

Abstrak

Arsitektur monolitik mendesain sistem aplikasi hanya terdiri dari satu bagian besar dan berjalan pada satu *computational instance* [1]. Hal tersebut menyebabkan, proses meningkatkan atau memperbaiki sistem aplikasi harus secara menyeluruh sekecil apapun peningkatan yang dilakukan. Heartenly.com sebagai aplikasi pencari jodoh asal Indonesia, masih menggunakan arsitektur monolitik. Heartenly.com, membutuhkan arsitektur yang dapat melakukan peningkatan atau perbaikan sistem dapat dilakukan secara independen tanpa harus mematikan seluruh layanan. Untuk itu Heartenly.com perlu melakukan perubahan pada arsitekturnya, dari monolitik menjadi *microservice*. Proses perubahan tersebut melibatkan proses dekomposisi untuk memecah layanan-layanan yang ada pada Heartenly.com. Metode dekomposisi menggunakan prinsip *Domain-driven Design* (DDD) dipadukan dengan arsitektur *Representational State Transfer* (REST) untuk mengoptimalkan jalur komunikasi karena Heartenly.com berbasis *web*. Hasil dekomposisi diuji dengan metode *Single Responsibility Principle* (SRP) dan *Common Closure Principle* (CCP) adaptasi dari prinsip *Object Oriented Programming* (OOP). Dalam penelitian ini, hasil analisis membuktikan bahwa, *microservice* dapat melakukan peningkatan layanan dan *deployment* secara independen

Kata kunci : *Microservice*,dekomposisi,domain-driven design, REST