BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Pada penelitian ini ada beberapa referensi terkait *Single Sign On.*

Agus Suhardi, Erfanti Fatkhiyah, dan Muhammad Sholeh (2017) dalam "Perancangan Dan Implementasi SSO (Single Sign On) Menggunakan Protokol Oauth 2.0". Pada penelitian ini menggunakan bahasa pemrograman Java dengan framework Spring boot, sebagai *Authorization* aplikasi ini menggunakan protocol OAuth 2, sedangkan basis data yang digunakan adalah MariaDB.

Indri Handayani, Erick Febriyanto, dan Fina Andhara Khumaida (2018) dalam "SSO (Single Sign On) pada Sistem PESSTA+ Berbasis Yii Framework di Perguruan Tinggi". Pada penelitian yang dilakukan pada Perguruan Tinggi Raharja, mengimplementasikan *Google Authclient* untuk mempermudah proses login bagi civitas Perguruan Tinggi Raharja.

Tio Aditya Putra (2019) dalam "Penerapan Sistem *Single Sign On* pada Institut Informatika dan Bisnis Darmajaya". Penelitian ini dibuat untuk membangun *sistem Single Sign On* pada website yang dimiliki kampus yang dapat diakses mahasiswa. Aplikasi sistem ini dibuat menggunakan Authentication Ouath2. Pada penelitian ini juga menerapkan autentifikasi login menggunakan Captcha, dimana jika *user* salah memasukkan Captcha, maka otomatis web akan me-refresh atau mengulang seperti semula (memasukkan *username* dan *password* kembali).

Theta Dinnarwaty, Winarno Sugeng, dan Resi Katri (2019) dalam "Sistem Otentifikasi *Login* dengan *Single Sign On* Untuk Mengakses Banyak Sistem". Pada

penelitian ini, sistem terdiri dari CAS dan LDAP sebagai server serta aplikasi explicite FTPS. CAS merupakan tempat ditanamkannya sistem SSO. CAS diintegrasikan dengan server LDAP yang merupakan tempat diletakkan basis data *user* dari aplikasi berbasis explicite FTPS.

I Kadek Dendy Senapartha (2021) dalam "Implementasi *Single Sign On* Menggunakan *Google Identity*, REST dan Oauth 2.0 Berbasis Scrum". Dalam penelitian ini menggunakan protokol Oauth 2.0 dengan menggunakan *Application Service Provider* (ASP) yang disediakan Google yaitu Google Identity. Sistem dini diimplementasikan ke dalam sistem *Student Relationship Management* (SRM) untuk meningkatkan keamanan akses instansi pendidikan dengan mahasiswa.

Afwan Ghiffari dan Purwono Hendradi (2023) dalam "Implementasi Single Sign On (SSO) Menggunakan Representational State Transfer (REST) dan Open Authorization (OAuth 2.0) (Studi kasus: Universitas Muhammadiyah Magelang)". Penelitian ini dibuat untuk mempermudah akses mahasiswa dan staf ke berbagai sistem layanan web menggunakan satu akun tunggal. Pada penelitian ini menggunakan protokol OAuth 2.0, menggunakan Laravel Passport yang dibangun dengan arsitektur REST Sistem yang akan dirancang menggunakan Laravel Passport sebagai Server dan arsitektur REST, terdiri dari dua jenis aplikasi, yaitu aplikasi Backend yang menggunakan bahasa pemrograman PHP dengan menggunakan Framework Laravel, dan aplikasi Frontend yang dibangun dengan menggunakan JavaScript, CSS, dan HTML.

Dari penelitan yang diusulkan penulis, terdapat beberapa perbedaan dengan penelitian-penelitian sebelumnya. Disini penulis melakukan penelitian Single Sign On (SSO) menggunakan platform Zitadel.

Dari penelitian yang telah dilakukan oleh peneliti-peneliti sebelumnya terdapat perbedaan yang bisa dilihat pada tabel berikut.

Tabel 2.1 Perbandingan Penelitian

Peneliti	**Objek**	**Teknologi**	**Bahasa Pemrograman**
Agus Suhardi, Erfanti Fatkhiyah, dan Muhammad Sholeh (2017)	IST Akprind	OAuth 2.0	Java
Indri Handayani, Erick Febriyanto, dan Fina Andhara Khumaida (2018)	Perguruan Tinggi Raharja	Google Authclient	PHP (*Hypertext Preprocessor*)
Tio Aditya Putra (2019)	Institut Informatika dan Bisnis Darmajaya	OAuth 2.0	PHP (*Hypertext Preprocessor*)

Theta Dinnarwaty, Winarno Sugeng, dan Resi Katri (2019)	Institut Teknologi Nasional Bandung	LDAP	PHP (*Hypertext Preprocessor*)
I Kadek Dendy Senapartha (2021)	Universitas Kristen Duta Wacana	OAuth 2.0	(*Hypertext Preprocessor*), JavaScript
Afwan Ghiffari dan Purwono Hendradi (2023)	Universitas Muhammadiyah Magelang	OAuth 2.0	PHP (*Hypertext Preprocessor*), JavaScript
Ardy Pramudyatama (sekarang)	Indonesia Indicator	OAuth 2.0	Java

2.2 Dasar Teori

2.2.1 Authentication dan Authorization

Authentication dan *Authorization* adalah dua konsep penting dalam manajemen keamanan informasi dan sistem komputer. Berperan penting dalam memastikan bahwa hanya *user* yang sah dan berwenang yang dapat mengakses data yang sesuai dalam suatu sistem.(Gupta, Tech Scholar, dan Sharma, t.t.)

Authentication adalah proses verifikasi *user* seperti data *user*, perangkat, atau sistem untuk memastikan bahwa identitasnya benar. Tujuan dari *authentication* adalah untuk memastikan bahwa *user* memiliki hak akses pada sistem website atau aplikasi tertentu. Ini memerlukan pembuktian *user* dengan menggunakan berbagai metode salah satunya dengan *JSON Web Token* (JWT).

Authorization adalah proses pengaturan izin dan hak akses yang diberikan kepada *user* setelah melewati proses *authentication.* Tujuan *authorization* adalah memastikan bahwa *user* hanya memiliki akses pada website atau aplikasi yang sesuai dengan *role* yang sudah ditetapkan pada sistem, seperti *user* biasa atau administrator.

2.2.2 Single Sign On (SSO)

Single Sign On adalah solusi *authorization* yang memungkinkan *user* untuk masuk ke beberapa aplikasi atau website dengan *authentication user* hanya cukup sekali *login*. Sistem SSO adalah sebuah sistem terpusat yang berbasis pada server, tidak ada akses lain selain SSO untuk bisa mengakses sebuah jaringan terstruktur yang terintegrasi. SSO juga meningkatkan keamanan dan mengurangi risiko yang disebabkan oleh kata sandi yang lemah, berulah, atau hilang.(Bazaz dan Khalique 2016)

2.2.3 OAuth 2.0

OAuth 2.0 adalah protokol standar industri untuk *authorization* yang banyak digunakan pada SSO dikarenakan kemudahan implementasi dan banyaknya dukungan dari *Application Service Provider* (ASP). OAuth 2.0 menggunakan token akses untuk memberikan izin kepada *user* yang sudah login untuk mengakses layanan yang dilindungi oleh sistem SSO.

2.2.4 JSON Web Token (JWT)

JWT adalah sistem yang mengelola identitas *user* dalam keamanan informasi. JWT digunakan untuk mengirimkan informasi yang dapat diverifikasi

antara dua pihak. Dua pihak utama adalah pihak yang menghasilkan token yaitu Identity Provider (IDP) dan pihak yang menerima token yaitu *user*.

Token JWT terdiri dari bagian utama *header*, *payload*, *dan signature*. Bagian *header* JWT berisi informasi tentang jenis token (JWT) dan algoritma yang digunakan untuk menandatangani token. *Payload* JWT berisi claim informasi identit yang mencakup ID *user*, peran, dan waktu kadaluwarsa. *Signature* JWT adalah tanda tangan digital yang digunakan untuk memastikan keaslian token yang hanya diketahui oleh user yang menghasilkan token.

2.2.5 OpenID Connect (OIDC)

OIDC adalah protokol *Authentication user*. Tujuanya adalah memberikan satu akses login untuk beberapa aplikasi atau website. Setiap kali *user* masuk ke aplikasi maupun website menggunakan OIDC, *user* akan diarahkan ke situs OpenID dan kemudian dibawa kembali ke aplikasi atau website tersebut.

OIDC termasuk Identity Provider (IDP) yang menghasilkan token identitas yang disebut ID Token. Token ini berisi *claim* identitas *user* dan dapat digunakan untuk mengidentifikasi dan mendapatkan informasi dari *user*.

2.2.6 Zitadel

Zitadel adalah *platform Identity and Acces Management* (IAM) yang menyediakan solusi keamanan dan manajemen akses untuk aplikasi dan layanan digital. Zitadel memungkinkan organisasi untuk membuat, mengelola, dan *authentication* user identitas *user*.

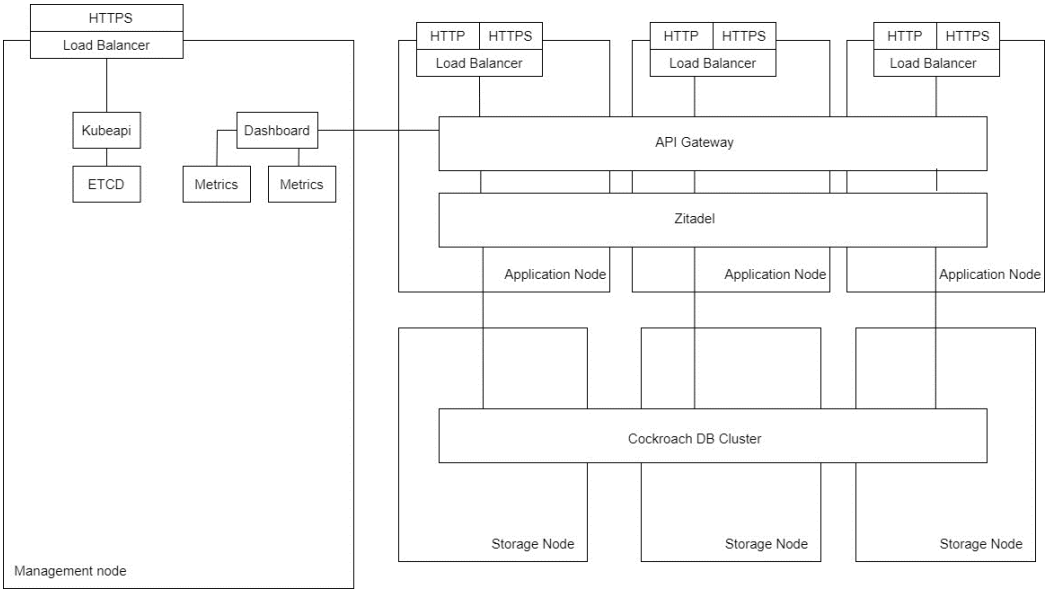
Zitadel juga mendukung *Single Sign On* (SSO), sehingga organisasi dapat mengintegrasikan aplikasi dan layanannya ke dalam platform Zitadel dan mengatur *user* untuk masuk beberapa aplikasi ataupun website dengan hanya satu *login* tunggal.

Single Sign On (SSO) menggunakan *Relational Database Management System* (RDBMS) dalam menyimpan informasi terkait *authentication* dan *authorization user*. Dalam arsitektur zitadel, RDBMS termasuk dalam *storage layer*. Berikut storage layer yang didukung oleh zitadel:

2.2.6.1 CockroachDB

CockroachDB adalah sebuah sistem basis data relasional terdistribusi yang dirancang untuk beroperasi dalam arsitektur terdistribusi, dengan kemampuan untuk menjalankan cluster multi-node yang menghindari titik kegagalan tunggal.

Zitadel menggunakan CockroachDB sebagai rekomendasi karena CockroachDB memiliki sejumlah fitur yang sesuai dengan kebutuhan manajemen *authentication dan authorization user*. CockroachDB dirancang untuk memberikan skalabilitas horizontal yang sangat baik. Dalam konteks manajemen authentication, di mana jumlah pengguna dan aplikasi dapat sangat bervariasi, kemampuan tersebut adalah fitur penting.



Gambar 2. 1 Cara Kerja CockroachDB

2.2.6.2 PostgreSQL

PostgreSQL adalah basis data relasional terpusat yang dirancang untuk berjalan pada satu server atau beberapa server yang terhubung dalam pengaturan terpusat. Transaksi dapat dijalankan pada satu node atau dalam konfigurasi yang terbatas.

Selain CockroachDB, ZITADEL juga memberikan opsi untuk menggunakan PostgreSQL. PostgreSQL adalah sistem basis data relasional yang mendukung tingkat dukungan yang lebih tinggi dalam bentuk *Enterprise Support*. Ini adalah pilihan alternatif bagi pengguna yang membutuhkan tingkat dukungan tambahan.