# Penggunaan Aplikasi *Selenium* Untuk *Automatic Testing* Pada Metodologi *DevOps*

**CANDRA GUNAWAN[1], DENY PRATAMA[2], DEWI APRIYANI[3], PUTRI SARI SIGIRO[4], AND ANESIA PUJI KINANTI[5]**

[1] *Magister Manajement Sistem Informasi, Universitas Gunadarma, 2016, Jakarta*
[1] *Corresponding author: sherard@outlook.com*
[2] *Magister Manajement Sistem Informasi, Universitas Gunadarma, 2016, Jakarta*
[2] *Corresponding author: denyuno@gmail.com*
[3] *Magister Manajement Sistem Informasi, Universitas Gunadarma, 2016, Jakarta*
[3] *Corresponding author: uwiart93@gmail.com*
[4] *Magister Manajement Sistem Informasi, Universitas Gunadarma, 2016, Jakarta*
[4] *Corresponding author: putri.sigiro@gmail.com*
[5] *Magister Manajement Sistem Informasi, Universitas Gunadarma, 2016, Jakarta*
[5] *Corresponding author: Anesiaakinanti@gmail.com*

*Compiled using LaTeX August 12, 2016*

***DevOps* adalah suatu metode pengembangan *software* yang menekankan komunikasi, kolaborasi dan integrasi antara pengembang *software* dan profesional TI dengan menjadikan proses pembuatan *software* dan perubahan infrastruktur secara otomatis. Metode ini merupakan implementasi dari *"Agile Infrastructure"* yang sebelum nya sudah diperkenalkan. *Selenium* IDE adalah *Integrated Tool* untuk agile testing. Mekanismenya bekerja dengan mereka semua aktifitas user saat mengakses suatu aplikasi berbasis web. Dan selanjut nya, test bisa dilakukan secara otomatis. Tool ini merupakan salah satu tool pendukung di dalam *DevOps*, yaitu untuk melakukan *Test Automation*.**



***submision codes:***

http://Gunadarma.ac.id

## 1. PENDAHULUAN

### A. Latar Belakang Masalah

*Agile* merupakan salah satu metodologi yang diperkenalkan untuk menutup kekurangan metodologi tradisional. Dimana prinsip dari Agile adalah mampu ber-adaptasi terhadap perubahan kebutuhan / (*Requirement*). *Agile* sebagai metodologi, dapa di implementasikan di berbagai macam kebutuhan. Untuk menunjang metodologi *Agile* ini, maka diperlukan berbagai macam tool yang bisa digunakan oleh praktisi *Agile*. Untuk menunjang metode *Agile* tersebut, maka diperlukan suatu alat penunjang. Maka, para praktisi *Agile* sepakat untuk membuat yang disebut *Agile Infrastructure*.

*Agile Infrastructure* adalah suatu cara mendefinisikan berbagai macam alat / *tools* baik berupa software, hardware maupun proses yang dapat menunjang *Agile*. Dengan *Agile Infrastructure*, diharapkan bisa menunjang agar proses *Agile* bisa lebih efisien. Pertama kali, *Agile Infrastructure* ini diperkenalkan oleh Patrick Debois pada 2009. Dan akhirnya, saat ini lebih dikenal dengan *DevOps*.

*Devops* adalah metode pengembangan *software* yang menekankan komunikasi, kolaborasi dan integrasi antara pengembang *softwware* dan profesional TI dengan menjadikan proses pembuatan *software* dan perubahan infrastruktur secara otomatis. Adopsi dari *DevOps* ini mencakup :

- Menggunakan *Agile* sebagai metodologi
- Terdapat nya kebutuhan untuk mempercepat waktu pengembangan aplikasi
- *Wide Availability* dan penggunaan *cloud* infrastructure
- Proses otomatis *Environment* dan konfigurasi
- Proses testing otomatis dan *Continuous Integration*

Dari area *DevOps* di atas, dalam penulisan ini yang dibahas adalah proses testing otomatis / *Automation Testing*. Tool yang akan digunakan adalah *Selenium Web Browser*.

*Selenium Web Browser* adalah salah satu contoh *tool* untuk melakukan *automated test* untuk aplikasi web. Jadi dengan tool ini, aplikasi web dapat dilakukan test secara otomatis.

### B. Rumusan Masalah

Rumusan masalah pada jurnal **"Penggunaan Aplikasi *Selenium* Untuk *Automatic Testing* Pada Metodologi *DevOps*"** adalah untuk menjelasakan bagaimana implementasi *automated testing* dengan menggunakan *tool selenium* dan kaitan nya dengan metodologi *DevOps*.

### C. Tujuan Penulisan

Tujuan penulisan ini adalah untuk menjelaskan implementasi dari *DevOps* di salah satu area, yaitu area *testing*. Dimana untuk tool yang digunakan adalah selenium.

## 2. LANDASAN TEORI

### A. Capability Maturity Model

*Capabilty Maturity Model* disingkat **CMM** adalah suatu model kematangan kemampuan (kapabilitas) proses yang dapat membantu pendefinisian dan pemahaman proses - proses suatu organisasi. Pengembangan model ini dimulai pada tahun 1986 oleh **SEI** *(Software Enginering Institute)*. CMM awalnya digunakan sebagai alat ukur untuk secara objectif menilai kemampuan dalam menangani proyek perangkat lunak, dan diharapkan dapat digunakan sebagai suatu model umum untuk membantu pemahaman kematangan kapabilitas proses organisasi.

| | Initial | Managed | Defined | Quantitatively Managed | Optimizing |
|---|---|---|---|---|---|
| Culture & Organization | • Teams organized based on platform/ technology<br>• Defined and documented processes | • One backlog per team<br>• Adopt agile methodologies<br>• Remove team boundaries | • Extended team collaboration<br>• Remove boundary dev/ ops<br>• Common process for all changes | • Cross-team continuous improvement<br>• Teams responsible all the way to production | • Cross functional teams |
| Build & Deploy | • Centralized version control<br>• Automated build scripts<br>• No management of artifacts<br>• Manual deployment<br>• Environments are manually provisioned | • Polling CI builds<br>• Any build can be re-created from source control<br>• Management of build artifacts<br>• Automated deployment scripts<br>• Automated provisioning of environments | • Commit hook CI builds<br>• Build fails if quality is not met (code analysis, performance, etc.)<br>• Push button deployment and release of any releasable artifact to any environment<br>• Standard deployment process for all environments | • Team priorities keeping codebase deployable over doing new work<br>• Builds are not left broken<br>• Orchestrated deployments<br>• Blue Green Deployments | • Zero touch Continuous Deployments |
| Release | • Infrequent and unreliable releases<br>• Manual process | • Painful infrequent but reliable releases | • Infrequent but fully automated and reliable releases in any environment | • Frequent fully automated releases<br>• Deployment disconnected from release<br>• Canary releases | • No rollbacks, always roll forward |
| Data Management | • Data migrations are performed manually, no scripts | • Data migrations using versioned scripts, performed manually | • Automated and versioned changes to datastores | • Changes to datastores automatically performed as part of the deployment process | • Automatic datastore changes and rollbacks tested with every deployment |
| Test & Verification | • Automated unit tests<br>• Separate test environment | • Automatic Integration Tests<br>• Static code analysis<br>• Test coverage analysis | • Automatic functional tests<br>• Manual performance/ security tests | • Fully automatic acceptance tests<br>• Automatic performance/security tests<br>• Manual exploratory testing based on risk based testing analysis | • Verify expected business value<br>• Defects found and fixed immediately (roll forward) |
| Information & Reporting | • Baseline process metrics<br>• Manual reporting<br>• Visible to report runner | • Measure the process<br>• Automatic reporting<br>• Visible to team | • Automatic generation of release notes<br>• Pipeline traceability<br>• Reporting history<br>• Visible to cross-silo | • Report trend analysis<br>• Real time graphs on deployment pipeline metrics | • Dynamic self-service of information<br>• Customizable dashboards<br>• Cross-reference across organizational boundaries |

**Fig. 1.** Capability Maturity Model

### B. Agile Development Method

***Agile Development Methods*** adalah sekelompok metodologi pengembangan perangkat lunak yang didasarkan pada prinsip-prinsip yang sama atau pengembangan sistem jangka pendek yang memerlukan adaptasi cepat dari pengembang terhadap perubahan dalam bentuk apapun. Agile development methods merupakan salah satu dari Metodologi pengembangan perangkat lunak yang digunakan dalam pengembangan perangkat lunak. Agile memiliki pengertian bersifat cepat, ringan, bebas bergerak, dan waspada. Sehingga saat membuat perangkat lunak dengan menggunakan agile development methods diperlukan inovasi dan responsibiliti yang baik antara tim pengembang dan klien agar kualitas dari perangkat lunak yang dihasilkan bagus dan kelincahan dari tim seimbang.

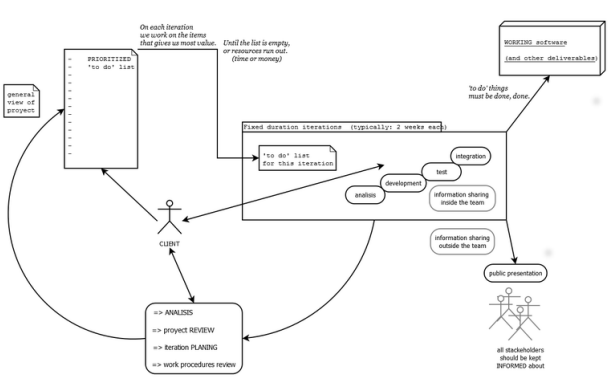


**Fig. 2.** Agile Methodology

Agile development methods terdefinisi dalam empat nilai, biasa di sebut ***Agile Alliances Manifesto***. Berikut ini isi dari ***Agile Alliances Manifesto [1]*** :

*Manifesto for Agile Software Development*

**We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:**

**Individuals and interactions** over processes and tools
**Working software** over comprehensive documentation
**Customer collaboration** over contract negotiation
**Responding to change over** following a plan

**That is, while there is value in the items on the right, we value the items on the left more.**

Dari ***Agile Manifesto*** terdefinisi dalam empat nilai :

- **Interaksi dan personel** lebih penting daripada proses dan alat.
- **Perangkat lunak yang berfungsi** lebih penting daripada dokumentasi yang lengkap.
- **Kolaborasi dengan klien** lebih penting daripada negosiasi kontrak.
- **Respon terhadap perubahan** lebih penting daripada mengikuti rencana.

### C. DevOps

DevOps adalah metode pengembangan software yang menekankan komunikasi, kolaborasi dan integrasi antara pengembang software dan profesional TI dengan menjadikan proses pembuatan software dan perubahan *infrastructure* secara otomatis. Metode ini pertama kali diperkenalkan pada tahun 2008 oleh **Patrick Debois** pada diskusi nya tentang *"Agile Infrastructure"*, tetapi metode ini baru dikenal sebagai *DevOps* sejavak *DevOpsDay* tahun 2009 di belgia.Bisa disimpulkan DevOps adalah suatu proses *Agile* yang di dukung oleh berbagai *tool* atau infrastructure.

Dengan *DevOps*, bagian pengembangan dan operasional akan bekerja lebih dekat. Tujuan nya adalah mengurangi friksi antara kedua tim serta meningkatkan kecepatan kerja dengan meng-otomatiskan beberapa proses, sehingga dapat mencapai kinerja optimal. Salah satu lembaga analis ternama

di dunia - *Gartner* - merumuskan *DevOps* ini dalam suatu *Blue Print DevOps Architecture*[2].

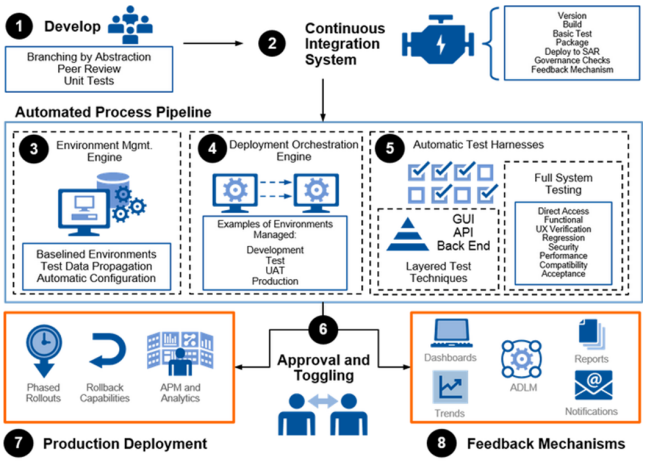


**Fig. 3.** DevOps Blueprint Architecture By Gartner

Di dalam blueprint menggambarkan suatu siklus *software development* yang terdiri atas :

- *Software Development Process* dan *versioning system*
- *Continuous Integration System*
- *Automatic Environment Management Engine*
- *Automatic Deployment Orchestration Engine*
- *Automatic Test*
- *Approval*
- *Production Deployment*
- *Feedback Mechanism*

khusus untuk di pembahasan ini, yang menjadi fokus utama area yang dibahas adalah *Automatic Test*. Berdasarkan *Gartner*, *Automatic test* area di definisikan sebagai : *Integrate the orchestration engine with automated test frameworks that check quality and efficacy for all aspects of the software-testing process.* [2] Yang pada prinsip nya adalah *Automatic Test* adalah proses integrasi terhadap suatu test framework dan digunakan untuk cek kualitas dan efisiensi semua aspek yang ada di dalam *software-testing* proses.

## 3. PEMBAHASAN

### A. Selenium Web Browser

Selenium adalah salah satu aplikasi yang digunakan untuk membuat *automated testing* khusus nya untuk aplikasi web. Tool ini pada prinsip nya digunakan oleh seorang *Quality Assurance Engineer* untuk membuat proses testing suatu aplikasi yang berbasi web dibuat secara otomatis. Di tool ini, bisa disiapkan suatu script yang nanti nya akan menjalankan *engine* selenium untuk melakukan tes terhadap aplikasi web secara otomatis bertahap.

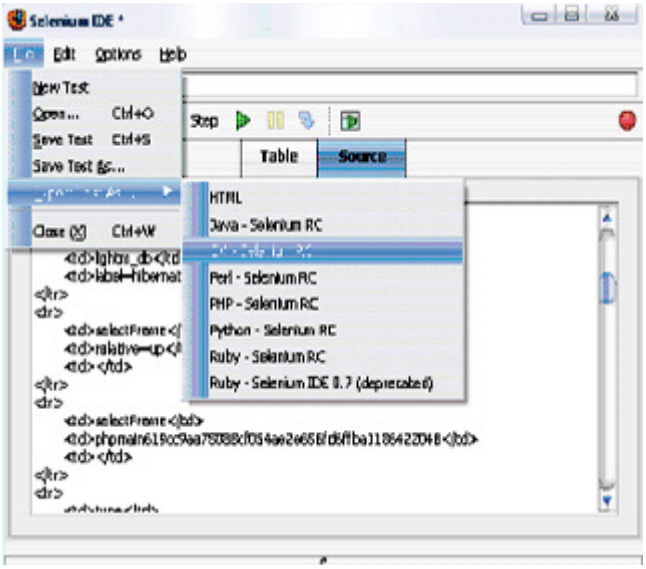


**Fig. 4.** Selenium IDE

Proses *automated test* ini merupakan salah satu dari komponent yang ada di dalam *DevOps blue print*. Dengan ada nya komponen ini, target dari *DevOps* yaitu agar mempercepat waktu siklus *development software* diharapkan bisa tercapai.

### B. Menggunakan Selenium

Mekanisme *Automated Test* disebut juga *Agile Test*, yaitu software tester berperilaku sebagai end-user dan memeriksa apakah sistemnya dapat berfungsi dengan baik sesuai requirement dari user. terjadi nya *Agile Test* tersebut, karena semua aktivitas test direkam oleh sebuah tools, dimana saat melakukan test ulang, bisa secara langsung menjalankan rekaman aktivitas kita, dan tools tersebut akan mengetes ulang secara otomatis. Dalam selenium setiap aktivitas akan terekam masuk ke dalam tiga *tag*, yaitu *command*, *target*, *value*.

- *Command*, berisi apa yang dilakukan user pada aplikasi.
- *Target*, berisi tujuan dari *command* dilakukan.
- *Value*, berisi nilai yang dimasukkan dan di akses saat melakukan test.

#### *B.1. Fitur yang terdapat pada Selenium IDE*

- Dapat merekam dan *re-play* hasil rekaman dengan mudah
- *Autocomplete* untuk selenium *command*
- dapat dijalankan selama testing berlangsung
- *Debug* dan *set breakpoint*

- bisa disimpan dalam file format berbagai bahasa pemrograman.
- mendukung *javascript*
- secara otomatis menambahkan *title* pada setiap halaman

### *B.2. Cara Menggunakan Selenium*

Selenium IDE bisa langsung di download sebagai *add-ons* **mozilla firefox** atau browser lain nya yang mendukung. Selenium IDE ter-integrasi dengan *browser* dan bisa langsung digunakan dengan menuliskan alamat *URL* yang akan ditest, kemudian klik tombol *record*.


**Fig. 5.** IDE

untuk menjalankan dan mengetes kembali secara otomatis hanya meng-klik tombol *play*.

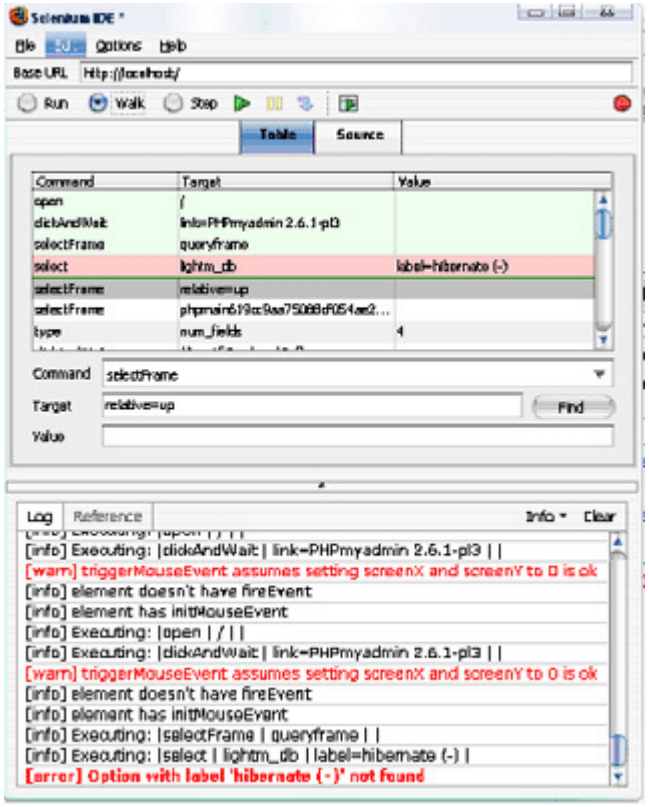


**Fig. 6.** Reply IDE

apabila sukses, test akan berwarna hijau, sedangkan kalau gagal, test akan berwarna merah. Kemudian tester menjalankan sistem selayaknya *end-user*, setelah selesai hasilnya bisa disimpan dalam bentuk HTML dan beberapa format yang bisa dipilih seperti pada ilustrasi berikut :

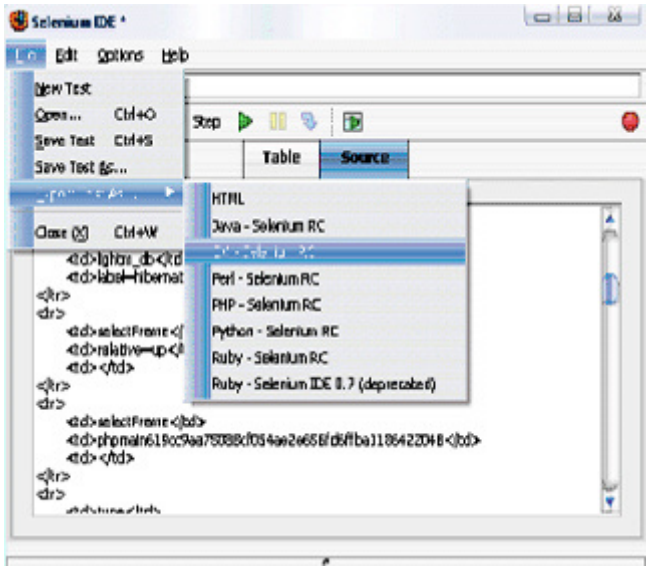


**Fig. 7.** save

Hasil output *selenium IDE* tadi merupakan satu buah unit test. Kemudian beberapa unit test disatukan dalam suatu *test suite*. *Test Suite* akan meng-otomatisasi software testing secara keseluruhan. Tools yang digunakan yaitu selenium core.

## 4. KESIMPULAN

### A. Rangkuman

*DevOps* sebagai implementasi *Agile Infrastructure* di siklus nya membuat beberapa aktivitas menjadi otomatis. Salah satu nya adalah otomatis testing unit. Untuk mencapai proses otomatis tersebut, maka salah satu tool yang bisa digunakan adalah *Selenium*. Dengan menggunakan *selenium* maka proses testing, terutama untuk aplikasi web, yang sebelum nya hanya dilakukan manual, bisa dilakukan otomatis. Cara nya adalah dengan me-*record* aktivitas testing sebelum nya untuk digunakan di kemudian hari. Sehingga proses nya lebih optimal, dan bisa lebih meng-optimalkan siklus *development*.

### B. Saran

Adapun dari penulisan berikut ini, ada beberapa saran yang penulis rangkum di dalam penulisan ini :

- Penulis baru menggambarkan relasi antara maturity model, Agile, DevOps dan *Testing Automation*.
- Penulis baru menerangkan proses DevOps di siklus *Testing Automation* dengan menggunakan selenium.
- Penulis belum menggambarkan perbandingan antara sebelum dan sesudah menggunakan *Automation Testing*.

## 5. DAFTAR PUSTAKA

## REFERENCES

1. B. Kent and et al, “Manifesto for agile software development,” Manifesto for Agile Software Development (2010).
2. S. Kenefick, “Blueprint for architecting a devops continuous delivery pipeline,” Blueprint for Architecting a DevOps Continuous Delivery Pipeline (2015).