

Vierdy Sulfianto Rahmadani, 2015. *Penerapan Reverse Engineering Di Dalam Penentuan Pola Interaksi Sequence Diagram Pada Perancangan Sampel Perangkat Lunak Sistem Operasi Android*. Skripsi ini di bawah bimbingan Indra Kharisma Raharjana, S.Kom, M.T. dan Taufik, S.T, M.Kom, Program Studi S1 Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Airlangga.

ABSTRAK

Dalam penelitian ini, dilakukan penerapan *reverse engineering* di dalam penentuan pola interaksi *Sequence Diagram* pada perancangan sampel perangkat sistem operasi Android. Dalam proses perancangan sebuah perangkat lunak diperlukan konsistensi pada setiap tahapan perancangannya, baik ketika perancangan desain UML hingga penerapannya pada penulisan baris kode perangkat lunak, agar dalam pengembangan perangkat lunak selanjutnya dapat dilaksanakan sesuai dengan desain UML perangkat lunak. Tujuan penelitian ini adalah untuk menerapkan proses *reverse engineering* dari sebuah perangkat lunak, sehingga bisa menghasilkan pola interaksi desain UML *sequence diagram* sebagai dasar desain arsitektur perancangan perangkat lunak.

Sampel aplikasi diperoleh dari sampel yang terdapat pada *Integrated Development Environment (IDE) Eclipse*. Penerapan proses *reverse engineering* dilakukan dengan menggunakan bantuan *plug-in ModelGoonUML4Java* pada IDE Eclipse. Penerapan *reverse engineering* dilakukan melalui beberapa tahapan, tahap pertama adalah pengumpulan sampel pada IDE Eclipse. Tahap kedua adalah pengujian *run* pada setiap sampel yang terkumpul, kriteria uji *run* adalah ketika perangkat lunak bisa berjalan tanpa ada *error* dalam *simulator* atau *device*. Tahap ketiga adalah melakukan identifikasi fitur dari setiap sampel yang lolos uji *run*. Tahap keempat adalah melakukan *reverse engineering* untuk mendapatkan desain *sequence diagram* sesuai dengan fitur yang terkumpul pada tahap sebelumnya. Tahap kelima adalah menjabarkan algoritma setiap fitur. Tahap keenam adalah melakukan identifikasi pola interaksi pada *sequence diagram*. Tahap terakhir adalah melakukan analisis terhadap pola interaksi yang didapat. Kemudian dilakukan evaluasi pada pola-pola interaksi yang didapat dengan merancang suatu contoh perangkat lunak dengan menggunakan pola interaksi yang sama.

Berdasarkan hasil evaluasi, dapat disimpulkan bahwa pola interaksi pada *sequence diagram* yang didapatkan dari proses *reverse engineering* dapat diimplementasikan pada perancangan perangkat lunak yang memiliki fitur-fitur yang sama dengan pola-pola interaksi pada fitur-fitur yang terdapat pada penelitian ini.

Kata kunci : *reverse engineering*, pola interaksi *sequence diagram*

Vierdy Sulfianto Rahmadani, 2015. *Penerapan Reverse Engineering Di Dalam Penentuan Pola Interaksi Sequence Diagram Pada Perancangan Sampel Perangkat Lunak Sistem Operasi Android*. Skripsi ini di bawah bimbingan Indra Kharisma Raharjana, S.Kom, M.T. dan Taufik, S.T, M.Kom, Program Studi S1 Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Airlangga.

ABSTRACT

In this research, reverse engineering executed to determine interaction patterns inside sequence diagram toward development of Android Operating System software samples. In software development process, a consistency is required on each step of its development, both in term of its UML designs and its logical code, so that future developments is able to be applied in accordance with its UML designs. This research aimed to execute reverse engineering from a software, so that it is able to produce interaction patterns of UML sequence diagram designs as a base toward software development design architecture.

Samples were obtained from local sample library contained in the Eclipse Integrated Development Environment (IDE). Execution of the reverse engineering process was assisted by ModelGoonUML4Java plug-in for Eclipse IDE. Reverse engineering executed into several steps. The first step was collecting the sample from Eclipse IDE. The second step was a run-test on each collected sample and run-test criteria is when the software succeeds to run without any error in the simulator or device. The third step was identify features from each sample that passed run-test. The fourth step was execute reverse engineering according to the features collected in the previous step. The fifth step was identify algorithm of each feature. The sixth step was identify interaction patterns on each sequence diagram. The last step was conduct an analysis of obtained sequence diagram interaction patterns. Then the obtained interaction patterns evaluated by designing a software example that uses the same interaction pattern.

According to the evaluation results, it concludes that interaction patterns inside sequence diagram designs obtained in reverse engineering steps, is able to be implemented in software development that contained the similar features with the obtained interaction patterns of features in this research.

Keyword : *reverse engineering, sequence diagram interaction pattern*