

PPL 2017

PANDUAN

ARSITEKTUR & DEPLOYMENT

30 JANUARY 2017

ARSITEKTUR SISTEM

ENVIRONMENT SISTEM YANG ADAPTIF

Arsitektur sistem sangat mempengaruhi kualitas dari produk, mulai saat pengembangan, testing, deployment, hingga *delivery* produk. Strategi keterhubungan antara aplikasi dengan komponen pendukungnya sangat mempengaruhi kemampuan sistem dalam menyesuaikan kebutuhan atas *enhancement* di masa yang akan datang. Sangatlah penting untuk memahami apa yang terjadi pada tingkat platform & komponen pendukung sebelum perancangan system dimulai.

1) *Staging, SIT & UAT environment*

Lingkungan pengembangan dan pengujian harus dibuat agar sistem dapat berjalan dalam skala yang dibutuhkan namun tidak merusak tujuan dari setiap lingkungan yang dibuat. Keberadaan environment Staging, SIT, dan UAT yang lengkap akan memudahkan fase-fase pengembangan dapat berjalan dengan baik.

2) Arsitektur system sudah harus mewakili kebutuhan non-fungsional

Kebutuhan non-fungsional sistem sangat mempengaruhi pemilihan dan susunan komponen-komponen yang diperlukan.

3) Perhatikan ketersediaan lisensi environment Development dan Deployment

Apabila ada, pastikan bahwa Anda diperbolehkan untuk mempergunakan komponen sesuai lisensi yang berlaku, baik tools untuk pengembangan, library, hingga tema. Posisikan diri Anda sebagai pengembang dengan status komersial (bukan mengandalkan lisensi mahasiswa).

4) Strategi arsitektur untuk performance dan kapasitas

Terkait dengan poin no 2, implementasi kebutuhan tersebut tidak hanya dilakukan pada aplikasi/program yang dibuat saja, melainkan harus meliputi seluruh komponen yang terlibat. Perlu diperhatikan bahwa sistem secara keseluruhan harus disusun untuk siap melayani dengan performance dan kapasitas yang diharapkan.

PRODUCT DEPLOYMENT

DELIVERY PRODUK YANG BERKUALITAS

1. **Automated Deployment Pipeline/Scripting**



Tujuan dari *deployment pipeline* ini adalah untuk menempatkan setiap perubahan yang Anda buat harus melalui proses automated build, deploy, dan test system (bukan cuma unit test) untuk memverifikasi kesiapannya sebelum rilis. Pastikan Anda memiliki mekanisme automated deployment / compilation / testing pada rangkaian environment Anda.

2. **Release Notes.**

Aplikasi tidak jadi hanya dalam 1 (satu) tahapan pengembangan. Apabila dalam pengembangan system ini terjadi 3 kali rilis, maka setiap rilis harus memiliki dokumen *summary* atas hal-hal yang baru, diganti, hingga hal-hal yang *obsolete*.

3. Product Packaging & **Installer.**

Setiap aplikasi yang sudah jadi harus memiliki panduan instalasi, dokumen cara instalasi, dan berkas yang diperlukan. Pastikan agar aplikasi yang Anda buat dapat diinstall dengan baik oleh orang lain yang sama sekali tidak mengerti bagaimana system ini dibuat. Panduan instalasi yang lengkap mulai dari *basic requirement*, FAQ, hingga Troubleshooting harus terbungkus rapih di dalam paket instalasi ini.

4. **Skenario Distribusi**

Apabila sebuah berkas installer sudah dihasilkan bukan berarti seluruh proses sudah selesai. Harus dipastikan bahwa penyebaran paket program dan instalasi

dapat dijangkau, diamankan, didapatkan, dan dipastikan berfungsi dengan baik bagi pengguna. Seluruh hal yang dapat terjadi saat pendistribusian aplikasi harus dapat diprediksi dan direncanakan dengan baik.