

Browser Army++: Sistem Eksekusi Test Terdistribusi dengan Antarmuka Pengembangan dan Pengelolaan Test Selenium = Browser Army++: Distributed Test Execution System with Selenium Test Development and Management Interface

Steven Novaryo, author

Deskripsi Lengkap: <https://lib.ui.ac.id/detail?id=9999920551719&lokasi=lokal>

Abstrak

Dalam pengembangan aplikasi web, pengujian merupakan komponen penting untuk memastikan aplikasi yang dapat diandalkan dan meminimalisasi kesalahan. Salah satu bentuk automasi pengujian aplikasi web adalah menggunakan Selenium. Masalah yang dihadapi dalam penggunaan Selenium adalah kasus ketika automasi memerlukan tenaga komputasi yang besar dan dapat dikustomisasi. Selain itu, dalam pemeliharaan test, dibutuhkan waktu dan kompetensi yang memadai untuk membuat test. Pada penelitian ini, penulis membuat sistem infrastruktur test terdistribusi berbasis Selenium yang memperbolehkan kustomisasi tipe infrastruktur, mesin, dan lokasi perjalanan test dari pengguna. Terdapat dua opsi arsitektur yang diberikan untuk menyesuaikan kebutuhan pengguna, yaitu infrastruktur on-demand dan persistent. Sistem infrastruktur tersebut dilengkapi dengan antarmuka berbasis Selenium IDE yang memberikan aktivitas record dan playback untuk mempermudah pembuatan test. Terakhir, karena perbedaan sistem eksekusi skrip di Selenium IDE, sistem dilengkapi dengan worker yang memiliki kapabilitas untuk melakukan eksekusi skrip dengan metode yang menyesuaikan eksekusi skrip dalam Selenium IDE (menggunakan side-runner) dan metode yang menyesuaikan eksekusi skrip menggunakan bahasa pemrograman populer, Python. Dari hasil evaluasi, infrastruktur on-demand memiliki durasi runtime 2.27 sampai 3.21 kali lipat dibandingkan infrastruktur persistent. Infrastruktur on-demand juga menggunakan 7.3% CPU dan 3.9% RAM lebih banyak dibandingkan persistent. Namun, biaya yang diperlukan infrastruktur persistent lebih besar 45% daripada on-demand ketika dievaluasi. Di sisi lain, komponen antarmuka dinilai memudahkan pembuatan test pada use case yang ditetapkan. Fitur yang berpengaruh akan hal tersebut adalah fitur record dan playback. Terakhir, dalam evaluasi, worker Python memiliki durasi runtime yang relatif lebih rendah daripada eksekusi side-runner. Pada kategori use case berita dan navigasi, side-runner memiliki tingkat keberhasilan eksekusi yang lebih rendah dibandingkan dengan Python karena permasalahan timeout. Di sisi lain, pada kategori use case sosial media, Python memiliki tingkat keberhasilan eksekusi yang lebih rendah dibandingkan dengan side-runner karena permasalahan StaleElementReferenceException.

.....In web application development, testing is a crucial component to ensure a reliable application and minimize errors. One form of web application testing automation is using Selenium. A problem encountered when using Selenium is the need for significant computational power and customization. Additionally, maintaining tests requires adequate time and competence to create tests. In this research, the author developed a distributed test infrastructure system based on Selenium that allows customization of infrastructure type, machine, and location by the user. Two architectural options are provided to meet user needs: on-demand infrastructure and persistent infrastructure. The system is equipped with a Selenium IDE-based interface that offers record and playback activities to simplify test creation. Lastly, due to differences in script execution systems in Selenium IDE, the system is equipped with a worker capable of executing scripts using methods that align with Selenium IDE (using side-runner) and methods that align with popular

programming languages, Python. From evaluation, on-demand infrastructure has a runtime duration of 2.27 to 3.21 times that of persistent infrastructure. On-demand infrastructure also uses 7.3% more CPU and 3.9% more RAM compared to persistent infrastructure. However, the cost required for persistent infrastructure is 45% higher than on-demand when evaluated. On the other hand, the interface components are considered to simplify test creation in the specified use cases. The features that influence this are the record and playback features. Lastly, in the evaluation, the Python worker has a relatively lower runtime duration compared to side-runner execution. In the news and navigation use case category, side-runner has a lower execution success rate compared to Python due to timeout issues. Whereas, in the social media use case category, Python has a lower execution success rate compared to side-runner due to `StaleElementReferenceException` issues.