BAB I
PENDAHULUAN

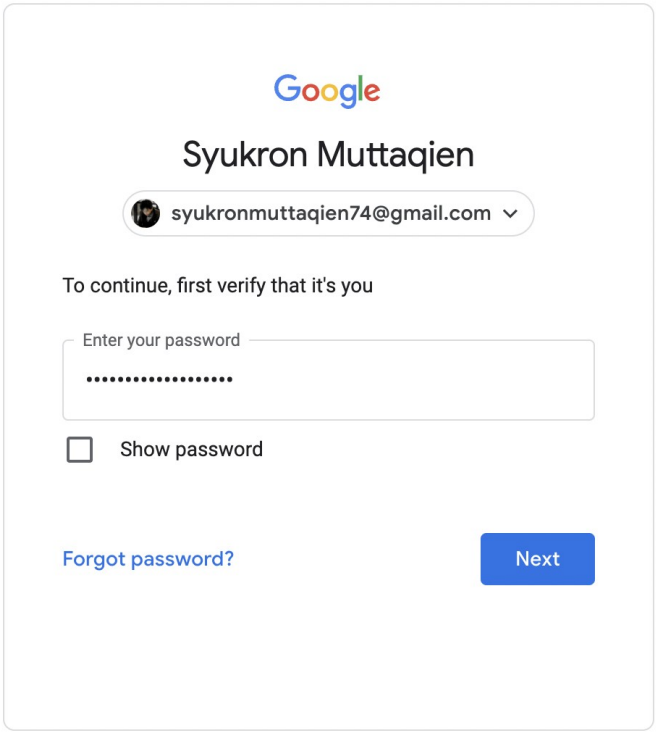
A. Latar Belakang

Dalam sebuah perusahaan informasi merupakan hal yang penting karena dengan adanya informasi perusahaan dapat melakukan pengambilan keputusan berdasarkan informasi yang diperoleh, (Jogiyanto, 2004) dalam bukunya yang berjudul Analisis dan Desain Sistem Informasi, berpendapat bahwa informasi adalah data yang diolah menjadi bentuk yang lebih berguna bagi yang menerimanya. Pada dasarnya data merupakan sekumpulan informasi atau juga keterangan– keterangan dari suatu hal yang diperoleh dengan melalui pengamatan atau juga pencarian ke sumber – sumber tertentu. Data yang diperoleh namun belum diolah lebih lanjut dapat menjadi sebuah fakta atau anggapan. Sebagai contoh, data yang diperoleh dari sebuah penelitian dengan menggunakan metode-metode tertentu, dapat menjadi lebih kompleks untuk menyajikan sebuah informasi baru atau bahkan solusi untuk menyelesaikan masalah tertentu. Data di Bagi ke dalam 2 Jenis data yaitu data kualitatif dan data kuantitatif. Data Kualitatif yaitu data yang disajikan dalam bentuk verbal (lisan/kata) bukan dalam bentuk angka. Data Kuantitatif yaitu jenis data yang dapat diukur atau dihitung secara langsung, yang berupa informasi atau penjelasan yang dinyatakan dengan bilangan atau berbentuk angka. Suatu data dapat diperoleh berdasarkan sumber, yang dikelompokkan menjadi dua yakni data primer dan data sekunder. Data Primer atau data asli merupakan data yang dikumpulkan dan berasal dari sumber asli atau tangan pertama. Data ini harus dicari melalui narasumber atau responden yaitu orang yang dijadikan obyek penelitian atau orang yang kita jadikan sebagai sarana mendapatkan informasi ataupun data. Contoh data primer yakni hasil wawancara. Data Sekunder adalah data yang mengacu pada informasi yang dikumpulkan dari sumber yang telah diolah. Contoh data sekunder antara lain catatan atau dokumentasi perusahaan; publikasi pemerintah seperti buku, laporan, berita; analisis oleh media, situs web, jurnal, dan lainnya.

Informasi dalam perusahaan bersifat sensitif dan tidak semua orang dalam sebuah perusahaan dapat mengakses informasi tersebut, hanya orang yang telah di autentikasi dengan otorisasi tertentu yang dapat mengakses informasi tersebut, sehingga informasi itu dapat digunakan sesuai kebutuhan. Informasi dapat ditemukan dalam format dan bentuk apa pun, baik di media cetak maupun media online. Sebuah data dapat dikatakan sebagai informasi ketika benar-benar berfungsi atau bisa benar-benar digunakan. Seperti dikutip dari Arkanasas State University, ada beragam jenis informasi yang bisa kita temui, seperti informasi nyata, analisis, subjektif, dan objektif. Informasi nyata atau

faktual adalah informasi yang hanya berhubungan dengan fakta. Biasanya, jenis informasi ini jarang memberikan latar belakang yang mendalam tentang suatu topik tertentu. Adapun informasi analisis ialah sebuah informasi yang biasanya dihasilkan peneliti dalam studi tertentu. Sementara itu, informasi subjektif adalah informasi yang hanya dilihat dari satu sudut pandang. Biasanya, informasi ini berisi tentang pendapat atau argumentasi dari pihak tertentu. Sedangkan, informasi objektif merupakan informasi yang dapat dipahami dari berbagai sudut pandang. Informasi menyediakan peristiwa dan kondisi dalam masyarakat tertentu, menunjukkan hubungan kekuasaan, serta memudahkan berbagai macam inovasi. Dengan begitu, masyarakat umum bisa memperoleh informasi yang yang berkaitan dengan kebutuhan dan kepentingannya dan sebagai sumber pengetahuan baru.

Autentikasi merupakan bagian dari sistem, Menurut (Sujarwo, 2010) Autentikasi adalah sebuah usaha pengecekan identitas seseorang pengguna sistem komunikasi pada proses login ke dalam sebuah sistem, sehingga dalam proses penggunaan sebuah sistem diperlukan Autentikasi terlebih dahulu agar orang dapat masuk ke sistem tersebut. Dalam konteks keamanan data, authentication atau autentikasi adalah sebuah metode untuk memeriksa kebenaran identitas pengakses data secara autentik.bagi pengguna ketika ingin melakukan login atau membuka file yang ter-enkripsi, autentikasi banyak diterapkan misalnya pada website atau aplikasi. Autentikasi menyediakan kontrol akses untuk sistem dengan mencocokkan apakah kredensial pengguna sesuai kredensial pada database pengguna yang berwenang (*server data*). Apabila kredensial (data/identitas) pengguna sesuai dengan yang terekam di kredensial sistem, maka pengguna tersebut diizinkan untuk mengakses file, aplikasi, atau sistem. Autentikasi dikembangkan lebih jauh lagi dengan meminta beberapa informasi pribadi misalnya sidik jari biometrik untuk memastikan keamanan akun dari orang-orang yang memiliki kemampuan teknis untuk membobol kelemahan sistem.Bisa dibilang, autentikasi adalah enkripsi ganda, dan dengan adanya autentikasi ini dapat menjamin bahwa sistem dan data akan jauh lebih aman dari tangan orang yang tidak bertanggung jawab. Faktor autentikasi adalah kredensial milik pengguna yang sangat spesifik, seperti ID dan password.



Gambar 0.1 Autentikasi Pada Google

Gambar di atas adalah contoh faktor autentikasi dalam website google dimana pengguna harus memasukan password agar dapat masuk ke dalam sistem google.

Otorisasi juga merupakan bagian dari sistem, Menurut (Raga Maran, 2001) Max Weber mendefinisikan kekuasaan sebagai kecenderungan seseorang untuk berperilaku sesuai kehendaknya, sehingga dalam sebuah sistem informasi harus memiliki otorisasi tertentu untuk menggunakan informasi yang ada dalam sebuah sistem. Otorisasi dalam sistem berperan sebagai pengatur dimana beberapa aktifitas dibatasi sesuai dengan fungsi dari pengguna nya, dalam hal ini dapat menjadi tolak ukur suatu keamanan internal agar informasi yang ada di internal tidak dapat digunakan oleh pengguna yang tidak berwenang maupun bertanggung jawab. Otorisasi dalam setiap perusahaan berbeda - beda dalam praktik nya, sehingga setiap perusahaan menentukan otorisasi terhadap sistem nya sendiri, otorisasi dalam sistem informasi juga berperan penting dalam keamanan sebuah informasi, selain diterapkan nya autentikasi sebagai pelindung dari luar, otorisasi berperan sebagai pelindung dari dalam suatu perusahaan, sehingga keamanan ganda terbentuk untuk mengamankan sebuah informasi.

Berikut merupakan contoh level otorisasi dalam sebuah sistem informasi di dalam suatu perusahaan:

Tabel 0.1 Contoh Otorisasi Dalam Sistem Informasi

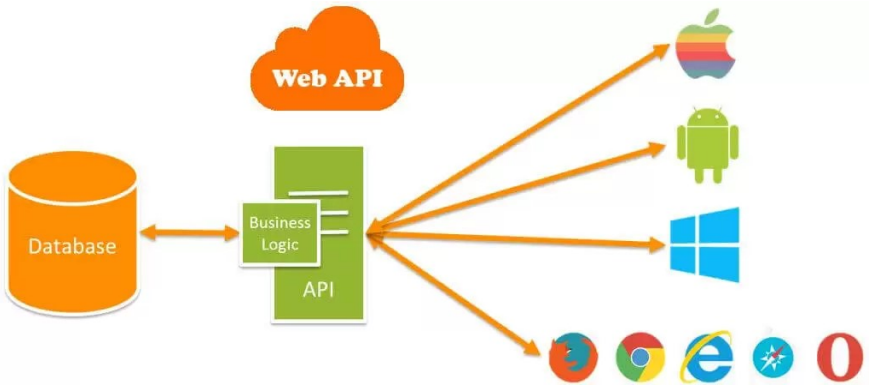
Subjek	**Objek**	**Aktivitas yang dapat dilakukan**	**Kekangan / Limitasi**
Owner	Data Karyawan	Registrasi Data Karyawan	Tidak Ada
		Melihat Semua Data Karyawan	
		Menambah Hak Ases Pada Data Karyawan	
Manager	Data Karyawan	Registrasi Data Karyawan	Hanya bisa registrasi karyawan yang berada di bawah manager
		Melihat Semua Data Karyawan	Tidak Ada
Marketing	Data Karyawan	Melihat Sebagian Data Karyawan	Hanya bisa melihat sebagian data karyawan sesama marketing saja, data sensitif karyawan tidak dapat dilihat
Admin	Data Karyawan	Melihat Sebagian Data Karyawan	Hanya bisa melihat sebagian data karyawan sesama admin saja, data sensitif karyawan tidak dapat dilihat

Pada Tabel 1.1 di atas dapat dilihat beberapa otorisasi yang telah ditentukan dalam sistem informasi untuk mengamankan data karyawan dari pengguna yang tidak berwenang sehingga data karyawan di jamin keamanan nya di dalam perusahaan tersebut.

Dalam Sistem Informasi pasti memiliki *Database* yang digunakan untuk menyimpan data yang nantinya akan diolah menjadi informasi, database yang biasa digunakan juga merupakan *RDBMS (Relational Database Management System)* yaitu melayani sistem basis data yang entitas utamanya terdiri dari tabel-tabel yang mempunyai relasi dari satu tabel ke tabel yang lain.

Dalam Ilmu Komputer terdapat teknologi yang dapat menghubungkan suatu aplikasi dengan aplikasi lain yang disebut dengan *API (Application Programming Interface)*, merupakan sebuah *interface* yang dapat diimplementasikan menggunakan *software* sehingga beberapa aplikasi yang terhubung dapat berbagi informasi satu sama lain.

API adalah singkatan dari *Application Programming Interface*. API sendiri merupakan interface yang dapat menghubungkan satu aplikasi dengan aplikasi lainnya.Dengan kata lain, peran API adalah sebagai perantara antar berbagai aplikasi berbeda, baik dalam satu platform yang sama atau pun lintas platform. Perumpamaan yang bisa digunakan untuk menjelaskan API adalah seorang pelayan di restoran. Tugas pelayan tersebut adalah menghubungkan tamu restoran dengan juru masak. Jadi, tamu cukup memesan makanan sesuai daftar menu yang ada dan pelayan memberitahukannya ke juru masak. Nantinya, pelayan akan kembali ke tamu tadi dengan masakan yang sudah siap sesuai pesanan. pada Gambar 1.2 di bawah ini *Database* dapat di umpamakan sebagai dapur dimana tempat juru masak berada, *API* sebagai pelayan nya, dan *Platform* itu sebagai tamu yang memesan makanan.



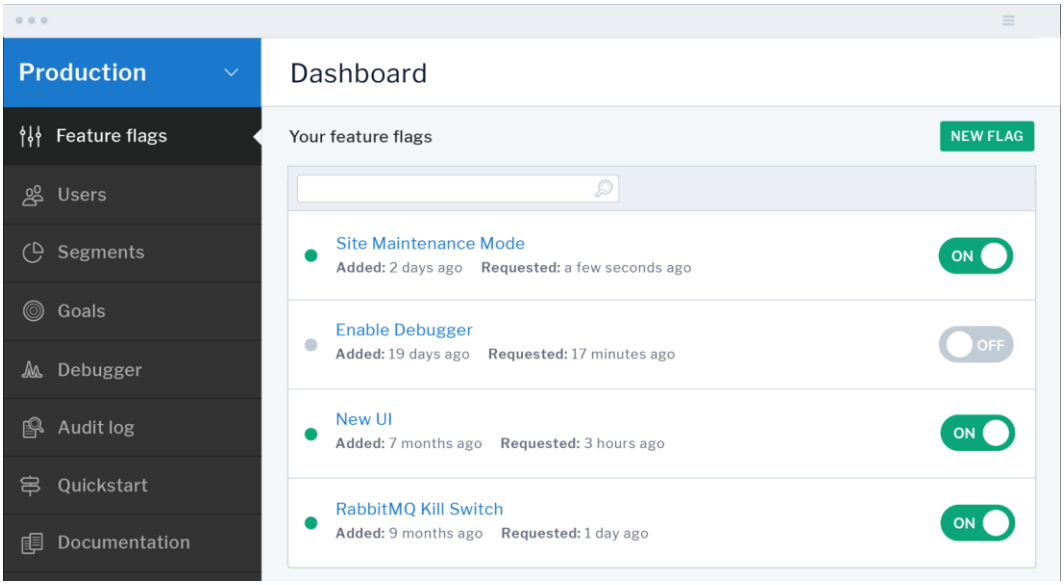
Gambar 0.2 API sebagai Penghubung Antar Aplikasi

API yang akan dibuat menggunakan format *JSON (Javascript Object Notation)* untuk memudahkan komunikasi antar aplikasi, dan untuk Autentikasi menggunakan data dari *Database* lalumenggunakan *JWT (JSON WEB Token)* agar data yang digunakan untuk Autentikasi tidak dapat di rubah atau di rusak dengan mudah. Dengan menggunakan API, akan lebih mudah untuk membuat aplikasi yang fungsional dan

kompleks. Tanpa perlu menambahkan data secara manual, aplikasi yang dikembangkan akan memiliki fitur dari aplikasi tujuan. Sebagai contoh, pada aplikasi Gojek. Sebagai sebuah platform layanan transportasi, peran peta sangatlah penting. Namun, Gojek tidak perlu mengembangkan aplikasi peta sendiri. Dengan API, aplikasi tersebut cukup mengambil data dari Google Maps. Dengan adanya API, developer aplikasi tidak perlu melakukan komunikasi langsung dengan aplikasi lain yang ingin dihubungkan. Cukup dengan komunikasi melalui API. Hal ini sangat membantu, terutama jika Anda ingin membangun aplikasi lintas platform dengan berbagai layanan sekaligus.

Kini pemikiran developer aplikasi bukanlah hanya mengenai bagaimana mengembangkan aplikasi dengan arsitektur yang baik. Salah satu masalah yang penting untuk selalu dipikirkan adalah ketika aplikasi sudah rilis ke tahap produksi maka kita tidak memiliki kuasa penuh akan aplikasi yang kita rilis tersebut. Meskipun kita telah memperbaiki error pada aplikasi kita, kita tidak bisa menjamin semua user yang memakai akan langsung selesai masalah nya.

Maka dari itu ada teknik *Feature Flag (Feature Toggles)* diperkenalkan, dimana kita dapat mengontrol aplikasi kita. Salah satu penerapan untuk aplikasi adalah kita dapat mengontrol komponen mana yang seharusnya ditampilkan ke pengguna. *Feature Flag* merupakan metode yang digunakan oleh pengembang perangkat lunak yang bertujuan untuk mengurangi risiko, melakukan iterasi lebih cepat, dan mendapatkan lebih banyak kontrol. *Feature Flag* memungkinkan untuk memisahkan peluncuran fitur dari penerapan kode. Pemisahan ini memungkinkan untuk mengontrol siapa saja yang dapat melihat fitur tersebut, terlepas dari rilis *Feature Flag* juga memberikan kontrol atas rilis dari perangkat lunak. Berikut contoh metode *Feature Flag* yang telah banyak digunakan oleh pengembang *software*:



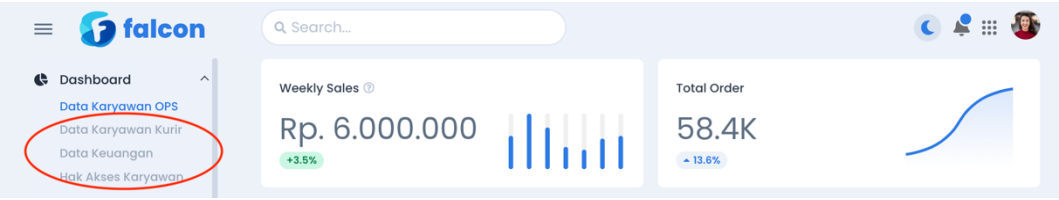
Gambar 0.3 Contoh Feature Flag

Pada Gambar 1.3 di atas dapat dilihit bahwa semua fitur yang ada pada sebuah website dapat ditampilkan dan dapat dihilangkan hanya menggunakan tombol on/off, sehingga fitur ini memberikan kontrol terhadap fitur yang akan di akses oleh pengguna. Developer aplikasi dapat membuat sebuah API yang berisi komponen mana atau fitur mana yang seharusnya ditampilkan, API tersebut di panggil ketika aplikasi pertama kali dibuka. Jadi, ketika aplikasi tersebut dibuka maka aplikasi akan melakukan *request* kepada *Feature Flag* API, dan API akan mengeluarkan response dalam bentuk *JSON* yang berisi fitur atau komponen mana saja yang harus disajikan kepada user.

Developer aplikasi juga dapat membuat web portal untuk mengontrol komponen atau fitur mana saja yang seharusnya di tampilkan kepada user. Dengan menggunakan *Feature Flag* seperti ini jika terdapat bug pada aplikasi, developer dapat dengan mudah menghilangkan sementara fitur tersebut, tanpa perlu menunggu. sehingga developer dapat memperbaikinya tanpa memberikan pengalaman buruk kepada pengguna meskipun nanti tetap memperbaiki dan merilis ulang aplikasi tersebut, namun dengan menonaktifkan fitur tersebut dari aplikasinya, developer dapat mengurangi pengalaman buruk pengguna ketika menggunakan aplikasi.

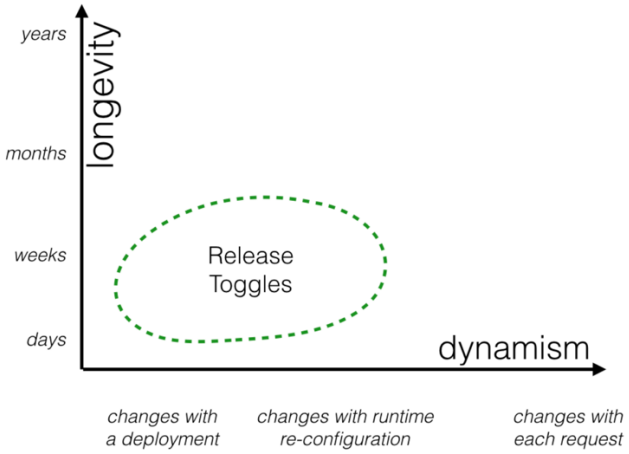
B. Permasalahan

Dalam pengembangan Sistem Informasi pada masa kini. masih sering terjadi beberapa hal yang dianggap biasa oleh *software developer* seperti masalah otorisasi yang diterapkan masih belum di terapkan secara baik dalam mengakses suatu fitur yang ada dalam sistem informasi. sebagai contoh ada pada gambar berikut:



Gambar 0.4 Penerapan Otorisasi Yang Belum Tepat Pada Menu Dashboard

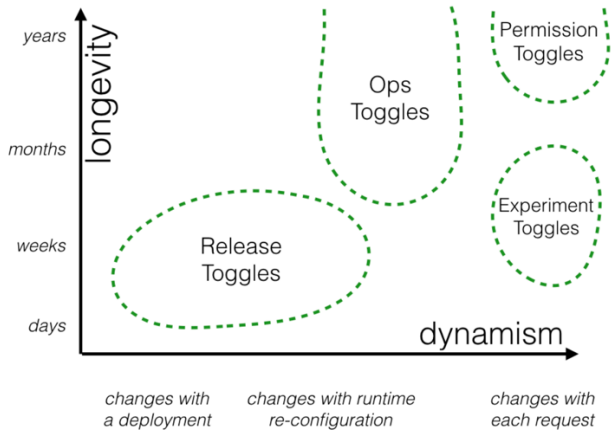
Pada Gambar 1.4 di atas, penerapan nya masih kurang tepat karena menu yang seharusnya tidak dapat diakses masih bisa terlihat, walaupun tidak bisa di klik karena menu tersebut terlihat tidak aktif, tetapi sebaiknya menu tersebut dihilangkan saja agar tidak terjadi rasa penasaran dari pengguna.



Gambar 0.5 Masa dan Dinamisme Feature Flag Kategori Release

Pada sistem yang berjalan saat ini, untuk pergantian proses otorisasi dapat dilakukan pada proses *release* atau bisa disebut sebagai proses *deployment,* sehingga proses pergantian otorisasi akan memakan waktu yang lebih lama, dan berdasarkan gambar 1.5 di atas, fitur nya hanya dapat bertahan dalam waktu yang singkat, dan perubahan nya pun hanya bisa di lakukan pada saat proses *deployment*, sehingga belum bisa dikatakan efektif.

Menurut (Fowler, 2017) ada grafik yang menunjukan bahwa *Feature Toggles* dengan Kategori *Permission* memiliki Masa dan Dinamisme yang tinggi sebagai berikut:



Gambar 1.6 Masa dan Dinamisme Feature Flag Kategori Permission

Berdasarkan gambar grafik di atas, *Feature Flag* dengan kategori *Permission Toggles* memiliki Masa dan Dinamisme yang cukup tinggi dimana Masa nya bisa sampai tahunan dan dinamisme nya sendiri lebih flexibel karena dapat di rubah dalam setiap *request*.

Dari gambar dan grafik di atas dapat disimpulkan bahwa proses penerapan tersebut masih bisa dibilang belum tepat dan untuk masa dan dinamisme nya relatif rendah karena perubahan pengembangan hanya dapat dilakukan dalam tahap deployment sehingga dapat dikatakan kurang efektif, berdarkan gambar dan grafik tersebut terdapat beberapa proses yang perlu diperbaiki.

Selain itu juga pengembangan suatu fitur lebih baik memikirkan masa dan dinamisme yang tinggi sehingga pengembangan aplikasi tidak harus berganti metode dalam jangka waktu yang singkat.

1. Identifikasi Masalah

   Dari Latar Belakang di atas identifikasi masalahnya sebagai berikut:

   a. Belum tepatnya penerapan pengendalian atau kontrol dalam menampilkan informasi
   b. Belum efektifnya proses otorisasi untuk mengakses informasi pada sistem informasi

2. Pernyataan Masalah

   Dalam perusahaan terdapat beberapa regulasi terkait dengan informasi sensitif dan tercantum dalam kebijakan privasi perusahaan, maka fitur otorisasi dalam sistem informasi masih bisa dibilang belum tepat dalam penerapannya, karena informasi sensitif seharusnya hanya dapat dilihat oleh orang dengan otorisasi tertentu.

3. Pertanyaan Penelitian

Dalam penelitian ini terdapat dua pertanyaan penelitian, yaitu:

a. Bagaimana penerapan metode *Feature Flag* agar informasi yang ditampilkan sesuai dengan otorisasi pengguna?
b. Seberapa efektif informasi yang telah diterapkan menggunakan *Feature Flag* tersebut?

C. Maksud dan Tujuan Penelitian

1. Maksud

   Maksud dari penelitian ini adalah menerapkan metode *Feature Flag* dalam proses otorisasi sehingga informasi yang dibutuhkan sesuai dengan otorisasi pengguna.

2. Tujuan
   a. Mengembangkan proses dalam penggunaan otorisasi yang belum tepat menjadi lebih tepat
   b. Mengembangkan prototype Sistem Informasi menggunakan Metode *Feature Flag* untuk proses otorisasi yang lebih efektif, dan juga informasi yang ditampilkan hanya akan tampil sesuai dengan orang yang memiliki otoritas tertentu.

D. Spesifikasi Produk yang diharapkan

Peneliti melakukan pengembangan *prototype* dengan menggunakan Bahasa Pemrograman *PHP (Hypertext Pre Processor)* dengan *Framework Lumen* menggunakan *Database Mysql* dan akan dibuat menggunakan *Javascript* di mana nanti akan berupa Sistem yang dapat mengatur otorisasi user dengan mudah dan cepat dengan adanya *Middleware*.

E. Signifikansi Penelitian

Banyak sistem informasi yang menerapkan otorisasi dalam sebuah fitur dengan menggunakan cara yang paling cepat tanpa memikirkan efek setelah pengembangan nya, dalam penelitian ini proses sistem informasi akan di buat dengan menggunakan metode *Feature Flag* sistem informasi yang menggunakan metode ini dapat melakukan kontrol lebih dalam setiap fitur dan penggunaan nya.

F. Asumsi dan Keterbatasan

1. Asumsi

   Pada penelitian ini diasumsikan bahwa informasi dalam Sistem Informasi dapat diakses sudah sesuai dengan orang yang memilki otorisasi tertentu, dan fitur yang ditampilkan sesuai pengguna yang mengakses Sistem Informasi tersebut, karena pengguna hanya mendapatkan hak akses yang sesuai dengan yang diberikan oleh admin perusahaan yang telah diterapkan kedalam sistem, sehingga informasi yang ditampilkan dapat lebih terkontrol .

2. Keterbatasan Pengembangan

   Dalam penelitian ini memiliki keterbatasan diantara lain:

   a. Pengembangan yang akan dilakukan hanya sebatas prototype yang dibuat ulang berdasarkan Aplikasi Web yang telah ada.
   b. Web Aplikasi yang dibuat tidak terlalu kompleks karena menyangkut konfidensial perusahaan.
   c. Data penelitian yang didapatkan merupakan data bayangan dari data asli yang telah di edit karena menyangkut konfidensial perusahaan.

G. Definisi Istilah dan Definisi Operasional

1. *API (*Application *Programming Interface)*

   merupakan sebuah penghubung yang dapat diimplementasikan menggunakan *software* sehingga beberapa aplikasi yang terhubung dapat berbagi informasi satu sama lain.

2. *Database*

   kumpulan data yang dikelola sedemikian rupa berdasarkan ketentuan tertentu yang saling berhubungan sehingga mudah dalam pengelolaannya. Melalui pengelolaan tersebut pengguna dapat memperoleh kemudahan dalam mencari informasi, menyimpan informasi dan membuang informasi.

3. *JSON (Javascript Object Notation)*

   adalah format yang digunakan untuk menyimpan dan mentransfer data yang terdiri dari sepasang *Key* dan *Value*.

4. *JWT (JSON WEB TOKEN)*

   yang berarti token ini menggunakan JSON (Javascript Object Notation) berbentuk string panjang yang sangat random, lalu token ini memungkinkan kita untuk mengirimkan data yang dapat diverifikasi oleh dua pihak atau lebih.

5. *Feature Flag*

Feature Flag adalah proses pengembangan *software* yang digunakan untuk mengaktifkan atau menonaktifkan fungsionalitas dari tanpa harus merubah *source code* pada sebuah sistem informasi, sehingga fitur tertentu tidak dapat terlihat oleh pengguna. *Feature Flag* juga membantu untuk kontrol penerapan siklus fitur secara penuh.

6. *Middleware*

Middleware adalah perangkat lunak komputer yang memberikan layanan untuk menghubungkan bagian-bagian berbeda dari sebuah sistem yang pada umumnya digunakan dalam sistem terdistribusi untuk memudahkan pengembang perangkat lunak dalam melakukan komunikasi *input/outpu*