

Abstrak

Pembangunan perangkat lunak sekarang ini lebih banyak dilakukan secara dengan pembangunan berbasis objek. Dalam perkembangannya, selain untuk permodelan UML yang menggambarkan perilaku sistem juga dapat digunakan untuk pengujian atau biasa disebut *model-based testing*. Sebelum melakukan pengujian, perlu diketahui bagaimana *test case* pengujiannya. Dalam *model-based testing*, digunakannya UML adalah pada tahap ini, yaitu tahap pembangkitan *test case*. Karena menggunakan UML model, keuntungan testing ini adalah dapat membangkitkan *test case* walaupun pembangunan perangkat lunak masih ada dalam tahap desain.

Dalam membangkitkan *test case* digunakan *activity diagram*. *Activity diagram* adalah representasi grafis dari seluruh tahapan alur kerja. Keunggulan dipilihnya *activity diagram* adalah diagram ini menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Kelebihan tersebut tergambar dalam dengan notasi dasar *activity diagram*. Selain ada notasi dasar, *activity diagram* juga mempunyai notasi-notasi tambahan untuk memperkaya informasi dirinya, misalkan notasi objek. Notasi objek menggambarkan alur objek dalam *activity diagram*. Notasi objek jarang dipergunakan oleh *developer* karena objek sudah tergambar dalam *sequence diagram*. Karena notasi tambahan dalam penelitian ini diperlukan, maka penyempurnaan dengan notasi tambahan dilakukan dalam *pre-processing*.

Pada tugas akhir ini dihasilkan tools *test case generator*, sehingga terbukti bahwa *model-based testing* menghilangkan beberapa proses pembangkitan *test case* pada *traditional testing* yang berarti waktu pembangkitan *test case* lebih cepat, tetapi harus dipastikan *activity diagram* sudah diverifikasi. *Test case* yang dihasilkan telah diuji kesesuaiannya equivalence partitioning menghasilkan kesesuaian selalu benar jika membandingkan *expected result* dan *actual result* pada *activity diagram valid* dan selalu salah pada *activity diagram invalid*. Dari *Activity diagram Bandung Pintar* dan *ATM Simulation* menghasilkan 7 *test suite* yang sudah sesuai jika dijalankan hasilnya dalam *system under test*, yang berarti tidak ditemukan *defect* sehingga tidak ada yang perlu dibenahi.

Kata Kunci: *Model-based testing, UML, activity diagram, test case.*