

Nama : Nurul Cesar Ardayanti
Program : Studi Independen

SUMMARY

Self Paced Learning : Unit Testing Javascript

Instruktur KODE Dibawakan oleh RizaFAHMI (Co-Founder HACKTIV8)

SELAMAT DATANG : Perkenalan dan Prasyarat. Unit Testing Javascript adalah hal yang penting dalam pemrograman terutama dalam job description. seorang developer saat ini sudah sedikit berubah, jadi dengan kemampuan melakukan tes secara otomatis value sebagai developer akan meningkat jauh. Skill yang dibutuhkan, Javascript Dasar dan NodeJS Dasar, GIT&GitHub. Karena akan mendemokan aplikasi client browser dan juga di sisi server.

TENTANG TESTING : Testing, memastikan kode berjalan dengan semestinya, testing menghindari kesalahan fatal, memudahkan proses refactor, testing memaksa menulis kode yang lebih baik, testing memiliki analogi yang sama seperti pengaman kesalahan atau safety net dan secara umum dapat mengurangi bugs atau defect pada aplikasi. Beberapa Contoh Bug Krusial : The Heartbleed Bug, dimana user dan password orang tersebar karena ada bug di SSL/TLS. karena programmer lupa menyatakan panjang dari sebuah string, Rocket NASA, karena kesalahan ketik atau typo. roket tiba-tiba meledak yang merugikan. Dan Orbiter NASA yang hilang, karena salah kalkulasi. bagaimana Menghindari Bug dengan cara Static Code Analysis (Linter), bisa menangkap kesalahan sederhana variable yang lupa di definisikan. Static Type seperti Flow, TypeScript, dan ReasonML bisa menghindari perbandingan data dan variable. Dan yang terakhir melakukan testing.

Jenis - Jenis Testing. Beberapa Tipe Testing yaitu Unit Testing yang akan mengetes fungsi tunggal dari sebuah aplikasi, Integration Testing yang melakukan tes terhadap dua atau lebih fungsi secara menyeluruh dan terakhir End to End test yang melakukan testing dengan simulasi perspektif pisert. Ada juga Testing Trophy.

UNIT TESTING. Tools Unit Testing adalah melakukan tes terhadap unit terkecil dari sebuah aplikasi atau project dalam hal ini fungsi, membantu dalam hal mengelola kode, dan terakhir unit testing dapat mengurangi bugs dan defect di aplikasi. Options yang dipilih yaitu ada Test Runner, Assertion Library, Mock, Stub dll, dan juga Cove Coverage.

JEST UNTUK UNIT TESTING/JEST TESTING FRAMEWORK. Jest adalah testing framework Javascript yang fokus kepada penggunaannya yang cukup mudah. Jest di buat oleh Facebook. Ada beberapa fitur di jest yaitu test runner, menginput library Assertion, memiliki mocking fitur dan memiliki code Coverage yang di suport oleh jest.

Demo Penggunaan Jest. Describe yaitu kumpulan berbagai testcase. mendemokan penggunaan jest mulai dari instalasi kemudian menggunakan grup cash discribe kemudian test equal, membuat tes case. Melakukan test demo dari instalasi, toBe, toEqual, listOfApperentices dan toMatchObject.

Unit Testing Aplikasi Kalkulator. melakukan testing terhadap aplikasi yang sudah running yaitu aplikasi sederhana kalkulator. aplikasi ini dapat menambah,mengurangi,membagi dan mengkalikan dua angka yang diberikan. pertama meskipun aplikasi sederhana yang menambahkan, mengurangi, kali dan bagi. tes yang pertama yaitu fungsi penambahan dan dari fungsi penambahan harus tes apakah penawaran yang benar penambahan desimal penambahan dengan string dan lain sebagainya. sehingga aplikasi menjadi jumlah bugs yang jadi sedikit. mendemokan bagaimana menambahkan antara number dengan string menjadi string itu adalah bugs dengan testing berhasil menghindari bugs tersebut. mensimulasikan bagaimana membuat sebuah bugs dan kemudian gimana cara membuat simulasi membasmi bug dan menghindari bahaya tersebut terjadi lagi dengan cara menjalankan testing.

UNIT Testing Aplikasi Cart. menghindari beberapa bugs, ada banyak bugs yang sudah ditandai yaitu pengurangan item dari satu menjadi nol dihindari itu ataupun nol menjadi minus 1, kemudian menghindari bugs dari penjumlahan item barang yang ditambahkan dengan stream, lalu implementasi satu total dan membuat tesnya, serta implementasi fitur kode promo dan membuat testnya.

Testing Back end NodeJS. Testing ini menggunakan aplikasi E-commerce sederhana. melakukan testing terhadap model yang diambil dari get itemsnya, kemudian melakukan demo terhadap dua fitur penting di dalam testing yaitu beforeEach dan afterEach.

TEKNIK TESTING : Test-Driven Development dan Teknik Dalam Testing. Ada beberapa teknik dalam testing yaitu : TDD - Aplikasi Testing Driven Development, di testing ini biasanya bertanya apakah cara yang dilakukan sudah benar atau belum? biasanya dilakukan oleh developer atau programmer. BDD - behind Demo Development, teknik ini biasanya menanyakan apakah fungsi ini perlu di test? Biasanya dilakukan oleh developer customer/enr user dan

seorang tester. **Dan** ATDD - Acts as Test Demo Development, teknik ini biasanya menanyakan apakah kode ini sudah berjalan sesuai yang diinginkan atau belum?. biasanya dilakukan oleh customer/end user kemudian developer dan seorang tester. TDD adalah pendekatan untuk membangun sebuah aplikasi, TDD memaksa untuk berpikir lebih jelas dan lebih jernih tentang apa yang mau dibuat. Red-Green-Refactor yaitu red membuat testing dulu dan lihat testingnya benar-benar failed atau gagal.

Demo TDD. Melakukan testing terhadap Rest API yang kita testing adalah method GET dan method POST.

UNIT TESTING DAN CONTINUOUS INTEGRATION : Continuous Integration Yaitu memastikan kode yang di-push atau di-merge ke repositori aman, bebas dari bugs, error dan melewati fase testing. mengimplementasikan CI harus menggunakan testing baik itu tdd ataupun tanpa tdd.

Demo TDD. mendemonstrasikan cara kerja continuous integration, melakukan set up terhadap Workflow GIT, GitHub, dan Continuous Integration dengan practice. mencoba beberapa fitur TDD, continuous integration dan juga github untuk memproteksi apabila ada bugs yang muncul di kode sehingga semua kode yang ada di master tetap aman.

CODE COVERAGE : cara perhitungannya cukup simpel karena tools ini akan menghitung baris kode mana yang sudah dieksekusi oleh testing dan mana yang belum, menjadi tolak ukur seberapa banyak tes yang dibutuhkan dari sebuah unit ataupun fungsi.

Code Coverage : Demo. Melakukan code Coverage aplikasi kalkulator, menggunakan Workflow GitHub dan CI, proteksi branch master di GitHub dan mensimulasikan bug & proses review kode.