

Studi Kasus: Sistem Notifikasi dan Log Aktivitas Fleksibel

Sebuah platform media sosial baru sedang berkembang pesat. Dengan pertumbuhan pengguna, kebutuhan akan sistem notifikasi yang dinamis dan pencatatan log aktivitas yang komprehensif menjadi sangat penting. Namun, arsitektur yang ada saat ini menghadapi beberapa tantangan.

Masalah:

1. **Beragamnya Jenis Notifikasi:** Platform ini perlu mengirimkan berbagai jenis notifikasi kepada pengguna (misalnya, pesan baru, permintaan pertemanan, komentar pada postingan, *like*, *update* sistem, pengumuman promosi). Setiap jenis notifikasi mungkin memiliki detail konten yang berbeda dan mungkin perlu dikirim melalui saluran yang berbeda (email, SMS, *push notification* dalam aplikasi). Menambah jenis notifikasi baru atau saluran pengiriman baru memerlukan modifikasi pada banyak bagian kode notifikasi inti.
2. **Konfigurasi Log yang Tidak Fleksibel:** Sistem perlu mencatat berbagai aktivitas pengguna dan sistem (misalnya, *login*, *logout*, pembuatan postingan, pengiriman pesan, perubahan profil, kesalahan sistem). Tingkat detail log yang diperlukan bisa bervariasi, dan kadang-kadang log perlu ditambahkan informasi kontekstual tambahan (misalnya, IP address, user agent) secara dinamis tanpa mengubah struktur dasar objek log.
3. **Proses Validasi Data yang Berurutan:** Sebelum suatu aktivitas dicatat atau suatu notifikasi dikirim, seringkali ada serangkaian validasi yang perlu dilakukan (misalnya, apakah pengguna aktif, apakah format data benar, apakah ada batasan pengiriman). Urutan dan jenis validasi ini bisa bervariasi tergantung pada aktivitas atau notifikasinya. Saat ini, logika validasi tersebar dan sulit untuk menambahkan atau mengubah urutan validasi.
4. **Akses Tunggal ke Pengaturan Global:** Ada beberapa pengaturan sistem penting yang harus diakses secara global dan konsisten di seluruh aplikasi (misalnya, kunci API untuk layanan SMS, *threshold* untuk notifikasi *spam*, *flag debug*). Saat ini, objek pengaturan ini bisa dibuat berkali-kali di berbagai tempat, berpotensi menyebabkan ketidakkonsistenan dan kesulitan dalam pembaruan.

Platform ini membutuhkan solusi yang memungkinkan penambahan jenis notifikasi dan saluran pengiriman baru dengan mudah, mekanisme yang fleksibel untuk mencatat log dengan detail yang bervariasi, cara terstruktur untuk menerapkan validasi berurutan, dan manajemen pengaturan sistem yang konsisten secara global.

Soal 1 (35 point): Jelaskan design pattern apa saja yg digunakan & apa alasannya?

Soal 2 (65 point): Buat solusinya, beserta kelas test-nya (TestMedsos.java/.py).