

## **SOAL REMEDIAL UCP 2**

Sebuah Universitas terkemuka sedang melakukan perombakan total pada Sistem Manajemen Data Buku mereka untuk mengatasi masalah redundansi dan ketidakefisiensi pelacakan aset. Tantangan utama terletak pada pengelolaan puluhan ribu buku di mana setiap satu judul buku memiliki banyak bentuk fisik dengan ID unik yang harus dipantau ketersediaannya secara digital. Struktur organisasi buku menjadi sangat kompleks karena satu buku dapat ditulis oleh banyak pengarang (harus memiliki relasi disalah satu RAT), serta kategori buku yang kini dirancang secara hierarkis dengan sub-kategori yang memiliki kedalaman tak terbatas. Pengembang dituntut untuk menciptakan skema database yang mampu melakukan pencarian agregat secara rekursif, di mana pencarian pada satu kategori induk harus secara otomatis menampilkan seluruh buku yang berada di bawah sub-kategori turunannya tanpa mengorbankan performa aplikasi.

Dari sisi keamanan dan keandalan sistem, aplikasi Android berbasis Kotlin ini wajib menerapkan protokol validasi yang ketat. Setiap data yang dimasukkan oleh pustakawan harus divalidasi berdasarkan tipe datanya untuk mencegah kesalahan entri. Selain itu, sistem harus mampu mendeksi dan mencegah cyclic reference pada struktur kategori agar tidak terjadi perulangan tak terbatas yang dapat merusak logika pencarian. Integritas data menjadi prioritas melalui penerapan Audit Log yang menyimpan rekaman data sebelum dan sesudah perubahan, serta mekanisme Soft Delete untuk memastikan data tidak hilang permanen sebelum periode selesai.

Tantangan krusial muncul pada logika penghapusan data yang saling terkait. Jika seorang pustakawan mencoba menghapus sebuah kategori yang masih memiliki buku dengan status "dipinjam", sistem harus mampu menjalankan mekanisme rollback otomatis melalui Database Transaction untuk membatalkan seluruh operasi demi mencegah ketidakkonsistenan status. Namun, jika buku tersebut tidak sedang dipinjam, sistem harus memberikan opsi dinamis: apakah buku dalam kategori tersebut akan ikut dihapus secara logis atau dialihkan statusnya menjadi "Tanpa Kategori". Seluruh proses ini, termasuk migrasi data besar-besaran dari sistem lama, harus berjalan secara asynchronous dengan tetap menjaga aplikasi responsif menggunakan wadah container untuk membatasi data pada perangkat pengguna.