

ABSTRAK

Kasus uji adalah bagian penting untuk melakukan pengujian perangkat lunak karena memainkan peran yang sangat penting dalam menentukan langkah-langkah pengujian yang akan dilakukan. Kasus uji dapat dibuat secara manual atau otomatis. Membuat kasus uji secara manual memiliki beberapa kelemahan, seperti probabilitas yang lebih tinggi dalam menghasilkan hasil yang tidak akurat. Selain itu, pembuatan kasus uji secara manual tidak efektif karena cara manual akan memakan waktu lebih lama dan membutuhkan tenaga yang besar. Oleh karena itu, membuat kasus uji secara otomatis lebih baik untuk membatasi masalah-masalah sebelumnya. Pengujian otomatis juga dapat memberikan hasil yang lebih akurat daripada pengujian manual karena tidak terlalu rentan terhadap kesalahan manusia sehingga mendorong pengembang perangkat lunak untuk membuat kasus uji secara otomatis. Kasus uji dapat dihasilkan dari berbagai sumber, termasuk diagram UML. Dalam makalah ini, kami mengusulkan sebuah metode untuk menghasilkan kasus uji secara otomatis dengan memanfaatkan algoritma *depth-first search*. Diagram UML yang digunakan dalam penelitian ini adalah diagram aktivitas. Makalah ini juga memberikan metode untuk mengurai diagram aktivitas yang dibuat menggunakan aplikasi *visual paradigm* yang telah dikonversi ke dalam format XML. Metode yang kami usulkan mampu mengurai dan mengubah format XML menjadi bentuk grafik atau *tree*. Dengan menggunakan metode yang kami usulkan, pencarian kasus uji akan dilakukan dengan menggunakan *depth-first search* pada bentuk *tree* untuk menghasilkan kasus uji. Terakhir, kami juga mengevaluasi metode kami dengan membandingkan kasus uji yang dihasilkan dengan kompleksitas siklomatis, metode ini akan dilakukan secara manual dan otomatis untuk memvalidasi hasil dari berbagai skenario. Hasil penelitian kami menunjukkan bahwa metode kami dapat menghasilkan cakupan skenario yang sama 100% dengan metode validasi manual.

Kata Kunci : Kasus uji, *Depth-First Search*, Diagram UML, Diagram aktivitas, Pengujian otomatis.