

Writing Programmatic Test
by: Dikie Rendra Aditya

Testing has always been a common process of software development. In general, it usually take place by running the application, then the programmer or tester navigate into the corresponding use case to test it by specifying input and analyzing the output. This routine is easy to perform and developer see the same way that end user sees from the application. However there are some drawbacks to this method which are: it is performed manually, repetitive, limited ability to test non-visible components, and other developer can't simply verify proper functioning of the application.

Writing programmatic test is about writing another code to test the application code. It seems like more work for developer at first, but actually it can really help in the long run especially for a large scale application. There are two way of programmatic testing: functional testing and unit testing. Functional testing is somewhat similar to manual testing by performing test just into the higher level by omitting knowledge of the internal implementation of specific features. The difference is that functional testing are written into a code not performed manually by hand. In contrast with functional testing, unit testing works in a lower level of code functionality, it required knowledge of an internal components. To make it simple, functional test perform simulation of input and button click of the application, where unit test perform simulation by call of the components function or method.

With programmatic testing developer can minimize the burden of repetitive task from manual testing by using the computer to perform them. Because the testing task is perform by computer it would run by the same sequence of actions, this reduce the error that is potentially generated by human which tends to overlook details. It is also possible to test the internal detail of the components such as the business logic, or software library. Developer can inspect the implementation details such as algorithms efficiency by calculating the time each time the tests is invoke. Finally this method can greatly prevent bugs that might arise in the application.

There are several tools out there that is specifically used to make writing programmatic test even more appealing, one of the most popular tools JUnit is a simple open source Java testing framework written by Kent Beck and Erich Gamma, available to download at www.junit.org. JUnit feature including assertion for testing the expected result and reporting of testing result. Most of the Java IDE right now provide built in JUnit support that make writing programmatic tests is easy and fun.