

PENGEMBANGAN SINGLE PAGE APPLICATION PADA SISTEM INFORMASI AKADEMIK

Poetri Lestari Lokapitasari Belluano

Poe3.setiawan@gmail.com
Universitas Muslim Indonesia

Abstrak

Sistem Informasi Akademik (SIKA) adalah sistem pengelola data sehingga memudahkan pengguna dalam kegiatan administrasi akademik secara *online*. *Single Page Application (SPA)* adalah aplikasi *web* yang sesuai pada satu halaman *web* dinamis tanpa *refresh page*. Penelitian bertujuan untuk mengaplikasikan SIKA sesuai kebutuhan pengguna, bermanfaat, dan mudah digunakan, mengembangkan pengelolaan informasi *realtime* dan membantu komunikasi data lebih cepat serta menghemat sumber daya perangkat lunak. Metode penelitian meliputi studi lapangan dan kepustakaan sehubungan pembahasan terkait *SPA*, SIKA, Forlap, PDPT dan *Feeder* Dikti. Perancangan sistem informasi menggunakan paradigma *Convention Over Configuration*. Pengelolaan aplikasi *DBMS* menggunakan *PostgreSQL*, dan *port TCP-IP* sebagai sarana komunikasi data. Hasil penelitian berupa aplikasi Sistem Informasi Akademik (xSIA) yang *user friendly* dengan tampilan informasi berbentuk teks dan grafik yang sinkron. Serta pengelolaan informasi *realtime* yang menghemat sumber daya perangkat lunak, karena *Client side* mengelola sendiri data yang dibutuhkan sesuai permintaan tanpa membebani kinerja server dengan tidak melakukan proses *Rendering Data* berulang-ulang.

Kata Kunci : SIKA, SPA, PostgreSQL, xSIA

1. Pendahuluan

Sistem Informasi Akademik (SIKA) adalah suatu sistem yang dibangun untuk mengelola data-data sehingga memberikan kemudahan kepada pengguna dalam kegiatan administrasi akademik kampus secara *online*. SIKA memberikan beberapa manfaat antara lain terjadinya Integrasi Data, sebagai pusat informasi, alat rekam kegiatan akademik, serta sebagai media komunikasi pengguna yakni seluruh pelaku dalam ruang lingkup akademik baik dari tingkatan Program Studi hingga Perguruan Tinggi/Universitas.

Kebutuhan akan Sistem informasi terkait langsung dengan kepentingan kinerja dari Pangkalan Data Pendidikan Tinggi (PDPT) bagi tiap-tiap Perguruan Tinggi (PT) sebagai media pelaporan akademik yang terintegrasi dengan PDDIKTI. Dalam hal ini seluruh aktivitas akademik yang telah direncanakan/ditetapkan/diatur oleh Program Studi adalah wajib untuk dilaporkan secara *realtime* ke PDPT Universitas yang kemudian diteruskan secara otomatis ke PDDIKTI.

Seperti pada penelitian sebelumnya [2], dimana sistem informasi akademik berbasis *web service* telah mengimplementasikan adanya integrasi, sinkronisasi, dan interoperabilitas sistem menjadi faktor dasar penerapan Sistem Informasi Akademik berbasis *website*. Begitu penting kewajiban pelaporan yang harus dilaksanakan oleh PT sehubungan dengan pelaporan akademik baik di tingkat Program Studi (Prodi), Fakultas, hingga Perguruan Tinggi (PT) itu sendiri. Sehingga dibutuhkanlah pengembangan aplikasi pengelolaan data akademik yang mampu terhubung dan berkomunikasi secara *real-time* dan otomatis. Untuk mendukung kinerja program studi, fakultas, dan perguruan tinggi agar dapat secara konsisten melaksanakan pelaporan akademik ke PD-DIKTI, tidak cukup hanya menerepakan aplikasi berbasis *Web* dengan desain Antarmuka Pengguna (*User Interface*) sebagaiperangkat virtual semata-mata berfokus pada memaksimalkan pengalaman pengguna.

Namun tuntutan kebutuhan akan segala bentuk transaksi oleh pengguna khususnya terhadap performansi aplikasi *web* sistem informasi akademik, dimana kendala-kendala akan beban kerja server *web* untuk mentransfer berkas atas permintaan pengguna melalui protokol komunikasi yang telah ditentukan. Dikarenakan sebuah halaman *web* dapat terdiri atas berkas teks, gambar, video, dan lainnya. Serta tidak jarang pula aplikasi sistem informasi akademik nampak tidak *user-friendly* sehingga para pengguna sulit untuk mengoperasikan aplikasi.

Oleh karenanya dibutuhkan peningkatan keandalan bentuk aplikasi pada sisi *User Experience* yakni proses meningkatkan kepuasan pengguna (pengguna aplikasi, pengunjung *website*) dalam meningkatkan kegunaan dan kesenangan yang diberikan dalam interaksi antara pengguna dan produk dalam mengurangi beban kerja *Server*.

Hal tersebut akan didukung dengan dibangunnya suatu aplikasi *web* menggunakan teknologi *Single page application (SPA)* yakni menggunakan satu halaman *web* saja sebagai tampilan dari aplikasinya, sehingga dapat mengurangi beban kerja dari server dan browser serta menjadikan kinerja aplikasi *web* nampak bersifat aplikasi desktop dengan proses *Rendering* terhadap semua data-data yang sudah dimasukkan yang kemudian diterjemahkan dalam sebuah bentuk *output*.

Rumusan masalah pada penelitian ini, berfokus tentang bagaimana meningkatkan User Experience menggunakan teknologi Single Page Applications, serta mengurangi beban kerja server pada sistem informasi akademik. Bertujuan untuk membantu mengaplikasikan sistem informasi akademik yang *user-friendly* sesuai kebutuhan dan keinginan pengguna (*user*), bermanfaat, dan mudah digunakan, dan mengembangkan pengelolaan informasi secara *realtime* dan membantu komunikasi data lebih cepat serta menghemat sumber daya perangkat lunak.

Penelitian [2][3] menyimpulkan bahwa Sistem Informasi Akademik xSIA berupaya mewujudkan interoperabilitas sistem akademik yakni mampu melaksanakan pertukaran data/informasi melalui suatu protokol *webservice* yang disetujui bersama melalui bermacam-macam jalur komunikasi dengan berbagai ragam sistem untuk bekerja sama dan berinteraksi dengan aplikasi lainnya, baik dari level Program Studi ke PDPT Universitas yang kemudian diteruskan ke Feeder PDDIKTI.

Dalam penelitian [1] menyatakan bahwa *Single Page Application (SPA) is composed of individual components that can be replaced/updated independently, without refreshing/reloading whole page so that the entire page needs not to be reloaded on every user action.*

Aplikasi Halaman Tunggal (*Single Page Application*) terdiri dari komponen individual yang dapat diganti atau diperbaharui secara mandiri, tanpa tindakan *refresh* atau *reload* pada seluruh halaman, sehingga seluruh halaman tidak perlu dimuat ulang pada setiap tindakan oleh pengguna.

Banyak situs web masih diperkenalkan secara tradisional dengan tujuan semata-mata untuk melayani secara statis halaman (konten) ke pengguna, termasuk konten seperti Gambar, CSS, JavaScript, dan sebagainya. Namun seiring perkembangan akan kebutuhan, maka fungsi situs web berkembang untuk mempublikasikan bisnis dan mengeksplorasi, sehingga persyaratan terhadap situs *web* meningkat untuk melayani secara dinamis serta memuat data dari *database*.

2. Metode

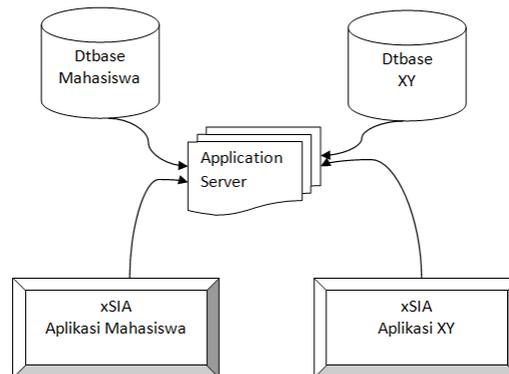
Metode penelitian meliputi studi lapangan dan kepustakaan sehubungan pembahasan terkait *Single Page Application (SPA)*, Sistem Informasi Akademik (SIKA), Forlap, Pangkalan Data Pendidikan Tinggi (PDPT) dan dokumen Feeder Dikti.

Tahapan penelitian yang dijalankan antara lain ;

1. Identifikasi Masalah, pada tahapan identifikasi masalah peneliti berusaha menguraikan permasalahan yang ditemui di tingkat Program Studi Teknik Informatika dan Universitas Muslim Indonesia.
2. Pengumpulan Data, setelah tahapan mengidentifikasi masalah, kemudian dilakukan pengumpulan data yang diperlukan untuk merancang form-form yang dibutuhkan oleh sistem agar diperoleh output sesuai kebutuhan user, Seperti data-data akademik tentang mahasiswa, mata kuliah, program studi, Aktivitas Kuliah Mahasiswa, Nilai Semester Mahasiswa dan Riwayat Status Mahasiswa. Populasi data akademik yang digunakan sebagai *sample* adalah data transaksi akademik periode tahun 2012, 2013, 2014, dan 2015.
3. Analisis Kebutuhan, tahapan ini merupakan tahapan pembuatan analisa aliran kerja sistem yang sedang berjalan untuk disesuaikan dengan model sistem yang akan dibangun. Antara lain menggunakan Model *Prototyping* untuk merepresentasikan secara grafis alur kerja sistem. Pemodelan sistem memanfaatkan tools *Unified Modeling Language*, Pengelolaan aplikasi *DBMS* menggunakan *PostgreSQL*, dan *port TCP-IP* sebagai sarana komunikasi data.
4. Desain Sistem, pada tahap desain sistem peneliti akan merancang aliran kerja manajemen, desain pemrograman, dan pengembangan sistem yang akan diterapkan sesuai dengan hasil analisis kebutuhan. Oleh karena itu, Perancangan sistem informasi menggunakan paradigma *Convention Over Configuration*.
5. Pengujian sistem, tahapan ini adalah tahapan paling penting setelah aplikasi dibangun dan dikembangkan sehingga diharapkan sistem telah dapat berjalan sesuai dengan alur kerja sistem dan menghasilkan data keluaran berdasarkan kebutuhan pengguna akhir. Dalam hal ini menggunakan metode *Blackbox Testing* untuk menguji keandalan sistem aplikasi sesuai penerapan teknologi *Single Page Application* terhadap kinerja dari web sistem informasi akademik.

Single Page Application (SPA) atau Aplikasi Halaman Tunggal adalah aplikasi *web* yang sesuai pada satu halaman *web* dengan tindakan dinamis tanpa menyegarkan halaman (*refresh page*). Interaksi Aplikasi Aplikasi *Single Page* dapat ditangani tanpa mencapai server, serta dapat meningkatkan kinerja dalam beberapa cara seperti waktu muat, menggunakan AJAX, mudah dinavigasi ke halaman dan sebagainya. Pengguna akhir (*end user*) akan lebih nyaman, karena sangat mudah untuk menavigasi atau mengarahkan ke seluruh halaman *web* dan filter konten yang berbeda.

Penerapan teknologi *Single Page Application (SPA)* yang digunakan bertujuan untuk mengurangi beban kerja *server* saat terjadi permintaan data dari *client*, dan meminimalisir penggunaan sumberdaya oleh *server*. Perbandingan antara *Web Pages Non-SPA* dengan yang menggunakan prinsip kerja *SPA*.



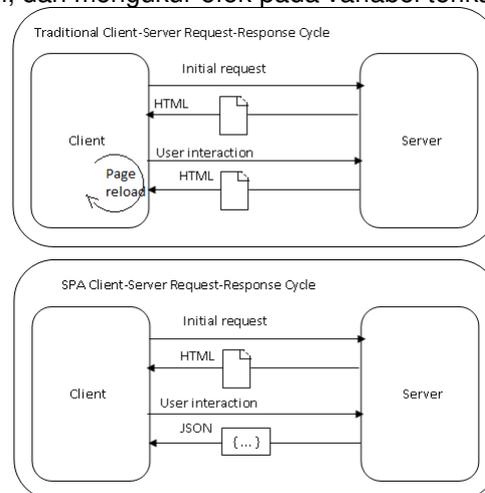
Gambar 1. *Single Page Application Control*

Gambar 1 menunjukkan pola kerja aplikasi yang menggunakan *SPA* dimana pada aplikasi induk yang biasanya dibangun dari sejumlah modul-modul, maka setiap modul dibangun kembali menjadi aplikasi-aplikasi kecil yang memiliki *database* tersendiri dan saling terintegrasi dengan sistem terdistribusi.

Penelitian ini menggunakan data primer yang diperoleh dari dokumen “Feeder Versi 1F” sudah tercatat lengkap semua Elemen Data setiap tabel PDPT. Berdasarkan dokumen teknis *webservice* feeder PDPT Ditjen-DIKTI telah diatur beberapa parameter dan elemen data yang dikirim (*push*) sebagai bentuk standarisasi dalam proses pengiriman data dari seluruh entitas internal yang terkait dengan PDPT.

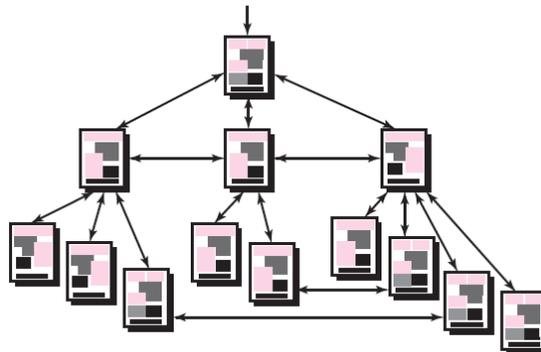
Dimana *sample data* berupa aktivitas akademik mahasiswa tahun angkatan 2012, 2013, 2014, dan 2015 kategori aktif terdaftar pada Fakultas Ilmu Komputer UMI, namun tidak termasuk data mahasiswa transfer/pindahan.

Rancangan Sistem didasarkan pada adanya manipulasi variabel bebas meliputi informasi yang akan disajikan bersifat dinamis sehingga pengguna dapat melakukan perubahan data informasi sewaktu-waktu jika diperlukan, dan mengukur efek pada variabel terikatnya.



Gambar 2. *Client-Server Request-Response Cycle*

Gambar 2 menunjukkan lalu lintas browser dikendalikan oleh file berekstensi **.js** dengan format data **[string]** dan bekerja untuk mengatur keluar masuknya data antar komponen aplikasi. Dimana proses render data Client side mengelola dirinya sendiri tanpa ketergantungan erat dari Server side.



Gambar 3. Struktur Hirarki Desain Arsitektur Web

Struktur Hirarki memungkinkan (melalui percabangan hypertext) Aliran kontrol secara horisontal melintasi cabang vertikal struktur. sehingga konten yang disajikan di sisi kiri dari hierarki dapat memiliki hubungan *hypertext* yang mengarah langsung ke konten yang ada di tengah atau kanan cabang dari struktur.

3. Hasil dan Pembahasan

Pada dokumen "Feeder Versi 1F" sudah tercatat lengkap semua Elemen Data setiap tabel PDPT. Berdasarkan dokumen teknis *web service feeder* PDPT Ditjen-DIKTI telah diatur beberapa parameter dan elemen data yang dikirim (*push*) sebagai bentuk standarisasi dalam proses pengiriman data dari seluruh entitas internal yang terkait dengan PDPT.

Daftar parameter elemen data yang dikirim, yakni ; Master Mahasiswa, Master Mata Kuliah, Transaksi Kuliah Mahasiswa, Transaksi Nilai Semester Mahasiswa, dan Transaksi Riwayat Status Mahasiswa.

Secara keseluruhan penelitian ini diimplementasikan menggunakan PostgreSQL 10.x DBMS engine. Aplikasi berjalan dengan OS Centos 7 yang bersifat *open-source*. Menggunakan Laravel sebagai *framework* pembuat aplikasi, dimana framework yang digunakan adalah Laravel 5.5.x untuk server site, dan Vue.js versi 2.5.x untuk client side. Bahasa pemrograman yang digunakan untuk merancang aplikasi yakni PHP 7.2 yang mampu meningkatkan kecepatan aplikasi. Pemanfaatan teknologi HTML 5 dan CSS 4 dalam aplikasi berbasis web. Menggunakan NGINX sebagai web server engine yang bersifat *opensource*. Serta metode *MVVM* untuk antarmuka pengguna yang bersifat lebih interaktif.

PDDIKTI Feeder menyediakan service yang dapat dimanfaatkan oleh Perguruan Tinggi agar dapat melakukan interoperabilitas system informasi yang sudah berjalan di lingkungan perguruan tinggi masing-masing. Sumber data yang digunakan untuk memenuhi kebutuhan PDDIKTI bisa berasal dari sebuah system informasi atau multi-sistem informasi, dimana data yang berasal dari sistem-sistem tersebut perlu dilakukan mapping terlebih dahulu untuk disesuaikan dengan standar yang ditentukan PDDIKTI.

Sistem informasi akademik (xSIA) ini dirancang secara khusus untuk membantu kebutuhan pengelolaan kegiatan administrasi kegiatan akademik yang masih dijalankan secara semi manual (Software Word Processing ex: word, excell) dan sistem informasi ini juga bisa berkomunikasi dengan aplikasi pelaporan resmi DITJEN – DIKTI (FEEDER DIKTI) melalui fitur web service FEEDER – DIKTI. Beberapa fitur yang dimiliki oleh xSIA adalah sebagai berikut: [3]

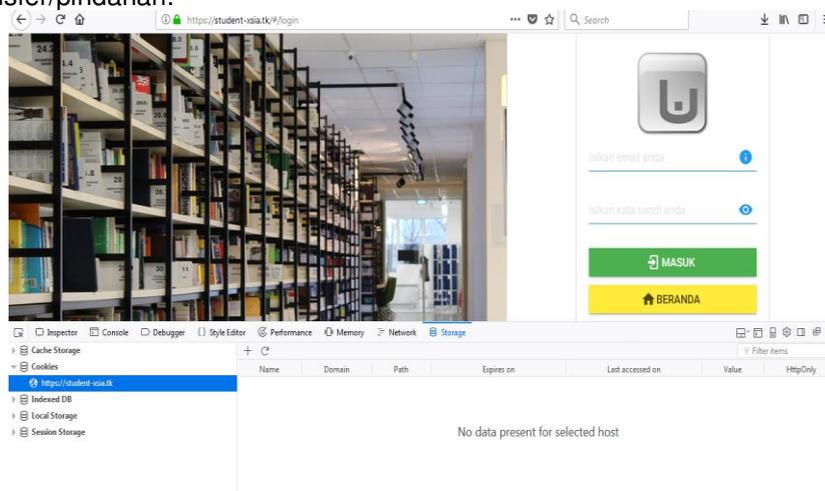
1. Berjalan pada *cloud system based*, dimana pengguna tidak perlu menyediakan perangkat keras (Server) dalam pengoperasian sistem ini, hal ini dapat memberikan penjaminan bahwa system berjalan setiap saat tanpa menghadapi kendala non teknis seperti listrik padam, keamanan server, biaya operasional yang lebih murah, dan pengaksesan dari dan kapan saja sepanjang pengguna mempunyai koneksi internet

2. Antar muka pengguna yang dirancang agar lebih mudah dioperasikan, baik dari *desktop work station* maupun *mobile devices* (ex: handphone dan tablet).
3. Perbaikan *Bug/Error* Jarak Jauh (jika ditemukan) akan ditangani dan diperbaiki dengan segera
4. Pengembangan Secara Terus – Menerus mengikuti perkembangan teknologi dan kebutuhan / penyesuaian dengan aplikasi pelaporan resmi DITJEN – DIKTI (FEEDER DIKTI), sehingga baik secara performansi dan kebutuhan, sistem akan beroperasi dengan optimal

Sedangkan untuk penerapan teknologi *Single Page Application (SPA)* yang digunakan pada implementasi web sistem informasi akademik ini, bertujuan untuk mengurangi beban kerja server saat terjadi permintaan data dari client, dan meminimalisir penggunaan sumberdaya oleh server.

Implementasi aplikasi sistem informasi akademik (xSIA) merupakan aplikasi utama yang mengintegrasikan sub-sub aplikasi sebagai media fungsi implemetansi dari modul-modul terkait dengan sistem akademik, sesuai prinsip kerja dari *Single Page Application*.

Data sampling berupa aktivitas akademik mahasiswa tahun angkatan 2012, 2013, 2014, dan 2015 kategori aktif terdaftar pada Fakultas Ilmu Komputer UMI, namun tidak termasuk data mahasiswa transfer/pindahan.

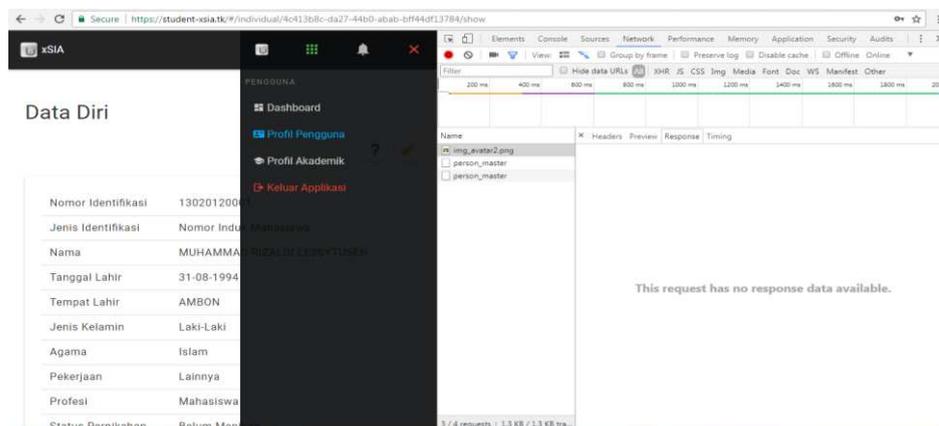


Gambar 4. Storage SPA

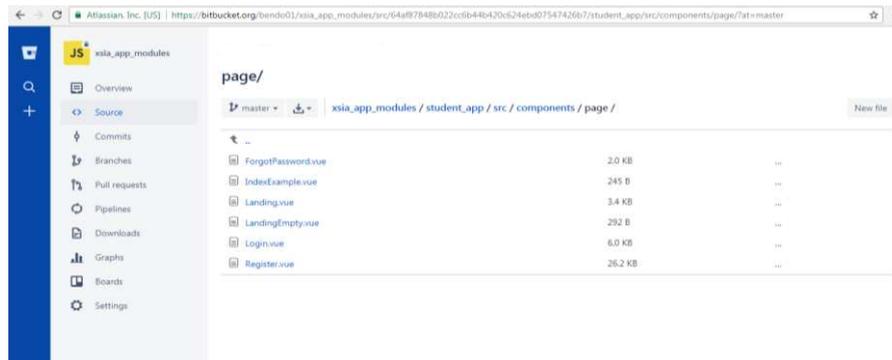
Gambar 4 pada storage *Client side* terlihat **No data present for selected host** yakni tidak adanya permintaan pemrosesan data dari *client side*, dikarenakan prinsip *SAP* memberikan loyalitas pada client side untuk mengelola datanya sendiri sesuai Request, dimana pada uji coba tersebut tidak ada perlakuan untuk mengakses data namun hanya menampilkan *Login page*.

Sedangkan *Server side* cukup dengan memberikan *Permission* terhadap *Request* yang datang dari *Clients*, sehingga tidak membebani kinerja server.

Untuk response Aplikasi *SAP*, dimana *client side* hanya menampilkan dan mengembalikan data yang diminta tanpa proses render dari *server side*. Seperti terlihat pada Gambar 5



Gambar 5. Response Page SPA



Gambar 6. *components testing*

Gambar 6 merupakan ciri khusus bahwa aplikasi web menerapkan model SPA yakni dengan memperhatikan terbentuknya sejumlah komponen-komponen yang berisi sub-sub komponen lainnya dalam satu aplikasi. Dimana setiap page menggunakan *file source code* berekstensi **Vue**. Serta penggunaan sumber daya yang kecil sehingga memudahkan tampilan data informasi secara *realtime*. Dikarena *Client side* mengelola sendiri data yang dibutuhkan sesuai permintaan tanpa membebani kinerja *server* yang tidak melakukan proses *Rendering* Data berulang-ulang.

4. Kesimpulan dan Saran

Dari hasil pengamatan antara aplikasi Single Page Application (SPA) yang diterapkan pada aplikasi Sistem Informasi Akademik(xSIA) dengan aplikasi serupa Non-SPA maka disimpulkan bahwa aplikasi web yang menerapkan teknologi SPA bertujuan untuk :

1. Membantu mengaplikasikan sistem informasi akademik yang user friendly sesuai kebutuhan dan keinginan pengguna (*user*), bermanfaat, dan mudah digunakan. Dilihat dari tampilan data informasi yang terstruktur yakni memperlihatkan informasi dalam bentuk teks dan grafik yang sinkron.
2. Mengembangkan pengelolaan informasi secara realtime dan membantu komunikasi data lebih cepat serta menghemat sumber daya perangkat lunak. Karena *Client side* mengelola sendiri data yang dibutuhkan sesuai permintaan tanpa membebani kinerja *server* yang tidak melakukan proses *Rendering* Data berulang-ulang.

Penelitian ini dapat mengembangkan aplikasi sistem informasi akademik dengan penerapan teknologi *Progressive Web Application (PWA)* untuk mengelola komponen-komponen modul sehingga dapat lebih mengimplementasikan fungsi *download*. Jika terjadi kehilangan koneksi jaringan, data tersimpan pada *client* dan berikutnya akan terjadi sinkronisasi data secara otomatis.

Berikutnya juga dapat mengembangkan aplikasi dengan menggunakan GraphQL sebagai penerapan protokol komunikasi data yang baru. aplikasi xSIA juga dapat meningkatkan performance dengan mengimplementasikan teknologi *IoT (internet of things)* untuk pemrosesan data yang menerapkan prinsip kerja *mechine learning*.

Daftar Pustaka

- [1] M.A.Jadhav, S. Balkrishna, and A. Deshmukh. 2015. *Single Page Application using AngularJS*. International Journal of Computer Science and Information Technologies., vol. 6(3), pp. 2876–2879 : Mumbai.
- [2] Poetri L.L.Belluano. 2016. Pengembangan Interoperabilitas Sistem Informasi Akademik Dengan Feeder Dikti Menggunakan Teknologi *Web Service*. Proceedings SNRIK Vol.1 Desember 2016, ISSN : 2443-048X.
- [3] Poetri L.L.Belluano. 2017. Penerapan Sistem Replikasi Dan Integrasi Basis Data Terdistribusi Pada Pangkalan Data Pendidikan Tinggi (PDPT). ILKOM Jurnal Ilmiah Vol.9 No.1 April 2017 Hal.42-48, ISSN: 2087-1716