

Pengembangan RESTful API Backend Website Customer Relation Management LoyalCust Menggunakan Laravel 11

Tedy Firmansyah^{*1}, Bambang Agus Herlambang²

¹Program Studi Informatika, Universitas Persatuan Guru Republik Indonesia Semarang, Kota Semarang

²Program Studi Informatika, Universitas Persatuan Guru Republik Indonesia Semarang, Kota Semarang

*Email: tedysyhh07@gmail.com

Abstract.

Digital transformation drives the development of Customer Relationship Management (CRM) systems that require structured and integrated customer data management. To support this need, this study develops the backend of a web-based CRM system called LoyalCust using Laravel 11. Laravel 11 was chosen for its performance improvements, modular structure, and ease of managing routing and dependency injection. The development process utilized Visual Studio Code (VS Code), Postman, and Laragon, and applied Representational State Transfer Application Programming Interface (RESTful API) along with system modeling using Unified Modeling Language (UML). The results show that the developed backend system is capable of providing efficient, well-structured API services that support the digital management of customer relationships.

Keywords: *Customer Relationship Management; RESTful API; Backend; Laravel 11.*

Abstrak

Transformasi digital mendorong perkembangan sistem *Customer Relationship Management* (CRM) yang menuntut pengelolaan data pelanggan secara terstruktur dan terintegrasi. Untuk mendukung hal tersebut, penelitian ini mengembangkan backend sistem CRM berbasis web bernama LoyalCust menggunakan Laravel 11. Laravel 11 dipilih karena peningkatan performa, struktur modular, dan kemudahan dalam pengelolaan routing serta *dependency injection*. Pengembangan dilakukan dengan memanfaatkan Visual Studio Code (VS Code), Postman, dan Laragon, serta menerapkan *Representational State Transfer Application Programming Interface* (RESTful API) dan pemodelan sistem menggunakan *Unified Modeling Language* (UML). Hasil pengembangan menunjukkan bahwa sistem backend yang dibangun mampu menyediakan layanan API yang efisien, terstruktur, dan mendukung pengelolaan relasi pelanggan secara digital.

Keywords: *Customer Relationship Management; RESTful API; Backend; Laravel 11.*

1. Pendahuluan

Di era digital yang kompetitif, perusahaan tidak hanya dituntut untuk menarik pelanggan baru, tetapi juga mempertahankan pelanggan lama melalui hubungan jangka panjang [1]. Strategi *Customer Relationship Management* (CRM) terbukti efektif dalam meningkatkan loyalitas dan nilai pelanggan jangka panjang [2], karena memungkinkan pengelolaan data secara terintegrasi untuk layanan yang lebih personal dan efisien [3].

Transformasi digital turut mendorong perkembangan CRM melalui teknologi seperti cloud computing, big data, dan kecerdasan buatan. Cloud memfasilitasi pengelolaan data secara fleksibel, big data membantu analisis perilaku pelanggan, dan kecerdasan buatan digunakan untuk prediksi serta rekomendasi layanan [4][5]. Namun, banyak Usaha Kecil dan Menengah (UKM) menghadapi tantangan dalam mengadopsi teknologi ini akibat keterbatasan anggaran, SDM, dan infrastruktur [6]. Karena itu, diperlukan solusi CRM yang ringan namun fungsional, khususnya pada sisi backend [7].

Pengembangan backend yang kuat dan terstruktur menjadi penting untuk memastikan sistem CRM berjalan stabil, responsif, dan mudah diintegrasikan. *Representational State Transfer* (REST) atau RESTful API merupakan pendekatan arsitektur umum yang efektif

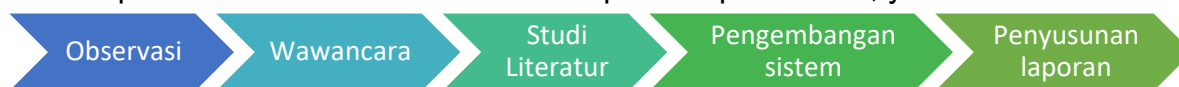
meningkatkan efisiensi dan skalabilitas, seperti ditunjukkan pada pengembangan sistem informasi admisi di Universitas Islam Indonesia [8] dan portal akademik di Universitas Sam Ratulangi [9]. RESTful API menggunakan protokol HTTP yang sederhana, fleksibel, dan mendukung pemisahan frontend-backend, sehingga cocok untuk CRM berbasis web dan mobile [10]. Laravel dipilih sebagai framework karena fitur lengkap, kemudahan penggunaan, dan dukungan komunitas luas, menjadikannya pilihan populer untuk membangun RESTful API [11]. Dalam penelitian ini, penulis mengembangkan backend sistem CRM bernama LoyalCust menggunakan Laravel 11 dengan pendekatan RESTful API. Sistem ini mendukung integrasi real-time dengan frontend dan dikembangkan menggunakan Visual Studio Code, Postman, dan Laragon.

Tujuan proyek ini adalah menghasilkan backend CRM yang andal, fleksibel, dan cocok digunakan oleh UKM, dengan pengelolaan data pelanggan yang terstruktur, layanan API yang konsisten, serta mendukung pengembangan berkelanjutan sesuai kebutuhan bisnis. Pengembangan dilakukan menggunakan metode Scrum untuk memastikan proses yang iteratif, terorganisir, dan adaptif terhadap perubahan kebutuhan selama proyek berlangsung.

2. Metode

2.1 Metode Penelitian

Metode penelitian ini dilakukan melalui empat tahapan utama, yaitu:



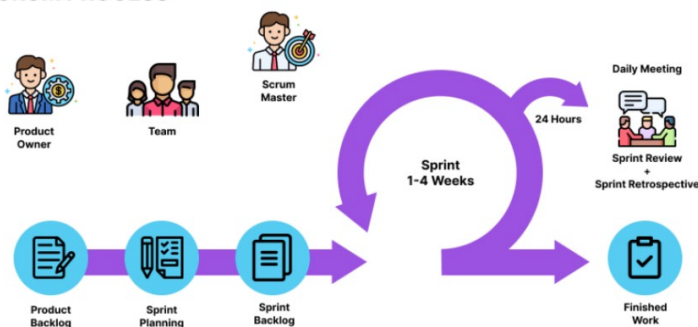
Gambar 1. Tahapan metode penelitian.

- a. **Observasi**
Dilakukan untuk memahami kebutuhan sistem backend dan alur kerja internal. Ditemukan masalah seperti integrasi data yang buruk, dokumentasi minim, dan bug pada validasi serta autentikasi.
- b. **Wawancara**
Dilakukan dengan manajer untuk menggali permasalahan backend lama dan ekspektasi terhadap sistem baru, seperti kestabilan, integrasi, dan efisiensi.
- c. **Studi Literatur**
Mengkaji teori terkait RESTful API, desain API, dokumentasi Laravel 11, debugging, dan testing sebagai dasar pengembangan yang kuat
- d. **Pengembangan Sistem**
Backend CRM dikembangkan menggunakan Laravel 11 dengan fokus pada struktur modular, validasi, penanganan error, dan standar RESTful API. Tools yang digunakan antara lain VS Code, Postman, dan Laragon.
- e. **Penyusunan Laporan**
Seluruh proses dari identifikasi masalah hingga pengujian didokumentasikan sebagai bentuk pertanggungjawaban akademik dan teknis.

2.2 Metode Pengembangan Sistem

Metode rekayasa perangkat lunak scrum diciptakan oleh Jeff Sutherland dan Ken Schwaber dengan tujuan untuk mempercepat perilisasi sebuah produk siap pakai kepada pengguna dengan kualitas produk yang tinggi sesuai dengan prinsip agile [12]. Scrum dipilih karena menggunakan sprint dan tim terstruktur, sehingga memudahkan pengelolaan, kolaborasi, serta komunikasi. Metode ini fleksibel terhadap perubahan, memungkinkan pengujian berkala, dan menghasilkan pengembangan yang efisien serta sesuai kebutuhan pengguna.

SCRUM PROCESS



Gambar 2. Scrum Process.

Dalam metodologi Scrum, setiap Sprint terdiri dari empat tahapan utama yang saling berkesinambungan untuk memastikan iterasi pengembangan berjalan efektif dan terukur. Berikut adalah tahapan dari sebuah sprint [13][14]:

a. *Sprint Planning*

Tahapan ini bertujuan untuk merancang pekerjaan yang akan diselesaikan selama periode sprint berlangsung. Dalam sesi perencanaan ini, *product owner* mengajukan ide-ide untuk meningkatkan nilai dan manfaat dari produk dalam sprint tersebut. Selanjutnya, seluruh anggota tim bekerja sama untuk menentukan target *sprint* yang bernilai bagi para stakeholder.

b. *Daily Scrum*

Daily Scrum adalah pertemuan singkat harian berdurasi maksimal 15 menit yang bertujuan memantau progres, mengidentifikasi hambatan, dan menyelaraskan rencana kerja tim menuju Sprint Goal.

c. *Sprint Review*

Pada tahap ini, tim menyajikan hasil kerja kepada *stakeholder* dan berdiskusi mengenai kemajuan dalam pencapaian *product goal*. Tahapan ini digunakan untuk mengevaluasi hasil dari sprint yang telah selesai dan merancang rencana untuk sprint berikutnya.

d. *Sprint Retrospective*

Tahap ini berfokus pada refleksi terhadap pelaksanaan sprint, mencakup aspek individu, interaksi, proses kerja, alat bantu, serta *definition of done* yang telah disepakati. Tujuan utamanya adalah menyusun langkah-langkah untuk meningkatkan kualitas kerja dan efektivitas tim pada sprint mendatang.

3. Hasil dan Pembahasan

3.1. Hasil

Pengembangan RESTful API backend sistem *Customer Relationship Management* (CRM) di PT. Campus Digital Indonesia menghasilkan beberapa temuan penting yang diperoleh melalui tahapan Scrum. Hasil ini mencakup *Sprint Planning*, *Daily Scrum*, *Sprint Review* dan *Sprint Retrospective*.

a. Hasil *Sprint Planning*

Sistem *Customer Relationship Management* (CRM) memiliki empat *role* yaitu *Super Admin Developer*, *Super Admin*, *Admin* dan *Employee*. Masing-masing *role* memiliki beberapa akses yang berbeda. Pada Pertemuan awal dengan *Product Owner* yaitu manajer PT. Campus Digital Indonesia menghasilkan daftar perencanaan perbaikan dan pengembangan dari RESTful API backend sistem CRM. Daftar tersebut dijelaskan pada Tabel 1 sebagai berikut.

Tabel 1. Daftar Perencanaan Perbaikan dan Pengembangan.

Role	Fitur	Perbaikan/Pengembangan
<i>Super Admin Developer</i> , <i>Super Admin</i> ,	<i>Authorization</i>	Mengubah fungsi pembuatan OTP dapat dilakukan secara paralel dengan fungsi lain yang membutuhkan OTP yaitu edit email oleh <i>Super Admin</i> .

<i>Admin & Employee</i>		Menghilangkan pilihan “lainnya” dalam menu pengisian data jenis kelamin
	Profil	Email dapat diperbarui dengan menambahkan keamanan menggunakan One Time Password (OTP)
	Import dan Export Dokumen	Menambahkan fitur ekspor data penjualan berdasarkan waktu yang diinginkan dalam bentuk file excel
<i>Super Admin, Admin & Employee</i>	Manajemen <i>Leads</i>	Memperbaiki agar sistem dapat melakukan menambah dan menghapus data <i>leads</i>
	Manajemen <i>Contact</i>	Memperbaiki agar sistem dapat melakukan menambah dan menghapus data <i>contact</i>
	Manajemen <i>Customer Company</i>	Memperbaiki agar sistem dapat melakukan tambah dan menghapus data <i>customer company</i>
	Manajemen <i>Deals</i>	Melakukan perbaikan perhitungan penjualan di setiap tahapan dengan akurat
		Memperbaiki agar sistem dapat menghapus data penjualan
<i>Super Admin & Admin</i>	Manajemen <i>Product/Service</i>	Melakukan penyesuaian aturan dalam menu edit data penjualan agar lebih efektif
		Memperbaiki agar sistem dapat menghapus data produk/jasa
<i>Super Admin</i>	Statistik Pelanggan	Menambahkan sistem ranking pada data karyawan berdasarkan banyaknya pelanggan yang didapat
		Menambahkan fitur statistik pelanggan berdasarkan karyawan yang memasukkannya dalam rentang waktu tahunan, bulanan dan mingguan
<i>Super Admin Developer</i>	Manajemen <i>User Company</i>	Membuat pembagian data pengguna yang berstatus aktif dan tidak aktif
	Manajemen <i>Prices</i>	Menambahkan fitur manajemen harga pada sistem
	Manajemen <i>Artikel</i>	Memperbaiki agar sistem dapat menambah, menghapus dan mengedit data artikel

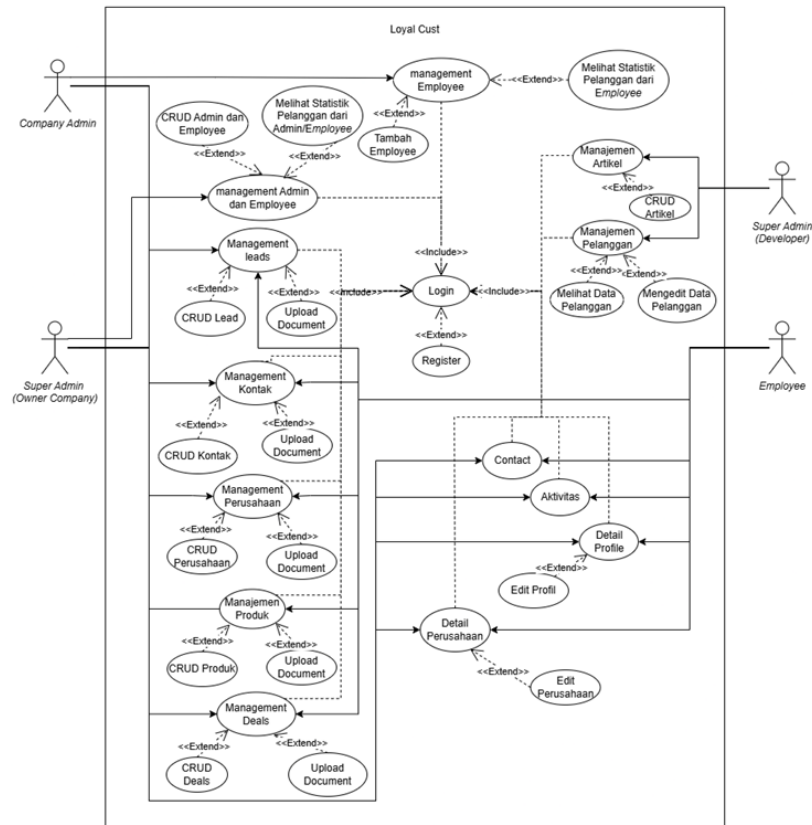
b. Hasil *Daily Scrum*

Pada tahap *Daily Scrum*, tim menyusun rencana kerja harian dan merancang sistem menggunakan UML. UML dipilih karena merupakan standar internasional untuk memvisualisasikan dan mendokumentasikan sistem. Pemodelan ini membantu tim memahami struktur dan alur kerja sistem secara selaras. Diagram UML yang digunakan meliputi use case, class, activity, dan sequence diagram.

1. *Use Case Diagram*

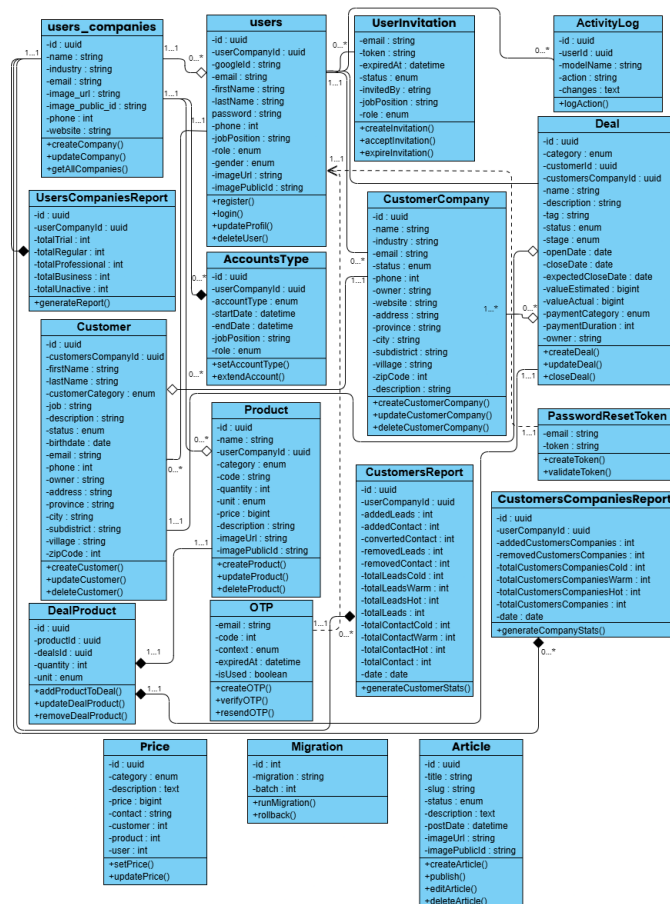
Use case diagram merupakan representasi visual dari perilaku sistem yang menggambarkan kebutuhan dan ekspektasi pengguna terhadap fungsi-fungsi yang disediakan oleh sistem [15]. Sistem terdiri dari empat *role*, yaitu *Super Admin Developer*, *Super Admin*, *Admin*, dan *Employee*, yang masing-masing memiliki hak akses berbeda.

Super Admin Developer memiliki akses penuh untuk manajemen pengguna dan konten sistem. *Super Admin*, sebagai pemilik perusahaan, dapat mengakses seluruh fitur CRM, termasuk rekrutmen, statistik karyawan, manajemen produk, pelanggan, dan penjualan. *Admin* memiliki akses serupa namun tanpa wewenang untuk merekrut karyawan. Sementara itu, *Employee* hanya dapat mengelola data pelanggan dan penjualan miliknya sendiri.



Gambar 3. Use Case Diagram.

2. Class Diagram

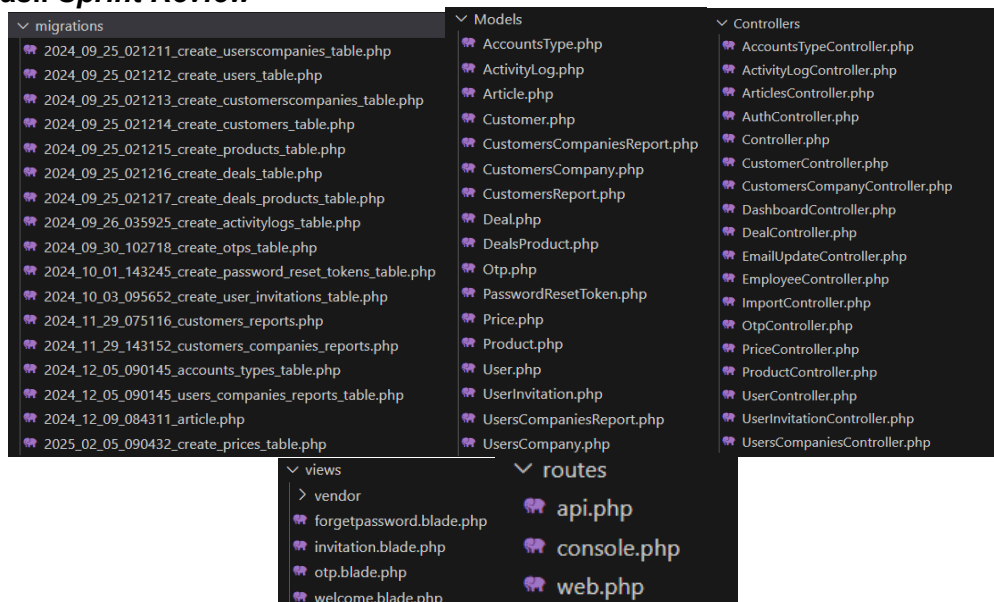


Gambar 4. Class Diagram.

Class Diagram merupakan representasi visual dari struktur statis suatu sistem perangkat lunak. Diagram ini sangat berguna dalam proses perancangan sistem berorientasi objek, karena membantu dalam mengidentifikasi dan memodelkan struktur data serta interaksi antar komponen sistem [16]. Gambar 4. menunjukkan relasi antar kelas dalam sistem melalui Class Diagram, menampilkan arah dan kardinalitas hubungan. Diagram ini membantu merancang sistem berorientasi objek dengan memodelkan struktur data dan interaksi antar komponen, seperti *users*, *users_companies*, dan *customers*.

Sebagian besar relasi bersifat One-to-Many, misalnya antara *users_companies* dengan *users*, *accounts_types*, dan *products*. Ada juga relasi Many-to-One dan One-to-One, seperti antara *deals* dan *products*, serta *users* dengan *otps/PasswordResetToken*. Pemetaan ini penting untuk menyusun diagram kelas UML yang sistematis dan konsisten.

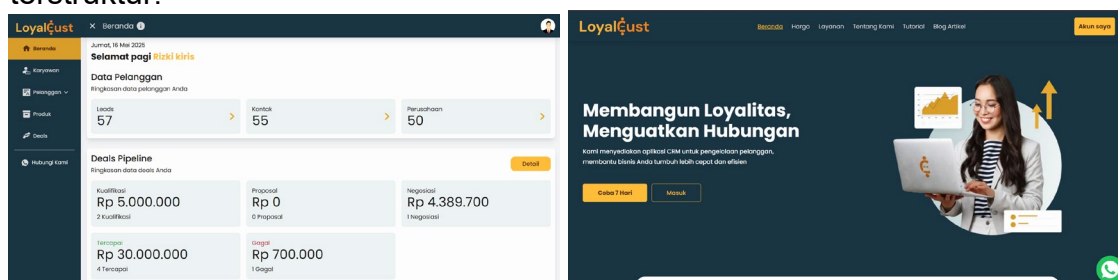
c. Hasil Sprint Review



Gambar 5. Struktur Migration, Model, View, Controller, Route.

Struktur Laravel 11 dalam sistem CRM terdiri dari beberapa komponen utama yang mendukung pengembangan modular dan terorganisir. Migrasi membentuk skema database dengan tabel utama seperti *users*, *customers*, dan *products*, serta tabel pendukung untuk aktivitas dan keamanan. Model merepresentasikan tabel-tabel tersebut, seperti *User*, *Customer*, dan *Product*, serta mendukung relasi dan fungsi sistem.

View menggunakan Blade untuk menyajikan antarmuka pengguna yang terpisah dari logika aplikasi, sesuai prinsip MVC. Controller menangani logika fitur seperti autentikasi, produk, pelanggan, dan artikel. Route dibagi menjadi *web.php*, *api.php*, dan *console.php*, memisahkan akses berdasarkan jenis layanan, sehingga sistem lebih terstruktur.



Gambar 6. Implementasi Frontend.

Setelah semua *route API* dibuat, dilakukan implementasi API ke *frontend*. Pada gambar 6. adalah tampilan dari implementasi API ke dalam tampilan sistem CRM LoyalCust. Dengan

user interface yang simple dan mudah dipahami, diharapkan pengguna dapat mudah menyesuaikan dalam menggunakan sistem CRM LoyalCust.

d. Hasil *Sprint Retrospective*

Evaluasi sprint menunjukkan kolaborasi tim berjalan baik, terutama dalam implementasi fitur backend dan penggunaan UML yang membantu pemahaman sistem. Laravel 11 dinilai efektif mendukung pengembangan modular. Kendala yang ditemukan meliputi kurangnya dokumentasi awal API dan komunikasi teknis yang belum optimal antar tim. Sebagai perbaikan, tim akan menyusun dokumentasi API secara paralel, menambah sesi sinkronisasi antar divisi, serta menggunakan alat bantu seperti Postman Collection untuk mempermudah pengujian.

3.2. Pembahasan

Pengembangan backend CRM LoyalCust dengan Laravel 11 berhasil diterapkan menggunakan RESTful API yang memisahkan frontend dan backend secara jelas. Laravel 11 dipilih karena kemudahan dalam *routing*, *middleware*, dan struktur modular yang mendukung efisiensi pengembangan. Melalui metode Scrum, pengembangan dilakukan secara iteratif. Sprint Planning merumuskan kebutuhan fitur tiap role, sementara Daily Scrum difokuskan pada pemodelan sistem menggunakan UML. Sprint Review menunjukkan sistem telah dibangun dengan pola MVC yang terstruktur. Evaluasi pada *Sprint Retrospective* mengidentifikasi kendala dokumentasi dan komunikasi, yang diatasi dengan penjadwalan sinkronisasi rutin dan dokumentasi API menggunakan Postman *Collection*.

Website dikembangkan menggunakan pola arsitektur MVC Laravel yang memisahkan aplikasi menjadi tiga komponen utama *Model* (pengelolaan data), *View* (tampilan antarmuka), dan *Controller* (logika proses), dengan fokus pada *Migration*, *Model*, *View*, *Controller*, dan *Route*. Pendekatan ini didasarkan pada prinsip pemisahan tanggung jawab, di mana setiap komponen memiliki peran spesifik yang memungkinkan pengorganisasian kode menjadi lebih modular dan terstruktur. Dengan pembagian tersebut, reusabilitas kode dapat ditingkatkan secara signifikan, yang menjadi salah satu keuntungan utama dari penerapan pola ini [17].

4. Kesimpulan

Proyek pengembangan backend CRM LoyalCust berbasis Laravel 11 berhasil menciptakan sistem yang efisien, modular, dan siap mendukung pengelolaan relasi pelanggan secara digital. RESTful API memberikan fleksibilitas dalam integrasi sistem, sementara penerapan metode Scrum memungkinkan pengembangan yang terstruktur dan responsif terhadap kebutuhan. Hasil akhir menunjukkan bahwa sistem ini sesuai untuk digunakan oleh UKM maupun organisasi lain yang membutuhkan solusi CRM ringan namun fungsional. Ke depan, peningkatan kualitas dokumentasi dan koordinasi tim akan menjadi aspek penting untuk mendukung keberlanjutan dan skalabilitas pengembangan.

5. Referensi

- [1] C. V. Andaryani and H. Alifahmi, "Penerapan Customer Relationship Management Dalam Sosial Media Untuk Mempertahankan Loyalitas Pelanggan," *Jurnal Pustaka Komunikasi*, vol. 6, no. 2, pp. 404–417, Sep. 2023, doi: 10.32509/pustakom.v6i2.3171.
- [2] S. S. Sholeha, T. Longga, and M. H. Chaska, "Penerapan Customer Relationship Management (CRM) di Alfamart," *Inisiatif: Jurnal Ekonomi, Akuntansi dan Manajemen*, vol. 4, no. 1, pp. 187–191, Dec. 2024, doi: 10.30640/inisiatif.v4i1.3481.
- [3] R. Roisah, I. Iskandar, R. Mahanka, A. Trijumansyah, and N. Yunita, "Upaya Meningkatkan Loyalitas Pelanggan Melalui Customer Relationship Management (Studi Kasus Pada Point Media Sejahtera)," *Jurnal Abdimas BSI*, vol. 2, no. 1, pp. 45–52, Feb. 2019.
- [4] T. Triana and I. Oktaviani, "Penerapan Cloud Computing Dalam Manajemen Pelanggan (Crm) Untuk Meningkatkan Loyalitas Konsumen," *Biner: Jurnal Ilmiah Informatika dan Komputer*, vol. 4, no. 1, pp. 39–43, Jan. 2025, doi: 10.32699/biner.v4i1.8893.

- [5] M. Shofiudin, T. M. P. Wiryawanto, Z. Hawani, and F. M. Amin, "TREN IMPLEMENTASI CUSTOMER RELATIONSHIP MANAGEMENT BERBASIS ARTIFICIAL INTELLIGENCE," *JSil (Jurnal Sistem Informasi)*, vol. 11, no. 1, pp. 27–32, Mar. 2024, doi: 10.30656/jsii.v11i1.8282.
- [6] C. V. Andaryani and H. Alifahmi, "Penerapan Customer Relationship Management Dalam Sosial Media Untuk Mempertahankan Loyalitas Pelanggan," *Jurnal Pustaka Komunikasi*, vol. 6, no. 2, pp. 404–417, Sep. 2023, doi: 10.32509/pustakom.v6i2.3171.
- [7] T. Gaur, "Applications of Micro-Frontend Application Development in a Customer Support CRM," *International Journal of Computer Trends and Technology*, vol. 72, no. 6, pp. 15–24, Jun. 2024, doi: 10.14445/22312803/IJCTT-V72I6P103.
- [8] M. F. Ramadhan and Z. Zuhri, "Pengembangan Rest Api Sistem Uiiadmisi Dengan Menggunakan Pendekatan Domain Driven Design," *Jurnal Ilmiah Informatika*, vol. 11, no. 02, pp. 176–182, Sep. 2023, doi: 10.33884/jif.v11i02.8017.
- [9] F. X. Senduk, X. B. N. Najoran, and S. R. U. A. Sompie, "Pengembangan Arsitektur Microservices dengan RESTful API Gateway menggunakan Backend-for-frontend Pattern pada Portal Akademik Perguruan Tinggi," *Jurnal Teknik Informatika*, vol. 18, no. 1, pp. 315–324, Mar. 2023, doi: 10.35793/jti.v18i1.50402.
- [10] M. A. Novianto and S. Munir, "Analisis dan Implementasi RESTful API guna Pengembangan Sistem Informasi Akademik pada Perguruan Tinggi," *Jurnal Informatika Terpadu*, vol. 8, no. 1, pp. 47–61, Mar. 2022, doi: 10.54914/jit.v8i1.409.
- [11] F. Sinlae, E. Irwanda, Z. Maulana, and V. Eka Syahputra, "Penggunaan Framework Laravel dalam Membangun Aplikasi Website Berbasis PHP," *Jurnal Siber Multi Disiplin*, vol. 2, no. 2, pp. 119–132, Jul. 2024, doi: 10.38035/jsmd.v2i2.186.
- [12] J. E. Firdaus and Y. Prayudi, "PENGEMBANGAN BACKEND MEDIA SOSIAL DENGAN MENGGUNAKAN METODE SCRUM," *EDUSAINTEK: Jurnal Pendidikan, Sains dan Teknologi*, vol. 11, no. 2, pp. 416–430, Sep. 2023, doi: 10.47668/edusaintek.v11i2.1020.
- [13] A. Mustika, "Permodelan Sistem Informasi Penjualan Barang Menggunakan Metode Scrum," *Journal of Data Science and Information System (DIMIS)*, vol. 2, no. 1, pp. 1–8, Feb. 2024.
- [14] S. Buyukguzel and R. Mitchell, "Progressivity in Hybrid Meetings: Daily Scrum as an Enabling Constraint for a Multi-Locational Software Development Team," *Computer Supported Cooperative Work (CSCW)*, vol. 32, no. 3, pp. 573–606, Sep. 2023, doi: 10.1007/s10606-022-09454-3.
- [15] K. D. Purnama and H. Susilowati, "The Evolution and Challenges of CRM Implementation in the Digital Economy: A Systematic Literature Review," *Journal of Management and Informatics*, vol. 3, no. 2, pp. 312–327, Aug. 2024, doi: 10.51903/jmi.v3i2.39.
- [16] S. Al-Fedaghi, "Classes in Object-Oriented Modeling (UML): Further Understanding and Abstraction," *IJCSNS International Journal of Computer Science and Network Security*, vol. 21, no. 5, pp. 139–150, May 2021.
- [17] H. S. Hadi and W. Yahyan, "Sistem Pengajuan Judul Tugas Akhir Di Universitas Ekasakti Arsitektur Model View Controller," *Prosiding SISFOTEK*, vol. 4, no. 1, pp. 221–224, Aug. 2020.