OPEN ACCESS

Volume 10 (2), April 2026, 475-488

E-ISSN:2580-1643

Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)

DOI: https://doi.org/10.35870/jtik.v10i2.5273

Perancangan Sistem *Single Sign-On (SSO)* Berbasis QR Code pada PT.XYZ dengan Pendekatan Model *Waterfall* untuk Lingkungan Korporasi

Agung Prasetyo Nugroho [1*], Felix David [2]

[1,2] Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana, Kota Salatiga, Provinsi Jawa Tengah, Indonesia.*

a r t i c l e i n f o

Article history:
Received 7 August 2025
Received in revised form
20 August 2025
Accepted 20 September 2025
Available online April 2026.

Keywords:
Single Sign-On;
Authentication; QR Code;
WebSocket; Passwordless;
Token.

Kata Kunci:
Single Sign-On; Autentikasi;
QR Code; WebSocket; Tanpa
Kata Sandi; Token.

a b s t r a c t

Hundreds of systems at PT. XYZ cause problems in managing credentials, as each user is required to remember multiple credentials. This study developed a QR Code-based Single Sign-On (SSO) system to simplify the login process using the Waterfall model Software Development Life Cycle (SDLC) approach. The login process is completed by scanning a QR Code that represents the user's authentication token, and the login status is synchronized in real-time using WebSocket. The scope of this study is limited to the development of a login page prototype and testing login time and QR scanning time, compared with other methods. This study does not include penetration testing or cryptography. The results from black-box testing show that all features function properly, and User Acceptance Testing (UAT) achieved an average score of 90.4%. Therefore, this QR Code-based SSO login system has proven to be effective, secure, and suitable for use as a modern authentication solution.

a b s t r a k

Ratusan sistem yang terdapat pada PT. XYZ menyebabkan masalah dalam mengelola kredensial, karena setiap user perlu mengingat banyak kredensial. Penelitian ini mengembangkan sistem Single Sign-On (SSO) berbasis QR Code untuk menyederhanakan proses login menggunakan pendekatan Software Development Life Cycle (SDLC) model Waterfall. Proses login dilakukan dengan memindai QR Code yang merepresentasikan token autentikasi user, kemudian status login disinkronkan secara real-time menggunakan WebSocket. Ruang lingkup penelitian ini terbatas pada pengembangan prototipe halaman login dan pengujian waktu login serta waktu pemindaian QR, yang dibandingkan dengan metode lainnya. Penelitian ini tidak mencakup uji penetrasi dan kriptografi. Hasil pengujian black-box menunjukkan bahwa semua fitur berfungsi dengan baik, dan User Acceptance Testing (UAT) menghasilkan skor rata-rata 90,4%. Dengan demikian, sistem login SSO berbasis QR Code ini terbukti efektif, aman, dan layak untuk digunakan sebagai solusi autentikasi modern.

Corresponding Author. Email: 672021077@student.uksw.edu [1].

1. Pendahuluan

Di era perkembangan teknologi yang pesat, banyak aspek telah mengalami digitalisasi yang membantu menyelesaikan suatu persoalan dengan mudah melalui integrasi teknologi digital (Chacko *et al.*, 2023). PT. XYZ menjadi salah satu perusahaan yang bergerak dalam bidang retail, telah melakukan digitalisasi untuk mendukung operasionalnya. Dengan terdapatnya ratusan sistem yang telah dibuat menyebabkan setiap user memiliki akun yang berbeda di setiap layanannya. Mengelola banyak kredensial menjadi sebuah tantangan tersendiri, selain itu akan menjadi masalah jika *password* yang diberikan terlalu sederhana atau *password* yang sama digunakan kembali untuk berbagai aplikasi (Kumar *et al.*, 2024). Hal ini menyebabkan masalah dalam mengelola kredensial, karena *user* perlu membuat dan mengingat banyak *password*. Salah satu solusi untuk mengatasi masalah ini adalah dengan menggunakan *Single Sign-On* (SSO). SSO memungkinkan *user* untuk mengakses berbagai aplikasi dengan satu kali login (Fauzi *et al.*, 2023). Dalam penelitian ini, kami mengembangkan sistem SSO berbasis *Quick Response Code* (QR Code), yang memungkinkan *user* untuk login tanpa harus memasukkan *username* atau *password* secara berulang.

Quick Response Code (QR Code) merupakan gambar dua dimensi yang merepresentasikan data text *(Nazar et al., 2022)*. Berdasarkan penelitian yang dilakukan oleh Triyanto Anggoro dan Budi Slamet Rianto, *QR Code* terbukti efektif dan efisien pada sistem absensi karena karyawan hanya perlu men scan karyawan ID ke *scanner* (Anggoro & Rianto, 2025). Karena itulah *QR Code* dipilih pada skenario autentikasi ini, *QR Code* bisa digunakan untuk mengenkode *One Time Password* (OTP) menjadi *QR Code*. Dengan begitu *user* hanya perlu memindai *QR Code* untuk bisa masuk ke aplikasi tanpa harus mengisi kredensial, hal ini akan meningkatkan keamanan dan kenyamanan. Penerapan SSO berbasis *QR Code* dalam lingkungan korporasi dengan sinkronisasi *real-time* masih terbatas, terutama yang menggunakan pendekatan *passwordless* yang aman dan efisien. Gap penelitian ini terletak pada belum adanya implementasi SSO QR berbasis sinkronisasi *real-time* di PT. XYZ, yang bertujuan untuk mengurangi jumlah kredensial yang perlu diingat oleh *user* dan meningkatkan kemudahan akses. Penelitian ini bertujuan untuk mengembangkan sistem SSO berbasis *QR Code* yang dapat memenuhi kebutuhan tersebut, serta menguji efektivitasnya melalui *User Acceptance Testing* (UAT) dengan target skor ≥85% dan waktu login rata-rata < 5 detik. Ruang lingkup penelitian ini terbatas pada pengembangan prototipe halaman login untuk aplikasi web dan mobile. Penelitian ini difokuskan pada aspek fungsionalitas dasar sistem login dan efisiensi penggunaannya. Oleh karena itu, penelitian ini tidak mencakup uji penetrasi dan pengujian kriptografi. Batasan ini ditetapkan untuk memfokuskan penelitian pada pengembangan sistem autentikasi yang praktis dan efisien. Berdasarkan latar belakang tersebut, penelitian ini berfokus untuk mengatasi permasalah sistem autentikasi yang berulang pada PT. XYZ. Penelitian ini bertujuan untuk merancang sistem *Single Sign-On (SSO)* berbasis *QR Code* agar user hanya perlu login satu kali untuk bisa mengakses aplikasi-aplikasi kantor, sehingga user tidak perlu memasukkan kredensial secara berulang-ulang. Penelitian ini diharapkan dapat memberikan manfaat baik itu secara teoritis, yang menjadi referensi pengembangan sistem autentikasi modern tanpa kredensial dengan memanfaatkan teknologi *QR Code*. Secara praktis, membantu PT. XYZ menyederhanakan proses login dan meningkatkan kenyamanan pengguna.

Penelitian ini dibatasi pada pengembangan prototipe halaman login dan integrasinya, tanpa membahas secara rinci terkait aspek keamanan aplikasi mobile serta tidak membahas terkait aspek kriptografi data. Autentikasi adalah proses penting dalam sistem untuk memastikan bahwa *user* yang mengakses sistem dikenali dan memiliki akses yang sah (Syarif Aziz *et al.*, 2021). Metode autentikasi umumnya menggunakan *username* dan *password*, namun metode ini memiliki berbagai kelemahan, seperti *user* lupa *password* dan manajemen kredensial yang rumit ketika *user* harus mengakses banyak aplikasi (Fernando *et al.*, 2023). Permasalahan ini mendorong munculnya pendekatan autentikasi terpusat seperti *Single Sign-On* (SSO) yang mampu menyederhanakan proses login lintas aplikasi. Berbagai penelitian terkait implementasi *Single Sign-On (SSO)* telah dilakukan, hal tersebut menghasilkan berbagai cara pengimplementasian. Salah satunya adalah penelitian yang mengembangkan *SSO* dengan kombinasi *OAuth 2.0* dan *One Time Password (OTP)* (Gumeraruloh Arianto *et al.*, 2025).

Pendekatan serupa diterapkan pula dengan penggunaan *magic link* sebagai metode autentikasi (Ruswandi & Alijoyo, 2024). Kedua metode ini terbukti efektif dalam meningkatkan keamanan, namun masih mengharuskan user untuk mengetikkan kredensial awal atau menunggu data validasi yang dikirim melalui jalur komunikasi eksternal. Padahal, prinsip utama dari SSO adalah mengakses beberapa aplikasi dengan sekali login agar user tidak perlu menginput kredensial secara berulang-ulang (Fauzi *et al*., 2023). Penelitian oleh Suwannakom dan Buranasaksee (2023) mengembangkan autentikasi tanpa password (*passwordless*) berbasis protokol *OpenID Connect* menggunakan *QR Code*, yang memungkinkan proses login tanpa memasukkan *password* dan tetap memenuhi standar keamanan tinggi (Buranasaksee, 2024). Penelitian lain juga mengusulkan metode *smart login* melalui pemindaian *QR Code* yang dipadukan dengan *OTP* sebagai alternatif login tanpa perlu mengetikkan kredensial manual (Adelson *et al*., 2020). Kedua pendekatan tersebut menunjukkan potensi nyata penggunaan QR Code sebagai metode autentikasi dalam sistem SSO yang lebih praktis, efisien, dan tetap aman.

Saat ini teknologi QR Code telah digunakan secara luas di berbagai aspek digital karena kecepatan dan kemudahannya. Penelitian yang berjudul "Perancangan Sistem Absensi Pintar Mahasiswa Menggunakan Teknik QR Code dan Geolocation" menghubungkan *QR Code* dengan geolokasi menghindari absensi palsu, menegaskan bahwa *QR Code* cocok untuk skenario yang membutuhkan validasi lintas peringkat (Marlein Tamtelahitu *et al*., 2021). Meskipun konteksnya bukan autentikasi sistem login, keberhasilan ini menunjukkan bahwa *QR Code* mampu menangani proses otentikasi secara instan dan unik. Penelitian-penelitian tersebut memperlihatkan potensi *QR Code* sebagai alternatif login, khususnya dalam sistem SSO yang bertujuan menyederhanakan proses autentikasi. Untuk mendukung komunikasi *real-time* saat proses pemindaian QR Code, diperlukannya mekanisme komunikasi dua arah antara client dan server, salah satunya menggunakan WebSocket. Menurut penelitian yang berjudul "Perancangan dan Pembuatan Website Kuis Daring dengan Menggunakan *Websocket Communication Protocol*" WebSocket memungkinkan komunikasi *full-duplex* dimana komunikasi dua arah melalui satu koneksi *Transmission Control Protocol (TCP)* (Sena *et al*., 2024). Dalam konteks sistem login dengan QR Code, WebSocket dapat digunakan untuk memantau status pemindaian secara langsung dan refresh token secara berkala. Begitu QR Code dipindai dari aplikasi, server dapat langsung mengirimkan notifikasi ke browser untuk menyelesaikan login. Teknologi ini menghindari kebutuhan *polling* berkala yang akan membebani server, sehingga lebih efisien (RESWARA, 2024). Sistem server dan halaman login SSO akan dibangun menggunakan bahasa bahasa pemrograman Python, dan framework Flask. Flask yang merupakan *micro-framework* dipilih karena memiliki core inti yang sederhana, hal tersebut membuat *Flask* menjadi ringan (Ningtyas & Setiyawati, 2021). Selain itu Flask juga secara default telah menyertakan jinja sebagai template-engine, hal ini akan berguna untuk membangun sistem yang memiliki tampilan seperti halaman login (Gea & Susetyo, 2023).

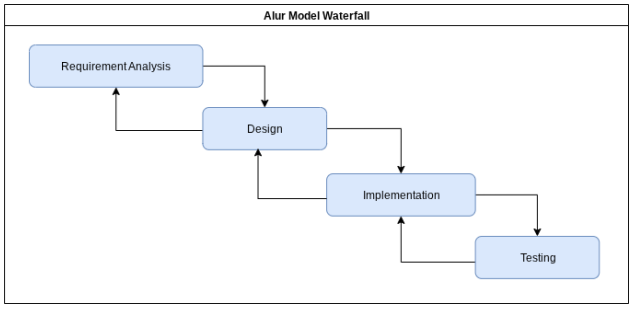
Menurut penelitian yang berjudul "*Comparative Analysis of React Native, Kotlin, and Flutter for Cross-Platform Mobile Development*" Flutter merupakan *Software Development Kit* (SDK) yang dirancang oleh Google dengan menggunakan bahasa pemrograman Dart untuk membangun aplikasi *cross-platform* (Usama Riaz, 2025). Flutter juga memiliki performa yang tinggi dalam lingkungan mobile karena dapat dikompilasi menjadi bahasa mesin (native) (Husain, 2023). Disamping itu, Flutter juga mudah digunakan dan memiliki *widget system* yang fleksibel untuk dikustomisasi (Hendriawan *et al*., 2023). Hal ini akan membantu dalam membuat aplikasi untuk memindai QR Code baik itu untuk ISO maupun Android. Dalam pengembangan sistem, pemilihan metode rekayasa perangkat lunak menjadi hal penting yang perlu diperhatikan.

Model pengembangan *Software Development Life Cycle* (SDLC) menjadi salah satu pendekatan yang sering digunakan dalam penelitian sistem informasi karena alurnya yang sistematis dan terdokumentasi dengan baik. Menurut penelitian yang berjudul "*Perancangan Aplikasi Batam Travel Menggunakan Metode Software Development Life Cycle (SDLC)*", SDLC merupakan proses sistematis yang digunakan oleh analis sistem

untuk membangun software, mulai dari penentuan kebutuhan, perancangan, validasi, sampai pelatihan dan penyerahan kepada konsumen (Hasanah & Nahrul Indriawan, 2021). Salah satu model populer dari SDLC adalah waterfall, yang menawarkan pendekatan linear dan sekuensial. Karakteristik Waterfall yang terstruktur menjadikan proses pengembangan menjadi sistematis.

2. Metodologi Penelitian

Penelitian ini menggunakan metode eksperimen dengan pendekatan pengembangan prototipe yang berfokus pada sistem login *Single Sign-On* (SSO) dengan QR Code. Model yang dipilih dalam pengembangan adalah model *Waterfall,* yang merupakan bagian dari *Software Development Life Cycle* (SDLC) (Rengganis *et al*., 2024). Model ini dipilih karena kebutuhan sistem telah didefinisikan secara jelas sejak awal berdasarkan hasil studi awal dan diskusi dengan tim IT internal PT XYZ. Dengan pendekatannya yang sistematis dan terstruktur, *Waterfall* memudahkan peneliti untuk mendokumentasikan setiap tahapan pengembangan mulai dari analisis kebutuhan, desain, implementasi, hingga pengujian.



Gambar 1. Alur model Waterfall

Analisis Kebutuhan (*Requirement analysis*)
Tahap analisis kebutuhan dimulai dengan pengumpulan dan analisis kebutuhan fungsional dan non-fungsional untuk sistem autentikasi SSO berbasis *QR Code*. Proses ini melibatkan stakeholder yang representatif dari PT. XYZ, terutama perwakilan dari departemen IT dan beberapa karyawan sebagai *user* akhir. Dalam sebuah rapat internal IT, permasalahan terkait proses login yang rumit diidentifikasi, di mana *user* seringkali harus mengingat banyak kredensial untuk aplikasi yang berbeda. Salah satu solusi yang dibahas adalah penggunaan *QR Code* untuk autentikasi, mengingat teknologi QR juga telah dipakai di sistem lain seperti absensi di PT. XYZ. Dalam tahap ini, kami juga melakukan validasi kebutuhan dengan melibatkan stakeholder melalui diskusi. Tujuannya adalah untuk memastikan bahwa semua kebutuhan sistem teridentifikasi dengan jelas, termasuk kebutuhan untuk kemudahan akses dan keamanan. Proses validasi dilakukan secara iteratif, dengan umpan balik dari stakeholder pada setiap tahap untuk memastikan bahwa solusi yang diusulkan benar-benar sesuai dengan masalah yang dihadapi *user* dan kebutuhan perusahaan.

Perancangan (*Design*)
Tahap ini mencakup pengembangan arsitektur sistem yang menggambarkan gambaran umum aplikasi dan komponen-komponen utama yang terlibat. Untuk merepresentasikan alur dan interaksi antar komponen secara visual, digunakannya pendekatan *Unified Modeling Language* (UML), dikarenakan UML menyediakan diagram yang bervariasi (Sumiati *et al*., 2021). Sistem ini terdiri dari tiga komponen utama, yaitu *frontend, backend*, dan aplikasi mobile. Komponen *frontend* mencakup *Client SSO* dan *Page Login SSO* yang berfungsi untuk menampilkan halaman login dengan *QR Code* serta memantau status login secara *real-time* melalui *WebSocket*. Komponen *backend*, yaitu *Server SSO*, bertugas untuk menghasilkan token *QR Code* dan menangani proses autentikasi pengguna. Sementara itu, aplikasi mobile digunakan sebagai alat identifikasi pengguna serta pemindai *QR Code* yang ditampilkan pada halaman login. Format token dalam sistem ini menggunakan UUID sebagai token autentikasi yang dihasilkan secara acak dan disimpan di *Redis*. Setiap *QR Code* akan diperbarui secara otomatis setiap 20 detik, sehingga menghasilkan UUID baru untuk mencegah *reuse token*. Setelah pengguna berhasil login, UUID tersebut disimpan di *Redis* dengan nilai *Time to Live* (TTL) 5 menit, dan TTL akan diperbarui setiap kali proses verifikasi dilakukan. Untuk menghindari *replay attack*, setiap UUID yang dipindai hanya dapat digunakan untuk satu sesi. Mekanisme ini memastikan bahwa UUID bersifat unik dan tidak dapat digunakan ulang, sehingga mencegah penyalahgunaan token oleh pihak yang tidak berwenang.

Implementasi (*Implementation*)

Pada tahap ini Flask akan digunakan sebagai framework backend untuk menangani API dan pengelolaan token. Aplikasi mobile dikembangkan menggunakan Flutter untuk authentikasi user dan untuk memindai QR Code. Untuk menjaga keamanan komunikasi, diterapkan *Hypertext Transfer Protocol Secure* (HTTPS) untuk akses web/API dan *WebSocket Secure* (WSS) untuk komunikasi *real-time* antar halaman login dan server. Penggunaan protokol terenkripsi ini bertujuan untuk mencegah serangan seperti *man-in-the-middle* dan *packet sniffing*. Sistem ini menerapkan dua skema logout. Pertama, logout aplikasi (web) yang berfungsi untuk mengakhiri sesi dengan cara menghapus token *Redis* hanya pada aplikasi yang sedang diakses oleh *user*. Kedua, logout global melalui aplikasi mobile, di mana seluruh token *user* yang tersimpan pada *Redis* akan dihapus sehingga seluruh akses ke aplikasi lain yang terhubung melalui mekanisme SSO juga ikut terhenti. Penerapan skema logout ini sejalan dengan prinsip *token revocation* dan *session termination* yang direkomendasikan dalam standar keamanan NIST SP 800-63-3, sehingga token yang telah kedaluwarsa ataupun yang berpotensi disalahgunakan tidak dapat digunakan kembali (National Institute of Standards and Technology (NIST), 2020).

Pengujian (*Testing*)

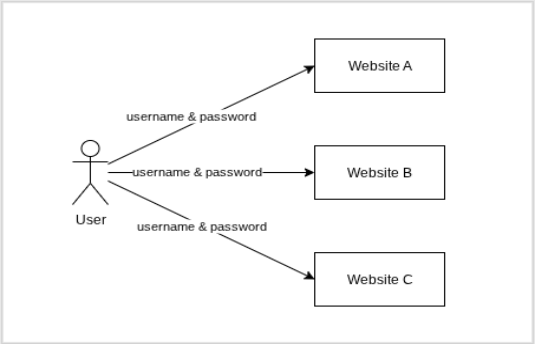
Tahap setelah sistem diimplementasikan, langkah untuk memastikan apakah aplikasi telah berjalan sesuai dengan yang diharapkan. Pengujian akan menggunakan *black box* testing untuk memastikan aplikasi yang dikembangkan sesuai dengan syarat fungsional, kebutuhan, dan spesifikasi yang telah ditentukkan oleh *user* (Ariyana *et al.*, 2023). Selain itu, dilakukan juga *User Acceptance Testing* (UAT) untuk memastikan aplikasi memenuhi harapan *user* akhir, dengan melibatkan responden yang menguji antarmuka, keamanan, kinerja, dan fungsionalitas aplikasi (Aliyah Aliyah *et al.*, 2024).

3. Hasil dan Pembahasan

Hasil

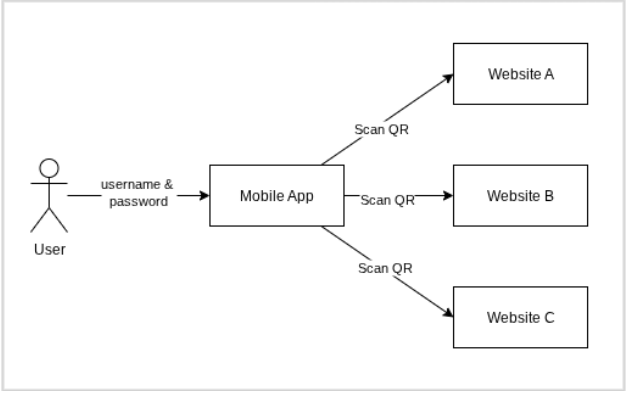
Analisis Kebutuhan (*Requirement Analysis*)

Dari hasil analisis, diketahui bahwa proses login konvensional menuntut *user* untuk mengingat kredensial. Dikarenakan PT. XYZ memiliki banyak aplikasi, maka user harus mengingat banyak kredensial. Solusi SSO berbasis *QR Code* bertujuan menyederhanakan login dengan cukup memindai QR menggunakan aplikasi *mobile* tanpa perlu mengingat ulang kredensial. Sistem harus mampu mengelola *QR Code* dengan batas waktu aktif, validasi *QR Code* sekali pakai, dan komunikasi *real-time* antara aplikasi mobile dengan halaman login. Untuk memahami analisis kebutuhan yang dikembangkan, penting untuk memahami kondisi autentikasi sebelum dan sesudah menerapkan SSO berbasis *QR Code*. Kondisi sebelum penelitian seperti Gambar 2, yang mana user harus memasukkan kredensial (*username* dan *password*) secara manual setiap ingin login ke aplikasi yang berbeda.



Gambar 2. Sistem login sebelum SSO

Solusi yang dikembangkan adalah sistem autentikasi SSO berbasis QR Code, seperti yang ditampilkan pada Gambar 3. Dalam pendekatan ini, halaman login menampilkan QR Code yang akan dipindai oleh aplikasi mobile yang telah terautentikasi oleh *user*. Proses ini memungkinkan user untuk login ke berbagai aplikasi tanpa perlu memasukkan *username* dan *password* secara berulang.

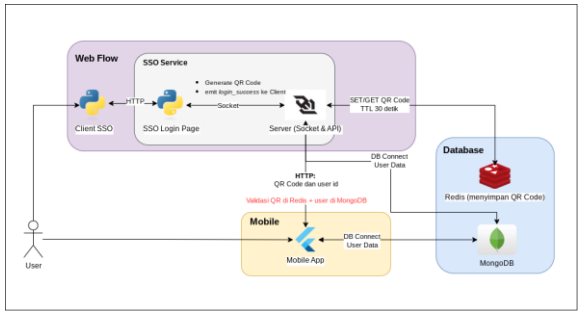


Gambar 3. Sistem login setelah SSO

Perancangan (*Design*)

1) Arsitektur Sistem

Berdasarkan kebutuhan yang telah dianalisis, tahap selanjutnya adalah merancang solusi secara teknis yang sesuai dengan hasil analisis kebutuhan. Perancangan ini digambarkan pada gambar arsitektur sistem dibawah yang menggambarkan alur komunikasi antar komponen utama, seperti *Frontend (Web Flow), Backend (SSO Service), Mobile App,* dan *Database*. Gambar arsitektur sistem ini menjadi dasar dalam pengembangan sistem secara menyeluruh.



Gambar 4. Arsitektur Sistem

Komponen arsitektur sistem diatas dapat dijelaskan sebagai berikut:

a) Client SSO (*Frontend* Integrasi)

Komponen ini merupakan bagian dari aplikasi pihak ketiga, aplikasi yang ingin menggunakan sistem SSO berbasis QR Code. Komponen ini bertujuan untuk mengarahkan user dari aplikasi pihak ketiga ke SSO Login Page saat autentikasi diperlukan. Komponen ini bertugas mengirimkan data yang terenkripsi yang mencakup informasi aplikasi seperti callback URL (alaman tujuan setelah login) dan client ID (identitas aplikasi yang didaftarkan pada sistem SSO). Dengan begitu SSO Login Page dapat mengetahui aplikasi apa yang melakukan request autentikasi dan kemana user harus diarahkan setelah proses autentikasi.

b) SSO *Login Page* (*Frontend Web*)

Komponen ini menampilkan QR Code yang akan dipindai oleh aplikasi mobile. QR Code didapatkan dari Server yang akan di refresh secara berkala. Selain itu, halaman ini akan membuka koneksi WebSocket dengan server untuk menerima notifikasi status login secara real-time.

c) Server (Web*Socket* dan *API Backend*)

Komponen ini merupakan *backend service* sebagai inti dari sistem *SSO QR Code*. Tugas utama komponen ini meliputi pembuatan *QR Code* baru untuk halaman *SSO Login Page* yang akan aktif selama disimpan di Redis. Selain itu, server menyediakan *endpoint API* yang berkaitan dengan autentikasi, seperti memvalidasi token dan autentikasi untuk aplikasi mobile. Komponen ini juga bertanggung jawab dalam mengelola koneksi *WebSocket* dan memberikan notifikasi status login secara *real-time*.

d) Aplikasi Mobile

Komponen ini merupakan aplikasi yang bertindak sebagai identitas user yang telah login. Tugas utamanya meliputi penyediaan proses autentikasi untuk memastikan pengguna dapat dikenali secara valid, serta menyediakan fitur pemindaian *QR Code* pada halaman login yang kemudian dikirimkan ke server untuk divalidasi.

e) Database

Komponen untuk menyimpan data, baik itu data user yang akan di simpan di *MongoDB* dan data *QR Code* yang akan disimpan sementara di *Redis*.

2) Teknologi yang digunakan

Berdasarkan arsitektur sistem yang telah dibuat sebelumnya beserta komponenya, masing-masing komponen perlu diimplementasikan dengan teknologi yang sesuai. Teknologi yang akan digunakan telah dicantumkan pada diagram arsitektur. Pada bagian ini akan dijelaskan lebih rinci mengenai spesifikasi dan alasan dari teknologi yang digunakan.

a) Client SSO

Memanfaatkan *Python* karena bahasa ini telah dipakai luas di lingkungan PT. XYZ dan mudah diintegrasikan dengan komponen lain.

b) Halaman login SSO

Dibangun dengan *Flask* dan *template engine Jinja* untuk merender *QR Code*, sedangkan komunikasi *real-time* dilakukan melalui *WebSocket*. Kombinasi ini memberi antarmuka ringan sekaligus interaktif.

c) Server API

Menggunakan *Flask* sebagai kerangka kerja utama untuk penyediaan endpoint dan pembuatan *QR Code*; dipilih karena ringan, intuitif, dan terintegrasi langsung dengan *Jinja*.

d) Komunikasi soket

Ditangani oleh pustaka *Flask-SocketIO*, yang

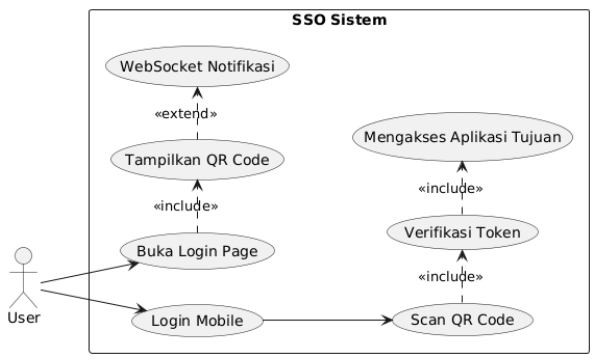
populer dalam ekosistem *Flask* dan mendukung koneksi dua arah berbasis *WebSocket* tanpa konfigurasi rumit.

e) Aplikasi mobile
Dikembangkan menggunakan *Flutter* (bahasa *Dart*) agar menghasilkan aplikasi *cross-platform* yang sama-sama optimal di Android dan iOS.

f) Database data user
Disimpan di *MongoDB*, karena PT. XYZ menggunakan *MongoDB* sebagai standar database utama dan kapabel menangani data dokumen skema-longgar.

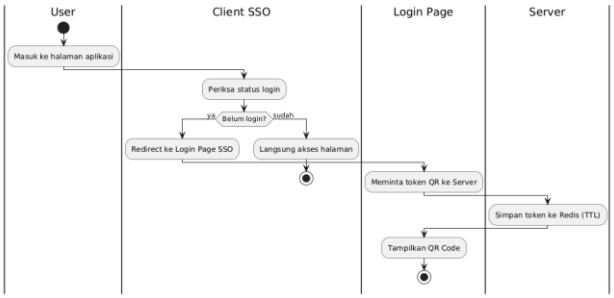
g) Penyimpanan token sementara
Mengandalkan *Redis* karena basis memori berkecepatan tinggi ini mendukung TTL (*Time-To-Live*), cocok untuk token sementara yang harus kedaluwarsa otomatis.

3) *Unified Modeling Language* (UML)
Setelah merancang arsitektur sistem dan menentukan teknologi apa yang dipakai, tahap selanjutnya merancang alur komunikasi antar komponen melalui diagram UML. Diagram UML yang digunakan meliputi *Use Case Diagram, Activity Diagram,* dan *Sequence Diagram*. Masing-masing diagram digunakan untuk menggambarkan skenario autentikasi *QR Code* dari berbagai sudut pandang, mulai dari gambaran fungsional hingga interaksi antar komponen. *Use Case Diagram* pada Gambar 5 menggambarkan hubungan antara aktor (*user*) dan sistem SSO. Diagram ini menunjukkan fitur-fitur utama yang digunakan user, seperti membuka halaman login yang menampilkan *QR Code*, login aplikasi mobile, serta proses pemindaian *QR Code* yang disertai verifikasi token.



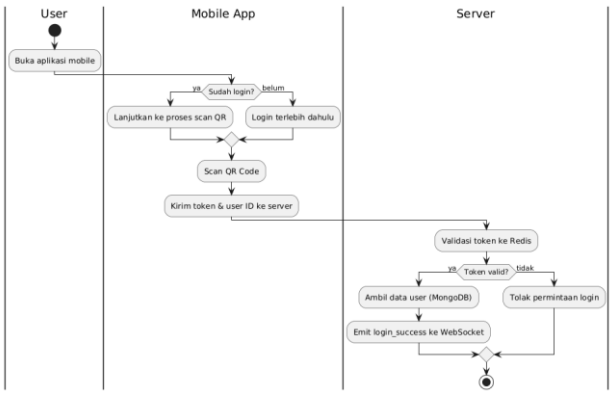
Gambar 5. Use Case Diagram

Sementara itu, *Activity Diagram* digunakan untuk menggambarkan alur aktivitas pada dua proses utama. Diagram Gambar 6 menunjukkan alur aktivitas user mulai dari mengakses aplikasi sampai berhasil menampilkan *QR Code*. Proses diawali dari sistem memastikan apakah user telah login. Jika sudah, maka langsung menuju halaman yang diharapkan. Jika belum, maka user akan dialihkan ke halaman login SSO yang akan mengirimkan permintaan token QR ke server untuk ditampilkan. Token *QR Code* akan otomatis di refresh setiap 20 detik untuk menjaga keamanan token.



Gambar 6. Activity Diagram Client Website

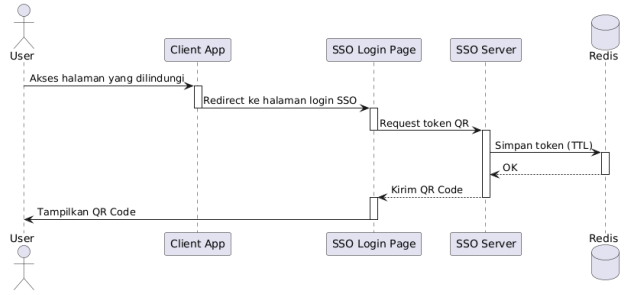
Diagram Gambar 7 menunjukkan alur aktivitas user mulai dari membuka aplikasi mobile sampai proses autentikasi berhasil. Proses diawali dari sistem memastikan apakah user telah login ke aplikasi mobile. Jika sudah, maka user dapat mengakses fitur pemindaian QR dan mengirimkan hasil pemindaian ke server untuk divalidasi. Jika belum, maka user harus login terlebih dahulu sebelum mengakses fitur pemindaian QR.



Gambar 7. Activity Diagram Aplikasi Mobile

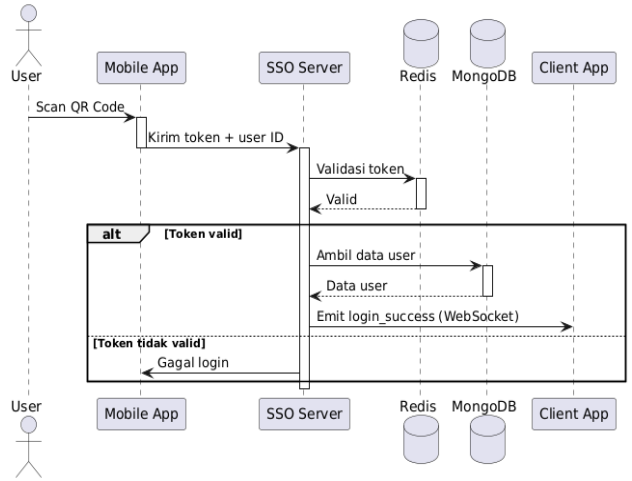
Sequence diagram digunakan untuk menggambarkan komunikasi antar komponen sistem berdasarkan urutan waktu. Dalam sistem SSO yang dikembangkan,

terdapat dua sequence diagram utama yang merepresentasikan dua proses kritis, yaitu login melalui client SSO dan autentikasi melalui aplikasi mobile. Gambar 8 menunjukkan *sequence diagram* proses login melalui client SSO. Proses autentikasi dimulai dari user mengakses website yang belum di otentikasikan. *Client SSO* akan mengarahkannya ke halaman login dengan menyertakan enkripsi data aplikasi (*url callback* dan *client id*), yang akan membuat koneksi WebSocket dengan server untuk meminta QR Code baru setiap beberapa saat. *Server SSO* akan membuat token QR Code baru dan menyimpannya pada Redis. dan mengembalikan *QR Code* ke halaman login untuk ditampilkan.



Gambar 8. Sequence Diagram Client Website

Sementara itu, Gambar 9 memperlihatkan *sequence diagram* autentikasi melalui pemindaian *QR Code* pada aplikasi mobile. Diagram ini menunjukkan alur mulai dari pemindaian *QR Code*, pengiriman token ke server, proses verifikasi token, hingga pengiriman notifikasi keberhasilan login kepada *Client SSO* untuk mengarahkan pengguna ke halaman utama.



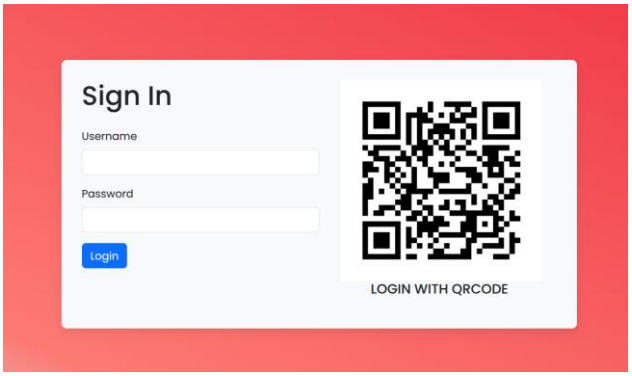
Gambar 9. Sequence Diagram Aplikasi Mobile

Implementasi (*Implementation*)

Setelah perancangan selesai, selanjutnya akan masuk ke tahap implementasi yang akan merubah desain sistem menjadi program yang fungsional. Pada tahapan ini, dilakukannya pengembangan dan integrasi dari seluruh komponen sistem sesuai dengan teknologi yang telah dipilih sebelumnya.

1) Implementasi Komponen

Implementasi dimulai dari masing-masing komponen utama yang telah dirancang sebelumnya. Komponen pertama adalah *Client SSO*, yaitu program yang ditanamkan ke aplikasi pihak ketiga yang menggunakan sistem login berbasis *QR Code*. Fungsi utama dari komponen ini adalah sebuah *decorator* yang digunakan untuk mengecek apakah *user* sudah login atau belum. Jika belum maka pindahkan ke halaman login dengan menambahkan *query param* 'data' pada URL sebagai identitas aplikasi yang sedang akses. Komponen berikutnya adalah halaman login SSO yang diimplementasikan menggunakan *HTML, CSS*, dan *JavaScript*. Komponen ini akan membuka koneksi ke *WebSocket* server untuk mendapatkan *QR Code* sekaligus menunggu notifikasi status login. Hasil tampilan dapat dilihat pada Gambar 11, terdapat dua cara untuk login menggunakan *QR Code* dan *username* dan *password* konvensional.



Gambar 10. Tampilan login QR Code

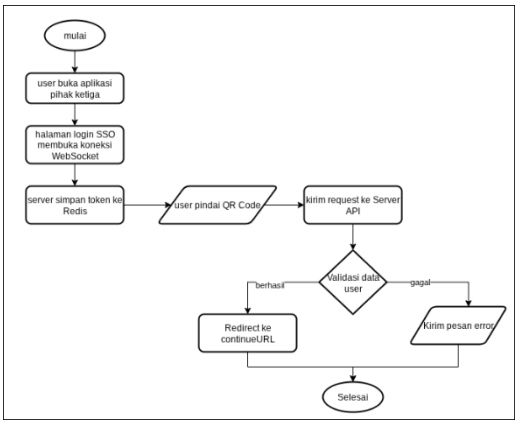
Selanjutnya adalah komponen server yang bertugas menangani API dan komunikasi *socket*. Server dikembangkan menggunakan *framework Flask* untuk menangani permintaan API, serta pustaka *Flask-SocketIO* untuk membangun koneksi *WebSocket*. Saat halaman login terhubung melalui *WebSocket*, server akan membuat *thread* baru untuk mengirimkan *QR Code* baru setiap 20 detik. Token QR disimpan di *Redis* dengan mengatur *Time to Live* (TTL) selama 20 detik

agar token bersifat sementara. Hasil tampilan utama aplikasi mobile, terdapat bagian informasi terkait user yang berhasil login ke aplikasi. Terdapat juga bagian yang berisikan informasi terkait Riwayat login aplikasi kantor dengan QR. Lalu bagian yang terpenting yaitu tombol untuk memindai QR, yang akan meminta permission untuk mengakses kamera untuk memindai QR.

Gambar 11. Tampilan aplikasi mobile

2) Implementasi Integrasi

Setelah setiap komponen sistem berhasil diimplementasikan secara terpisah, selanjutnya dilakukan proses integrasi antar komponen sistem. Integrasi ini memastikan seluruh komponen dapat berinteraksi satu sama lain. Tujuannya agar terbentuknya alur fungsionalis sesuai dengan rancangan awal. Alur proses integrasi antar sistem digambarkan pada Gambar 12.



Gambar 12. Flowchart system

Alur diawali ketika *user* membuka aplikasi pihak ketiga (*Client SSO*) yang belum terautentikasi, lalu diarahkan ke halaman login *SSO*. Halaman login *SSO* akan membuka koneksi dengan *WebSocket* untuk mendapatkan *QR Code* baru secara berkala, sekaligus menunggu notifikasi status login. Sementara itu, server akan menyimpan token sementara ke dalam *Redis*. Ketika user memindai *QR Code* melalui aplikasi mobile, aplikasi akan mengirimkan *request* ke *Server API* yang berisikan token *QR Code* dan informasi user. *Server* akan memverifikasi apakah data user sesuai, jika sesuai maka server akan mengirimkan notifikasi ke halaman *login SSO* melalui *WebSocket* kalau proses login telah berhasil. Setelah menerima notifikasi, halaman login *SSO* akan memindahkan page ke *continueURL* yang telah terenkripsi pada *query param* 'data'. Proses ini akan membuktikan bahwa integrasi antar komponen sistem telah berhasil dilakukan.

Pengujian (Testing)

Setelah sistem autentikasi *SSO QR Code* berhasil diimplementasikan, langkah berikutnya adalah melakukan uji coba aplikasi. Pengujian dilakukan dengan menggunakan metode pengujian *black box*. Hasil pengujian dapat dilihat pada *Table 1* berikut.

Tabel 1. Hasil pengujian black box

Langkah Pengujian	Hasil yang Diharapkan	Status
Akses halaman home aplikasi pihak ketiga tanpa login	Halaman login SSO dan QR Code tampil	Lulus
Menunggu QR Code setelah TTL	QR Code otomatis diperbarui	Lulus
Login ke aplikasi mobile menggunakan username dan password	Berhasil masuk ke halaman utama aplikasi mobile	Lulus
Memindai QR Code valid dari aplikasi mobile	Data terkirim dan mendapat respons sukses dari server	Lulus

Memindai QR Code yang tidak valid	Respons gagal dari server karena QR tidak valid	Lulus
Token valid dan user valid	Browser diarahkan ke halaman utama aplikasi pihak ketiga	Lulus
Token QR digunakan ulang	Server memberikan respons gagal karena token telah digunakan	Lulus

Setelah seluruh skenario pengujian *black-box* pada Tabel 1 dinyatakan lulus, tahap selanjutnya adalah melakukan pengujian laboratorium untuk mengevaluasi kinerja autentikasi. Pengujian ini bertujuan untuk membandingkan waktu login pada sistem *SSO QR Code* yang dikembangkan dengan dua metode autentikasi konvensional, yaitu OTP (email) dan *Magic Link*. Setiap metode diuji sebanyak 10 kali dalam kondisi jaringan yang sama, dengan waktu pengukuran dimulai sejak pengguna mengakses halaman login hingga berhasil diarahkan ke halaman dashboard.

Tabel 2. Metode Autentifikasi

Metode Autentikasi	Rata-rata waktu (detik)
SSO QR Code	3,2 detik
OTP Email	8,4 detik
Magic Link	7,6 detik

Untuk *SSO QR Code*, *user* hanya perlu membuka aplikasi mobile lalu memindai QR Code yang ditampilkan di halaman login, yang mengarah langsung ke dashboard yang diharapkan setelah pemindaian. Kecepatan login ini terbukti lebih cepat, dengan rata-rata 3,2 detik. Untuk *OTP Email*, *user* terlebih dahulu memasukkan *username* dan *password*, kemudian menunggu untuk menerima OTP yang dikirim melalui email, dan akhirnya memasukkan OTP tersebut. Proses ini memakan waktu lebih lama, sekitar 8,4 detik, karena adanya langkah tambahan seperti menunggu email OTP dan memasukkan kode OTP. Sedangkan untuk *Magic Link*, *user* harus memasukkan *email* untuk menerima tautan login. Setelah membuka email dan mengklik link, mereka diarahkan ke aplikasi. Proses ini memakan waktu lebih lama, yaitu 7,6 detik, karena menunggu pengiriman *email* dan membuka tautan. Dari hasil ini, dapat disimpulkan bahwa *SSO QR Code* lebih efisien dalam hal kecepatan dan pengalaman *user* yang lebih baik dibandingkan dengan kedua metode lainnya. Setelah uji laboratorium, langkah berikutnya adalah *User Acceptance Testing* (UAT). Pengujian ini bertujuan untuk memastikan bahwa aplikasi sesuai dengan kebutuhan pengguna dan dapat berfungsi dengan baik dalam lingkungan nyata. Dalam pengujian UAT ini, sebanyak 21 responden terlibat, yang terdiri dari 8 karyawan dan 13 karyawan magang dengan rentang usia antara 21 hingga 28 tahun. Semua responden memiliki pengalaman dalam penggunaan aplikasi berbasis web dan mobile. Pengujian dilakukan dengan menggunakan skala penilaian dari 1 (sangat tidak setuju) hingga 5 (sangat setuju) untuk menilai berbagai aspek fungsionalitas dan usability aplikasi. Pertanyaan yang diajukan terkait dengan kemudahan penggunaan, kinerja, keamanan, dan kepuasan pengguna terhadap aplikasi yang diuji.

Tabel 3. Pertanyaan UAT

(P)	Pertanyaan	Skor
P1	Proses login SSO menggunakan QR Code mudah digunakan	1-5
P2	Saya dapat dengan mudah memindai QR Code menggunakan aplikasi mobile	1-5
P3	Saya dapat dengan mudah memindai QR Code menggunakan aplikasi mobile	1-5
P4	Saya merasa login menggunakan QR Code lebih aman daripada mengetikkan password.	1-5
P5	Saya merasa yakin bahwa data saya tetap aman saat menggunakan sistem login ini.	1-5
P6	Sistem merespons dengan cepat setelah saya memindai QR Code.	1-5
P7	Tidak terdapat kendala secara teknis selama proses login berlangsung	1-5
P8	Sistem login ini mempermudah saya dalam mengakses berbagai aplikasi tanpa perlu memasukkan kredensial berulang kali.	1-5
P9	Penggunaan sistem login ini menghemat waktu saya dibanding metode login tradisional.	1-5
P10	Saya ingin sistem login ini digunakan juga di aplikasi kantor lainnya.	1-5

Berdasarkan hasil pengisian kuesioner, diperoleh total skor keseluruhan sebesar 950 dari seluruh responden.

$$Rata\ Rata\ UAT = \left(\frac{Total\ skor\ semua\ responden}{Jumlah\ responden\ x\ Jumlah\ pertanyaan\ x\ 5}\right) x\ 100$$
$$= \left(\frac{950}{21\ x\ 10\ x\ 5}\right)\ x\ 100 = 90{,}4\%$$

Rata-rata UAT menunjukkan hasil 90,4%, yang mencerminkan penerimaan yang sangat baik terhadap aplikasi. Hasil ini menunjukkan bahwa aplikasi dinilai sangat baik oleh responden dalam beberapa aspek utama, yaitu kemudahan penggunaan, kinerja, keamanan, dan kepuasan pengguna secara keseluruhan.

1) Kemudahan penggunaan (P1-P3)
   Pengujian pada akspek kemudahan penggunaan memperoleh rata-rata skor 4,6, yang menunjukkan bahwa proses login menggunakan QR Code dan tampilan halaman login sangat mudah dipahami oleh *user*. Aplikasi ini dirancang dengan antarmuka intuitif, memudahkan pengguna untuk autentikasi secara cepat dan efisien.
2) Keamanan dan Kepercayaan (P4–P5)
   Aspek keamanan memperoleh rata-rata skor 4,3, yang menunjukkan bahwa meskipun *user* merasa sistem lebih aman dibandingkan dengan login konvensional, masih ada sedikit kekhawatiran terkait kepercayaan terhadap sistem. Hasil UAT menunjukkan bahwa mayoritas pengguna merasa login menggunakan *QR Code* lebih aman dan yakin data mereka tetap aman, dengan skor yang menunjukkan tingkat kepuasan tinggi pada kedua pernyataan terkait. Namun, meskipun persepsi keamanan ini positif, penting untuk memastikan bahwa sistem ini terus diuji untuk mengatasi potensi risiko yang mungkin belum terlihat dalam pengujian *user*, seperti ancaman keamanan yang lebih teknis.
3) Kinerja Sistem (P6–P7)
   Pada akspek kinerja sistem memperoleh rata-rata skor 4,4, menunjukkan respons cepat dan kestabilan tinggi selama proses login. Kecepatan dan keandalan ini sangat penting dalam konteks organisasi yang mengutamakan efisiensi dan pengalaman pengguna yang lancar.
4) Kepuasan Umum (P8–P10)
   Dengan rata-rata skor 4,5, sistem berhasil mempermudah akses ke berbagai aplikasi lain tanpa perlu memasukkan kredensial berulang kali, menghemat waktu *user*. Keinginan untuk mengimplementasikan sistem ini di aplikasi kantor lainnya menunjukkan potensi adopsi yang luas di lingkungan kerja.

Pembahasan

Pembahasan ini mengonfirmasi bahwa pengembangan sistem *Single Sign-On* (*SSO*) berbasis *QR Code* di PT. XYZ memberikan solusi praktis atas permasalahan pengelolaan kredensial yang kompleks, sebagaimana dikemukakan oleh Fauzi *et al.* (2023) yang menekankan pentingnya penyederhanaan proses autentikasi lintas aplikasi. Pendekatan menggunakan *QR Code* mengacu pada temuan Triyanto Anggoro dan Budi Slamet Rianto (2025) yang membuktikan efektivitas pemindaian *QR Code* dalam sistem absensi, memperkuat argumen bahwa teknologi ini mampu menghadirkan autentikasi cepat dan mudah. Selain itu, mekanisme token unik berbasis *UUID* dengan pembaruan berkala yang diterapkan sesuai standar keamanan NIST (2020) memperkuat aspek perlindungan terhadap penyalahgunaan, sejalan dengan rekomendasi keamanan yang diusulkan oleh National Institute of Standards and Technology. Pengujian fungsional menggunakan metode *black-box* menunjukkan kesesuaian hasil dengan studi Ariyana *et al.* (2023) yang menegaskan pentingnya validasi fitur secara menyeluruh untuk memastikan kualitas aplikasi. Dalam hal performa, waktu autentikasi rata-rata 3,2 detik yang dicapai sistem ini mengungguli metode OTP email dan *Magic Link*, yang juga diamati oleh Adelson *et al.* (2020) dalam penelitian mereka mengenai alternatif login berbasis *QR Code* dan OTP. Temuan ini menegaskan potensi *QR Code* sebagai metode autentikasi yang tidak hanya efisien tetapi juga mengurangi beban pengguna dalam mengelola kredensial, sesuai dengan hasil studi Fernando *et al.* (2023) terkait tantangan manajemen password. Evaluasi melalui *User Acceptance Testing* (*UAT*) dengan skor 90,4% mengindikasikan tingkat kepuasan pengguna yang tinggi, konsisten dengan hasil Aliyah Aliyah *et al.* (2024) yang menyoroti pentingnya penerimaan pengguna dalam keberhasilan implementasi sistem informasi. Namun, persepsi

keamanan yang positif tetap perlu diimbangi dengan penguatan mekanisme keamanan, terutama untuk mengantisipasi ancaman seperti *man-in-the-middle* dan *replay attack*, sebagaimana disarankan oleh Syarif Aziz *et al.* (2021). Integrasi autentikasi *multi-factor* dan audit keamanan lanjutan menjadi langkah penting untuk menjaga keandalan sistem, mengacu pada praktik terbaik dalam pengembangan sistem autentikasi modern. Selain itu, perluasan dukungan *Client SSO* agar kompatibel dengan berbagai bahasa pemrograman akan meningkatkan fleksibilitas integrasi, memperhatikan kebutuhan beragam platform teknologi yang digunakan di PT. XYZ, sebagaimana dianjurkan oleh Gumeraruloh Arianto *et al.* (2025). Penelitian ini juga membuka peluang pengembangan lebih lanjut dengan mengadopsi standar federasi seperti *SAML* dan *OpenID Connect*, yang telah terbukti meningkatkan interoperabilitas dan skalabilitas sistem autentikasi dalam berbagai konteks (Buranasaksee, 2024). Secara keseluruhan, hasil penelitian ini memperkuat dan memperluas temuan-temuan terdahulu dengan menghadirkan solusi autentikasi berbasis *QR Code* yang praktis, efisien, dan aman, khususnya dalam lingkungan korporasi yang memiliki kebutuhan akses aplikasi yang kompleks dan beragam.

4. Kesimpulan dan Saran

Penelitian ini berhasil merancang sistem *Single Sign-On* (*SSO*) berbasis *QR Code* sebagai solusi untuk mengatasi proses login yang berulang menjadi terpusat, di mana pengguna hanya perlu melakukan satu kali *login* pada aplikasi *mobile* dan kemudian dapat mengakses berbagai aplikasi kantor lainnya hanya dengan memindai *QR Code*. Penggunaan *QR Code* terbukti mampu menyederhanakan proses autentikasi tanpa mengorbankan aspek keamanan. Pendekatan model *Waterfall* diterapkan secara konsisten dalam pengembangan sistem ini. Hasil pengujian *black box* dan *User Acceptance Testing* (*UAT*) menunjukkan bahwa seluruh fungsi sistem berjalan dengan baik dan tingkat kepuasan pengguna mencapai 90,4%. Dengan demikian, sistem ini dinilai siap untuk diimplementasikan dan mampu memenuhi ekspektasi pengguna dalam meningkatkan efisiensi serta keamanan proses *login*. Secara kuantitatif, sistem berhasil mencapai target skor *UAT* ≥85% dan waktu *login* rata-rata kurang dari 5 detik, menandakan efisiensi dan keamanan yang memadai. Secara praktis, sistem ini berpotensi meningkatkan efisiensi operasional serta memperbaiki pengalaman pengguna dalam mengakses berbagai aplikasi tanpa perlu memasukkan kredensial secara berulang. Meskipun demikian, pengembangan lebih lanjut tetap diperlukan, khususnya dalam memperkuat keamanan aplikasi *mobile* untuk mencegah serangan seperti *man-in-the-middle* dan *replay attacks*, meskipun komunikasi sudah terlindungi oleh protokol *HTTPS* dan *WSS*. Implementasi autentikasi *multi-factor* (*MFA*) pada aplikasi *mobile* dapat menambah lapisan perlindungan tambahan, mengingat aplikasi tersebut menjadi titik akses utama ke berbagai aplikasi kantor. Selain itu, perluasan dukungan sistem *Client SSO* yang saat ini hanya kompatibel dengan bahasa pemrograman Python agar dapat mendukung berbagai bahasa lain akan meningkatkan fleksibilitas dan memperluas integrasi dengan platform teknologi yang beragam di lingkungan perusahaan. Penelitian lanjutan disarankan untuk mengintegrasikan standar federasi seperti *Security Assertion Markup Language* (*SAML*) atau *OpenID Connect* (*OIDC*) guna meningkatkan interoperabilitas dan skalabilitas sistem. Audit keamanan melalui *penetration testing* dan *vulnerability scanning* menggunakan alat seperti *OWASP ZAP* atau *Burp Suite* juga perlu dilakukan untuk mengidentifikasi potensi celah keamanan yang belum terdeteksi. Selain itu, *load testing* penting untuk memastikan sistem mampu menangani ribuan pengguna secara simultan tanpa mengalami penurunan kinerja, serta penguatan sisi *mobile* untuk melindungi perangkat dari risiko keamanan yang mungkin muncul.

5. Daftar Pustaka

Adelson, L., Siregar, D., & Chiuloto, K. (2020). Smart Login Pada Website Dengan Menggunakan Qr Code Dan Otentikasi One Time Password. *SNASTIKOM*, *2020*, 425-430.

Alijoyo, F. A. (2024). PENGEMBANGAN SINGLE SIGN ON (SSO) MENGGUNAKAN TEKNOLOGI MAGIC LINK DI UNIVERSITAS MUHAMMADIYAH SUKABUMI (UMMI). *Jurnal Informatika dan Rekayasa Elektronik*, *7*(1), 115-123.

Aliyah Aliyah, N., Hartono, N., & Muin, A. A. (2024). Penggunaan User Acceptance Testing (UAT) Pada Pengujian Sistem Informasi Pengelolaan Keuangan Dan Inventaris Barang. *Switch: Jurnal Sains Dan Teknologi Informasi, 3*(1), 84–100. https://doi.org/10.62951/switch.v3i1.330.

ANGGORO, T., & RIANTO, B. S. (2025). Perancangan Sistem Absensi Kehadiran Karyawan Menggunakan Barcode Berbasis Client Server. *JURNAL ILMIAH TEKNIK INDUSTRI DAN INOVASI*, *3*(2), 49-57.

Ariyana, R. Y., Susanti, E., Ath-Thaariq, M. R., & Apriadi, R. (2023). Penerapan Uji Fungsionalitas Menggunakan Black Box Testing pada Game Motif Batik Khas Yogyakarta. *JUMINTAL: Jurnal Manajemen Informatika Dan Bisnis Digital, 2*(1), 33–43. https://doi.org/10.55123/jumintal.v2i1.2371.

Aziz, A. S., & Safriatullah, S. (2021). Perancangan Dan Analisis Keamanan Pada Sistem Autentikasi Terpusat Freeradius. *Journal of Informatics and Computer Science*, *7*(2), 106-112.

Buranasaksee, P. (2024). Online Conference) Procedia of Multidisciplinary Research Article No (Vol. 2, Issue 5).

Chacko, R., Gossler, H., Riedel, J., Schunk, S. A., & Deutschmann, O. (2023). Digitalization in Catalysis and Reaction Engineering: Automatizing Work Flows. *Proceedings of the Conference on Research Data Infrastructure, 1*. https://doi.org/10.52825/cordi.v1i.412.

Fauzi, E., Yuliani, S., Syukriyah, Y., & Zakiah, A. (2023). Model Implementasi Nft Pada Web Sso Melalui Protokol Openid Connect Dan Oauth 2.0 Model of Nft Implementation on Web Sso Over Openid Connect and Oauth 2.0 Protocols. *Journal of Information Technology and Computer Science (INTECOMS)*, *6*(2).

Fernando, W. P. K., Dissanayake, D. A. N. P., Dushmantha, S. G. V. D., Liyanage, D. L. C. P., & Karunatilake, C. (2023). Challenges and Opportunities in Password Management: A Review of Current Solutions. *Sri Lanka Journal of Social Sciences and Humanities, 3*(2), 9–20. https://doi.org/10.4038/sljssh.v3i2.96.

Gea, J., & Susetyo, Y. A. (2023). Implementasi Framework Flask Pada Modul Beta-App Pada Aplikasi Sistem Informasi Helpdesk (Sih) Studi Kasus Pt XYZ. *J. Inform*, *23*(2), 243-258.

Gumeraruloh Arianto, I., Witanti, W., Ashaury, H., Informatika, T., & Jenderal Achmad Yani, U. (2025). Sistem Keamanan Otentikasi Pengguna pada Modul Single Sign On Menggunakan OAuth 2.0 dan One Time Password. *6*(1).

Hasanah, N., & Nahrul Indriawan, M. (2021). Rancangan Aplikasi Batam Travel Menggunakan Metode Software Development Life Cycle (SDLC). *Vol. 1, Issue 1*.

Hendriawan, M., Haryono, H., & Budiman, T. (2023). Development of water level monitoring applications in smart home systems using Flutter. *Journal of Information System, Informatics and Computing*, *7*(2), 213-240. https://doi.org/10.52362/jisicom.v7i2.1197.

Husain, I., Purwantoro, P., & Carudin, C. (2023). Analisis Performa State Management Provider Dan Getx Pada Aplikasi Flutter. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *7*(2), 1417-1422.

National Institute of Standards and Technology (NIST). (2020). *Digital Identity Guidelines* (Special Publication 800-63-3).

Nazar, M., Nurmalahayati, N., Rusman, R., Puspita, K., & Haris, A. (2022). Introducing Chemical Instruments through Quick Response Code (QR-Code) Based Website. *Jurnal Penelitian Pendidikan IPA, 8*(3), 1083–1088. https://doi.org/10.29303/jppipa.v8i3.1361.

Ningtyas, D. F., & Setiyawati, N. (2021). Implementasi Flask Framework pada Pembangunan Aplikasi Purchasing Approval Request. *Jurnal Janitra Informatika Dan Sistem Informasi, 1*(1), 19–34. https://doi.org/10.25008/janitra.v1i1.120.

Rengganis, P. N., Suhayati, M. S. M., & Sutara, B. S. B. (2024). THE APPLICATION OF THE SDLC WATERFALL METHOD IN DEVELOPING AN AUDIT APPLICATION FOR THE SUMEDANG REGENCY INSPECTORATE. *Jurnal Riset Teknik Informatika*, *1*(2), 139-145.

Reswara, R. M. (2024). *RANCANG BANGUN APLIKASI WEB UNTUK PELACAKAN KENDARAAN MENGGUNAKAN NODEJS DAN FRAMEWORK LARAVEL* (Doctoral dissertation, Sekolah Tinggi Teknologi Terpadu Nurul Fikri).

Riaz, M. U. (2025). Comparative Analysis of React Native, Kotlin, and Flutter for Cross-Platform Mobile Development.

Sena, I. G. W., Pattiasina, T. J., Basatha, R., & Reinaldo, N. G. (2024). Perancangan dan Pembuatan Website Kuis Daring dengan Menggunakan Websocket Communication Protocol. *4*(1).

Sumiati, M., Abdillah, R., & Cahyo, A. (2021). Pemodelan UML untuk Sistem Informasi Persewaan Alat Pesta. *Jurnal Fasilkom, 11*, 79–86.

Tamtelahitu, T. M., Sambono, J., & Unenor, J. E. (2021). Perancangan Sistem Absensi Pintar Mahasiswa Menggunakan Teknik Qr Code Dan Geolocation. *JIPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, *6*(1), 114-125.

Venkata, A. K. P., Palaparthy, H., Gudala, L., Reddy, V. K., & Vangoor, S. C. Balancing Security And Convenience: Sso And Oauth For Healthcare Data In Aws Govcloud.