

Dalam sebuah *software development lifecycle* pada pengembangan aplikasi *mobile*, tahap *maintenance* merupakan tahap yang membutuhkan biaya dan waktu yang sangat besar. Hal ini ditunjukkan dengan banyaknya aplikasi yang harus dihapus oleh Google pada toko aplikasi mereka karena tidak di-*maintain* dengan baik. Hal ini menunjukkan bahwa masih banyak aplikasi yang tidak dipersiapkan dengan memikirkan aspek *maintainability*. Penelitian sebelumnya menunjukkan *maintainability* pada sistem berorientasi objek seperti aplikasi *mobile* dapat ditingkatkan dengan penggunaan *design pattern* yang sesuai. Sayangnya penelitian tersebut tidak memaparkan secara detail analisis terhadap pemilihan masing-masing *design pattern* yang digunakan serta pengaruh pemilihan tersebut terhadap aspek-aspek *maintainability*. Penelitian ini berfokus pada pemilihan *design pattern* yang dilakukan pada sebuah aplikasi *mobile* dan melihat pengaruhnya terhadap *maintainability* seperti *analyzability*, *changeability*, *stability*, serta *testability*. *Source code* aplikasi dianalisis untuk mengetahui *design problem* yang ada serta pengaruhnya terhadap *OO metric*, lalu mencari *design pattern* yang sesuai untuk melihat pengaruh dari penggunaan beberapa *design pattern* secara terpisah, serta kombinasi dari *design pattern* tersebut terhadap aspek *maintainability*. Dari analisis tersebut ditemukan 2 hal. Pertama pemilihan *design pattern* pada aplikasi *mobile*, khususnya Android dapat dilakukan dengan mencari *design problem* dengan penyesuaian seperti merubah *constructor* menjadi *method-method* penunjang *activity lifecycle*, serta tidak semua *design problem* dapat diselesaikan seperti jumlah *parent class* dari *Activity* yang pada dasarnya melebihi batas yang diterima. Kedua implementasi *design pattern* dapat memperbaiki nilai *maintainability* aplikasi *mobile* yang dipilih antara 11.35% sampai 17.83% karena dapat menerapkan *separation of concern*. Walau begitu, aspek *stability* dapat memburuk karena perlu ada penambahan *class* yang dilakukan sehingga meningkatkan *dependency*.

Kata kunci : design pattern, pengembangan aplikasi mobile, maintainability, non-functional requirement, rekayasa perangkat lunak