

Penerapan Google Auth Sebagai Sistem Login Pada Website Berbasis React dengan Framework NextJS

Daniel Giovani Liem^{*1}, Dr. Ir. Anang Kukuh Adisusilo ST., MT..²,

^{1,2} Universitas Wijaya Kusuma Surabaya

Email: ¹danielgiovanieliem457@gmail.com, ²anang65@uwks.ac.id

^{*}Penulis Korespondensi

Abstrak

Perkembangan teknologi website modern mendorong penggunaan sistem login yang terintegrasi dengan database untuk mengelola data pengguna secara efektif dan efisien. Sistem autentikasi konvensional yang menggunakan username dan password masih banyak diterapkan karena mudah diimplementasikan. Namun, metode tersebut memiliki berbagai kelemahan, seperti risiko serangan SQL Injection, kebocoran data akun, serta human error ketika pengguna membuat atau mengelola akun, misalnya penggunaan password yang lemah atau lupa kredensial login. Selain itu, proses registrasi manual sering dianggap kurang praktis karena pengguna harus mengisi data dan melakukan verifikasi secara terpisah.

Permasalahan tersebut menunjukkan bahwa sistem autentikasi tradisional masih memiliki celah keamanan dan kurang memberikan kemudahan bagi pengguna. Kerentanan pada pengelolaan database dapat dimanfaatkan oleh pihak tidak bertanggung jawab untuk memperoleh akses ilegal, sedangkan kesalahan pengguna dalam pembuatan akun dapat mengurangi efektivitas keamanan sistem. Oleh karena itu, diperlukan solusi autentikasi yang lebih aman, praktis, dan efisien untuk meningkatkan pengalaman pengguna sekaligus meminimalkan risiko keamanan.

Tujuan dari penelitian atau pengembangan ini adalah menerapkan autentikasi menggunakan Google Authentication (Google Auth) sebagai alternatif sistem login modern. Dengan memanfaatkan layanan autentikasi Google, proses login dapat dilakukan secara lebih cepat dan praktis tanpa perlu membuat akun baru secara manual. Selain itu, penggunaan Google Auth membantu meningkatkan keamanan sistem melalui mekanisme autentikasi yang telah terstandarisasi, sehingga dapat mengurangi risiko SQL Injection, penyalahgunaan akun, serta human error dalam pengelolaan kredensial pengguna.

Kata kunci: Google Authentication, Human Error, Keamanan Website, SQL Injection, Sistem Login

Abstract

The development of modern website technology has encouraged the use of login systems integrated with databases to manage user data effectively and efficiently. Conventional authentication systems that use usernames and passwords are still widely implemented because they are easy to develop. However, these methods have several weaknesses, such as the risk of SQL Injection attacks, account data leaks, and human error when users create or manage accounts, including weak password usage and forgotten login credentials. In addition, manual registration processes are often considered impractical because users must fill in data and perform separate verification steps.

These problems indicate that traditional authentication systems still have security vulnerabilities and lack user convenience. Weaknesses in database management can be exploited by unauthorized parties to gain illegal access, while user errors in account creation can reduce system security effectiveness. Therefore, a more secure, practical, and efficient authentication solution is needed to improve user experience while minimizing security risks.

The purpose of this research is to implement Google Authentication (Google Auth) as an alternative modern login system. By utilizing Google authentication services, the login process can be performed more quickly and conveniently without requiring users to manually create new accounts. In addition, the use of Google Auth improves system security through standardized authentication mechanisms, thereby reducing the risks of SQL Injection, account misuse, and human error in user credential management.

Keywords: Google Authentication, Human Error, Login System, SQL Injection, Website Security

I. PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi telah menjadikan website sebagai media utama dalam berbagai layanan digital, seperti pendidikan, bisnis, dan pemerintahan. Dalam implementasinya, sebagian besar website modern menggunakan sistem login yang terhubung dengan database untuk melakukan autentikasi pengguna dan pengelolaan data akun. Sistem autentikasi berbasis username dan password masih menjadi metode yang paling umum digunakan karena mudah diterapkan dan mendukung pengelolaan data pengguna secara terstruktur. Namun, semakin meningkatnya penggunaan aplikasi berbasis web juga meningkatkan risiko ancaman keamanan terhadap data pengguna dan sistem database[1][2].

Salah satu ancaman keamanan yang paling sering terjadi pada aplikasi web adalah SQL Injection. Serangan ini memanfaatkan kelemahan validasi input pada sistem sehingga penyerang dapat menyisipkan perintah SQL berbahaya ke dalam query database. Dampak dari serangan tersebut dapat berupa pencurian data pengguna, manipulasi database, hingga pengambilalihan akses sistem secara ilegal. Halfond, Viegas, dan Orso dalam penelitiannya menjelaskan bahwa SQL Injection merupakan salah satu ancaman terbesar pada aplikasi web karena banyak sistem masih belum menerapkan mekanisme validasi dan sanitasi input secara optimal[1].

Selain permasalahan keamanan, sistem login konvensional juga memiliki kelemahan dari sisi pengguna. Proses registrasi akun manual sering menyebabkan human error, seperti penggunaan password yang lemah, pengulangan password pada berbagai layanan, maupun lupa terhadap kredensial akun. Penelitian Bonneau menunjukkan bahwa autentikasi berbasis password memiliki berbagai kelemahan pada aspek keamanan dan kenyamanan pengguna sehingga diperlukan metode autentikasi yang lebih praktis dan aman untuk mendukung penggunaan layanan digital modern[3].

Salah satu solusi yang dapat diterapkan untuk mengatasi permasalahan tersebut adalah penggunaan Google Authentication berbasis OAuth 2.0 dan Single Sign-On (SSO). Teknologi ini memungkinkan pengguna melakukan login menggunakan akun Google tanpa perlu membuat akun baru secara manual pada website. Dengan mekanisme autentikasi terstandarisasi, penggunaan Google Authentication dapat membantu meningkatkan keamanan sistem, mengurangi risiko penyalahgunaan password, meminimalkan human error, serta memberikan proses login yang lebih praktis dan efisien bagi pengguna website modern.

II. METODE PENELITIAN

Bagian ini menjelaskan pendekatan dan tahapan yang digunakan dalam pengembangan sistem login website menggunakan Google Authentication dan cookies sebagai mekanisme autentikasi pengguna. Metode penelitian ini dirancang untuk mengatasi permasalahan keamanan login konvensional seperti SQL Injection, human error dalam pembuatan akun, serta meningkatkan kepraktisan pengguna dalam mengakses website modern. Pendekatan yang digunakan berfokus pada implementasi autentikasi berbasis OAuth 2.0 dengan dukungan session cookies untuk menjaga keamanan sesi pengguna.

2.1 Pendekatan Google Authentication

Penelitian ini menggunakan pendekatan autentikasi modern berbasis Google Authentication atau Google Auth. Google Authentication merupakan metode login yang memanfaatkan akun Google pengguna melalui protokol OAuth 2.0 dan OpenID Connect. Dengan metode ini, pengguna tidak perlu membuat akun baru secara manual pada website karena proses autentikasi dilakukan langsung melalui layanan Google. Pendekatan ini dipilih karena mampu meningkatkan keamanan sistem login dengan mengurangi penyimpanan password pengguna di database aplikasi. Selain itu, Google Authentication juga memberikan kemudahan dan efisiensi dalam proses login karena pengguna hanya perlu menggunakan akun Google yang telah dimiliki sebelumnya.

2.2 Implementasi Session Cookies

Dalam sistem ini, cookies digunakan sebagai mekanisme penyimpanan sesi autentikasi pengguna setelah proses login berhasil dilakukan. Setelah pengguna berhasil melakukan autentikasi menggunakan akun Google, server akan membuat session token yang disimpan dalam bentuk HTTP Cookies. Cookies berfungsi untuk mempertahankan status login pengguna selama pengguna masih aktif menggunakan website. Dengan penggunaan session cookies, pengguna tidak perlu melakukan login berulang kali setiap berpindah halaman. Selain meningkatkan kenyamanan pengguna, penggunaan cookies

juga membantu menjaga keamanan autentikasi dengan membatasi akses hanya kepada pengguna yang memiliki token sesi valid.

2.3 Metode Prototype

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah metode Prototype. Metode ini dipilih karena memungkinkan proses pengembangan sistem dilakukan secara bertahap melalui pembuatan model awal sistem autentikasi yang dapat diuji langsung sebelum dikembangkan menjadi sistem final.

Tahapan metode prototype dalam penelitian ini meliputi:

1. Analisis Kebutuhan
Tahap ini dilakukan untuk mengidentifikasi kebutuhan sistem login website modern, termasuk kebutuhan keamanan autentikasi, pengelolaan sesi pengguna, dan integrasi Google Authentication.
2. Perancangan Prototype
Pada tahap ini dilakukan perancangan alur autentikasi login menggunakan Google Auth serta mekanisme penyimpanan session menggunakan cookies.
3. Implementasi Sistem
Sistem mulai dikembangkan menggunakan integrasi Google OAuth dan pengelolaan session cookies pada website.
4. Pengujian Sistem
Pengujian dilakukan untuk memastikan proses login berjalan dengan baik, session cookies dapat digunakan dengan aman, serta mengurangi risiko keamanan seperti SQL Injection dan human error.
5. Evaluasi dan Perbaikan
Setelah pengujian dilakukan, sistem dievaluasi berdasarkan keamanan, kepraktisan, dan kenyamanan pengguna, kemudian dilakukan perbaikan apabila ditemukan kekurangan pada sistem.

2.4 Alur Sistem Autentikasi

Sistem autentikasi dimulai ketika pengguna memilih tombol login menggunakan Google. Website kemudian mengarahkan pengguna ke halaman autentikasi Google untuk melakukan verifikasi akun. Setelah proses autentikasi berhasil, Google akan mengirimkan token autentikasi kepada server website. Server kemudian melakukan validasi token dan membuat session pengguna dalam bentuk cookies. Session tersebut digunakan untuk mempertahankan status login pengguna selama mengakses website. Jika session cookies sudah tidak valid atau pengguna melakukan logout, maka sistem akan menghapus session sehingga pengguna harus melakukan login kembali.

2.5 Keamanan Sistem

Penerapan Google Authentication dan cookies dalam penelitian ini bertujuan untuk meningkatkan keamanan autentikasi website. Dengan penggunaan OAuth 2.0, sistem tidak perlu menyimpan password pengguna secara langsung pada database sehingga dapat mengurangi risiko kebocoran data akun. Selain itu, penggunaan session cookies membantu mengelola autentikasi pengguna secara lebih aman dibandingkan sistem login konvensional. Pendekatan ini juga membantu meminimalkan human error dalam proses registrasi akun dan penggunaan password yang lemah.

III. HASIL DAN PEMBAHASAN

Pada bagian ini menjelaskan hasil implementasi sistem login website menggunakan Google Authentication dan cookies sebagai mekanisme autentikasi pengguna. Sistem yang dikembangkan bertujuan untuk meningkatkan keamanan login website sekaligus memberikan kemudahan dan kepraktisan bagi pengguna dalam proses autentikasi. Pengujian dilakukan terhadap proses login, pengelolaan session cookies, serta alur autentikasi pengguna pada website.

3.1 Implementasi Sistem Login Google Authentication

Sistem login diimplementasikan menggunakan layanan Google Authentication berbasis OAuth 2.0. Pengguna dapat melakukan login menggunakan akun Google tanpa perlu membuat akun baru secara manual pada website. Ketika tombol login ditekan, pengguna akan diarahkan menuju halaman autentikasi Google untuk melakukan verifikasi akun.

Setelah proses autentikasi berhasil, sistem menerima data pengguna berupa nama, email, dan token autentikasi dari Google. Data tersebut kemudian digunakan untuk membuat sesi login pengguna pada server website menggunakan cookies. Implementasi ini membantu meningkatkan keamanan sistem karena password pengguna tidak disimpan secara langsung pada database website.

3.2 Data Flow Diagram (DFD) Sistem

3.2.1 DFD Level 0

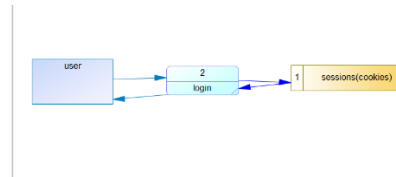
DFD Level 0 menggambarkan alur utama sistem autentikasi website menggunakan Google Authentication dan session cookies secara umum. Pada diagram ini terdapat tiga komponen utama, yaitu *user* sebagai entitas eksternal, proses *login* sebagai proses utama sistem, dan *sessions (cookies)* sebagai media penyimpanan sesi pengguna.

Pada proses ini, user melakukan login ke dalam website melalui proses login. Sistem kemudian membuat dan memvalidasi session pengguna menggunakan cookies yang tersimpan pada browser. Session cookies tersebut digunakan untuk mempertahankan status login pengguna selama masih aktif menggunakan website.

Keterangan alur pada DFD Level 0:

1. User mengirimkan permintaan login ke sistem.
2. Sistem memproses autentikasi login pengguna.
3. Sistem membuat dan menyimpan session login ke dalam sessions (cookies).
4. Cookies digunakan kembali untuk memvalidasi sesi pengguna.
5. Sistem memberikan akses kepada user setelah sesi dinyatakan valid.

DFD Level 0 ini menunjukkan hubungan umum antara pengguna, proses login, dan penyimpanan session cookies pada sistem autentikasi website.



Gambar 1. DFD Level 0 Sistem Login Google Authentication

3.2.2 DFD Level 1

DFD Level 1 menjelaskan proses autentikasi login secara lebih rinci dibandingkan DFD Level 0. Pada diagram ini terdapat beberapa proses utama yaitu *login request*, *user verification*, *auth callback*, *generate sessions*, dan *sending cookies*. Selain itu terdapat dua data store yaitu *users database* dan *sessions (cookies)*, serta entitas eksternal berupa *user* dan *Google Auth*.

Proses dimulai ketika user melakukan *login request* pada website menggunakan akun Google. Sistem kemudian meneruskan permintaan autentikasi ke layanan Google Auth untuk melakukan verifikasi identitas pengguna.

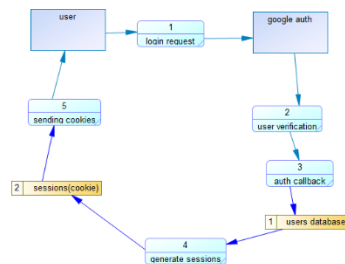
Setelah pengguna berhasil melakukan autentikasi, Google Auth mengirimkan data autentikasi kembali ke sistem melalui proses *auth callback*. Data pengguna seperti email, nama, dan token autentikasi kemudian diperiksa dan disimpan ke dalam *users database* apabila pengguna baru pertama kali login.

Selanjutnya sistem melakukan proses *generate sessions* untuk membuat session login pengguna. Session tersebut kemudian disimpan pada *sessions (cookies)* dan dikirim kembali ke browser pengguna melalui proses *sending cookies*. Cookies ini digunakan untuk mempertahankan status login pengguna selama sesi masih aktif.

Keterangan alur pada DFD Level 1:

1. User melakukan login request ke website.
2. Sistem mengirimkan permintaan autentikasi ke Google Auth.
3. Google Auth melakukan verifikasi akun pengguna.
4. Google mengirimkan auth callback ke sistem.
5. Sistem memeriksa dan menyimpan data pengguna ke users database.
6. Sistem membuat session login pengguna.
7. Session disimpan ke sessions (cookies).

8. Cookies dikirim kembali ke browser user.
 9. User berhasil mengakses website dalam keadaan login.
- DFD Level 1 ini menunjukkan proses autentikasi Google secara lebih detail mulai dari login pengguna hingga pengelolaan session cookies pada website.



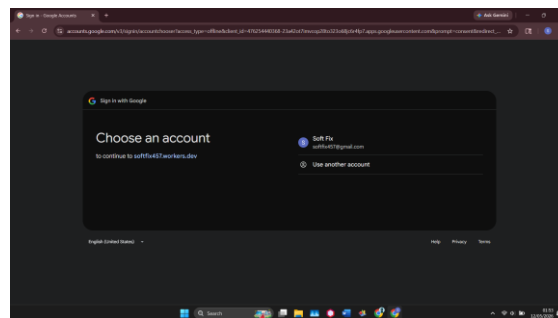
Gambar 2. DFD Level 1 Sistem Login Google Authentication

3.3 Pengujian Login Menggunakan Google Authentication

Pengujian pertama dilakukan pada proses login menggunakan akun Google. Sistem berhasil mengarahkan pengguna menuju halaman autentikasi Google dan mengembalikan data akun pengguna setelah proses login berhasil dilakukan.

Hasil pengujian menunjukkan bahwa proses autentikasi berjalan dengan baik tanpa perlu melakukan registrasi akun secara manual. Pengguna hanya perlu memilih akun Google yang ingin digunakan untuk login ke website.

Berdasarkan hasil pengujian tersebut, implementasi Google Authentication mampu meningkatkan kepraktisan sistem login karena pengguna tidak perlu mengingat username dan password tambahan untuk mengakses website.



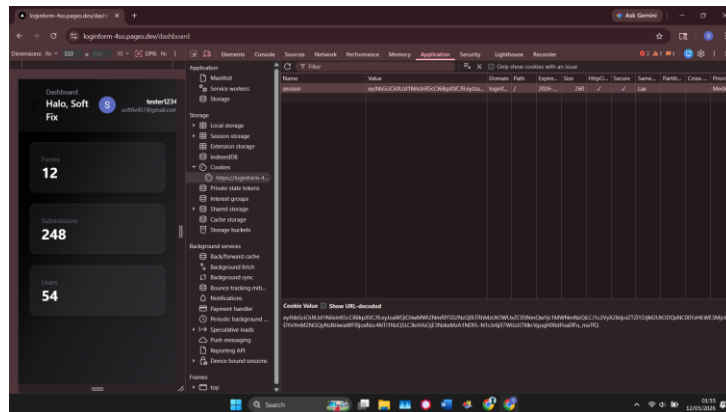
Gambar 3. Pengujian Login Menggunakan Google Auth

3.4 Pengujian Session Cookies

Pengujian kedua dilakukan terhadap mekanisme session cookies yang digunakan untuk mempertahankan status login pengguna. Setelah pengguna berhasil login, sistem membuat cookies session yang tersimpan pada browser pengguna.

Hasil pengujian menunjukkan bahwa pengguna tetap dapat mengakses halaman website tanpa perlu melakukan login ulang selama cookies session masih aktif dan valid. Ketika pengguna melakukan logout, session cookies berhasil dihapus sehingga akses ke halaman website dibatasi kembali.

Penggunaan cookies dalam sistem ini membantu meningkatkan kenyamanan pengguna sekaligus menjaga keamanan sesi autentikasi pengguna pada website.



Gambar 4. Pengujian Cookies Setelah Login

3.5 Pembahasan

Berdasarkan hasil implementasi dan pengujian yang telah dilakukan, sistem login menggunakan Google Authentication dan cookies berhasil memberikan solusi autentikasi yang lebih aman dan praktis dibandingkan sistem login konvensional. Integrasi OAuth 2.0 memungkinkan pengguna melakukan login menggunakan akun Google tanpa perlu membuat akun baru pada website.

Penggunaan session cookies membantu mempertahankan sesi login pengguna secara efisien sehingga meningkatkan kenyamanan saat mengakses website. Dari sisi keamanan, sistem ini mampu mengurangi risiko SQL Injection karena proses autentikasi tidak bergantung pada query login manual berbasis username dan password.

Selain meningkatkan keamanan, implementasi Google Authentication juga membantu meminimalkan human error dalam pengelolaan akun pengguna. Dengan demikian, sistem autentikasi modern ini dapat menjadi solusi yang efektif untuk diterapkan pada website modern yang membutuhkan keamanan dan kemudahan akses pengguna.

IV. KESIMPULAN

Berdasarkan hasil implementasi dan pengujian sistem login website menggunakan Google Authentication dan session cookies, dapat disimpulkan bahwa sistem autentikasi modern ini mampu memberikan solusi yang lebih aman dan praktis dibandingkan metode login konvensional. Integrasi Google Authentication berbasis OAuth 2.0 memungkinkan pengguna melakukan login menggunakan akun Google tanpa perlu membuat akun baru maupun mengingat password tambahan pada website. Penerapan session cookies juga berhasil mempertahankan status login pengguna secara efisien sehingga meningkatkan kenyamanan dalam mengakses website. Mekanisme ini memungkinkan pengguna tetap berada dalam keadaan login selama session masih aktif tanpa harus melakukan autentikasi ulang pada setiap perpindahan halaman.

Dari sisi keamanan, penggunaan Google Authentication membantu mengurangi risiko penyalahgunaan password dan meminimalkan human error dalam proses registrasi maupun login pengguna. Selain itu, penggunaan prepared statement pada query database membantu mengurangi potensi serangan SQL Injection karena input pengguna tidak langsung digabungkan ke dalam query SQL. Berdasarkan hasil tersebut, sistem login menggunakan Google Authentication dan cookies dapat menjadi solusi autentikasi yang efektif untuk website modern karena mampu meningkatkan keamanan, efisiensi, serta kepraktisan dalam proses login pengguna.

V. UCAPAN TERIMA KASIH

Penulis mengucapkan puji dan syukur kepada Tuhan Yang Maha Esa karena atas rahmat dan karunia-Nya penelitian dan penulisan karya ilmiah ini dapat diselesaikan dengan baik. Penulis juga menyampaikan terima kasih kepada semua pihak yang telah memberikan dukungan, bantuan, serta bimbingan selama proses penyusunan penelitian ini.

Ucapan terima kasih disampaikan kepada dosen pembimbing yang telah memberikan arahan, masukan, dan motivasi dalam proses penelitian dan penulisan karya ilmiah ini. Selain itu, penulis juga

berterima kasih kepada keluarga dan teman-teman yang selalu memberikan dukungan moral maupun semangat selama proses pengerjaan penelitian.

Penulis juga mengucapkan terima kasih kepada seluruh pihak yang telah membantu dalam pengembangan sistem login berbasis Google Authentication ini, baik secara langsung maupun tidak langsung. Semoga penelitian ini dapat memberikan manfaat serta menjadi referensi bagi pengembangan sistem autentikasi website yang lebih aman dan praktis di masa mendatang.

REFERENSI

- [1] W. G. J. Halfond, J. Viegas, and A. Orso, "A Classification of SQL Injection Attacks and Countermeasures," 2006.
- [2] J. P. Singh, "Analysis of SQL Injection Detection Techniques." [Online]. Available: <http://exploitable-web.com/link.php?id=1>
- [3] J. Bonneau, C. Herley, P. C. Van Oorschot, and F. Stajano, "The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes *."