

---

## IMPLEMENTASI SINGLE SIGN ON DENGAN OAUTH 2.0 UNTUK EFEKTIFITAS LOGIN

<sup>1</sup>Mukrodin, <sup>2</sup>Rohmat Taufiq, Deskal Dwi Rayananda

<sup>1</sup>Sistem Informasi/Universitas Peradaban, Jawa Tengah

<sup>2</sup>Teknik Informatika/Universitas Muhammadiyah Tangerang, Banten

<sup>3</sup>Sistem Informasi/Universitas Peradaban, Jawa Tengah

Email : [mukrodins@gmail.com](mailto:mukrodins@gmail.com) (email korespondensi)

Received: 25 September 2023

Accepted: 17 Oktober 2023

### *Abstract*

The increasing function of the internet has an impact on increasing internet users. Users will have many user accounts to access the different systems they own. This often causes problems on the part of users who forget user accounts and passwords. Through the single sign on method using Oauth 2.0 authentication. In the Laravel framework, users can log in to their system with just one account. So users will not experience problems related to forgetting their user account. By using the research method, namely RAD (Rapid Application Development) in implementing a single sign on system with Oauth on PHP laravel. Users won't have to worry about forgetting their user names or passwords thanks to the study's findings since it will be simpler for them to log in to the system.

**Keywords:** *Single sign on, Oauth, Framwork Laravel, Log-in*

### **Abstrak**

Meningkatnya fungsi dari internet berdampak pada meningkatnya pengguna internet. Pengguna akan memiliki banyak *user account* dalam mengakses sistem yang berbeda - beda yang dimiliki. Hal ini sering menimbulkan masalah dari sisi *user* yang lupa *user account* dan *password*. Melalui metode *single sign on* dengan memanfaatkan autentikasi *Oauth 2.0* pada *framework* laravel pengguna dapat *login* ke sistem yang dimiliki hanya dengan satu akun saja. Sehingga pengguna tidak akan mengalami masalah terkait lupa *user account*. Dengan menggunakan metode penelitian yaitu *RAD (Rapid Application Development)* dalam mengimplementasikan sistem *single sign on* dengan *Oauth* pada *PHP laravel*. Hasil dari penelitian ini akan mempermudah user dalam melakukan aktifitas *log-in* sistem tanpa khawatir akan masalah lupa *user account* atau *password*.

**Kata Kunci:** *Single sign on, Oauth, Framwork Laravel, Log-in*

## PENDAHULUAN

Perkembangan internet yang semakin pesat memungkinkan untuk setiap orang terlibat dalam perkembangannya. Beberapa layanan di internet banyak di sukai oleh para pengguna salah satunya adalah kemudahan dan tampilan yang menarik bagi penggunanya, dengan demikian fungsi internet akan semakin banyak, begitupun dengan penggunaannya, seiring dengan meningkatnya penggunaan internet jumlah *user* juga akan semakin bertambah. Dan dengan semakin banyaknya jumlah *user*, secara tidak langsung menimbulkan masalah yang baru, tidak hanya dari sisi user itu sendiri, namun juga dari sisi admin sebagai pengelola *user*. Masalah yang dihadapi pengguna yaitu sering terjadinya lupa *user account* dan *password*, karena harus mengingat tidak hanya satu *user account*, melainkan banyak *user account* untuk masuk kedalam sistem yang berbeda – beda.

Keamanan akun internet yang nyaman dan baik, akan menyebabkan persolana baru bagi penggunaan tingkat akhir yaitu masalah rumitnya pengelolaan admin yang sulit di tembus akan memberikan langkah yang lebih panjang demi tujuan kenyamanan dan kewanaman. akhir-akhir ini mulai banyak pengembangan tentang mudahnya pengelolaan akun di internet untuk meningkatkan sistem keamanan, salah satu cara dengan pemanfaatan SSO (*Single Sign On*). Begitu cepatnya proses pengembangan sistem keamanan dan permintaan akan aktifitas penggunaan internet yang begitu masif serta didorong adanya kebutuhan sistem yang nyaman dan cepat dalam aktifitas di dunia maya, maka perlu adanya pengembangan sistem jaringan kedalam SSO. Tugas utama SSO adalah mengautentifikasi beberapa aktifitas *login website* dalam satu aktifitas *login* sehingga pengguna internet tidak perlu melakukan banyak *login* di setiap *website*.

SSO adalah sebuah cara bagaimana membuat *user* hanya perlu mengingat satu *username* dan *password* yang autentik untuk membuka beberapa layanan sekaligus. Dari dua pengertian tersebut yang diketahui bahwa *single sign on system* merupakan suatu mekanisme autentikasi untuk dapat mengakses keseluruhan sumber daya seperti beberapa situs atau layanan lainnya hanya dengan satu kali *login* saja. Sistem SSO menghilangkan *login double* dengan cara mengidentifikasi login secara ketat. Dengan menerapkan sistem SSO ini ada efisiensi waktu dan dapat meningkatkan kegunaan jaringan secara keseluruhan dan pada saat yang sama dapat memusatkan pengelolaan dari parameter sistem yang relevan. (Nugroho, 2012).

Hasil pembahasan diatas, maka penulis berinisiatif melakukan penelitian. Adapun judul penelitian yang diambil adalah “**Implementasi Single Sign On Dengan Oauth 2.0 Untuk Efektifitas Login**”. Penelitian ini diharapkan dapat menjadi sumber pengetahuan serta sebagai pendukung dalam menerapkan *Single Sign On* dengan Oauth 2.0 dalam meningkatkan efisiensi *user*, keamanan sistem informasi, dan produktivitas IT Serta kemudahan akses pengguna keseluruhan sistem yang dimiliki.

## LITERATURE REVIEW

Penelitian (Guntoro, 2018) Perancangan Aplikasi *Single Sign-On (SSO)* Menggunakan Otentikasi Gambar, Penelitian membahas tentang : analisa sistem *single sign-on (SSO)*, serta otentifikasi sistem, proses mode analisa *least significant bit* digunakan untuk menggunakan gambar sebagai otentifikasi dan analisa penggunaan *web service REST*. Penelitian ini membangun sebuah sistem yang dapat melakukan login hanya sekali pada banyak aplikasi web menggunakan otentikasi gambar. Dan dari penelitian tersebut dihasilkan bahwa hasil perancangan sistem SSO memudahkan aktifitas bagi pengguna, yaitu dengan hanya sekali melakukan otentikasi menggunakan gambar, yaitu melalui metode *least significant*, untuk masuk ke beberapa aplikasi *web*.

Penelitian (Elsera, 2021), Implementasi *single sign on* pada *web* menggunakan protokol *oauth facebook*, pada penelitian ini memiliki tujuan dimana dengan merancang *single sign on* menggunakan *authentication facebook* sebagai kemudahan dalam melakukan aktivitas *login* pada aplikasi *e-commerce*, hasil dari penelitian ini yaitu, dimana pengguna tidak lagi memasukkkn login yang sama pada aplikasi yang berbeda, dengan adanya layanan SSO ini jauh lebih cepat dan lebih akurat dengan menambah layanan facebook sebagai akun untuk autentifikasinya

Penelitian (Senapartha, 2021), Implementasi *Single Sign-On* Menggunakan *Google Identity, REST* dan *OAuth 2.0* Berbasis *Scrum*. Penelitian ini merancang sebuah sistem SSO dengan menggunakan *oauth 2.0* pada *google identity* dan arsitektur dari *REST* sebagai keamanan dari sistem. Hasil penelitian ini mengurangi adanya proses login berkali-kali,. Selain itu, layanan *Google Identity* mempercepat dan mempermudah proses implementasi SSO karena lebih berfokus pada bagaimana mengamankan transmisi data SSO pada sistem.

### METODE PENELITIAN

Menurut Kendall (2010), Rapid Application Development (RAD) adalah pendekatan pengembangan sistem berorientasi objek yang mencakup perangkat lunak dan metode pengembangan. Antara perancangan dan penerapan sistem informasi, RAD bertujuan untuk mempersingkat waktu yang biasanya diperlukan dalam siklus hidup pengembangan sistem tradisional. Pada akhirnya, RAD ingin memenuhi syarat bisnis yang berubah secara cepat.

Pressman memberikan penjelasan tambahan tentang presentasi ide yang lebih khusus. RAD adalah proses model perangkat lunak inkremental yang berfokus pada siklus pengembangan cepat. Adaptasi "kecepatan tinggi" dari model waterfall, model RAD mencapai perkembangan pesat melalui pendekatan konstruksi berbasis komponen. Proses Rapid Application Development (RAD) memungkinkan tim pengembang membuat sebuah "sistem yang berfungsi penuh" dalam waktu yang sangat singkat jika semua kebutuhan dan batasan ruang lingkup proyek telah diketahui dengan baik (Pressman, 2012).



Gambar 1. Metode rapid application development (RAD)

Dalam metode pengembangan aplikasi cepat (RAD), ada langkah-langkah berikut:

1. Perencanaan Persyaratan :

Fase ini memungkinkan pengguna dan penganalisis untuk menentukan tujuan aplikasi atau sistem serta syarat informasi yang dihasilkan dari tujuan tersebut. Dalam tahap ini,

fokusnya adalah menyelesaikan masalah yang berkaitan dengan perusahaan. Fokus tetap pada pencapaian tujuan perusahaan, meskipun beberapa sistem yang diajukan dapat dipengaruhi oleh teknologi informasi dan sistem (Kendall, 2010).

2. Desain Workshop Rapid Application Development (RAD) :  
Fase ini adalah fase desain dan perbaikan yang dapat didefinisikan sebagai workshop. Penganalisis dan pemrogram dapat bekerja sama untuk membuat representasi visual pola kerja dan desain. Bergantung pada ukuran aplikasi yang akan dikembangkan, workshop desain ini dapat berlangsung selama beberapa hari. Pengguna merespon prototipe yang ada dan penganalisis memperbaiki modul-modul yang ada selama workshop desain rapid application development (RAD).
3. Implementasi :  
Pada fase implementasi ini, penganalisis bekerja sama dengan pengguna di workshop untuk merancang aspek bisnis dan nonteknis perusahaan. Sistem baru atau bagian dari sistem diuji dan diperkenalkan kepada organisasi setelah elemen ini disetujui dan dibangun dan disaring (Kendall, 2010).

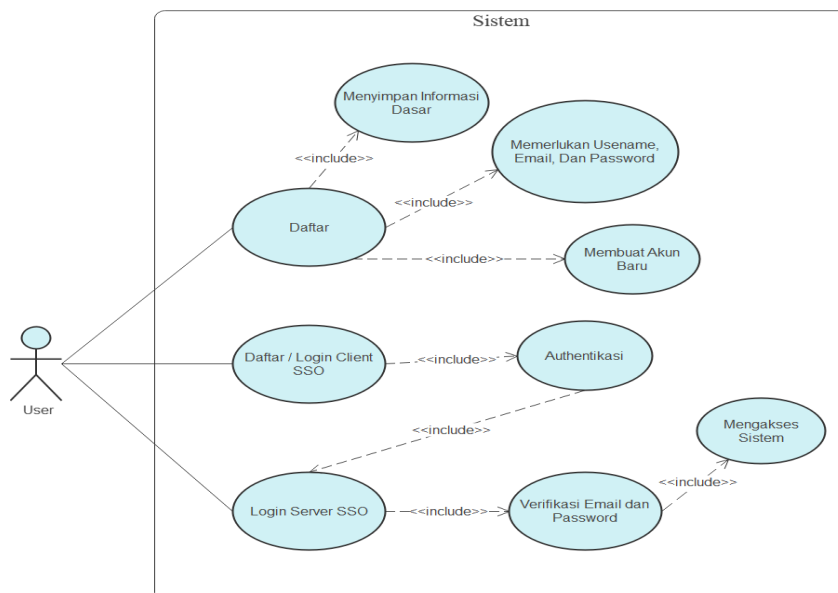
## PEMBAHASAN DAN HASIL

Pembahasan hasil penelitian adalah diskusi tentang hasil penelitian yang dibuat oleh peneliti untuk menjelaskan dan menerangkan temuan penelitian untuk menjawab pertanyaan penelitian.

1. Perencanaan Unified Modelling Language (UML) :

1. Diagram Use case

Ada use case diagram yang menjelaskan bagaimana aktor berinteraksi dengan sistem dan menjelaskan fungsi-fungsi yang ada pada aktor..

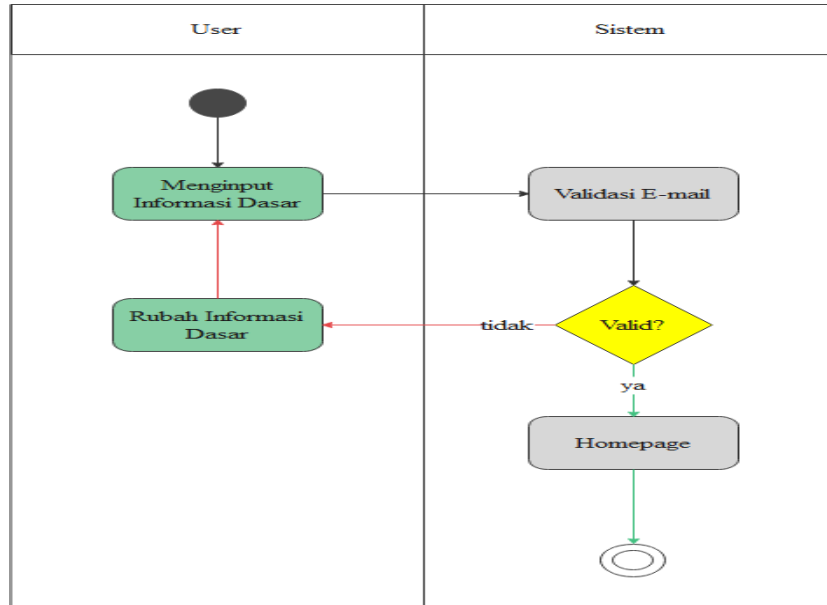


Gambar 2. Use case diagram sistem

Pada sistem single *sign on* terdapat satu aktor yaitu *user*, dimana *user* memiliki berapa fungsi yaitu melakukan pendaftaran, *log-in* ke *client SSO*, dan *Log-in* ke *Server SSO*, dimana pada saat *login* atau mendaftarkan harus mengisi sebuah informasi dasar pada sistem. Deskripsi dari *actor* dan deskripsi *use case* pada *use case* diagram.

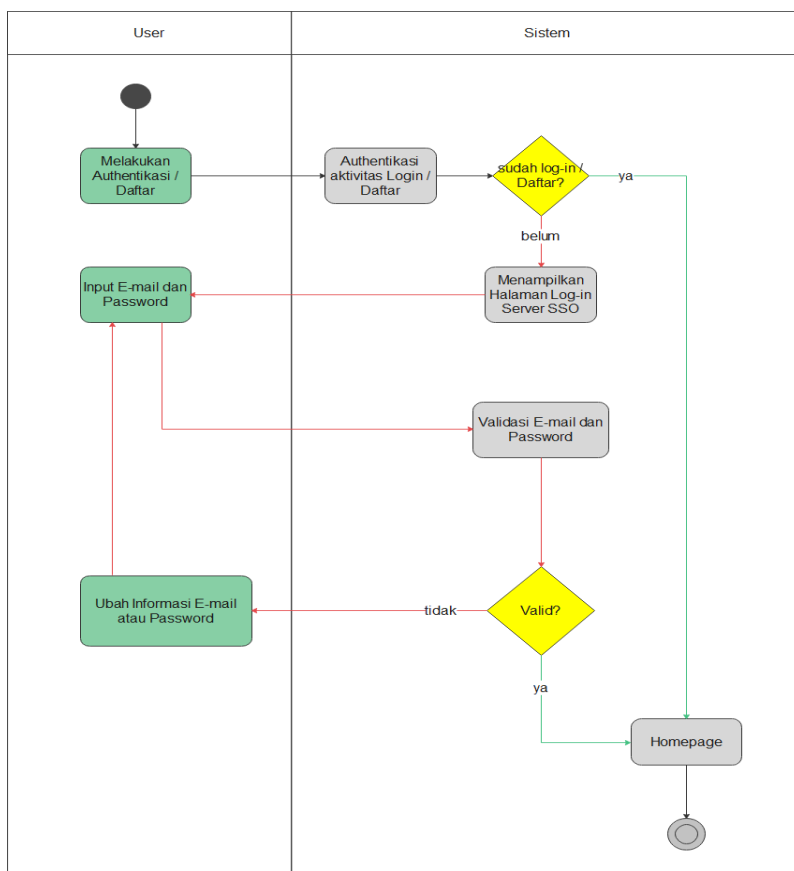
## 2. Diagram Activity

Berdasarkan use case yang telah dibuat, aktivitas diagram untuk implementasi satu tanda pada.



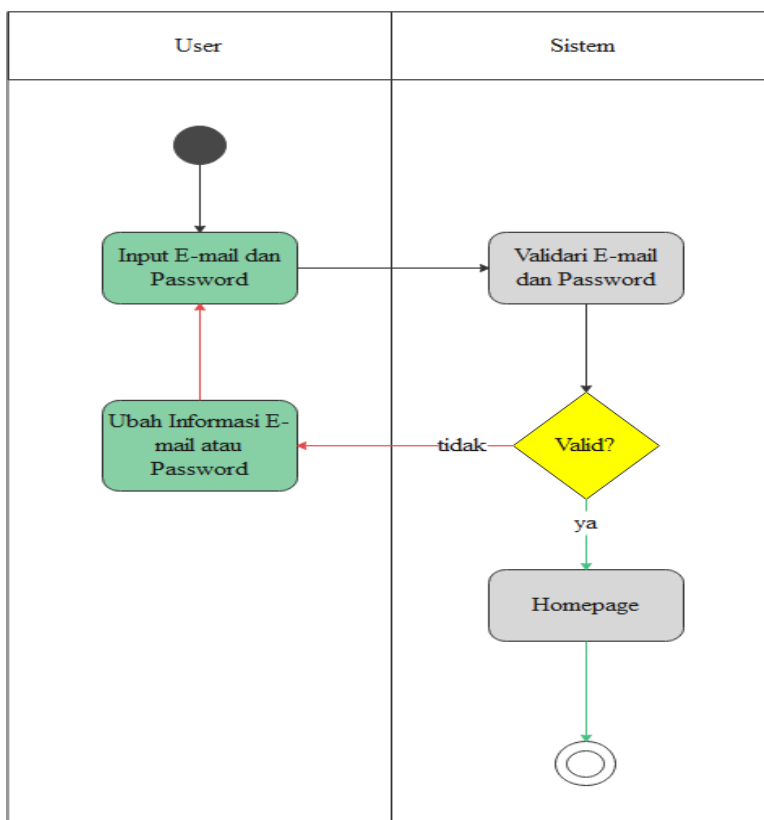
Gambar 3. Activity diagram Daftar

*Activity Diagram* Daftar menggambarkan yaitu alur yang dilakukan seorang *user* dalam melakukan aktivitas daftar ke sistem, *user* menginput informasi dasar kedalam sistem, kemudian sistem akan memvalidasi informasi yang diinputkan khususnya yaitu *e-mail*, jika informasi yang diinputkan telah sesuai maka sistem akan langsung menampilkan halaman *homepage*, namun jika terjadi suatu kesalahan pada informasi, maka *user* akan menerima peringatan kesalahan, sehingga *user* akan memperbaiki informasi dan melakukan validasi kembali



Gambar 4. Activity Diagram Log-in client SSO

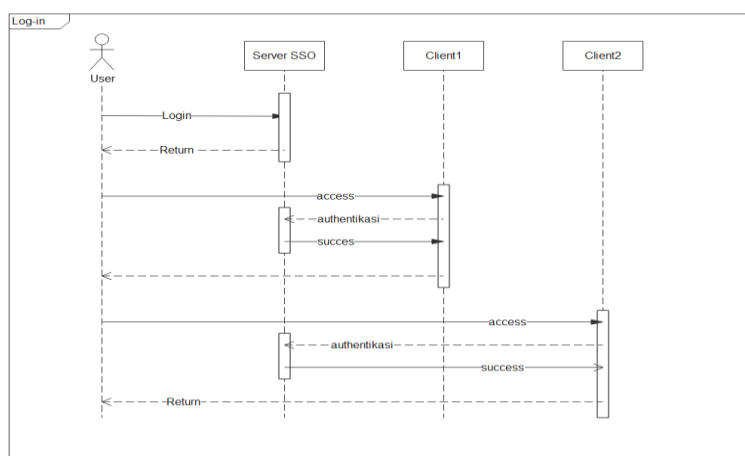
*Activity Diagram Log-in client SSO* menggambarkan alur proses dari *log-in* pada client dari sistem *SSO* oleh aktor *user*, dimana ketika *user* mengakses *client* dari *SSO*, maka akan dimintai suatu proses autentikasi, autentikasi akan diproses oleh sistem *SSO*, ketika *user* telah *log-in* pada *server SSO* maka sistem akan menampilkan halaman *homepage* sistem *Client SSO* terkait, namun ketika *user* belum melakukan *log-in* pada *server SSO*, maka sistem akan mengarahkan *user* untuk *log-in* pada *server SSO*, *user* akan meng-input *e-mail* dan *password*, yang kemudian akan divalidasi oleh sistem, jika *valid* maka sistem akan langsung menampilkan *homepage* sistem *client SSO* terkait, namun ketika *user account* kurang tepat maka *user* akan menerima peringatan kesalahan, sehingga *user* akan memperbaiki informasi dan melakukan validasi kembali.



Gambar 5. Activity diagram Log-in Server SSO

Activity Diagram Log-in Server Single Sign On menggambarkan alur proses dari log-in pada server dari sistem single sign on oleh aktor user, dimana user akan mengakses sistem dari server SSO, user akan menginput username dan password, user akan menginput e-mail dan password, yang kemudian akan divalidasi oleh sistem, jika valid maka sistem akan langsung menampilkan homepage sistem server SSO, namun ketika user account kurang tepat maka user akan menerima peringatan kesalahan, sehingga user akan memperbaiki informasi dan melakukan validasi kembali.

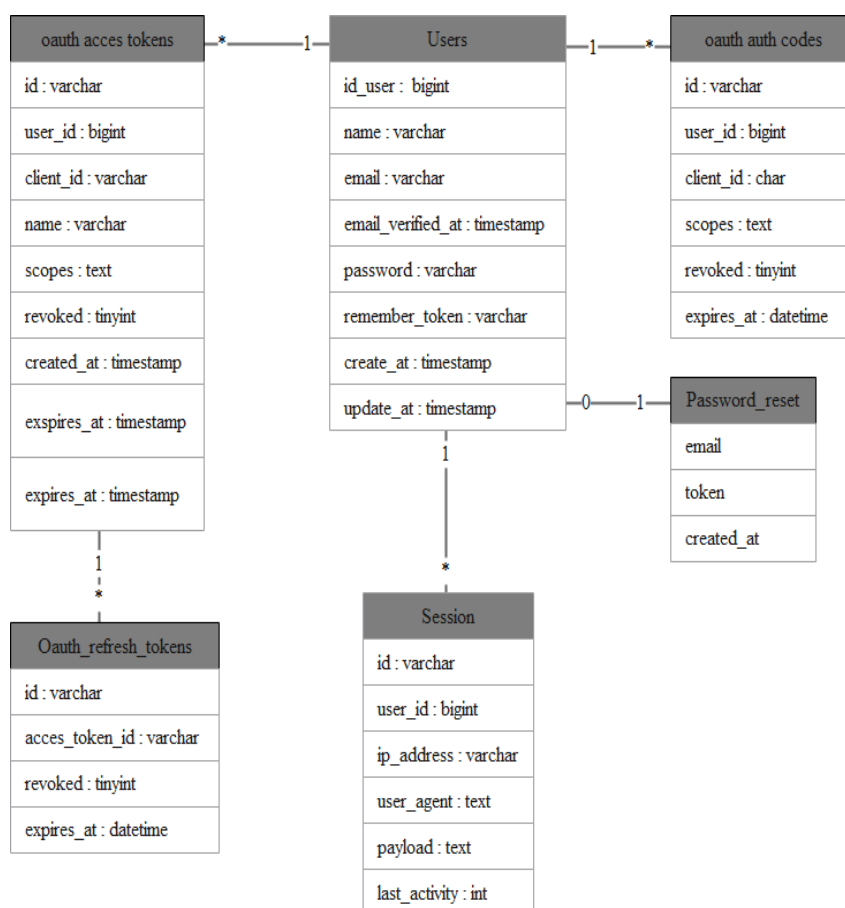
### 3. Squence diagram



Gambar 6. Squence diagram

User mengakses server sso dan melakukan *log-in.*, ketika user mengakses klien 1, maka akan melalui proses *otentikasi* ke server sso, jika berhasil akan ditampilkan *homepage* dari sistem klien 1. Begitupun ketika user mengakses klien 2, maka akan melalui proses *otentikasi* ke server sso, jika berhasil akan ditampilkan *homepage* dari sistem klien 2.

4. Class diagram

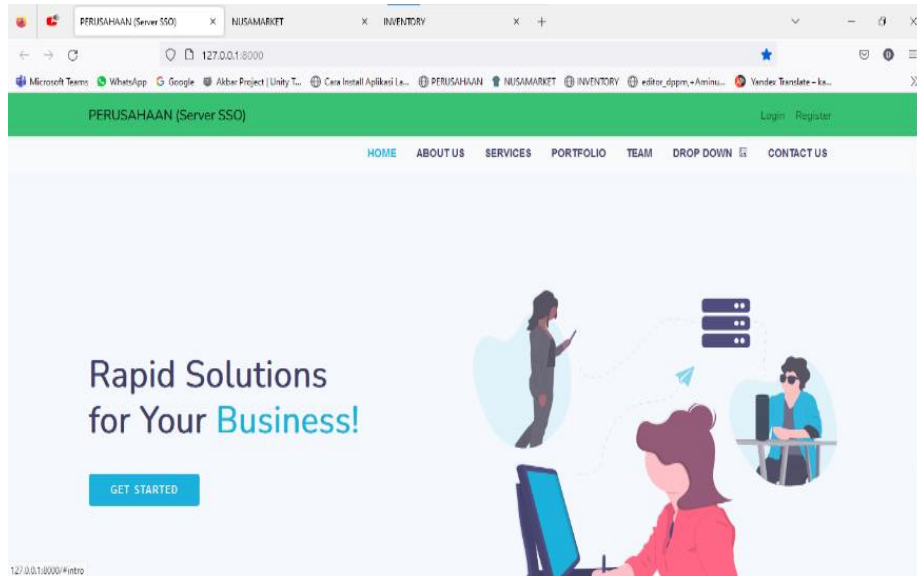


Gambar 7. Class diagram perencanaan sistem

Diagram tersebut menjelaskan *class* yang terdapat dalam sistem *single sign on* dan gambaran dari hubungan antar *class*. *Class* yang terdapat pada sistem *single sign on* adalah *class user*, *class session*, *class password reset*, *class oauth refresh tokens*, *class oauth auth codes*, dan *oauth access tokens*. Pada diagram *class* tersebut juga menggambarkan atribut atribut untuk setiap *class*

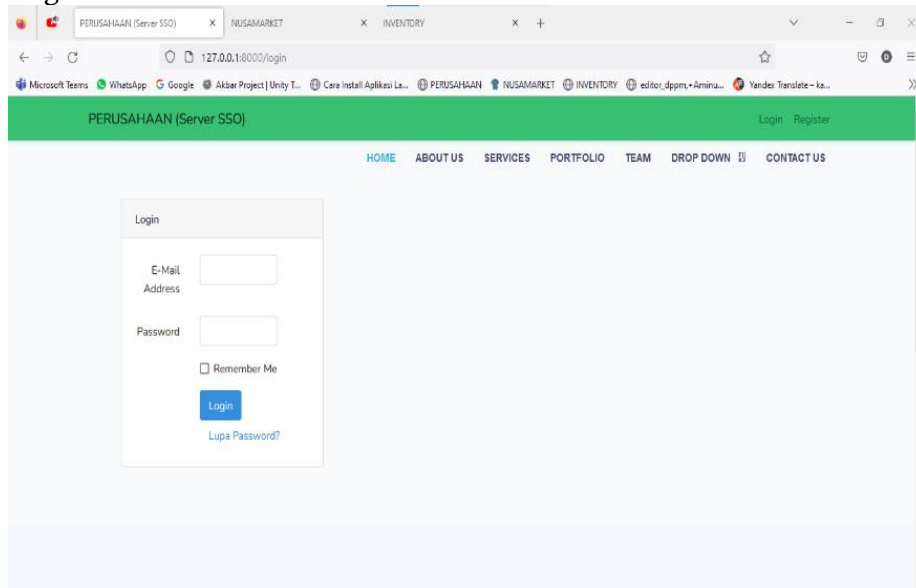


2. Implementasi sistem  
1. Homepage perusahaan



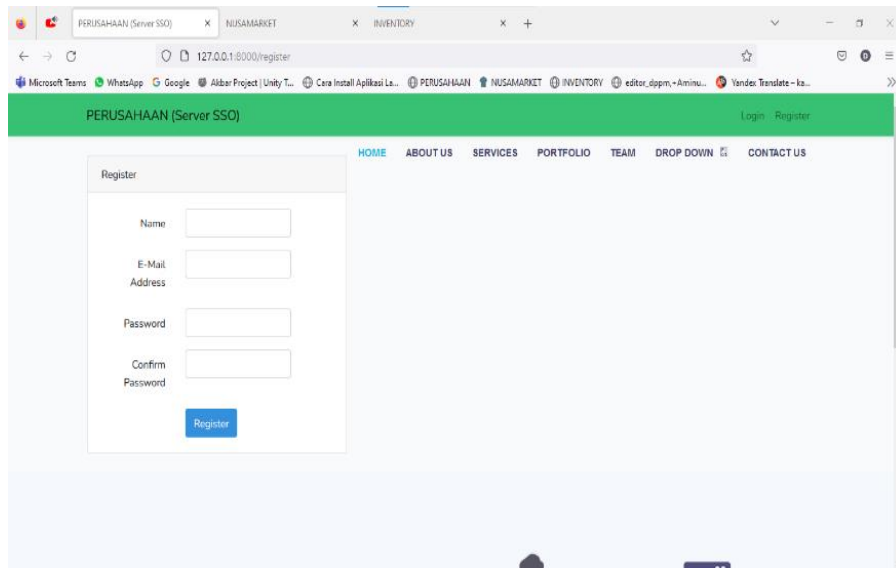
Gambar 8. Homepage Perusahaan

2. Log-in Perusahaan



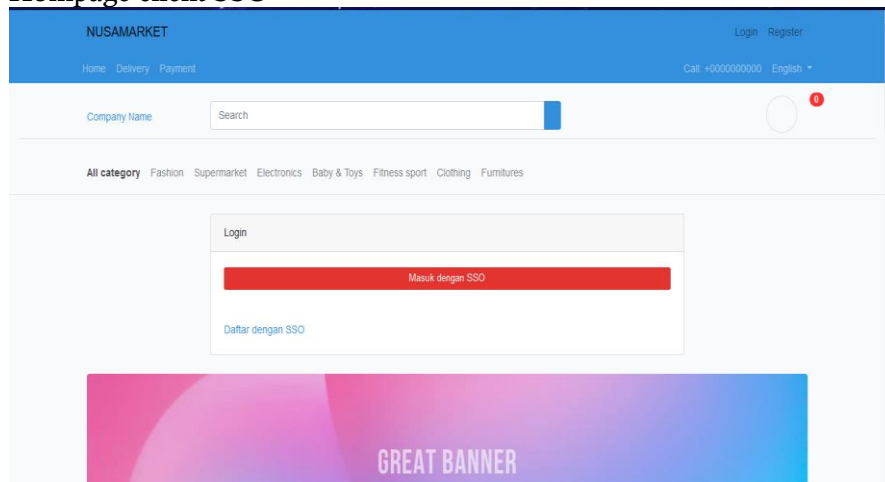
Gambar 9. Log-in Perusahaan

### 3. Register Perusahaan



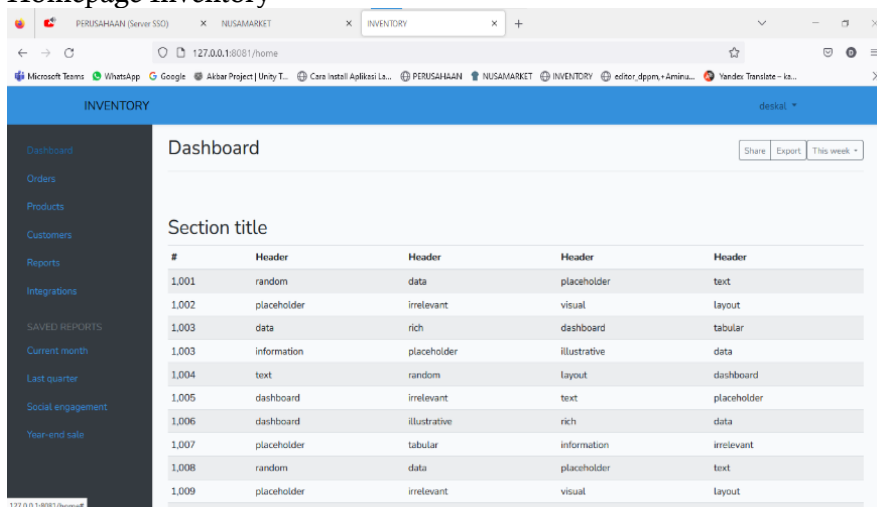
Gambar 10. Register Perusahaan

### 4. Homepage client SSO



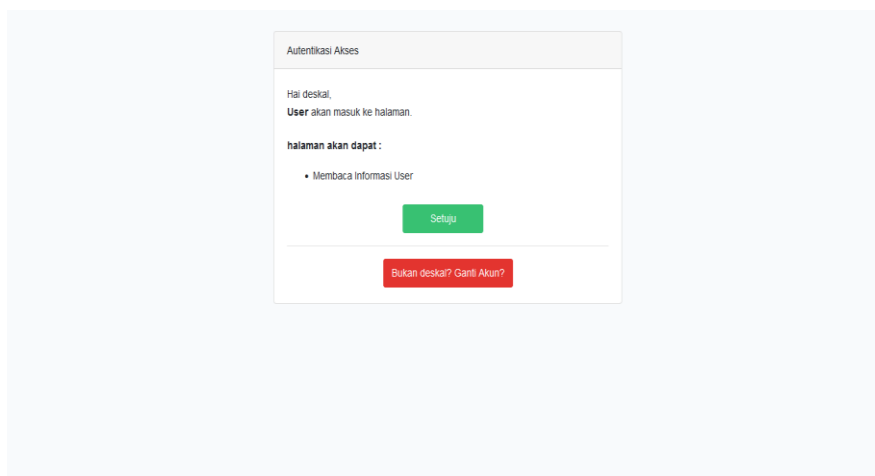
Gambar 11. Homepage Client SSO

### 5. Homepage Inventory



Gambar 12. Homepage client SSO

### 6. Autentifikasi Lo-in

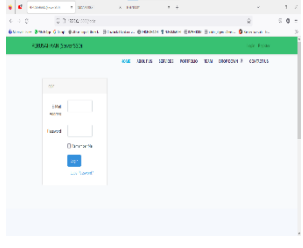
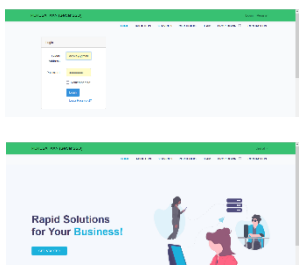
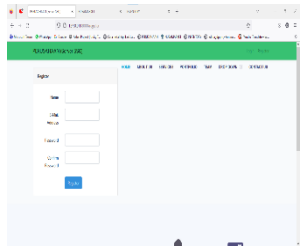


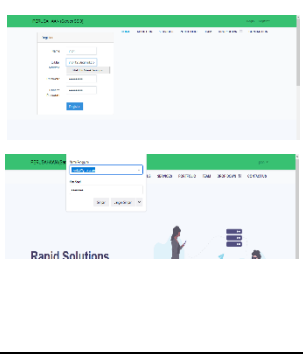
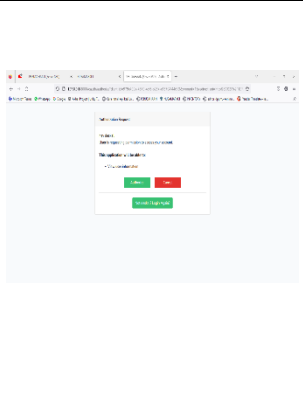
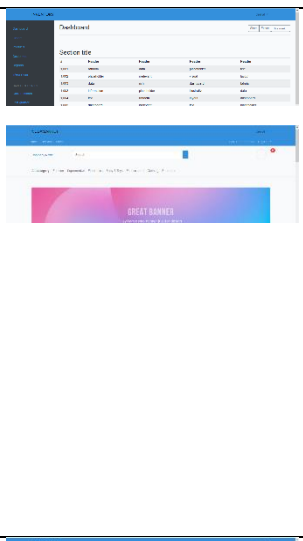
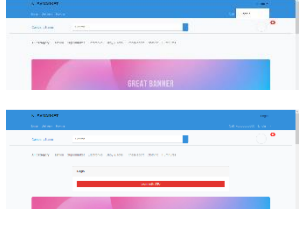
Gambar 13. Autentifikasi Log-in Client SSO

### 3. Pengujian sistem

Pengujian sistem dilakukan untuk mengetahui bagaimana keberhasilan dari sistem *single sign on* yang telah dirancang dan diimplementasikan, apabila pada tahap pengujian terjadi kendala, maka akan dilakukan proses pengulangan penelitian ke tahap sebelumnya. Pengujian ini bertujuan mengetahui apakah sistem *single sign on* telah memenuhi kriteria dan sesuai dengan tujuan yang diinginkan. Pengujian sistem *single sign on* pada penelitian ini menggunakan *black box testing* yang berfokus pada persyaratan fungsional sistem. Berikut adalah tabel hasil pengujian menggunakan *black box* :

Tabel 1. Hasil Pengujian Black Box

No	Modul	Output yang diharapkan	Hasil	Validasi	
				Y	T
1.	<i>Log-in</i>	Tampil <i>form log-in</i> untuk masuk ke dalam sistem.		✓	
2.	Proses <i>Log-in</i>	Dapat melakukan log-in ke dalam sistem.		✓	
3.	<i>Register</i>	Tampil <i>form register</i> untuk melakukan pendaftaran <i>user</i> ke sistem.		✓	

4.	Proses Register	Dapat melakukan register ke dalam sistem.		✓	
5.	Authentikasi	Tampil form autentikasi, untuk masuk ke sistem klien sso.		✓	
6.	Proses Authentikasi	Tampil halaman homepage sistem client setelah proses autentikasi.		✓	
7.	Log-out	User berhasil keluar dari sistem.		✓	

Dari pengujian melalui *black box* dapat disimpulkan bahwa sistem berjalan dengan baik sesuai dengan kebutuhan yang diinginkan, yaitu melakukan *login* dan *register* ke dalam sistem, serta berhasil melakukan *autentikasi login* klien melalui proses *single sign on* dengan *autentikasi*, dan diakhir *user* berhasil *log out* dan mengakhiri proses *autentikasi* dari server

### UCAPAN TERIMAKASIH

Alhamdulillah Penelitian dapat berjalan lancar dan selesai tepat waktu, dengan bantuan beberapa pihak, terutama kepada mahasiswa sistem informasi, teman-teman dosen dan institusi Universitas Peradaban yang selalu mendukung penuh terlaksananya penelitian ini.

### SIMPULAN DAN SARAN

Sistem SSO yang dibangun berhasil diterapkan dengan menggunakan *Oauth 2.0* pada *PHP laravel*, dan dapat memudahkan pengguna dalam melakukan login ke beberapa sistem hanya dengan menggunakan satu akun saja, selain itu pengguna tidak perlu lagi untuk mengingat banyak *user* akun, jadi akan meminimalisir masalah pada pengguna terkait dengan lupa *user* akun dan *password*. Untuk saran perlu adanya tambahan fitur yang dapat memilih akses website mana saja yang dijadikan tanpa melakukan autentifikasi lagi

### DAFTAR PUSTAKA

- Aminudin. (2014). Implementasi Single Sign On (Sso) Untuk Mendukung Interaktivitas Aplikasi E-Commerce Menggunakan Protocol Oauth. *Jurnal Gamma*.
- Dede Wira Trise Putra, R. A. (2019). Unified Modelling Language (UML) dalam Perancangan Sistem Informasi Permohonan Pembayaran Restitusi SPPD. *Jurnal TEKNOIF*.
- Elsera, M. (2021). Implementasi Single Sign On Pada Web Menggunakan Protocol Oauth Facebook. *Buletin Utama Teknik*.
- Enterprise, J. (2019). *Belajar Pemrograman dengan Visual Studio*. Elex Media Komputindo.
- Faqih, H. (2021). *Modul Dasar GIT dan Github Untuk Matakuliah Praktikum*. BSI.
- Ginting, E. (2013). Aplikasi Penjualan Berbasis Web (e-Commerce) menggunakan Joomla pada mutiara fashion.
- Guntoro, M. F. (2018). Perancangan Aplikasi Single Sign-On (SSO) Menggunakan Otentikasi Gambar. *Jurnal Teknologi Informasi & Komunikasi Digital Zone*.
- Hamidillah Ajie, M. I. (2019). Pengembangan Teknologi Single Sign On Pada Sistem Informasi Dosen dan Sistem Informasi Kurikulum di Universitas Negeri Jakarta. *Jurnal Pinter*.
- Marwan Darwish, A. O. (2015). Evaluation of an OAuth 2.0 Protocol Implementation for Web Server Applications . *IEEE* .
- Muhammad Zidan, S. N. (2022). Black Box Testing pada Aplikasi Single Sign on (SSO) di Diskominfostandi Menggunakan. *WJIT: Walisongo Journal of Information Technology*, 129.
- Nugroho, P. P. (2012). pengembangan model single sign on untuk layanan internet dan proxy ipb. *UT - cOMPUTER sCIENCE*.
- Pratama, A. (2015). *Bootstrap Uncover - Panduan Belajar Framework Bootstrap 5*. DuniaIlkom.
- Pratama, A. (2015). *CSS Unover - Paduan Belajar CSS untuk Pemula*. DuniaIlkom.
- Pratama, A. (2015). *HTML Uncover - Panduan belajar HTML untuk Pemula*. DuniaIlkom.
- Pratama, A. (2019). *Laravel Uncover - Paduan Belajar Framework Laravel 9*. DuniaIlkom.

- 
- Senapartha, I. K. (2021). Implementasi Single Sign-On Menggunakan Google Identity, REST dan OAuth 2.0 Berbasis Scrum. *Jurnal Teknik Informatika dan Sistem Informasi*.
- Tua, B. (2018). *Node.js Handbook Untuk Pemula*. BonzDev.
- Wibowo, O. T. (2019). Sistem Informasi Penjualan, Pelayanan, Gudang, Tracking Dokumen, Penagihan Menggunakan Sms Gateway Dan Android Messenger Berbasis Web ( Studi Kasus Pt. Teruna Gema Nusa ).