



# JURNAL IPTEK

## MEDIA KOMUNIKASI TEKNOLOGI

homepage URL : [ejurnal.itats.ac.id/index.php/iptek](http://ejurnal.itats.ac.id/index.php/iptek)



### Pengujian Aplikasi Web - Tinjauan Pustaka Sistematis

Dimas Widya Liestio Pamungkas<sup>1</sup>, Siti Rochimah<sup>2</sup>

<sup>1,2</sup>Departemen Informatika, Institut Teknologi Sepuluh Nopember (ITS) Surabaya

#### INFORMASI ARTIKEL

Jurnal IPTEK – Volume 23  
Nomer 1, Mei 2019

Halaman:  
17–24  
Tanggal Terbit :  
31 Mei 2019

DOI:  
[10.31284/j.iptek.2019.v23i1.459](https://doi.org/10.31284/j.iptek.2019.v23i1.459)

#### EMAIL

[dimaswlp.18051@mhs.its.ac.id](mailto:dimaswlp.18051@mhs.its.ac.id)  
[siti@if.its.ac.id](mailto:siti@if.its.ac.id)

#### PENERBIT

LPPM- Institut Teknologi  
Adhi Tama Surabaya  
Alamat:  
Jl. Arief Rachman Hakim  
No.100,Surabaya 60117,  
Telp/Fax: 031-5997244

*Jurnal IPTEK by LPPM-  
ITATS is licensed under a  
Creative Commons  
Attribution-ShareAlike 4.0  
International License.*

#### ABSTRACT

*Web-based applications in the past few decades have experienced very rapid development. The use of technology in software engineering for web application development has also increased several changes following the needs of the community. Web applications that are built required testing so that there are no problems that can disrupt services and information. In software engineering, performance testing holds an important role in overcoming problems in developing web applications. Many studies related to performance testing have been carried out in the past few years. This research discusses studies that have been published in the form of journal articles and conference proceedings in the period 2012-2017. This research selects 22 research articles related to Web Application Testing. Then divided into several categories and do reviews of each of these categories. It is expected that the results of the discussion can be parsed in detail so that in the future related research will get valuable direction and resources. The systematic literature review of this paper concludes that 68% of studies related to testing are Load-Tests and 32% other tests. Trend testing leading to the ability of the web to load information is now more emphasized following the needs of software engineering testing in web applications.*

**Keywords:** *Systematic literature review; Software engineering; Web performance test; Web application test*

#### ABSTRAK

Aplikasi berbasis web dalam beberapa dekade terakhir mengalami perkembangan yang sangat pesat. Pemanfaatan teknologi dalam rekayasa perangkat lunak untuk pengembangan aplikasi web juga mengalami banyak perubahan mengikuti kebutuhan masyarakat. Aplikasi web yang dibangun tentu memerlukan pengujian agar tidak terjadi permasalahan yang dapat menyebabkan terganggunya pelayanan dan informasi. Dalam rekayasa perangkat lunak, *performance testing* memegang peranan penting untuk mengatasi masalah-masalah pada pengembangan aplikasi web. Banyak penelitian terkait *performance testing* yang sudah dilakukan dalam beberapa tahun terakhir. penelitian ini membahas penelitian-penelitian yang telah dipublikasikan dalam bentuk artikel jurnal dan prosiding seminar dalam rentang waktu tahun 2012–2017. Sebanyak 22 artikel penelitian terkait *Web Application Testing* terpilih dalam artikel ini. Kemudian, artikel dibagi dalam beberapa kategori dan dilakukan ulasan dari setiap kategori tersebut. Diharapkan hasil pembahasan bisa diurai secara mendetail sehingga di masa yang akan datang penelitian terkait akan mendapatkan arah dan sumber berharga. Tinjauan pustaka sistematis dari paper ini menghasilkan kesimpulan 68% penelitian terkait pengujian merupakan Load-Test dan 32% pengujian lainnya. Trend pengujian mengarah pada kemampuan dari web dalam memuat informasi saat ini lebih ditekankan mengikuti kebutuhan dari pengujian rekayasa perangkat lunak dalam aplikasi web.

**Kata kunci:** *Tinjauan pustaka sistematis; Rekayasa perangkat lunak; Pengujian performa web; Pengujian aplikasi web*

## PENDAHULUAN

*Website* telah berkembang sangat pesat selama beberapa dekade terakhir. Hal ini berdampak pada banyak aspek dalam masyarakat, bisnis, pendidikan, pemerintahan, hiburan, industri, bahkan dalam aspek kehidupan pribadi. Keuntungan utama dalam mengadopsi web untuk pengembangan perangkat lunak di antaranya adalah minim biaya instalasi, peningkatan fitur baru untuk pengguna secara otomatis, dan akses secara universal dari semua perangkat yang terhubung dengan internet. Namun, terdapat sisi negatif dari web, yaitu rentan terjadinya eror dan pengujian yang tidak mudah sehingga diperlukan pengujian yang memadai [1]. *Website* yang baik seharusnya dapat diakses secara cepat. Pengunjung *website* tidak akan mau menunggu lama. Pada *website* bisnis, persaingan dalam menyajikan informasi produk dan jasa bersaing ketat. Kecepatan dalam menampilkan isi *website* menjadi hal yang perlu dipertimbangkan. *Website* yang cepat akan memperoleh pengunjung yang lebih banyak sehingga berdampak pada jumlah *customer* dan transaksi yang ditangani [2].

*Search engine* sekilas Google juga lebih memilih *website* yang dapat diakses dengan cepat. Google dalam menentukan *ranking* juga menilai dari kecepatan akses dari web yang diindeks [2]. Sehingga, pengujian *website* yang baik memerlukan metode dan teknik yang memadai. Pengujian aplikasi web memiliki banyak kesulitan. Pertama, web didistribusikan melalui model *client/server* yang rentan bermasalah. Kedua, web bersifat heterogen, yaitu dikembangkan dengan berbagai bahasa pemrograman yang berbeda, misalnya PHP, Ruby, Java di sisi server dan HTML, CSS, Javascript di sisi *client*. Ketiga, web bersifat dinamis sehingga pengembangan yang dilakukan perlu diuji [1].

Sesuai penelitian yang dilakukan sebelumnya oleh [3], Systematic Mapping (SM) adalah metode untuk melakukan peninjauan, pengklasifikasian dan penyusunan *paper* yang terkait dengan bidang penelitian tertentu dalam rekayasa perangkat lunak. Dengan tujuan untuk mendapatkan gambaran dengan pendekatan yang sesuai, menguraikan bidang penelitian dalam berbagai segi skema klasifikasi. Ini berfungsi juga sebagai dasar untuk penelitian lebih lanjut di masa depan pada keilmuan rekayasa perangkat lunak. Hasil studi SM juga dapat menjadi sumber penelitian baru bagi peneliti dengan memberikan gambaran tentang area penelitian tertentu. Sementara *Systematic Literature Review* yang berarti Tinjauan Pustaka Sistematis adalah metode peninjauan pustaka (*literature review*) yang mengidentifikasi, menilai dan menginterpretasi seluruh temuan-temuan pada suatu topik penelitian, untuk menjawab pertanyaan penelitian (*research questions*) yang telah ditetapkan sebelumnya [4]. Perbedaan antara SM dan SLR bisa agak kabur. Kami menggunakan metode SLR pada artikel ilmiah ini.

Penelitian yang dilakukan menyajikan kemajuan terbaru dalam teknik pengujian web dan membahas kekuatan dan kelemahan dari masing-masing teknik tersebut. Saat ini perkembangan aplikasi web sangat pesat dan semakin kompleks. Selain itu, Aplikasi web saat ini mampu menyelesaikan transaksi secara kompleks dengan cara yang aman dan waktu yang relatif singkat. Sehingga peran aplikasi web menjadi sentral dan penting. Sehingga perlu dipastikan bahwa aplikasi web harus aman, tepat dan efisien dengan cara melakukan pengujian. Karena aplikasi web menjadi lebih penting dan kompleks, cacat pada perangkat lunak dapat berdampak signifikan terhadap pengguna dan juga penyedia layanan. Pentingnya pengujian pada aplikasi web tidak dapat diremehkan. Fakta menunjukkan bahwa pengujian perangkat lunak adalah bagian yang sangat penting dari keseluruhan proses pembuatan perangkat lunak berkualitas tinggi. Sebanyak 40% waktu pembuatan perangkat lunak digunakan untuk melakukan pengujian untuk memastikan kualitas perangkat lunak yang dihasilkan [5].

Publikasi dalam *web application performance testing* menunjukkan peluang penelitian yang menjanjikan, dengan mengimplementasikan pengujian kinerja pada aplikasi web. Dalam *Systematic Literature Review* (SLR) ini, kami bertujuan untuk mencakup teknik yang telah digunakan dalam aktivitas pengujian kinerja aplikasi web dan untuk menganalisis keuntungan dan kerugian dari masing-masing teknik. Setelah melakukan pemilihan literatur berdasarkan kriteria tertentu, studi yang dipilih kemudian diproses dan dianalisis untuk menjawab pertanyaan penelitian atau *research question* (RQ). Dua perpustakaan digital *online* digunakan, ScienceDirect dan IEEE Xplore, untuk mendapatkan literatur yang diterbitkan dari tahun 2012 hingga 2017. Kami juga membatasi hanya artikel jurnal atau prosiding yang kami analisis. Hasilnya disajikan dan disimpulkan pada bagian akhir makalah ini.

Artikel ini disusun sebagai berikut: Bagian *Tinjauan Pustaka* menyajikan penelitian yang relevan yang telah dilakukan sebelumnya di bidang pengujian kinerja aplikasi web. Bagian *Metode* memberikan metode penelitian tentang pengujian kinerja aplikasi web yang digunakan dalam SLR dan

laporan hasil temuan berdasarkan pertanyaan penelitian. Bagian *Hasil dan Pembahasan* menyajikan hasil dan analisis dari penelitian ini. *Kesimpulan* penelitian disajikan pada bagian akhir.

## TINJAUAN PUSTAKA

Penelitian dalam studi *Systematic Mapping* telah dilakukan oleh beberapa peneliti sebelumnya. Penelitian terkait *Web Performance Testing* dalam bentuk *Literature Review* juga telah dilakukan oleh [1] dan [6]. Penelitian tersebut menghasilkan beberapa paper terkait untuk dilakukan *review* mendalam sehingga dapat diperoleh analisis yang menjawab pertanyaan terkait penelitian (*research question*). V. Garousi dkk. dalam proses *review* menganalisis 147 *paper* terkait *web application testing*. Setelah dilakukan *screening* mendalam, diperoleh 79 *paper* yang akan diteliti lebih mendalam.

Pada publikasi penelitian lainnya terkait *Systematic Literature Review*, S. Doğan dkk. [6] pada 2013 menganalisis 114 *paper* terkait *web application testing* yang didapatkan dari pencarian dari beberapa *digital library*. Hasilnya, didapatkan 95 *paper* final setelah melakukan *screening* dan penelitian lebih mendalam. Dalam publikasinya, S. Doğan dkk. [6] menggunakan juga hasil dari *screening paper* pada penelitian *Systematic Mapping* yg dilakukan [1] sebelumnya.

Di tahun 2014, kami menemukan sebuah artikel jurnal terkait survei terhadap teknik dan metode *testing* pada aplikasi web. Survei oleh [7] ini dilakukan dengan meneliti terkait pengujian aplikasi web selama dua dekade terakhir. Teknik pengujian yang disurvei untuk memastikan bahwa aplikasi web berfungsi secara konsisten dengan spesifikasi sesuai kebutuhan. Pengujian yang dilakukan diharapkan dapat memberi informasi kelemahan sebanyak-banyaknya dari aplikasi web sehingga celah ini dapat ditutup sebelum aplikasi web dirilis ke publik. Aplikasi web saat ini menjadi perhatian karena sudah mencakup ke transaksi keuangan, reservasi, hingga komunikasi. Aplikasi web perlu pengamanan karena data dipertaruhkan. Sehingga, pengujian web yang saat ini cukup intens dilakukan yaitu pengujian keamanan [7].

Dalam penelitian, metode pencarian artikel dilakukan secara otomatis dengan kata kunci tertentu terhadap dua *digital library* yang cukup populer 5 tahun terakhir. Penelitian juga dibatasi hanya untuk artikel jurnal dan prosiding yang ditelaah untuk mendapatkan penelitian terbaru di pengujian sistem aplikasi web. Artikel SLR ini diharapkan mampu memberikan gambaran bahwa tren pengujian *website* saat ini mengarah ke pengujian jenis tertentu atau metode yang lebih baru yang sesuai dengan perkembangan teknologi web.

## METODE

*Systematic Literature Review* (SLR) dilakukan mengikuti pedoman yang diusulkan oleh [3]. Tujuan dari SLR adalah untuk mengidentifikasi teknik terbaik dengan prosedur spesifik, teknologi, metode, atau *tools* dengan mengumpulkan informasi dari studi perbandingan. Sedangkan tujuan dari *Systematic Mapping* (SM) adalah klasifikasi dan analisis tematik literatur pada topik rekayasa perangkat lunak. SLR biasanya lebih mendalam daripada SM dan seringkali SLR termasuk dalam bagian dari SM. Hal ini berarti hasil dari SM dapat dimasukkan ke dalam studi SLR yang lebih spesifik untuk mendukung rekayasa perangkat lunak. Dalam studi SLR, terdapat beberapa proses yang dapat dilakukan yaitu: (1) Menentukan *Research Questions* (RQ); (2) Menentukan *Search Query* (SQ) dan melakukan pencarian *paper* terkait; (3) *Screening Paper* (SP) yang diperoleh dengan membaca abstrak dan isi *paper*, lalu menentukan *paper* yang relevan untuk diproses; (4) Melakukan pencarian kata kunci dengan abstrak yang menghasilkan skema klasifikasi; serta (5) Melakukan ekstraksi data dan proses *review* secara sistematis terhadap *paper* yang diperoleh.

*Research questions* atau pertanyaan penelitian berfungsi untuk mengekstraksi informasi mendetail dari setiap studi yang dilakukan. Empat formulasi pertanyaan penelitian dibuat terkait hal ini:

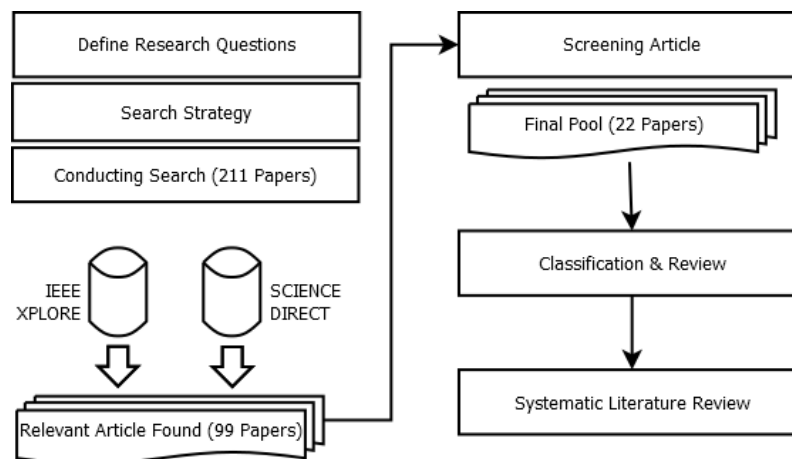
- **RQ1:** *Apa jenis metode web performance testing yang digunakan?* Hal ini penting diidentifikasi agar penelitian ini dapat menunjukkan informasi terkait tipe dan karakteristik metode pengujian yang telah dilakukan sehingga peneliti akan memperoleh ide bagaimana melakukan penelitian dengan metode terkait untuk penelitian lebih lanjut.
- **RQ2:** *Apakah web performance testing menggunakan tools tertentu?* Perangkat pengujian biasanya digunakan untuk melakukan pengujian sehingga dengan mengetahui *tools* yang digunakan dapat memudahkan pengaplikasian *tools* untuk penelitian selanjutnya.

- **RQ3:** Apakah *web performance testing* telah diaplikasikan dalam kasus riil? Berbeda dengan pengujian yang bersifat usulan, pengujian pada kasus riil dapat membuktikan bahwa suatu metode atau tools dapat menjanjikan hasil yang lebih nyata dalam pengujian web.
- **RQ4:** Apa *trend metode pengujian web yang berkembang saat ini*? Dengan mengetahui tren metode pengujian, diharapkan dapat diketahui perkembangan metode dan *tools* yang digunakan untuk melakukan pengujian terbaru. Tentunya, ini sangat erat kaitannya dengan perkembangan teknologi *website* yang menuntut pengujian lebih adaptif untuk performa dari aplikasi web.

Untuk memperoleh artikel dengan studi yang relevan, dilakukan pencarian ke dalam dua *digital library* yaitu IEEE Xplore dan ScienceDirect. Pencarian juga dibatasi hanya untuk artikel berbahasa Inggris dalam bentuk artikel ilmiah atau jurnal dan juga artikel prosiding dalam rentang waktu lima tahun yaitu 2012 hingga 2017. Formulasi kata kunci dalam pencarian yaitu "web performance testing" OR "web performance test" OR "web performance" OR "web test". Kata kunci ini berguna untuk mendapatkan artikel yang fokus hanya terkait dengan tes performa web. Hasilnya, diperoleh 168 artikel. Namun, setelah dilakukan filter awal, diperoleh beberapa artikel yang kurang terkait. Di antara artikel yang dieliminasi yaitu artikel jurnal atau prosiding yang tidak berhubungan, topik yang tidak sesuai maupun artikel yang tidak berkaitan sehingga didapati 99 artikel yang lebih spesifik. Artikel tersebut di antaranya didapatkan dari ScienceDirect sebanyak 56 artikel dan dari IEEE Xplore sebanyak 43 artikel.

Pada tahap *screening paper*, dilakukan pembacaan abstrak dan isi dari setiap artikel dari proses sebelumnya. Hasilnya, didapatkan 22 *paper* terkait yang dibahas secara mendalam pada metode SLR ini. Artikel hasil dari proses ini setidaknya harus dapat menjawab *research question* yang diformulasikan dari tahap sebelumnya.

Langkah selanjutnya adalah membaca seluruh artikel penelitian yang diperoleh. Terdapat 22 artikel yang sesuai yang akan diproses lebih lanjut untuk penilaian kualitas penelitian sebelum dilakukan proses ekstraksi data. Alur proses kerja yang dilakukan untuk memperoleh artikel penelitian hingga dihasilkan *systematic literature review* dapat dilihat pada Gambar 1.



Gambar 1. Proses yang Dilakukan untuk Memperoleh *Paper Systematic Literature Review*

### Penilaian Kualitas Artikel Penelitian

Kualitas dari SLR ini erat kaitannya dengan kualitas data yang diperoleh dari hasil ekstraksi artikel penelitian yang telah dipilih sebelumnya. Kualitas artikel dapat dilihat dari relevansi dan kredibilitas penelitian yang telah dipaparkan [30]. Berikut ini pertanyaan terkait penilaian kualitas artikel penelitian untuk menentukan kualitas data.

**P1:** Apakah metode yang digunakan dalam proses pengujian performa web jelas dan teliti? Pertanyaan ini bertujuan menilai aspek kejelasan artikel penelitian. Deskripsi metode harus diberikan dengan jelas di artikel, baik itu sebagian maupun secara jelas. Kurangnya evaluasi atau implementasi dapat mengurangi kualitas dari literatur tersebut. **P2:** Apa tipe publikasi dari literatur yang terpilih? Pertanyaan ini ditujukan untuk informasi apakah artikel dipublikasikan pada jurnal atau seminar. **P3:** Kapan literatur ini dipublikasikan? Pertanyaan ini bertujuan untuk menjawab bahwa artikel apakah termasuk yang kurang relevan dengan perkembangan teknologi. Artikel yang baru biasanya

lebih relevan dan sesuai dengan perkembangan dibandingkan dengan artikel yang telah diterbitkan satu dekade yang telah lalu. **P4: Berapa jumlah sitasi dari artikel penelitian?** Pertanyaan ini menunjukkan kontribusi dan relevansi dari artikel penelitian, penelitian yang baik akan digunakan oleh peneliti lain untuk dikembangkan.

Artikel terpilih dilakukan analisis dengan pertanyaan penilaian kualitas penelitian yang disajikan dalam Tabel 1.

Tabel 1. Penilaian Kualitas Artikel Penelitian

No.	Ref.	P1	P2	P3	P4	No.	Ref.	P1	P2	P3	P4
1	[8]	Yes	Jurnal	2013	3	12	[19]	Yes	Seminar	2012	-
2	[9]	No	Jurnal	2016	-	13	[20]	Yes	Seminar	2016	2
3	[10]	Yes	Jurnal	2013	17	14	[21]	Yes	Seminar	2015	22
4	[11]	Yes	Jurnal	2013	5	15	[22]	Yes	Seminar	2015	2
5	[12]	No	Seminar	2015	2	16	[23]	Yes	Seminar	2014	23
6	[13]	Yes	Jurnal	2013	5	17	[24]	Yes	Seminar	2016	7
7	[14]	Yes	Jurnal	2013	15	18	[25]	Yes	Seminar	2014	1
8	[15]	Yes	Jurnal	2016	2	19	[26]	Yes	Seminar	2014	2
9	[16]	Yes	Seminar	2016	-	20	[27]	Yes	Seminar	2015	23
10	[17]	Yes	Seminar	2013	3	21	[28]	No	Seminar	2013	-
11	[18]	No	Seminar	2017	1	22	[29]	Yes	Seminar	2015	4

### Ekstraksi Data dan Analisis

Artikel yang telah terpilih kemudian dilakukan ekstraksi data untuk menjawab pertanyaan penelitian (RQ). Data yang diekstraksi dari artikel penelitian membahas terkait pengujian performa web, baik itu metode maupun teknik dan pendekatan lainnya, implementasi atau studi kasus, penggunaan alat pengujian (*tools*) maupun evaluasi kekurangan dan kelebihan dari masing-masing metode yang digunakan. Dengan membaca 22 artikel terpilih, akan dihasilkan analisis yang dapat menjawab pertanyaan penelitian. Proses ini akan dijelaskan ke dalam bagian selanjutnya secara mendetail.

## HASIL DAN PEMBAHASAN

### RQ1: Apa Jenis Metode Web Performance Testing yang Digunakan?

Terdapat beberapa jenis metode yang dibahas dalam pengujian performa web yang telah diklasifikasikan ke dalam 3 skema dari 22 artikel yang dipilih. Paper yang telah diperoleh dibagi ke dalam proses *screening* dalam beberapa skema klasifikasi [3] sebagai berikut.

- *Proposal Solusi*: Solusi untuk masalah diusulkan, dapat berupa usulan perbaikan teknik pengujian yang telah ada. Manfaat potensial dan penerapan solusi ditunjukkan oleh contoh kecil yang bagus dalam penelitian.
- *Penelitian Validasi*: Teknik yang diselidiki masih baru dan belum diimplementasikan dalam praktik. Teknik yang digunakan misalnya eksperimen, yaitu pengujian yang dilakukan di laboratorium.
- *Penelitian Evaluasi*: Teknik diimplementasikan dalam praktik dan evaluasi pengujian dilakukan. Hal ini menunjukkan bagaimana teknik ini diterapkan dalam praktik (implementasi solusi) dan konsekuensi dari implementasi dalam hal manfaat dan kekurangan (evaluasi implementasi).

Hasil klasifikasi ditampilkan dalam Tabel 2.

Tabel 2. Pembagian Artikel Berdasarkan Skema Klasifikasi Penelitian

Type	Referensi	Jumlah
Proposal Solusi	[8] [10] [13] [14] [19] [20] [21] [23] [26]	9
Penelitian Validasi	[9] [11] [12] [15] [16] [17] [18] [22]	8
Penelitian Evaluasi	[24] [25] [27] [28] [29]	5

Terdapat sembilan penelitian yang masuk dalam klasifikasi Proposal Solusi. Kesembilan penelitian ini memperbaiki teknik yang digunakan untuk pengujian maupun optimasi dalam memuat *website*. Hasil dari penelitian ini di antaranya penelitian [8] melakukan pendekatan yang berbeda dari pengujian yang biasa dilakukan peneliti lainnya yaitu menggunakan *log* atau catatan kesalahan untuk

diuji. Peneliti membagi skema pengujian dari sisi *user* dengan algoritma K-Means dan menggunakan model Markov untuk melakukan estimasi *reliability* dari aplikasi web. Pengujian beban kerja hasil dari perbaikan terhadap teknik yang telah ada juga telah dilakukan dalam artikel penelitian [10], [13], [14], [19], [20], dan [26]. Penelitian Proposal Solusi dengan melakukan optimasi *xpath locator* yang diujikan dalam enam aplikasi web dipaparkan oleh artikel penelitian [21] dan [23].

Delapan penelitian masuk dalam kategori Penelitian Validasi. Penelitian ini sebagian besar berupa pengujian eksperimen dilakukan oleh peneliti dengan tipe pengujian beban kerja pada aplikasi web oleh peneliti [9], [15]–[18], dan [22], sedangkan untuk pengujian error dalam eksperimen dipaparkan oleh penelitian [11]. Pada penelitian [12], peneliti mencoba untuk menguji aksesibilitas dari web aplikasi yang tentu masih jarang dilakukan dalam pengujian performa web. Untuk Penelitian Evaluasi, terdapat lima artikel penelitian yang mencoba mengevaluasi teknik sekaligus mengimplementasikan ke dalam perangkat uji coba oleh peneliti [24], [25], dan [27]–[29].

Metode pengujian performa web yang dilakukan para peneliti sebagian besar menggunakan metode *Load-Test*. *Load-Test* atau pengujian performa dilakukan saat memuat aplikasi dengan parameter tertentu sehingga dapat diketahui kemampuan dari aplikasi web. Ini berguna untuk mengetahui seberapa jauh beban yang dapat ditangani oleh aplikasi web yang diuji terhadap *database* atau server sehingga *resource* yang digunakan dapat memenuhi kebutuhan. Metode kedua yang terbanyak digunakan yaitu *Fault-Test* atau pengujian performa dilakukan dengan mengetahui kemampuan dari aplikasi web hingga muncul error saat dilakukan pengujian beban. Selain kedua metode tersebut, juga diperoleh beberapa optimasi dan hasil pengujian dari pengembangan *framework* tertentu maupun pendekatan dari aksesibilitas terhadap aplikasi web. Hasilnya disajikan dalam Tabel 3.

Tabel 3. Metode Dalam Pengujian Performa Web

Metode	Referensi	Jumlah
Load-Test	[9][10][13][14][15][16][17][18][19][20][22][25][26][28][29]	15
Fault-Test	[8][11][24][27]	4
Other	[12][21][23]	3

### RQ2: Apakah Web Performance Testing Menggunakan Tools Tertentu?

Pada analisis sebelumnya web *performance testing* telah dibagi dalam beberapa metode pengujian. Banyak metode pengujian yang dilakukan menggunakan metode *Load-Test* yang dilakukan dalam pengujian performa web yaitu sebanyak 12 artikel. Selain itu, *Fault-Test* yang digunakan oleh 7 penelitian dan 3 penelitian menggunakan metode lainnya. Peneliti dalam bagian ini mengungkapkan *tools* yang digunakan oleh artikel dalam melakukan pengujian. Sebagian besar adalah jenis pengujian dengan aplikasi pengujian seperti JMeter, LoadRunner, GridView, WebPro, dan sebagainya yang merupakan jenis *Load-Test*.

Untuk pengujian jenis *Fault-Test*, beberapa menggunakan maupun memodifikasi algoritma untuk melakukan pengujian. Di antaranya, modifikasi algoritma jenis Genetic Algorithm (GA) untuk melakukan *fail-safe testing* [11]. Peneliti [10] juga melakukan modifikasi dengan menggunakan metode Turning Bands (TB) dan Sequential Gaussian Simulation (SGS) untuk mengevaluasi dan memprediksi performa web. Peneliti [12] melakukan pengujian aksesibilitas *website* untuk pengguna difabel. Sedangkan pada penelitian [13] dilakukan pengujian pada *framework* web 2.0 dengan melakukan improvisasi pada *web cache* dengan *tools Cofridias framework*.

### RQ3. Apakah Web Performance Testing telah Diaplikasikan dalam Kasus Riil?

Pada bagian ini, telah disimpulkan bahwa tidak banyak penelitian yang menerapkan teknik pengujian dengan menggunakan web riil. Sebagian besar pengujian dilakukan dengan simulasi yang di-*setup* pada perangkat *server-client*. Namun, pada penelitian [13], [15], [20]–[22], [24], dan [28], pengujian juga telah dilakukan pada web riil selain menggunakan simulasi. Peneliti [13] menggunakan dua *website* populer yaitu PageFlakes dan Yahoo! Pipes untuk dilakukan pengujian *framework* web *cache*. Peneliti [15] menggunakan 20 *top website* di negara Kanada dalam daftar *ranking* Alexa, peneliti [20] menggunakan 4 *top rank* Alexa untuk dilakukan pengujian beban kerja secara otomatis dengan WePR. Sedangkan peneliti [21] mengimplementasikan pengujian pada 6 aplikasi web yang terdapat pada Sourceforge.net. Peneliti [22] menggunakan 959 URL dari Alexa untuk mengukur

performa web dari hasil *render cache web*. Peneliti [24] juga menggunakan 100 *website* dari daftar Alexa untuk menguji kesamaan konten *website* dalam beberapa kategori. Peneliti [28] menggunakan *website* dari Sistem Administrasi pada Wuhan Landscaping Bureau untuk dilakukan pengujian performa dengan perspektif pengguna.

#### **RQ4. Apa Tren Metode Web Performance Testing yang Berkembang Saat Ini?**

Penelitian ini menemukan bahwa sebagian besar artikel yang diperoleh dari hasil pencarian adalah metode pengujian jenis *Load-Test*. Metode pengujian tipe ini digunakan dalam penelitian lebih dari separuh artikel yang didapatkan. Ini menunjukkan bahwa *Load-Test* menjadi tren yang sedang diminati oleh para peneliti. Hal ini sesuai dengan analisis bahwa penelitian untuk meningkatkan kecepatan, kestabilan, kemampuan dalam skalabilitas, maupun reliabilitas dalam beban berat menjadi tantangan yang banyak diteliti untuk mendapatkan solusi. Teknik-teknik dalam pengujian juga dikembangkan misalnya dengan melibatkan algoritma prediksi yang dipaparkan oleh peneliti [14] untuk *web prefetching* dalam mengurangi *web latency*. Pengujian beban kerja secara dinamis juga lebih ditekankan dalam penelitian [10], [13], dan [19] dengan membandingkan pengujian secara statis. Pengujian dinamis akan menghasilkan analisis yang lebih mendekati penyelesaian masalah pada kasus nyata. Metode *Load-Test* digunakan oleh 68% dari hasil penelitian ini.

#### **KESIMPULAN**

Penelitian ini menyimpulkan dari hasil analisis terhadap data bahwa performa dari aplikasi web memerlukan pengujian yang serius dan *ter-update*, mengingat setiap aplikasi web berkembang mengikuti kebutuhan *user*. Beberapa metode dilakukan modifikasi dan pengujian menghasilkan beberapa instruksi perbaikan untuk meningkatkan performa aplikasi web. Penggunaan perangkat lunak pengujian (*tools*) dapat meningkatkan hasil pengujian, namun memerlukan beberapa modifikasi dan penyesuaian agar diperoleh hasil yang maksimal. Pengujian juga sebaiknya dilakukan dengan pengembangan adaptif mengingat tingkat kepadatan akses dan beban kerja aplikasi web erat kaitannya dengan kepuasan pengguna.

Pengujian pada aplikasi web yang nyata dapat merepresentasikan hasil yang lebih baik karena perangkat pengujian eksperimental tidak selalu dapat diterapkan pada studi kasus nyata. Sehingga, pengujian, baik *load-test*, *fault-test*, maupun lainnya akan lebih berkontribusi bila dilakukan pada area *web application* yang riil. Tren pengujian dalam penelitian ini menunjukkan sebagian besar artikel yang terjaring menggunakan metode *Load-Test*. Hal ini tentunya menyediakan sumber penelitian yang lebih mudah didapatkan. Namun, tidak menutup kemungkinan, pada metode yang lain dapat membuka kontribusi yang lebih bila dilakukan pengembangan metode.

Dalam penelitian ini, terdapat beberapa keterbatasan, di antaranya yang telah disebutkan bahwa hanya digunakan dua *digital library*. Jumlah ini lebih sedikit dibandingkan dengan yang digunakan dalam metode SLR yang dilakukan oleh [6]. Hal ini dikarenakan keterbatasan akses yang dimiliki oleh institusi Penulis.

#### **REFERENSI**

- [1] V. Garousi, A. Mesbah, A. Betin-Can, and S. Mirshokraie, "A systematic mapping study of web application testing," *Inf. Softw. Technol.*, vol. 55, no. 8, pp. 1374–1396, Aug. 2013.
- [2] "Why should you be serious about web performance testing? - CloudQA." [Online]. Available: <https://cloudqa.io/serious-web-performance-testing/>. [Accessed: 26-Nov-2018].
- [3] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," *Proceedings of the 12th international conference on Evaluation and Assessment in Software Engineering*. BCS Learning & Development Ltd., pp. 68–77, 2008.
- [4] B. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature Reviews in Software Engineering."
- [5] G. J. Myers, C. Sandler, and T. Badgett, *The art of software testing*. John Wiley & Sons, 2012.
- [6] S. Doğan, A. Betin-Can, and V. Garousi, "Web application testing: A systematic literature review," *J. Syst. Softw.*, vol. 91, pp. 174–201, May 2014.
- [7] Y.-F. Li, P. K. Das, and D. L. Dowe, "Two decades of Web application testing—A survey of recent advances," *Inf. Syst.*, vol. 43, pp. 20–54, Jul. 2014.

- [8] C. Chen, Y. Chen, H. Miao, and H. Wang, "Usage-pattern based Statistical Web Testing and Reliability Measurement," *Procedia Comput. Sci.*, vol. 21, pp. 140–147, Jan. 2013.
- [9] L. Borzemski, M. Danielak, and A. Kamińska-Chuchmała, "Spatio-temporal Web Performance Prediction: Turning Bands Method and Sequential Gaussian Simulation," *Procedia Comput. Sci.*, vol. 96, pp. 568–576, Jan. 2016.
- [10] R. Peña-Ortiz, J. A. Gil, J. Sahuquillo, and A. Pont, "Analyzing web server performance under dynamic user workloads," *Comput. Commun.*, vol. 36, no. 4, pp. 386–395, Feb. 2013.
- [11] S. Boukhris, A. Andrews, A. Alhaddad, and R. Dewri, "A case study of black box fail-safe testing in web applications," *J. Syst. Softw.*, vol. 131, pp. 146–167, Sep. 2017.
- [12] P. Ackermann, E. Vlachogiannis, and C. A. Velasco, "Developing Advanced Accessibility Conformance Tools for the Ubiquitous Web," *Procedia Comput. Sci.*, vol. 67, pp. 452–457, Jan. 2015.
- [13] C. Guerrero, I. Lera, and C. Juiz, "Performance improvement of web caching in Web 2.0 via knowledge discovery," *J. Syst. Softw.*, vol. 86, no. 12, pp. 2970–2980, Dec. 2013.
- [14] B. de la Ossa, J. A. Gil, J. Sahuquillo, and A. Pont, "Referrer Graph: A cost-effective algorithm and pruning method for predicting web accesses," *Comput. Commun.*, vol. 36, no. 8, pp. 881–894, May 2013.
- [15] A. Sehati and M. Ghaderi, "Network assisted latency reduction for mobile web browsing," *Comput. Networks*, vol. 106, pp. 134–150, Sep. 2016.
- [16] K. S. Shailesh and P. V. Suresh, "Web performance optimization through smart resource and asset optimizations," in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, 2016, pp. 277–281.
- [17] P. Li, D. Shi, and J. Li, "Performance test and bottle analysis based on scientific research management platform," in *2013 10th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, 2013, pp. 218–221.
- [18] M. Flores, A. Wenzel, and A. Kuzmanovic, "Oak: User-Targeted Web Performance," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, 2017, pp. 2654–2655.
- [19] R. Pena-Ortiz, J. A. Gil, J. Sahuquillo, and A. Pont, "The Impact of User's Dynamic Behavior on Web Performance," in *2012 IEEE 11th International Symposium on Network Computing and Applications*, 2012, pp. 143–150.
- [20] A. S. Asrese, P. Sarolahti, M. Boye, and J. Ott, "WePR: A Tool for Automated Web Performance Measurement," in *2016 IEEE Globecom Workshops (GC Wkshps)*, 2016, pp. 1–6.
- [21] M. Leotta, A. Stocco, F. Ricca, and P. Tonella, "Using Multi-Locators to Increase the Robustness of Web Test Cases," in *2015 IEEE 8th International Conference on Software Testing, Verification and Validation (ICST)*, 2015, pp. 1–10.
- [22] Y. Nakano, N. Kamiyama, K. Shiimoto, G. Hasegawa, M. Murata, and H. Miyahara, "Web performance acceleration by caching rendering results," in *2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2015, pp. 244–249.
- [23] M. Leotta, A. Stocco, F. Ricca, and P. Tonella, "Reducing Web Test Cases Aging by Means of Robust XPath Locators," in *2014 IEEE International Symposium on Software Reliability Engineering Workshops*, 2014, pp. 449–454.
- [24] S. J. Eravuchira, V. Bajpai, J. Schonwalder, and S. Crawford, "Measuring web similarity from dual-stacked hosts," in *2016 12th International Conference on Network and Service Management (CNSM)*, 2016, pp. 181–187.
- [25] V. Jain and A. Kolambkar, "Modeling Web Attachment Storage for Web Applications," in *2014 21st Asia-Pacific Software Engineering Conference*, 2014, pp. 98–102.
- [26] I. Iliev and G. P. Dimitrov, "Front end optimization methods and their effect," in *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2014, pp. 467–473.
- [27] S. Mirshokraie, A. Mesbah, and K. Pattabiraman, "JSEFT: Automated Javascript Unit Test Generation," in *2015 IEEE 8th International Conference on Software Testing, Verification and Validation (ICST)*, 2015, pp. 1–10.
- [28] H. Zhang, S. Zhang, M. Xiong, and S. Tang, "Improving Performance of E-Government System from the User Perspective," in *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, 2013, pp. 2108–2113.
- [29] P. Ahammad et al., "A flexible platform for QoE-driven delivery of image-rich web applications," in *2015 IEEE International Conference on Multimedia and Expo (ICME)*, 2015, pp. 1–6.
- [30] S. Rochimah, S. Arifiani, and V. F. Insanittaqwa, "Non-Source Code Refactoring: A Systematic Literature Review," *Int. J. Softw. Eng. Its Appl.*, vol. 9, no. 6, pp. 197–214, Jun. 2015.