

BAB IV KESIMPULAN

4.1 Kesimpulan

Berdasarkan penelitian mengenai implementasi sistem informasi JANAKA untuk legalisasi ijazah dan surat keterangan pengganti ijazah mahasiswa di LLDIKTI Wilayah V dengan metode *Agile*, dapat disimpulkan bahwa sistem ini berhasil dikembangkan untuk mempermudah proses administrasi yang sebelumnya dilakukan secara manual. Dengan menggunakan *framework Laravel* dan database *MySQL*, sistem ini mampu memberikan solusi yang efektif dalam pengelolaan permohonan legalisasi ijazah dan surat keterangan pengganti ijazah. Metode *Agile* yang diterapkan memungkinkan pengembangan sistem berjalan secara iteratif sehingga kebutuhan pengguna dapat terpenuhi dengan lebih cepat dan akurat.

Hasil pengujian menggunakan metode *Black Box Testing* menunjukkan bahwa seluruh fitur yang dikembangkan telah berfungsi dengan baik sesuai spesifikasi yang dirancang. Proses review aplikasi bersama stakeholder juga menunjukkan bahwa sistem ini tidak hanya memenuhi kebutuhan teknis, tetapi juga membantu meningkatkan efisiensi kerja, khususnya dalam mengurangi waktu dan biaya yang diperlukan oleh pemohon, terutama mereka yang berada di luar wilayah Yogyakarta.

4.2 Saran

Penulis menyadari bahwa implementasi sistem informasi JANAKA masih memiliki beberapa kekurangan yang perlu diperbaiki agar dapat memberikan layanan yang lebih optimal. Oleh karena itu, penulis memberikan beberapa saran untuk peningkatan sistem informasi di masa mendatang, antara lain:

1. Menambahkan notifikasi otomatis baik melalui *email*, *SMS*, maupun bot *WhatsApp/Telegram* untuk memberikan pemberitahuan kepada pengguna terkait status pengajuan, seperti ketika pengajuan diterima, ditolak, atau telah selesai diproses.
2. Mengintegrasikan sistem dengan layanan pihak ketiga seperti layanan kurir untuk mempermudah pengiriman dokumen ijazah langsung kepada

pengguna. Integrasi ini juga dapat mencakup pelacakan pengiriman secara real-time agar pengguna dapat mengetahui status pengiriman dokumen mereka.

3. Meningkatkan efisiensi dan kinerja sistem melalui optimasi kode program. Optimasi ini mencakup pemeliharaan kode agar lebih terstruktur, mudah dipahami, dan konsisten. Selain itu, perlu dilakukan penghapusan redundansi kode yang tidak diperlukan untuk menjaga kualitas serta meningkatkan performa sistem.

