**TOOLS AUTOMATION**

1. **KATALON STUDIO**

**Kelebihan menggunakan Katalon Studio**

1. Terintegrasi dengan alur kerja CI/DevOps dan alat lainnya

Katalon tidak memerlukan ekstensi tambahan untuk menjalankan pengujian karena menawarkan berbagai [integrasi](https://www.katalon.com/Integrations/) praktis. Misalnya, menghubungkan ke qTest, alat manajemen pengujian terpusat yang memfasilitasi komunikasi tim QA, memungkinkan untuk mengendalikan persyaratan pengujian, perencanaan kasus uji, dan manajemen cacat. Pengguna juga dapat dengan mudah berintegrasi dengan platform manajemen SDLC (software development lifecycle) lainnya seperti JIRA, TestRail, dan TestLink.

Katalon terintegrasi dengan sejumlah alat yang digunakan untuk mendukung CI /CD dan [DevOps](https://www.altexsoft.com/blog/engineering/devops-principles-practices-and-devops-engineer-role/) seperti Jenkins, Bamboo, TeamCity, CircleCI, dan Travis CI. Diperkenalkan pada Mei 2021, [Katalon Studio 8](https://docs.katalon.com/katalon-studio/new/what-is-new-800.html" \l "continuous-testing-with-azure-test-plans-integration) menyertakan integrasi asli dengan Azure DevOps Services, yaitu, Azure Test Plans. Sekarang, pengguna dapat memetakan Kasus Uji di Azure DevOps ke Kasus Pengujian otomatis di Katalon Studio serta secara otomatis mengirim log dan laporan Eksekusi Pengujian dari Katalon Studio ke Test Run di Azure DevOps. Katalon Studio juga mendukung integrasi dengan platform kolaborasi (Git, Microsoft Teams, dan Slack) dan platform eksekusi (SauceLabs, BrowserStack, Selenium Grid, dan Kobiton).

1. Dasbor dan laporan analitik intuitif

Modul pengujian berbasis data bawaan memungkinkan pengguna untuk menganalisis pengujian berdasarkan skenario yang direkam. Katalon menampilkan hasil analitik dalam bentuk laporan bawaan yang dapat diekspor dalam PDF, HTML, Excel, atau CSV. Laporan ini intuitif secara visual dan mudah digunakan.

Katalon Studio Enterprise menawarkan [lebih banyak opsi pelaporan](https://docs.katalon.com/katalon-studio/docs/test-suite-report.html" \l "test-suite-report), yaitu, laporan pengumpulan rangkaian uji dan riwayat laporan, video untuk browser tanpa kepala, dll.Bahkan lebih banyak kemampuan pelaporan dan analitis tersedia melalui integrasi alat TestOps.

1. Mendukung beberapa jenis pengujian

Di Katalon Studio Anda dapat menjalankan jenis pengujian berikut:

* **Kata kunci-driven**. Pendekatan ini bekerja untuk penguji yang tidak mahir dalam scripting. Berdasarkan kata kunci yang mewakili tindakan pengguna pada AUT (Applications Under Test), penguji dapat menjalankan tes otomatis. Kata kunci dapat disesuaikan.
* **Berbasis data**. Strategi eksekusi pengujian yang menampilkan pembacaan skrip uji dari berbagai sumber data. Data diambil dari kumpulan data, file Excel, objek ADO, file CSV, dan sumber ODBC.
* **Pengujian API.** Ini mendukung XLS, sumber data XML, dan database dengan pemetaan dinamis untuk cakupan pengujian yang lebih baik. Selain itu, tes ini sesuai dengan Cucumber, kerangka kerja berbasis BDD (pengembangan berbasis perilaku) untuk tes penerimaan aplikasi web.
* **Lintas browser.** Katalon mendukung semua browser populer dan memungkinkan menjalankan tes otomatis di Chrome, Firefox, Internet Explorer, Safari, Edge, Remote, Headless, dan lingkungan khusus.

1. GUI yang ramah pengguna

UI Katalon menawarkan grafis yang kaya dengan tampilan pohon, tabel, dan menu yang memungkinkan pengguna untuk mengelola artefak uji dengan mudah. Penguji menemukan antarmuka alat yang nyaman, ramah pengguna, bersih, dan mudah dipahami bahkan untuk pengguna non-teknis.

1. Kurva belajar ringan + materi pembelajaran

Katalon Studio terkenal dengan kesederhanaannya, baik instalasi maupun penggunaannya. Sangat mudah untuk mengatur berkat banyak paket terintegrasi. Ini bagus untuk pemula dan pengguna tingkat lanjut, yang dapat menggunakan seluruh IDE dan menulis skrip lanjutan. Pengembang memiliki akses ke sejumlah video pelatihan dan dokumentasi untuk pengguna: Katalon memiliki repositori GitHub, daftar kursus online di Udemy, dan kursus video di YouTube.

1. Tersedia versi gratis

Katalon memiliki versi gratis yang berfungsi penuh, yang dianggap sebagai keuntungan oleh sebagian besar penggunanya. Namun, versi Enterprise menawarkan lebih banyak fitur dan akses ke semua plugin Katalon dan lisensi offline, jadi disarankan untuk tim dan proyek yang dapat diskalakan, sementara edisi dasar gratis ditawarkan untuk penggunaan individual.

**Kontra menggunakan Katalon Studio**

1. Kurangnya bahasa scripting

Tidak seperti Selenium dan TestComplete, satu-satunya bahasa scripting yang didukung oleh Katalon adalah Groovy. Bahasa scripting ini ada dalam keluarga Java sehingga siapa pun yang tahu Java dapat menggunakannya. Namun, pengguna ingin melihat lebih banyak bahasa yang didukung.

1. Komunitas kecil

Sejak Katalon dikembangkan pada tahun 2015, komunitas ini lebih kecil daripada pesaing yang lebih matang, dan Anda mungkin mengalami kurangnya dukungan dari penguji sebaya. Jadi, hanya untuk perbandingan, pada hari pembaruan ini, ada [706 pertanyaan](https://stackoverflow.com/questions/tagged/katalon-studio) yang ditandai Katalon Studio di Stack Overflow, sedangkan untuk Selenium ada 87.555 pertanyaan. Namun, menurut Capterra, Gartner, dan Stack Overflow, Katalon jauh lebih populer daripada Ranorex dan alat otomatisasi uji lainnya. Ada [forum Katalon](https://forum.katalon.com/) yang berdedikasi dan komunitas yang berkembang [di GitHub](https://github.com/katalon-studio/katalon-studio), sehingga ketika produk matang dan memperoleh lebih banyak pengagum, menjadi lebih mudah untuk menemukan jawaban atas pertanyaan yang muncul.

1. Bukan open source

Alat ini memiliki kode sumber tertutup, menghasilkan sejumlah kecil pengembang di komunitas. Selenium, salah satu pesaing utama Katalon, adalah alat open-source yang memungkinkan insinyur untuk menyesuaikannya atau menggunakan paket buatan komunitas. Namun, Katalon Studio memiliki beberapa [komponen terbuka](https://forum.katalon.com/t/more-open-source-components-from-katalon-studio/15764), kerangka kerja Katalium sumber terbuka, dan Toko Katalon sumber terbuka dengan plugin yang telah kami jelaskan di atas.

1. Masalah kinerja

Pengguna melaporkan beberapa bug yang menghambat pengujian, memperlambatnya. Misalnya, terkadang alat membeku atau mungkin mulai tertinggal, verifikasi teks dan objek dalam iframe bermasalah. Pengujian seluler membutuhkan lebih banyak waktu karena kebutuhan untuk menangkap dan menulis kode.

1. **SELENIUM**

**Kelebihan menggunakan Selenium**

1. Gratis

Selenium bukan satu-satunya alat pengujian otomatisasi di pasaran, tetapi satu-satunya yang gratis yang dapat bersaing dengan produk berbayar. Seperti yang mungkin telah Anda lihat dalam [perbandingan alat pengujian otomatis](https://www.altexsoft.com/blog/engineering/comparing-automated-testing-tools-selenium-testcomplete-ranorex-and-more/" \t "_blank). Katalon Studio adalah satu-satunya alternatif yang layak tetapi tidak memberikan pilihan bahasa yang besar, tidak berfungsi di Linux, dan sama sekali tidak memiliki basis penggemar di belakangnya. Status open-source Selenium membuat alat dengan kurva belajar yang curam menjadi titik masuk yang rendah untuk startup dan pengembang independen. Tidak mengherankan bahwa bahkan perusahaan yang lebih besar tidak terburu-buru untuk beralih ke opsi berbayar dan memberikan beberapa ribu dolar ditambah pemeliharaan ketika Selenium masih bertahan.

1. Terintegrasi dengan alur kerja Agile, DevOps, Continuous Delivery

Selenium mengikuti narasi Agile dan DevOps bahkan sebelum kata-kata ini dipraktikkan. Seluruh sifat Selenium jatuh dengan sempurna ke dalam prinsip-prinsip utama di balik [Agile](https://www.altexsoft.com/whitepapers/agile-project-management-best-practices-and-methodologies/), [DevOps](https://www.altexsoft.com/blog/engineering/devops-principles-practices-and-devops-engineer-role/), atau [Continuous Delivery](https://www.altexsoft.com/blog/business/continuous-delivery-and-integration-rapid-updates-by-automating-quality-assurance/" \t "_blank). Bagaimana tepatnya ini dicapai?

* Karena portabel di semua platform dan seringkali tidak memerlukan pembelajaran bahasa baru, Selenium memungkinkan fleksibilitas yang tak tertandingi.
* Selenium mudah diintegrasikan dengan berbagai platform pengembangan seperti Jenkins, Maven, TestNG, QMetry, SauceLabs, dll.
* Pengujian paralel dan lintas browser dengan Selenium Grid dan di luar cloud-grid seperti [LambdaTest](https://www.lambdatest.com/) memungkinkan pengembang untuk menerima umpan balik lebih cepat dan bekerja pada perubahan segera daripada menunggu semalam untuk lulus tes.

1. Mendukung pengujian seluler

Dengan Selenium, juga dapat menguji [aplikasi seluler asli, hibrida, atau web](https://www.altexsoft.com/blog/mobile/key-approaches-to-mobile-development-explained/), meskipun Anda memerlukan perangkat lunak tambahan. Ada dua pilihan utama [Appium](http://appium.io/" \t "_blank) dan [Selendroid](http://selendroid.io/" \t "_blank). Keduanya didasarkan pada Selenium sehingga pengembang yang sudah berpengalaman di dalamnya dapat menerapkan prinsip yang sama saat menguji aplikasi seluler.

1. Berbagai bahasa, platform, dan browser yang didukung

Fleksibilitas yang diberikan Selenium hampir tak tertandingi di dunia otomatisasi pengujian. Pertama, kami memiliki sepuluh bahasa yang didukung, di antaranya adalah Java, Ruby, C #, PHP, JavaScript, dan Python, semuanya di antara [bahasa pemrograman yang paling banyak digunakan](https://insights.stackoverflow.com/survey/2020" \l "most-popular-technologies" \t "_blank) saat ini. Ini juga satu-satunya alat utama yang mencakup pengujian Linux. Untuk daftar terperinci dan terbaru versi browser yang didukung.

1. Komunitas besar

Selenium mengambil pangsa pasar 27,48 persen yang mengejutkan dari semua alat pengujian perangkat lunak, dengan pesaing terdekatnya Apache Jmeter mengambil lebih dari 10 persen statistik luar biasa.

Sebagai salah satu pelopor dalam pengujian otomatis modern, Selenium mengakuisisi basis penggemar pengembang di pemain utama seperti Google dan startup juga. Daftar pekerjaan untuk profesional QA termasuk kemahiran Selenium sebagai salah satu keterampilan yang diperlukan.

1. Perpustakaan plugin dan ekstensi yang besar

Selenium dapat diperluas di luar fungsi standarnya dengan berbagai plugin. Beberapa dari mereka agak secara resmi didukung oleh Selenium (seperti Appium dan Selendroid).

**Kekurangan menggunakan Selenium**

1. Kurva belajar yang curam

Salah satu tren pengujian otomatis saat ini adalah pengujian tanpa kode. Pendekatan ini memungkinkan siapa saja tanpa pengetahuan pemrograman yang mendalam untuk melakukan tes. Selenium tidak memungkinkan untuk pengujian tanpa kode. Pengguna memerlukan pemahaman yang baik tentang salah satu bahasa pemrograman yang mempersempit kumpulan orang yang dapat terlibat dalam proses pengujian. Banyak perusahaan, terutama startup, cenderung menggunakan coders terbaik mereka untuk menulis fitur produk dan melibatkan orang-orang yang kurang terampil dalam tes otomatis. Ini tidak akan bekerja dengan Selenium.

1. Hanya digunakan untuk aplikasi berbasis web

Selenium adalah alat yang tidak dapat digunakan untuk mengotomatisasi pengujian aplikasi desktop karena tidak dapat mengenali objek di aplikasi desktop. Ini hanya ditargetkan untuk melakukan tes aplikasi web, menggunakan berbagai browser yang tercantum di atas. Jadi, untuk menguji aplikasi desktop, harus menemukan alat terpisah seperti WinAppDriver dan menggunakannya bersama dengan Selenium (mereka dapat diintegrasikan dengan mudah), atau instrumen pengujian komprehensif seperti [Katalon](https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-katalon-studio-automation-testing-tool/) yang dapat mengotomatiskan tes berbasis web dan non-web.

1. Tidak ada perbandingan gambar bawaan

Dalam proses QA, ada kesulitan dalam mengotomatisasi verifikasi gambar. Cara sederhana dan paling efektif untuk memutuskan apakah gambar ditampilkan dengan benar di layar adalah dengan memeriksa secara manual. Meskipun normal bagi sebagian besar produksi untuk menggunakan [kombinasi pengujian manual dan otomatis](https://www.altexsoft.com/blog/engineering/striking-a-balance-between-manual-and-automated-testing-when-two-is-better-than-one/" \t "_blank), ini adalah sumber bug yang besar karena tugas segera menjadi berulang dan penguji cenderung memeriksa lebih jarang.

1. Tidak ada dukungan teknis

Sisi buruk dari ini adalah bahwa komunitaslah yang memberikan bantuan, bukan pembuat produk. Menemukan solusi untuk masalah yang tidak konvensional atau bantuan dengan permintaan khusus adalah mungkin, tetapi sulit. Jika diperlukan, dapat menghubungi perusahaan konsultan atau layanan untuk dukungan komersial. Selenium menyediakan daftar singkat perusahaan tersebut di [halaman ekosistem mereka.](https://www.seleniumhq.org/ecosystem/" \t "_blank).

1. Tidak ada kemampuan pelaporan

Kurangnya laporan yang dihasilkan secara otomatis adalah salah satu tantangan terbesar selenium. Untuk menangkap kegagalan pengujian di Selenium, harus mengambil tangkapan layar pada saat kegagalan. Ini jauh dari format yang dapat dibaca bahwa tim perlu dengan cepat mendiagnosis masalah. Sementara vendor yang berbeda menawarkan fungsionalitas pelaporan dengan wawasan berbasis data dan integrasi alat kerja tim, Selenium harus bergantung pada solusi pihak ketiga.

1. **APPIUM**

**Kelebihan menggunakan Appium**

1. Dengan menggunakan API yang sama, Appium akan memungkinkan Anda untuk menulis tes yang bertentangan dengan platform seluler.
2. Dengan menggunakan segala jenis pekerjaan kerangka uji atau bahasa Anda dapat menulis dan menjalankan tes.
3. Appium adalah platform open source sehingga Anda dapat berkontribusi dengan mudah.
4. Untuk aplikasi seluler hibrida dan Native, Appium menyediakan lintas platform.
5. Appium mendukung protokol kawat JSON.
6. Appium tidak memerlukan rekompilasi Aplikasi.
7. Appium juga mendukung uji otomatisasi pada perangkat fisik dan juga untuk simulator atau emulator keduanya.
8. Appium tidak memiliki ketergantungan pada perangkat seluler.

**Kekurangan penggunaan Appium**

1. Pengujian android yang lebih rendah dari 4.2 tidak diperbolehkan.
2. Appium memiliki dukungan terbatas untuk pengujian aplikasi hibrida. Anda tidak akan dapat menguji tindakan yang memungkinkan pengalihan aplikasi dari aplikasi asli ke aplikasi web dan dari aplikasi web ke native.
3. Tidak ada dukungan yang memungkinkan Anda menjalankan appium inspector di Microsoft windows.
4. **CYPRESS.IO**

**Kelebihan Menggunakan Cypress**

1. Menginstal Cypress tidak memerlukan dependensi atau unduhan tambahan.
2. Cypress digunakan oleh pengembang dan insinyur QA, sedangkan Selenium semata-mata untuk otomatisasi uji. Cypress didasarkan pada JavaScript, yang merupakan bahasa pemrograman front-end yang populer.
3. Di Cypress, tidak ada OVERHEAD IDE tambahan. Saat Anda meluncurkan Cypress, ia meminta Anda untuk memilih IDE untuk memodifikasi skrip uji.
4. Jika dibandingkan dengan Selenium, kerangka Cypress menghasilkan hasil yang lebih akurat. Itu karena Cypress memiliki kontrol yang lebih besar atas seluruh proses otomatisasi, memungkinkannya untuk memiliki pemahaman yang baik tentang apa yang terjadi di dalam dan di luar browser.
5. Instans Cypress merespons secara real-time terhadap peristiwa dan perintah aplikasi. Reload real-time di Cypress secara otomatis memuat ulang tes saat perubahan dilakukan dalam aplikasi.

**Kekurangan menggunakan Cypress**

1. Cypress saat ini hanya didukung untuk browser Chrome, Firefox, Edge, Brave, dan Electron. Akibatnya, Cypress adalah pilihan yang kurang disukai untuk pengujian lintas browser.
2. Untuk pembuatan kasus uji, itu hanya mendukung kerangka kerja JavaScript.
3. Cypress tidak mendukung eksekusi jarak jauh.
4. **ROBOT FRAMEWORK**

**Kelebihan menggunakan Robot Framework**

1. Pengguna tidak memerlukan bahasa pemrograman untuk menulis kasus uji Robot Framework & menjalankannya
2. Mendukung pendekatan *Keyword driven*, *Data-driven* dan *Behavior-driven* (BDD)
3. Penggunaan pustaka Selenium2 dalam RF akan membantu seseorang mengotomatiskan kasus pengujian manual yang ada dari sebuah proyek dalam Fase Cepat
4. Sistem Pelaporan Luar Biasa: RF secara otomatis menghasilkan laporan dan log html setelah menjalankan setiap build
5. Robot Framework menyediakan banyak pustaka untuk menguji aplikasi yang berbeda seperti Appium Library for Mobile Automation, Database Library untuk pengujian DB, Android Library dll.
6. Robot Framework adalah alat otomatisasi sumber terbuka gratis dan mudah dipasang.
7. Robot Framework memungkinkan penggunaan Gherkin.
8. Robot Framework bekerja untuk aplikasi seluler dan web.
9. Dasar-dasar Robot Framework mudah dipelajari.
10. Robot Framework menggunakan pendekatan low-code.
11. Robot Framework kompatibel dengan Windows, macOS, Unix, dan Linux.
12. Robot Framework mendukung Jython, IronPython, dan PyPy.
13. Robot Framework mendukung semua browser utama.
14. Robot Framework membuat laporan hasil dalam tiga format.

Nilai tambah yang besar adalah bahwa Robot Framework bekerja pada semua sistem operasi utama. Meskipun browser Microsoft Edge tersedia untuk Windows, macOS, Android, dan iOS, ada baiknya untuk melakukan skrip uji Robot pada semua sistem operasi yang didukung. Kelebihan lain untuk menggunakan Robot Framework adalah ia bekerja dengan baik dengan perpustakaan Appium dan Selenium. Hal ini memungkinkan penguji dan pengembang untuk menggunakan dan meningkatkan pengetahuan mereka yang ada tentang perpustakaan tersebut.

**Kekurangan menggunakan robot framework**

1. Robot Framework sulit untuk dipertahankan.
2. Laporan HTML Kerangka Kerja Robot sulit untuk disesuaikan.
3. Robot Framework tidak mendukung eksekusi uji paralel.
4. Beberapa kesalahan Kerangka Kerja Robot sulit untuk di-debug.
5. Robot Framework memiliki aturan lekukan yang ketat.
6. Kurang fleksibilitas - Tidak ada fitur seperti tag penulis, skip test, No nested loop, dll
7. Menjalankan skrip uji di beberapa browser dan platform seluler secara bersamaan adalah fitur hemat waktu yang tidak dimiliki Robot Framework.[https://ssl.microsofttranslator.com/static/27420612/img/tooltip_logo.gif](http://www.bing.com/translator)https://ssl.microsofttranslator.com/static/27420612/img/tooltip_close.gif

**Original**

Running test scripts on multiple browsers and mobile platforms simultaneously is a time-saving feature that Robot Framework lacks

[https://ssl.microsofttranslator.com/static/27420612/img/tooltip_logo.gif](http://www.bing.com/translator)https://ssl.microsofttranslator.com/static/27420612/img/tooltip_close.gif

**Original**

+ Free