

Reading Material

Memahami Testing Principles - Pengenalan Testing tools



READING

Testing Tools

Tahukah kamu jika pengujian software zaman sekarang sudah didukung oleh berbagai tools yang memudahkan dan membuat pekerjaan lebih efektif?

Sumber Testing Tools?

Kita dapat dengan mudah mencari tools testing di internet, ada berbagai tipe tools, mulai dari berbayar hingga gratis. Sebetulnya tidak ada tools yang sama tapi sebagian tools memiliki solusi untuk hal yang sama dengan cara berbeda sesuai dengan teknologi yang dipakai dari tools tersebut. Contohnya pengujian pada API dapat dilakukan dengan berbagai tools, salah satunya adalah Postman. Postman memiliki kemampuan yang berbeda dengan tools lain yakni Insomnia meskipun keduanya bisa digunakan untuk testing REST API namun insomnia belum mendukung websocket API. Jadi, tentukan software testing tools yang memang cocok untuk proses pengujian-mu.

Kategori Testing Tools

Software testing tools bisa dikategorikan berdasarkan kategori, teknologi yang digunakan, tipe testing dan lain sebagainya. Secara general software testing tool dapat dibagi menjadi beberapa antara lain:

1. Test management
2. Bug tracking tool
3. Automated testing tool
4. Performance testing tool
5. Unit testing tool

Berbagai Testing Tools

Test Management Tool

Tool ini digunakan pada dasarnya untuk menyimpan semua testcase yang kita miliki. Sebetulnya berbeda tool pasti memiliki perbedaan fitur namun secara garis besar manfaat dari test management tool antara lain

1. Meningkatkan performa testing dan software quality assurance
2. Mempermudah dokumentasi data dan proses test
3. Meningkatkan kualitas dari software

Contoh:

1. Qase.io

Qase memiliki tampilan yang menarik, user experience yang modern, minimalis dan dilengkapi dengan berbagai widget yang dapat disesuaikan dengan kebutuhan masing-masing cara testing atau pengguna. Qase memiliki fitur antara lain:



- a. Test case management
- b. Bug/Defect Management
- c. Test Plans
- d. Informatif report
- e. API
- f. Webhooks

2. Test Rail

Memiliki semboyan *centralized and scalable* yang berarti penggunaan test rail memang untuk membuat proses testing



dan seluruh data ada di dalam sebuah platform yang sama. TestRail memiliki fitur antara lain:

- a. Mendukung berbagai tipe test dan metode test.
- b. Semua data akan terpusat, memiliki fitur import dari excel yang akan memudahkan untuk migrasi test case ke testrail.

- c. Memiliki fitur pencarian dengan berbagai filter bantuan

Bug Tracking Tool

Bug atau cacat pada sebuah software harus ditulis secara detail, hal ini karena tim pengembang sangat butuh informasi tersebut untuk proses perbaikan dan mengurangi waktu back and forth pertanyaan ke penguji. Informasi tersebut harus disimpan ke sebuah database penyimpanan yang nantinya akan bermanfaat antara lain:

1. Membantu kita untuk melacak bug-bug pada version tertentu
2. Memberikan laporan kepada client apabila dibutuhkan
3. Mempermudah proses analisis bug pada software yang akan bermanfaat untuk tahap pengujian dan pengembangan software kedepan
4. Akan mempermudah end-user untuk melaporkan bug secara langsung dari aplikasi mereka

Contoh dari tool untuk bug tracking:

1. BugHerd

Software tool yang khusus digunakan untuk mengumpulkan, mengorganisir, dan manajemen feedback selama proses fixing bug. Bugherd memiliki feature antara lain:



- a. Fitur assign bug to developer
- b. Mendukung KanBan workflow
- c. Mendukung integrasi dengan pihak ketiga

2. Jira

Tool dari atlassian yang saat ini banyak digunakan oleh pengembang aplikasi dalam manajemen task ataupun issue/bug. Tampilan sederhana dominan biru dan putih dengan tulisan yang jelas akan membuat user experience lebih nyaman. Jira mendukung berbagai macam tipe SDLC. Berikut beberapa manfaat Jira:



- a. Menyediakan berbagai tipe isu sehingga memudahkan untuk manajemen project, task, dan tentunya bug.
- b. Bisa langsung berkolaborasi dengan semua anggota team yang ada di Jira dan saling memberikan umpan balik hingga masalah selesai.
- c. Jira memiliki berbagai 3rd party tool yang dapat diintegrasikan secara langsung.

Automation Testing Tool

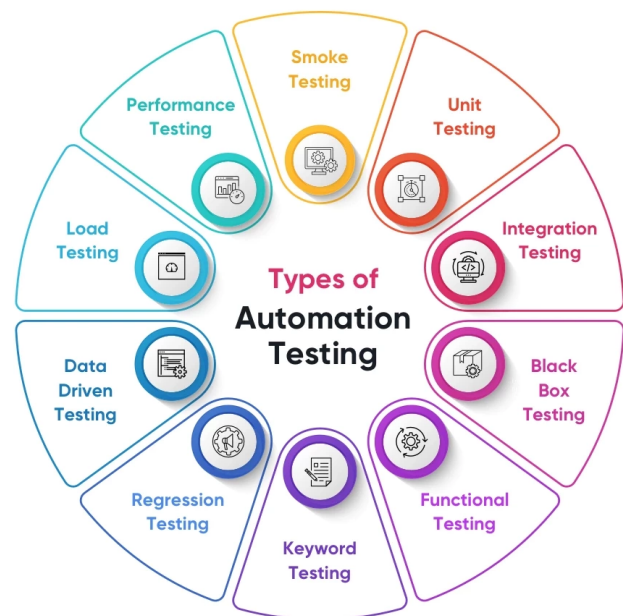
Tool ini digunakan untuk memberikan peningkatan produktivitas software dan meningkatkan tingkat keakuratan dari sebuah proses pengujian. Banyak sekali tipe automation tool, ada yang high level seperti dalam bentuk tool GUI atau code level yang pembuatannya menggunakan teknik scripting. Contoh tool adalah cypress.io untuk membuat UI web automation, postman untuk testing API, dll. Beberapa manfaat dari automation testing tool:

- Menghemat waktu
- Hemat biaya
- Lebih Akurat
- Concurrent Test
- Pelaporan yang jelas
- Pelacakan bug
- Mempercepat proses development

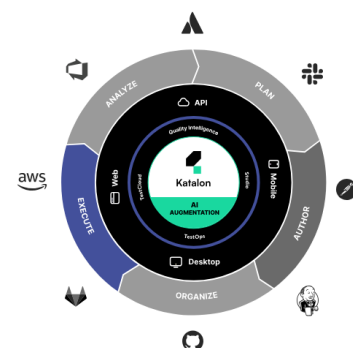
Contoh tool pada automation testing:

1. Katalon Studio

Merupakan salah satu aplikasi dengan fitur cukup lengkap dan sering digunakan oleh software developer. Katalon selalu aktif memperbaiki teknologi dan sekarang sudah menggunakan AI



net solutions

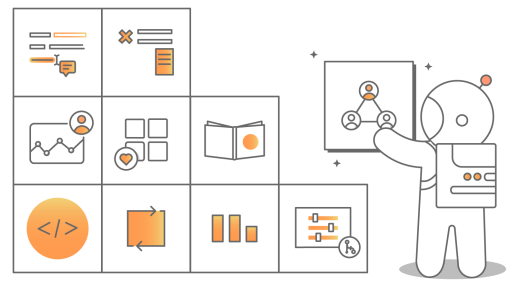


Augmentation, sangat layak untuk dicoba. Beberapa fitur unggulan katalon adalah:

- a. Desktop App Testing bisa memudahkan untuk pemula automation tester karena bisa menggunakan GUI untuk menjalankan test sesuai kebutuhan, memanfaatkan fitur record, playback, dan object spying untuk UI testing.
- b. Private plugin disediakan oleh Katalon untuk tester yang ingin mendesain dan membuat plugin di dalam internal team.
- c. Test suite collection digunakan untuk mempermudah proses manajemen testing dan report.

2. Postman

Sebuah tool untuk API testing yang sangat populer dan memiliki fitur sangat lengkap sehingga cukup untuk membantu tester melakukan API test dan bekerja dalam tim dengan mudah.



Berikut beberapa fitur dari Postman:

- a. Postman Visualizer, berguna untuk membuat hasil test lebih informatif dan mudah dibaca dengan chart yang menarik.
- b. Built-in Libraries, Postman menyediakan berbagai fitur seperti Faker.js yang bisa digunakan untuk membuat data-data test, Lodash yang menyediakan berbagai fungsi untuk testing, Moments digunakan untuk format data, dll.
- c. Collaboration Features yang bisa digunakan untuk bekerja bersama tim, memberikan komentar, feedback, dan melakukan revisi test bersama.
- d. Collection-Level Informative, merupakan kelebihan postman karena bisa manajemen data test, variable, scripts, auth credential sehingga bisa dipakai untuk multiple requests.

Performance Testing Tool

Tool yang digunakan untuk melakukan performance test. Performance test bertujuan untuk menguji ketahanan aplikasi dari berbagai sisi seperti sisi server dan sisi keamanan. Contoh performance testing tool adalah K6, Gatling, JMeter, dll. Contoh tool:

1. K6

Merupakan sebuah open-source tool yang bisa digunakan untuk performance test. Dengan menggunakan k6 dapat menguji performance system dan mengetahui issue terkait performance system lebih awal. k6 akan membantu develop software yang tangguh dan berperforma tinggi.



2. Apache JMeter

Sebuah multipurpose tool selain untuk performance testing bisa digunakan untuk spike testing dan unit test. JMeter memiliki beberapa keunggulan antara lain open source, sudah mendukung browser plugin untuk web testing, bisa menganalisa report secara offline, memiliki GUI dan sudah support report visualize ke chart, tables atau trees.



Unit Testing Tool

Unit testing tool digunakan untuk menguji bagian terkecil dari sebuah aplikasi yaitu fungsi. Unit testing erat kaitannya dengan programmer atau software developer, yang berarti sebagian besar akan dikerjakan oleh pembuat software tersebut. Contoh unit test tool adalah pyunit, mocha, chai, NUnit, JUnit, JMockit, HtmlUnit, dll.

Manfaat dan Resiko Penggunaan Testing Tool

Software testing tool dapat bermanfaat untuk

1. meningkatkan performance software
2. Deliver produk berkualitas tinggi
3. mengurangi durasi testing
4. membantu proses pengujian lebih terorganisir dan terdokumentasi

5. Meningkatkan efisiensi kegiatan pengujian
6. Mengotomatiskan aktivitas yang membutuhkan sumber daya signifikan saat dilakukan secara manual
7. Mengotomatiskan aktivitas yang tidak dapat dijalankan secara manual.

Selain itu testing tool bisa memberikan potensi resiko seperti:

1. Terlalu berharap adanya perubahan untuk mencapai manfaat yang signifikan
2. Upaya untuk memelihara test assets
3. Terlalu mengandalkan alat
4. Mengabaikan interoperabilitas antara alat-alat penting
5. Dukungan vendor yang buruk
6. Mengabaikan kontrol versi
7. Risiko penangguhan proyek alat sumber terbuka/gratis
8. Security data, credentials, semua informasi tentang aplikasi atau perusahaan kita.