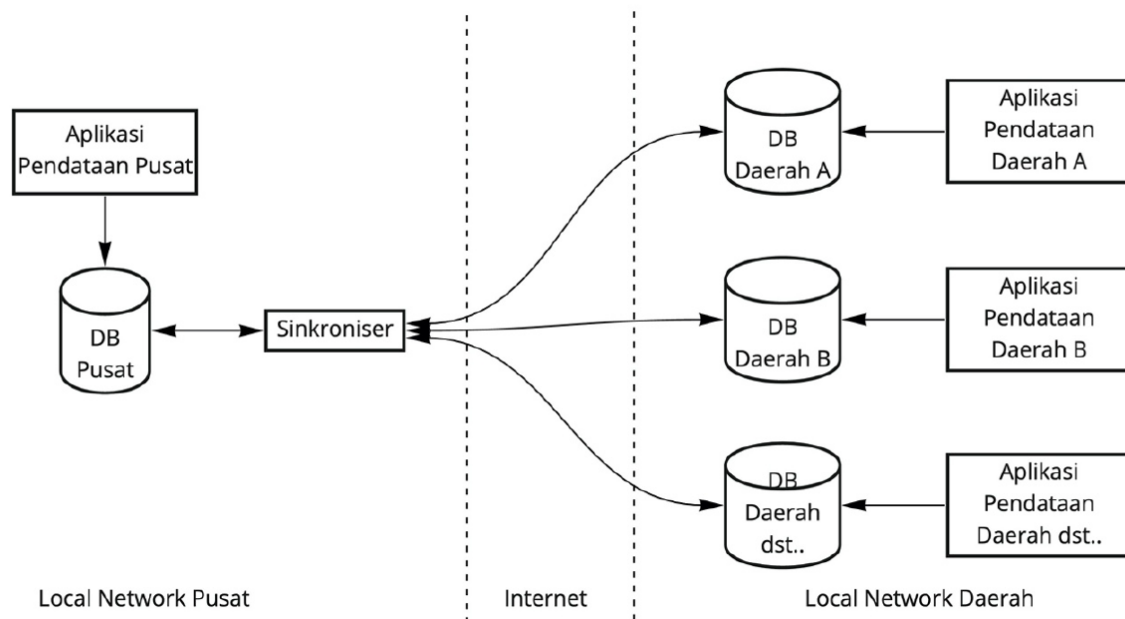


**Evaluasi Akhir Semester Genap 2021/2022**  
**Arsitektur Perangkat Lunak (A)**  
**Dosen: Rizky Januar Akbar, S.Kom., M.Eng**  
**Sifat: Open book, kelompok terdiri dari 3 orang**

Sebuah instansi pemerintahan pusat membawahi 4670 instansi daerah yang tersebar di seluruh wilayah Indonesia. Instansi pusat ini harus melakukan pendataan aktivitas yang terjadi di setiap instansi daerah. Untuk melaksanakan pendataan itu dan juga banyaknya instansi daerah yang ada dan tidak semua memiliki akses internet yang memadai maka dibuatlah arsitektur sistem sebagai berikut.



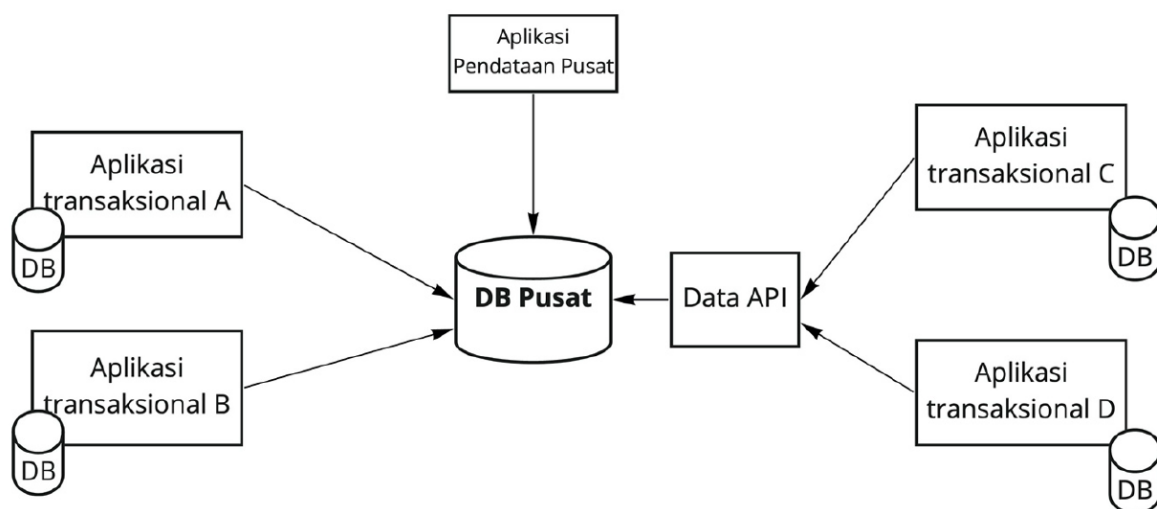
Pada arsitektur di atas, instansi pusat memiliki sebuah database pusat menggunakan RDBMS SQL Server yang diakses oleh aplikasi pendataan pusat. Instansi daerah memiliki database daerah menggunakan RDBMS PostgreSQL yang diakses oleh aplikasi pendataan daerah. Database pusat dan database daerah memiliki struktur database yang identik meski berbeda RDBMS. Database pusat menyimpan seluruh data instansi daerah sedangkan database daerah hanya menyimpan data yang terkait instansinya saja. Aliran data berjalan secara dua arah dimana data seperti data referensi mengalir dari pusat ke daerah, sedangkan data master dan transaksional mengalir dari daerah ke pusat. Untuk menjaga konsistensi data antara pusat dan daerah maka dibuat sebuah perangkat lunak sinkroniser yang berfungsi untuk menyamakan data diantara dua database pusat dan daerah. Sinkroniser melayani sinkronisasi data untuk seluruh instansi yang berjumlah 4670.

Primary use cases dari sistem pendataan daerah adalah sebagai berikut:

Use Case	Description
UC-1: Melakukan pendataan mahasiswa	Petugas pendataan di daerah dapat menginputkan data mahasiswa, mengedit data mahasiswa, atau menghapus data mahasiswa.

UC-2: Melakukan pendataan dosen	Petugas pendataan di daerah dapat menginputkan data dosen beserta penugasannya.
UC-3: Melakukan pendataan perkuliahan	Petugas pendataan di daerah dapat menginputkan data kurikulum, kelas, aktivitas mahasiswa, dan lainnya.
UC-4: Men-generate reports	Petugas pendataan di daerah dapat melihat rekapitulasi data jumlah dosen, jumlah mahasiswa, KRS mahasiswa, dan lainnya.
UC-5: Memonitor status sinkroniser	Petugas pendataan di daerah dapat melihat status sinkroniser DB pusat apakah menyala atau tidak.
UC-6: Melakukan sinkronisasi ke DB pusat	Petugas pendataan di daerah dapat melakukan sinkronisasi
UC-7: Ekspor data	Petugas pendataan di daerah dapat mengekspor data dalam dua format yaitu HTML dan XLS

Setelah sistem berhasil dibangun dan dijalankan, data mengalir dengan baik dari daerah ke pusat, maka tahap selanjutnya dibutuhkan mekanisme untuk pemanfaatan data yang telah masuk ke DB pusat. Untuk memanfaatkan data dan juga memperkaya data, maka dibuat juga aplikasi-aplikasi lain yang akan menggunakan data dari database pusat dan juga mengembalikan data ke database pusat dengan arsitektur sebagai berikut.



Pada arsitektur di atas terdapat dua cara aplikasi dapat mengakses data yaitu cara pertama dengan mengakses database pusat secara langsung dan cara kedua dengan mengakses lewat API. Aplikasi transaksional A dan B melakukan akses data dengan mengakses database secara langsung, sehingga struktur internal database DB pusat mempengaruhi struktur database aplikasi transaksional A dan B. Untuk aplikasi transaksional eksternal lainnya (C dan D) dapat mengakses data melewati Data API. Aplikasi transaksional selain dapat mengakses data dari database pusat, juga dapat mengirimkan/menyimpan data ke database pusat. Selain itu, database pusat menampung data untuk seluruh domain data baik di pusat dan di daerah,

sehingga bisa disebut sebagai all-purpose single model karena menangani banyak domain permasalahan dalam sebuah database.

Seiring dengan berjalannya waktu terdapat kelemahan-kelamahan yang menyebabkan operasional tidak berjalan dengan lancar dan pengembangan tidak dapat berjalan dengan cepat, sebagai contoh beberapa hal sebagai berikut:

1. Aplikasi transaksional yang mengakses DB pusat secara langsung mengakibatkan adanya strong coupling antara DB pusat dengan aplikasi transaksional.
2. Perubahan/alter database pusat tidak dapat dilakukan secara independen karena adanya strong coupling database dengan aplikasi-aplikasi lain, sehingga menghambat pengembangan aplikasi baru/penyesuaian aplikasi untuk mengikuti perubahan proses bisnis.
3. Performa menurun karena banyaknya transaksi dan locking yang terjadi di database pusat.
4. Perubahan yang dilakukan di satu aplikasi transaksional yang menyebabkan perubahan di database pusat akan mengakibatkan ketidakstabilan di sistem-sistem lain yang berhubungan.
5. Database selain menyimpan data terstruktur (structured) juga menyimpan data unstructured seperti file (binary/object) yang menyebabkan ukuran database sangat besar dan jika terjadi database backup atau database failure, membutuhkan waktu yang lama untuk melakukan backup dan recovery.
6. Jika database pusat mati, maka seluruh layanan mati.

Anda saat ini berperan sebagai arsitek dan diminta untuk menyelesaikan permasalahan diatas dan menghasilkan rekomendasi arsitektur baru dan bagaimana evolusinya dari arsitektur yang saat ini dengan menjawab pertanyaan-pertanyaan dibawah ini.

1. Apakah model arsitektur saat ini monolitik atau terdistribusi? Jelaskan dengan rinci.
2. Identifikasi kelebihan dan kekurangan arsitektur saat ini!
3. Bagaimana model arsitektur yang akan Anda rekomendasikan?
4. Jelaskan dengan rinci arsitektur yang akan Anda rekomendasikan beserta penjelasan alasan keputusan arsitekturalnya!
5. Jelaskan bagaimana arsitektur baru yang Anda rekomendasikan dapat menyelesaikan masalah yang timbul!
6. Identifikasi kelebihan dan kelebihan arsitektur yang Anda rekomendasikan!
7. Bagaimana langkah-langkah evolusi yang harus diambil untuk berpindah dari arsitektur lama ke arsitektur baru?

Jawablah pertanyaan-pertanyaan diatas dengan menggunakan cara yang dijelaskan pada Case Study di Buku Designing Software Architectures: A Practical Approach. Humberto Cervantes, Rick Kazman. Bab 4-6.

Definisikan bagian-bagian berikut:

- a. Use Case Model (bisa menggunakan primary use cases yang telah didefinisikan)
- b. Quality Attribute Scenarios
- c. Constraints
- d. Architectural Concerns
- e. Attribute-driven design steps (antara 2-4 iterasi)

Tuliskan jawaban dan penjelasannya di sebuah dokumen berformat PDF dengan identitas nama dan NRP dengan format sebagai berikut:

**EAS\_APL\_2022\_NRP.pdf**

Contoh:

**EAS\_APL\_2022\_0511100000.pdf**

Dokumen paling lambat dikirimkan pada tanggal 22 Juni 2022 Pk. 23:00 WIB di myITS Classroom.

**--Selesai--**