

BAB 4. PENUTUP

4.1. Kendala Pengembangan Proyek

Dalam proses pengembangan Sistem Repositori dan Demonstrasi Karya Mahasiswa FIK UPN “Veteran” Jakarta, terdapat beberapa kendala yang dihadapi, yaitu sebagai berikut.

1. Pada awal pengembangan, sistem belum memiliki data akademik utama, seperti data mahasiswa, dosen Fakultas Ilmu Komputer, program studi, dan periode akademik. Data tersebut diperoleh dari UPA Teknologi Informasi dan Komunikasi Universitas Pembangunan Nasional “Veteran” Jakarta, kemudian disesuaikan dengan struktur basis data sistem. Sinkronisasi data belum dapat dilakukan secara otomatis dan terintegrasi karena terdapat kendala konektivitas dan konfigurasi antarmuka pada server. Akibatnya, pembaruan data belum dapat diterima secara langsung dan sebagian data masih perlu diperiksa sebelum digunakan.
2. Pengelolaan peran dan hak akses menjadi lebih kompleks karena sistem digunakan oleh Superadmin, Admin Program Studi, Koordinator Program Studi, Dosen, dan Civitas Akademika dengan kewenangan yang berbeda. Satu pengguna juga dapat memiliki lebih dari satu peran sehingga sistem harus menentukan peran aktif, menampilkan halaman yang sesuai, dan membatasi data berdasarkan program studi. Meskipun mekanisme tersebut telah diterapkan, jenis peran dan kewenangannya masih ditentukan dalam kode program sehingga perubahan struktur organisasi masih memerlukan penyesuaian pada sistem.
3. Kebutuhan sistem mengalami perubahan selama proses pengembangan berdasarkan hasil diskusi, evaluasi internal, dan pelaksanaan *User Acceptance Testing* (UAT). Masukan yang diterima berkaitan dengan data referensi, jenis berkas, statistik, format ekspor, pengaturan demonstrasi, dan penyajian informasi. Perubahan tersebut dapat memengaruhi struktur basis data, proses *backend*, respons API, dan integrasi dengan *frontend* sehingga setiap penyesuaian perlu diikuti dengan pengujian ulang.
4. Pengelolaan kapasitas penyimpanan menjadi perhatian karena jumlah berkas akan terus bertambah seiring dengan meningkatnya jumlah karya mahasiswa dan pengajuan ulang setelah revisi. Setiap versi pengajuan tetap disimpan agar riwayat dokumen dapat ditelusuri. Pada pelaksanaan UAT, pengguna juga menyampaikan kekhawatiran terhadap kapasitas penyimpanan apabila sistem digunakan secara berkelanjutan. Sistem belum menyediakan informasi mengenai kapasitas yang telah digunakan, sisa ruang penyimpanan, dan pemberitahuan ketika kapasitas mendekati batas maksimum.
5. Sistem bergantung pada ketersediaan *backend*, PostgreSQL, Gitea, dan sistem berkas server. Ketika salah satu layanan tidak aktif, proses pengambilan dan

penyimpanan data, akses repositori, serta pengelolaan dokumen tidak dapat dijalankan. Penanganan dilakukan dengan memeriksa konfigurasi dan mengaktifkan kembali layanan yang diperlukan. Namun, sistem belum dilengkapi dengan pemantauan layanan, pemberitahuan otomatis, serta mekanisme pencadangan dan pemulihan yang dikelola secara khusus.

4.2. Kesimpulan

Dalam proyek ini, telah berhasil dikembangkan Sistem Repositori dan Demonstrasi Karya Mahasiswa FIK UPN “Veteran” Jakarta dengan fokus pada pengembangan basis data dan Sistem Manajemen Admin. Sistem dikembangkan menggunakan metode *Extreme Programming* agar dapat menyesuaikan perubahan kebutuhan selama proses pengembangan. Hasil implementasi menunjukkan bahwa:

1. Arsitektur basis data terintegrasi dirancang melalui identifikasi kebutuhan informasi dan pemodelan hubungan antarentitas menggunakan *Entity Relationship Diagram*, kemudian diimplementasikan pada PostgreSQL 15 melalui Prisma ORM. Struktur tersebut menghubungkan metadata akademik, data referensi sistem, pengguna, peran, program studi, tugas akhir, repositori, berkas pendukung, status pengelolaan, dan riwayat aktivitas. Hierarki hak akses dibentuk melalui keterkaitan antara pengguna, peran, dan program studi. PostgreSQL digunakan untuk menyimpan data terstruktur dan metadata, sedangkan kode sumber tetap dikelola melalui Gitea dan dokumen pendukung disimpan pada sistem berkas server. Hasil pengujian basis data menunjukkan bahwa hubungan antardata dan aturan integritas telah berjalan sesuai dengan rancangan.
2. Logika *backend* Sistem Manajemen Admin diimplementasikan menggunakan Node.js, Express.js, Prisma ORM, dan PostgreSQL. Setiap permintaan diproses berdasarkan peran aktif dan ruang lingkup kewenangan pengguna. Admin Program Studi hanya menangani data pada program studinya, sedangkan Superadmin dapat melakukan pengelolaan pada tingkat fakultas. Mekanisme tersebut mendukung proses pendataan mahasiswa, pengelolaan pengguna dan peran, verifikasi repositori, permintaan revisi, perubahan status karya, pengelolaan data referensi, penyajian statistik, serta ekspor data. Hasil *unit testing* dan *User Acceptance Testing* menunjukkan bahwa fungsi administratif dan pembatasan akses telah berjalan sesuai dengan kebutuhan pengguna.
3. Mekanisme transformasi data repositori dibangun dengan menghubungkan informasi teknis dari Gitea dengan metadata akademik yang tersimpan dalam basis data. Proses tersebut menghasilkan informasi karya mahasiswa yang dilengkapi dengan identitas pemilik, program studi, periode akademik, kategori, metode, dosen terkait, status pengelolaan, dokumen pendukung, dan informasi demonstrasi. Informasi yang dihasilkan dapat ditelusuri, disaring, dikelompokkan, diagregasi, disajikan dalam bentuk statistik, dan diekspor

untuk mendukung pelaporan serta pemantauan akademik pada tingkat program studi dan fakultas.

Secara keseluruhan, sistem yang dikembangkan telah mendukung pengelolaan karya mahasiswa secara lebih terstruktur serta membantu Fakultas Ilmu Komputer UPN “Veteran” Jakarta dalam menyediakan data karya akhir untuk kebutuhan pengarsipan, pemantauan, pelaporan, dan evaluasi akademik.

4.3. Saran

Berdasarkan kendala yang ditemukan selama proses pengembangan, saran yang dapat diberikan untuk pengembangan sistem selanjutnya adalah sebagai berikut.

1. Sistem perlu diintegrasikan dengan layanan akademik yang telah digunakan, seperti SIM UPNVJ (Sistem Informasi Proses Pembelajaran) atau LENTERA FIK (Layanan Elektronik Tugas Akhir). Integrasi tersebut dapat mendukung pembaruan data akademik secara otomatis sehingga data tidak perlu diperiksa atau disesuaikan kembali secara manual.
2. Pengelolaan peran dan hak akses perlu dikembangkan secara lebih dinamis. Superadmin dapat diberikan fungsi untuk mengatur jenis peran, kewenangan, dan lingkup akses pengguna tanpa melakukan perubahan langsung pada kode program. Pengguna yang memiliki lebih dari satu peran tetap perlu memilih peran aktif agar fungsi dan data yang diakses sesuai dengan kewenangannya.
3. Evaluasi dan pembaruan sistem perlu dilakukan secara berkala dengan melibatkan pengguna dari setiap peran. Setiap perubahan kebutuhan, data referensi, atau kebijakan akademik perlu diikuti dengan penyesuaian dan pengujian ulang pada *frontend*, *backend*, dan basis data agar tidak mengganggu fungsi yang telah berjalan.
4. Sistem perlu dikembangkan dengan pengaturan kapasitas penyimpanan dan batas ukuran unggahan yang dapat disesuaikan untuk setiap jenis berkas. Informasi mengenai kapasitas yang telah digunakan, sisa ruang penyimpanan, dan pemberitahuan ketika kapasitas mendekati batas maksimum juga perlu disediakan agar penggunaan penyimpanan dapat dipantau.
5. Pemantauan terhadap ketersediaan *backend*, PostgreSQL, Gitea, dan sistem berkas server perlu diterapkan untuk memberikan pemberitahuan ketika terjadi gangguan. Selain itu, diperlukan prosedur pencadangan dan pemulihan data yang terstruktur agar layanan dapat dipulihkan dan risiko kehilangan data dapat dikurangi.