

1. Pendahuluan

Latar Belakang

Pengujian perangkat lunak menjadi hal yang penting dalam menentukan kualitas perangkat lunak. Lebih dari 50% bagian dari pengembangan perangkat lunak dihabiskan untuk pengujian [1] [2]. Pengujian dilakukan untuk menemukan kesalahan/*defect* pada perangkat lunak. Penguji perlu untuk merencanakan strategi pengujian, mulai dari pembuatan kasus uji, eksekusi dari kasus uji dan juga harus dicek apakah *expected result* dengan *actual result* sudah sesuai yang mana apabila dilakukan secara manual akan memakan waktu dan memerlukan kontribusi manusia [3]. Maka pembuatan kasus uji merupakan proses penting yang dilakukan pertama kali pada proses pengujian [4]. Pembuatan kasus uji dapat dilakukan secara manual ataupun otomatis.

Kasus uji dapat dihasilkan dengan dua pendekatan yaitu, dari desain kode program perangkat lunak/*code based* atau desain spesifikasi kebutuhan/*model based* menggunakan *Unified Model Language (UML) behavioral diagram*. Pembangkitan kasus uji dari spesifikasi kebutuhan/*model based* lebih efisien dari waktu dan usahanya, dibandingkan *code based* [4]. Terdapat beberapa model diagram pada UML yang umum digunakan untuk membangkitkan kasus uji seperti *statechart diagram*, *sequence diagram*, *activity diagram* [5].

Pada penelitian yang dilakukan hashim [5], model *statechart diagram* lebih baik dibandingkan model diagram lainnya karena diagram ini menunjukkan perilaku sistem dan perubahan yang terjadi antara *event* dan transisi. *Statechart diagram* digunakan sebagai model untuk membangkitkan kasus uji. Maka pada penelitian ini akan melakukan pembangunan sistem untuk pembangkitan kasus uji otomatis menggunakan model *statechart diagram* dengan algoritma *Depth First Search (DFS)* dan menerapkan metode *equivalence partitioning* untuk memvalidasi antara spesifikasi hasil kasus uji.

Topik

Pengujian diperlukan untuk menjamin kualitas suatu perangkat lunak. Pembangunan kasus uji dengan model *statechart diagram* diperlukan validasi agar mendapatkan kasus uji yang benar, maka diperlukan pendefinisian kasus uji untuk memudahkan proses pembangunan kasus uji untuk melakukan pengujian perangkat lunak. Sehingga dapat dirumuskan pada rumusan masalah sebagai berikut

1. Bagaimana membangkitkan kasus uji berbasis *statechart diagram*?
2. Bagaimana kesesuaian antara kasus uji yang dibangkitkan?

Batasan Masalah

Adapun batasan masalah pada penelitian ini sebagai berikut:

1. Pembangkitan kasus uji pada 5 fitur aplikasi web kepegawaian yaitu *login*, *view profile*, *change password*, *change photo*, dan *view absen*, karena hanya ada 5 fitur yang tersedia pada aplikasi web.
2. Untuk menganalisis kasus uji yang dihasilkan sistem menggunakan selenium.

Tujuan

Tujuan dari penelitian ini adalah melakukan pembangunan sistem pembangkitan kasus uji otomatis pengujian perangkat lunak dengan menerapkan model *statechart diagram*.

Organisasi Tulisan

Pada bagian selanjutnya, bagian 2 mengenai studi pustaka tentang teori dan metode terkait penelitian, bagian 3 mengenai rancangan dari sistem yang akan dibangun pada penelitian, bagian 4 mengenai hasil dan analisis dari sistem, dan pada bagian 5 merupakan kesimpulan dari penelitian.