

Usulan Pemanfaatan AJAX untuk Membangun *Rich Internet Application* pada Aplikasi *Smartcity*

R. Trimanadi¹⁾, Fransiskus Adikara²⁾, Husni S. Sastramihardja³⁾

Universitas Esa Unggul

Jalan Arjuna Utara No.9, Kebon Jeruk, Jakarta 11510

raden.trimanadi@gmail.com, fransiskus.adikara@esaunggul.ac.id, husni@esaunggul.ac.id

Abstrak

Dalam sebuah '*smartcity*', data yang dikumpulkan akan sangat banyak dan beragam. Beberapa konsep dan metode penyimpanan data tersebut telah dikembangkan oleh beberapa peneliti. Demikian juga dengan teknik pengolahan data menjadi sebuah informasi yang berguna telah banyak dikemukakan. Dengan tersedianya beragam informasi yang harus ditampilkan, akan memerlukan banyak web programmer handal untuk membangun *Rich Internet Application (RIA)* pada aplikasi *smartcity*. Kebutuhan akan programmer handal ini akan menjadi semakin kritis dan dapat menghambat berkembangnya sebuah *smartcity*. Dalam tulisan ini diangkat usulan untuk menyederhanakan teknik pembangunan ataupun pemeliharaan *RIA* sehingga dapat dilakukan oleh bahkan orang programmer pemula. Sebagai hasil dari sebuah penelitian awal, sebuah alternatif pemecahan dengan menggunakan *AJAX* yang disederhanakan telah berhasil dilakukan. Hasil dari penelitian dapat digunakan untuk penelitian selanjutnya dalam menyelesaikan masalah-masalah yang lebih kompleks saat mengembangkan aplikasi untuk *smartcity*.

Kata kunci: *Smartcity*; *rich internet application*; *web programmer*; *AJAX*; *rekayasa perangkat lunak*

1. Pendahuluan

Saat ini *smartcity*[1] adalah tren global strategi perkotaan yang bertujuan untuk memulihkan kualitas penduduk yang tinggal di daerah perkotaan dan memanfaatkan inovasi dan teknologi tinggi untuk memecahkan masalah sulit yang dihasilkan oleh kepadatan penduduk yang meningkat. Ini membantu memecahkan masalah urbanisasi, terutama pencemaran lingkungan, konsumsi lahan, *urban sprawl*, kemacetan transportasi, kebutuhan energi, kesulitan dalam mengakses layanan publik dan mengandung beragam inisiatif publik: membentuk sistem transportasi yang lebih baik untuk mendukung inovasi kreatif, pengetahuan untuk merancang kebijakan hemat energi.

Definisi pasti tentang *smartcity*, masih belum dapat ditetapkan secara pasti. Chourabi[2] merangkum 6 (enam) definisi *smartcity* dalam tulisannya yang dapat disimpulkan sebagai sebuah kota yang berperforma baik dengan cara mengintegrasikan semua infrastruktur kritisnya menggunakan *Information and Communication Technologies (ICT)* secara optimal hingga dapat memberikan layanan maksimal kepada masyarakat atas dasar dukungan dan aktivitas warga yang mandiri, peduli dan sadar.

Sedangkan menurut Deakin dan Al Wear[3], ada 4 (empat) faktor besar yang berkontribusi dalam pendefinisian sebuah *smartcity*, yaitu:

- Penerapan sejumlah besar teknologi digital dan elektronik dalam komunitas dan perkotaan.
- Penggunaan ICT untuk membentuk lingkungan kerja dan kehidupan dalam wilayah.
- Memasukkan ICT dalam sistem pemerintahan.
- Teritorialisasi praktek yang membawa komunitas masyarakat bersama-sama untuk meningkatkan inovasi dan pengetahuan yang mereka tawarkan.

Dengan semakin berkembangnya ICT, peralatan digital dan elektronik semakin beragam dalam pengertian jumlah dan ragam data yang dihasilkan sebagai input dalam sebuah *smartcity* semakin meningkat. Teknik penyimpanan data dan pengolahannya pun semakin berkembang, sebagai contoh dengan dikembangkannya konsep *Big Data*[5]. Hasil pengolahan informasi inilah yang diharapkan dapat mengoptimalkan layanan sebuah *smartcity*.

Tulisan ini tidak membahas tentang bagaimana data tersebut terbentuk, ataupun bagaimana para penentu kebijakan dalam menentukan informasi apa yang dapat diperoleh. Tetapi lebih mengulas tentang bagaimana informasi yang telah dihasilkan dapat sampai pada para penggunanya. Untuk mesin dapat dirancang konverter yang menerima data atau informasi tersebut secara langsung dan memberikan reaksi (SCADA, Automatisasi Mesin, LED Display). Sedangkan bagi manusia diperlukan *interface* ataupun media untuk membaca informasi tersebut sebelum menggunakannya. Secara visual digambarkan pada Gambar 1 sebagai hasil studi di atas.

Perangkat yang dipergunakan oleh manusia untuk membaca/mendapatkan informasi tersebut dapat berupa display monitor, TV, *gadget*, atau lainnya yang terhubung dengan sumber informasi tersebut (server). Media komunikasi yang saat ini paling luas digunakan adalah jaringan internet. Dengan

banyaknya ragam layanan informasi yang dibutuhkan, akan menuntut para *web programmer* untuk menciptakan berbagai macam situs yang dapat berinteraksi dengan pengguna yang memerlukan informasi sesuai dengan yang diinginkan.

Asynchronous Javascript and XML (AJAX) diperkenalkan pada tahun 2005, merupakan sebuah terobosan yang sangat berpengaruh pada konsep pembangunan website pada masa kini. Dengan AJAX[4][6] sebuah browser tidak perlu mengganti keseluruhan halaman jika ingin meng-*update* sebagian dari tampilannya sebagai respon. AJAX akan berinteraksi dengan server dan melakukan update tampilan pada DOM[7][8] pada bagian tampilan yang diinginkan. Hal ini yang menjadikan konsep ini tepat dalam menghadapi dinamika data serta respon pengguna dalam penampilan display informasi di sebuah *Smartcity* yang banyak memerlukan penerapan konsep *Rich Internet Application (RIA)*[9]. Dalam pelaksanaannya penerapan AJAX memerlukan pengetahuan khusus

(*javascript*, *XML*, *JSON* dan lainnya), sehingga seorang *web programmer* tingkat pemula tidak dapat serta merta mengerjakannya. Berdasarkan permasalahan inilah timbul pertanyaan: Apakah dapat dikembangkan sebuah format standar penerapan AJAX yang disederhanakan sehingga dapat dengan mudah digunakan oleh para *web programmer*?

Dengan berasumsi bahwa penggunaan AJAX dapat disederhanakan sehingga dapat dengan mudah dilakukan oleh *programmer* pemula, penelitian ini berhasil membuktikannya bahwa *programmer* dapat membuat sebuah aplikasi sederhana tanpa harus mengetahui secara mendalam tentang *javascript*. Tentunya penelitian ini masih sangat perlu untuk disempurnakan lebih lanjut untuk dapat diterapkan pada aplikasi RIA dalam sebuah aplikasi *smartcity*.

2. Metode Penelitian

Metode penelitian yang dilaksanakan untuk membuktikan asumsi awal di atas bahwa AJAX dapat digunakan dengan cara yang sederhana diuraikan sebagai berikut:

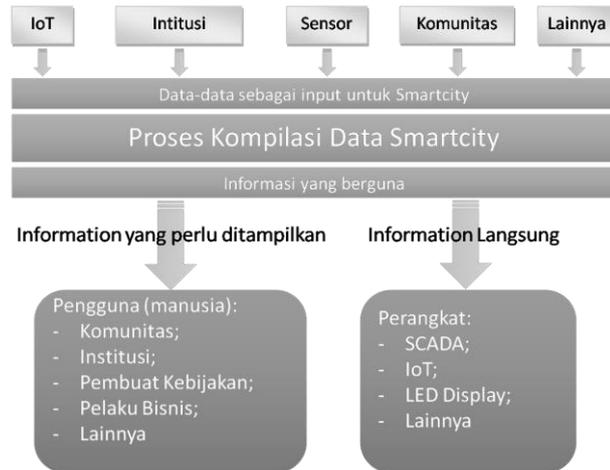
- Studi Pustaka.
Pengumpulan teori dan penelitian yang pernah dilakukan berkenaan dengan penerapan AJAX terkait asumsi yang ada.
- Pengembangan dan atau penyempurnaan konsep.
- Penelitian untuk pembuktian konsep di atas dengan cara penerapan pada situs web (dari yang sederhana hingga yang cukup rumit).
- Hasil dan pembahasan berkenaan dengan penelitian lanjutan.

Langkah ini akan berulang ke poin (a) hingga dapat ditemukan cara penyederhanaan yang benar-benar dapat diterapkan pada pembangunan RIA sebuah aplikasi *smartcity*. Pada tulisan ini akan dipaparkan hasil yang telah dilakukan pada proses awal penelitian ini.

3. Hasil dan Pembahasan

3.1. Hasil Studi Pustaka

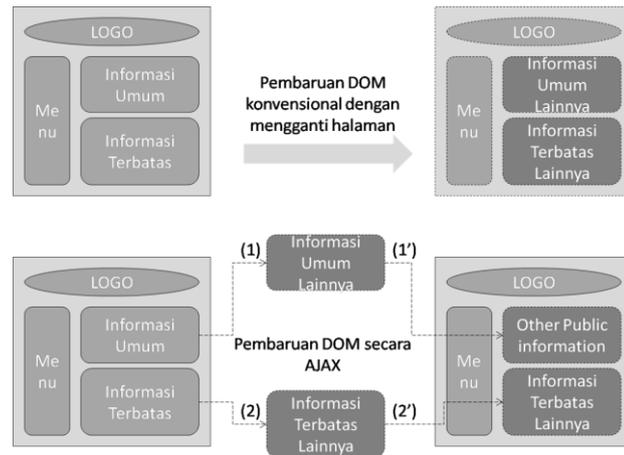
Setelah dilakukan studi literatur dengan kata kunci yang berkaitan dengan penulisan AJAX serta penerapannya pada 5 tahun terakhir, dapat disimpulkan bahwa belum ada peneliti yang telah mencoba untuk menyederhanakan penerapan AJAX seperti yang diharapkan. Dengan demikian, perlu disusun sebuah konsep penyederhanaan yang dinamakan '*Simplified AJAX*'.



Gambar 1. Proses pengolahan data menjadi informasi pada Smart City.

Prinsip dasar dalam AJAX adalah pemanfaatan perintah XMLHttpRequest yang tertanam pada halaman web browser guna melakukan interaksi dengan server. Sedangkan untuk memberikan tampilan akan sangat membantu proses unduh sebuah halaman web dengan penggunaan metode CSS3 dan HTML5.

Pada Gambar 2 ditunjukkan perbandingan antara metode konvensional dan metode AJAX. Dikarenakan besarnya bagian file yang tidak dibutuhkan untuk berganti data (menjadi informasi baru), dapat dibayangkan bagaimana metode AJAX dapat menghemat waktu unduh yang dibutuhkan. Angka (1) merupakan perintah untuk melakukan *updating* sebuah tampilan sedangkan (1') merupakan hasil interaksi browser dengan server berisi data dan informasi yang akan ditampilkan untuk tampilan terkait. Hal yang sama berlaku untuk mengubah tampilan yang lain (2) menjadi (2').



Gambar2. Perbandingan metode pembaharuan tampilan secara konvensional dengan secara AJAX.

Pada Gambar 2 dicontohkan apabila ingin 2 (dua) kali mengganti tampilan. Dalam sebuah RIA[4][10][11] terutama pada *Single Page Application*, pengguna akan meminta browser untuk melakukan penggantian tampilan (*data updating*) puluhan kali atau bahkan lebih. Dengan metode AJAX akan terasa perbedaan waktu unduh yang diperlukan pengguna dalam menggali informasi yang dibutuhkan[10].

3.2. Pengembangan Konsep Penyederhanaan AJAX

Sebelum mengkaji tentang AJAX lebih jauh, ada baiknya dipahami tentang bagaimana struktur dari sebuah halaman web. Para peneliti telah sepakat, bahwa sebuah halaman web dapat diumpamakan sebagai sebuah pohon dengan banyak ranting dan daunnya. Setiap bagian dari halaman web merupakan sebuah ranting dan dapat berupa sebuah *DOM tree* tersendiri[6][7].

Bila ditelaah lebih dalam tentang penulisan AJAX, setiap kali browser hendak mengganti tampilan suatu DOM, selalu menggunakan perintah yang susunannya tidak jauh dari berikut ini:

```
var req = new XMLHttpRequest();
```

atau

```
if (window.XMLHttpRequest) {
    // code for modern browsers
    xmlhttp = new XMLHttpRequest();
} else {
    // code for old IE browsers
    xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
}
```

Dengan perintah di atas, browser mengajukan sebuah permintaan data ke server.

```
xhttp.open("GET", "nama_file_pengolah_data", true);
xhttp.send();
```

Kemudian hasil tersebut ditampilkan dengan cara :

```
document.getElementById("DOM_id").innerHTML = xhttp.responseText;
```

Perlu dipahami bahwa nama fungsi *XMLHttpRequest*, tidak semata-mata hanya bekerja dengan format data XML. Pada praktiknya fungsi ini juga dapat bekerja dengan format data lain seperti JSON, HTML, text dan lainnya.

Pada paragraf di atas dapat dilihat bahwa AJAX meminta server dengan perintah *xhttp.open("GET", "nama_file_pengolah_data", true)*. Dengan cara yang sama seperti pada komunikasi antara browser dengan server menggunakan metode "GET", dikembangkan fungsi *Simplified AJAX* yang berfungsi untuk menampung nama file yang akan dijalankan, parameter yang diperlukan dan tempat (DOM) yang dituju. Dengan penggunaan metode "GET" pada script ini, pengujian file yang akan dipanggilpun dapat dengan mudah dilakukan. Sehingga setiap proses pembuatan halaman yang dibutuhkan untuk mengisi DOM dapat dilakukan secara parallel bersamaan. Adapun detil dari penulisan konsep ini secara ekperimental ditulis sebagai berikut:

```
function ajaxScript(url, divid, tambahan, ani){
    var rndm=Math.floor(Math.random()*11)
    var myLineBreak = tambahan;
    myLineBreak = myLineBreak.replace(/\n/g, '<br />');
    var url_ajax=url + "?rnd=" + rndm + myLineBreak;
    var ani_load = document.getElementById(ani);
```

```
if (ani_load) ani_load.style.visibility = 'visible';
if (window.XMLHttpRequest) {
    // Mozilla, Safari,...
    http_temp = new XMLHttpRequest();
    if (http_temp.overrideMimeType) {
        http_temp.overrideMimeType('text/xml');
    }
} else if (window.ActiveXObject) {
    // IE
    try {
        http_temp = new ActiveXObject("Msxml2.XMLHTTP");
    }
    catch (e) {
        try{
            http_temp = new ActiveXObject("Microsoft.XMLHTTP");
        } catch (e) {}
    }
}
if (!http_temp) {
    alert("Giving up :( Cannot create an XMLHTTP instance");
    return false;
}
http_temp.onreadystatechange=function(){
    if (http_temp.readyState == 4) {
        if (http_temp.status == 200 || http_temp.status == 0) {
            var dataku= http_temp.responseText;
            document.getElementById(divid).innerHTML=dataku;
            var isiparse = document.getElementById(divid);
            parseScript(isiparse.innerHTML);
        } else{
            var dataku="The server is under maintenance, you can visit
in a few hours later.....!!!!";
            document.getElementById(divid).innerHTML=dataku;
        }
        if (ani_load != null) ani_load.style.visibility = 'hidden';
    }
}
http_temp.open("GET", url_ajax, true);
http_temp.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
http_temp.send(null);
}
```

3.3. Uji Coba Konsep Penyederhanaan AJAX

Sebuah eksperimen dilakukan dengan cara membuat sebuah *Simplified AJAX script* yang sederhana menggunakan XAMPP server sebagai server dan Notepad++ sebagai perangkat editor. Dalam uji-coba, dibuat sebuah fungsi seperti berikut ini:

```
OnEvent="ajaxScript(URL,'NameOfDOM','ParameterValueofURL','AdditionalValue)'"
```

OnEvent adalah kejadian yang mengaktifkan fungsi AJAX ini, seperti aktifitas mouse atau keyboard oleh pengguna. Fungsi yang diaktifkan adalah *ajaxScript()* yang disusun dalam penelitian ini dengan menggunakan *javascript*. *URL* adalah alamat file yang hendak dijalankan. *ParameterValueofURL* digunakan untuk parameter imbuhan yang dibutuhkan file tersebut agar memberikan hasil yang diinginkan. Sedangkan *AdditionalValue* disiapkan untuk pemberian parameter lain yang sekiranya diperlukan. Dalam uji-coba, parameter ini digunakan untuk mengaktifkan animasi menunggu saat proses pengolahan hasil yang diminta.

Sebagai contoh, hasil dari eksperimen sederhana yang dilakukan dapat dilihat pada gambar3. DOM dengan nama "*tunggu*" adalah tempat dimana ditampilkan simbol/gambar menunggu yang menandakan bahwa sedang terjadi proses permintaan data dari server. Simbol ini akan menghilang dengan sendirinya dengan berhentinya proses permintaan AJAX ke server. *Menu 00*, *Menu 01* dan *Menu 02* merupakan DOM tempat pengguna mengaktifkan AJAX script untuk merubah tampilan pada DOM bernama "*tujuan00*". Bila pengguna meng-klik *menu 00*, maka browser akan meminta fungsi *ajaxScript()* untuk melakukan permintaan ke server agar menjalankan *menu_00.txt* dan ditampilkan di DOM *tujuan00*. Demikian juga bila pengguna mengklik pada *Menu 01*, maka akan menampilkan hasil *menu_01.txt* pada DOM *tujuan00*.

Bila seorang programmer ingin merubah tampilan dari sebuah DOM atas aktivasi yang dilakukan pengguna pada *Menu 02*, maka dia cukup mengganti parameter-parameter yang ada tanpa perlu menuliskan *javascript* pada fungsi AJAX. Sebagai contoh, bila ingin mengganti DOM bernama *tujuan_lain* dengan sebuah daftar pustaka, maka programmer cukup menuliskan sebagai berikut:

```
<a onClick = "ajaxScript('daftar_pustaka.php', 'tujuan_lain', '', 'tunggu') "> Menu 02 </a>
```

Bermodalkan fungsi `ajaxScript()` di atas, programmer akan bisa segera menggunakan AJAX, untuk membuat sebuah situs yang interaktif. Dengan demikian diharapkan kebutuhan akan programmer untuk pengembangan sebuah *Rich Internet Application* (RIA) ataupun *Single Page Application* (SPA) dalam rangka pengembangan aplikasi *smartcity* dapat dengan dipenuhi. Dengan terpenuhinya kebutuhan akan programmer, tentunya akan membawa ke kesempurnaan dalam penyampaian informasi yang dibutuhkan oleh para pelaku dalam *smartcity*.

Sebagai penelitian awal, pada paragraf-paragraf di atas, telah dibahas dan diuji-cobakan bagaimana penerapan *Simplified AJAX* dengan sebuah contoh yang sangat sederhana. Pembuktian asumsi awal berhasil dilakukan untuk halaman sederhana. Perubahan DOM yang diinginkan dapat dilakukan dengan tata cara penulisan yang relatif mudah dimengerti oleh programmer pemula.

Namun dengan kesimpulan ini masih belum dapat diyakini bahwa konsep ini dapat diterapkan dalam sebuah *smartcity*. Untuk itu perlu dilakukan penelitian yang lebih mendalam berkenaan dengan konsep *Simplified AJAX* seperti berikut ini:

- Bagaimana isi dari sebuah *Simplified AJAX* yang baik?
- Bagaimana penerapan *Simplified AJAX* dalam menggali informasi dari database?
- Bagaimana penerapannya pada sebuah situs yang relatif rumit seperti RIA ataupun Single Page Application?
- Bagaimana dilihat dari segi keamanannya saat terjadi komunikasi *browser* dan *server*.
- Apakah dapat diterapkan untuk proses *Login Authentication*?
- Seberapa besar kemudahan yang dihasilkan oleh konsep *Simplified AJAX* ini dalam melakukan modifikasi atau pemeliharaan sebuah situs?
- Bagaimana penerapan *Simplified AJAX* pada kenyataan dalam sebuah aplikasi *smartcity*?

Tentunya untuk menjawab pertanyaan-pertanyaan di atas akan diperlukan banyak penelitian lain. Dapat dibayangkan bila proses penyampaian informasi berkenaan dengan *smartcity* kepada para penggunanya dapat dipermudah tentunya akan mempercepat penyempurnaan sebuah aplikasi *smartcity*. Untuk itu kontribusi para peneliti terkait pada penyederhanaan ini akan sangat diharapkan mengingat tren pengembangan *smartcity* sedang digalakkan di seluruh dunia.

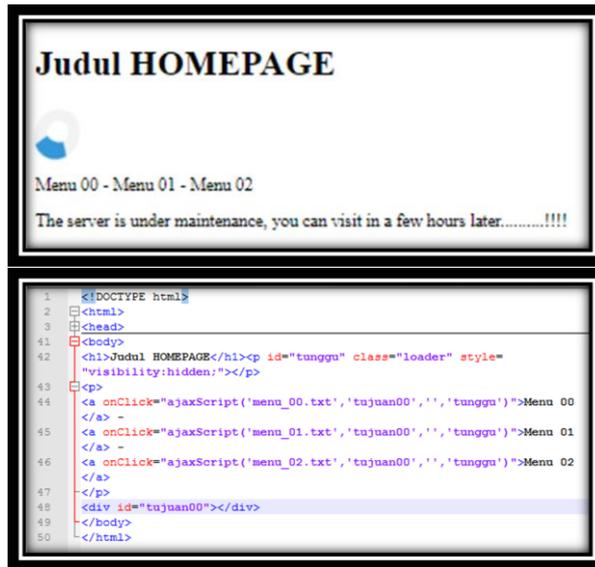
4. Simpulan

Dari pembahasan di atas disimpulkan bahwa penyederhanaan AJAX script dapat digunakan oleh web programmer dengan pengetahuan minimal. Walaupun sudah dibuktikan bahwa penerapan konsep AJAX sudah dapat disederhanakan, penelitian lebih lanjut tetap sangat diperlukan untuk melakukan penyempurnaan.

Sangat diharapkan partisipasi para peneliti agar dapat menyempurnakan konsep ini. Penyempurnaan konsep ini akan dapat mempercepat terbentuknya aplikasi *smartcity* secara utuh yang berujung pada peningkatan kesejahteraan yang lebih baik bagi masyarakat.

Daftar Pustaka

- [1] Hamblen, M., *Just what IS a smart city?* 2015.
- [2] Chourabi, H., et al., *Understanding Smart Cities: An Integrative Framework*. 2012.
- [3] Deakin, M. and H.A. Waer, *From Intelligent to Smart Cities*. Journal of Intelligent Buildings International: From Intelligent Cities to Smart Cities, 2011.
- [4] Dissanayake, N.R. and G.K.A. Dias, *Best Practices for Rapid Application Development of AJAX based Rich Internet Applications*. 2014 International Conference on Advances in ICT for Emerging Regions (ICTer), 2014.



Gambar3. Sebuah penerapan AJAX script pada code HTML halaman situs percobaan.

- [5] Dedić, N.; Stanier, C. (2017). "Towards Differentiating Business Intelligence, Big Data, Data Analytics and Knowledge Discovery". 285. Berlin ; Heidelberg: Springer International Publishing. ISSN 1865-1356. OCLC 909580101.
- [6] Yuping, J., *Research and Application of Ajax Technology in Web Development*. 2014 IEEE Workshop on Electronics, Computer and Applications, 2014.
- [7] BING, L., W. LAM, and T.-L. WONG, *Robust Detection of Semi-Structured Web Records Using a DOM Structure-Knowledge-Driven Model*. ACM Transactions on the Web, Vol. 7, No. 4, Article 21, 2013.
- [8] Liu, C.-H., C.-J. Wu, and H.-M. Chen, *Testing of AJAX-based Web Applications Using Hierarchical State Model*. 2016 IEEE International Conference on e-Business Engineering, 2016.
- [9] Rauti, S. and V. Leppänen, *Browser Extension-Based Man-in-the-Browser Attacks against Ajax Applications with Countermeasures*. International Conference on Computer Systems and Technologies - CompSysTech'12, 2012.
- [10] Dissanayake, N.R., G.K.A. Dias, and M. Jayawardena, *An Analysis of Rapid Application Development of AJAX based Rich Internet Applications*. 2013 International Conference on Advances in ICT for Emerging Regions (ICTer), 2013.
- [11] Jourdan, G.-V., et al., *Workshop on the application of security and testing to rich internet applications*, in *Proceedings of the 25th Annual International Conference on Computer Science and Software Engineering*. 2015, IBM Corp.: Markham, Canada. p. 320-322.