

Penerapan Single Page Application Pada Proses Pengisian Online Data Rencana Studi Mahasiswa

Aryani Ristyabudi^{1*}, Husni Thamrin¹

¹Program Studi Informatika

Universitas Muhammadiyah Surakarta

*aryani.ristyabudi@gmail.com, husni.thamrin@ums.ac.id

ABSTRAK

Universitas Muhammadiyah Surakarta (UMS) telah membangun aplikasi berbasis web untuk pengisian data rencana studi mahasiswa dan menerapkannya selama belasan tahun. Paper ini mendeskripsikan sebuah studi untuk mempertimbangkan penerapan konsep Single Page Application (SPA) ke dalam aplikasi tersebut. SPA adalah merupakan teknik baru yang menggunakan satu halaman web saja untuk beberapa tahap dalam satu kesatuan aktivitas. Penerapan SPA dalam proses pengisian data rencana studi dilakukan menggunakan teknologi web termutakhir yaitu HTML5 dan AngularJS. Kinerja aplikasi yang baru diukur dengan mencermati jumlah data yang ditransfer dan waktu yang dibutuhkan untuk melakukan pengisian data rencana studi seorang mahasiswa. Pengukuran dilakukan menggunakan Wireshark. Hasil pengujian membuktikan bahwa pengisian data rencana studi menggunakan aplikasi dengan SPA membutuhkan transfer data kurang dari sepersepuluh dari proses menggunakan aplikasi tanpa SPA. Aplikasi dengan SPA menghemat total waktu yang dibutuhkan selama proses pengisian data menjadi sepertiga dari waktu yang diperlukan jika proses dilakukan menggunakan aplikasi tanpa SPA.

Kata kunci: single page application, JavaScript, AngularJS, rencana studi

1. Pendahuluan

Setiap awal semester, seorang mahasiswa membuat rencana studi yang harus disetujui oleh pembimbing akademik [1]. Pada masa lalu, sebelum komputerisasi merambah proses perekaman data rencana studi, rencana studi mahasiswa ditulis pada selembar kertas yang disebut Kartu Rencana Studi atau lebih dikenal dengan KRS. Ketika proses pengisian KRS berubah menggunakan komputer, istilah KRS masih sering digunakan meskipun proses tersebut sama sekali tidak menggunakan kertas berbentuk kartu. Untuk membedakan dengan proses pengisian kartu rencana studi yang menggunakan kertas, pengisian data rencana studi menggunakan komputer sering diistilahkan sebagai KRS *online*. Universitas Muhammadiyah Surakarta (UMS) telah menerapkan sistem KRS *online* selama belasan tahun. Buku Panduan Akademik menyebutkan bahwa pengambilan KRS mahasiswa dilaksanakan langsung melalui prosedur KRS *online* melalui komputer. Sistem KRS *online* UMS merupakan salah satu dari bagian aplikasi terpadu UMS yang dirancang untuk memfasilitasi aktivitas penyusunan atau pengaksesan sistem administrasi dan informasi akademik [2].

Pada sistem yang sedang berjalan sekarang, proses pengisian data rencana studi memerlukan akses ke beberapa halaman web, yaitu halaman login, halaman penyuntingan profil, halaman daftar mata kuliah, halaman penambahan data rencana studi, dan lain-lain. Beberapa halaman ditampilkan secara berulang. Setiap halaman

web dibuat atau dikonstruksi oleh *browser* setelah selesai mengambil (*loading*) seluruh data dan metadata dari server terkait halaman yang akan ditampilkan. Metadata yang dimaksud termasuk *tag* HTML (*hypertext markup language*) dan kode CSS (*content style sheet*). Menampilkan beberapa halaman web berarti mengambil data dan metadata dari server. Data yang berasal dari basis data selalu diambil ulang dari server basis data untuk memastikan kemutakhiran informasi yang ditampilkan. Oleh karena itu, menampilkan beberapa halaman secara berulang memerlukan banyak proses transfer data dan waktu.

Belakangan ini telah dikembangkan teknik antar muka yang disebut *Single Page Application* (SPA). Istilah SPA merujuk pada aplikasi berbasis web yang hanya menampilkan satu halaman saja. Artinya, meskipun pengguna beralih ke menu lain, URL tidak menunjukkan perubahan. Semua komponen utama pada SPA, seperti CSS, skrip dan sumber daya lain yang diperlukan diambil pada satu waktu di halaman utama. Komponen atau konten yang sesuai diambil secara dinamis tergantung pada kebutuhan interaksi atau permintaan pengguna [3]. Server hanya perlu mengirim dan menyebarkan bagian tertentu sesuai permintaan pengguna sehingga diharapkan dapat menghemat *bandwidth*. Browser dapat mengubah tampilan pada bagian tertentu saja secara dinamis. Karena *bandwidth* yang digunakan dalam SPA tergolong kecil, SPA dapat memberikan responsivitas yang tinggi dan kenyamanan bagi pengguna, tidak membingungkan dan menjengkelkan bagi pengguna [4].

Tulisan ini menguraikan upaya membuat *Single Page Application* untuk proses perekaman data rencana studi mahasiswa UMS dan menguji aplikasi tersebut dibandingkan dengan sistem KRS *online* di Universitas Muhammadiyah Surakarta. Objek penelitian adalah sistem KRS *online* UMS yang diakses oleh mahasiswa aktif UMS. Hasil penelitian menunjukkan bahwa aplikasi SPA dapat membantu mengurangi masalah yang muncul terkait lamanya proses KRS *online* dengan melakukan penghematan *bandwidth* dan waktu akses terhadap sistem.

2. Metode

Metode penelitian yang digunakan adalah eksperimen dan observasi, yaitu dengan membuat *Single Page Application* untuk merekam data rencana studi mahasiswa dan membandingkan kinerja aplikasi SPA dibanding aplikasi KRS *online* yang sedang berjalan. Kinerja aplikasi diukur dari jumlah data yang ditransfer antara server dan *browser* pada proses perekaman pengambilan satu mata kuliah di KRS. Selain itu, kinerja aplikasi diukur dari prediksi waktu yang dibutuhkan untuk menyelesaikan proses pengambilan mata kuliah.

1) Waktu dan Tempat

Waktu yang dibutuhkan untuk menyelesaikan penelitian ini kurang lebih lima bulan terhitung dari bulan November 2015 sampai dengan Maret 2016, dan bertempat di Universitas Muhammadiyah Surakarta.

2) Alat dan Bahan

Peralatan yang digunakan selama penelitian terdiri dari *software* dan *hardware*. *Software* yang digunakan antara lain Sublime Text 2, XAMPP, Google Chrome, Visual Paradigm Community Edition, DBDesigner 4, dan Wireshark. Sedangkan *hardware* yang digunakan adalah laptop dengan spesifikasi prosesor Intel Core i3, RAM 2 GB, *harddisk* 500 GB dan sistem operasi Windows 7 serta *smartphone* dengan spesifikasi prosesor 8 Core 1,5 GHz, RAM 2 GB, ROM 8 GB dan memori eksternal 4 GB.

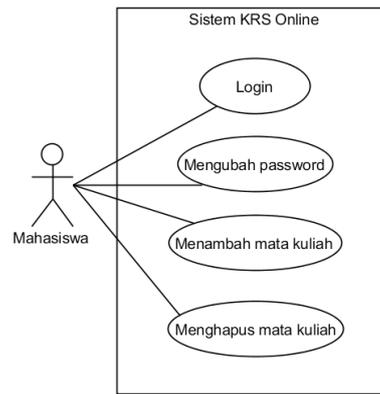
Bahan yang digunakan untuk penelitian ini adalah data mahasiswa aktif, data mata kuliah dan data pengambilan studi (Kartu Rencana Studi, KRS) mahasiswa aktif Fakultas Komunikasi dan Informatika Universitas Muhammadiyah Surakarta semester gasal tahun 2015.

3) Perancangan SPA

Perancangan *Single Page Application* untuk merekam proses pengambilan mata kuliah meliputi desain atau rancangan sistem dengan menggunakan model *Unified Modelling Language* (UML). Diagram yang digunakan antara lain: *use case diagram*, *class diagram*, *sequence diagram*, dan *activity diagram*. Berikut model UML dari aplikasi SPA yang dibuat untuk penelitian ini.

a. Use Case Diagram

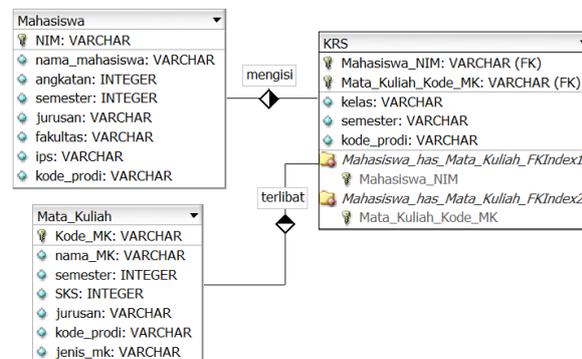
Use case diagram menggambarkan fungsionalitas yang diharapkan dari suatu sistem. *Use case diagram* merepresentasikan sebuah interaksi antara aktor dengan sistem, seperti yang digambarkan dalam Gambar 1 terkait interaksi antara mahasiswa dengan sistem. Aktor utama dalam aplikasi SPA ini adalah mahasiswa. Mahasiswa dapat melakukan empat aktivitas meliputi login ke dalam sistem, mengubah *password*, menambah mata kuliah dan menghapus mata kuliah.



Gambar 1. Use Case Diagram Mahasiswa

b. Class Diagram

Class diagram adalah sebuah diagram yang menggambarkan kelas/objek beserta atribut atau properti objek yang ada dalam sistem. Diagram ini dapat digunakan untuk mendeskripsikan aktivitas yang dilakukan objek atau layanan yang tersedia untuk mengubah nilai atribut objek. Aktivitas dalam kelas atau objek diwujudkan dalam bentuk fungsi, prosedur atau method. Aplikasi SPA yang dibuat dalam penelitian ini menggunakan *class diagram* seperti ditunjukkan pada Gambar 2. Diagram yang dimaksud menggambarkan bagaimana kelas/objek direkam dalam basis data (diwujudkan dalam tabel dan keterkaitan antar objek (diwujudkan dalam bentuk *relationship*). Terdapat dua kelas utama dalam aplikasi yaitu kelas Mahasiswa dan kelas Mata_Kuliah. Kedua kelas mempunyai hubungan yaitu “Mahasiswa mengambil Mata_Kuliah” dan hubungan itu diwujudkan dalam bentuk tabel bernama “KRS”.



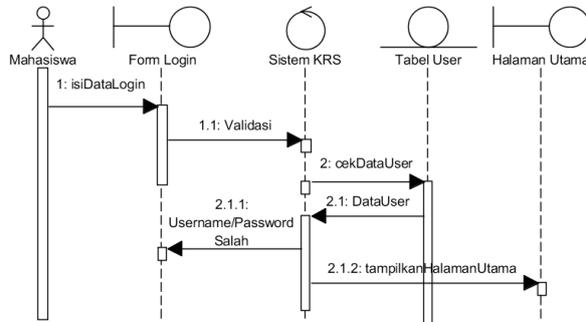
Gambar 2. Class Diagram Sistem KRS

c. Sequence Diagram

Sequence diagram adalah suatu diagram yang menggambarkan alur komunikasi antar obyek dengan waktu penyampaian pesan yang digambarkan secara vertikal. Terdapat empat *sequence diagram* dalam perancangan sistem KRS *online*, yaitu proses login, proses pengubahan *password*, proses penambahan mata kuliah dan proses penghapusan mata kuliah dari daftar KRS.

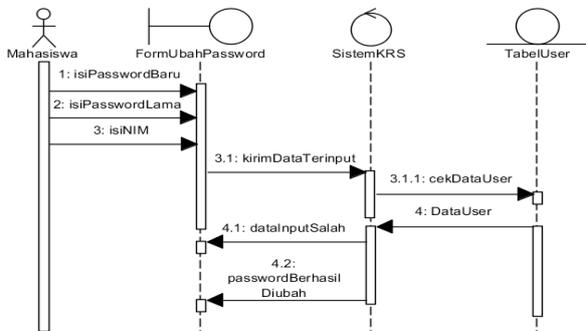
Sequence diagram untuk proses login diterapkan saat mahasiswa melakukan *login* seperti pada Gambar 3. Proses diawali dengan mengisi data *login* seperti

username dan *password* pada *form login*, kemudian sistem akan melakukan validasi data ketika mahasiswa mengklik tombol “Login”. Validasi oleh sistem dilakukan dengan cara mencocokkan data yang dimasukkan oleh mahasiswa dengan data yang ada pada basis data. Apabila data tidak cocok, maka akan ditampilkan peringatan bahwa data yang dimasukkan salah. Apabila data sesuai, sistem akan menampilkan halaman utama.



Gambar 3. *Sequence Diagram Login*

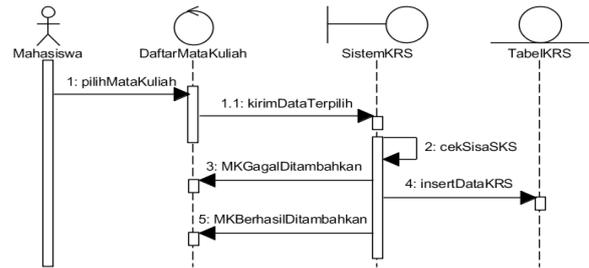
Sequence diagram mengubah *password* menampilkan proses ketika mahasiswa mengubah *password* yang digunakan sebagai data *login* seperti pada Gambar 4. Mahasiswa mengisi data *password* baru dan dikonfirmasi dengan data *password* lama. Melalui proses *request*, data user, *password* lama dan *password* baru dikirim ke sistem (server), server kemudian akan menguji kecocokan data *password* lama yang dimasukkan oleh mahasiswa dengan data yang ada pada basis data. Jika sesuai, maka sistem akan memutakhirkan *password* dengan menyimpan *password* baru ke basis data.



Gambar 4. *Sequence Diagram Ubah Password*

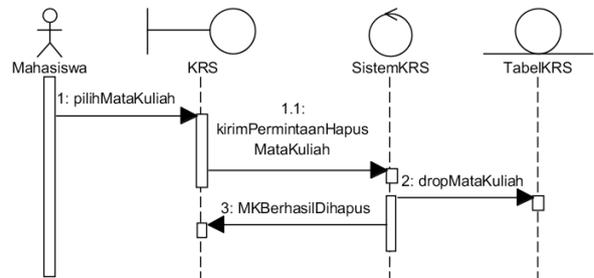
Sequence diagram menambah mata kuliah menampilkan proses ketika mahasiswa menambah data mata kuliah yang diambil ke dalam KRS, atau menambah matakuliah ke dalam daftar rencana studinya (Gambar 5). Proses dimulai dengan memilih mata kuliah dari sebuah daftar mata kuliah yang ditawarkan. Selanjutnya setelah mengklik tombol kirim, data mata kuliah baru akan dikirim ke sistem (server). Selanjutnya sistem akan mengecek daftar matakuliah yang telah dipilih mahasiswa dan melihat apakah penambahan data diperkenankan (yaitu terkait jumlah kredit maksimal yang boleh diambil mahasiswa). Jika kondisi memungkinkan, server akan

menambah data mata kuliah pada daftar rencana studinya di dalam basis data.



Gambar 5. *Sequence Diagram Tambah Mata Kuliah*

Sequence diagram menghapus mata kuliah menampilkan proses ketika mahasiswa menghapus mata kuliah yang sebelumnya telah dipilih (ada dalam KRS), seperti yang ditampilkan pada Gambar 6. Mahasiswa memilih mata kuliah yang ada dalam daftar mata kuliah yang telah direncanakan. Kemudian sistem akan menghapus data mata kuliah yang telah dipilih sebelumnya dari tabel pada basis data.



Gambar 6. *Sequence Diagram Hapus Mata Kuliah*

3. Hasil dan Diskusi

3.1. Hasil Penelitian

Rancangan *Single Page Application* (SPA) yang dibuat dalam penelitian ini sebelum diujikan di-*deploy* pada sebuah server web berikut dengan basis data yang diperlukan dalam pengujian. Sesuai dengan tahapan perancangan, tampilan dari aplikasi terbagi menjadi lima halaman tampilan yaitu, tampilan login, tampilan halaman utama, tampilan daftar rencana studi (atau halaman KRS), tampilan tambah mata kuliah, dan tampilan ubah *password*.

Tampilan login merupakan tampilan yang muncul pertama kali ketika membuka aplikasi SPA. Tampilan login ini digunakan untuk memvalidasi pengguna yang diizinkan dan yang tidak diizinkan mengakses halaman utama. Dalam tampilan ini, terdapat *form* yang terdiri dari dua *field* untuk input data yaitu *username* dan *password*, serta sebuah tombol untuk mengirim data dalam dua *field* tadi ke dalam sistem (server).

Tampilan utama merupakan halaman yang pertama kali ditampilkan setelah mahasiswa melakukan login dan diizinkan masuk ke dalam sistem (Gambar 7). Tampilan ini terdiri dari *header*, konten dan *footer*. Bagian *header* berisi judul aplikasi yang disertai dengan logo dan nama universitas. Judul aplikasi SPA tetap menggunakan nama “Sistem KRS Online” yang merupakan nama yang telah dikenal luas. Bagian konten terdiri atas tiga bagian, yaitu

bagian sebelah kiri, kanan atas, dan kanan bawah. Bagian konten sebelah kiri mengandung menu sistem, yaitu menu “KRS”, “TAMBAH MATA KULIAH”, “UBAH

PASSWORD”, dan “LOGOUT”. Bagian konten sebelah kanan mengandung kotak informasi berisi deskripsi singkat mengenai sistem KRS *online* UMS.



Gambar 7. Tampilan Utama

Tampilan daftar rencana studi atau KRS merupakan halaman yang ditampilkan ketika mahasiswa mengklik menu “KRS” (lihat Gambar 8). Di bagian konten sebelah kanan atas terdapat informasi singkat mengenai data diri mahasiswa yang bersangkutan. Di bagian konten sebelah kanan bawah berisi daftar mata kuliah yang telah direncanakan (tabel KRS) oleh mahasiswa yang

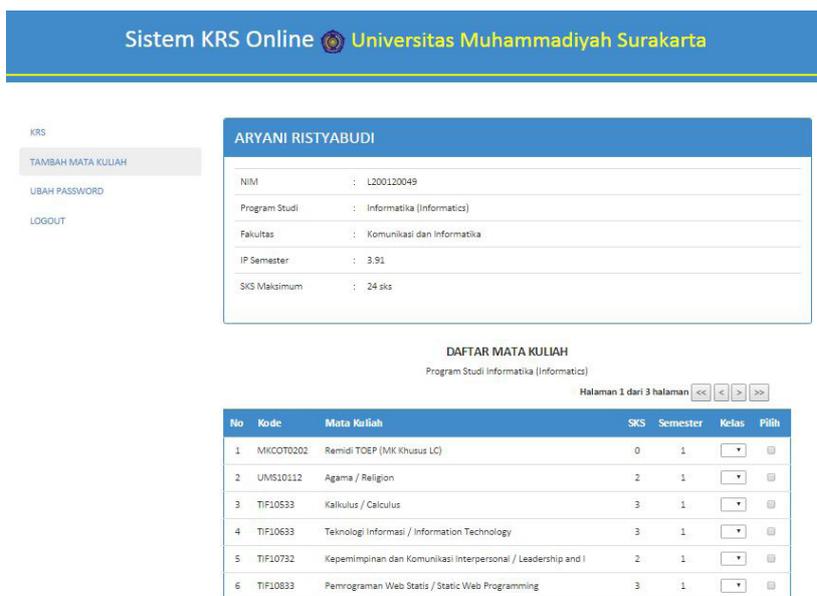
bersangkutan. Konten sebelah kanan akan berubah-ubah sesuai aksi yang dilakukan oleh mahasiswa. Tabel KRS yang ditampilkan memuat informasi mengenai kode mata kuliah, nama mata kuliah, semester mata kuliah yang bersangkutan, jumlah SKS, kelas, dan sebuah pilihan untuk menghapus mata kuliah yang bersangkutan



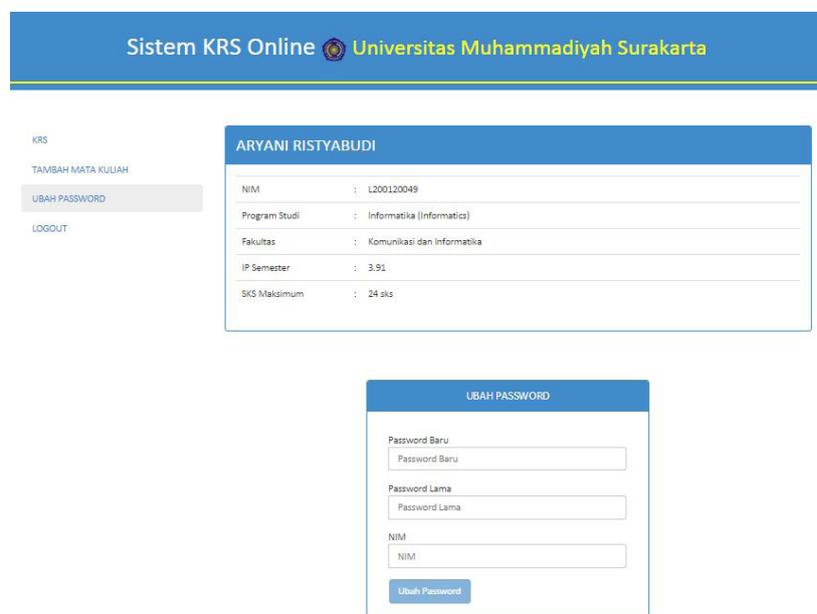
Gambar 8. Tampilan KRS

Tampilan tambah mata kuliah merupakan tampilan ketika mahasiswa mengklik menu “TAMBAH MATA KULIAH” (Gambar 9). Tampilan ini mirip dengan tampilan utama. Perbedaannya ada pada bagian konten sebelah kanan bawah yang menampilkan daftar mata kuliah yang bisa ditambahkan oleh mahasiswa pada

semester aktif. Tabel ini berisi beberapa informasi seperti kode mata kuliah, nama mata kuliah, semester mata kuliah yang bersangkutan, jumlah SKS, pilihan kelas, dan pilihan untuk menambahkan dengan mengklik tombol “Tambahkan”.



Gambar 9. Tampilan Tambah Mata Kuliah



Gambar 10. Tampilan Ubah Password

Tampilan ubah *password* ditampilkan setelah mahasiswa mengklik menu “UBAH PASSWORD”. Tampilan ini mirip dengan tampilan utama (lihat Gambar 10). Yang membedakan adalah pada bagian konten sebelah kanan bawah, yaitu menampilkan sebuah *form* untuk mengubah *password*. Dalam *form* ini, terdapat tiga *field* untuk input data *password* baru, *password* lama dan NIM mahasiswa yang bersangkutan, serta sebuah tombol untuk eksekusi data yang diinputkan oleh mahasiswa untuk kemudian diproses apakah *password* berhasil untuk diubah atau tidak.

3.2. Pengujian dengan Wireshark

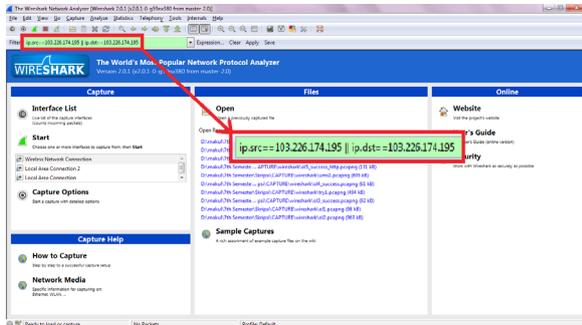
Untuk menunjukkan perbaikan kinerja aplikasi SPA yang telah dibuat, aplikasi dibandingkan dengan sistem KRS *online* yang masih berjalan. Pengujian menggunakan dua metrik yaitu total waktu dan ukuran data yang ditransfer (dalam *byte*). Kedua metrik tersebut diukur menggunakan

software Wireshark. Wireshark digunakan karena merupakan aplikasi penganalisa jaringan, yang mampu menangkap (*capturing*) setiap data yang lewat melalui salah satu antar muka jaringan dan menampilkannya dengan rinci [5]. Wireshark dapat menangkap data waktu (dalam detik) dan jumlah data transfer (dalam *byte*) yang lewat ketika melakukan proses KRS. Tujuannya adalah untuk membandingkan data transfer antara sistem KRS *online* dan aplikasi SPA sehingga dapat mengetahui sistem mana yang mempunyai data transfer lebih kecil serta sistem mana yang lebih cepat (hemat waktu).

1) Pengujian pada sistem KRS online

Pengujian dengan Wireshark dilakukan dengan pertama kali mencari alamat IP dari sistem KRS *online* yaitu dengan alamat web “akademik.ums.ac.id” yang didapatkan dengan menggunakan perintah “ping akademik.ums.ac.id” pada Command Prompt. Dari hasil

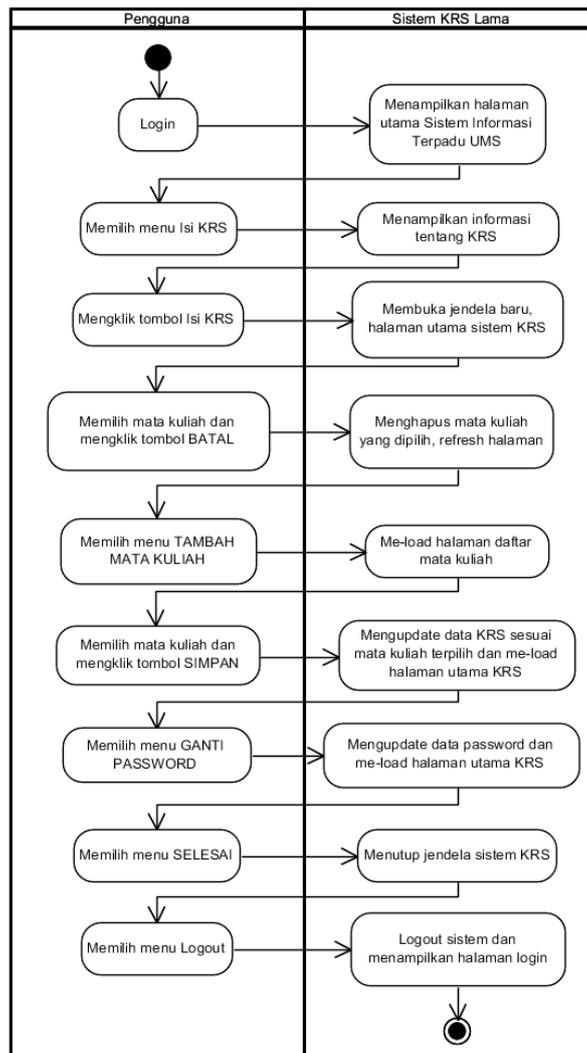
ping, didapatkan alamat IP sistem adalah 103.226.174.195. Alamat IP tersebut kemudian digunakan sebagai filter pada Wireshark dengan menggunakan perintah filter `ip.src==103.226.174.195 || ip.dst==103.226.174.195` (lihat Gambar 11). Filter ini bertujuan untuk menyaring data yang ditampilkan Wireshark yaitu hanya yang bersumber dari dan menuju ke alamat IP yang difilter.



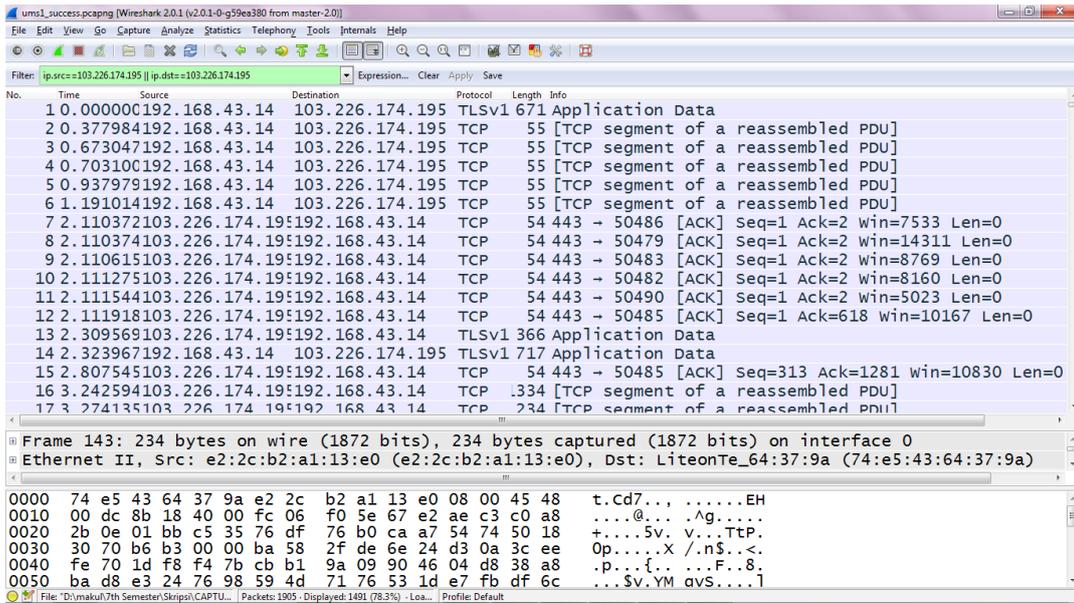
Gambar 11. Analisa dengan Filter `ip.src==103.226.174.195 || ip.dst==103.226.174.195`

Setelah memberi filter, langkah selanjutnya adalah menginisiasi proses *capturing* pada Wireshark. Ketika proses tersebut berjalan, barulah dilakukan aktivitas-aktivitas yang ada dalam aplikasi atau dalam sistem KRS *online*. Pada penelitian ini, proses *capturing* dilakukan selama terjadi aktivitas login, mengisi rencana studi, membatalkan data rencana studi, mengganti *password*, dan *logout*. Detil tahapan aktivitas yang dilakukan saat pengujian sistem KRS *online* diperlihatkan pada Gambar 12. Setelah seluruh tahapan aktivitas dikerjakan, proses *capturing* pada Wireshark diakhiri dan hasilnya disimpan pada sebuah berkas berekstensi *pcap* (*.pcap).

Aktivitas-aktivitas dibawah ini yang dianalisis menggunakan Wireshark. Wireshark akan menangkap semua data yang lewat ketika sistem diakses, tiap data tersebut memiliki informasi seperti waktu, alamat IP asal, alamat IP tujuan, protokol yang digunakan, panjang data (dalam *byte*) serta informasi lebih lanjut mengenai data tersebut. Informasi ini terlihat seperti pada Gambar 13. Setelah aktivitas yang dilakukan saat mengakses sistem telah selesai, maka selanjutnya menghentikan proses *capturing* pada Wireshark.



Gambar 12. Aktivitas Selama Mengakses Sistem KRS *Online*



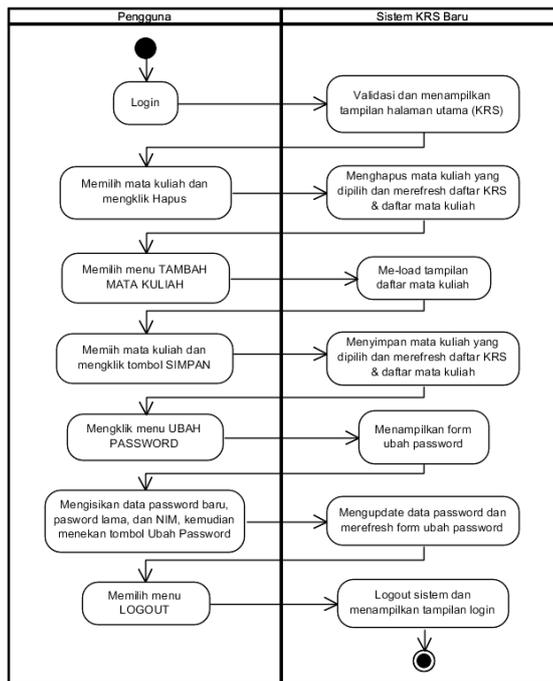
Gambar 13. Hasil Analisa Wireshark pada Sistem KRS Online

Dari hasil analisa dengan menggunakan Wireshark, didapatkan jumlah data yang ditransfer selama mengakses sistem tersebut adalah sebesar 670581 bytes atau 654,86 kilobytes. Jumlah tersebut terbagi dalam dua jenis data, yaitu data incoming adalah sebesar 531515 bytes atau 519,06 kilobytes dan data outgoing sebesar 139066 bytes atau 135,81 kilobytes. Selain itu, didapatkan data mengenai total waktu yang dibutuhkan selama mengakses sistem yaitu sebesar 178,70 detik dengan asumsi waktu direkam berdasarkan aktivitas-aktivitas yang dilakukan dari awal hingga akhir beserta posisi diam (*idle*) pada saat pengaksesan sistem.

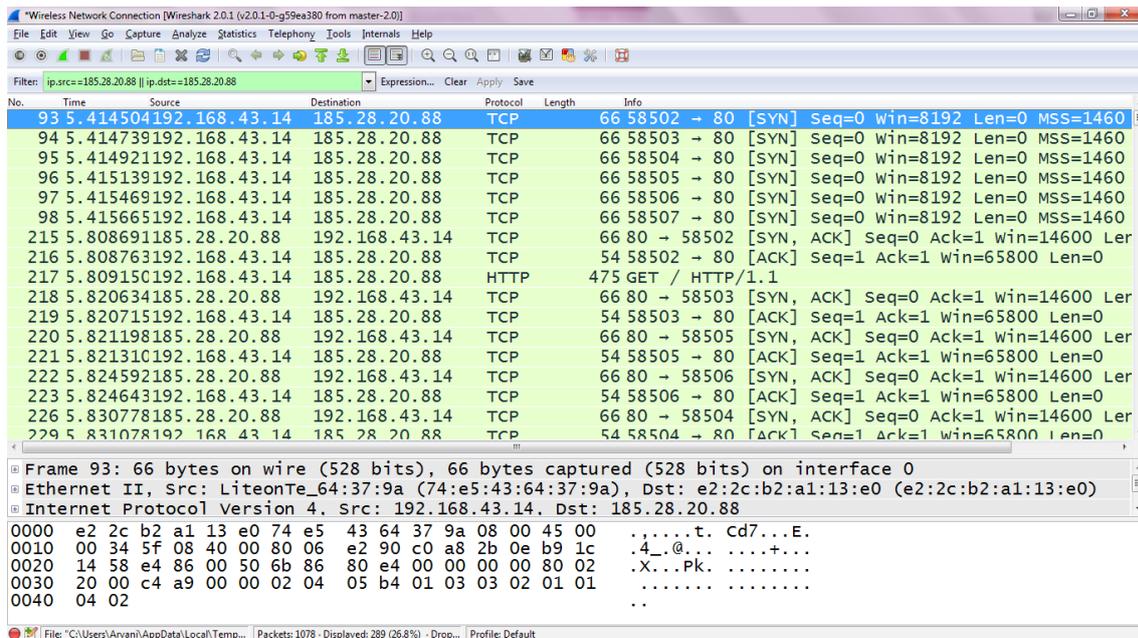
2) Pengujian pada Aplikasi SPA

Pengujian terhadap aplikasi SPA dilakukan dengan cara yang sama dengan pengujian pada sistem KRS

online. Hal pertama yang dilakukan adalah menentukan IP dari server tempat aplikasi dijalankan. Server yang digunakan untuk men-*deploy* aplikasi SPA mempunyai alamat IP 185.28.20.88. Alamat IP tersebut kemudian digunakan sebagai filter IP pada Wireshark dengan menggunakan perintah filter `ip.src==185.28.20.88 || ip.dst==185.28.20.88`. Langkah kedua pengujian adalah menginisiasi proses *capturing* pada Wireshark. Langkah ketiga adalah mengakses aplikasi SPA dengan melakukan tahapan aktivitas seperti pada Gambar 14. Setelah seluruh tahapan aktivitas dijalankan, langkah berikutnya adalah mengakhiri proses *capturing* Wireshark lalu menyimpan hasilnya ke sebuah berkas berekstensi pcap



Gambar 14. Aktivitas Selama Mengakses aplikasi SPA



Gambar 15. Hasil Analisa Wireshark pada aplikasi SPA

Aktivitas-aktivitas tersebutlah yang dianalisis menggunakan Wireshark. Hasil analisa menggunakan Wireshark meliputi informasi mengenai paket data yang ditransfer selama mengakses sistem seperti yang terlihat pada Gambar 15. Dari hasil analisa menggunakan Wireshark, didapatkan jumlah data yang ditransfer selama mengakses aplikasi SPA adalah 54182 bytes atau 52,91 kilobytes. Jumlah tersebut terbagi dalam dua jenis data, yaitu data *incoming* sebesar 27000 bytes atau 26,37 kilobytes dan data *outgoing* sebesar 27182 bytes atau 26,54 kilobytes. Selain itu didapatkan total waktu yang diperlukan selama mengakses sistem yaitu sebesar 64,71 detik dengan

asumsi waktu direkam berdasarkan aktivitas-aktivitas yang dilakukan dari awal hingga akhir beserta posisi diam (*idle*) pada saat mengakses sistem.

Data pengujian sistem KRS *online* dan aplikasi SPA kemudian dihipunkan ke dalam satu tabel untuk mempermudah perbandingan kinerja keduanya (Tabel 1). Pada tabel tersebut, total data transfer merupakan hasil penjumlahan antara data *incoming* dan *outgoing*. Sedangkan waktu transfer menunjukkan waktu yang dibutuhkan dalam melakukan keseluruhan aktivitas di mana waktu *idle* diasumsikan minimum karena pengujian dilakukan secepat mungkin.

Tabel 1. Kinerja Sistem KRS *online* UMS dan aplikasi SPA

Data kinerja untuk menyelesaikan seluruh tahapan aktivitas	Sistem KRS Online	Aplikasi SPA	Perbandingan kinerja (Sistem KRS Online: Aplikasi SPA)
Data Incoming (kilobyte)	519,06	26,37	20 : 1
Data Outgoing (kilobyte)	135,81	26,54	5 : 1
Total data transfer (kilobyte)	654,86	52,91	12 : 1
Waktu transfer (detik)	178,70	64,71	3 : 1
Rata-rata kecepatan transfer (kbps)	29,32	6,54	

3.3. Diskusi

Tabel 1 menampilkan hasil pengujian aplikasi SPA, yang dibuat untuk penelitian ini, dibandingkan dengan pengujian sistem KRS *online*, yang sedang digunakan di Universitas Muhammadiyah Surakarta. Data pengujian diperoleh menggunakan perangkat lunak Wireshark. Pengujian kinerja menggunakan Wireshark terutama dilakukan jika dibutuhkan detil data yang ditransfer antara server dan *client* (yaitu *browser* internet) beserta waktu kejadiannya, misalnya pengujian kualitas video streaming pada [6] dan pengujian kinerja server email pada [7]. Pada penelitian ini, Wireshark digunakan untuk mendapatkan jumlah data yang ditransfer saat dilakukan proses KRS, yaitu perekaman data rencana studi mahasiswa, baik pada aplikasi SPA maupun pada sistem KRS *online*.

Tabel data kinerja (Tabel 1) menunjukkan bahwa aplikasi SPA memerlukan transfer data sebesar 52,91 KB, yang lebih kecil dibanding transfer data pada sistem KRS *online* sebesar 654,86 KB. Terjadi perbaikan signifikan terkait jumlah data yang ditransfer untuk melakukan aktivitas yang setara. Perbaikan tersebut mungkin terjadi karena beberapa alasan. Pertama, jumlah tahapan kegiatan pada sistem KRS *online* lebih banyak dibanding pada aplikasi SPA untuk melakukan pekerjaan yang sama. Gambar 12 menunjukkan bahwa sistem KRS online membutuhkan sembilan langkah untuk melakukan satu proses login, menghapus satu mata kuliah dari daftar KRS, menambah satu mata kuliah yang lain, mengubah password dan melakukan logout. Sembilan langkah tersebut harus dilakukan oleh pengguna dan ditanggapi

oleh sistem. Sedangkan aplikasi SPA hanya membutuhkan tujuh langkah untuk melakukan hal yang sama (lihat Gambar 14). Pada sistem KRS *online*, setiap langkah membutuhkan respon berupa pemuatan (*loading*) halaman web sehingga jumlah langkah yang lebih banyak berarti jumlah data yang harus ditransfer dari server ke *browser* (yaitu data *incoming*) menjadi lebih banyak.

Data yang ditransfer ke server dari *browser* (atau data *outgoing*) pada sistem KRS *online* juga lebih besar sampai lima kali lipat lebih banyak dibanding data *outgoing* pada aplikasi SPA. Namun perbedaannya tidak seekstrem pada data *incoming*, yang mencapai 20 kali lipat. Hal ini karena data *outgoing* yang berasal dari *browser* tidak banyak berbeda antara sistem KRS *online* dan aplikasi SPA, yaitu berupa data perubahan daftar rencana studi (KRS) dan data permintaan perubahan *password*. Sebaliknya perbedaan muncul karena jumlah byte *overhead* yang dikirim oleh *browser* karena jumlah permintaan (*request*) yang berbeda terkait jumlah langkah aktivitas yang berbeda.

Selanjutnya waktu yang dibutuhkan untuk melakukan 9 langkah aktivitas pada sistem KRS *online* sebesar sekitar 3 menit sedangkan pada aplikasi SPA dibutuhkan hanya 1 menit. Sebenarnya perbedaan waktu tersebut tidak dapat diperbandingkan secara *fair* karena kedua aplikasi berada pada server yang berbeda di lokasi yang berbeda. Kecepatan transfer data saat akses sistem KRS online bisa mencapai 29,32 kbps sedangkan kecepatan akses ke server aplikasi SPA hanya 6,54 kbps. Walaupun begitu, karena data yang ditransfer pada akses sistem KRS *online* lebih banyak, maka waktu yang dibutuhkan menjadi lebih lama. Meskipun sistem KRS *online* didukung oleh perangkat dan *bandwidth* yang lebih baik, namun ternyata kinerjanya tidak lebih baik.

Uraian paragraf terdahulu memberikan indikasi bahwa jika infrastruktur yang melayani aplikasi SPA setara dengan infrastruktur yang melayani sistem KRS *online*, akan diperoleh penghematan *bandwidth* dan waktu baik dari sisi server maupun dari sisi pengguna. Indikasi itu sesuai dengan pernyataan yang dikemukakan Jadhav, dkk. [3] bahwa penerapan *Single Page Application* (SPA) dapat menghemat *bandwidth* serta waktu karena seluruh *resources* dibebankan pada pengaksesan sistem di awal dan berubah sesuai respon server atas interaksi klien tanpa menyebarkan seluruh halaman.

Penggunaan *bandwidth* yang kecil dan respon waktu yang singkat akan memberikan kenyamanan kepada pengguna aplikasi. Hal ini dapat memberikan pengalaman pengguna yang tinggi dan tidak akan mengecewakan ketika menggunakan SPA [8]. Selain itu penggunaan AngularJS untuk penerapan SPA membuat kode program mudah untuk dibaca dan dimengerti yang merupakan bagian paling penting dalam hal responsivitas antar muka pengguna [3].

4. Penutup

4.1. Kesimpulan

Penelitian ini menghasilkan aplikasi yang tergolong Single Page Application, yaitu aplikasi yang menempatkan

sebuah tugas yang terdiri atas beberapa tahap dalam satu halaman web dengan satu URL. Aplikasi yang dibuat terdiri atas lima tampilan yaitu tampilan login, tampilan utama, tampilan daftar rencana studi (KRS), tampilan tambah mata kuliah dan tampilan ubah *password*. Kinerja aplikasi SPA dibandingkan dengan kinerja sistem KRS *online* yang masih digunakan. Pengukuran data kinerja dilakukan menggunakan Wireshark. Hasil penelitian menunjukkan bahwa aplikasi SPA memerlukan transfer data yang jauh lebih sedikit dibanding sistem KRS *online* untuk melakukan pekerjaan yang setara. Tahapan yang harus ditempuh lebih sedikit dan waktu yang dibutuhkan untuk menyelesaikan pekerjaan yang sama juga lebih sedikit. Artinya, penerapan *single page application* pada proses KRS terbukti dapat memperbaiki kinerja sistem dengan mengurangi bandwidth yang dibutuhkan dan mengurangi waktu total proses KRS.

4.2. Saran

Penelitian ini berfokus pada jumlah data yang ditransfer dan waktu untuk melakukan proses KRS. Sebelum konsep ini diterapkan, ada baiknya dilakukan penelitian tentang keamanan aplikasi karena SPA menggunakan Javascript yang kodenya dapat dilihat/dibaca oleh pengguna yang mempunyai pengetahuan tentang pemrograman.

5. Daftar Pustaka

- [1] Tim Penyusun, *Buku Panduan Akademik*, Surakarta: Fakultas Komunikasi & Informatika, 2012.
- [2] _____, *Sistem Informasi Akademik*, 2014. Diakses dari <https://akademik.ums.ac.id/> tanggal 25 Maret 2016.
- [3] M. A. Jadhav, B.R. Sawant & A. Deshmukh, "Single Page Application using AngularJS," dalam *International Journal of Computer Science and Information Technologies*, 6 (3), hal. 2876, 2015.
- [4] M.S. Mikowski & J.C. Powell, *Single Page Web Application*. Shelter Island: Manning Publications Company, 2014.
- [5] U. Lamping, R. Sharpe & E. Warnicke, *Wireshark User's Guide: For Wireshark 2.1*. Boston: The Free Software Foundation, 2014.
- [6] D.T. Massandy, & I.R. Munir, "Secured video streaming development on smartphones with android platform," In *Telecommunication Systems, Services, and Applications (TSSA), 7th International Conference on*, hal. 339-344, IEEE, 2012.
- [7] H. Thamrin, "Beban Jaringan Saat Mengakses Email Dari Beberapa Mail Server," in *Prosiding Seminar Nasional Sains Dan Teknologi Fakultas Teknik*, 1(1), 2010.
- [8] R.J. Joseph, "Single Page Application and Canvas Drawing," *International Journal of Web & Semantic*, 6 (1): 29 – 30, 2015.