

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Program ini adalah sebuah aplikasi berbasis web yang menggunakan Laravel sebagai framework utamanya, dengan pendekatan MVC (Model-View-Controller) untuk mengelola logika bisnis, data, dan tampilan. Aplikasi ini memiliki berbagai fitur untuk mengelola data Penduduk, Kriteria, SubKriteria, dan menghitung peringkat berdasarkan kriteria tertentu. Penggunaan paket Voyager memberikan kemudahan dalam mengelola panel admin, khususnya untuk operasi CRUD.

Program ini menunjukkan struktur yang baik dengan pemisahan tanggung jawab yang jelas antara model, controller, dan view. Penggunaan relasi antar model yang kompleks memungkinkan integrasi dan analisis data yang lebih mendalam. Namun, kompleksitas ini juga memerlukan perhatian khusus untuk memastikan integritas data dan performa sistem yang optimal.

Perhitungan peringkat di `RankingController` adalah fitur utama yang mencerminkan kemampuan program dalam menangani logika bisnis yang rumit. Meskipun demikian, aspek performa dan keamanan harus terus dipantau dan ditingkatkan, terutama jika jumlah data meningkat atau jika program ini digunakan dalam skala yang lebih besar.

Secara keseluruhan, program ini adalah solusi yang cukup komprehensif untuk pengelolaan data berbasis kriteria, dengan potensi untuk diadaptasi dan

dikembangkan lebih lanjut sesuai dengan kebutuhan spesifik pengguna. Aspek UX/UI, validasi data, dan keamanan menjadi elemen penting yang perlu diperhatikan untuk memastikan pengalaman pengguna yang baik dan keandalan sistem secara keseluruhan.

## **5.2 Saran**

Mengingat program ini dapat menangani sejumlah besar data, disarankan untuk mengoptimalkan performa kueri database. Ini dapat dilakukan dengan menggunakan teknik seperti eager loading (`with()`), indexing pada kolom yang sering diakses, dan pengoptimalan kueri kompleks. Selain itu, jika perhitungan peringkat melibatkan data dalam jumlah besar, pertimbangkan penggunaan batch processing atau job queues untuk mengurangi beban pada server.

Pastikan sistem aman dari berbagai ancaman keamanan seperti SQL Injection, Cross-Site Scripting (XSS), dan Cross-Site Request Forgery (CSRF). Implementasikan validasi input yang kuat dan gunakan fitur bawaan Laravel seperti middleware untuk autentikasi dan otorisasi. Pastikan juga data sensitif dienkripsi baik saat disimpan maupun dalam transmisi.

Implementasikan validasi data yang lebih komprehensif di setiap endpoint yang menerima input dari pengguna. Gunakan form request classes Laravel untuk memisahkan logika validasi dari controller. Selain itu, sediakan mekanisme penanganan error yang baik dengan pesan yang informatif untuk pengguna dan logging yang memadai untuk pengembang.

Tingkatkan pengalaman pengguna dengan menyediakan antarmuka yang intuitif dan responsif. Pastikan navigasi mudah digunakan, dan informasi yang relevan mudah diakses. Gunakan notifikasi yang jelas dan konsisten untuk memberikan feedback kepada pengguna tentang status tindakan mereka, seperti sukses atau gagal.

Implementasikan unit testing dan integration testing untuk memastikan setiap komponen berfungsi dengan baik dan tidak terjadi regresi setelah pembaruan kode. Gunakan PHPUnit, yang sudah menjadi standar di Laravel, untuk menulis dan menjalankan tes. Selain itu, pertimbangkan untuk mengimplementasikan end-to-end testing untuk menguji alur pengguna secara keseluruhan.

Dokumentasikan kode dan logika bisnis dengan baik untuk memudahkan pemeliharaan dan pengembangan di masa depan. Buat dokumentasi API jika ada, dan pertimbangkan untuk menggunakan alat seperti Swagger untuk dokumentasi interaktif. Selain itu, pastikan kode mengikuti standar coding yang baik dan konsisten.

Jika program ini digunakan dalam skala yang lebih besar, pertimbangkan arsitektur yang dapat diskalakan seperti microservices atau penggunaan caching untuk mengurangi beban pada database. Implementasikan monitoring dan logging untuk memantau performa aplikasi dan mendeteksi masalah sejak dini.

Dengan menerapkan saran-saran ini, program dapat menjadi lebih efisien, aman, dan mudah digunakan, sekaligus siap untuk menghadapi peningkatan skala

penggunaan di masa depan