BAB V

SIMPULAN DAN SARAN

5.1 Simpulan

Berdasarkan rangkaian perancangan, implementasi, serta pengujian yang telah dilakukan, dapat disimpulkan bahwa pengembangan modul Business Case (*Bizcase*) pada sistem PRISMA berhasil dilaksanakan secara sistematis dengan menggunakan pendekatan *Database System Development Life Cycle* (DBSDLC). Seluruh tahapan mulai dari analisis kebutuhan, perancangan basis data, implementasi backend, hingga pengujian menyeluruh menghasilkan modul yang stabil, terintegrasi, dan memenuhi kebutuhan operasional PMO IT. Simpulan penelitian ini dijabarkan berdasarkan rumusan masalah sebagai berikut:

1. **Menjawab Rumusan Masalah Pertama: Rancangan Basis Data yang Terstruktur dan Konsisten**

   Penelitian ini berhasil menghasilkan rancangan basis data relasional yang terstruktur, bebas redudansi, dan memenuhi prinsip *Third Normal Form* (3NF). Seluruh entitas turunan *bc_* dirancang dalam pola *one-to-many* terhadap *Bizcase*, sehingga integritas data dapat dipertahankan dan tidak ditemukan lagi kasus *orphan record* seperti yang terjadi pada sistem sebelumnya. Implementasi dengan PostgreSQL dan Prisma ORM memastikan struktur tabel terdokumentasi dan mudah dikembangkan.

2. **Menjawab Rumusan Masalah Kedua: Implementasi Backend yang Modular dan Andal**

   Layanan *backend* yang dikembangkan menggunakan NestJS berhasil memisahkan logika bisnis secara modular, mendukung validasi input yang ketat, serta menyediakan mekanisme transaksi yang kuat. Fungsi krusial updateProjectV3 berhasil diimplementasikan menggunakan pendekatan upsert dan **Transaksi Atomik**. Pendekatan ini terbukti menghilangkan risiko inkonsistensi data yang sebelumnya timbul akibat penggunaan pola *delete-and-recreate* pada sistem lama, sehingga

memberikan fondasi sistem yang lebih aman dan mudah dipelihara (*maintainable*).

3. **Menjawab Rumusan Masalah Ketiga: Efektivitas dan Performa Sistem**

   Hasil pengujian fungsional dan kinerja menunjukkan bahwa sistem mampu menangani *payload* data kompleks dengan ukuran lebih dari 150 KB tanpa *error*. Pengujian kinerja mencatat waktu respons rata-rata 1,67 detik untuk proses pengambilan data (*Read*) dan 9,75 detik untuk proses pembaruan data kompleks (*Write*) pada lingkungan pengembangan. Meskipun waktu pembaruan dipengaruhi oleh keterbatasan sumber daya server pengembangan, mekanisme transaksi atomik terbukti berhasil menjamin data tersimpan secara konsisten 100%. Dengan demikian, implementasi modul ini terbukti meningkatkan akurasi data dan efisiensi proses pelaporan proyek secara signifikan dibandingkan metode sebelumnya.

Secara umum, penelitian ini berhasil menghasilkan rancangan dan implementasi *backend Bizcase* yang akurat, efisien, terintegrasi, dan siap digunakan sebagai dasar pengembangan fungsionalitas lanjutan dalam sistem PRISMA.

5.2 Saran

Sebagai tindak lanjut dari pengembangan modul yang telah dilakukan, terdapat beberapa saran yang dapat dijadikan acuan untuk pengembangan sistem PRISMA pada tahap berikutnya:

1. **Pengujian Lanjutan pada Staging atau Production Environment**

   Hasil pengujian saat ini diperoleh pada lingkungan pengembangan. Pengujian tambahan pada *staging* atau *production environment* diperlukan untuk memperoleh metrik performa aktual, terutama ketika sistem menangani permintaan simultan dari banyak pengguna.

2. **Optimasi Performa dan Skalabilitas Sistem**

Untuk menangani waktu respons pada pemrosesan *payload* besar, pengembangan selanjutnya dapat menerapkan strategi *Database Indexing* pada kolom yang sering diakses dan mekanisme *Caching* (menggunakan Redis) untuk data yang jarang berubah. Hal ini diharapkan dapat memangkas waktu eksekusi transaksi secara signifikan.

3. **Peningkatan Keamanan dan Pengendalian Akses**

Implementasi *Role-Based Access Control* (RBAC) yang lebih granular dan fitur *Audit Logging* menjadi penting untuk diterapkan. Hal ini bertujuan agar setiap perubahan pada data *Bizcase* dapat ditelusuri riwayatnya (*traceability*), terutama pada proses yang melibatkan persetujuan anggaran sensitif.

4. **Integrasi Lebih Lanjut dengan Modul Internal PRISMA**

Pengembangan selanjutnya dapat memperluas integrasi *Bizcase* dengan modul *Approval, Project Reporting*, serta modul *Risk Management* agar seluruh siklus proyek dapat tercatat secara otomatis dan saling sinkron tanpa *input* manual.

5. **Automasi Pengujian API dan Validasi Payload**

Penerapan *automated API testing* menggunakan Jest, Newman, atau Cypress dapat mempercepat proses QA dan meminimalkan risiko regresi pada setiap perubahan versi backend, terutama untuk fungsi multi-entitas seperti *updateProjectV3()*.

6. **Pengembangan Validasi Logika pada Sisi *Frontend***

Mengingat penelitian ini berfokus pada sisi *backend* dan basis data, penelitian selanjutnya disarankan untuk mengembangkan logika validasi yang lebih kuat pada antarmuka pengguna (*frontend*). Hal ini diperlukan untuk mencegah terjadinya *human error* yang bersifat semantik, seperti

kesalahan pengetikan (*typo*) pada nama proyek atau input data duplikat yang tidak sepenuhnya dapat dideteksi oleh *constraint* basis data.

Dengan menerapkan saran-saran tersebut, sistem PRISMA dapat berkembang menjadi platform manajemen proyek yang lebih komprehensif, efisien, dan mampu mendukung proses bisnis perusahaan secara berkelanjutan dalam jangka panjang.