

## BAB 4. KESIMPULAN

### 4.1. Kendala

Berdasarkan proses selama implementasi dan pengujian proyek, terdapat beberapa kendala yang dialami selama pengembangan *frontend*. Kendala tersebut adalah sebagai berikut:

1. Terdapat beberapa kali perubahan alur validasi yang menimbang hierarki validator, penyesuaian jumlah suara minimum yang diperlukan, dan penambahan mekanisme *override* oleh superadmin apabila terjadi kebutuhan lain pada proses validasi. Setiap perubahan tersebut berdampak langsung kepada beberapa lapisan *frontend* yang telah dikembangkan, baik dari segi antarmuka maupun logika penentuannya pada `lib/permissions/`.
2. Dikarenakan migrasi data berskala besar beberapa kali, penulis butuh mengubah puluhan *file* yang telah di-*refactor* menjadi beberapa komponen dan bersangkutan dengan adanya migrasi data tersebut. Perubahan tersebut mencakup nama entitas (dari `Proyek` menjadi `TugasAkhir`) di seluruh lapisan sistem, nama atribut (BahasaPemrograman menjadi `Metode`), dan nama kolom (`ekstensi_diizinkan` menjadi `ekstensi_file`, `jenis_berkas_id` menjadi `ref_jenis_berkas`) pada *database*. Karena *frontend* dan *backend* saling terikat melalui tipe data yang dihasilkan oleh skema basis data, maka setiap migrasi memaksa pembaruan pada puluhan *file* Typescript di *frontend*.
3. *Race condition* pada pemilihan peran di mana penulisan *cookie* sesi NextAuth di sisi *server* dan propagasi *cookie* di *browser*. Setelah pengguna memilih peran melalui pemanggilan (`{ role: peran }`), sistem langsung melakukan *redirect* ke halaman *dashboard* sesuai peran terkait. Namun, pada saat pengalihan terjadi, *cookie* JWE yang berisi peran baru belum sepenuhnya diterima oleh *browser*. Hal ini menyebabkan *middleware* Next.js membaca *cookie* dengan peran lama dan mengarahkan pengguna ke halaman yang salah. Kendala ini diatasi dengan menambahkan verifikasi sesi (pembaruan peran dan pemberian jeda 300 milidetik `getSession()` setelah `setTimeout`) sebagai mekanisme *fallback* jika *cookie* belum terpropagasi.

### 4.2. Kesimpulan

Berdasarkan hasil rancangan serta implementasi dan pengujian proyek yang telah dipaparkan, dapat ditarik kesimpulan sebagai berikut:

1. Antarmuka sistem repositori dan demonstrasi karya mahasiswa FIK UPNVJ telah berhasil dirancang dan diimplementasikan sebagai pusat penyajian luaran tugas akhir mahasiswa secara terpusat. Hal ini dibuktikan melalui perancangan *Information Architecture* yang memetakan hierarki menu bagi setiap peran pengguna, penyediaan halaman pencarian dan pengelolaan repositori, serta kapabilitas menjalankan demonstrasi karya yang telah diuji melalui *unit testing* pada tiap komponen sebelum diintegrasikan dengan *backend* dan dinyatakan sesuai dalam pengujian *User Acceptance Testing* (UAT) setelah diintegrasikan dengan *backend*. Dengan demikian, identifikasi masalah pertama mengenai perancangan antarmuka yang mampu menyajikan dan mendemonstrasikan luaran tugas akhir mahasiswa secara terpusat, terstruktur, dan mudah dipahami dapat dinyatakan terjawab.
2. Antarmuka sistem telah berhasil mengakomodasi alur kerja akademik pasca-sidang secara berjenjang dan terintegrasi, mencakup tahap verifikasi administratif oleh admin, tahap validasi akademik oleh dosen dan koorprodi, hingga mekanisme *override* untuk mencegah hambatan pada proses validasi. Kesesuaian antarmuka dengan kebutuhan tiap pengguna diwujudkan melalui penerapan *Role-Based Access Control* (RBAC) dua tingkat, yaitu pada level rute melalui *middleware* Next.js dan pada level fitur melalui modul `lib/permissions/`. Hal ini menghasilkan *dashboard*, *sidebar*, dan hak akses yang ditampilkan menyesuaikan peran masing-masing pengguna. Seluruh alur kerja ini telah divalidasi melalui pengujian UAT dengan tingkat kesesuaian 100% pada seluruh peran yang menguji. Meskipun alur validasi sempat mengalami beberapa kali perubahan selama pengembangan, arsitektur *permission functions* yang bersifat *pure* dan *modular* memungkinkan penyesuaian pada *frontend* maupun *backend* dilakukan secara independen. Kendala *race condition* pada pemilihan peran pengguna multi-peran juga berhasil diatasi melalui mekanisme verifikasi sesi dan jeda waktu untuk propagasi *cookie*. Dengan demikian, identifikasi masalah kedua dapat dinyatakan terjawab.

3. Antarmuka sistem telah berhasil diintegrasikan dengan basis data, Gitea FIK, dan Docker melalui arsitektur REST API yang dilengkapi mekanisme *loading state*, validasi dari sisi pengguna, serta penanganan *error* untuk menjaga responsivitas dan pengalaman pengguna. Tingkat kegunaan (*usability*) sistem turut terbukti baik, ditunjukkan oleh hasil UAT yang mencapai kesesuaian 100% pada seluruh peran pengguna yang diuji. Sebagai bukti kuantitatif dari aspek performa, dilakukan pengukuran menggunakan *Lighthouse Testing*. Meskipun beberapa kali migrasi skema data memaksa pembaruan tipe secara luas di sisi *frontend*, penggunaan TypeScript dan integrasi tipe data Prisma memastikan seluruh ketidakcocokan tipe data terdeteksi pada waktu kompilasi, bukan pada waktu eksekusi. Dengan demikian, identifikasi masalah ketiga dari sisi kegunaan, aspek performa, dan keberhasilan integrasi fungsional antarmuka dapat dinyatakan terjawab secara menyeluruh.

#### 4.3. Saran

Berdasarkan proses perancangan, implementasi, dan pengujian yang telah dilakukan, terdapat beberapa hal yang dapat menjadi bahan pertimbangan untuk penyempurnaan maupun pengembangan sistem pada penelitian atau proyek lanjutan, yaitu:

2. Perubahan alur validasi yang berulang menunjukkan perlunya dokumentasi aturan bisnis dan spesifikasi API yang disepakati sejak awal antara pemangku kepentingan dan tim pengembang. Disarankan untuk menyusun dokumen *business rules* yang secara jelas mendefinisikan hierarki validator, siapa saja yang berhak memberikan validasi, kondisi minimum suara validasi yang dibutuhkan, dan mekanisme *override* sebelum tahap implementasi dimulai. Selain itu, diperlukan juga penyusunan spesifikasi API menggunakan standar OpenAPI/Swagger yang akan menjadi kontrak integrasi yang mengikat antara *frontend* dan *backend* supaya mengurangi ambiguitas format respons dan memfasilitasi selama pengujian.

3. Diperlukannya perencanaan skema data yang matang sebelum implementasi. Migrasi data berskala besar yang berdampak pada puluhan file di *frontend* dapat dihindari apabila skema basis data dirancang dan distabilkan secara matang sebelum pengembangan antarmuka dimulai. Disarankan untuk menerapkan proses *database design review* yang melibatkan perwakilan tim *frontend* dan *backend* sebelum skema ditetapkan. Sehingga, kebutuhan penamaan, relasi, dan tipe data telah dipertimbangkan dari semua sisi.
4. Kendala *race condition* pada pemilihan peran menunjukkan perlunya pengujian integrasi *end-to-end* yang mencakup skenario autentikasi dan otorisasi. Disarankan penambahan rangkaian pengujian integrasi (*integration testing*) yang memvalidasi beberapa hal, yaitu: pengguna dengan peran tunggal mengakses rute di luar kewenangannya, pengguna multi-peran berganti peran dan mengakses fungsionalitas spesifik, pencabutan peran oleh admin yang harus terdeteksi dalam interval waktu yang ditentukan, dan propagasi *cookie* sesi pada saat pemilihan peran untuk mencegah *race condition* serupa.
5. Sistem yang dikembangkan pada penelitian ini masih dioptimalkan untuk penggunaan pada perangkat *desktop* atau *laptop*. Penelitian atau pengembangan selanjutnya dapat mengadaptasi antarmuka agar responsif terhadap perangkat lainnya, sehingga pengguna dapat mengakses repositori dan melakukan validasi melalui *smartphone* atau tablet.
6. Saat ini video demonstrasi karya mahasiswa hanya diunggah sebagai berkas MP4 pada formulir pengumpulan tugas akhir. Pada pengembangan selanjutnya, disarankan untuk mengintegrasikan YouTube API agar video demonstrasi dapat secara otomatis diunggah ke *channel* YouTube Fakultas Ilmu Komputer UPN Veteran Jakarta ketika repositori telah divalidasi dan dipublikasikan. Integrasi ini akan memperluas jangkauan audiens dan memudahkan akses publik terhadap luaran karya mahasiswa.

7. Pengujian non-fungsional tingkat lanjut seperti *stress testing* dan *load testing* belum dilakukan pada penelitian ini. Penelitian selanjutnya dapat mengevaluasi kinerja sistem secara keseluruhan di bawah beban tinggi untuk mengidentifikasi *bottleneck* dan batas kapasitas sistem sebelum dioperasikan secara luas.