melakukan autentikasi. Berikut adalah hasil pengujian menunjukkan serangan *brute* *force* yang dilakukan selama 5 menit.



Gambar 4. 7 Hasil Pengujian Brute Force

Hasil dari pengujian *brute* *force* menunjukkan bahwa metode ini tidak berhasil menemukan *password* yang valid dalam waktu 5 menit. Selama pengujian, program *brute* *force* mencoba berbagai kombinasi *password*, dan respon dari *server* menunjukkan beberapa pesan kesalahan seperti "crypto/bcrypt: hashedPassword is not the hash of the given password" dan "Password Expired". Selain itu, mekanisme *rate* *limiting* yang diterapkan pada *server* mengakibatkan beberapa permintaan mendapatkan respon "Too Many Requests", yang berarti server berhasil membatasi jumlah permintaan yang bisa diterima dalam jangka waktu tertentu.

Penting untuk dicatat bahwa tidak semua baris respon dari server dimasukkan dalam hasil ini, namun dapat disimpulkan bahwa selama periode pengujian, *brute* *force* terus mencoba berbagai kemungkinan *password* untuk menebak kata sandi pengguna saat autentikasi, namun tidak ada yang berhasil.

1. Hasil Pengujian Serangan *Dictionary*

Pengujian dengan teknik *dictionary* *attack* dilakukan untuk mengevaluasi keamanan sistem STP (*Secure* *Temporary* *Password*) yang telah dibuat. Teknik *dictionary* *attack* adalah metode serangan di mana penyerang menggunakan daftar kata-kata atau kombinasi yang umum digunakan sebagai kata sandi untuk menebak kata sandi pengguna. Dalam penelitian ini, pengujian *dictionary* *attack* akan dijalankan selama 5 menit, di mana program akan mencoba mencocokkan setiap kata dalam daftar (<https://github.com/danielmiessler/SecLists/blob/master/Passwords/2023-200_most_used_passwords.txt>) dengan kata sandi pengguna yang sebenarnya. Tujuan dari pengujian ini adalah untuk menilai seberapa baik sistem dapat bertahan terhadap serangan yang menggunakan kata-kata umum atau yang sering digunakan sebagai kata sandi.



Gambar 4. 8 Hasil Pengujian Serangan Dictionary

Hasil dari pengujian *dictionary* *attack* menunjukkan bahwa teknik ini tidak berhasil menebak kata sandi pengguna dalam waktu 5 menit. Pengujian ini dimulai dengan mencoba berbagai kata sandi yang umum digunakan, yang diambil dari daftar kata yang disimpan dalam file dictionary.txt. Selama pengujian, program menerima beberapa respon dari server, termasuk pesan kesalahan seperti "crypto/bcrypt: hashedPassword is not the hash of the given password," yang menunjukkan bahwa kata sandi yang dicoba tidak cocok dengan hash kata sandi yang disimpan. Selain itu, server juga mengirimkan pesan "Please try again later." yang menunjukkan bahwa batas permintaan telah tercapai, serta pesan "Password Expired" yang menunjukkan bahwa kata sandi sementara yang digunakan telah kedaluwarsa.

Secara keseluruhan, meskipun program mencoba berbagai kombinasi kata sandi dari daftar kata selama hampir 5 menit, teknik *dictionary* *attack* ini tidak berhasil menebak kata sandi pengguna yang sebenarnya.

1. Program yang Berperan sebagai Keamanan

Sistem *backend* yang telah dikembangkan memiliki beberapa bagian yang berperan penting dalam keamanan. Bagian-bagian tersebut meliputi:

* GeneratePasswordHandler

Fungsi ini bertanggung jawab untuk menghasilkan password baru dan menyimpannya di database. CAPTCHA juga digunakan di sini untuk memverifikasi bahwa request berasal dari manusia. Berikut adalah *Pseudocode* dari program.



Gambar 4. 9 Pseudocode GeneratePasswordHandler

* VerifyPasswordHandler

Fungsi ini digunakan untuk memverifikasi *password* yang dimasukkan oleh pengguna. Password yang sudah kadaluarsa tidak akan diterima. Berikut adalah *Pseudocode* dari program.



Gambar 4. 10 Pseudocode VerifyPasswordHandler

* ResendPasswordHandler

Fungsi ini memungkinkan pengguna untuk meminta pengiriman ulang *password* jika belum diterima atau kadaluarsa. Namun, fungsi ini hanya dapat digunakan setelah waktu tertentu untuk mencegah spam. Berikut adalah *Pseudocode* dari program.



Gambar 4. 11 Pseudocode ResendPasswordHandler

1. Perbandingan Keamanan

Dalam subjudul ini, kita akan membahas perbandingan antara program yang memiliki keamanan lemah dengan program yang memiliki keamanan kuat berdasarkan beberapa aspek utama dari sistem autentikasi. Aspek-aspek tersebut mencakup pengelolaan kata sandi, penggunaan CAPTCHA, perlindungan terhadap serangan *brute* *force*, dan penanganan permintaan dari pengguna.

* Pengelolaan Kata Sandi

Pengelolaan kata sandi merupakan aspek penting dalam keamanan aplikasi. Kata sandi yang lemah atau tidak dikelola dengan baik dapat membuka celah bagi penyerang untuk mengakses data sensitif. Oleh karena itu, penting untuk memahami bagaimana program yang lemah dan kuat dalam menangani kata sandi guna memastikan keamanan yang optimal.

Program Lemah:

* + 1. Menggunakan kata sandi dalam bentuk teks biasa tanpa enkripsi.
    2. Menyimpan kata sandi dalam database tanpa *hashing*.
    3. Menggunakan kata sandi yang sederhana dan mudah ditebak karena berasal dari input pengguna.



Gambar 4. 12 Ilustrasi program lemah dalam menangani password

Program Yang Seharusnya:

1. Menggunakan algoritma hashing yang aman, seperti bcrypt, untuk menyimpan kata sandi.
2. Menghasilkan kata sandi yang kompleks dan panjang secara acak.
3. Memanfaatkan salt dalam proses hashing untuk meningkatkan keamanan.



Gambar 4. 13 Program yang digunakan dalam penelitian untuk menangani password

* 1. Validasi CAPTCHA

Validasi CAPTCHA adalah langkah penting untuk mencegah serangan otomatis dari bot. CAPTCHA yang efektif dapat membedakan antara pengguna manusia dan bot, sehingga melindungi aplikasi dari serangan brute force dan spam. Bagian ini membahas perbedaan antara program yang lemah dan kuat dalam implementasi CAPTCHA.

Program Lemah:

1. Tidak menggunakan CAPTCHA untuk validasi pengguna.
2. Mengabaikan validasi CAPTCHA pada langkah-langkah kritis, seperti login atau pengiriman kata sandi.



Gambar 4. 14 Ilustrasi program yang tidak memiliki validasi CAPTCHA

Program yang seharusnya:

1. Mengintegrasikan CAPTCHA yang kuat, yang dalam hal ini, penelitian ini menggunakan Cloudflare Turnstile, sehingga mengenali pola serangan otomatis dan bot.
2. Memastikan CAPTCHA selalu aktif dan diperiksa pada setiap permintaan autentikasi.



Gambar 4. 15 Program yang digunakan penelitian untuk validasi CAPTCHA

1. Penanganan Permintaan (*Request*)

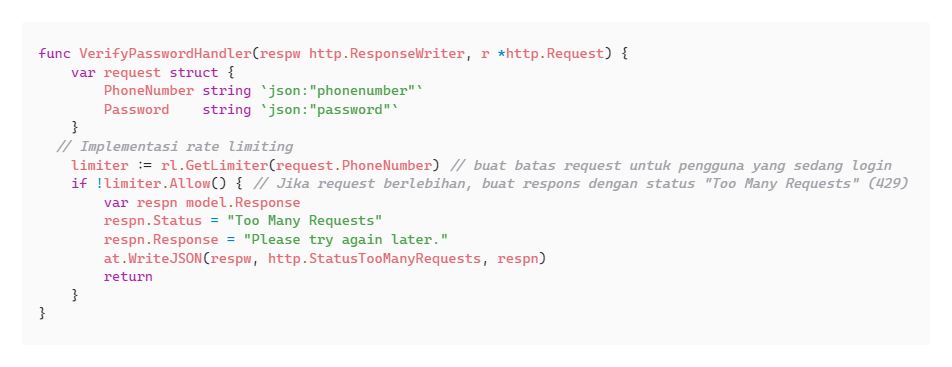
Penanganan permintaan autentikasi dengan baik adalah kunci untuk menjaga keamanan aplikasi. Mekanisme rate limiting, seperti penguncian akun dan pelambatan login, merupakan metode umum untuk mencegah serangan cracking password[17]. Di bagian ini, kita akan melihat bagaimana program yang lemah dan kuat menangani permintaan autentikasi.

Program yang lemah terhadap aspek ini adalah program yang tidak memiliki mekanisme untuk membatasi jumlah permintaan yang dapat dilakukan dalam jangka waktu tertentu, membuatnya rentan terhadap serangan *brute* *force*.



Gambar 4. 16 Ilustrasi program yang tidak memiliki rate limiter

Sementara itu, program yang digunakan oleh penelitian ini, sudah mengimplementasikan mekanisme *rate* *limiting* untuk membatasi jumlah permintaan autentikasi dalam jangka waktu tertentu.



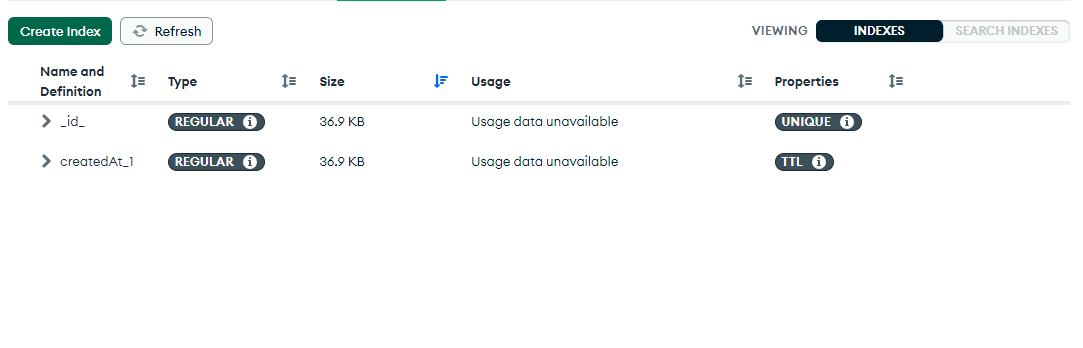
Gambar 4. 17 Program yang menerapkan mekanisme rate limiter

1. Kesimpulan

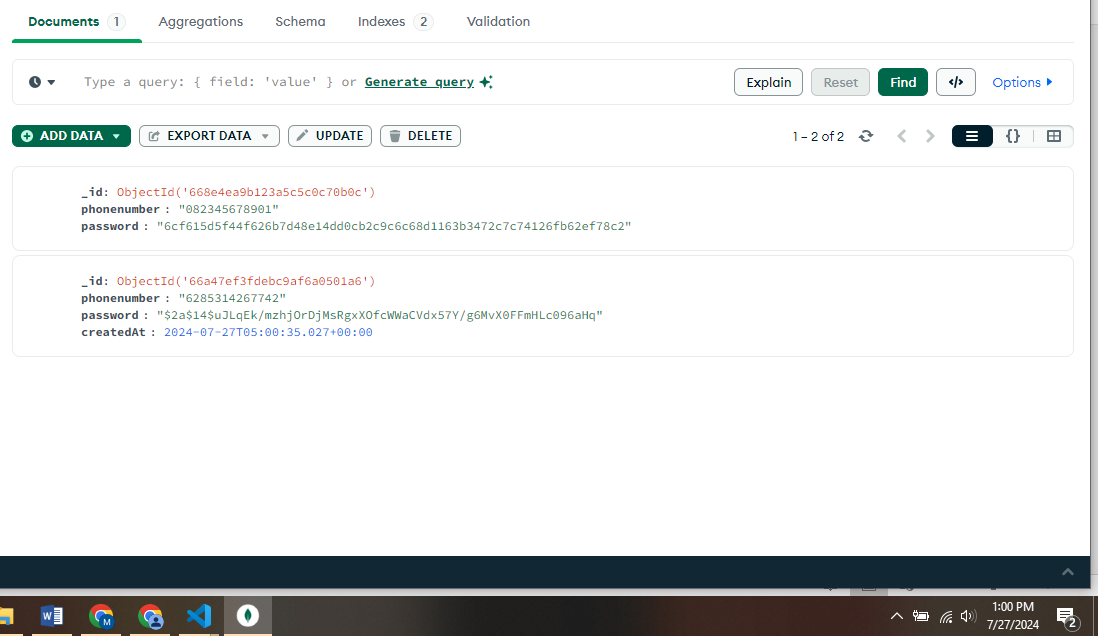
Untuk memastikan keamanan yang optimal dalam sistem autentikasi, sangat penting untuk:

1. Menggunakan hashing untuk menyimpan kata sandi.
2. Melakukan pengindeksan pada setiap data, yang merupakan salah satu fitur MongoDB, agar data autentikasi yang tersimpan dapat dihapus secara berkala. Hal ini juga memastikan keamanan dari upaya pembobolan akun, meskipun akun dicoba ditebak password-nya berkali-kali menggunakan Teknik *brute force*.

Berikut gambar yang menunjukkan konfigurasi dari *collection* yang digunakan autentikasi, dimana *field* createdAt yang diberikan indeks selama 5 menit.

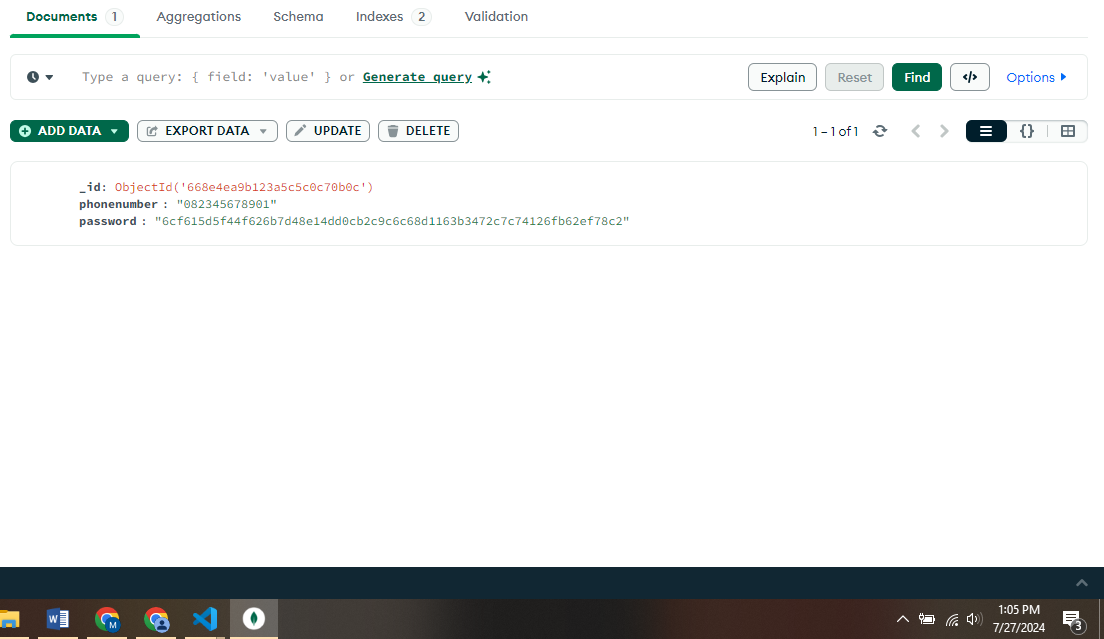


Gambar 4. 18 Konfigurasi indeks MongoDb



Gambar 4. 19 Database ketika ada proses autentikasi

Ini memungkinkan, setiap adanya proses autentikasi pengguna, data hanya akan disimpan dalam waktu 5 menit, setelah tenggang waktu tersebut, maka data akan otomatis hilang, sehingga password tidak akan lagi valid, dan mencegah adanya upaya pembobolan akun. Hal ini juga untuk memastikan data tidak kedaluwarsa terlalu lama dan tetap relevan untuk tujuan autentikasi sementara.



Gambar 4. 20 Database setelah 5 menit dari proses autentikasi

1. Mengintegrasikan CAPTCHA untuk memverifikasi bahwa permintaan berasal dari manusia.
2. Menerapkan batasan pada jumlah percobaan login untuk mencegah serangan brute force.
3. Melakukan validasi dan sanitasi input dengan benar.

Dengan mengikuti praktik terbaik ini, aplikasi akan lebih terlindungi dari berbagai ancaman keamanan, termasuk serangan brute force dan serangan berbasis web lainnya.

# BAB V

# KESIMPULAN

## Kesimpulan Masalah

Bagian ini merangkum kesimpulan dari permasalahan yang diidentifikasi selama penelitian. Kesimpulan ini akan menjawab permasalahan yang telah diuraikan di bagian sebelumnya.

1. Bagaimana mengamankan autentikasi dan otorisasi *user* pada *website* statis?

Penelitian ini berhasil mengimplementasikan skema autentikasi dan otorisasi yang aman pada *website* statis menggunakan berbagai metode autentikasi, seperti autentikasi QR, autentikasi Google, dan autentikasi STP. Setiap metode telah diuji dan menunjukkan efektivitas dalam mengamankan akses pengguna.

1. Bagaimana menghitung resiko keamanan yang terjadi dari setiap skema autentikasi dan otorisasi?

Penelitian ini juga menganalisis risiko keamanan dari setiap skema autentikasi yang digunakan. Risiko keamanan dihitung dengan menguji setiap metode terhadap serangan *brute* *force* dan *dictionary* *attack*. Hasil pengujian menunjukkan bahwa metode QR code, OAuth2, dan STP dapat mengurangi risiko pembobolan akun secara signifikan. Metode STP, yang mengirimkan *password* sementara melalui WhatsApp, terbukti paling efektif dalam menambah lapisan keamanan karena password hanya berlaku untuk satu sesi dan memiliki masa berlaku yang singkat. Pengujian *brute* *force* dan *dictionary* *attack* menunjukkan bahwa *password* yang dihasilkan oleh sistem STP cukup kuat dan tidak mudah ditebak dalam waktu yang wajar. Berikut adalah hasil pengujian:

1. *Brute Force Attack*

Mencoba semua kemungkinan kombinasi karakter hingga panjang tertentu. Pengujian menunjukkan bahwa *password* tidak ditemukan dalam waktu 5 menit.

1. *Dictionary Attack*

Mencoba menebak *password* menggunakan daftar kata-kata yang sering digunakan sebagai *password*. Pengujian menunjukkan bahwa *password* tidak ditemukan dalam waktu 5 menit.

## Kesimpulan Metode

Bagian ini menyimpulkan metode yang digunakan dalam penelitian ini. Kesimpulan ini akan menilai efektivitas metode yang diterapkan dalam mencapai tujuan penelitian.

1. Metode Autentikasi dan Otorisasi pada *Website* Statis

Skema autentikasi dan otorisasi yang dikembangkan mencakup berbagai metode yang sesuai untuk *website* statis, seperti autentikasi QR, autentikasi Google, dan autentikasi STP. Implementasi metode ini terbukti meningkatkan keamanan tanpa mengurangi kenyamanan pengguna.

1. Menghitung Resiko Pembobolan Akun

Dengan menghitung resiko pembobolan akun dari setiap metode autentikasi, penelitian ini berhasil mengidentifikasi metode yang aman untuk diterapkan pada *website* statis. Hasilnya menunjukkan bahwa kombinasi berbagai metode autentikasi dapat mengurangi resiko pembobolan akun secara signifikan.

## Kesimpulan Eksperimen

Bagian ini menyimpulkan hasil eksperimen yang telah dilakukan selama penelitian. Kesimpulan ini akan menilai keberhasilan sistem berdasarkan hasil pengujian yang telah dilakukan.

1. Autentikasi QR

Eksperimen autentikasi QR menunjukkan bahwa metode ini sangat efektif dalam mengamankan akses pengguna. Namun, metode ini memerlukan perangkat tambahan seperti ponsel yang mendukung QR *code* *scanner*.

1. Autentikasi Google

Autentikasi menggunakan Google *Sign*-*In* terbukti mudah digunakan dan aman. Metode ini memanfaatkan infrastruktur keamanan Google, yang sudah terbukti handal dalam menangani autentikasi pengguna.

1. Autentikasi STP
   1. Berhasil

Pengguna yang memasukkan *password* dengan benar dalam batas waktu yang ditentukan dapat berhasil masuk ke sistem.

* 1. *Password* Kedaluwarsa

Jika password telah *kedaluwarsa*, pengguna harus meminta *password* baru.

* 1. Resend *Password*

Pengguna dapat mengirim ulang *password* jika tidak menerima atau jika password telah kedaluwarsa. Metode ini memastikan pengguna tetap dapat mengakses sistem meskipun terjadi kendala pada pengiriman *password* sebelumnya.

1. Pengujian *Brute* *Force* dan *Dictionary* *Attack*

Pengujian tambahan dengan *brute* *force* dan *dictionary* *attack* menunjukkan bahwa *password* yang dihasilkan oleh sistem STP cukup kuat dan tidak mudah ditebak dalam waktu yang wajar. Hal ini menunjukkan bahwa sistem STP yang dikembangkan mampu memberikan keamanan yang tinggi terhadap serangan tersebut.

Kesimpulan dari penelitian ini menunjukkan bahwa metode yang diterapkan efektif dalam mengatasi permasalahan yang diidentifikasi. Sistem autentikasi yang dirancang berhasil meningkatkan keamanan dan efisiensi, serta memberikan pengalaman pengguna yang lebih baik. Implementasi teknologi yang digunakan, seperti *Bcrypt*, MongoDB, *React*, dan WhatsApp, terbukti berhasil dalam mencapai tujuan penelitian.

# BAB VI

# SARAN

Bagian ini memberikan saran untuk pengembangan lebih lanjut berdasarkan temuan penelitian. Saran ini bertujuan untuk meningkatkan sistem yang telah dikembangkan serta mengatasi kendala yang mungkin dihadapi di masa mendatang.

* Penggunaan *Embedded* *Message* di WhatsApp

Untuk meningkatkan kenyamanan pengguna, disarankan untuk mengirimkan pesan yang di-embed dengan tombol salin *password* di WhatsApp. Dengan adanya tombol salin, pengguna dapat menyalin password dengan sekali klik tanpa harus menyeleksi teks secara manual. Hal ini dapat mengurangi kesalahan pengguna dalam menyalin password dan mempercepat proses login.

* Pengembangan Fitur *Multi*-*Factor* *Authentication* (MFA)

Meskipun sistem autentikasi saat ini telah cukup aman, penambahan fitur MFA dapat meningkatkan keamanan lebih lanjut. Pengguna dapat memverifikasi identitas mereka melalui metode tambahan seperti kode OTP yang dikirim melalui SMS atau *email*.

* Optimisasi Performa Sistem

Untuk menghadapi peningkatan jumlah pengguna di masa mendatang, disarankan untuk terus mengoptimalkan performa sistem. Penggunaan teknologi *caching* dan *load* *balancing* dapat membantu meningkatkan kecepatan respon server dan mengurangi beban pada server utama.

* Integrasi dengan Layanan Pihak Ketiga

Menambahkan integrasi dengan layanan pihak ketiga seperti Github untuk opsi login dapat meningkatkan fleksibilitas dan kenyamanan pengguna. Pengguna dapat memilih metode login yang paling sesuai dengan preferensi mereka.

* Peningkatan UX/UI

Meningkatkan desain antarmuka pengguna (UI) dan pengalaman pengguna (UX) agar lebih intuitif dan mudah digunakan. Melakukan pengujian dengan pengguna (*user* *testing*) untuk mendapatkan *feedback* langsung dapat membantu dalam perbaikan desain.

* Penerapan Teknik *Machine* *Learning*

Menggunakan teknik *machine* *learning* untuk mendeteksi pola penggunaan yang mencurigakan dan mencegah akses yang tidak sah. Algoritma *machine* *learning* dapat membantu dalam meningkatkan keamanan sistem secara otomatis.

* Penyempurnaan Dokumentasi

Menyediakan dokumentasi yang lebih lengkap dan mudah dipahami bagi pengembang lain yang mungkin akan mengembangkan sistem ini lebih lanjut. Dokumentasi yang baik akan mempermudah proses pengembangan dan pemeliharaan di masa mendatang.

* Penelitian Lanjutan

Melakukan penelitian lanjutan untuk mengeksplorasi metode autentikasi lain yang lebih aman dan efisien. Penggunaan teknologi terbaru dan pendekatan inovatif dapat membantu dalam terus meningkatkan keamanan dan kenyamanan pengguna.

Dengan implementasi saran-saran di atas, diharapkan sistem autentikasi ini dapat terus berkembang dan memberikan keamanan serta kenyamanan yang lebih baik bagi pengguna. Penelitian ini telah menunjukkan potensi besar dalam penggunaan teknologi untuk meningkatkan keamanan dan efisiensi dalam autentikasi pengguna, dan masih banyak ruang untuk pengembangan lebih lanjut.

# DAFTAR PUSTAKA

[1] J. Shahid, M. K. Hameed, I. T. Javed, K. N. Qureshi, M. Ali, and N. Crespi, “A Comparative Study of Web Application Security Parameters: Current Trends and Future Directions,” *Applied Sciences*, vol. 12, no. 8, 2022, doi: 10.3390/app12084077.

[2] “Cybercrime To Cost The World $10.5 Trillion Annually By 2025.” Accessed: Jul. 22, 2024. [Online]. Available: https://cybersecurityventures.com/cybercrime-damage-costs-10-trillion-by-2025/

[3] B. Narwal and A. K. Mohapatra, “A survey on security and authentication in wireless body area networks,” *Journal of Systems Architecture*, vol. 113, p. 101883, 2021, doi: https://doi.org/10.1016/j.sysarc.2020.101883.

[4] D. V Kornienko, S. V Mishina, and M. O. Melnikov, “The Single Page Application architecture when developing secure Web services,” in *Journal of Physics: Conference Series*, 2021, p. 12065.

[5] K.-C. Liao and W.-H. Lee, “A Novel User Authentication Scheme Based on QR-Code,” *JNW*, vol. 5, pp. 937–941, Jul. 2010, doi: 10.4304/jnw.5.8.937-941.

[6] A. Arvind, P. Mahajan, and R. Chalke, “TOTP Based Authentication Using QR Code For Gateway Entry System,” *International Journal of Engineering and Computer Science*, vol. 9, pp. 25023–25028, Jul. 2020, doi: 10.18535/ijecs/v9i05.4481.

[7] R. Yang, W. Lau, and S. Shi, “Breaking and Fixing Mobile App Authentication with OAuth2.0-based Protocols,” Jul. 2017, pp. 313–335. doi: 10.1007/978-3-319-61204-1\_16.

[8] D. Nolan and D. Temple Lang, “Authentication for Web Services via OAuth,” 2014, pp. 441–461. doi: 10.1007/978-1-4614-7900-0\_13.

[9] D.-S. Oh, B.-H. Kim, and J.-K. Lee, “A Study on Authentication System Using QR Code for Mobile Cloud Computing Environment,” vol. 184, Jul. 2011, doi: 10.1007/978-3-642-22333-4\_65.

[10] D. Mahansaria and U. K. Roy, “Secure Authentication Using One Time Contextual QR Code,” in *International Symposium on Security in Computing and Communications*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:219008525

[11] C. E. (Catherine) Lee, H. H. Chern, and D. A. Azmir, “WhatsApp Use in a Higher Education Learning Environment: Perspective of Students of a Malaysian Private University on Academic Performance and Team Effectiveness,” *Educ Sci (Basel)*, vol. 13, no. 3, 2023, doi: 10.3390/educsci13030244.

[12] J. Sen, *Theory and Practice of Cryptography and Network Security Protocols and Technologies*. Rijeka: IntechOpen, 2013. doi: 10.5772/56823.

[13] D. Fett, R. Küsters, and G. Schmitz, “A Comprehensive Formal Security Analysis of OAuth 2.0,” *CoRR*, vol. abs/1601.01229, 2016, [Online]. Available: http://arxiv.org/abs/1601.01229

[14] K.-C. Liao, W.-H. Lee, M.-H. Sung, and T.-C. Lin, “A One-Time Password Scheme with QR-Code Based on Mobile Phone,” in *2009 Fifth International Joint Conference on INC, IMS and IDC*, 2009, pp. 2069–2071. doi: 10.1109/NCM.2009.324.

[15] V. Grover and others, “An Efficient Brute Force Attack Handling Techniques for Server Virtualization,” in *Proceedings of the International Conference on Innovative Computing & Communications (ICICC)*, 2020.

[16] T. Gautam and U. Singh, “AN APPROACH FOR DETECTING PASSWORD PATTERN IN DICTIONARY ATTACK,” 2022.

[17] B. Lu, X. Zhang, Z. Ling, Y. Zhang, and Z. Lin, “A Measurement Study of Authentication Rate-Limiting Mechanisms of Modern Websites,” in *Proceedings of the 34th Annual Computer Security Applications Conference*, in ACSAC ’18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 89–100. doi: 10.1145/3274694.3274714.