**JETI**
*Jurnal Elektronika dan Teknologi Informasi*
**Vol. 5, No. 2, September 2024**
**e-ISSN: 2721-9380**

# Implementasi Single Sign On (SSO) Menggunakan Protokol OAuth Pada Sistem Informasi Kampus

**Anis Raysa[1], Imam Muslem[2], Sriwinar[3]**
[1,2,3] Program Studi Informatika,Fakultas Ilmu Komputer, Universitas Almuslim

*Coresponding Email: itsanisraysa@gmail.com

**ABSTRAK**

Single Sign-On (SSO) merupakan solusi autentikasi terpusat yang memungkinkan pengguna mengakses beberapa aplikasi hanya dengan satu kali login. Studi ini mengimplementasikan SSO pada sistem informasi kampus menggunakan protokol OAuth 2.0, yang memungkinkan otorisasi aman tanpa berbagi kredensial secara langsung. OAuth digunakan untuk menghasilkan token akses yang valid bagi aplikasi terintegrasi melalui akun Google pengguna. Implementasi dilakukan dengan mengintegrasikan OAuth melalui Google Cloud Platform, membangun sistem backend menggunakan PHP dan MySQL, serta mendesain antarmuka dengan Figma. Sistem diuji melalui skenario login lintas pengguna (admin, dosen, mahasiswa, staf, dan pengguna otomatis), serta integrasi ke berbagai aplikasi kampus pihak ketiga. Hasil menunjukkan bahwa sistem mampu menyederhanakan proses autentikasi, mengurangi kebutuhan login berulang, dan meningkatkan efisiensi akses. Penerapan ini juga memperkuat aspek keamanan dan kenyamanan pengguna melalui mekanisme otorisasi token berbasis OAuth. SSO berbasis OAuth terbukti efektif diterapkan sebagai solusi otentikasi terintegrasi dalam lingkungan sistem informasi kampus.

**Kata Kunci:** *Single Sign-on (sso), OAuth 2.0, Sistem Informasi Kampus, Autentikasi Terpusat, Token Akses, Google API*

**ABSTRACT**

*Single Sign-On (SSO) is a centralized authentication solution that enables users to access multiple applications with a single login. This study implements SSO in a campus information system using the OAuth 2.0 protocol, which allows secure authorization without directly sharing user credentials. OAuth is used to generate valid access tokens for integrated applications via users' Google accounts. The implementation was carried out by integrating OAuth through Google Cloud Platform, developing a backend system using PHP and MySQL, and designing the interface with Figma. The system was tested through login scenarios involving various user types (admin, lecturers, students, staff, and automatic registrants), and integrated with several third-party campus applications. The results show that the system simplifies the authentication process, reduces repetitive login requirements, and improves access efficiency. This implementation also enhances security and user convenience through a token-based authorization mechanism. OAuth-based SSO has proven to be an effective solution for integrated authentication within academic information systems..*

***Keywords:*** *Single Sign-On (SSO), OAuth 2.0, Campus Information System, Centralized Authentication, Access Token, Google API*

## 1. PENDAHULUAN

Perkembangan teknologi internet yang sangat pesat telah mendorong terciptanya berbagai inovasi dalam bidang sistem jaringan, keamanan data, dan integrasi layanan digital. Salah satu tantangan yang sering muncul dalam pengelolaan sistem berbasis web adalah kebutuhan untuk melakukan autentikasi berulang kali ketika mengakses berbagai platform yang berbeda. Hal ini tidak hanya menyulitkan pengguna, tetapi juga berpotensi menurunkan efisiensi dan kenyamanan penggunaan sistem.

Untuk mengatasi tantangan tersebut, konsep Single Sign-On (SSO) menjadi salah satu solusi yang banyak digunakan. Teknologi SSO memungkinkan pengguna untuk mengakses berbagai layanan dan aplikasi dalam satu jaringan hanya dengan satu kali proses autentikasi [1]. Teknologi ini sangat efektif diterapkan pada lingkungan jaringan yang besar dan heterogen, seperti di institusi pendidikan tinggi, di mana terdapat banyak sistem informasi akademik, keuangan, layanan mahasiswa, dan lainnya yang saling terpisah.

Dalam implementasinya, SSO biasanya memanfaatkan protokol otorisasi modern seperti OAuth 2.0.

OAuth merupakan protokol otorisasi terdelegasi (delegated authorization protocol) yang memungkinkan aplikasi pihak ketiga mengakses data pengguna tanpa memerlukan kredensial secara langsung [2]. Protokol ini bekerja dengan memberikan *access token* melalui alur otorisasi yang melibatkan persetujuan eksplisit dari pengguna. Dengan sistem ini, keamanan data pengguna lebih terjaga karena kredensial utama tidak perlu dibagikan ke banyak layanan.

OAuth 2.0 dirancang dengan pendekatan *three-legged authorization*, yang melibatkan tiga komponen utama: penyedia layanan (service provider), pengguna (user), dan aplikasi pihak ketiga (client) [3]. Setiap komponen memiliki peran penting dalam proses otorisasi berbasis token. Keunggulan OAuth terletak pada fleksibilitasnya, skalabilitasnya, serta kemampuannya dalam diintegrasikan ke berbagai platform, baik berbasis web, mobile, maupun desktop.

Penerapan OAuth dalam sistem informasi kampus diharapkan dapat meningkatkan efisiensi operasional dan kenyamanan pengguna, khususnya dalam proses login ke berbagai aplikasi yang digunakan sivitas akademika. Dengan adanya sistem SSO berbasis OAuth, pengguna cukup melakukan login satu kali menggunakan akun yang sudah mereka miliki (misalnya akun Google), dan langsung mendapatkan akses ke berbagai sistem terintegrasi kampus seperti sistem akademik, pembayaran, atau e-learning.

Berdasarkan latar belakang tersebut, artikel ini membahas implementasi Single Sign-On (SSO) menggunakan protokol OAuth 2.0 pada Sistem Informasi Kampus. Tujuannya adalah untuk merancang dan mengembangkan sistem login yang terintegrasi, aman, dan efisien bagi pengguna kampus, sekaligus mengurangi kompleksitas dalam pengelolaan akun dan meningkatkan pengalaman pengguna secara keseluruhan.

## 2. KAJIAN TEORITIS

### *2.1 Single Sign-On (SSO)*

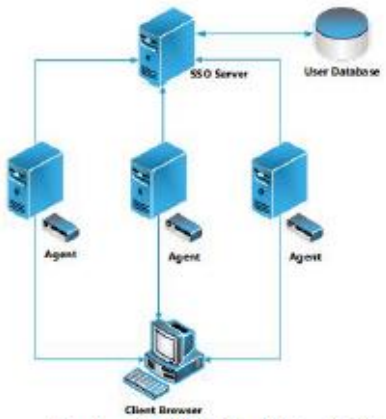
Single Sign-On (SSO) merupakan mekanisme autentikasi yang memungkinkan pengguna untuk mengakses berbagai aplikasi dalam satu sistem hanya dengan satu kali proses login [1]. Hal ini mempermudah pengguna, mengurangi kebutuhan login ulang ke aplikasi yang berbeda, serta meningkatkan efisiensi dan kenyamanan dalam penggunaan sistem informasi.

Menurut Menurut Djayusman (2023) [2], SSO sangat berguna pada sistem berskala besar dan heterogen. Satu kali autentikasi memberikan akses ke berbagai layanan yang saling terintegrasi, tanpa perlu login berulang-ulang. Selain itu, menurut Hursti (1997) [3], SSO juga memberikan keuntungan dalam hal manajemen akun dan peningkatan keamanan.

Keuntungan SSO:
1. Pengguna hanya login satu kali untuk banyak layanan.
2. Mengurangi Kelelahan akibat banyak password
3. Mendukung autentikasi konvensional.
4. Meningkatkan efisiensi dan produktivitas.
5. Menyediakan keamanan tambahan melalui token dan OTP.


Program Studi Informatika,Fakultas Ilmu Komputer, Universitas Almuslim
Email: itsanisraysa@gmail.com

### *2.2 Arsitektur SSO*




Gambar: 2.3 Arsitektur Sistem SSO

(Sumber: Springer-verlag Berlin Heidelberg, 2003)

**Sistem SSO memiliki dua komponen utama:**

1. **Agent**: Berfungsi di sisi web server untuk memproses permintaan pengguna dan meneruskannya ke sever otentikasi**.**
2. **SSO Server**: Bertugas mengelola sesi pengguna melalui cookie sementara yang berisi user-id, session-id, dan informasi lainnya [4].

### *2.3 OAuth 2.0*

OAuth 2.0 adalah protokol otorisasi yang memungkinkan aplikasi pihak ketiga mengakses data pengguna tanpa harus mengetahui username dan password pengguna [5]. Protokol ini berbasis token dan digunakan secara luas dalam sistem modern untuk mengamankan akses ke sumber daya pengguna.

**Komponen OAuth 2.0:**

1. **Authorization Server** – Bertugas melakukan autentikasi dan menerbitkan token.
2. **Client** – Aplikasi pihak ketiga yang meminta akses.
3. **Resource Owner** – Pemilik data (pengguna).
4. **Resource Server** – Menyediakan data yang diminta dan memverifikasi token.

**Keunggulan OAuth 2.0:**

1. Memberikan akses tanpa berbagi kredensial.
2. Token dapat dibatasi masa aktifnya.
3. Mendukung berbagai metode autentikasi.
4. Mudah diintegrasikan.

### *2.4 Sistem Informasi Kampus dan OAuth*

Sistem Informasi Kampus mengelola berbagai layanan seperti akademik, keuangan, dan administrasi. Dengan mengintegrasikan OAuth 2.0, pengguna hanya perlu login sekali untuk mengakses semua layanan seperti SIAKAD, LMS, dan perpustakaan digital [6].

### *2.5 Teknologi yang Digunakan*

1. Google Cloud Platform – Menyediakan layanan OAuth dan API [7].
2. PHP & MySQL – Untuk membangun aplikasi web kampus [8].
3.Website – Menjadi antarmuka akses utama pengguna [11].
4. Diagram Konteks & DFD – Menggambarkan alur data sistem [9].
5. ERD – Merancang basis data yang digunakan [10].

***2.6 Sistem Basis Data* sistem** adalah kesatuan elemen yang saling berinteraksi untuk mencapai tujuan tertentu. Basis data digunakan untuk menyimpan informasi pengguna dan log aktivitas sistem dengan aman dan efisien [12].

**2.7 Manfaat Penerapan SSO OAuth 2.0**

1. Akses mudah dan cepat.
2. Efisiensi dalam pengelolaan akun.
3. Keamanan meningkat.
4. Konsistensi data pengguna.
5. Sistem mudah diintegrasikan.

***3. Analisi dan Perancangan Sistem***

**3.1 Jabaran Masalah**

a. Masalah utama adalah: *"Bagaimana mengoptimalkan media SSO untuk mempermudah akses Sistem Informasi Kampus?"*. Solusi:

- Gunakan protokol standar seperti OAuth, SAML, OpenID atau Kerberos yang mendukung autentikasi lintas sistem [1][2].
- Integrasikan SSO dengan sistem kampus seperti SIM, KRS, PMB sehingga pengguna login sekali untuk semua layanan [3][4].
- Gunakan layanan SSO terpercaya seperti Satu Dikti dari Ditjen Diktiristek [5].

b. Masalah kedua adalah: *"Bagaimana menerapkan metode Protokol OAuth untuk membangun sistem SSO pada Sistem Informasi Kampus?"*. Langkah penerapan:

1. Pilih penyedia OAuth seperti Google, Facebook, GitHub [1][6].
2. Daftarkan aplikasi untuk memperoleh *Client ID* dan *Client Secret* [4].
3. Implementasikan alur otorisasi:
    1. Arahkan pengguna ke login penyedia OAuth
    2. Setelah login, pengguna menyetujui akses data
    3. Sistem menerima *authorization code*, menukarnya dengan *access token*
    4. Gunakan token untuk mengakses data pengguna [2][6]
4. Terapkan mekanisme SSO:
    1. Cek apakah pengguna memiliki sesi aktif
    2. Jika tidak, arahkan ke login OAuth
    3. Setelah otorisasi, simpan data pengguna dan token ke sesi lokal [2][7]

**3.2 Strategi Penyelesaian Masalah**

a. Rancang sistem sederhana dan mudah dipahami dengan mengikuti tahapan metode OAuth mulai dari analisis, implementasi, hingga pengujian [6].

b. Strategi implementasi:

- Pilih platform OAuth 2.0 yang sesuai (misal Google OAuth API)
- Rancang sistem aman dan efisien (dengan enkripsi dan token handling)
- Laksanakan tahapan implementasi mulai dari setup proyek, konfigurasi, integrasi, pengujian
- Sediakan dokumentasi pengguna berupa panduan atau FAQ [2][6][8]

**3.3 Deskripsi Sistem**

Sistem Single Sign-On (SSO) berbasis OAuth 2.0 memungkinkan pengguna login ke sistem kampus dengan akun media sosial (Google, Facebook). SSO menyederhanakan proses login, meningkatkan kenyamanan, efisiensi, dan keamanan karena tidak perlu menyimpan banyak password [1][6].

**3.4 Analisis Masalah Sistem Saat Ini**

- Proses login masih manual pada setiap aplikasi
- Membutuhkan banyak akun dan kredensial login
- Tidak terintegrasi antar sistem
- Risiko keamanan tinggi karena pengelolaan akun terpisah [3][9]

**3.5 Analisis Kebutuhan Sistem**

**a. Kebutuhan Data**

- Data user: ID, username, password, nama, email, role, profile_img

**b. Kebutuhan Output**

- Tampilan data user yang telah login
- Informasi akun pengguna dan aplikasinya

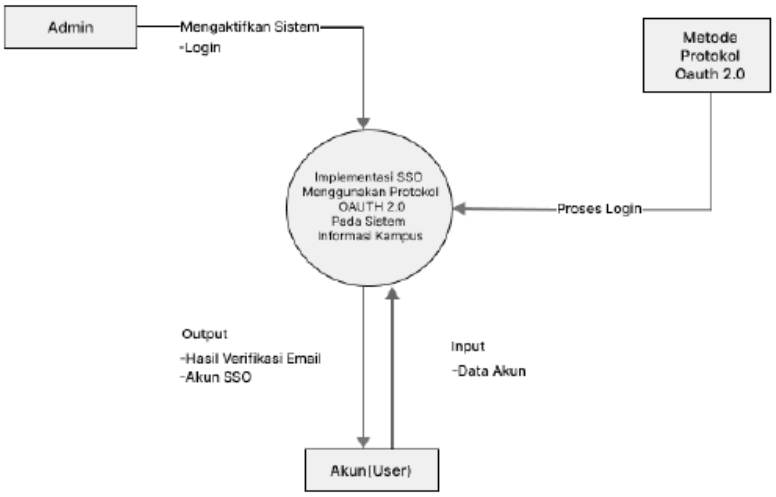
**3.6 Kebutuhan Perangkat Keras**

- Komputer: Intel Core 2.0 GHz, RAM 2 GB, HDD 250 GB
- Koneksi internet, printer, monitor, smartphone Android

**3.7 Kebutuhan Perangkat Lunak**

- Sistem Operasi: Windows 10 64-bit
- Web browser: Google Chrome
- Web server lokal: XAMPP
- Code editor: Visual Studio Code
- Bahasa pemrograman: PHP, HTML, CSS
- Desain antarmuka: Figma
- Database: MySQL [6][10]

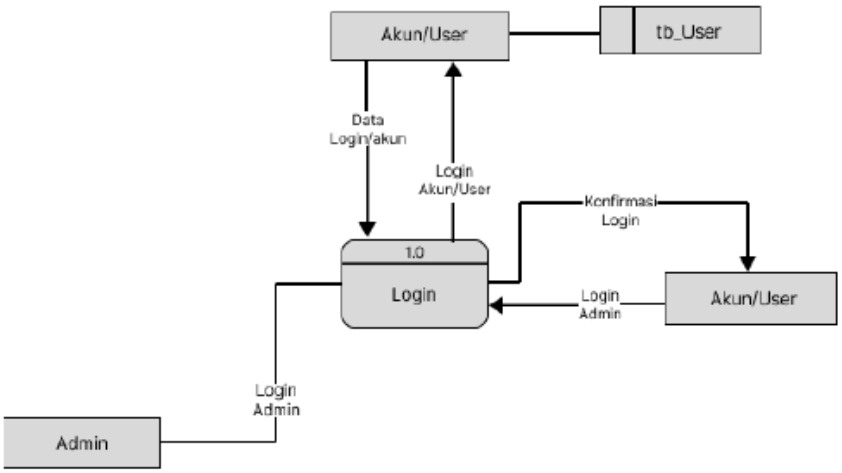
**3.8 Diagram Konteks** Diagram konteks menjelaskan aliran data antara pengguna, sistem, dan administrator. Diagram ini menunjukkan bagaimana data input dan output berinteraksi dalam sistem SSO (tidak ditampilkan dalam dokumen).




Gambar 3.1 Diagram Konteks Data Input dan Data Output

**3.9 Data Flow Diagram (DFD)** Menampilkan proses dan alur data utama dalam sistem SSO, seperti:

**e-ISSN: 2721-9380**




Gambar 3.2 *Data Flow Diagram*

- Proses: Login/logout, verifikasi login, pembuatan riwayat login
- Entitas: Admin, pengguna, sistem
- Aliran data: Akun pengguna, status login, token OAuth [2]

### 3.10 Desain Database

Database utama: `rayj6124_ssoo`

**Tabel user:**

Tabel 3.2 Tabel Login

| Kolom | Tipe | Size | Keterangan |
|---|---|---|---|
| id | Int | 11 | Id Login |
| username | varchar | 20 | username |
| Password | varchar | 50 | password |
| Nama | varchar | 200 | nama |
| Email | varchar | 255 | email |
| Npm | varchar | 100 | npm |
| Role | varchar | 255 | role (admin,dosen,mahasiswa,staf,none) |
| profile_img | varchar | 500 | Profile image |

- id (PK)
- username
- password
- nama

- email
- npm
- role (admin, dosen, mahasiswa, staf, none)
- profile_img

**Tabel applications:**

| Kolom | Tipe | Size | Keterangan |
|---|---|---|---|
| Id | Int | 11 | id |
| Name | Varchar | 255 | Nama aplikasi |
| Linkapp | Varchar | 500 | Link aplikasi |
| imgapp | Varchar | 500 | Gambar aplikasi |
| create_at | Timestamp | - | Waktu dibuat |

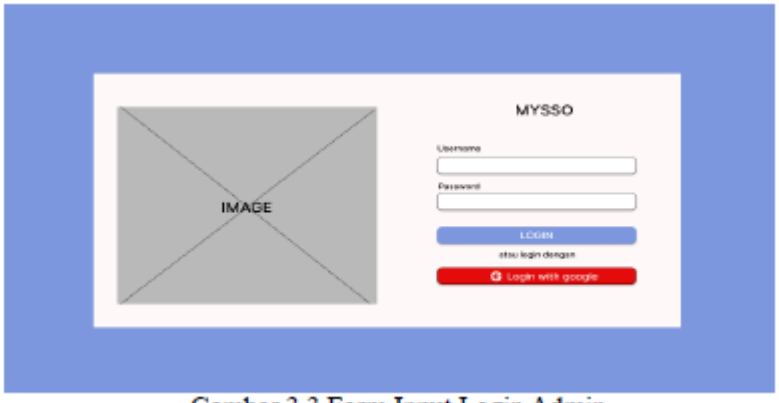
Tabel 3.3 Tabel Aplikasi

- id (PK)
- name
- linkapp
- imgapp
- created_at [6]

### 3.11 Perancangan Input dan Output

### a. Tampilan Admin

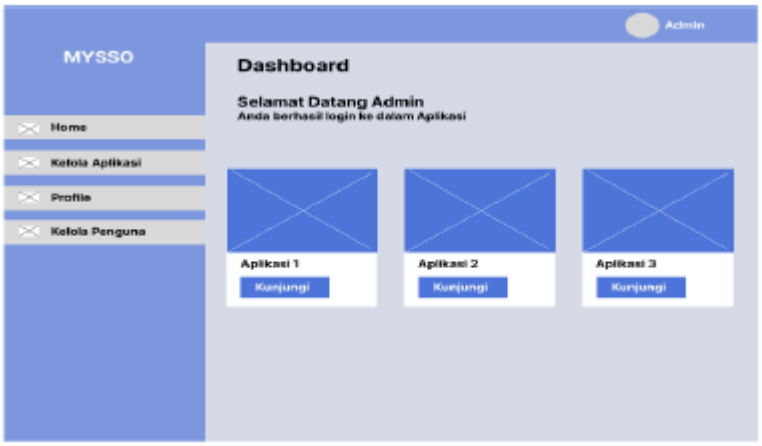
- Form login




Gambar 3.3 Form Input Login Admin

- Dashboard admin

**e-ISSN: 2721-9380**




Gambar 3.4 Form Dashboard Admin

- Kelola aplikasi


Gambar 3.5 Form Kelola Aplikasi

- Kelola pengguna

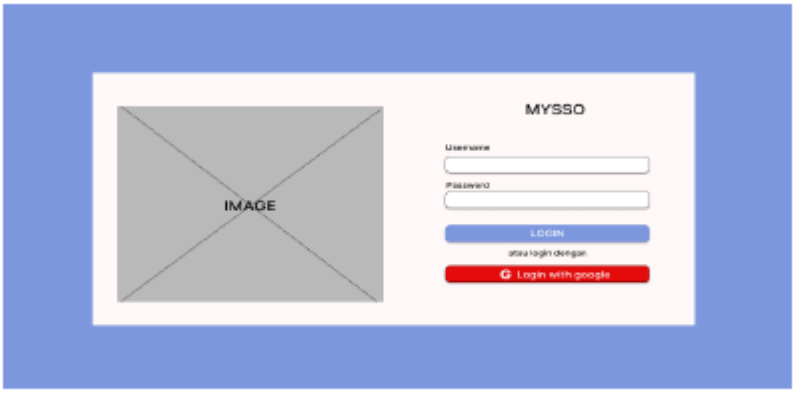

Gambar 3.7 Form Kelola Pengguna

- Profil admin


Gambar 3.6 Form Profile Admin

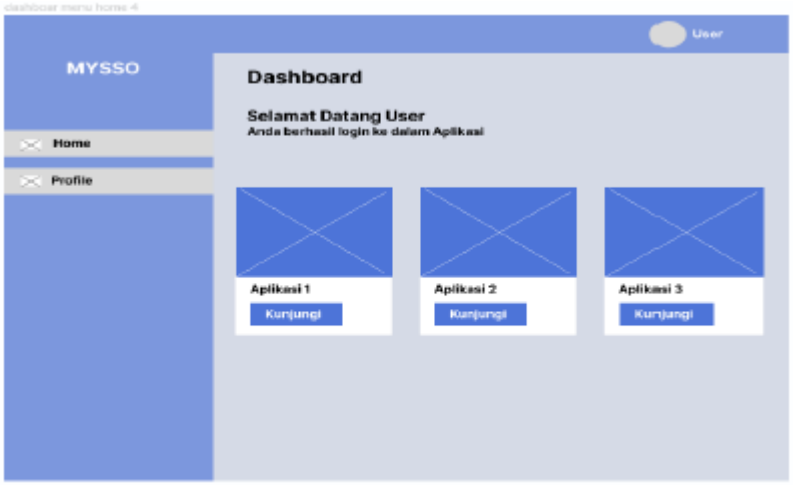
**b. Tampilan User (Dosen/Mahasiswa/Staf/None)**

- Form login



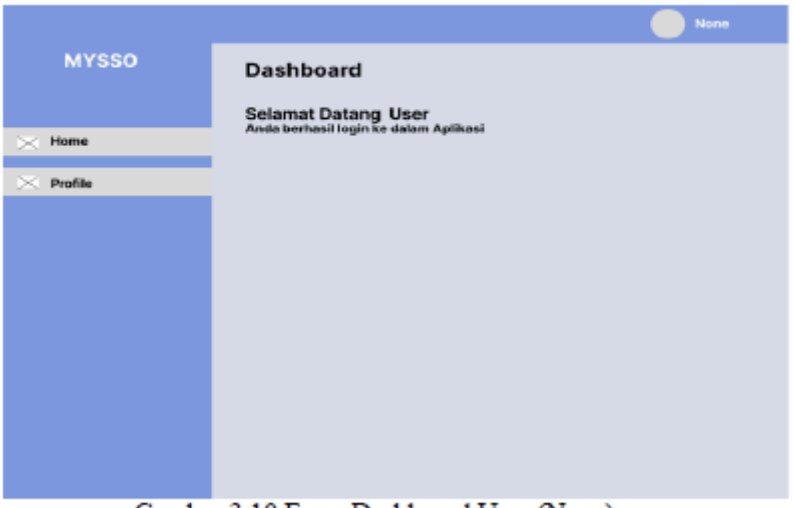

Gambar 3.8 Input Login User (Dosen/Mahasiswa/Staf/None)

- Dashboard user




Gambar 3.9 Form Dashboard User (Dosen/Mahasiswa/Staf)

- Profil pengguna




Gambar 3.10 Form Dashboard User (None)

***4. Implementasi Sistem***

Implementasi sistem ini adalah tahap penerapan dari rancangan sistem Single Sign-On (SSO) menggunakan protokol OAuth 2.0 pada Sistem Informasi Kampus. Tujuannya adalah untuk memudahkan pengguna mengakses berbagai aplikasi layanan kampus tanpa perlu login berulang. Implementasi melibatkan persiapan perangkat keras, perangkat lunak, dan integrasi sistem login berbasis OAuth dari Google [1][2].

**4.1 Perangkat Keras**

Implementasi dilakukan pada perangkat laptop dengan spesifikasi:

- Prosesor: AMD PRO A6-7350B
- RAM: 4 GB
- Penyimpanan: HDD 500 GB
- Koneksi internet

**4.1.2 Perangkat Lunak**

Perangkat lunak yang digunakan:

- Windows 10 x64
- Visual Studio Code
- XAMPP (Apache & MySQL)
- Google Chrome
- Google Cloud Console (untuk konfigurasi OAuth)
- Figma (desain antarmuka)
- Bahasa pemrograman: PHP, HTML, CSS [6][10]

**4.2 Proses Implementasi OAuth 2.0**

Langkah-langkah:

1. **Membuat Form Login**: Form login mengarahkan user ke halaman login Google. User yang terdaftar diverifikasi dari database.
2. **Konfigurasi Google API Console**: Membuat project, mengatur OAuth consent screen, dan mendapatkan Client ID serta Secret [6].
3. **Integrasi Kode**: Client ID dan Secret dimasukkan ke file konfigurasi `gpconfig.php`.
4. **Alur Otentikasi**:
   1. User diarahkan ke login Google
   2. Google mengirimkan authorization code
   3. Code ditukar dengan access token
   4. Data user diambil dan disimpan di sesi [2][6]
5. **Integrasi ke Sistem Kampus**: Token digunakan untuk mengakses aplikasi tanpa login ulang. Hal ini meningkatkan efisiensi pengguna [3][6].
6. **Antarmuka Sistem**:
   1. Halaman login
   2. Dashboard user & admin
   3. Pengelolaan aplikasi dan profil [6]
7. **Basis Data**:
   1. Tabel `user`: menyimpan akun pengguna
   2. Tabel `applications`: menyimpan data aplikasi pihak ketiga
8. **Pengujian**:
   1. User terdaftar dapat login menggunakan Google
   2. Login ulang tidak diperlukan ke aplikasi lain
   3. User yang tidak terdaftar ditolak [3][6]

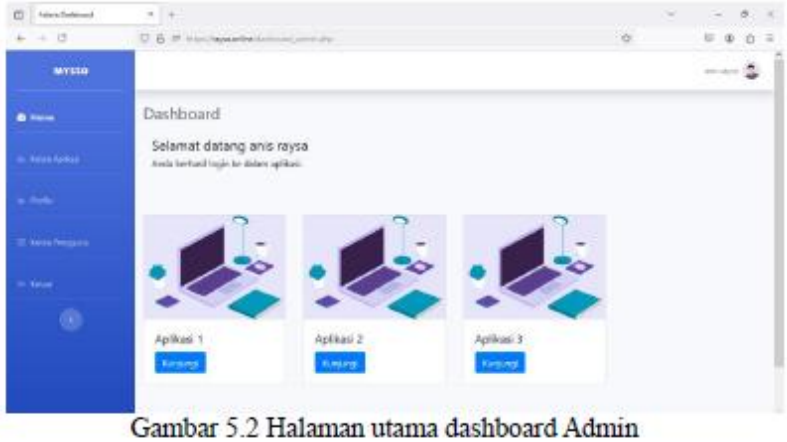
## *5. Pengujian dan Pembahasan*

**5.1 Pengujian Sistem**

Pengujian dilakukan untuk memastikan bahwa sistem Single Sign-On (SSO) dengan protokol OAuth 2.0 berjalan dengan baik dan sesuai kebutuhan pengguna pada Sistem Informasi Kampus. Tujuan dari pengujian ini adalah mengevaluasi efektivitas login tunggal, kemudahan penggunaan, dan keamanan akses [2][3].

**1. Halaman Login**

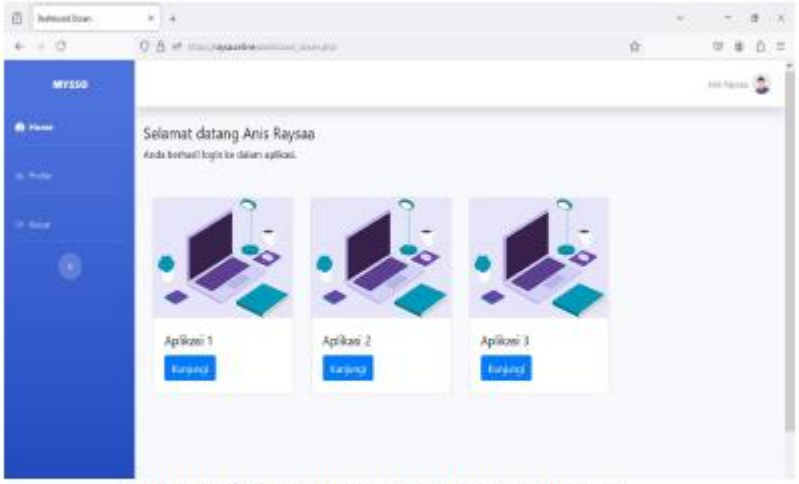
**JETI**
*Jurnal Elektronika dan Teknologi Informasi*
**Vol. 5, No. 2, September 2024**
**e-ISSN: 2721-9380**




Gambar 5.2 Halaman utama dashboard Admin

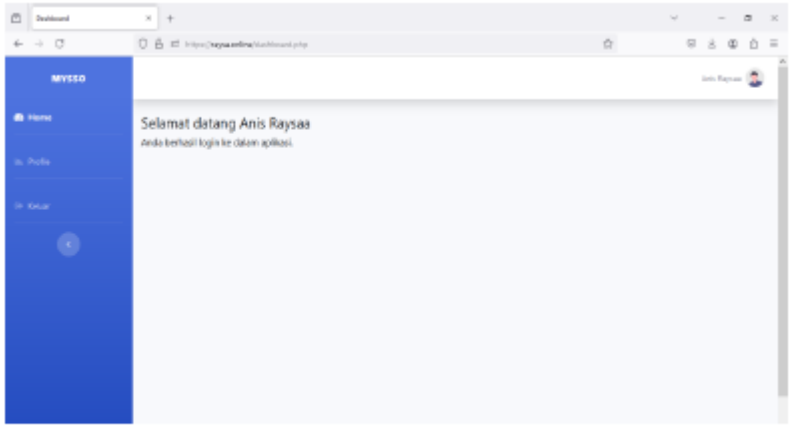
Halaman login digunakan oleh admin dan user (dosen, mahasiswa, staf, maupun user otomatis yang belum memiliki hak akses). Pengujian dilakukan dengan login melalui Google OAuth. Hasilnya, akun yang telah terdaftar di database dapat mengakses sistem, sedangkan akun yang tidak terdaftar akan ditolak. Ini menunjukkan sistem otentikasi bekerja sesuai harapan [1][6].

**2. Dashboard Pengguna**




Gambar 5.3 Halaman utama dashboard User
(Dosen/Mahasiswa/Staf)

**JETI** e-ISSN: 2721-9380
*Jurnal Elektronika dan Teknologi Informasi*
**Vol. 5, No. 2, September 2024**




Gambar 5.4 Halaman utama dashboard User (None)

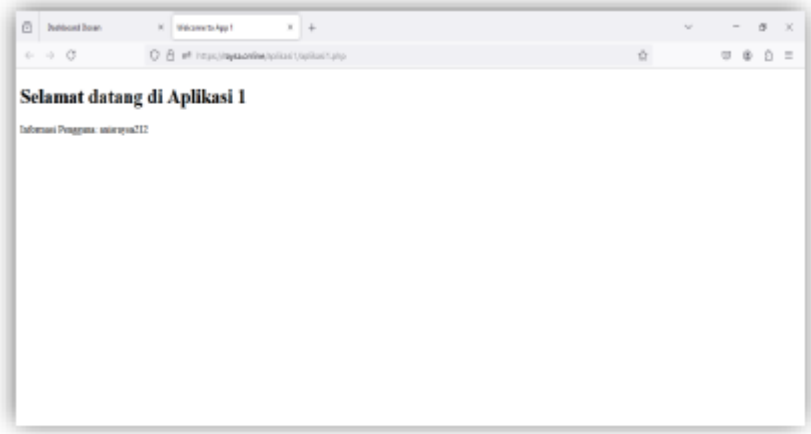
Setelah login berhasil, user diarahkan ke dashboard utama:

- **Admin**: dapat melihat dan mengelola aplikasi pihak ketiga dan data pengguna.
- **User (dosen/mahasiswa/staf)**: melihat aplikasi kampus dan profil pengguna.
- **User (None)**: hanya melihat profil karena belum diberi akses aplikasi.

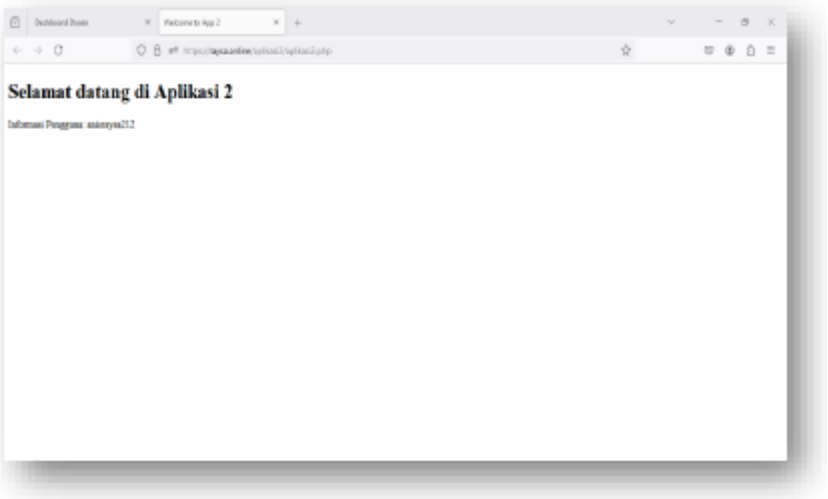
Dashboard memuat layanan sesuai peran pengguna, dengan antarmuka sederhana dan informatif [3][6].

**3. Akses Aplikasi Pihak Ketiga**

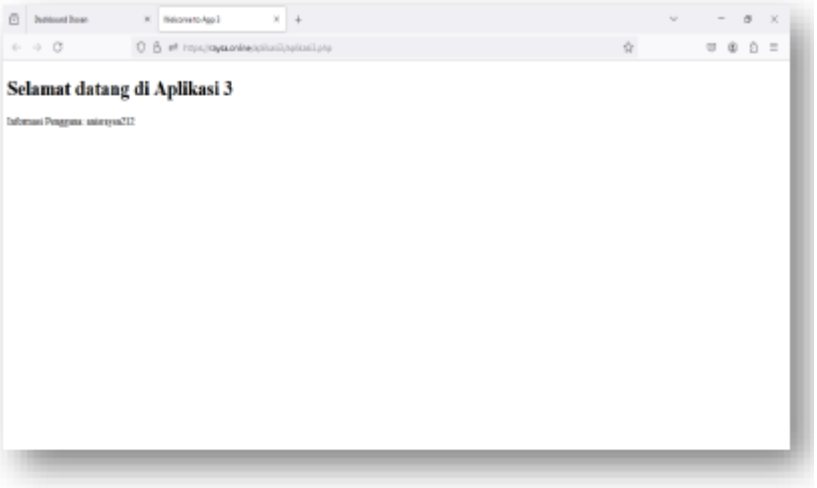
Pengujian dilakukan dengan mengakses beberapa aplikasi pihak ketiga dari dashboard. Hasilnya, pengguna tidak perlu login ulang ke setiap aplikasi karena sudah diotentikasi melalui token OAuth yang aktif. Ini membuktikan keberhasilan konsep SSO [2][4].




Gambar 5.5 Aplikasi satu



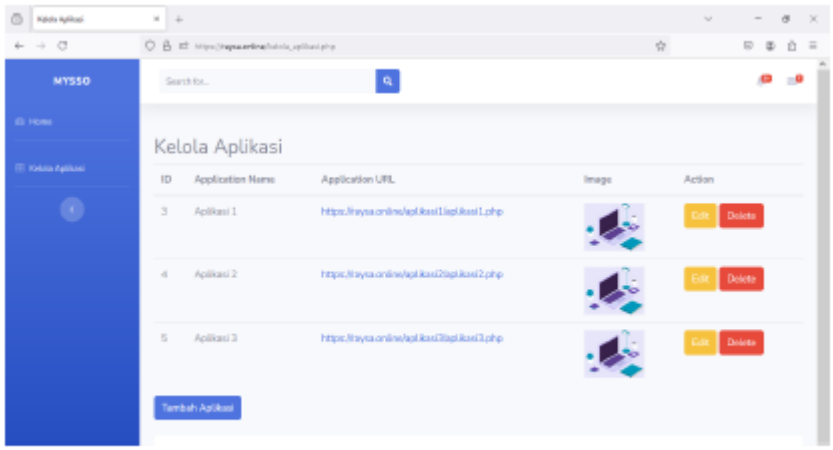

Gambar 5.6 Aplikasi dua




Gambar 5.7 Aplikasi tiga

**4. Halaman Kelola Aplikasi (Admin)**

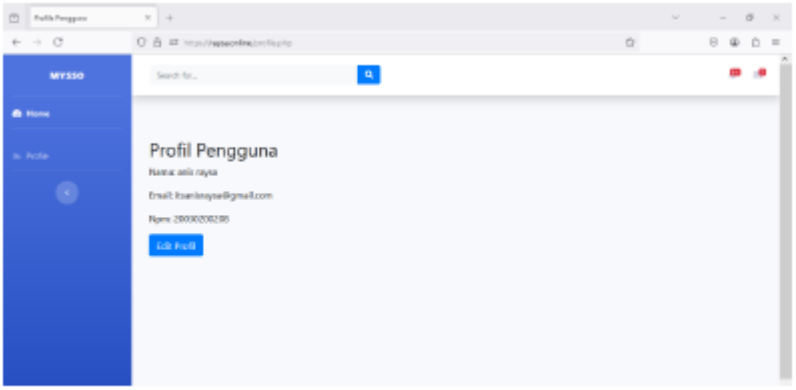
**JETI**
*Jurnal Elektronika dan Teknologi Informasi*
**Vol. 5, No. 2, September 2024**
**e-ISSN: 2721-9380**



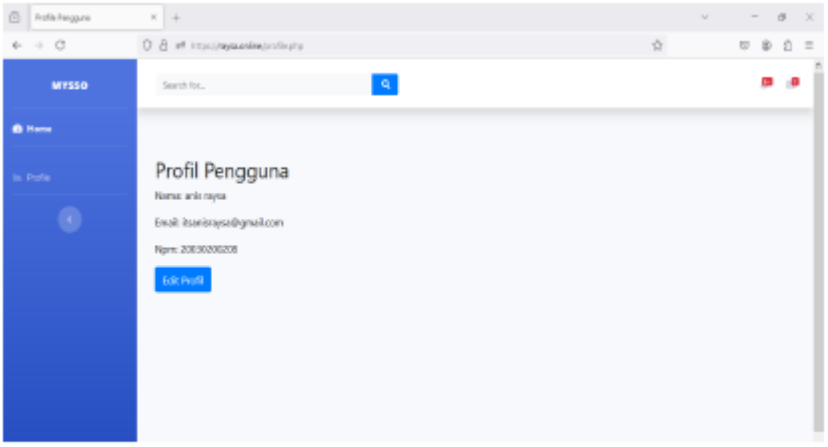

Gambar 5.8 Halaman kelola aplikasi

Admin dapat menambah, mengedit, dan menghapus aplikasi pihak ketiga yang terintegrasi. Halaman ini berfungsi dengan baik dan memastikan fleksibilitas dalam pengelolaan aplikasi [6].

**5. Halaman Profil**



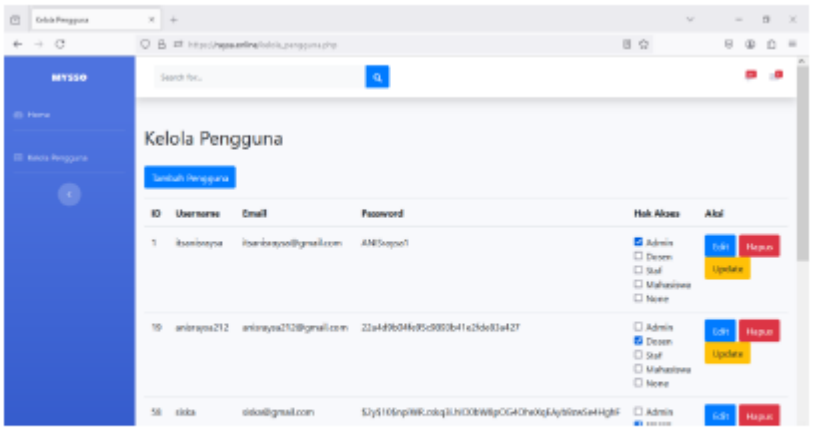

Gambar 5.9 Halaman Profile Admin




Gambar 5.10 Halaman Profile User (Dosen/Mahasiswa/Staf/None)

Halaman ini menampilkan informasi pengguna sesuai data yang ditarik dari Google API dan disimpan ke basis data lokal. Pengujian menunjukkan data profil tampil sesuai dan konsisten antara admin dan user [3][6].

**6. Halaman Kelola Pengguna (Admin)**




Gambar 5.11 Halaman Kelola Pengguna

Admin dapat melihat seluruh data user yang telah login ke sistem. Fungsi ini penting untuk memantau penggunaan sistem dan memberikan hak akses jika diperlukan [6].

## 3. SIMPULAN

Hasil pengujian dan implementasi menunjukkan bahwa penerapan Single Sign-On (SSO) menggunakan protokol OAuth 2.0 pada sistem informasi kampus berhasil meningkatkan efisiensi login, keamanan data, serta kenyamanan pengguna dalam mengakses berbagai layanan hanya dengan satu kali autentikasi, sehingga sistem ini dinilai efektif dan sesuai kebutuhan pengguna kampus.

## DAFTAR PUSTAKA

[1] Aini, Q., Rahardja, U., Naufal, R. S., Stmik, D., Jurusan, R., Informasi, S., Stmik, M., & Komputer, S. (2018). Penerapan Single Sign On dengan Google pada Website berbasis YII Framework. *Application Single Sign On with Google the Website Based on Yii Framework*, 8(1).

[2] Aminudin. (2014). Implementasi Single Sign On (SSO) untuk Mendukung Interaktivitas Aplikasi E-Commerce Menggunakan Protokol OAuth. *Jurnal Teknologi Informasi*, 10(1).

[3] Dinnarwaty Putri, T., Sugeng, W., & Katri, R. (2019). Sistem Otentikasi Login dengan Single Sign-On untuk Mengakses Banyak Sistem. *MIND Journal*, 2(2), 96–110. https://doi.org/10.26760/mindjournal

[4] Fatman, Y., & Octaviawati, R. (n.d.). Implementasi Metode Open Authorization (OAuth2) untuk Pengelolaan Data Dosen di Universitas Islam Nusantara. *Jurnal Informatika*, 2(1).

[5] Fikri, M. (2018). Perancangan Aplikasi Single Sign-On (SSO) Menggunakan Otentikasi Gambar. *Jurnal Teknologi Informasi & Komunikasi Digital Zone*, 9(1).

[6] Ghiffari, A. P. (2023). Implementasi Single Sign On (SSO) Menggunakan Representational State Transfer (REST) dan Open Authorization (OAuth 2.0): Studi Kasus Universitas Muhammadiyah Magelang.

[7] Marina, E. (2021). Implementasi Single Sign On pada Web Menggunakan Protokol OAuth Facebook. *Buletin Utama Teknik*, 16(3).

[8] Salmuasih, & Setiawan, M. A. (2023). Evaluasi Penerapan Single Sign-On SAML dan OAuth 2.0: Studi pada Perguruan Tinggi Yogyakarta. *JSiI (Jurnal Sistem Informasi)*, 10(1), 41–49. https://doi.org/10.30656/jsii.v10i1.6186

[9] Senapartha, I. K. D. (2021). Implementasi Single Sign-On Menggunakan Google Identity, REST, dan OAuth 2.0 Berbasis Scrum. *Jurnal Teknik Informatika dan Sistem Informasi*, 7(2). https://doi.org/10.28932/jutisi.v7i2.3437

[10] Suhardi, A., Fatkhiyah, E., & Sholeh, M. (2017). Perancangan dan Implementasi SSO (Single Sign On) Menggunakan Protokol OAuth 2.0. *Jurnal Teknologi Informasi*, 5(2).

[11] Rameshbabu, V., Vijayakumaran, C., & Prabhakar, B. E. (2023). Machine Learning. *Character Lab Tips*. https://doi.org/10.53776/tips-gratitude-machine-learning

[12] Dinata, R., Akbar, H., & Hasdyna, N. (2020). Algoritma K-Nearest Neighbor dengan Euclidean Distance dan Manhattan Distance untuk Klasifikasi Transportasi Bus. *ILKOM Jurnal Ilmiah*, 12(2), 104–111. https://doi.org/10.33096/ilkom.v12i2.539.104-111

[13] Noorbehbahani, F., Rasouli, F., & Saberi, M. (2019). Analysis of Machine Learning Techniques for Ransomware Detection. In *2019 27th Iranian Conference on Electrical Engineering (ICEE)* (pp. 128–133). https://doi.org/10.1109/ISCISC48546.2019.8985139

[14] Adamu, U., & Awan, I. (2019). Ransomware Prediction Using Supervised Learning Algorithms. In *2019 IEEE 7th International Conference on Future Internet of Things and Cloud (FiCloud)* (pp. 57–63). https://doi.org/10.1109/FiCloud.2019.00016

[15] Dinata, R. K., Fajriana, F., Zulfa, Z., & Hasdyna, N. (2020). Klasifikasi Sekolah Menengah Pertama/Sederajat Wilayah Bireuen Menggunakan Algoritma K-Nearest Neighbors Berbasis Web. *CESS (Journal of Computer Engineering, System and Science)*.