

## Pengembangan Sistem Penghubung Layanan (SPL) Berbasis *Service Oriented Architecture* (SOA) dengan Metode *Rapid Application Development* (RAD)

Aditya Darmawan<sup>1</sup>, Aini Suri Talita<sup>2</sup>

Program Studi Perangkat Lunak Sistem Informasi, Fakultas Pascasarjana, Universitas Gunadarma,  
Jl. Margonda Raya No. 100, Depok 16424, Jawa Barat.  
e-mail: <sup>1</sup>adit.dars@gmail.com, <sup>2</sup>ainisuri@staff.gunadarma.ac.id

Submitted Date: August 03<sup>rd</sup>, 2023  
Revised Date: October 05<sup>th</sup>, 2023

Reviewed Date: August 11<sup>th</sup>, 2023  
Accepted Date: October 05<sup>th</sup>, 2023

### Abstract

Ministry of Trade is one of the central government agencies that has many service information systems, but these systems are still separate, making it difficult to share data and information. The lack of integrated service information systems makes it very difficult for the public to access public services because to access one public service, the public has to switch to several service information systems or even have to upload the same documents to several service information systems. To overcome this, this research carried out the development of a Enterprise Service Bus (ESB) using the Rapid Application Development (RAD) method based on Service Oriented Architecture (SOA) to connect services from various service information systems. The result of this research is to produce SPL which is a management solution for exchanging data and information between service information systems that can simplify public service business processes to the public.

Keywords: Service Oriented Architecture; Rapid Application Development; Enterprise Service Bus

### Abstrak

Kementerian Perdagangan merupakan salah satu Instansi Pusat pada pemerintahan memiliki banyak sistem informasi pelayanan namun sistem-sistem tersebut masih terpisah-pisah sehingga menyulitkan berbagi pakai data dan informasi. Tidak terintegrasinya sistem informasi pelayanan sangat menyulitkan bagi masyarakat ketika mengakses layanan publik dikarenakan untuk mengakses satu layanan publik, masyarakat harus berpindah ke beberapa sistem informasi pelayanan atau bahkan harus mengunggah dokumen-dokumen yang sama ke beberapa sistem informasi pelayanan. Untuk mengatasi hal tersebut pada penelitian ini dilakukan pengembangan Sistem Penghubung Layanan (SPL) menggunakan metode Rapid Application Development (RAD) berbasis Service Oriented Architecture (SOA) untuk menghubungkan layanan dari berbagai sistem informasi pelayanan. Hasil penelitian ini adalah menghasilkan SPL yang menjadi solusi manajemen pertukaran data dan informasi antar sistem informasi pelayanan yang dapat mensimplifikasi bisnis proses layanan publik kepada masyarakat.

Kata Kunci: Service Oriented Architecture; Rapid Application Development; Sistem Penghubung Layanan

### 1. Pendahuluan

Teknologi informasi merupakan bagian yang tidak dapat dipisahkan dari kehidupan era digital saat ini. Begitupun pada sektor pemerintahan dengan hadirnya Sistem Pemerintahan Berbasis Elektronik (SPBE). Hal ini seperti yang tertuang pada Peraturan Presiden No. 95 Tahun 2018 tentang Sistem Pemerintahan

Berbasis Elektronik. SPBE merupakan penyelenggaraan pemerintahan yang memanfaatkan teknologi informasi dan komunikasi untuk memberikan layanan kepada pengguna, SPBE ditujukan untuk untuk mewujudkan tata kelola pemerintahan yang bersih, efektif, transparan dan akuntabel serta

pelayanan publik yang berkualitas dan terpercaya.

SPBE mencakup sistem informasi pelayanan pemerintah elektronik, aplikasi seluler, dan berbagai layanan elektronik lainnya, memudahkan bagi masyarakat untuk mengakses dan menelusuri informasi dan layanan publik secara *online*. Dengan adanya sistem ini, masyarakat tidak lagi terbatas oleh hambatan fisik dan geografis yang mungkin menghalangi mereka untuk mendapatkan akses ke pelayanan publik yang mereka butuhkan. Masyarakat dapat mengakses layanan tersebut dari mana saja, kapan saja, tanpa harus mengunjungi kantor pemerintah secara langsung. (Arief dan Abbas, 2021).

Di samping itu, SPBE juga memberikan potensi peningkatan efisiensi dalam penyelenggaraan pelayanan publik. Dengan menggunakan platform elektronik, proses administrasi dapat dilakukan secara *online*, mengurangi birokrasi dan waktu yang diperlukan untuk pengurusan dokumen dan permohonan layanan. Masyarakat dapat mengajukan permohonan, melacak status permohonan, dan menerima pemberitahuan secara *online*, menghemat waktu dan tenaga baik bagi pemerintah maupun masyarakat. (Warman et al., 2022).

Peraturan Presiden No. 95 Tahun 2018 tentang SPBE mengamanatkan Kementerian/Lembaga/Daerah/Instansi untuk menerapkan SPBE, yaitu dengan menyediakan infrastruktur dan sistem informasi instansi yang terpadu dengan Arsitektur SPBE Nasional dalam penyelenggaraan pemerintahan. Salah satu infrastruktur yang perlu dibangun yaitu Sistem Penghubung Layanan (SPL) yang berfungsi sebagai sistem yang menghubungkan berbagai layanan yang ada di sektor pemerintahan untuk dapat berbagi pakai data dan informasi, sehingga peran SPL ini sangat vital dalam meningkatkan layanan publik yang prima berbasis elektronik kepada masyarakat.

Sebagai contoh implementasinya, pemerintah telah mengambil tindakan-tindakan eksekutif strategis sejak awal tahun 2020 sebagai usaha dalam mengatasi situasi krusial menghadapi wabah covid-19. Salah satu cara pemerintah adalah mengintensifkan layanan elektronik dalam manajemen pemerintahan. Hal

tersebut terbukti dapat mengoptimalkan efisiensi pelayanan publik dan meminimalisir persebaran dan penjangkitan Covid-19 pada masyarakat. (Rahmadany, 2021).

Layanan publik Kementerian Perdagangan terdiri dari layanan administrasi pemerintahan dan layanan perizinan khusus di bidang perdagangan. Masyarakat dapat memperoleh layanan publik tersebut dengan mengakses sistem informasi yang dikelola Kementerian Perdagangan seperti layanan perizinan ekspor-impor melalui Sistem INATRADE, layanan surat keterangan asal bagi eksportir melalui Sistem e-SKA, layanan sarana perdagangan melalui Sistem Informasi Sarana Perdagangan (SISP) dan layanan-layanan lainnya. Kehandalan dan ketersediaan sistem-sistem tersebut tentunya menjadi hal yang sangat penting, karena apabila sistem tersebut terganggu atau *down* bahkan dalam waktu yang tidak lama akan mengakibatkan kerugian bagi masyarakat khususnya pelaku usaha ekspor/impor yang pada akhirnya akan mempengaruhi kegiatan usahanya dan meningkatkan biaya produksi penyediaan barang dan jasa yang akan diedarkan di Indonesia maupun diekspor ke luar negeri. Hal ini tentunya akan sangat berdampak pada perekonomian Indonesia.

Selain itu, tidak terintegrasinya sistem informasi pelayanan pemerintahan juga sangat menyulitkan bagi masyarakat dalam mengakses layanan publik dikarenakan untuk mengakses satu layanan publik, masyarakat harus berpindah ke beberapa aplikasi atau bahkan harus mengunggah dokumen-dokumen yang sama ke beberapa aplikasi pemerintah. Sebagai contoh apabila pelaku usaha ingin memperoleh perizinan importasi dan izin distributor, pelaku usaha tersebut masih harus mengunggah dokumen nomor induk berusaha (NIB) di setiap sistem informasi perizinan di Kementerian Perdagangan, padahal dokumen NIB tersebut telah tersedia pada *Sistem Online Single Submission* (OSS) yang dikelola Kementerian Investasi/BKPM. Untuk mengatasi hal tersebut, maka diperlukannya SPL sebagai *middleware* yang dapat mengelola semua aspek yang berkaitan dengan lalu lintas, analisis, monitoring dan keamanan pertukaran data dan informasi baik lingkup internal maupun eksternal Kementerian Perdagangan.

SPL adalah perangkat integrasi/penghubung untuk melakukan pertukaran layanan SPBE (Perpres 95, 2018). Pemanfaatan SPL bertujuan untuk memudahkan dalam melakukan integrasi antar layanan SPBE. Dalam perpektif bisnis, *Service Oriented Architecture* (SOA) merupakan satu set layanan yang digunakan untuk meningkatkan kemampuan organisasi atau perusahaan untuk melakukan bisnis dengan konsumen. Dari sudut pandang teknologi SOA ditandai dengan adanya modularitas yaitu penggunaan kembali layanan, metode pemrograman baru berdasarkan standar, tools yang melibatkan layanan web. (Hizviani, 2020). Dengan memanfaatkan konsep SOA suatu sistem informasi dipandang dari layanan-layanan yang disediakan. Metode pengembangan perangkat lunak *Rapid Application Development* (RAD) adalah model proses pengembangan perangkat lunak yang bersifat incremental terutama untuk waktu pengerjaan yang pendek. (Wulandari dan Siti, 2022). Dari definisi konsep RAD ini, dapat dilihat bahwa pengembangan aplikasi dengan menggunakan metode RAD dapat dilakukan dalam waktu yang relatif lebih cepat.

Penelitian ini menggunakan *Web Server* NGINX. NGINX menjadi pilihan berkat kinerja dan stabilitasnya yang tinggi. Kinerja *web server* dengan menggunakan NGINX memiliki performansi lebih baik dibandingkan dengan APACHE, hal ini dibuktikan dengan pengukuran kinerja *web server* NGINX dan APACHE dengan menggunakan IPv6 dengan parameter *Time Taken for Tests*, *Request Per Second*, *Transfer Rate* (Kb/s), *Time per Request* (ms), *Memory Usage*, dan *Load Maximum*. (Satwika dan Semadi, 2020).

## 2. Penelitian Terdahulu

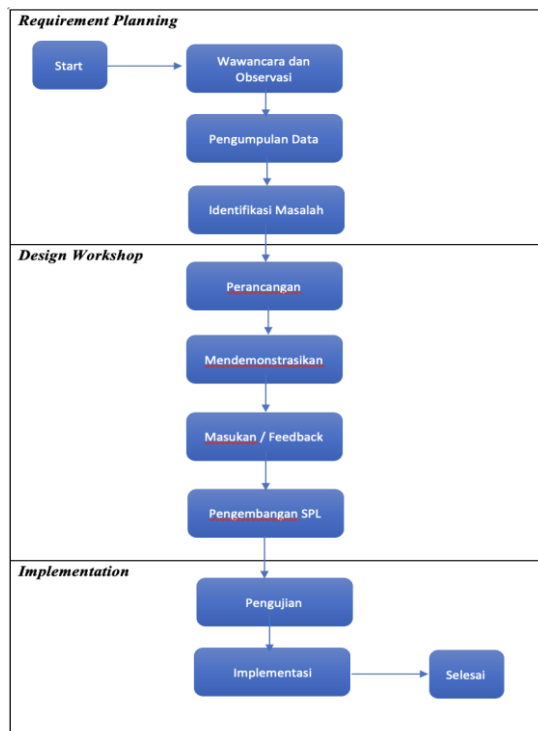
Utama et al. tahun 2019 melakukan penelitian "*E-Government Integration of Sidoarjo Regency using Service Oriented Architecture (SOA)*", menyebutkan bahwa pembangunan aplikasi *e-Government* menggunakan SOA, proses dapat menjadi lebih cepat karena arsitektur ini membuat aplikasi ke dalam bentuk *web service* sehingga bisa terhubung antara satu layanan dengan lainnya (Utama et al., 2019).

Deni Murdiani dan Muhamad Sobirin tahun 2022 melakukan penelitian "Perbandingan Metodologi *Waterfall* dan RAD (*Rapid Application Development*) dalam Pengembangan Sistem Informasi", menyebutkan bahwa metode RAD dapat digunakan untuk pengembangan sistem atau perangkat lunak yang berskala besar dan memerlukan waktu lebih singkat, di mana software dibuat berdasarkan permintaan dan kebutuhan tertentu (Murdiani dan Sobirin, 2022).

Kyosti dan Lindstrom tahun 2022 melakukan penelitian "*SOA-Based Platform Use in Development and Operation of Automation Solutions: Challenges, Opportunities, and Supporting Pillars towards Emerging Trends*", menyebutkan bahwa platform. SOA mendukung teknologi *microservices* yang menjadi tren di organisasi dan perusahaan yang terlibat dalam pengembangan dan pengoperasian solusi otomatis (Kyosti dan Lindstrom, 2022).

## 3. Metodologi

Penelitian ini terdiri dari 3 (tiga) tahap, yaitu fase perencanaan persyaratan (*requirement planning*) meliputi wawancara, observasi, studi literatur dan pengumpulan data primer dan sekunder serta identifikasi masalah. Tahapan selanjutnya yaitu fase perancangan sistem (*design workshop*) meliputi perancangan, mendemonstrasikan kepada pengguna, masukan/*feedback* dari pengguna, penyaringan/*filtering* dilanjutkan pembuatan kode program SPL. Selanjutnya pada tahapan terakhir yaitu fase implementasi, sistem diuji dengan menggunakan metode *blackbox testing* dilanjutkan dengan implementasi sistem baru, tahapan penelitian ditunjukkan pada Gambar 1



Gambar 1. Tahap Pengembangan SPL

a. Fase Perencanaan Persyaratan (*Requirement Planning*)

Fase ini dilakukan pengumpulan data, wawancara dan observasi dilanjutkan dengan analisa dan pengelompokan layanan-layanan yang ada pada Kementerian Perdagangan dilanjutkan mengumpulkan *log* pertukaran data antar sistem. Wawancara dan observasi dilakukan langsung kepada pemangku kepentingan terkait sistem informasi di internal dan eksternal Kementerian Perdagangan.

b. Fase Perancangan Sistem (*Design Workshop*)

Proses pada fase ini yaitu analis bekerja sama dengan pengguna sistem untuk melakukan *workshop* mengenai perancangan sistem yang akan dilakukan. Kegiatan yang dilakukan seperti analisis kebutuhan sistem, pembuatan alur kerja sistem, perancangan tampilan sistem dan mendemonstrasikan rancangan serta pembuatan kode program sistem. Berikut ini beberapa hal-hal yang dilakukan pada fase ini yaitu analisis standar dan kebutuhan sistem, desain rancangan alur kerja sistem, desain rancangan visual sistem, penyempurnaan rancangan kode program dan visualisasi.

c. Fase Penerapan (*Implementation*)

Pada fase ini dilakukan tahap pengujian fungsi dan implementasi sistem. Pengujian sistem

menggunakan metode *blackbox testing*. Pengujian ini merupakan pengujian fungsionalitas dari seluruh fitur sistem dilanjutkan dengan tahap implementasi SPL sebagai *middleware* yang menjadi sistem pertukaran layanan.

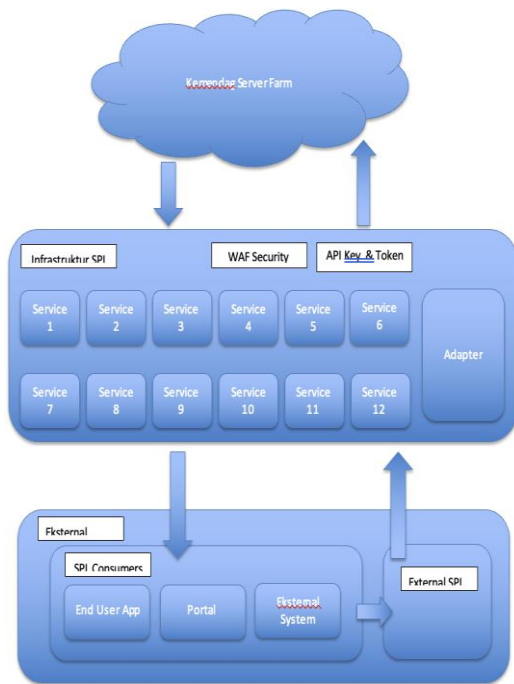
#### 4. Hasil dan Pembahasan

Pada fase pertama yaitu perencanaan persyaratan dilakukan pengumpulan data, wawancara dan observasi kepada beberapa orang pejabat dan pegawai dari Unit Sekretariat Jenderal dan Unit-Unit Teknis di internal Kementerian Perdagangan. Untuk mengetahui layanan yang ada pada Kementerian Perdagangan, tahap pertama yaitu melihat gambaran tugas pokok dan fungsi Kementerian Perdagangan, hal ini dapat diperoleh dari proses bisnis Kementerian Perdagangan secara keseluruhan. Selain peta proses bisnis, peneliti juga memperoleh informasi tentang klasifikasi data perdagangan dan informasi perdagangan serta *road map* pengembangan Sistem Informasi Perdagangan (SIP). SIP merupakan sistem informasi yang akan menjadi sistem informasi utama bidang perdagangan. Wawancara dan observasi dilakukan kepada beberapa orang pejabat dan pegawai, pengembang sistem di internal Kementerian Perdagangan dan eksternal yaitu Kementerian Pendayagunaan Aparatur Negara dan Reformasi Birokrasi (Kemendagri), Kementerian Komunikasi dan Informatika, Kementerian Investasi/BKPM, Kementerian Perindustrian, Kementerian Investasi/Badan Koordinasi Penanaman Modal (BKPM) dan Lembaga Nasional Single Window (LNSW) untuk memperoleh data dan informasi kebutuhan pertukaran data dalam rangka implementasi layanan perizinan berbasis resiko dan layanan non perizinan.

Berdasarkan analisis dari data-data yang telah dikumpulkan pada fase perencanaan persyaratan, maka pada perancangan sistem dapat dikelompokkan dan didefinisikan 12 (dua belas) kelompok layanan/services yang akan dikembangkan pada SPL ini yaitu Layanan *Single Sign-On*, Layanan Kepegawaian, Layanan Data Pelaku Usaha, Layanan Sistem Perizinan, Layanan Non Perizinan, Layanan Sistem Keuangan, Layanan Sistem Pelaporan Harga, Layanan Surveyor, Layanan *E-Payment*,

Layanan Sistem Pengawasan, Layanan Hukum dan Perundang-Undangan, Layanan *Digital Signature*. Selain itu pada fase ini juga ditetapkan fitur-fitur yang akan dikembangkan di dalam SPL yaitu *Dashboard*, *User Managemen*, *API Management*, *API List*, *API Client*, *API Access*, *API Credential*, *Access Limit* dan *Tools Monitoring*.

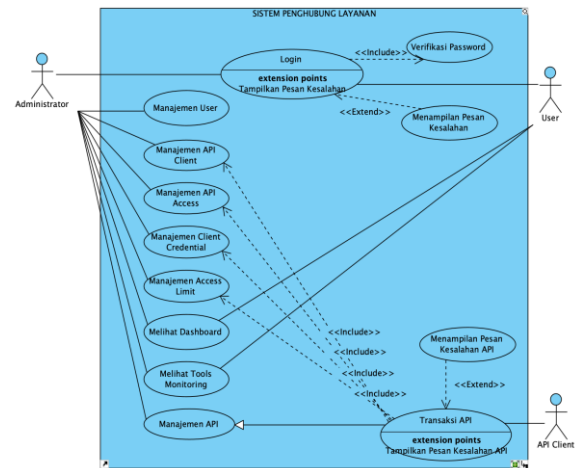
Tahapan selanjutnya yaitu dilakukan perancangan arsitektur SPL, infrastruktur SPL terdiri dari satu bus yang di dalamnya terdapat seluruh kelompok layanan/*services* dalam bentuk *API services* yang dapat berkomunikasi dengan sistem-sistem yang ada internal maupun eksternal. Untuk menjamin keamanan pertukaran data dan informasi, *API services* tersebut dilengkapi dengan *API Key* dan *Token*. Selain itu bus tersebut juga mengimplementasikan *Web Application Firewall (WAF)*. Rancangan Arsitektur SPL ditunjukkan pada Gambar 2.



Gambar 2. Rancangan Arsitektur SPL

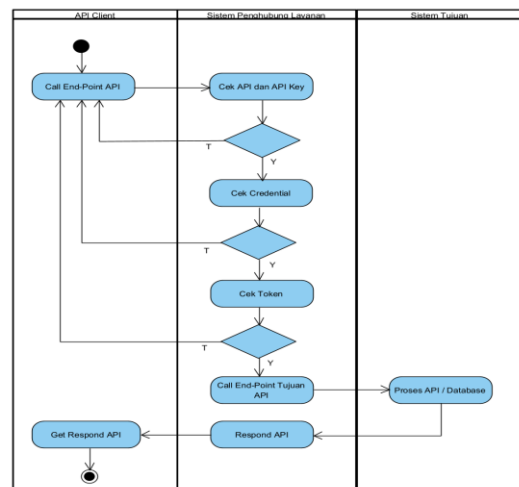
Tahapan selanjutnya yaitu melakukan desain alur aktor pada Sistem dengan menggunakan *Use Case Diagram*. SPL terdiri dari 3 (tiga) aktor yaitu *Administrator*, *User* dan *API Client*. Aktor *Administrator* merupakan pengelola SPL yang dapat mengelola dan mengakses seluruh fitur dalam sistem, Aktor *User* merupakan pengguna sistem yang dapat

melakukan monitoring sistem yang terdiri dari *Dashboard* dan *Tools Monitoring*, yang terakhir yaitu Aktor *API Client* merupakan Aktor/sistem lain yang memanfaatkan API pada SPL ini. *Use Case* seluruh aktor ditunjukkan pada Gambar 3.



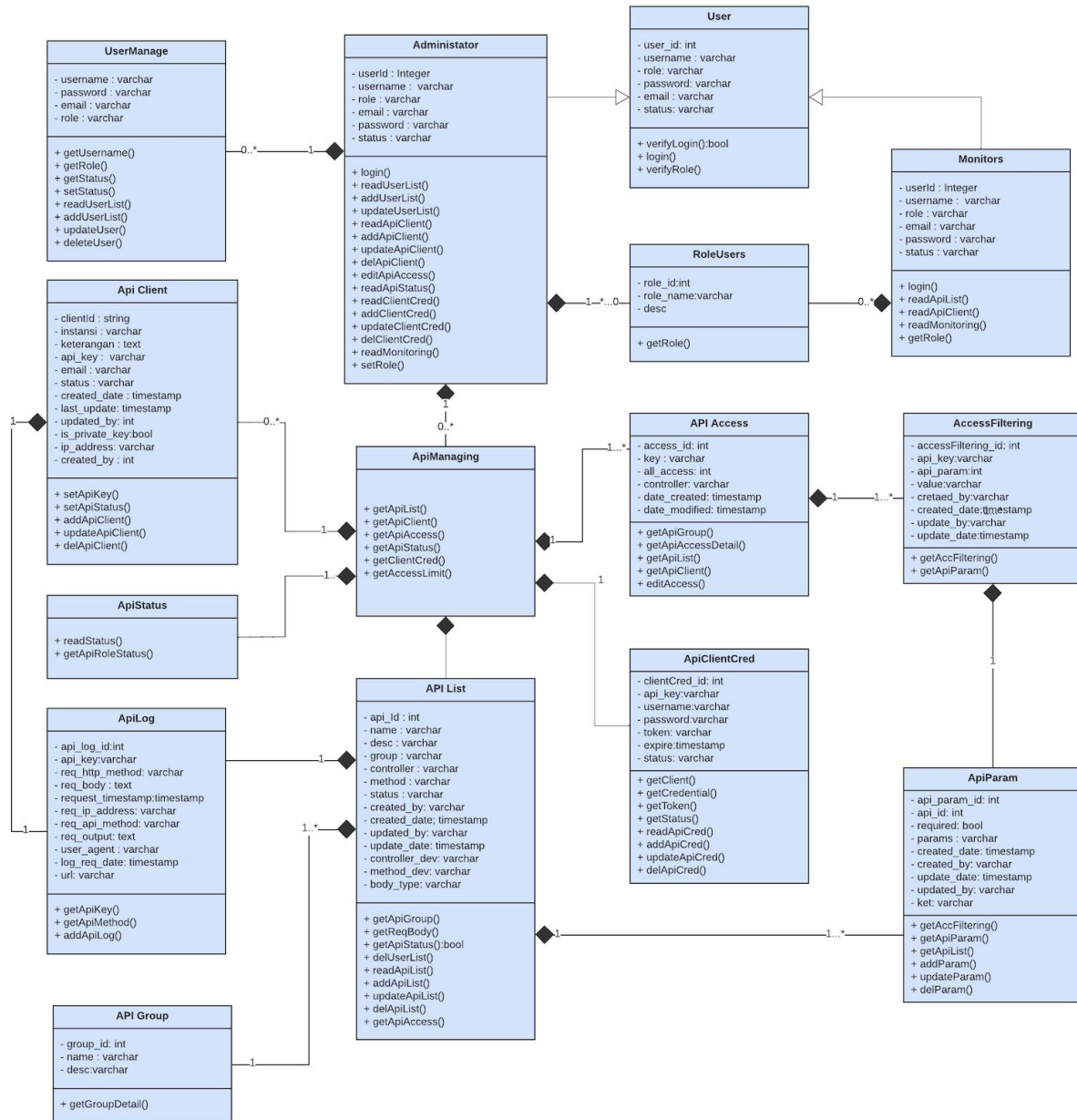
Setelah dilakukan desain alur aktor, tahapan selanjutnya yaitu melakukan desain aktivitas seluruh aktor dengan menggunakan *Activity Diagram*, hal yang cukup penting dalam sistem ini yaitu alur transaksi API, alur transaksi API dimulai dari *API client* melakukan *call end-point* API pada SPL yang dilanjutkan dengan pengecekan *API key*, *credential* dan *token* serta API tujuan, apabila hasil pengecekan tersebut sesuai maka dilanjutkan dengan *call end-point* tujuan API dan merespon *API request* dengan mengirimkan data kepada *API client*. Alur transaksi API dapat dilihat pada Tabel 1.

Tabel 1. Alur Transaksi API



Proses perancangan model data SPL dilakukan dengan menggunakan *Class Diagram*. Pemodelan data SPL terdiri 8 (delapan) tabel yaitu tabel *user*, *client*, *client\_cred*, *access*,

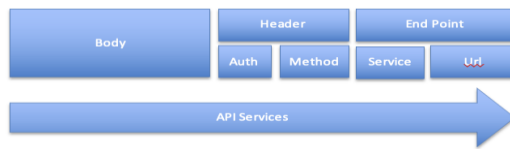
*access filtering*, *api*, *api\_params* dan *api\_log*. Seluruh tabel tersebut digunakan untuk seluruh proses dalam sistem. Pemodelan data SPL ditunjukkan pada Gambar 4.



Gambar 4. Class Diagram SPL

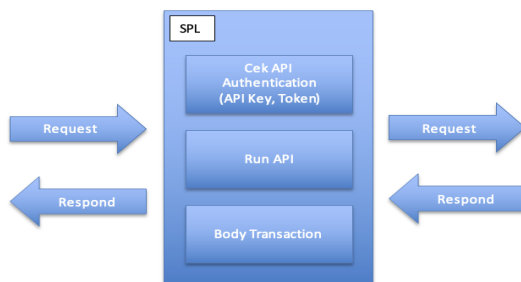
Pada fase perancangan sistem dilakukan juga desain struktur API yang akan digunakan dalam proses pertukaran data dan informasi. Struktur API merupakan standar API yang digunakan dalam transaksi API untuk melakukan pertukaran data dan informasi. Struktur API terdiri dari 3 (tiga) bagian utama yaitu *end-point*,

*header* dan *body*. *End-point* merupakan url dan API yang akan diakses, *header* merupakan API elemen autentikasi dan *method* yang digunakan sedangkan *body* merupakan data dan informasi yang dipertukarkan. Berikut ini adalah desain struktur API yang akan digunakan ditunjukkan pada Gambar 5.



Gambar 5. Struktur API

Proses selanjutnya dilakukan desain alur transaksi API yang menggambarkan proses yang dilakukan sistem dalam melayani suatu transaksi mulai dari proses autentifikasi hingga proses respon, alur transaksi API ditunjukkan pada Gambar 6.



Gambar 6. Alur Transaksi API

Langkah selanjutnya pada fase perancangan sistem yaitu melakukan perancangan tampilan pada sistem dilanjutkan dengan melakukan pembuatan kode program. Setelah seluruh rangkaian pada fase perencanaan persyaratan dan perencanaan sistem selesai tahapan selanjutnya yaitu dengan fase penerapan.

Tahap pertama pada fase penerapan, sebelum sistem digunakan oleh user, sistem harus dilakukan pengujian terlebih dahulu untuk memastikan spesifikasi, fitur-fitur dan fungsi telah sesuai dengan rancangan. Pengujian SPL ini menggunakan metode *blackbox testing*. Pengujian *blackbox* merupakan pengujian yang dilakukan untuk mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari sistem. Tabel 2. menunjukkan hasil pengujian sistem.

Tabel 2. Hasil Pengujian SPL

No	Aktifitas Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
<b>Halaman Login</b>			
1	Input <i>username/ password</i> , dilanjutkan klik tombol <i>Sign In</i>	Masuk ke halaman utama	Sesuai
<b>Halaman Homepage</b>			
2	Klik Dashboard pada menu di sisi kiri	Masuk ke halaman Dashboard	Sesuai
3	Klik User pada menu di sisi kiri	Masuk ke halaman User	Sesuai
4	Klik Client Manage pada menu di sisi kiri	Keluar tampilan menu : API List, API Client, API Access, Client Cred dan Access Limit	Sesuai
5	Klik Monitoing pada menu di sisi kiri	Keluar tampilan menu tools monitoring berdasarkan layanan	Sesuai
<b>Halaman Client Manage</b>			
6	Klik Client Manage : API List pada menu di sisi kiri	Masuk ke halaman API List	Sesuai
7	Klik Client Manage : API Client pada menu di sisi kiri	Masuk ke halaman API Client	Sesuai
8	Klik Client Manage : API Access pada menu di sisi kiri	Masuk ke halaman API Access	Sesuai
9	Klik Client Manage : Client Creds pada menu di sisi kiri	Masuk ke halaman Client Creds	Sesuai
10	Klik Client Manage : Acces Limit pada menu di sisi kiri	Masuk ke halaman Access Limit	Sesuai

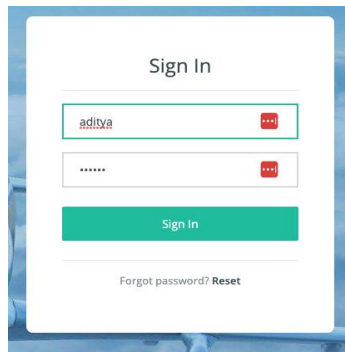
No	Aktifitas Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
<b>Halaman API List</b>			
11	Klik Tambah API	Masuk ke halaman Tambah API	Sesuai
12	Klik Edit API	Masuk ke halaman Edit API	Sesuai
13	Klik Tambah/Edit Parameter API	Masuk ke halaman Tambah/Edit Parameter	Sesuai
14	Klik Aktif/Non Aktif API	Aktifkan / Non Aktifkan API	Sesuai
15	Klik Delete API	Masuk ke halaman Delete API	Sesuai
<b>Halaman API Client</b>			
16	Klik Tambah API Client	Masuk ke halaman Tambah API Client	Sesuai
17	Klik Edit API Client	Masuk ke halaman Edit API Client	Sesuai
18	Klik Aktif/Non Aktif API Client	Aktifkan / Non Aktifkan API Client	Sesuai
19	Klik Delete API Client	Masuk ke halaman Delete API Client	Sesuai
<b>Halaman API Access</b>			
20	Pilih API Client, melakukan konfigurasi privilege sesuai layanan	Konfigurasi privilege API Client berjalan sesuai layanan	Sesuai
<b>Halaman Client Creds</b>			
21	Klik Tambah Client Credential	Masuk ke halaman Tambah Client Credential	Sesuai
22	Klik Edit Client Credential	Masuk ke halaman Edit Client Credential	Sesuai
23	Klik Aktif/Non Aktif Client Credential	Aktifkan / Non Aktifkan Client Credential	Sesuai
24	Klik Delete Client Credential	Masuk ke halaman Delete Client Credential	Sesuai
<b>Halaman Access Limit</b>			
25	Klik Set Limit	Masuk ke halaman Set Limit Parameter	Sesuai
<b>Halaman Monitoring</b>			
26	Klik layanan pada menu monitoring	Muncul log API sesuai dengan layanan	Sesuai
<b>Pengujian Pertukaran Data</b>			
27	Akses API menggunakan layanan (request body) dengan API key yang sesuai	Mendapatkan respon request output dari API	Sesuai
<b>Pengujian Keamanan Sistem</b>			
28	Akses API menggunakan layanan (request body) dengan API key yang tidak sesuai	Mendapatkan respon penolakan dari API	Sesuai

SPL ini terdiri dari beberapa tampilan halaman diantaranya adalah sebagai berikut.

1. Halaman *Login*  
Halaman *login* dirancang untuk *user* untuk mengakses sistem dengan menginput



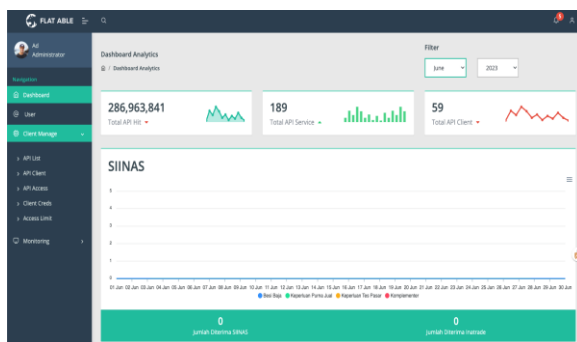
username dan password ditunjukkan pada Gambar 7.



Gambar 7. Tangkapan Layar Halaman Login

## 2. Halaman Homepage

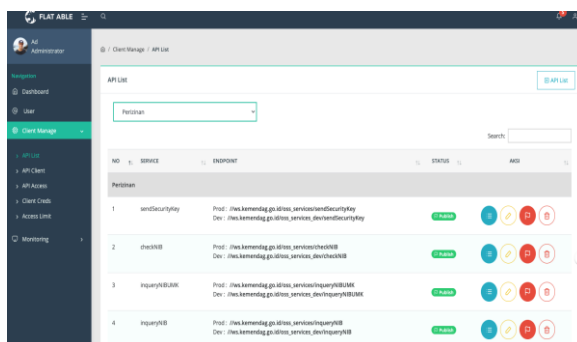
Halaman homepage akan menampilkan halaman utama user setelah user berhasil melakukan login ditunjukkan pada Gambar 8.



Gambar 8. Tangkapan Layar Halaman Homepage

## 3. Halaman API List

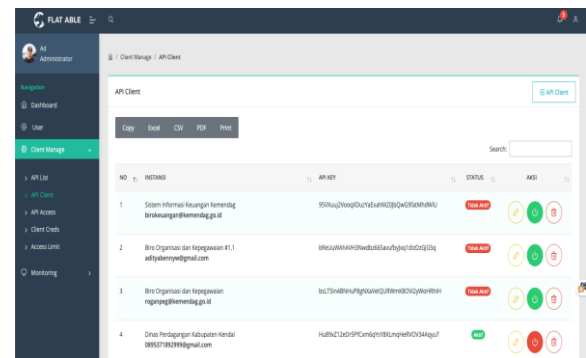
Halaman API List akan menampilkan, menambah, mengedit, mengaktifkan dan menghapus API yang ada di dalam sistem ditunjukkan pada Gambar 9.



Gambar 9. Tangkapan Layar Halaman API List

## 4. Halaman API Client

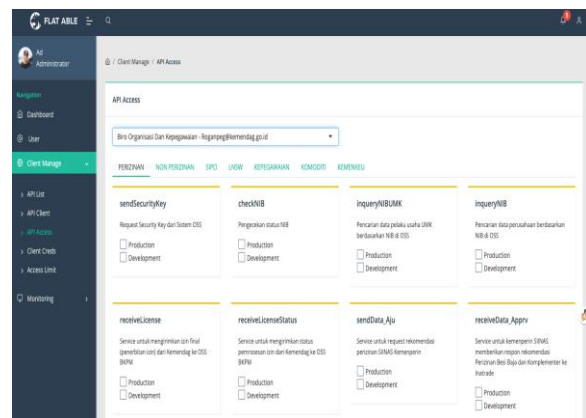
Halaman API Client akan menampilkan, menambah, mengedit, mengaktifkan dan menghapus API Client yang ada di dalam sistem ditunjukkan pada Gambar 10.



Gambar 10. Tangkapan Layar Halaman API Client

## 5. Halaman API Access

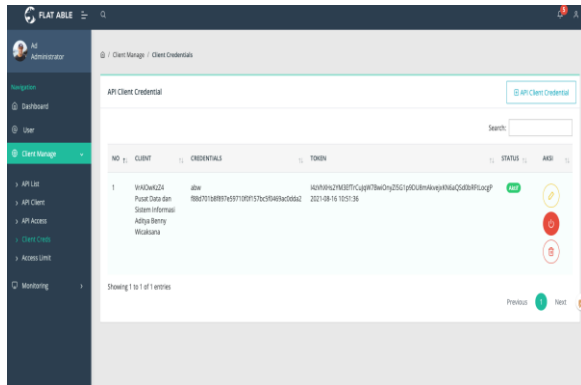
Halaman API Access digunakan untuk melakukan konfigurasi privilege API Client terhadap layanan yang diberikan ditunjukkan pada Gambar 11.



Gambar 11. Tangkapan Layar Halaman API Access

## 6. Halaman API Credential

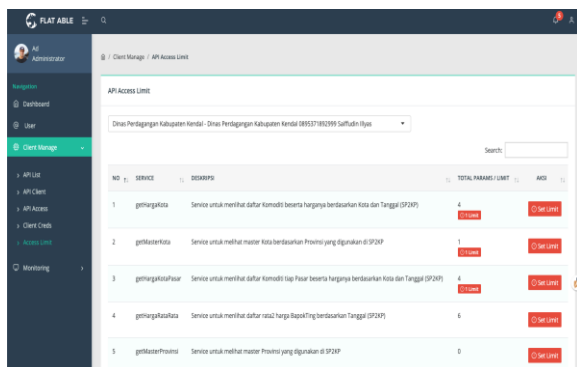
Halaman API Credential akan menampilkan, menambah, mengedit, mengaktifkan dan menghapus API Client Credential yang ada di dalam sistem ditunjukkan pada Gambar 12.



Gambar 12. Tangkapan Layar Halaman API Credential

#### 7. Halaman API Access Limit

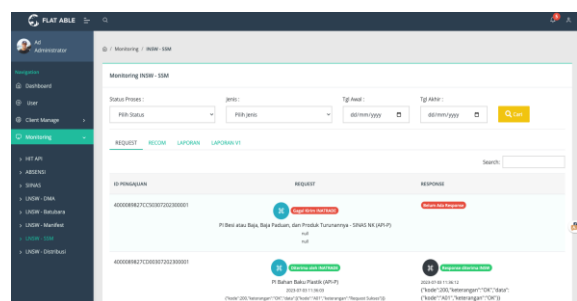
Halaman API Access Limit digunakan untuk melakukan konfigurasi *priviledge* API Client dalam mengakses parameter pada sebuah API yang ada di dalam sistem ditunjukkan pada Gambar 13.



Gambar 13. Tangkapan Layar Halaman API Access Limit

#### 8. Halaman Tools Monitoring

Halaman Tools Monitoring digunakan untuk menampilkan seluruh tools untuk memonitor ditunjukkan pada Gambar 14.



Gambar 14. Tangkapan Layar Halaman Tools Monitoring

Langkah kedua fase penerapan yaitu melakukan implementasi SPL sebagai middleware yang menjadi sistem pertukaran layanan di lingkungan Kementerian Perdagangan, beberapa hal yang harus menjadi perhatian dalam implementasi sistem ini yaitu :

1. Spesifikasi perangkat keras server yang digunakan harus sesuai dengan perencanaan sebelumnya.
2. Bandwidth jaringan lokal maupun internet yang digunakan harus memadai dan cukup untuk melakukan transaksi pertukaran data secara realtime.
3. Penyediaan server untuk web application dan database pada arsitektur SPL sebaiknya terpisah untuk memaksimalkan kapasitas layanan sistem.
4. Selama proses implementasi SPL harus didukung sumber daya manusia yang paham akan bisnis proses setiap layanan yang disediakan sistem.

#### 5. Kesimpulan

Pengembangan Sistem Penghubung Layanan (SPL) pada Kementerian Perdagangan telah selesai, beberapa hal yang dapat diambil kesimpulan yaitu konsep *Service Oriented Architecture* (SOA) memudahkan untuk memisahkan seluruh layanan menjadi layanan-layanan kecil sehingga memudahkan pertukaran layanan antar sistem informasi pelayanan. Dengan menggunakan metode *Rapid Application Development* (RAD) waktu pengembangan SPL dapat dilakukan dengan waktu yang lebih cepat. Hasil pengujian SPL dengan metode *blackbox testing* menunjukkan bahwa rancangan seluruh fitur berjalan dengan baik, hal ini dibuktikan juga dengan aliran data dan informasi yang melalui SPL ini dapat mengalir dengan baik.

Pemanfaatan SPL dapat menjadi solusi manajemen pertukaran data dan informasi, mensimplifikasi proses bisnis layanan yang pada akhirnya akan meningkatkan kualitas layanan publik Kementerian Perdagangan.

#### 6. Saran

Saran untuk penelitian pengembangan SPL selanjutnya yaitu perlunya penambahan fitur monitoring API yang dapat memberikan notifikasi status layanan ke aplikasi pesan singkat pada *mobile device* sehingga apabila terjadi

kegagalan pada suatu API yang mengakibatkan terganggunya suatu layanan, maka hal tersebut dapat langsung diketahui admin SPL dan dapat segera ditindaklanjuti untuk perbaikan.

Wedding. *Jurnal Rekayasa Informasi*, 11(1), 79-85

## References

- Arief, A. & Abbas, M. Y. (2021). Kajian Literatur (Systematic Literature Review): Kendala Penerapan Sistem Pemerintahan Berbasis Elektronik (SPBE). *PROtek : Jurnal Ilmiah Teknik Elektro*, (8)1, 1-6
- Hizviani, N. V. (2020), Teknologi Service Oriented Architecture sebagai jalur Komunikasi Data untuk penghubung Sistem lain dengan Aplikasi Datawarehouse KTP el. *Jurnal Esensi Infokom*, (4)1, 25-28
- Kyosti P. & Lindstrom J. (2022). SOA-Based Platform Use in Development and Operation of Automation Solutions: Challenges, Opportunities, and Supporting Pillars towards Emerging Trends, *MDPI Appl. Sci.*, (12)1074, 1-17
- Murdiani D. & Sobirin M. (2022). Perbandingan Metodologi Waterfall dan RAD (Rapid Application Development) dalam Pengembangan Sistem Informasi. *Jurnal Teknik Informatika*, (10)2, 95-104
- Peraturan Presiden. (2018). Peraturan Presiden Nomor 95 Tahun 2018 tentang Sistem Pemerintahan Berbasis Elektronik
- Utama, A. P., Asmara R. & Hasim, J. A. N. (2019), E-Government Integration of Sidoarjo Regency using Service Oriented Architecture (SOA). *International Journal of New Media Technology*, (6)2, 109-115
- Rahmadany, A. F. (2021). Implementasi Pelayanan Publik berbasis Electronic Government dalam Menghadapi Pandemi Covid-19 di Indonesia. *Jurnal Ilmiah Administrasi Pemerintahan Daerah*, (13)1, 118-125
- Satwika, I. K. S. & Semadi, K. N. (2020). Perbandingan Performansi Web Server Apache dan Nginx Dengan Menggunakan Ipv6. *Jurnal Teknologi Informasi dan Komunikasi*, (15)1, 10-15
- Warman, N. S., Syamsir, Maldini M., Nurhasanah, O., Oktariandani, N. R., & Syafikruzi, I. H. (2022). Implementasi Inovasi Kebijakan Dalam Penyelenggaraan Sistem Pemerintahan Berbasis Elektronik (SPBE) Di Kota Pekanbaru. *Prosiding Seminar Nasional Pendidikan, Bahasa, Sastra, Seni, Dan Budaya*, (1)2, 132– 148
- Wulandari, T. & Siti N. (2022), Rancang Bangun Sistem Pemesanan Wedding Organizer Menggunakan Metode RAD di Shofia Ahmad