

# Integrasi Sistem Manajemen Tata Usaha Dengan Metode *Service Oriented Architecture* (SOA) Pada SMP Negeri 10 Mataram

*(Integration of Administrative Management Systems with Service Oriented Architecture (SOA) Methods in SMP Negeri 10 Mataram)*

Jaka Bagus Wijanarko, Royana Afwani, Moh. Ali Albar\*

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Mataram  
Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA

Email: jakabagus65@gmail.com, royana@unram.ac.id, mohalialbar@unram.ac.id

\*Penulis korespondensi

**Abstract** - The business processes continue to change following the needs of the organization, so the development of existing systems and applications is needed. For more complex cases they need to be integrated with other systems and applications to deliver fast and quality educational processes and activities. SMP Negeri 10 Mataram has several problems, namely: to date, in every administrative section, all business processes or job desks are still done manually, so that they are considered inefficient, especially in terms of time and there is no data linkage between one section with other parts. To overcome these problems, the integration of the administrative management system with the Service Oriented Architecture (SOA) method. Because SOA is one way to design data integration from several different systems. SOA is a software development technology architecture with a service approach, allowing the relationship and exchange of data or information between parts to be easy. The existence of SOA that has been built at SMP Negeri 10 Mataram using a web service obtained an integrated information system architecture in each part of the administration, making it easier for administrative unit employees to exchange data.

**Key words:** Service Oriented Architecture (SOA), Web Service, SMP Negeri 10 Mataram, Integration, Administration Management.

## I. PENDAHULUAN

Pengelolaan data dan informasi adalah hal penting pada organisasi, khususnya organisasi pendidikan. Proses bisnis mengalami perubahan secara terus menerus dengan diikuti kebutuhan pada organisasi, sehingga sistem dan aplikasi perlunya dikembangkan. Pengembangan ini perlu pengintegrasian dengan sistem dan aplikasi yang lain sehingga dapat menciptakan proses dan aktifitas pelayanan yang cepat, berkualitas, dan berdaya saing. Pengembangan ini tergantung berdasarkan kompleksnya permasalahan yang ada. Integrasi sistem adalah penyatuan sistem yang ada dan teknologi baru untuk membentuk sistem yang lebih mampu yang dimaksud untuk mengambil tugas-tugas tambahan, menunjang peningkatan kinerja, dan atau meningkatkan sistem yang ada.

SMP Negeri 10 Mataram terletak di Kecamatan Ampenan Kota Mataram. Sekolah ini letaknya di pinggiran kota Mataram, namun sekolah ini menjadi salah satu sekolah menengah pertama yang memiliki cukup banyak peminat, diantaranya yang berasal dari Kecamatan Ampenan dan sekitarnya serta sebagian dari Kabupaten Lombok Barat. Berdasarkan observasi yang dilakukan di SMP Negeri 10 Mataram dan wawancara kepada pegawai tata usahanya, bahwa SMP Negeri 10 Mataram sebagai sebuah lembaga pendidikan memiliki beberapa permasalahan di setiap bagian pada tata usaha. Sampai saat ini, disetiap bagian-bagian tata usaha melakukan semua proses bisnis atau jobdesk nya masing-masing bagian masih dilakukan secara manual, sehingga di nilai kurang efisien khususnya dalam hal waktu. Selain itu permasalahan yang muncul juga misalnya pada saat bagian keuangan membutuhkan data dari bagian kesiswaan, atau melakukan pertukaran data harus melakukan pencatatan ulang data yang dibutuhkannya, atau meminta data ke bagian tata usaha lainnya yang memiliki data tersebut. Sehingga dibutuhkannya sebuah komunikasi antar sistem bagian tata usaha yang satu dengan bagian tata usaha lainnya.

*Service Oriented Architecture* (SOA) adalah salah satu cara atau solusi untuk merancang integrasi data dari beberapa sistem yang berbeda. SOA merupakan teknologi berbasis arsitektur dalam suatu pengembangan *software*. SOA dilakukan berdasarkan pendekatan layanan, sehingga memudahkan adanya *link* dan pertukaran data dan informasi antar bagian satu dengan bagian lainnya. Selain itu *service* yang ada pada SOA bersifat *reusable*, yang artinya *service* tersebut dapat digunakan pada beberapa fitur aplikasi yang berbeda dan dapat digunakan secara *multiplatform* [1]. Sementara itu, *web services* adalah merupakan *method* yang ada pada sebuah *server* dan dapat dipanggil oleh *clien*. Implementasi *web services* di SOA adalah sangat tepat. Hal ini, karena *web service* dapat mewakili layanan atau *service* dalam SOA. Oleh karena itu, konsep dan sifat-sifat SOA sangat cocok diterapkan pada *web services* [1].

Berdasarkan permasalahan pada bagian tata usaha SMP Negeri 10 Mataram, SOA merupakan solusi untuk pengembangan sistem antarmuka yang dapat di akses banyak *software*, karena data-data itu digunakan dan dihasilkan pada bagian tata usaha juga dibutuhkan dan dapat digunakan pada *sub* organisasi lain di SMPN Negeri 10 Mataram. SOA juga memiliki keunggulan yaitu kemampuan integrasi yang sudah terkelola. Selain itu, SOA unggul pada mudahnya melakukan tukar informasi pada beberapa aplikasi. SOA juga dapat memperkecil pengaruh ketika terjadi perubahan, dan dapat mengurangi biaya pengeluaran untuk integrasi dan modifikasi aplikasi bagi SMP Negeri 10 Mataram untuk jangka panjang. Oleh karena itu maka dibutuhkan perancangan dan pembangunan *web service* yang dapat melakukan integrasi data disetiap bagian-bagian tata usaha.

Berdasarkan penjelasan di atas, maka fokus penelitian ini adalah untuk merancang dan membangun sebuah integrasi sistem manajemen tata usaha dengan metode SOA pada SMP Negeri 10 Mataram sehingga didapatkan arsitektur dan *services* yang digunakan untuk pengembangan sistem secara berkesinambungan jika dibutuhkan dan memudahkan masing-masing bagian tata usaha untuk melakukan pertukaran data serta penyajian data sesuai keperluan.

## II. TINJAUAN PUSTAKA DAN LANDASAN TEORI

### A. Tinjauan Pustaka

Penelitian pertama dilakukan oleh Kris Iswanto pada tahun 2013 [2]. Dalam penelitian ini, peneliti merancang sebuah aplikasi web dengan memanfaatkan teknologi *Service Oriented Architecture* (SOA). Adanya teknologi ini memungkinkan komunikasi antar satu aplikasi dengan aplikasi yang lain. Perancangan aplikasi ini dibangun untuk memberi kemudahan pada bagian tata usaha dalam melakukan pembayaran maupun penyebaran data siswa serta pengecekan status siswa saat siswa lulus atau akan pindah sekolah. Peneliti menggunakan teknologi SOA karena sekolah memiliki beberapa permasalahan yaitu sampai saat ini baik bagian ketatausahaan, kesiswaan, maupun perpustakaan masih melakukan semua proses administrasi secara manual, sehingga dinilai kurang efisien khususnya dalam hal waktu. Karena masih dilakukan secara manual dalam penyediaan serta pertukaran data dan informasi yang diperlukan setiap unit menjadi sangat sulit, kesulitan yang muncul antara lain: 1) dalam hal penyebaran data siswa setiap unit pendukung sekolah harus melakukan pencatatan ulang data siswa yang dimiliki oleh bagian ketatausahaan, 2) dalam perolehan informasi tagihan siswa serta status pinjaman buku saat siswa akan pindah taupun lulus dari sekolah, harus dilakukan pengecekan yang lama. Dalam penelitian ini menggunakan metode "*Prototype*", SOA dibangun dengan mengintegrasikan *web service* perpustakaan dan *web service* kesiswaan sebagai hasilnya, SOA dapat diimplementasikan sebagai sarana dalam penyebaran data siswa atau pun perolehan status siswa agar menjadi lebih mudah dan cepat.

Penelitian kedua dilakukan oleh Dipalokaeswara dan Aradea Mubarak Husni juga melakukan penelitian [3]. Pada penelitian ini, mengelola *service* akademik dan keuangan merupakan lingkup masalah yang kompleks. Hal ini terjadi di SMK Bina Lestari PUI. SMK Bina Lestari PUI tidak didukung oleh fasilitas pelayanan yang memadai, seperti fasilitas computer, sehingga pelayanan hanya memenuhi kebutuhan jangka pendek saja dan tidak dapat mengatasi perubahan yang ada. Oleh karena itu, cara mengatasinya dengan pendekatan arsitektur SOA. SOA dapat meletakkan fungsi aplikasi sebagai bagian dari pelayanan. SOA dapat digunakan berulang kali dan sifatnya *independen*. Pengembangan fungsi aplikasi untuk akademik dan keuangan dapat dilakukan terpisah, sehingga memudahkan melakukan perubahan dan penambahan fungsi.

Penelitian ketiga dilakukan oleh Kapojos F dan kawan kawan juga melakukan penelitian dalam jurnalnya [4]. Penelitian ini bertujuan untuk pembuatan *interface web service* dengan penerapan SOA sehingga dapat diintegrasikan dengan sistem informasi akademik di Fakultas Teknik di Universitas Sam Ratulangi. Metodologi dari penelitian ini adalah *top-up* yang menggunakan konsep SOA. Metodologi *top-up* ini untuk bagian analisis dan perancangan dilakukan di bagian awal. Tahap analisis penelitian ini didapatkan empat *service* utama dan lima *method*. *Service* ini antara lain matakuliah, dosen, mahasiswa, dan nilai. Selanjutnya *service* dan *method* ini di *desain* dan diimplementasikan pada bagian perancangan dan teknologi *web service*. Hasilnya adalah *interface* aplikasi yang dapat diintegrasikan pada sistem informasi akademik. Setelah diintegrasikan, dapat diketahui bahwa sifat-sifat SOA berhasil diimplementasikan di penelitian ini, selain *autonomy* dan *composability*. Kesimpulan yang didapatkan bahwa penggunaan SOA dan *web service* pada sistem informasi organisasi sangat tepat untuk aplikasi yang memiliki tingginya sifat penggunaan kembali. Selain itu, dapat juga diimplementasikan antar *platform* dan antar sistem operasi.

Irwan Hadi juga melakukan penelitian dalam jurnalnya [5]. Kota Mataram merupakan kota yang sedang berkembang, sehingga akses informasi sangat diperlukan oleh masyarakatnya salah satunya adalah informasi mengenai fasilitas umum. Informasi dalam ruang lingkup pemerintahan terdapat berbagai badan publik yang merupakan penugasan desentralisasi yang di dukung oleh kewenangan dari sisi otonomi daerah dari berbagai bidang. Berdasarkan data rekapitulasi izin hingga tahun 2015 BPMP2T telah menerbitkan sebanyak 2.114 izin, namun belum ada media maupun sistem yang digunakan untuk mempublikasikan izin untuk keterbukaan informasi, sehingga masyarakat tidak memiliki gambaran informasi berkenaan dengan keberadaan izin. Informasi seperti lokasi izin, jenis izin, masa berlaku izin serta informasi yang berkaitan dengan izin tersebut sangat dibutuhkan oleh masyarakat, khususnya masyarakat yang akan membangun usaha dan mendaftarkan izin perusahaannya.

Oleh karena itu, solusi masalah ini adalah membutuhkan sistem yang dapat menggambarkan visualisasi lokasi tempat izin berlangsung dalam bentuk peta sekaligus menampilkan informasi yang berkaitan dengan izin tersebut. Dalam hal ini dapat dibangun sebuah Sistem Informasi Geografis (SIG) untuk memetakan lokasi izin. Dengan adanya SIG perizinan dapat digunakan untuk publikasi izin sebagai keterbukaan informasi kepada masyarakat. Dalam implementasi SIG Perizinan tentunya data yang digunakan adalah data perizinan yang terdapat pada BPMP2T. Sehubungan dengan dibuatnya Sistem Pelayanan Perizinan Terpadu (SPPT) pada BPMP2T untuk melayani perizinan secara *online*, maka data hasil pengolahan yang di simpan di SPPT bisa digunakan oleh SIG Perizinan sebagai informasi yang akan ditampilkan. Agar data pada SPPT dapat ditampilkan pada SIG Perizinan maka diperlukan sebuah arsitektur untuk integrasi antar kedua sistem. Arsitektur untuk integrasi sistem harus bersifat fleksibel dan adaptif apabila ada perubahan proses bisnis, oleh karena itu digunakan sebuah arsitektur yaitu *SOA*. *SOA* memudahkan beberapa sistem berbeda dapat berintegrasi karena *SOA* menyajikan arsitektur yang berorientasi *service*. *SOA* bersifat fleksibel dan adaptif terhadap perubahan dan dapat menyesuaikan dengan cepat dengan menggunakan *service-service* yang telah dibangun.

### B. Service Oriented Architecture (SOA)

*SOA* merupakan sisi penting pada *service computing platform* dengan konsep, teknologi, dan tantangan yang baru. *Service Oriented Architecture* dapat dijadikan infrastruktur jika memenuhi beberapa hal. Hal tersebut adalah enkapsulasi logika bisnis menjadi *service*, dan penggunaan *message* antar *service*. *Service layer* akan menjadi penghubung antara *business logic* dan *application logic* [1].

*Service* adalah bagian umum pada beberapa sistem aplikasi (*reusable*). Bentuk *service* ini adalah modul program, aplikasi, maupun gabungan dari beberapa aplikasi yang saling terintegrasi. Penambahan *interface* (*wrapper*) pada satu atau beberapa aplikasi merupakan realisasi dari *service*. Aplikasi ini melakukan komunikasi dengan *service interface* melalui API (*application programmer interface*).

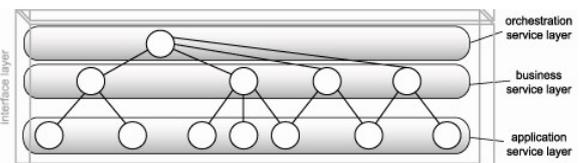
Berikut adalah prinsip-prinsip dasar layanan yang dimiliki oleh *SOA*:

- Service reusability*, penggunaan ulang layanan sehingga proses desain layanan yang tepat supaya mendukung penggunaan kembali.
- Service share formal contract*, yaitu setiap *service* memiliki kesepakatan mengenai cara untuk komunikasi. Kesepakatan ini berupa:
  - *Endpoint* dari *service*.
  - *Operations* di setiap *service*.
  - Dukungan setiap input dan *output message* oleh *operations*.
  - Aturan dan karakteristik pada *service* dan *operations* yang ada didalamnya.

- Service loose coupling*, yaitu *service* yang tidak terhubung erat. *Service* ini di desain supaya dapat beroperasi tanpa bergantung pada *service* lain. Dibutuhkan mekanisme komunikasi satu sama lain untuk meminimalisir kebergantungan ini.
- Service abstraction*, yaitu *service* yang membungkus logika. Kontrak layanan saja yang terlihat dari luar, sedangkan logika dan implementasinya tidak terlihat karena dirasa bukan hal penting bagi yang meminta layanan.
- Service autonomy*, yaitu *service* yang bersifat mandiri. Hal ini dapat dilakukan terpisah dalam pengembangan *service* nya. Logika dikendalikan secara penuh oleh *service*.
- Service statelessness*, yaitu *service* yang tidak memiliki status tertentu. Hal ini berkaitan terhadap aktifitas yang dilakukannya.
- Service discoverability*, yaitu *service* yang harus dapat ditemukan. Bagian yang meminta *service* dapat mencari dan mengerti mengenai layanan tersebut.

### C. Layering Pada SOA

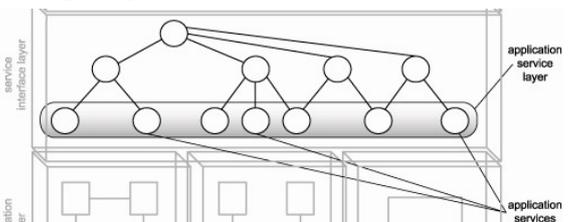
*Service interface layer* terbagi menjadi tiga bagian yang dapat di lihat pada Gambar 1, yaitu *orchestration service layer*, *business service layer*, dan *application service layer* [6].



Gambar 1. Abstraksi Dari Service Interface Layer [1]

#### 1. Application Service Layer

*Application service layer*, seperti yang ditunjukkan dalam Gambar 2 adalah *layer* yang terletak berbatasan dengan *application layer*. *Layer* ini menjadi fondasi dasar untuk membuat fungsionalitas-fungsionalitas aplikasi secara teknis. *Service* yang ada di dalam *layer* ini biasa disebut sebagai *application services*, dimana tugasnya adalah menyediakan fungsi-fungsi untuk pemrosesan data pada suatu lingkungan aplikasi baik aplikasi modern maupun aplikasi *legacy* (aplikasi yang sudah dipakai sejak lama). Fungsi yang disediakan bersifat *reusable* sehingga selalu dapat digunakan kembali.



Gambar 2. Application Service Layer [1]

Karakteristik yang dimiliki oleh *application services* adalah sebagai berikut:

- Menyediakan fungsionalitas yang spesifik terhadap tugas pemrosesan tertentu.

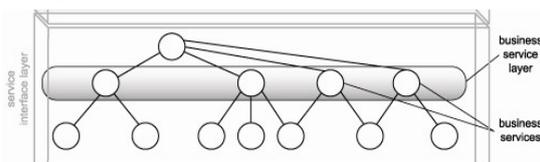
- b. Menggunakan sumber daya yang disediakan oleh *platform* teknologi tertentu.
- c. Berorientasi solusi.
- d. Bersifat generik dan *reusable*.
- e. Dapat digunakan untuk membuat integrasi *point-to-point* dengan *service* yang disediakan oleh aplikasi lain.
- f. Inkonsisten dalam tingkatan transparansi *interface* yang disediakan oleh *service* yang bersangkutan.
- g. Dapat terdiri dari *service* yang dikembangkan sendiri maupun dikembangkan oleh orang lain.

## 2. Business Service Layer

Berbeda dengan layer pertama yang bertanggung jawab untuk merepresentasikan teknologi yang digunakan dan logik dari aplikasi, *business service layer* yang ditunjukkan dalam Gambar 3 bertanggung jawab untuk menyediakan *service* yang berkaitan dengan representasi logik dari bisnis. Logik bisnis ini akan diatur dalam sebuah *service* bernama *business service*. Tanggung jawab dari *business service* adalah mengimplementasikan logik dan model bisnis perusahaan dalam sebuah *service*.

*Business service* sendiri terbagi menjadi dua katagori, yaitu:

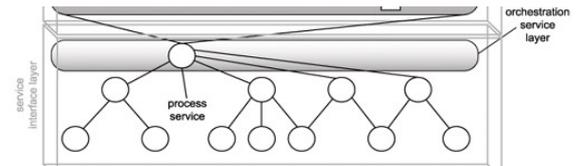
- a. *Task-centric business service*, yaitu *service* yang mengenkapsulasi logik bisnis yang spesifik terhadap sebuah kegiatan atau proses bisnis yang dimiliki perusahaan. *Service* ini paling mudah untuk dianalisis, namun potensi *reuse* yang ada terbatas.
- b. *Entity-centric business service*, yaitu *service* yang mengenkapsulasi sebuah entitas bisnis (seperti *invoice* atau jadwal kegiatan). *Service* ini memiliki potensi *reuse* yang tinggi, namun memerlukan proses analisis yang lebih sulit.



Gambar 3. Business Service Layer [1]

## 3. Service Orchestration Layer

*Orchestration service layer*, yang diilustrasikan dalam Gambar 4, adalah *layer* yang memberikan abstraksi paling tinggi dari logik dan aturan bisnis perusahaan serta bagaimana *service* harus berjalan. *Layer* ini digunakan untuk memenuhi kebutuhan perlunya sebuah *service* lain yang bertanggung jawab untuk mengatur bagaimana *service-service* yang ada dapat dieksekusi dengan urutan yang sesuai. *Orchestration service layer* mempunyai *process services*, yaitu *service* yang dapat mengatur *business service* dan *application service* untuk berjalan dengan aturan dan urutan yang sesuai dengan logik dan aturan bisnis perusahaan yang telah terdefinisi di dalam *process service* tersebut.



Gambar 4. Orchestration Service Layer [1]

## D. Service Oriented Analisis

Bagian ini adalah tahap perkembangan awal sebuah aplikasi *Service Oriented Architecture*. Pada tahap ini dilakukan penentuan lingkup dari aplikasi SOA yang akan dibuat. Kemudian melakukan identifikasi dan analisis *service* dari aplikasi.

## E. Service Oriented Desain

Hasil kandidat pada tahap analisis dari *service*, dan selanjutnya pembuatan desain lengkap dari kandidat *service*. Kemudian diimplementasikan pada sebuah komposisi yang mampu membentuk suatu proses bisnis. Pada SOAD, desain serta implementasinya pada *service* dengan memanfaatkan teknik web *service*. *Web service* ini dapat berbasis XML (*eXtensible Markup Language*), XSD (*XML Schema Definition*), SOAP (*Simple Object Access Protocol*), WSDL (*Web Service Definition Language*), dan WS-\* *extension*.

## F. Keuntungan SOA

SOA memiliki keuntungan yang tidak terbatas pada hematnya biaya serta tenaga pada pengembangan aplikasi. Selain itu, SOA juga dapat mewujudkan suatu organisasi dengan kemampuan yang cepat dalam mengadaptasi proses-proses bisnis di dalamnya, sehingga SOA mampu mengatasi tuntutan pasar pada saat ini [7]. Ada beberapa pendekatan yang bisa diambil yaitu:

1. Dengan SOA dapat dibuat pemrograman yang sudah berorientasi *service*. Program yang bersifat modular dapat ditambahkan bagian *web service* di dalamnya. Oleh karena itu, *service interface* di buat tanpa mengubah alur logika dari program sebelumnya. Sehingga, pengujiannya hanya dilakukan di bagian *interface* nya saja, sedangkan fungsi logika tidak perlu di uji karena tidak ada perubahan di dalamnya.
2. Aplikasi yang diperoleh dari pembelian secara paket, misalnya ERP, CRM dan lainnya dapat di *enkapsulasi* sehingga dapat dipanggil via *interface* dari *web service*. Paket aplikasi ini bias di buat *service* dari tingkatan *service* terendah seperti *customer lookup* hingga tingkatan tertinggi seperti *create invoice*. Cara ini sangat dapat diimplemetasikan karena beberapa vendor SOA telah menyiapkan adapter untuk paket aplikasi seperti SAP, Siebel dan lainnya. Oleh karena itu, maka *business logic* didalamnya dapat dipublikasikan sebagai *service*.

## G. Cara Kerja SOA

Inovasi membutuhkan perubahan dan SOA memudahkannya. SOA bekerja seperti *charger* untuk semua fungsi. SOA membangun *interface* yang bisa diakses oleh berbagai macam *software*. Selama ini, sebuah

*software* dibangun dengan cara mengikat data dan alat pemrosesnya dalam satu rangkaian. Tentu saja, semakin banyak *software* yang dibutuhkan akan membuat perusahaan mengeluarkan uang dan tenaga lebih banyak lagi. Untuk menggambarkan bagaimana SOA bekerja dalam sebuah perusahaan atau institusi bisnis, dapat mengambil contoh transaksi pembelian barang melalui internet yang dilakukan seorang netter atau pelanggan. Dalam sistem TI pengecer yang menggunakan sebuah SOA, pembelian barang secara *online* itu memicu serangkaian transaksi lainnya. Misalnya, kartu kredit pelanggan diverifikasi, bagian pengiriman barang diberitahu, gudang diminta untuk menyesuaikan persediaan barang, dan catatan-catatan pembukuan diperbaharui. Transaksi itu berupa input informasi yang dikirim melalui sistem atau *software* yang berlainan, yang kadang tidak sesuai dan tidak bisa berhubungan satu sama lain. Namun, teknologi SOA telah memungkinkan infrastruktur yang mendukung transaksi tersebut untuk dibaurkan dan dikombinasikan secara *integral* [8].

#### H. Web Service

*Web service* adalah cara menyediakan layanan. *Web service* dibuat bukan untuk berinteraksi langsung kepada client. *Web service* di sini memberikan layanan yang dapat di panggil oleh aplikasi. *Web service* dapat di akses dengan bantuan jaringan internet dengan bentuk XML sebagai format pengiriman pesan

Pada kondisi dimana *operating system* dan bahasa pemrograman beraneka ragam jenisnya, sehingga pertukaran data antar perangkat dengan beda *platform* menjadi permasalahan tersendiri. Untuk mengatasi permasalahan tersebut, maka perlu adanya *web service*, dimana *web service* memungkinkan *user* dari *platform* yang berbeda dapat bertukar data dan informasi dengan mudah.

Pembagian peranan *service* ada dua, yaitu *service provider* dan *service requestor*. *Service provider* menyediakan layanan yang akan digunakan, sedangkan *service requestor* yang meminta layanan tersebut berjalan. Alokasi dua pembagian ini tidak mutlak. Sebuah *service provider* bisa diubah menjadi *service requestor* yang bergantung pada peran yang ada di proses bisnis yang terkait [9].

Beberapa komponen pendukung *web service* adalah [10]:

##### 1. XML (*Extensilble Markup Languange*)

XML diciptakan pada akhir tahun 60-an oleh W3C. XML menjadi populer pada *ebusiness* pada akhir 90-an, pada saat traksaksi online menggunakan bahasa *server-side scripting* menjadi tren.

Arsitektur data yang direpresentasikan XML mencerminkan pemahaman dasar terhadap SOA. Dalam arsitektur data tersebut XML membangun format dan struktur dari *message* yang berjalan antar *service*.

##### 2. SOAP (*Simple Object Access Protocol*)

SOAP adalah bentuk standar dari dokumen dalam bentuk XML. SOAP melakukan proses *request* dan *response* antar *web service* pada aplikasi yang

memanggilnya. Komunikasi antar *services* yang menggunakan XML harus distandarisasi, dengan menggunakan SOAP, komunikasi antar *services* akan lebih teratur. Jika diibaratkan XML adalah bahasa Inggris, maka SOAP adalah *grammar* atau struktur bahasa.

Tujuan utama penggunaan SOAP adalah menstandarisasi format *message* yang berjalan antar *service*. Struktur dari SOAP terdiri dari SOAP *envelope*, *header* dan *body*.

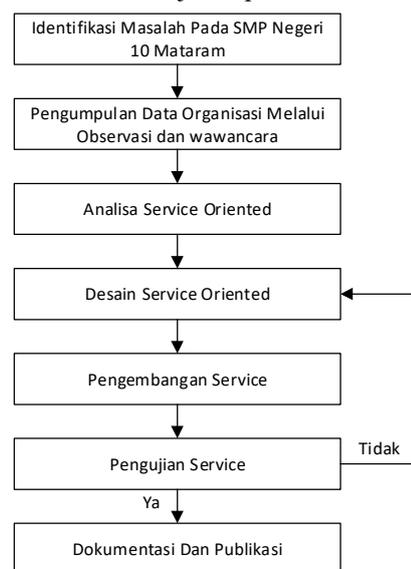
##### 3. WSDL (*Web Service Description Language*)

WSDL adalah dokumen dengan bentuk XML. Isi WSDL ini berupa penjelasan informasi detail dari sebuah *webservice* yaitu *method-method* yang ada pada *web service*, parameter yang diperlukan ketika memanggil sebuah *method*, dan hasil atau tipe data yang nilainya dikembalikan oleh *method* yang dipanggil tersebut.

### III. METODE PENELITIAN

#### A. Tahap Penelitian

Tahapan penelitian untuk membangun SOA pada SMP Negeri 10 Mataram ditunjukkan pada Gambar 5.



Gambar 5. Tahapan Metode Penelitian

##### 1) Identifikasi Masalah

Melakukan identifikasi masalah yang terjadi. Pada SMP Negeri 10 Mataram disetiap bagian-bagian tata usaha melakukan semua proses bisnisnya atau jobdesknya masing-masing bagian masih menggunakan *Microsoft Word* dan *Microsoft Excel* sehingga menyebabkan pegawai dan para guru kesulitan dalam mendapatkan data atau informasi yang dibutuhkan. Selain itu permasalahan yang muncul juga misalnya pada saat bagian keuangan membutuhkan data dari bagian kesiswaan, atau melakukan pertukaran data harus melakukan pertukaran data yang dibutuhkannya, atau meminta data ke bagian tata usaha lainnya yang memiliki data tersebut.

2) Pengumpulan Data

Tahapan ini digunakan untuk mendapatkan data dan informasi dari ruang lingkup masalah yang ada. Tahapan ini, yaitu:

a. Studi Pustaka

Pada tahap ini, dilakukan pencarian informasi pengetahuan dari sumber-sumber terpercaya. Sumber ini antara lain buku, dokumen, dan publikasi. Sumber pengetahuan ini menjadi dasar teori dalam pembuatan aplikasi SOA Untuk Tata Usaha SMP Negeri 10 Mataram.

b. Wawancara

Sumber informasi dari tahap ini adalah kepala tata usaha dan admin tata usaha. Informasi yang didapatkan berupa keterangan yang berhubungan dengan objek penelitian dan kebutuhan user pada aplikasi SOA ini.

c. Observasi

Tahap ini merupakan pengamatan secara langsung dari setiap user dan pengamatan langsung dari sistem yang sedang berjalan. Pada penelitian ini dilakukan di bagian semua unit tata usaha dari kegiatan yang dilakukan.

3) Service Oriented Analysis

Bagian ini merupakan fase awal dalam pembuatan aplikasi SOA. Pada tahap ini akan dilakukan identifikasi proses bisnis yang ada pada SMP Negeri 10 Mataram, identifikasi data apa saja yang dibutuhkan dalam proses bisnis yang ada pada SMP Negeri 10 Mataram, dan penentuan kandidat service.

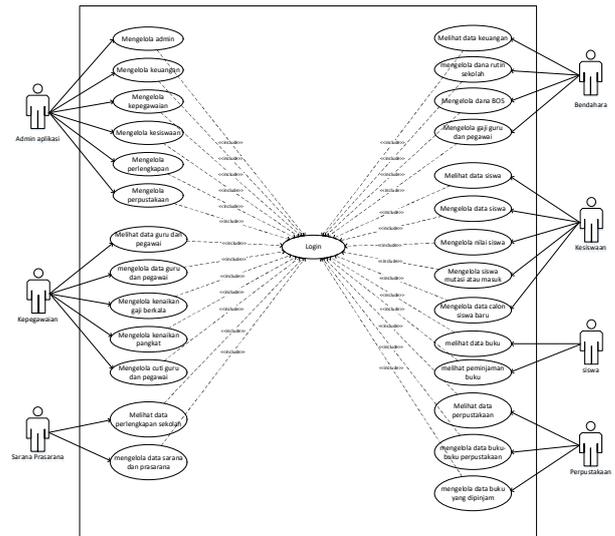
1. Proses Bisnis

Hasil tahap wawancara di SMP Negeri 10 Mataram dapat dilihat proses bisnis yang berlangsung pada unit Tata Usaha SMP Negeri 10 Mataram. Adapun beberapa bagian yang telah diidentifikasi proses bisnisnya antara lain adalah Kepala Tata Usaha, Bendara/Keuangan, Operator Sekolah, Kepegawaian, Kesiswaan, Sarana Prasarana/Perlengkapan dan perpustakaan.

2. Identifikasi Kebutuhan Data

Kebutuhan data diperoleh dari proses bisnis yang ada pada setiap bagian yang telah didefinisikan pada proses penentuan proses bisnis. Kebutuhan datanya berupa data administrator, data kebutuhan sekolah, data siswa dana bos, data gaji guru dan pegawai, data pegawai dan guru, data siswa, data orang tua siswa, data inventaris, data ruangan sekolah, data buku, data peminjaman buku.

Pada Gambar 6 ditunjukkan usecase yang didapatkan dari Analisa proses bisnis pada SMP Negeri 10 Mataram.

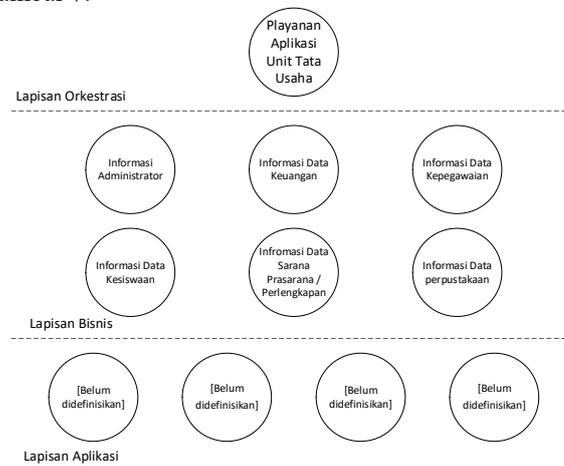


Gambar 6. Use Case Diagram Aplikasi Unit Tata Usaha SMP Negeri 10 Mataram

3. Penentuan Kandidat Service

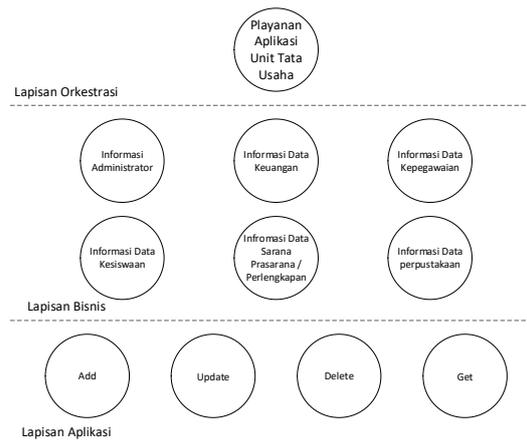
Pada bagian ini akan mengidentifikasi lapisan orkestrasi, lapisan bisnis dan lapisan aplikasi untuk pengembangan layanan perangkat lunak di SMP Negeri 10 Mataram. Identifikasi ini berdasarkan pada alur kerja yang terdapat di SMP Negeri 10 Mataram.

Berdasarkan pada pembahasan proses bisnis pada bagian arsitektur bisnis maka didapat kandidat service pada bagian orkestrasi adalah Pelayanan Sistem Informasi unit tata usaha SMP Negeri 10 Mataram yang ditunjukkan pada Gambar 7.



Gambar 7. Lapisan Orkestrasi dan Bisnis

Sedangkan untuk menentukan kandidat service pada lapisan aplikasi adalah dengan mendekomposisi proses bisnis dan kebutuhan data sehingga didapatkan lapisan aplikasi seperti pada Gambar 8.



Gambar 8. Lapisan Service

#### 4) Desain Service Oriented

Pada tahap desain *service oriented* dilakukan desain konkrit dari kandidat *service* yang didapatkan dari tahapan analisa *service oriented*. Pada tahap desain ini juga dilakukan pengimplemtasian kandidat *service* menggunakan teknologi *web service*. Pengimplentasian *web service* pada penelitian ini dilakukan dengan menggunakan *framework codeigniter*.

#### 5) Pengembangan Service

Pada tahap pengembangan *service*, hal-hal yang terkait *platform* pengembangan, seperti bahasa pemrograman yang digunakan, ditentukan tanpa memperhatikan tipe dari *service*. Secara spesifik, lingkungan implementasi akan menentukan bentuk atau implementasi fisik dari *service* dan bentuk orkestrasi *service* yang diperlukan.

Pada penelitian ini akan dilakukan pengimplementasian *service-service* yang didapatkan pada fase analisis kedalam aplikasi, yang dimana di sini pengimplentasian *service* hanya dilakukan pada *service* kesiswaan dan *service* perpustakaan. Pengimplementasian fitur akan dilakukan pada pembuatan aplikasi *web* dengan *framework codeigniter* dari fitur kesiswaan dan perpustakaan.

#### 6) Pengujian Service

Tahap ini bertujuan menguji *service* yang sudah diimplementasikan sesuai dengan yang dirancang. Pengujian dilakukan sebelum instalasi dilakukan. Pada tahap pengujian ini akan dilakukan pengujian *service* menggunakan metode *blackbox*. Sedangkan untuk pengujian lainnya dilakukan pengimplementasian *service* ke dalam dua buah sistem aplikasi yang menggunakan data yang sama yaitu sistem kesiswaan dan sistem perpustakaan. Kedua sistem tersebut menggunakan sebuah data yang sama yaitu data siswa yang ada di dalam *service* pengelolaan data siswa. Apabila terjadi perubahan data dari aplikasi kesiswaan, maka perubahan data tersebut akan berubah juga di aplikasi perpustakaan karena kedua aplikasi tersebut sudah terintegrasi dengan *service* yang sudah di implementasikan.

## IV. HASIL DAN PEMBAHASAN

### A. Desain Serive Oriented

Pada tahapan desain *service oriented* ini di lakukan pengimplementasian *service* yang sudah didapatkan dari tahapan analisa. Pada tahap analisa *service oriented* yang dilakukan di sebuah organisasi tata usaha SMP Negeri 10 Mataram didapatkan enam *service* yang akan diimplementasi pada tahap desain *service oriented*. Enam *service* tersebut telah sesuai dengan sifat SOA yang *reusable* sehingga dapat digunakan oleh beberapa fitur dan aplikasi yang dibutuhkan di SMP Negeri 10 Mataram. Pengimplementasian *service* pada penelitian ini dilakukan dengan menggunakan *framework codeigniter*. Berikut ini adalah penjelasan hasil dari tahap pengimplementasian *service* yang telah didapatkan:

#### 1) Service Pengelolaan Data Administrator

*Service* pengelolaan data administrator digunakan untuk melakukan pengelolaan data admin. Dalam *service* ini terdapat beberapa fungsi yang dapat digunakan untuk pengelolaan data administrator tersebut. Berdasarkan identifikasi aplikasi yang dibutuhkan pada SMP Negeri 10 Mataram *service* administrator dapat digunakan untuk semua aplikasi diantaranya adalah aplikasi keuangan, aplikasi kepegawaian, aplikasi kesiswaan, aplikasi sarana prasarana dan aplikasi perpustakaan.

#### 2) Service Pengelolaan Data Keuangan

*Service* pengelolaan data keuangan digunakan untuk melakukan pengelolaan data keuangan atau bendahara. Dalam *service* ini terdapat beberapa fungsi yang dapat digunakan untuk pengelolaan data keuangan tersebut. Berdasarkan identifikasi aplikasi yang dibutuhkan pada SMP Negeri 10 *service* keuangan dapat digunakan untuk membuat aplikasi keuangan yang ada pada SMP Negeri 10 Mataram berdasarkan proses bisnis yang berlaku.

#### 3) Service Pengelolaan Data Kepegawaian

*Service* pengelolaan data kepegawaian digunakan untuk melakukan pengelolaan data kepegawaian. Dalam *service* ini terdapat beberapa fungsi yang dapat digunakan untuk pengelolaan data kepegawaian tersebut. Berdasarkan identifikasi aplikasi yang dibutuhkan pada SMP Negeri 10 Mataram *service* kepegawaian dapat digunakan di beberapa aplikasi diantaranya adalah aplikasi kepegawaian dan aplikasi keuangan yang ada di SMP Negeri 10 Mataram berdasarkan proses bisnis yang berlaku.

#### 4) Service Pengelolaan Data Kesiswaan

*Service* pengelolaan data kesiswaan digunakan untuk melakukan pengelolaan data kesiswaan. Dalam *service* ini terdapat beberapa fungsi yang dapat digunakan untuk pengelolaan data kesiswaan tersebut. Berdasarkan identifikasi aplikasi yang dibutuhkan pada SMP Negeri 10 Mataram *service* kesiswaan dapat digunakan di beberapa aplikasi yaitu diantaranya adalah aplikasi kesiswaan, aplikasi perpustakaan dan aplikasi keuangan yang ada di SMP Negeri 10 Mataram.

#### 5) Service Pengelolaan Data Sarana Prasarana

*Service* pengelolaan data sarana prasarana berfungsi dalam aktifitas kegiatan mengelola data yang berhubungan

dengan sarana prasarana. Dalam *service* ini terdapat beberapa fungsi yang dapat digunakan untuk pengelolaan data sarana prasarana tersebut. Berdasarkan identifikasi aplikasi yang dibutuhkan pada SMP Negeri 10 Mataram *service* sarana prasarana dapat digunakan di beberapa aplikasi yaitu diantaranya adalah aplikasi sarana prasarana dan aplikasi keuangan.

#### 6) Service Pengelolaan Data Perpustakaan

*Service* pengelolaan data perpustakaan digunakan untuk melakukan pengelolaan data perpustakaan. Dalam *service* ini terdapat beberapa fungsi yang dapat digunakan untuk pengelolaan data perpustakaan tersebut. Berdasarkan identifikasi aplikasi yang dibutuhkan pada SMP Negeri 10 Mataram *service* perpustakaan dapat digunakan di aplikasi perpustakaan.

### B. Pengembangan Service Oriented Architecture

Pada SMP Negeri 10 Mataram membutuhkan sebuah sistem yang dapat terkomputerisasi untuk menunjang kinerja administrasi yang ada pada SMP Negeri 10 Mataram. Sistem atau aplikasi tersebut diharapkan dapat membantu kinerja para pegawai tata usaha SMP Negeri 10 Mataram dalam menjalankan proses bisnis yang berlaku. Sistem atau aplikasi yang dibutuhkan tata usaha SMP Negeri 10 Mataram saat ini diantaranya adalah aplikasi keuangan, aplikasi kepegawaian, aplikasi kesiswaan, aplikasi sarana prasarana dan aplikasi perpustakaan.

Pada penelitian ini akan dilakukan pengimplementasian *service-service* yang didapatkan pada fase analisis kedalam aplikasi, yang dimana di sini pengimplentasian *service* hanya dilakukan pada *service* pengelolaan kesiswaan dan *service* pengelolaan perpustakaan. Pengimplementasian fitur akan dilakukan pada pembuatan *web* dengan *framework codeigniter* dari fitur bagian kesiswaan dan bagian perpustakaan.

### C. Pegujian Service Oriented Architecture

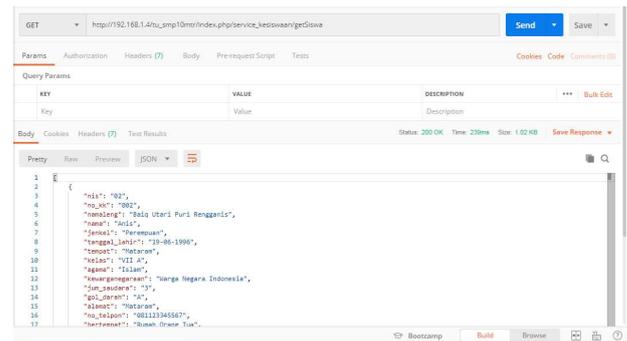
Pengujian *SOA* di penelitian ini terbagi menjadi dua. Pengujian pertama adalah *black box* dan yang kedua adalah pengujian pada aplikasi tata usaha bagian kesiswaan dan bagian perpustakaan.

#### 1) Black Box Testing

Berikut adalah uji menggunakan *black box* dari *service* yang telah diimplementasikan. Pengujian ini dilakukan dengan menggunakan *postman* untuk menguji REST API yang telah dibuat. Berikut contoh cara pengujian menggunakan aplikasi *postman*.

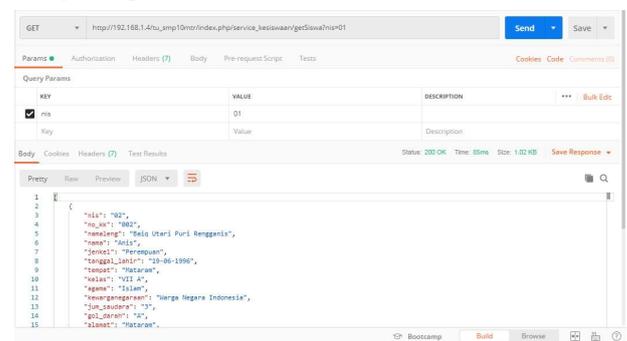
##### a. GET

Fungsi *get* digunakan untuk mengambil data pada *database*. Pada aplikasi ini menggunakan dua cara pengambilan data siswa secara keseluruhan dan mengambil data siswa dengan parameter tertentu. Untuk pengujian *postman* yang pertama dilakukan pilih metode *get* dan masukkan link pada *adres* bar lalu klik “*send*”. Berikut ini adalah hasil dari fungsi *get* yang diuji pada *tool postman* yang tidak menggunakan parameter yang ditunjukkan pada Gambar 9.



Gambar 9. Tampilan Fungsi GET Pada Postman

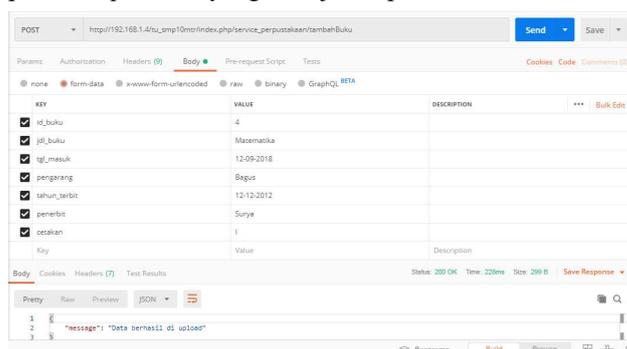
Berikut ini adalah hasil dari fungsi *get* yang diuji pada *tool postman* yang menggunakan parameter yang ditunjukkan pada Gambar 10.



Gambar 10. Tampilan Fungsi GET Pada Postman Berdasarkan Nis

##### b. POST

Fungsi *post* digunakan untuk menambah data ke dalam *database*. Cara untuk pengujian *postman* yang pertama dilakukan pilih fungsi *post* dan masukkan link pada *adres* bar, klik “*body*” di bawah *adres* bar, kemudian pilih form data dan masukkan *key* dan *value* yang diperlukan, lalu klik “*send*”. Berikut ini adalah hasil dari fungsi *post* yang diuji pada *tool postman* yang ditunjukkan pada Gambar 11.

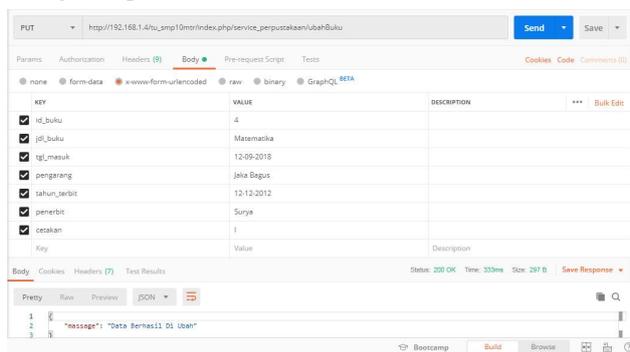


Gambar 11. Tampilan Fungsi POST Pada Postman

##### c. PUT

Fungsi *put* digunakan untuk mengubah data kedalam *database*. Cara untuk pengujian *postman* yang pertama dilakukan pilih fungsi *put* dan masukkan link pada *adres* bar, klik “*body*” di bawah *adres* bar, kemudian pilih form data dan masukkan *key* dan *value* yang ingin diubah, lalu klik “*send*”. Berikut ini adalah hasil

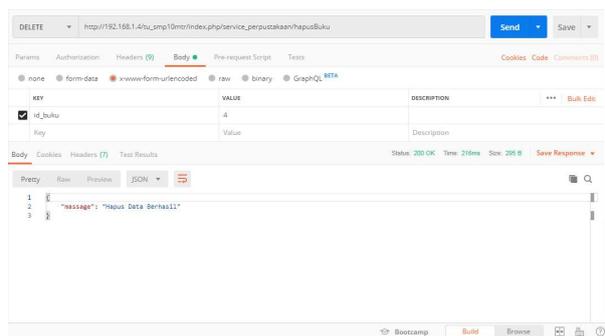
dari fungsi put yang diuji pada *tool postman* yang ditunjukkan pada Gambar 12.



Gambar 12. Tampilan Fungsi PUT Pada *Postman*

#### d. DELETE

Fungsi *delete* digunakan untuk menghapus data di dalam *database*. Cara untuk pengujian *postman* yang pertama dilakukan pilih fungsi *delete* dan masukkan link pada *adres* bar, klik “*body*” di bawah *adres* bar, kemudian pilih pilih *x-www-form-urlencoded* dan masukkan *key* dan *value* yang ingin dihapus, lalu klik “*send*”. Berikut ini adalah hasil dari fungsi put yang diuji pada *tool postman* yang ditunjukkan pada Gambar 13.



Gambar 13. Tampilan Fungsi DELETE Pada *Postman*

#### 2) Pengujian Aplikasi Tata Usaha Bagian Kesiswaa

Aplikasi tata usaha bagian kesiswaan SMP Negeri 10 Mataram ini merupakan aplikasi yang berfokus pada Tata Usaha bagian kesiswaan. Pada bagian kesiswaan dilakukan pengelolaan data siswa dari SMP Negeri 10 Mataram. Pengujian pada aplikasi ini menggunakan *service* admin dan *service* kesiswaan. Berikut pengujian *black box* untuk aplikasi tata usaha bagian kesiswaan yang dapat dilihat dari Tabel I.

#### 3) Pengujian Aplikasi Tata Usaha Bagian Perpustakaan

Aplikasi tata usaha bagian perpustakaan SMP Negeri 10 Mataram ini merupakan aplikasi yang berfokus pada Tata Usaha bagian perpustakaan. Pada bagian perpustakaan dilakukan pengelolaan data perpustakaan dari SMP Negeri 10 Mataram. Pengujian pada aplikasi ini menggunakan *service* admin, *service* kesiswaan dan *service* perpustakaan. Berikut pengujian *black box* untuk aplikasi tata usaha bagian perpustakaan yang dapat dilihat dari Tabel II.

TABEL I. Tabel Pengujian *Black Box* Untuk Aplikasi Tata Usaha Bagian Kesiswaan

No	Fitur	Hasil
1	Lihat Data Siswa	Valid
2	Tambah Data Siswa	Valid
3	Detail Data Siswa	Valid
4	Ubah Data Siswa	Valid
5	Hapus Data Siswa	Valid
6	Print Data Siswa	Valid
7	Lihat Data Nilai	Valid
8	Tambah Data Nilai	Valid
9	Detail Data Nilai	Valid
10	Ubah Data Nilai	Valid
11	Hapus Data Nilai	Valid
12	Print Data Nilai	Valid
13	Lihat Siswa Mutasi Masuk	Valid
14	Tambah Siswa Mutasi Masuk	Valid
15	Detail Siswa Mutasi Masuk	Valid
16	Ubah Siswa Mutasi Masuk	Valid
17	Hapus Siswa Mutasi Masuk	Valid
18	Lihat Siswa Mutasi Keluar	Valid
19	Tambah Siswa Mutasi Keluar	Valid
20	Detail Siswa Mutasi Keluar	Valid
21	Ubah Siswa Mutasi Keluar	Valid
22	Hapus Siswa Mutasi Keluar	Valid

TABEL II. Tabel Pengujian *Black Box* Untuk Aplikasi Tata Usaha Bagian Perpustakaan

No	Fitur	Hasil
1	Lihat Data buku	Valid
2	Tambah Data Buku	Valid
3	Detail Data Buku	Valid
4	Ubah Data Buku	Valid
5	Hapus Data Buku	Valid
6	Print Data Buku	Valid
7	Lihat Data Peminjaman	Valid
8	Tambah Data Peminjaman	Valid
19	Ubah Status Peminjaman	Valid
10	Hapus Data Peminjaman	Valid
11	Print Data Peminjaman	Valid
12	Lihat Data Rekap	Valid
13	Detail Data Rekap	Valid
14	Print Data Rekap	Valid

#### D. Pembahasan

Berdasarkan hasil pengujian di atas didapatkan enam *service* yang telah diimplementasikan ke dalam *web service* dengan *framework Codeigniter* yaitu: *service* administrator, *service* keuangan, *service* kepegawaian, *service* kesiswaan, *service*, sarana prasarana dan *service* perpustakaan. Pada pengujian *service* ada dua aplikasi yang digunakan sebagai pengujian yaitu aplikasi kesiswaan dan aplikasi perpustakaan.

Pada pengujian *service* yang dilakukan dengan dua cara yaitu menggunakan metode *black box* dengan *tool postman* dan pengujian pada aplikasi kesiswaan serta aplikasi perpustakaan. Pada pengujian *service* menggunakan

metode *black box* didapatkan hasil yang *valid* pada semua fungsi yang terdapat pada *service-service* tersebut. Sedangkan pada pengujian dengan menggunakan aplikasi kesiswaan dan aplikasi perpustakaan dapat menggunakan *service-service* tersebut dengan aplikasi yang berbeda. Pada pengujian aplikasi ini *service* yang digunakan secara bersamaan yaitu *service* administrator yang digunakan pada aplikasi kesiswaan dan aplikasi perpustakaan pada saat *login* aplikasi dan *service* kesiswaan digunakan untuk menampilkan data siswa yang ada di aplikasi kesiswaan dan aplikasi perpustakaan. Dari pengujian *service* yang dilakukan bisa disimpulkan bahwa SOA bersifat *reusable* yang artinya fungsi-fungsi yang terdapat dalam *service* yang telah dibuat dapat digunakan secara berulang-ulang. Selain itu SOA juga bersifat *interoperability* yang artinya *service* yang telah dibuat dapat digunakan di *platform* mana saja seperti *web*, *android*, *mobile*, dan lain-lain. Selain itu juga SOA bersifat *highly interoperable* yang artinya *service-service* yang sudah dibuat dapat mudah dioperasikan kedalam sistem atau aplikasi yang akan dibuat.

Penggunaan *service* pada aplikasi ini adalah sebagai *server* penyedia data yang dapat diambil oleh aplikasi dengan memanggil URL dari fungsi yang ada di dalam *service* tersebut. Dengan menggunakan *service* sebuah aplikasi berbasis MVC tidak lagi menggunakan model karena data-data yang diperlukan sudah disediakan oleh *service* yang ada.

## V. KESIMPULAN DAN SARAN

### A. Kesimpulan

Adapun kesimpulan dari penelitian di SMP Negeri 10 Mataram ini adalah:

1. SOA yang telah dibangun pada SMP Negeri 10 Mataram menggunakan *web service* didapatkan sebuah arsitektur sistem informasi yang saling terintegrasi pada setiap bagian-bagian tata usaha, sehingga memudahkan para pegawai unit tata usaha dalam melakukan pertukaran data.
2. Pegawai tata usaha SMP Negeri 10 Mataram dapat menggunakan *service* tersebut untuk pembuatan aplikasi-aplikasi yang akan datang atau mengembangkan aplikasi yang dibutuhkan organisasi tata usaha SMP Negeri 10 Mataram.
3. *Service* yang diimplementasikan pada SMP Negeri 10 Mataram yaitu *service administrator*, *service keuangan*, *service kepegawaian*, *service kesiswaan*, *service sarana prasarana* dan *service perpustakaan*.
4. Dilakukan pengujian *service* yang diimplementasikan di SMP Negeri 10 Mataram pada aplikasi kesiswaan dan aplikasi perpustakaan sudah terintegrasi.
5. SOA yang diimplementasikan dengan *web service* untuk sistem kesiswaan dan sistem perpustakaan menghasilkan sistem yang lebih adaptif terhadap perubahan, yang dimana apabila terjadi perubahan

*service* tersebut tidak mempengaruhi atau menghambat sistem tersebut.

### B. Saran

Saran dari perancangan dan pembangunan *service oriented architecture* pada SMP Negeri 10 Mataram ini adalah:

1. Untuk pengembangan lebih lanjut diharapkan aplikasi *client* dapat dibangun pada *platform* lain contohnya *android*, *ios*, *windows phone* dan lain-lain.
2. Untuk pengembangan lebih lanjut diharapkan adanya pembuatan keamanan dari sistem dalam *method-method* hak akses pada *web service*.
3. Pada tahapan analisa diharapkan melakukan analisa kebutuhan-kebutuhan dan proses bisnis pada SMP secara umum sehingga bisa digunakan oleh SMP-SMP lainnya.

## DAFTAR PUSTAKA

- [1] T. Erl, *Service-Oriented Architecture: Concepts, Technology, and Design*. 2005.
- [2] M. K. Kris Iswanto S, Hendry, "Integrasi Sistem Manajemen Data Siswa, Sistem Perpustakaan dan Sistem Keuangan Berbasis Service Oriented Architecture (SOA) (Studi Kasus: SMK Kristen Salatiga)," no. 672009163, 2013.
- [3] A. Dipalokareswara and H. Mubarak, "Implementasi Service Oriented Architecture untuk Pengintegrasian Fungsi Akademik dan Keuangan," *Res. Gate*, no. January 2014, pp. 0–6, 2014.
- [4] F. Kapojos, H. F. Wowor, a M. Rumagit, and a P. R. Wowor, "Implementasi Service-Oriented Architecture dengan Web Service untuk Aplikasi Informasi Akademik," *J. Fak. Tek. UNSRAT*, vol. 1, no. 1, pp. 1–5, 2012.
- [5] I. Hadi, I. B. K. Widiartha, and I. W. A. Arimbawa, "Desain dan Implementasi Service Oriented Architecture Pada Sistem Informasi Geografis Pemerintah Kabupaten Lombok Barat," *J. Comput. Sci. Informatics Eng.*, vol. 1, no. 1, p. 33, 2018.
- [6] H. Deviana, "Penerapan XML Web Service Pada Sistem Distribusi Barang," no. September, 2011.
- [7] Fachry Frisandy, "Implementasi Service-Oriented Architecture (SOA) Pada Pengembangan Sistem Pembelajaran Mobile," pp. 1–4, 2011.
- [8] Lucky, "XML Web Service: Aplikasi Desktop, Internet & Handphone," 2008.
- [9] I. Yoana, S. Tarigan, and R. Hartanto, "Perancangan Basis Data dan Layanan Akses Berbasis Service Oriented Architecture (SOA) Untuk Dinas Kesehatan Kabupaten Sleman," no. 1, pp. 15–28, 2008.
- [10] A. A. T. Lukhman, "Service Oriented Architecture (SOA)," *J. Manaj. Inform. Progr. Pendidik. Vokasi Univ. Brawijaya*, 2012.