

**Nama : Andira Faqih Muhammad**  
**NIM : 24060120140142**  
**Kelas UPL B**

***Resume Kuliah Tamu : Automation in Software Quality Control and Testing Process***

Dalam perkembangan industri perangkat lunak, penggunaan otomatisasi dalam kontrol kualitas dan pengujian perangkat lunak menjadi sangat penting. Otomatisasi ini melibatkan penggunaan alat dan *framework* khusus untuk menjalankan pengujian secara otomatis, dengan tujuan meningkatkan efisiensi, mengurangi upaya manual, dan meningkatkan kualitas perangkat lunak secara keseluruhan. Review ini akan memberikan gambaran tentang otomatisasi dalam kontrol kualitas dan pengujian perangkat lunak, manfaatnya, tantangannya, dan praktik terbaik yang dapat diikuti.

Pertama-tama, untuk memahami peran otomatisasi dalam kontrol kualitas dan pengujian perangkat lunak, penting untuk memahami konsep dari pengembangan perangkat lunak. Pengembangan perangkat lunak melibatkan proses terorganisir untuk menciptakan produk yang dapat memenuhi tujuan dan sasaran bisnis. Proses pengembangan perangkat lunak meliputi tahap-tahap seperti analisis kebutuhan, membuat *design* dan *prototype*, pengembangan *software*, pengujian, *deployment*, *maintenance*.

Salah satu aspek penting dalam pengembangan perangkat lunak adalah Quality Assurance (QA) atau jaminan kualitas. QA adalah praktik yang bertujuan untuk memastikan bahwa perangkat lunak yang dikembangkan memenuhi standar kualitas yang ditetapkan. Untuk memastikan pemenuhan faktor kualitas ini, perlu disusun Software Quality Assurance Plan (SQA Plan) yang merinci strategi, metode, dan alat yang akan digunakan dalam pengujian dan pengawasan kualitas perangkat lunak. Selain itu, juga terdapat standar SQA yang harus diikuti untuk memastikan keseragaman dan kualitas hasil akhir perangkat lunak.

Salah satu tahap penting dalam pengujian perangkat lunak adalah software testing atau pengujian perangkat lunak. Pengujian perangkat lunak adalah proses untuk memverifikasi dan memvalidasi bahwa perangkat lunak berfungsi sesuai dengan kebutuhan yang telah ditentukan. Tujuan dari pengujian perangkat lunak adalah untuk menemukan bug atau kesalahan dalam perangkat lunak sehingga dapat diperbaiki sebelum dirilis. Terdapat berbagai metode pengujian yang dapat digunakan, seperti pengujian fungsional, pengujian regresi, pengujian integrasi, dan pengujian kinerja. Proses pengujian melibatkan perencanaan pengujian, desain kasus pengujian, eksekusi pengujian, dan pelaporan hasil pengujian. Selain itu, pengujian perangkat lunak juga mengikuti siklus hidup pengujian yang mencakup tahap persiapan, perencanaan, desain, eksekusi, dan penutupan.

Dalam konteks otomatisasi pengujian perangkat lunak, pengujian otomatis memainkan peran penting. Pengujian otomatis melibatkan penggunaan alat dan kerangka kerja otomatisasi untuk menjalankan kasus pengujian secara otomatis. Manfaat utama dari pengujian otomatis adalah meningkatkan efisiensi dan efektivitas pengujian. Dengan otomatisasi, pengujian dapat dilakukan dengan cepat dan secara konsisten, mengurangi upaya manual yang repetitive, dan meningkatkan cakupan pengujian. Pengujian otomatis juga memungkinkan pengujian regresi yang efisien, dengan memastikan bahwa perubahan atau penambahan fitur baru tidak mempengaruhi fungsi yang ada. Dalam pengembangan perangkat lunak yang menggunakan metodologi Agile, pengujian otomatis juga memungkinkan integrasi yang lancar dengan praktik pengiriman berkelanjutan (continuous delivery).

Untuk mendukung otomatisasi pengujian perangkat lunak, terdapat berbagai jenis alat pengujian perangkat lunak yang tersedia. Alat-alat ini membantu dalam pembuatan dan eksekusi kasus pengujian otomatis, manajemen pengujian, pelaporan hasil pengujian, dan kolaborasi tim pengembang. Beberapa contoh alat pengujian perangkat lunak yang umum digunakan adalah Selenium, Appium, JUnit, TestNG, Cucumber, dan Jenkins. Pemilihan alat pengujian yang tepat sangat penting untuk mencapai tujuan pengujian otomatis yang efektif dan efisien.

Secara umum, otomatisasi dalam kontrol kualitas dan pengujian perangkat lunak merupakan komponen yang sangat penting dalam pengembangan perangkat lunak modern. Dengan menerapkan otomatisasi, tim pengembang perangkat lunak dapat meningkatkan efisiensi, meningkatkan kualitas perangkat lunak, dan mengurangi waktu pengembangan secara keseluruhan. Namun, ada beberapa tantangan yang perlu diatasi, seperti biaya awal, kompleksitas implementasi, dan pemeliharaan otomatisasi. Untuk mengatasinya, praktik terbaik harus diterapkan dan alat yang sesuai harus dipilih.