

# MODEL HALAMAN TUNGGAL UNTUK PENINGKATAN KINERJA APLIKASI PEREKAMAN DATA PENUGASAN JABATAN STRUKTURAL

**Husni Thamrin**

Program Studi Informatika, Universitas Muhammadiyah Surakarta  
Jl. A Yani, Pabelan, Surakarta, Telp. (271)717417  
husni.thamrin@ums.ac.id

## **Abstract**

*Convenience is one of the key factors that support the success of information system implementation, and it is affected by how the system supports users to quickly get their job done. For the case of institution XYZ, the process of data recording of a managerial mutation requires a number of steps. System operators found it cumbersome to go through too many steps when recording such data because it takes a long time to complete the task. The situation is exacerbated for recording a decree with several job assignment changes. To overcome this problem, we have developed an application that applies single page model and replaced the legacy system that applies standard web pages. The use of a single page web-based applications reduces the amount of data that are transferred between client and server so it is more efficient in terms of processing speed compared to the standard web application. The implementation of single page application in institution XYZ drops the number of data transferred to the level of 62% for recording a single job assignment. The percentage level is smaller when a process contains a lot of job assignment, as happened in the case of large-scale mutation.*

**Keywords:** *single page application, job assignment, information system*

## **Abstrak**

*Kenyamanan penggunaan sistem informasi merupakan salah satu faktor kesuksesan implementasi sistem, dan hal ini dipengaruhi oleh dukungan sistem agar pengguna dapat menyelesaikan pekerjaan dengan cepat. Proses perekaman data pergantian jabatan struktural pada aplikasi yang diterapkan di lembaga XYZ memerlukan beberapa tahap aktivitas untuk setiap data jabatan baru. Teknik ini memberi masalah terutama ketika merekam sebuah surat keputusan dengan banyak perubahan data penugasan, yaitu lamanya waktu untuk menyelesaikan pekerjaan. Untuk mengatasi permasalahan tersebut, diterapkan model halaman tunggal menggantikan model halaman web standar. Penggunaan model halaman tunggal pada aplikasi berbasis web mengurangi jumlah data yang harus ditransfer antara server dan client sehingga lebih efisien dari segi kecepatan proses dibanding penerapan model halaman standar. Pada aplikasi yang diterapkan di lembaga XYZ, perekaman satu data jabatan dengan model halaman tunggal memerlukan hanya 62% jumlah data yang ditransfer antara client dan server. Prosentase akan mengecil jika sebuah proses mengandung banyak data penugasan, seperti yang terjadi dalam kasus mutasi jabatan besar-besaran.*

**Kata kunci:** *aplikasi halaman tunggal, single page application, penugasan jabatan, sistem informasi*

## 1. PENDAHULUAN

Implementasi sistem informasi pada suatu lembaga tidak selalu berjalan mulus dan berhasil. Permasalahan bisa muncul dari sisi teknis, sosial dan organisasional [1]. Masalah sosial dan organisasional harus diselesaikan oleh pihak manajemen. Manajer atau pimpinan perusahaan harus mampu meyakinkan berbagai pihak bagaimana sistem yang baru merubah pola kegiatan lembaga, bagaimana visi dapat dicapai sesuai *road map* yang disusun, termasuk dengan memberi *reward* kepada pihak yang mendukung dalam implementasi sistem yang baru [2,3]. Sedangkan masalah teknis dapat diselesaikan oleh pengembang sistem yang melibatkan analis, desainer dan programer. Para engineer perlu memperhatikan berbagai faktor yang mempengaruhi kesuksesan implementasi sistem, seperti faktor kegunaan (*usability*), fleksibilitas, dan kinerja sistem [4].

Kinerja sistem yang baik dapat diperoleh jika laju aliran informasi tinggi, dan struktur informasi relatif sederhana [5]. Artinya, selang waktu yang dibutuhkan untuk menjalankan sebuah tugas haruslah singkat. Secara naluriah, seseorang ingin agar dapat menyelesaikan tugas dengan cepat. Interaksi yang nyaman memungkinkan seseorang memiliki daya tahan dalam bekerja sehingga dapat mengerjakan lebih banyak tugas dalam selang jam kerjanya. Hambatan terkait kinerja sistem terjadi di lembaga XYZ yang belum lama ini menerapkan sebuah sistem informasi berbasis web untuk pengelolaan sumber daya manusia. Sebagian operator lapangan mengeluhkan piranti lunak yang baru diimplementasikan karena operator merasa tidak nyaman dan membutuhkan waktu yang terlalu lama untuk melakukan pekerjaan perekaman data penugasan jabatan.

Pada sistem informasi berbasis web, terdapat beberapa faktor yang mempengaruhi waktu untuk menyelesaikan tugas. Faktor yang dimaksud adalah kecepatan aliran data dari server dan menuju server (*network delay*), kecepatan penyiapan data oleh server (*server response* atau *server load*), kecepatan operator dalam bekerja, dan ukuran data yang dialirkan dalam interaksi [6]. Untuk menghitung kecepatan penyelesaian sebuah tugas, kesemua faktor tersebut perlu diukur, atau setidaknya diprediksi.

Analisis terhadap sistem yang diterapkan di lembaga XYZ menunjukkan bahwa pengembang menggunakan *template* bawaan dari *framework* yang ada untuk menerapkan proses perekaman data. *Template* tersebut menggunakan formulir HTML standar sebagai antar muka. Selanjutnya telah diupayakan perbaikan terhadap sistem dengan menerapkan model proses yang berbeda, yaitu dengan model halaman tunggal. Aplikasi halaman tunggal dalam sistem berbasis web memerlukan implementasi *client side scripting* misalnya menggunakan JavaScript atau dalam bentuk AJAX (*Asynchronous JavaScript and XML*). Sedangkan aplikasi web standar dapat diwujudkan tanpa teknologi AJAX, yaitu melalui halaman-halaman dengan formulir HTML (*hypertext mark-up language*).

Tulisan ini mendeskripsikan penggunaan aplikasi halaman tunggal (*single page application*) untuk merekam data penugasan jabatan struktural di lembaga XYZ, dan bagaimana model halaman tunggal dapat memperbaiki kecepatan proses dibanding penggunaan halaman standar yang semula diterapkan.

Model halaman tunggal merupakan model sistem yang merujuk pada aplikasi berbasis web yang hanya menampilkan satu halaman saja untuk

memproses sebuah tugas. Model ini lebih dikenal dengan *Single Page Application* (SPA). Model yang kontras dengan SPA adalah aplikasi standar dengan formulir HTML. Model SPA menjadi terasa perbedaannya dengan model standar jika proses yang diolah oleh sistem informasi mempunyai sekuens atau terdiri atas beberapa sub-aktivitas. Meskipun terdapat banyak sub-aktivitas, semua komponen utama pada SPA, seperti CSS, skrip dan sumber daya lain yang diperlukan diambil pada satu selang waktu. Komponen atau konten yang sesuai diambil secara dinamis tergantung pada kebutuhan interaksi atau permintaan pengguna [7].

Pada model standar, setiap sub-aktivitas memerlukan halaman web dengan formulir HTML yang berlainan. 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 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. Perilaku seperti ini menyebabkan beban server bertambah dan terjadi pengiriman data secara berulang. Kecepatan sistem dalam menanggapi interaksi dengan pengguna menjadi lambat, baik karena beban server maupun karena kebutuhan bandwidth yang lebih besar. Sistem dengan model SPA dengan demikian dapat memberi respons yang lebih cepat sehingga menjadi lebih nyaman bagi pengguna [8].

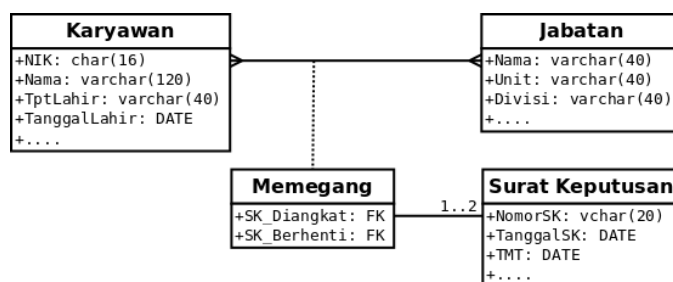
Model halaman tunggal direalisasikan dengan kode program yang dijalankan di komputer pengguna (*client side scripting*). Konsekuensinya adalah penggunaan bahasa pemrograman JavaScript, jika menggunakan teknologi web pada sebuah browser. Alternatif lain adalah penggunaan teknologi flash (dari Adobe), atau menggunakan framework JavaScript semacam AngularJS.

## 2. METODE PENELITIAN

Metode penelitian yang digunakan adalah eksperimen dan observasi, yaitu dengan membuat dua buah aplikasi. Aplikasi pertama mewujudkan model halaman tunggal (SPA) dan aplikasi kedua mewujudkan model standar. Kinerja kedua aplikasi diukur dari jumlah data yang ditransfer antara server dan *browser* pada proses perekaman data penugasan jabatan struktural di sebuah lembaga. Selanjutnya, kinerja aplikasi diukur dari prediksi waktu yang dibutuhkan untuk menyelesaikan proses perekaman data.

### 2.1. Model data

Model data menunjukkan bagaimana data ditata di dalam aplikasi atau dalam sistem basis data. Model data yang diterapkan pada lembaga XYZ dapat digambarkan dengan *class diagram* seperti pada 0 Diagram tersebut mengandung tiga kelas dan satu relasi. Ketiga kelas adalah Karyawan, Jabatan dan Surat Keputusan. Kelas Karyawan dan Jabatan memiliki relasi memegang sebagai wujud kalimat relasional "Karyawan memegang Jabatan". Relasi ini bersifat *many-to-many* dengan beberapa atribut dan merupakan wujud dari proses penugasan oleh lembaga kepada karyawan.



Gambar 1. Class Diagram pada aplikasi penugasan jabatan

Kelas Karyawan dan kelas Jabatan merupakan tabel master dalam aplikasi ini. Kedua kelas mempunyai banyak sekali atribut, dan hanya beberapa yang ditampilkan pada gambar. Kelas Surat Keputusan digunakan untuk menyimpan SK yang dikeluarkan pimpinan lembaga terkait pengangkatan pemberhentian jabatan. Setiap proses penugasan dan proses penghentian dari penugasan dikukuhkan dengan Surat Keputusan (SK) dari pimpinan yang berwenang. Oleh karena itu relasi “Memegang” mempunyai atribut SK\_diangkat dan SK\_berhenti yang masing-masing digunakan untuk mencatat SK pengangkatan dan pemberhentian karyawan dalam jabatan tertentu. Setiap SK mengandung setidaknya tiga informasi yaitu nomor SK, tanggal terbitnya SK dan tanggal efektif (dikenal dengan istilah TMT = terhitung mulai tanggal). Atribut SK\_berhenti dan SK\_diangkat pada relasi Memegang merupakan kunci asing (*foreign key*) menuju kelas Surat Keputusan.

Model seperti pada 0 dipilih karena satu surat keputusan dapat mengandung pengangkatan dan pemberhentian jabatan. Tidak jarang pula satu surat keputusan mengandung pengangkatan dan pemberhentian banyak orang dari banyak jabatan (misalnya dalam kasus mutasi besar-besaran).

## 2.2. Model proses halaman standar

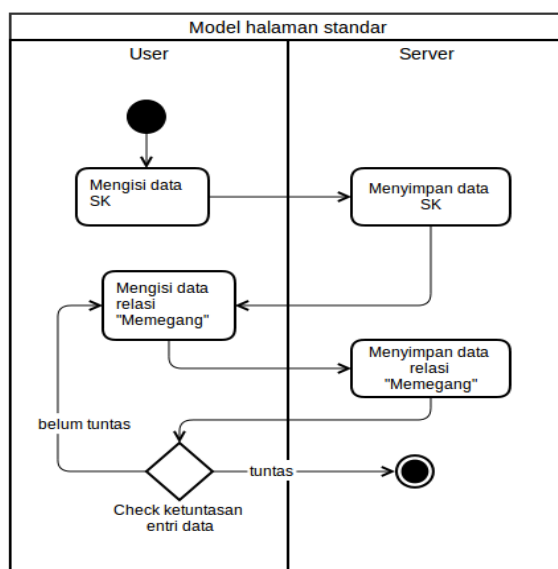
Berdasarkan model data pada 0, dapat dibuat alur proses ketika dilakukan perekaman data pengangkatan jabatan dan pemberhentian jabatan. Model proses yang digunakan di lembaga XYZ dapat ditulis dalam bentuk diagram aktivitas (*activity diagram*).

Kelas Karyawan dan kelas Jabatan merupakan tabel master dalam sistem ini dan telah memiliki data sebelum aplikasi di-*deploy*. Kelas Surat Keputusan merupakan bagian dari transaksi terkait relasi Memegang sehingga akan diisi jika ada proses pengangkatan maupun pemberhentian jabatan sebelum dilakukan pengisian data untuk relasi Memegang.

Langkah-langkah seperti pada 0 diterapkan pada aplikasi standar. Kegiatan yang pertama dilakukan adalah mengisi data surat keputusan mulai dari nomor SK, tanggal SK, maupun tanggal berlaku SK, termasuk mengunggah berkas SK. Aplikasi memanfaatkan formulir HTML sebagai antar muka. Setelah data diisi, user harus mengklik tombol *submit* dan kemudian server akan menyimpan data tersebut di basis data.

Selanjutnya user mengisi data penugasan untuk satu karyawan pada suatu jabatan (bisa dalam bentuk pengangkatan atau pemberhentian). Sekali lagi formulir HTML digunakan di sini sebagai antar muka. Setelah data terisi, user mengklik tombol *submit* dan server akan menyimpan data di basis data.

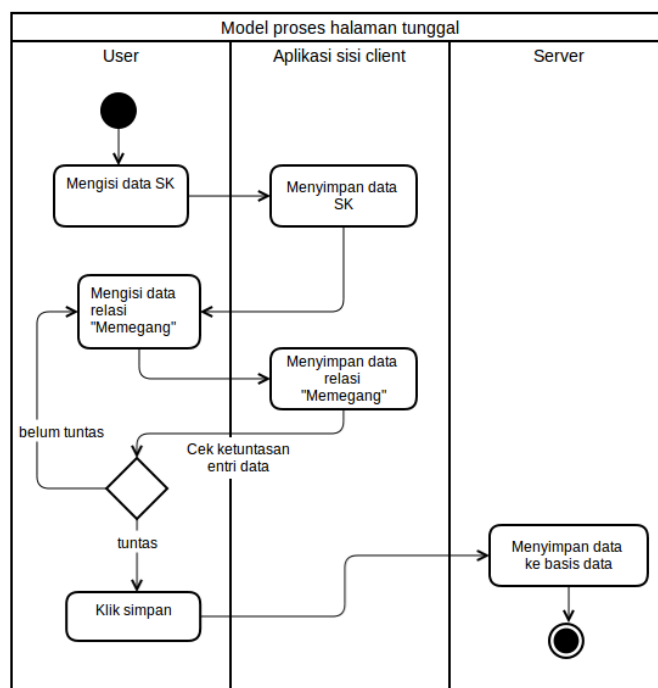
Proses yang disebutkan pada paragraf terakhir dapat berulang jika satu surat keputusan berisi banyak data penugasan. Semisal satu surat keputusan berisi pengangkatan 5 karyawan pada 5 jabatan, maka proses pada paragraf di atas akan berulang 5 kali.



Gambar 2. Diagram aktivitas pengisian data penugasan pada aplikasi standar

### 2.3. Model proses halaman tunggal

Model proses untuk aplikasi halaman tunggal mengikuti langkah-langkah pada diagram aktivitas 0. yang berlaku baik untuk pengangkatan maupun pemberhentian jabatan.



Gambar 3. Diagram aktivitas pengisian data penugasan pada aplikasi halaman tunggal

Seperti juga pada model proses pada aplikasi standar, kegiatan yang pertama dilakukan adalah user mengisi data surat keputusan mulai dari nomor SK, tanggal SK, maupun tanggal berlaku SK dan unggah berkas SK. Aplikasi dapat memanfaatkan formulir HTML untuk isian data di atas.

Pada halaman yang sama terdapat dua kotak edit untuk isian karyawan dan jabatan. Setelah kedua kotak diisi, user menekan sebuah tombol (misalnya tombol “pasangkan”). Penekanan tombol menyebabkan aplikasi membuat objek yang memasang karyawan dan jabatan. Proses terakhir ini dilakukan dengan JavaScript (atau kode program *client side scripting* lainnya).

Proses pada paragraf di atas dapat diulang sebanyak jumlah data penugasan pada sebuah surat keputusan. Setelah proses ini selesai, user mengklik tombol *submit* dan kemudian server akan menyimpan data tersebut di basis data.

## 2.4. Pengukuran kinerja

Upaya mengukur kinerja kedua model proses dilakukan dengan dua langkah. Langkah pertama adalah implementasi kedua model menggunakan bahasa pemrograman yang sama, *framework* yang sama, *template* HTML yang sama dan mengakses ke basis data yang sama. Langkah kedua adalah dengan menjalankan aplikasi dan mengukur seberapa besar data yang harus ditransfer antara server dan *client*. Idealnya, kinerja aplikasi web dilihat dari *steady-state throughput* dan *response times* [9] dan kedua variabel tersebut dipengaruhi oleh kecepatan transfer data di jaringan dan kecepatan server menanggapi *request*. Namun kecepatan transfer dan kecepatan server diasumsikan konstan selama pengamatan sehingga

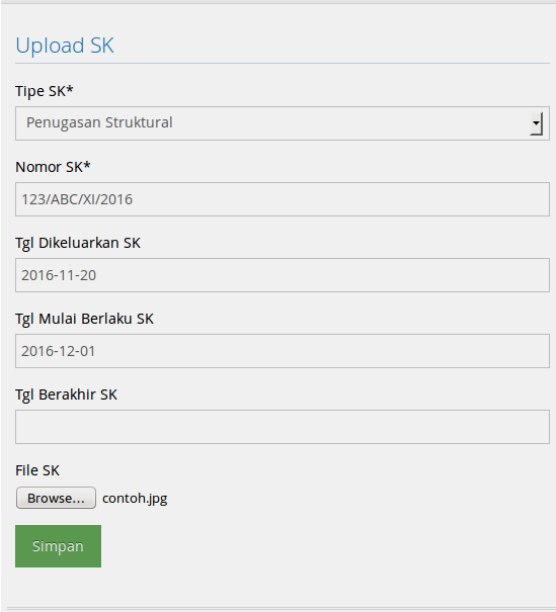
tidak memberi pengaruh yang berbeda terhadap kinerja aplikasi yang menerapkan kedua model proses.

Perangkat lunak (*tools*) yang digunakan dalam mengukur jumlah data yang ditransfer adalah fasilitas “Network” dari *browser* Mozilla Firefox. Fasilitas tersebut terdapat pada menu “Web Developer” yaitu menu untuk para pengembang halaman web dalam menguji aplikasi mereka. Perangkat ini dipilih karena relatif sederhana, dibandingkan Wireshark misalnya, dan sudah memadai untuk kebutuhan pengamatan. Keunggulan dari penggunaan fasilitas *browser* adalah sinkronisasi waktu memulai proses monitoring dan waktu berhenti. Selain itu, data yang terekam pada layar monitoring hanya data yang terkait dengan aktivitas pada halaman *browser* yang sedang diamati. Jika digunakan Wireshark atau perangkat semacam itu, kerja perangkat tidak sinkron dengan *browser* dan seluruh aktivitas di jaringan akan direkam sehingga harus dilakukan pemilahan hasil monitoring yang berasal dari halaman browser yang diamati dan yang berasal dari halaman lain atau aplikasi lain yang menggunakan jaringan.

### 3. PEMBAHASAN

#### 3.1. Hasil implementasi

Implementasi dari model proses halaman standar ditunjukkan pada 0 dan 0, yaitu halaman untuk mengisi data surat keputusan dan mengisi data penugasan. Proses mengisi data surat keputusan mencakup unggah berkas SK (bisa dalam bentuk salinan pindai atau berkas berformat pdf), pengisian nomor SK, tanggal SK diterbitkan dan tanggal SK mulai berlaku. Jika ada, tanggal SK berakhir juga dapat diisikan. Pengisian data penugasan mencakup pemilihan SK (yang telah diisikan sebelumnya), pengisian data karyawan dan nama jabatan.



Upload SK

Tipe SK\*  
Penugasan Struktural

Nomor SK\*  
123/ABC/XI/2016

Tgl Dikeluarkan SK  
2016-11-20

Tgl Mulai Berlaku SK  
2016-12-01

Tgl Berakhir SK

File SK  
Browse... contoh.jpg

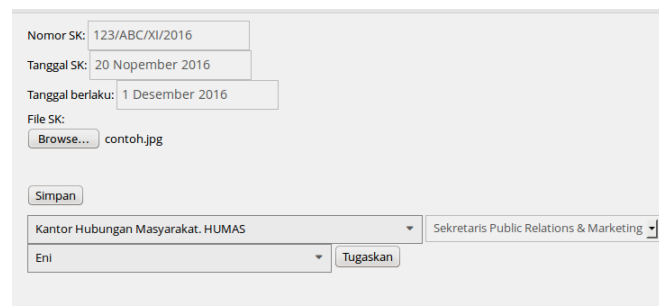
Simpan

Gambar 4. Tampilan halaman standar untuk pengisian data penugasan



Gambar 5. Tampilan halaman standar untuk pengisian relasi “Memegang”

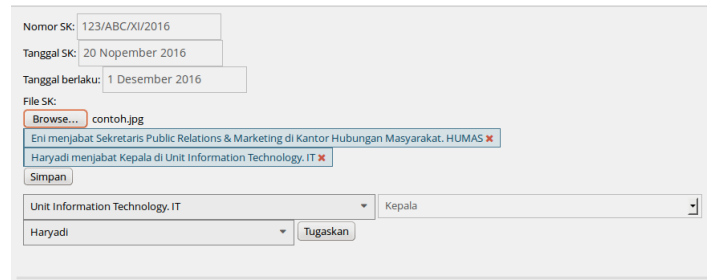
Implementasi dari model proses halaman tunggal ditunjukkan pada 0. Halaman web dapat dipilah menjadi dua bagian. Bagian pertama digunakan untuk mengisikan nomor SK, tanggal SK diterbitkan, tanggal SK berlaku, dan fasilitas mengunggah berkas SK. Bagian kedua digunakan untuk menambahkan data karyawan dan jabatan yang ditugaskan. Bagian pertama berada di sisi atas dan bagian kedua ada di sisi bawah.



Gambar 6. Tampilan halaman tunggal pengisian data dan SK penugasan

Ketika pengguna menambahkan data penugasan karyawan, pada bagian tengah akan muncul daftar karyawan dan jabatannya (lihat 0). Daftar ini dapat bertambah secara dinamis sesuai jumlah karyawan yang tercantum dalam surat keputusan. Jika pengguna melakukan kesalahan entri, *item* yang salah pada daftar dapat dihapus. Implementasi dari proses penambahan dan penghapusan data penugasan dilakukan dengan fungsi JavaScript. Daftar jabatan pada suatu unit diambil secara dinamis dengan menggunakan teknologi JSON (*JavaScript Object Notation*).





Gambar 7. Tampilan halaman tunggal dengan data penugasan karyawan di bagian tengah

### 3.2. Hasil pengamatan

Setelah aplikasi selesai dibuat, dilakukan pengamatan terhadap kinerja aplikasi untuk dua model proses, yaitu model proses standar dengan menggunakan formulir HTML dan model aplikasi halaman tunggal (*single page application* – SPA). Kinerja diukur dengan memonitor jaringan menggunakan fasilitas “Network” pada menu “Web Developer” dari aplikasi Mozilla Firefox. Aktivitas yang dimonitor setidaknya ada tujuh jenis, sesuai dengan banyaknya jenis aktivitas transfer data dari server dan client pada dua model proses yang diamati. Ketujuh aktivitas itu terdiri atas empat aktivitas pada model proses standar dan tiga aktivitas pada model SPA. Empat aktivitas pada model proses standar adalah:

- MPS1 - pengambilan halaman web berisi formulir untuk mengisi data surat keputusan,
- MPS2 - penyimpanan data surat keputusan,
- MPS3 - pengambilan halaman web berisi formulir untuk mengisi data penugasan,
- MPS4 - penyimpanan data penugasan.

Tiga aktivitas yang dimonitor merupakan aktivitas pada model SPA yaitu:

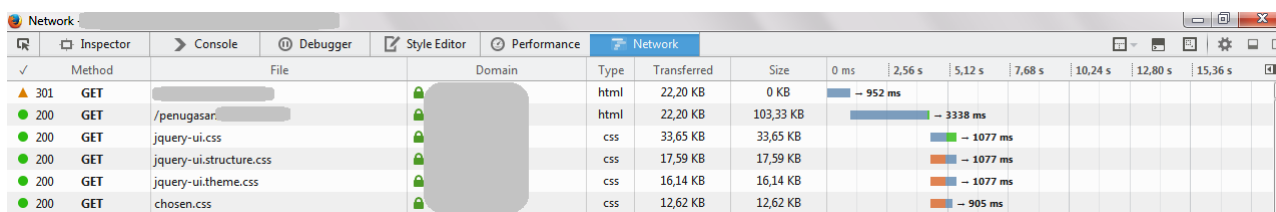
- SPA1 - pengambilan halaman web berisi aplikasi halaman tunggal untuk pengisian data surat keputusan dan data penugasan,
- SPA2 - proses pengisian data penugasan
- SPA3 - penyimpanan data surat keputusan dan data penugasan.

Semua aktivitas (kecuali aktivitas SPA2) mengandung proses pemuatan halaman web baru sehingga menghasilkan data monitoring baru. Data hasil monitoring pada Mozilla Firefox terdiri atas beberapa kolom atau atribut.

Aktivitas bagi pengembang web (*web developer*), termasuk monitoring jaringan, pada *browser* Mozilla Firefox difasilitasi dengan sebuah *window*. Tampilan monitoring jaringan ada pada tab Network dari *window* tersebut berwujud sebuah tabel dengan banyak kolom, seperti kolom kode respon server, metode transfer data (seperti GET atau POST), berkas, tipe berkas, dan diagram waktu (lihat 0). Dua kolom sebagai sumber data yang dianalisis dalam penelitian ini adalah data yang ditransfer (*transferred*) dan ukuran data (*size*). Data yang ditransfer menunjukkan jumlah data sebenarnya yang ditransfer melalui jaringan, baik data dari *client* ke server (*upstream*) maupun *downstream*. Ukuran data menunjukkan ukuran data

yang diperoleh (bersifat *downstream*) yang bisa lebih besar dari pada data yang ditransfer jika proses transfer melibatkan proses kompresi.

Kolom diagram waktu menunjukkan kapan sebuah data mulai diminta (*request*) dan kapan data tersebut selesai diperoleh. Diagram waktu berbentuk batang horizontal dengan 5 macam warna. Kelima macam warna mengindikasikan lima macam proses, yaitu menemukan alamat DNS dari server (*DNS Resolution*), menghubungi server (*Connecting*), mengirim data (*Sending*), menunggu respon (*Waiting*), dan menerima data (*Receiving*). Pada penelitian ini diasumsikan bahwa kinerja server dan jaringan konstan dan stabil sehingga proses menemukan DNS, menghubungi server dan menunggu respon selalu tetap, tidak berbeda untuk kedua macam aplikasi yang diamati. Perbedaan kinerja dipengaruhi hanya oleh proses mengirim data dan menerima data dan perbedaan itu muncul terutama karena ukuran data yang dikirim atau diterima.



Gambar 8. Contoh hasil monitoring jaringan

Tabel 1 memperlihatkan ukuran data yang dikirim atau diterima pada setiap aktivitas. Kolom "Ukuran data" dibagi menjadi dua. Kolom "Server (*cached*)" menunjukkan jumlah data sebenarnya yang ditransfer dari dan ke server jika *browser* sudah menyimpan sebagian data (atau berkas) di dalam *cache*-nya. Kolom "Total" menunjukkan total data yang harus ditransfer untuk menyelesaikan aktivitas, yang sebagian data bisa berasal dari *cache*. Angka pada kolom "Server (*cached*)" untuk aktivitas MPS2, MPS4 dan SPA3 merupakan ukuran data yang dikirim ke server pada proses menyimpan data SK atau penugasan (yaitu proses POST). Angka pada kolom "Total" mencakup data yang diterima *browser* untuk membentuk halaman lanjutan setelah proses POST selesai. Pada kegiatan operasional yang berulang, tentu saja kebanyakan berkas sudah di-*cache* oleh *browser* (jika ukuran *harddisk* memadai).

Tabel 1. Data yang ditransfer di tiap aktivitas

No	Aktivitas	Ukuran data (D, dalam KB)	
		Server ( <i>cached</i> )	Total
1	MPS1	20,08	1643,77
2	MPS2	45,43	1125,28
3	MPS3	167,3	1512,63
4	MPS4	45,26	1730,39
5	SPA1	125,6	1350,1

No	Aktivitas	Ukuran data (D, dalam KB)	
		Server ( <i>cached</i> )	Total
6	SPA2	0,75	0,75
7	SPA3	45,47	1743,35

Berdasarkan hasil pengamatan pada Tabel 1, dapat dihitung jumlah data yang harus ditransfer untuk merekam data penugasan seorang karyawan. Untuk model standar, data yang harus ditransfer dapat dihitung sebagai berikut.

$$D_{MPS} = D_{MPS1} + D_{MPS2} + D_{MPS3} + D_{MPS4}$$

$$= 287,07 \text{ KB}$$

Sedangkan untuk model halaman tunggal, data yang ditransfer adalah:

$$D_{SPA} = D_{SPA1} + D_{SPA2} + D_{SPA3}$$

$$= 171,82 \text{ KB}$$

Hasil perhitungan di atas menunjukkan bahwa proses yang menerapkan model halaman tunggal membutuhkan transfer data yang lebih sedikit dibanding proses dengan halaman standar, yaitu hanya 62%. Hal ini memberi konsekuensi proses pengerjaan yang secara signifikan akan lebih cepat.

### 3.3. Diskusi

Model proses yang berbeda telah diterapkan terhadap model data yang sama. Model proses digambarkan dengan diagram aktivitas (*activity diagram*) sedangkan model data digambarkan dengan diagram kelas (*class diagram*). Implementasi model proses yang berbeda diwujudkan dalam bentuk tampilan aplikasi yang berbeda.

Diagram kelas pada 0 bukanlah satu-satunya model yang dapat dirancang. Kelas "Surat Keputusan" dapat dihilangkan dan atribut lebih detil terkait surat keputusan dapat disisipkan sebagai atribut langsung dari relasi Memegang. Perubahan model data ini akan menyebabkan alur aktivitas menjadi berubah, begitu pula diagram aktivitasnya. Alur aktivitas akan lebih sederhana untuk kasus di mana setiap penugasan dan pemberhentian jabatan berkaitan dengan satu dan satu-satunya Surat Keputusan. Namun jika satu Surat Keputusan dapat mengandung pengangkatan dan pemberhentian beberapa orang sekaligus, maka model seperti ini akan menyebabkan terjadinya redundansi pada pencatatan nomor SK, tanggal SK dan tanggal berlaku SK. Kesulitan akan bertambah ketika berkas SK (misalnya dalam bentuk salin pindai) harus diunggah dan disimpan di server.

Dengan menggunakan model data yang telah dipilih, terbukti bahwa penggunaan model halaman tunggal lebih efisien dibanding penggunaan model halaman standar. Untuk merekam data penugasan seorang karyawan, model halaman standar membutuhkan transfer data sebesar 287,07 KB dan model halaman tunggal memerlukan transfer data sebesar 171,82 KB.

Efisiensi akan menjadi lebih terasa jika sebuah SK berisi data penugasan untuk banyak karyawan. Pada model halaman standar, untuk mengisi data penugasan n orang karyawan diperlukan perulangan pada aktivitas MPS3 dan MPS4 sehingga jumlah data yang ditransfer menjadi:

$$D_{MPS} = D_{MPS1} + D_{MPS2} + n (D_{MPS3} + D_{MPS4})$$

Sedangkan pada model halaman tunggal, diperlukan transfer data sebesar

$$D_{SPA} = D_{SPA1} + D_{SPA3} + n D_{SPA2}$$

Sebagai contoh jika pada sebuah SK penugasan terdapat mutasi 10 orang karyawan, maka jumlah data yang ditransfer menggunakan model halaman standar adalah 2191,11 KB sedangkan data yang ditransfer menggunakan model halaman tunggal adalah sejumlah 178,57 KB. Itu berarti pengurangan yang sangat signifikan hingga lebih dari 90%.

Hasil penelitian yang didiskusikan di atas sangat berseduaian dengan apa yang dihasilkan dalam berbagai penelitian lain, misalnya oleh [7] dan [10]. Kedua penelitian itu menyimpulkan bahwa aplikasi halaman tunggal memerlukan transfer data yang jauh lebih sedikit dibanding sistem dengan model standar untuk melakukan pekerjaan yang setara. Untuk konteks yang dicermati, tahapan yang harus ditempuh lebih sedikit dan waktu yang dibutuhkan untuk menyelesaikan pekerjaan yang sama juga lebih sedikit.

#### 4. SIMPULAN

Untuk menyelesaikan pekerjaan yang sama, dapat digunakan model data yang berbeda dan untuk model data yang sama dapat diterapkan model proses yang berbeda. Model halaman tunggal telah diterapkan untuk menggantikan model halaman standar pada proses perekaman data penugasan jabatan di lembaga XYZ. Model halaman tunggal tersebut menerapkan fungsi JavaScript dan teknologi JSON. Untuk merekam satu data penugasan jabatan, model halaman tunggal hanya mentransfer 62% jumlah data yang ditransfer pada model halaman standar.

#### DAFTAR PUSTAKA

- [1] Cresswell, K., dan Sheikh, A., "**Organizational issues in the implementation and adoption of health information technology innovations: an interpretative review**," International journal of medical informatics, vol. 82, no. 5, hal. e73-e86, 2013.
- [2] Fitzgerald, M., Kruschwitz, N., Bonnet, D., dan Welch, M., "**Embracing digital technology: A new strategic imperative**," MIT sloan management review, vol. 55, no. 2, 2014.
- [3] Knight, R., "**Convincing Skeptical Employees to Adopt New Technology**," Harvard Business Review, 2015. <https://hbr.org/2015/03/convincing-skeptical-employees-to-adopt-new-technology>. dilihat 10 Januari 2017
- [4] Cresswell, K.M., Bates, D.W., dan Sheikh, A., "**Ten key considerations for the successful implementation and adoption of large-scale health information technology**," Journal of the American Medical Informatics Association, vol. 20, no.1, hal. e9-e13, 2013.
- [5] Wilson, G.F., dan Russell, C.A., (2003). "**Real-time assessment of mental workload using psychophysiological measures and artificial neural networks**," Human Factors: The Journal of the Human Factors and Ergonomics Society, vol. 45, no. 4, hal. 635-644, 2003.
- [6] Krishnamurthy, B., dan Wills, C.E., "**Analyzing factors that influence end-to-end Web performance**," Computer Networks, vol. 33, no. 1, hal. 17-32, 2000.

- [7] Jadhav, M.A., Sawant, B.R., dan Deshmukh, A., "**Single Page Application using AngularJS**," International Journal of Computer Science and Information Technologies, vol. 6, no. 3, 2015.
- [8] Mikowski, M. S., dan Powell, J. C. "**Single page web applications**," B and W, 2013.
- [9] Shoaib, Y., dan Olivia D., "**Web application performance modeling using layered queueing networks**," Electronic Notes in Theoretical Computer Science no. 275, hal. 123-142, 2011.
- [10] Ristyabudi, A., dan Thamrin, H., "**Penerapan Single Page Application pada Proses Pengisian Online Data Rencana Studi Mahasiswa**," Khazanah Informatika: Jurnal Ilmu Komputer dan Informatika, vol. 2, no. 1, hal. 1-9, 2016.