

Sistem Manajemen Layanan Pelanggan Di CV *Asian Technology Inspiration* Berbasis *Website* Menggunakan *Framework Laravel*

Ardan Yafi Hamam¹, Rangga Sanjaya²

^{1,2}Program Studi Sistem Informasi, Universitas Adhirajasa Reswara Sanjaya
e-mail: ¹16200022@ars.ac.id, ²rangga@ars.ac.id

Abstrak

CV *Asian Technology Inspiration* mengalami kesulitan dalam pengelolaan data pelanggan secara manual, yang menyebabkan ketidaksesuaian dan kesulitan pencarian data. Penelitian ini bertujuan mengembangkan sistem manajemen layanan pelanggan berbasis *website* menggunakan *framework Laravel* untuk meningkatkan efisiensi dan hubungan dengan pelanggan. Metode yang digunakan adalah *waterfall*, dimulai dengan analisis kebutuhan melalui wawancara dan observasi. Rancangan sistem mencakup diagram *use case*, *class diagram*, dan desain *database* menggunakan *Entity Relationship Diagram (ERD)*, dengan fitur utama dalam *Customer Relationship Management (CRM)* dan pengelolaan data pelanggan. Sistem yang dikembangkan berhasil meningkatkan efisiensi pengelolaan data dan berfungsi sesuai harapan. Implementasi sistem ini memungkinkan CV *Asian Technology Inspiration* memberikan layanan yang lebih baik dan meningkatkan kepuasan pelanggan.

Kata kunci— Sistem Manajemen Layanan Pelanggan, Framework Laravel, Website

Abstract

CV *Asian Technology Inspiration* has difficulty in managing customer data manually, which causes discrepancies and difficulty searching for data. This research aims to develop a website-based customer service management system using the Laravel framework to improve efficiency and customer relations. The method used is *waterfall*, starting with needs analysis through interviews and observations. The system design includes *use case diagram*, *class diagram*, and *database design* using *Entity Relationship Diagram (ERD)*, with the main features in *Customer Relationship Management (CRM)* and customer data management. The developed system successfully improves the efficiency of data management and functions as expected. The implementation of this system allows CV *Asian Technology Inspiration* to provide better services and increase customer satisfaction.

Keywords— Customer Relationship Management, Laravel Framework, Website

Corresponding Author:

Rangga Sanjaya,

Email: rangga@ars.ac.id

1. PENDAHULUAN

Persaingan bisnis yang semakin ketat pada era globalisasi memaksa perusahaan untuk terus berinovasi guna meningkatkan pelayanan kepada pelanggan agar dapat mempertahankan reputasi dan eksistensi di dunia bisnis. Perusahaan harus fokus pada upaya mempertahankan pelanggan untuk menciptakan loyalitas, yang pada gilirannya akan mengurangi kemungkinan pelanggan berpaling kepada pesaing. Salah satu strategi utama yang dapat diterapkan adalah dengan menggunakan sistem *Customer Relationship Management (CRM)* [1].

CRM adalah pendekatan bisnis yang mengintegrasikan manusia, proses, dan teknologi untuk mengelola hubungan dengan pelanggan secara efisien. Sistem ini memungkinkan perusahaan memanfaatkan sumber daya secara maksimal untuk menjaga hubungan pelanggan melalui strategi yang terstruktur [2]. Menurut penelitian yang dilakukan oleh Silaswara [3], CRM memberikan dampak signifikan dalam meningkatkan keuntungan perusahaan, karena

memungkinkan analisis kebutuhan pelanggan dilakukan dengan lebih mudah. Selain itu, *CRM* mempercepat proses penyimpanan data, menghasilkan laporan penjualan yang akurat, serta membantu pengambilan keputusan yang lebih baik berdasarkan data yang terorganisir [4].

Berbagai studi sebelumnya telah mengkaji manfaat penerapan sistem *CRM*, Seperti penelitian oleh Prasetyo [5] yang mengembangkan sistem *CRM* elektronik berbasis *framework dynamic CRM* untuk toko *Clashy*. Penelitian ini menunjukkan bahwa penerapan *CRM* tidak hanya meningkatkan kepuasan dan loyalitas pelanggan, tetapi juga mendorong penjualan yang lebih baik. Penelitian lainnya yang dilakukan oleh Galih [1] pada PT Aguna Amreta Ardaya menggunakan sistem *CRM* berbasis website untuk meningkatkan struktur manajemen perusahaan. Hasil penelitian ini mengungkapkan bahwa sistem *CRM* membuat pengelolaan data konsumen lebih terorganisir sehingga perusahaan dapat dengan mudah mendapatkan informasi yang dibutuhkan.

Pada penelitian ini, penerapan sistem *Customer Relationship Management (CRM)* dilakukan pada perusahaan kondisi CV. *Asian Technology Inspiration*, sebuah perusahaan yang berlokasi di Kecamatan Bandung Kidul, Kota Bandung, Jawa Barat. Saat ini, sistem manajemen di perusahaan tersebut masih dilakukan secara manual, yang menimbulkan sejumlah kendala operasional. Data pelanggan sering kali tidak sesuai atau tidak ter-update, proses pencarian informasi membutuhkan waktu yang lama, dan evaluasi kinerja karyawan sulit dilakukan. Selain itu, perusahaan menghadapi kesulitan dalam menangani komunikasi dengan pelanggan dan karyawan yang berada di luar kota, sehingga keluhan pelanggan tidak selalu dapat diselesaikan secara efisien. Masalah ini menghambat perusahaan dalam mencapai efisiensi operasional dan kualitas layanan yang optimal.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem *CRM* berbasis *website* menggunakan *framework Laravel* dengan pendekatan *metode waterfall*. Sistem ini dirancang tidak hanya untuk mengelola data pelanggan, tetapi juga untuk memantau kinerja karyawan, menyelesaikan keluhan pelanggan, serta meningkatkan komunikasi dan koordinasi antar pihak yang terlibat. Sistem *CRM* yang diusulkan diharapkan dapat memberikan solusi terintegrasi yang menyatukan pengelolaan data pelanggan dan tugas karyawan dalam satu platform.

2. METODE PENELITIAN

2.1. Pengumpulan Data

Pada tahap pengumpulan data, peneliti menggunakan dua metode utama, yaitu wawancara dan observasi. Wawancara dilakukan dengan staf CV. *Asian Technology Inspiration*, khususnya pada bagian pengelolaan pelanggan, untuk menggali informasi terkait permasalahan yang dihadapi dalam manajemen data dan pelayanan pelanggan. Wawancara ini juga bertujuan untuk memahami kebutuhan spesifik perusahaan dalam merancang sistem *Customer Relationship Management (CRM)*.

Selain itu, peneliti juga melakukan observasi langsung ke lokasi perusahaan untuk mempelajari alur pengelolaan data pelanggan, proses pencatatan layanan, penanganan keluhan, dan pembuatan laporan aktivitas pelayanan. Metode ini membantu peneliti mendapatkan gambaran menyeluruh tentang proses operasional yang sedang berjalan, sehingga dapat dijadikan dasar yang kuat untuk pengembangan sistem yang sesuai dengan kebutuhan perusahaan.

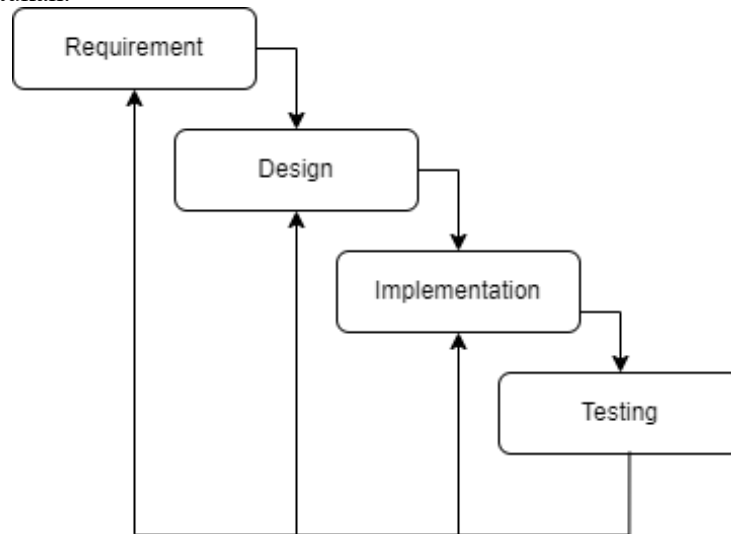
2.2. Studi Pustaka

Studi pustaka dilakukan dengan mengacu pada berbagai jurnal dan *e-book* yang relevan untuk memperoleh wawasan dan landasan teoretis yang mendukung pengembangan sistem.

2.3. Pengembangan Aplikasi

Pada tahapan ini, peneliti menggunakan metode *waterfall* karena pendekatannya yang terstruktur dan sistematis, yang memungkinkan setiap tahap pengembangan sistem dapat berjalan secara berurutan [6]. Tahapan pengembangan aplikasi akan dijelaskan secara mendalam setiap tahap pelaksanaan, mulai dari identifikasi kebutuhan sistem, perancangan arsitektur, hingga tahap

pengujian, guna memastikan bahwa sistem yang dikembangkan berfungsi sesuai dengan tujuan yang telah ditentukan.



Gambar 1. Tahapan Metode *Waterfall*

Berikut ini merupakan penjelasan mengenai setiap tahapan yang terdapat pada metode pengembangan aplikasi *waterfall*:

1. *Requirement*

Pada tahapan ini penulis memulai dengan mengumpulkan informasi mengenai kebutuhan sistem yang akan dikembangkan. Penulis melakukan wawancara dan observasi langsung di CV *Asian Technology Inspiration* sebagai bagian dari proses pengumpulan data. Tujuan utama dari kegiatan ini adalah untuk memahami kebutuhan perusahaan secara lebih mendalam dan memastikan bahwa sistem yang akan dikembangkan dapat menjawab permasalahan yang ada.

2. *Design*

Pada tahapan ini penulis merancang sistem menggunakan beberapa alat bantu perancangan untuk memvisualisasikan dan mengorganisasi alur serta struktur sistem. Penulis menggunakan *UML use case diagram* untuk menggambarkan interaksi antara pengguna dan sistem [7], *class diagram* untuk mendefinisikan struktur statis dan hubungan antar objek [8], serta *sequence diagram* untuk memodelkan alur kerja dan interaksi antar objek secara berurutan [9]. Selain itu, penulis juga merancang basis data menggunakan *ERD (Entity Relationship Diagram)* untuk memetakan hubungan antar entitas [10] dan *LRS (Logical Relational Schema)* untuk mendesain tabel serta kunci yang menghubungkan data [10].

3. *Implementation*

Pada tahapan ini penulis mulai mengembangkan sistem menggunakan *framework backend Laravel* dan *framework frontend Vue.js*. *Laravel* dipilih karena kemampuannya dalam menyediakan struktur *MVC (Model-View-Controller)* yang kuat serta fitur-fitur bawaan seperti *routing*, pengelolaan *database*, dan otentikasi, yang memudahkan proses pengembangan aplikasi [11]. Sementara itu, untuk bagian *frontend*, penulis menggunakan *Vue.js* sebagai *framework* yang berfokus pada tampilan dinamis dan interaktif [12].

4. *Testing*

Pada tahapan ini penulis melakukan pengujian sistem untuk memastikan bahwa fungsionalitasnya sesuai dengan spesifikasi yang telah ditetapkan. Dalam penelitian ini, penulis menggunakan metode *black box testing*. Metode ini berfokus pada pengujian fungsionalitas sistem tanpa memeriksa *source code* atau struktur internalnya [13]. Pengujian dilakukan dengan memberikan berbagai input ke sistem dan mengamati hasil output untuk memastikan bahwa sistem merespons sesuai dengan ekspektasi.

3. HASIL DAN PEMBAHASAN

3.1. Hasil Analisis Kebutuhan

Analisis kebutuhan adalah tahap penting dalam pengembangan sistem yang melibatkan pengumpulan informasi melalui wawancara, observasi, dan studi literatur untuk memahami kebutuhan pengguna dan pemangku kepentingan. Berdasarkan analisis tersebut, sistem memiliki tiga peran utama, yaitu:

1. Admin

Mengelola data domain, pelanggan, produk, tugas *support*, memantau transaksi, dan mengawasi karyawan *support*.

2. Support

Bertugas menerima tugas dari *Admin*, memproses pesanan, menangani masalah teknis, mengelola domain/produk, melaporkan hasil kepada *Admin*, dan memberikan dukungan teknis untuk *User*.

3. User

Menggunakan sistem untuk pembelian domain, dengan akses terbatas untuk pemesanan, pembayaran, dan memantau riwayat transaksi, serta dapat meminta bantuan *Support* jika mengalami masalah.

3.2. Hasil Perancangan Analisis Sistem

3.2.1. Desain Sistem

A. Usecase Diagram

Use Case Diagram pada gambar dibawah menggambarkan interaksi antara tiga aktor utama dalam Sistem Manajemen Layanan Pelanggan, yaitu *Admin*, *Support*, dan *Customer*. *Admin* memiliki berbagai tanggung jawab dalam sistem, seperti mengelola pengguna, domain, vendor, transaksi, pekerjaan karyawan, dan produk. *Admin* dapat menambah, menghapus, dan memperbarui data pengguna, serta mengelola data domain dan vendor. Selain itu, *Admin* bertugas untuk memantau transaksi yang dilakukan oleh pelanggan dan mengatur pekerjaan yang diberikan kepada *Support*. *Admin* juga mengelola produk yang dijual di dalam sistem.

Support berperan untuk membantu pelaksanaan tugas yang diberikan oleh *Admin*. Mereka memiliki kemampuan untuk membuat daftar tugas (*to-do list*) yang diberikan kepada mereka dan melaporkan progres atau penyelesaian pekerjaan kepada *Admin*. Sementara itu, *Customer* dapat melakukan dua hal utama dalam sistem, yaitu melakukan registrasi untuk mendaftarkan akun dan melakukan pembelian produk atau layanan yang tersedia di dalam sistem.

Secara keseluruhan, diagram ini menjelaskan bagaimana masing-masing aktor berinteraksi dengan berbagai fungsionalitas dalam sistem untuk menjalankan tugas dan peran mereka sesuai dengan akses yang diberikan.

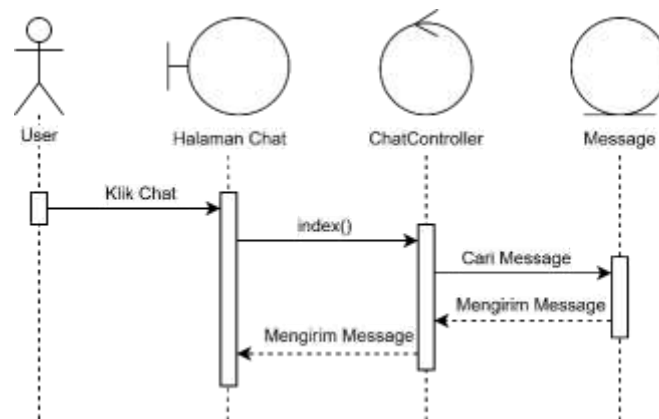


Gambar 2. Usecase Diagram

B. Sequence Diagram

Sequence diagram yang dibuat dalam penelitian ini bertujuan untuk memperjelas alur interaksi antara komponen sistem dalam sistem yang dirancang. Diagram yang dibuat pada penelitian ini mencakup beberapa diagram *sequence* utama, diantaranya *sequence diagram* hubungan layanan pelanggan berbasis *live chat*, dan pemesanan *website* hingga proses pembayaran. Berikut merupakan *sequence diagram* yang telah buat.

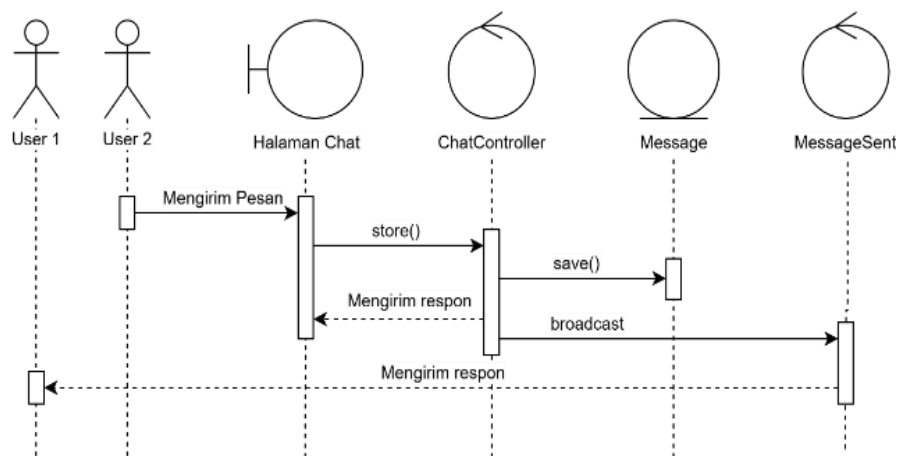
1. Live Chat



Gambar 3. Sequence Diagram Buka Halaman Chat

Diagram sequence ini menggambarkan alur komunikasi pada sistem chat. Proses dimulai ketika user mengklik chat di antarmuka, yang memuat halaman *Chat*. Halaman ini mengirimkan permintaan ke *ChatController* melalui fungsi *index()*, yang kemudian mencari pesan-pesan terkait di *Model Message*. Pesan-pesan yang ditemukan dikembalikan dan ditampilkan di halaman *Chat* untuk *user*.

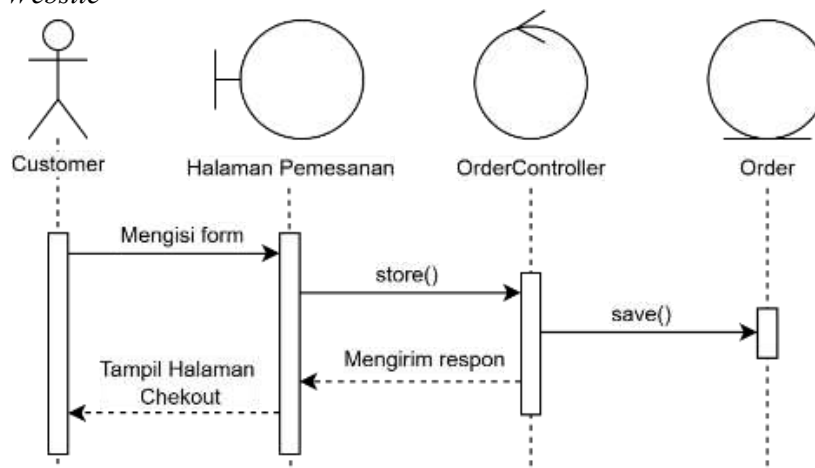
Dalam proses *live chat*, terdapat aktivitas lain yang dapat dilakukan, yaitu pengiriman yang akan dijelaskan pada gambar dibawah ini.



Gambar 4. Sequence Diagram Mengirim Pesan

Dalam diagram diatas, *User 1* atau *User 2* dapat berperan sebagai *support* atau *customer*. Alurnya dimulai ketika salah satu user mengirim pesan melalui halaman *Chat*. Pesan tersebut dikirim ke *ChatController* melalui fungsi *store()*, yang kemudian menyimpan pesan di *Model Message* menggunakan fungsi *save()*. Setelah pesan tersimpan, sistem melakukan *broadcast* pesan tersebut melalui event *MessageSent*, yang memungkinkan *user* lain (baik itu *support* atau *customer*) untuk menerima pesan dalam waktu nyata.

2. Pemesanan Website



Gambar 5. Sequence Diagram Membuat Pesanan Website

Diagram sequence ini menunjukkan proses pemesanan yang dilakukan oleh customer. Proses dimulai ketika customer mengisi formulir pemesanan di halaman pemesanan. Setelah formulir dikirim, data tersebut diteruskan ke *OrderController* melalui fungsi *store()*. *OrderController* kemudian menyimpan data pemesanan dengan memanggil fungsi *save()* pada *Model Order*. Setelah data berhasil disimpan, customer diarahkan ke halaman *checkout* untuk melihat detail pemesanan atau melanjutkan ke proses pembayaran.



Gambar 8. Diagram Logical Record Structure

3.2.3. Pengujian *Blackbox Testing*

Berikut merupakan tabel yang menunjukkan hasil dari pengujian menggunakan *blackbox testing* pada setiap aktivitas yang ada pada aplikasi.

Tabel 1. Hasil Pengujian *Blackbox Testing*

No.	Pengujian	Perintah	Hasil	Keterangan
1	Login	Pengguna mengisi form di halaman login.	Login berhasil dan mengarahkan ke halaman terkait.	Sesuai
2.	Validasi login	Pengguna mengisi form dengan kondisi username dan password salah di halaman login.	Login gagal dan kembali ke halaman login dengan pesan error.	Sesuai
3.	Register	Pengguna mengisi form di halaman register.	Registrasi berhasil dan mengarahkan ke halaman landing page dengan kondisi sudah login.	Sesuai
4.	Validasi register	Pengguna mengisi form di halaman register dengan username atau email yang sudah ada pada database.	Registrasi gagal dan kembali ke halaman register dengan pesan error.	Sesuai
5.	Logout	Pengguna melakukan klik logout.	Logout berhasil dan mengarahkan ke halaman landing page.	Sesuai

6.	Admin dashboard	Admin klik tombol dashboard pada navigasi.	Menampilkan halaman dashboard.	Sesuai
7.	Daftar user	Admin klik tombol user pada navigasi.	Menampilkan halaman user yang menampilkan seluruh data user.	Sesuai
8.	Tambah user	Admin mengisi form di halaman tambah user.	Data user berhasil di tambahkan dan mengarahkan ke halaman daftar user.	Sesuai
9.	Validasi tambah user	Admin mengisi form di halaman tambah user dengan kondisi username atau email sudah ada di database.	Tambah user gagal dan mengarahkan kembali ke halaman tambah user dengan pesan error.	Sesuai
10.	Edit user	Admin mengisi form di halaman edit user.	Data user berhasil diubah dan mengarahkan ke halaman daftar user.	Sesuai
11.	Validasi edit user	Admin mengisi form di halaman edit user dengan kondisi username atau email sudah ada di database.	Edit user gagal dan mengarahkan kembali ke halaman edit user dengan pesan error.	Sesuai
12.	Hapus user	Admin klik tombol hapus pada kolom user.	Data user berhasil di hapus.	Sesuai
13.	Daftar domain	Admin klik tombol domain pada navigasi.	Menampilkan halaman domain yang menampilkan seluruh data domain.	Sesuai
14.	Tambah domain	Admin mengisi form di halaman tambah domain.	Data domain berhasil di tambahkan dan mengarahkan ke halaman daftar domain.	Sesuai
15.	Validasi tambah domain	Admin mengisi form di halaman tambah domain dengan kondisi nama domain sudah ada di database.	Tambah domain gagal dan mengarahkan kembali ke halaman tambah domain dengan pesan error.	Sesuai
16.	Edit domain	Admin mengisi form di halaman edit domain.	Data domain berhasil diubah dan mengarahkan ke halaman daftar domain.	Sesuai

17.	Validasi edit domain	Admin mengisi form di halaman edit domain dengan kondisi nama domain sudah ada di database.	Edit domain gagal dan mengarahkan kembali ke halaman edit domain dengan pesan error.	Sesuai
18.	Hapus domain	Admin klik tombol hapus pada kolom domain.	Data domain berhasil di hapus.	Sesuai
19.	Daftar vendor	Admin klik tombol vendor pada navigasi.	Menampilkan halaman vendor yang menampilkan seluruh data vendor.	Sesuai
20.	Tambah vendor	Admin mengisi form di halaman tambah vendor.	Data vendor berhasil di tambahkan dan mengarahkan ke halaman daftar vendor.	Sesuai
21.	Edit vendor	Admin mengisi form di halaman edit vendor.	Data vendor berhasil diubah dan mengarahkan ke halaman daftar vendor.	Sesuai
22.	Hapus vendor	Admin klik tombol hapus pada kolom vendor.	Data vendor berhasil di hapus.	Sesuai
23.	Daftar paket	Admin klik tombol package pada navigasi.	Menampilkan halaman paket yang menampilkan seluruh data paket.	Sesuai
24.	Tambah paket	Admin mengisi form di halaman tambah paket.	Data paket berhasil di tambahkan dan mengarahkan ke halaman daftar paket.	Sesuai
25.	Edit paket	Admin mengisi form di halaman edit paket.	Data paket berhasil diubah dan mengarahkan ke halaman daftar paket.	Sesuai
26.	Hapus paket	Admin klik tombol hapus pada kolom paket.	Data paket berhasil di hapus.	Sesuai
27.	Daftar template	Admin klik tombol template pada navigasi.	Menampilkan halaman template yang menampilkan seluruh data template.	Sesuai
28.	Tambah template	Admin mengisi form di halaman tambah template.	Data template berhasil di tambahkan dan mengarahkan ke halaman daftar template.	Sesuai

29.	Edit template	Admin mengisi form di halaman edit template.	Data template berhasil diubah dan mengarahkan ke halaman daftar template.	Sesuai
30.	Hapus template	Admin klik tombol hapus pada kolom template.	Data template berhasil di hapus.	Sesuai
31.	Admin dashboard todolist	Admin klik tombol todolist pada navigasi.	Menampilkan dashboard todo list.	Sesuai
32.	Tambah todolist admin	Admin pilih user lalu klik add task, kemudian mengisi form yang dibutuhkan.	Data todo list berhasil disimpan sesuai dengan user yang dipilih.	Sesuai
33.	Admin Report todolist	Admin pilih user lalu klik report.	Menampilkan laporan todo list sesuai dengan user yang dipilih.	Sesuai
34.	Daftar transaksi	Admin klik tombol order pada navigasi.	Menampilkan halaman order yang menampilkan seluruh data transaksi.	Sesuai
35.	Edit status transaksi	Admin klik tombol progress pada kolom order, lalu mengisi form yang dibutuhkan.	Status transaksi berhasil diubah.	Sesuai
36.	Support Dashboard	Support klik tombol dashboard pada navigasi.	Menampilkan halaman dashboard untuk role support.	Sesuai
37.	Daftar todolist	Support klik tombol todolist pada navigasi.	Menampilkan daftar todolist sesuai dengan user yang login di halaman todolist.	Sesuai
38.	Tambah todolist	Support mengisi form di todo list di panel tambah todolist.	Data todolist berhasil ditambahkan secara langsung di halaman todolist.	Sesuai
39.	Hapus todolist	Support klik hapus pada konten todolist yang dipilih.	Data todolist berhasil dihapus secara langsung di halaman todolist.	Sesuai
40.	Buat pesanan website	Customer mengisi form di halaman pemesanan.	Menampilkan halaman checkout.	Sesuai

41.	Buat invoi cepayment gateway	Customer klik checkout di halaman checkout.	Menampilkan halaman pembayaran dari payment gateway.	Sesuai
42.	Ubah status order	Sistem melakukan pengecekan status order menggunakan callback dan return url setelah customer melakukan pembayaran.	Status order berubah sesuai dengan kondisi.	Sesuai
43.	Daftar trans aksicustomer	Customer klik pesanan saya pada navigasi.	Menampilkan halaman serta data transaksi sesuai dengan user yang login.	Sesuai

3.3. Pembahasan

Bagian ini membahas hasil penelitian pengembangan sistem manajemen layanan pelanggan berbasis *website* di CV *Asian Technology Inspiration*. Sistem ini berhasil memproses permintaan pengguna dengan cepat, meningkatkan pengalaman pengguna. Fitur seperti pengelolaan data pelanggan, pemantauan kinerja karyawan, dan manajemen keluhan dirancang sesuai spesifikasi dan diuji dengan metode *blackbox* untuk memastikan fungsionalitas tanpa *bug* signifikan.

Dibandingkan metode manual sebelumnya, sistem baru mengurangi waktu pengelolaan data, menangani keluhan lebih efisien, dan meminimalkan risiko kesalahan. Sistem ini juga meningkatkan produktivitas karyawan melalui alat manajemen tugas yang lebih baik. Data yang terorganisir mendukung pengambilan keputusan strategis, meningkatkan layanan, kepuasan pelanggan, dan pertumbuhan bisnis perusahaan.

4. KESIMPULAN

Sistem CRM yang dikembangkan mampu menggantikan metode manual, meningkatkan akurasi, kemudahan akses, dan pembaruan data pelanggan, sehingga efisiensi operasional meningkat. Fitur pemantauan kinerja karyawan membantu evaluasi efektivitas kerja dan kualitas layanan, sementara komunikasi dan koordinasi antara perusahaan, pelanggan, serta karyawan menjadi lebih efisien dan terstruktur, termasuk bagi yang bekerja di lokasi berbeda.

DAFTAR PUSTAKA

- [1] Galih, N. Alivia, and Sandi Apriandi, "Perancangan Sistem Customer Relationship Management (CRM) Berbasis Website di PT Aguna Amreta Ardaya," *TEMATIK*, vol. 10, no. 1, pp. 113–123, Jun. 2023, doi: 10.38204/tematik.v10i1.1328.
- [2] G. M. Tahir, "Penerapan CRM (Customer Relationship Management) Pada Sistem Reservasi Salon N'N Berbasis Web," *SENATIK*, pp. 461–472, 2023.
- [3] D. Silaswara, M. M. A. Kusnawan, and M. M. E. Hernawan, "Dampak Customer Relationship Management dan Kualitas Pelayanan Jasa Pengiriman terhadap Kepuasan Konsumen Marketplace di Masa Pandemi Covid 19," 2022.
- [4] R. Y. Endra, Y. Aprilinda, Y. Y. Dharmawan, and W. Ramadhan, "Analisis Perbandingan Bahasa Pemrograman PHP Laravel dengan PHP Native pada Pengembangan Website," *EXPERT: Jurnal Manajemen Sistem Informasi dan Teknologi*, vol. 11, no. 1, p. 48, Jun. 2021, doi: 10.36448/expert.v11i1.1202.
- [5] A. H. Prasetyo, A. U. Hamdani, and D. Anubhakti, "Penerapan Electronic Customer Relationship Management Berbasis Framework of Dynamic CRM Pada Toko Clashy

- Implementation of Electronic Customer Relationship Management based on the Framework of Dynamic CRM At Toko Clashy,” 2023. [Online]. Available: <https://jurnaldrpm.budiluhur.ac.id/index.php/Kresna/>
- [6] Lasimin, A. Haq, and Verry, “INFORMATION MANAGEMENT FOR EDUCATORS AND PROFESSIONALS Sistem Informasi Penggajian PT. Kalisha Utama Ghani Cilacap Menggunakan Framework Laravel,” *INFORMATION MANAGEMENT FOR EDUCATORS AND PROFESSIONALS*, vol. 4, no. 2, pp. 153–162, 2020.
 - [7] D. M. Ramdani and R. Sanjaya, “SISTEM INFORMASI MANAJEMEN PEMESANAN DAN PRODUKSI UMKM KONVEKSI DI CV. IDSOPIS,” *eProsiding Teknik Informatika (PROTEKTIF)*, vol. Vol. 1, no. 01, pp. 228–236, 2020.
 - [8] A. Noviantoro, A. B. Silviana, R. R. Fitriani, and H. P. Permatasari, “RANCANGAN DAN IMPLEMENTASI APLIKASI SEWA LAPANGAN BADMINTON WILAYAHDEPOK BERBASIS WEB,” *Jurnal Teknik dan Science*, vol. 1, no. 2, pp. 88–103, 2022.
 - [9] Y. D. Arimbi, D. Kartinah, A. Nila, and W. Della, “RANCANGAN SISTEM INFORMASI KOST PUTRI MALIKA BERBASIS WEBSITE MENGGUNAKAN FRAMEWORK LARAVEL DAN MYSQL,” *Jurnal Ilmiah Multidisiplin*, vol. 1, no. 3, pp. 93–103, 2022, [Online]. Available: <https://www.teamstart.my.id/>.
 - [10] A. Purnamawati and R. T. Prasetyo, “Sistem Informasi Penjualan Dan Pelelangan Pada Cv Java Ombus Dengan Metode Waterfall,” *Journal Computer Science*, vol. 1, no. 1, 2022.
 - [11] T. M. Sinaga, A. Zuhdi, and G. B. Santoso, “MVC Implementation in Laravel Framework for Development Web-Based E-Commerce Applications,” *Intelmatix*, pp. 37–42, 2021, doi: 10.24018/ejece.2021.6.4.448.
 - [12] A. S. Sari and R. Hidayat, “Designing website vaccine booking system using golang programming language and framework react JS,” *Journal of Information System, Informatics and Computing Issue Period*, vol. 6, no. 1, pp. 22–39, 2022, doi: 10.52362/jisicom.v6i1.760.
 - [13] J. Shadiq, A. Safei, R. Wahyudin Ratu Loly, C. sitasi, L. Rwr, and P. Aplikasi Peminjaman Kendaraan Operasional Kantor Menggunakan BlackBox Testing, “INFORMATION MANAGEMENT FOR EDUCATORS AND PROFESSIONALS Pengujian Aplikasi Peminjaman Kendaraan Operasional Kantor Menggunakan BlackBox Testing,” *Information Management for Educators and Professionals*, vol. 5, no. 2, pp. 97–110, 2021.