

# Strategi Migrasi

Setelah kita belajar beberapa kerangka kerja yang dapat membantu Anda dalam mempersiapkan proses migrasi ke AWS, sekarang mari kita lihat bagaimana cara implementasinya.

Ingatkah Anda? Proses migrasi tak terjadi hanya dengan menjentikkan jari lalu seketika semua elemen dari data center on-premise Anda berjalan di AWS.

Setiap aplikasi yang berarsitektur *tightly coupled*--telah kita bahas di Modul Komputasi di Cloud--akan memiliki 6 kemungkinan opsi terkait strategi migrasi, AWS menyebutnya dengan 6 R.

Saat Anda sudah yakin untuk bermigrasi ke AWS, pilihlah opsi strategi migrasi di antara 6 R yang paling sesuai menurut waktu, biaya, prioritas, dan kekritisannya beban kerja Anda. Mari kita jabarkan 6 R satu per satu:

- Rehosting

Rehosting juga dikenal sebagai *lift and shift* (angkat dan pindahkan). Opsi ini mudah dilakukan karena Anda tak perlu membuat perubahan apa pun. Cukup pindahkan aplikasi yang Anda miliki ke AWS.

Dengan rehosting, Anda bisa menghemat hingga 30% dari total biaya walaupun tidak ada pengoptimalan apa pun pada aplikasi Anda.

Aplikasi akan lebih mudah untuk dioptimalkan setelah ia berjalan di cloud. Ini karena kemampuan Anda dalam menggunakan AWS akan semakin berkembang seiring proses migrasi cloud dan bagian yang sulit--migrasi aplikasi, data, dan traffic--telah selesai sehingga Anda bisa fokus untuk mengoptimalkan aplikasi.

- Replatforming

Strategi migrasi yang satu ini disebut juga dengan *lift, tinker, and shift* (angkat, perbaiki, dan pindahkan). Pada dasarnya, opsi ini masih berupa lift and shift namun Anda dapat melakukan beberapa pengoptimalan cloud. Dalam strategi ini, Anda tak akan mengubah arsitektur inti (*core architecture*) apa pun pada proses migrasi aplikasi.

Sebagai contoh, Anda dapat mengurangi jumlah waktu yang dihabiskan untuk mengelola database instance dengan bermigrasi ke platform *database-as-a-service* seperti Amazon Relational Database Service (Amazon RDS). Atau, Anda dapat memigrasikan aplikasi Anda ke platform yang dikelola sepenuhnya seperti AWS Elastic Beanstalk.

- Retiring

AWS menemukan bahwa sebanyak 10% hingga 20% dari portofolio IT perusahaan menyertakan aplikasi yang tak lagi digunakan dan bisa dimatikan. Maka dari itu, hapuslah aplikasi yang tidak lagi Anda butuhkan.

Penghematan semacam ini dapat memberikan beberapa keuntungan, seperti meningkatkan *business case* (kasus bisnis), memfokuskan tim terhadap aplikasi yang digunakan, dan mengurangi jumlah aplikasi yang

harus dikelola.

- Retaining

Saat hendak melakukan migrasi ke AWS, mungkin Anda memiliki beberapa aplikasi yang akan segera deprecated (dihentikan). Tetapi, aplikasi tersebut masih perlu digunakan selama beberapa waktu.

Nah, daripada memindahkan aplikasi tersebut ke AWS, sebaiknya Anda hanya memigrasikan aplikasi yang sekiranya masuk akal dan berguna untuk bisnis.

- Repurchasing

Strategi migrasi ini umum terjadi pada perusahaan yang ingin meninggalkan vendor perangkat lunak lama dan memulai yang baru. Misalnya:

- Mengakhiri kontrak dengan vendor CRM (*Customer Relationship Management*) lama dan pindah ke yang baru seperti Salesforce.com.
- Beralih dari sistem HR (Human Resource/Sumber Daya Manusia) ke aplikasi Workday.
- Berpindah dari perangkat lunak CMS (Content Management System) ke software Drupal.
- Mengakhiri lisensi dengan vendor database yang usang dan migrasi ke database cloud.

Oke, mungkin ini terdengar bagus. Tapi ingat! Anda akan berurusan dengan *software package* (paket perangkat lunak) yang baru. Beberapa di antaranya mudah diimplementasikan tetapi ada juga yang membutuhkan waktu.

Oleh karena itu, total biaya di muka untuk strategi ini akan naik, tetapi potensi manfaatnya bisa sangat besar.

- Refactoring/re-architecting

Pada opsi inilah Anda menulis kode yang baru. Hal ini didorong oleh kebutuhan bisnis yang kuat untuk menambahkan fitur, skala, atau kinerja yang mungkin sulit diraih pada data center on-premise.

Perubahan dramatis pada arsitektur Anda bisa sangat bermanfaat. Namun, akan menimbulkan biaya awal yang paling tinggi dalam hal perencanaan dan usaha migrasi.

Oke. Itulah 6 strategi migrasi yang dapat Anda terapkan untuk migrasi ke AWS. Pilihlah opsi yang tepat demi suksesnya proses migrasi Anda.

---

[< Sebelumnya](#)[Selanjutnya >](#)