**Ready for Changes with Hexagonal Architecture**

**(**[**https://netflixtechblog.com/ready-for-changes-with-hexagonal-architecture-b315ec967749?source=false---------0**](https://netflixtechblog.com/ready-for-changes-with-hexagonal-architecture-b315ec967749?source=false---------0)**)**

**Latar Belakang**

Seiring dengan meningkatnya angka produksi konten Netflix Originals setiap tahunnya, maka kebutuhan akan konsep arsitektur yang efisien semakin mendesak. Saat ini terdapat beberapa aplikasi terpisah yang masing-masing menangani unit kerja tertentu, seperti *screenplay*, *playback*, *scripting*, *vendor management*, dan sebagainya. Namun seiring dengan beban kerja yang meningkat, maka dibutuhkan sebuah arsitektur yang berkonsep *highly integrated*, yang mampu menyatukan semua aspek bisnis dan unit kerja yang sebelumnya mengandalkan aplikasi independen. Dengan demikian, pekerjaan (*task*) yang sebelumnya berjalan pada masing-masing unit kerja, dapat digabungkan dan berjalan dalam sebuah aristektur yang terintegrasi. Selain integrasi antar-komponen yang tinggi, diperlukan sebuah repositori terpusat (*centralized repository*) agar masing-masing komponen unit kerja yang telah terintegrasi tersebut dapat mengakses sumber daya (*resource*) yang dibutuhkan. Walaupun konsep tersebut lebih dikenal dengan *shared data*, pengembang Netflix memilih untuk menamakan konsep tersebut dengan *swappable data sources*. Disebut demikian karena data awal (*initial data*) yang dibutuhkan oleh arsitektur tersebut harus diambil dari arsitektur yang terdahulu, untuk kemudian ditukar (*swap*) dengan data baru jika tersedia. Konsep tersebut dapat diterapkan tanpa mengubah alur bisnis yang telah ada, apabila diimplementasikan dalam sebuah arsitektur yang dinamakan Hexagonal Architecture. Arsitektur tersebut berfokus mengisolasi fungsi-fungsi utama (*core logic*) dari aplikasi dari pengaruh eksternal, sehingga kanal *input* dan *output* terletak pada bagian ujung terluar dari rancangan arsitektur. Implikasinya adalah proses pertukaran sumber data (*source swapping*) dapat dilakukan tanpa melakukan perubahan perubahan signifikan pada struktur kode program.

**Konsep Hexagonal Architecture**

*Hexagonal Architecture* bertumpu pada tiga komponen utama, yaitu 1) ***Entities****,* 2) ***Repositories*,** dan 3) ***Interactors***. Komponen *entities* objek yang dapat dilakukan operasi CRUD (*create, read, update, delete*). Data yang dipertukarkan oleh sebuah *entity* disimpan dalam komponen kedua, yaitu *repository*. Komponen tersebut menyimpan daftar *method* yang digunakan untuk berkomunikasi dengan sumber data. Sedangkan komponen ketiga, *interactors*, berfungsi untuk menjalankan fungsi dari sebuah *entity*. Ketiga komponen utama tersebut membentuk lapisan terdalam dari arsitektur heksagonal. Diluarnya terdapat lapisan *data sources* dan *transport layer*. Lapisan *data sources* berhubungan langsung dengan komponen *repository* untuk menyediakan data yang dibutuhkan oleh sebuah *entity*. Sedangkan *transport layer* berhubungan langsung dengan *interactors* sebagai *trigger* untuk menjalankan fungsi sebuah *entity*.

**Quality Attribute**

Implementasi arsitektur heksagonal berusaha menjawab tantangan *quality attributes* berupa ***performance****,* ***modifiability****,* dan ***availability***. Atribut *performance* membahas mengenai kemampuan sebuah sistem untuk memenuhi kebutuhan pengguna dalam suatu jangka waktu yang telah ditentukan. Agar dapat memenuhi batasan waktu tersebut, tentu diperlukan sebuah arsitektur yang efisien, dimana pertukaran data antar-komponen dapat berlangsung dengan efektif dan tanpa kendala yang berarti. Arsitektur heksagonal menjawab tantangan tersebut dengan memaksimalkan integrasi antar-komponen, sehingga sistem dapat melakukan sebuah pekerjaan dengan waktu tempuh yang singkat dan sumber daya yang minimal. Atribut kedua yang dibahas adalah *modifiability*, yang membahas mengenai biaya dan resiko perubahan yang dilakukan terhadap suatu sistem. Pada bagian awal artikel tersebut membahas bagaimana sebuah sistem yang telah ada (*existing*) dapat dimodifikasi untuk mendukung konsep arsitektur heksagonal tanpa melakukan perubahan-perubahan yang signifikan terhadap struktur kode program. Dengan demikian, arsitektur heksagonal memiliki tingkat *modifiability* yang tinggi dan resiko perubahan yang rendah. Atribut ketiga adalah *availability*, yang membahas mengenai tingkat ketersediaan sebuah sistem untuk melakukan pekerjaan ketika dibutuhkan. Aristektur heksagonal mendukung atribut *availability*, dengan membagi sistem menjadi banyak *entity* yang memiliki sifat *loose coupling*. Apabila terjadi *fault* pada sebuah *entity*, maka *entity* lainnya tidak akan terdampak secara signifikan dan dapat menjalankan fungsinya dengan normal. Dengan demikian, sistem dengan arsitektur heksagonal masih dapat melayani kebutuhan pengguna walau dengan kapasitas layanan yang tidak maksimal.

**Kesimpulan**

Hexagonal Architecture adalah sebuah arsitektur piranti lunak yang berkonsep *highly integrated* dan *swappable data sources*. Arsitektur tersebut bertumpu pada tiga komponen utama, yaitu 1) *Entities,* 2) *Repositories*, dan 3) *Interactors*. Ketiga komponen tersebut didukung oleh *data sources* dan *transport layer*. Dari segi *quality attributes*, arsitektur heksagonal mendukung atribut *performance* karena mengusung konsep *highly integrated*, *modifiability* karena kemudahan untuk memodifikasi sistem untuk mendukung arsitektur heksagonal, dan *availability*, karena sistem terbagi menjadi banyak komponen yang memiliki fungsinya tersendiri dan tidak saling berhubungan (*loose coupling*).