

Pengembangan Aplikasi Perawatan Sepatu Berbasis Website

Asroni^{1✉}, Dede Chandra², Cahya Damarjati³
¹⁻³Universitas Muhammadiyah Yogyakarta, Indonesia

✉Corresponding Author: asroni@umy.ac.id

ABSTRAK

Vist Clean adalah layanan perawatan sepatu yang bertujuan untuk membersihkan, merawat, dan memperbaiki sepatu pelanggan dengan teknik khusus. Namun, minimnya digitalisasi menyebabkan efisiensi operasional rendah dan banyak kesalahan, seperti pencatatan transaksi manual yang menyulitkan rekap data. Selain itu, pelanggan harus mengecek status pengerjaan secara manual karena durasi perawatan bergantung pada antrean dan kondisi sepatu. Untuk mengatasi hal ini, dikembangkan sistem informasi berbasis online yang memungkinkan pemesanan layanan, pencatatan transaksi digital, dan pemantauan status pengerjaan. Dengan sistem ini, transaksi tercatat otomatis, dan pelanggan dapat memeriksa status laundry sepatu secara mandiri. Pengembangan sistem menggunakan metode Waterfall SDLC, sementara analisis dan perancangan dilakukan dengan UML. Sistem ini dikembangkan menggunakan PHP (Laravel), Bootstrap, dan MySQL. Fitur utama mencakup pemesanan layanan dengan laporan yang dapat diunduh serta pelacakan status pengerjaan sepatu.

Kata Kunci: sistem informasi vist clean, PHP, laravel, bootstrap, MySQL, SDLC

A. Latar belakang

Dalam masyarakat yang semakin sibuk dan padat, banyak individu tidak memiliki waktu atau keterampilan untuk membersihkan sepatu mereka secara tepat. Walaupun sepatu merupakan elemen esensial dari penampilan dan citra diri individu, perawatannya sering kali diabaikan akibat kesibukan sehari-hari. Sehubungan dengan itu, terdapat peningkatan permintaan untuk layanan laundry sepatu yang efisien dan dapat diandalkan. Vist Clean merupakan salah satu industri laundry sepatu yang berlokasi di Sonopakis Lor, Ngestiharjo, Kasihan Bantul. Vist Clean adalah layanan khusus yang ditujukan untuk membersihkan, merawat, dan memperbaiki sepatu pelanggan. Walaupun sering dianggap sebagai layanan premium, penting untuk menyadari bahwa pencucian sepatu memiliki manfaat signifikan dalam perawatan sepatu. Walaupun layanan laundry sepatu telah memberikan estimasi waktu pengerjaan, proses pembersihan dan perawatan sepatu berkualitas tinggi memerlukan waktu yang bervariasi untuk setiap pasang sepatu. Durasi pengerjaan tergantung pada jumlah antrian, tingkat kontaminasi, atau kerusakan awal sebelum perawatan dilakukan. Ini yang menimbulkan ketidakpastian mengenai durasi proses perawatan sepatu.

Dalam industri laundry sepatu, Vist Clean masih mencatat data transaksi pelanggan secara manual dalam buku. Selain itu, pencatatan transaksi manual menggunakan buku memiliki risiko kehilangan atau kerusakan akibat berbagai peristiwa yang tidak diinginkan, seperti kebakaran, banjir, atau kecelakaan lainnya. Ini dapat mengakibatkan kehilangan data yang tidak dapat dipulihkan. Penelitian sebelumnya menjelaskan bahwa peran sistem informasi dalam dunia bisnis sangat krusial untuk menyampaikan informasi kepada konsumen mengenai produksi yang ditawarkan, dengan dapat berupa gambar, teks, dan

lainnya. Penerapan system informasi dalam bisnis memfasilitasi akses informasi yang diperlukan tanpa kehadiran fisik di toko, serta dapat dimanfaatkan untuk pembelajaran, promosi, penjualan, dan pelayanan kepada pengguna. Oleh karena itu, system informasi dirancang untuk memfasilitasi penyampaian data dengan cepat saat diperlukan. Sistem informasi dapat memfasilitasi proses transaksi pemesanan, pengiriman, pendaftaran pelanggan baru, dan penyajian laporan transaksi untuk pengusaha [1].

Penelitian sebelumnya menegaskan bahwa pengecekan status laundry sepatu melalui aplikasi chatting WhatsApp secara personal, atau dengan mengharuskan konsumen datang langsung ke penyedia jasa untuk menanyakan status laundry, jelas tidak efektif dan efisien. Oleh karena itu, diperlukan suatu system untuk memeriksa status laundry sepatu [2]. Selanjutnya, penelitian lain menjelaskan bahwa Framework Laravel memiliki keunggulan yang membedakannya dari framework lain, yaitu kinerja yang lebih cepat, stabilitas dalam memuat data, keamanan data, penggunaan fitur canggih seperti Blade dengan konsep HMVC (*Hierarchical Model View Controller*), ketersediaan pustaka yang siap pakai, serta fitur pengelolaan migrasi untuk pembuatan skema tabel dalam basis data [3]. Oleh karena itu, dapat disimpulkan bahwa penerapan framework Laravel memiliki potensi signifikan untuk pengembangan sistem informasi layanan dalam industry perawatan sepatu Vist Clean.

Dari latar belakang tersebut, dapat disimpulkan bahwa belum terdapat system informasi layanan perawatan sepatu berbasis website di Vist Clean. Pemanfaatan catatan tulisan tangan untuk mendokumentasikan data transaksi pelanggan di Vist Clean dapat mengakibatkan berkurangnya akurasi informasi. Ketidakpastian durasi penyelesaian dalam layanan laundry sepatu, yang dipengaruhi oleh faktor antrian, tingkat kotoran, dan kerusakan awal, berdampak pada kepuasan dan kenyamanan pelanggan.

Dalam menjalankan bisnis, selain strategi pemasaran yang baik, fasilitas pendukung seperti website sistem informasi sangat penting. Website membantu memasarkan produk dan jasa, memberikan informasi lengkap, serta menyediakan proses transaksi seperti pemesanan, pengiriman, dan laporan transaksi. Website juga memperlihatkan eksistensi usaha tersebut.

Industri laundry sepatu, yang meliputi cuci, perbaikan, dan pengecatan sepatu, memerlukan teknologi digital, salah satunya sistem informasi berbasis website untuk transaksi online. Sistem ini meningkatkan keamanan data, mempercepat pencatatan, dan mengurangi kesalahan manusia [4].

Namun, industri ini menghadapi masalah waktu pengerjaan, dan estimasi waktu yang tepat penting untuk pengalaman pelanggan. Sistem pelacakan status pengerjaan sepatu juga dibutuhkan untuk memantau proses cuci sepatu.

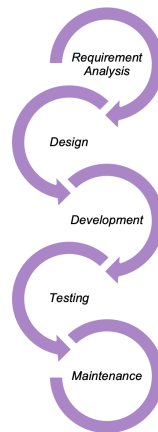
Sebelumnya, pengelolaan data di Jasa Laundry Sepatu Queen Shoes Cleaning dilakukan secara manual, yang tidak efisien dan sering terjadi kesalahan. Oleh karena itu, diperlukan sistem informasi berbasis web untuk mempercepat pencatatan transaksi dan pengelolaan data [5].

Sistem ini akan dikembangkan menggunakan framework Laravel untuk pengelolaan informasi keuangan yang lebih baik dan minimisasi kesalahan. Laravel memiliki kelebihan seperti CLI Artisan, PHP Composer, dan kode program yang rapi dan mudah dipahami [6].

Penulis akan mengembangkan sistem informasi berbasis website untuk Vist.Clean, yang sebelumnya belum diterapkan. Sistem ini akan menggantikan pencatatan transaksi manual menjadi digital, dengan fitur pencatatan transaksi harian, laporan bulanan, tracking status pengerjaan sepatu, serta fitur pick up dan delivery untuk memudahkan pelanggan [7].

B. Metode

Metode yang digunakan dalam perancangan dan pengembangan sistem informasi ini adalah metode *Waterfall* dalam *Software Development Life Cycle* (SDLC). *SDLC Waterfall* sering digunakan dalam pembuatan sistem perangkat lunak, termasuk sistem informasi, dengan tujuan menghasilkan kualitas sistem yang tinggi, biaya yang efisien, serta memastikan bahwa sistem berjalan sesuai dengan ekspektasi pengguna [8]. Terdapat lima tahapan dalam metode *waterfall*, yang digambarkan pada Gambar 1 berikut:



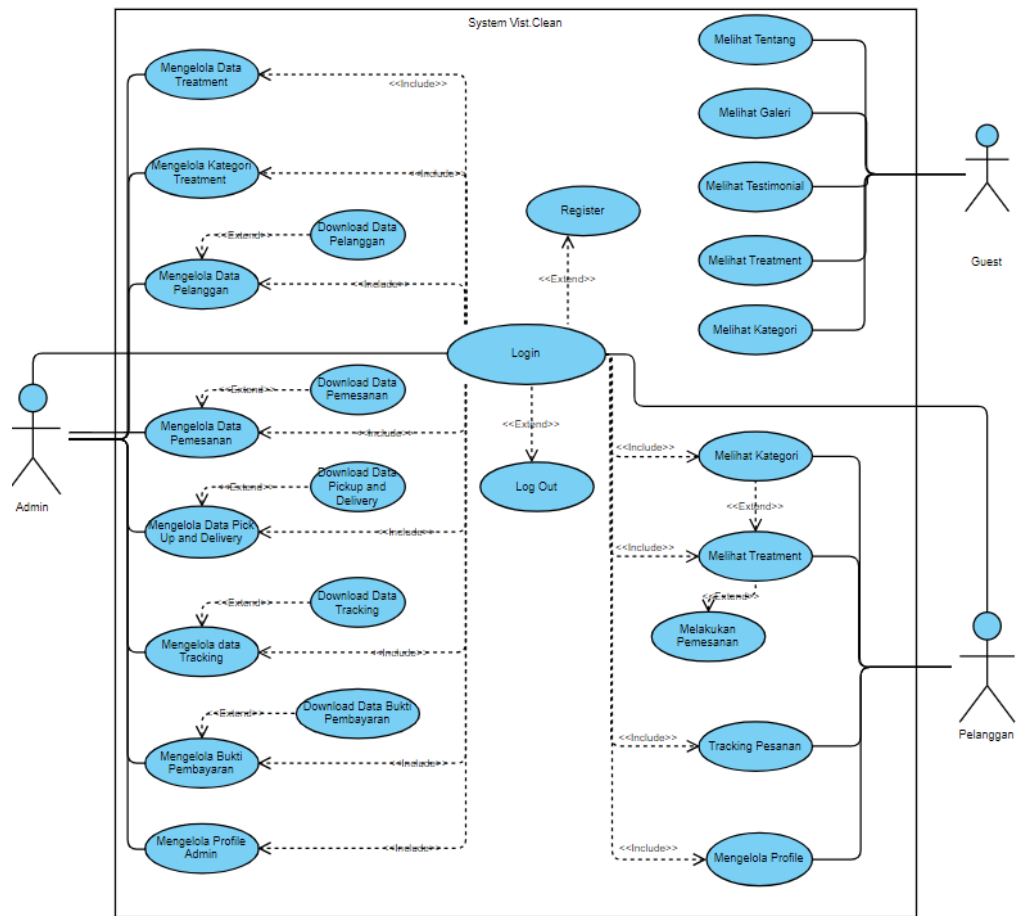
Gambar 1. Metode penelitian *waterfall* Vist.Clean

a. Requirements

Tahap pertama adalah analisis kebutuhan yang berkaitan dengan pembuatan website. Pada tahap ini, dilakukan pengumpulan informasi terkait rancangan sistem dan data yang diperlukan dalam pengembangan website Vist.Clean. Hasil dari analisis ini akan digunakan untuk merancang sistem dan menghasilkan kebutuhan pengguna (*user requirements*).

b. Design

Pada tahap kedua, dilakukan perancangan sistem berdasarkan kebutuhan yang telah ditetapkan. Tahapan ini mencakup desain website, arsitektur informasi, serta pemilihan teknologi yang akan digunakan, seperti bahasa pemrograman, *library*, dan *software* lainnya. Hasil dari tahap ini adalah *software requirement* yang menggambarkan kebutuhan teknis sistem. Desain yang dilakukan diantaranya berupa *use case diagram*, *entity relational diagram*, dan *class diagram* [9].

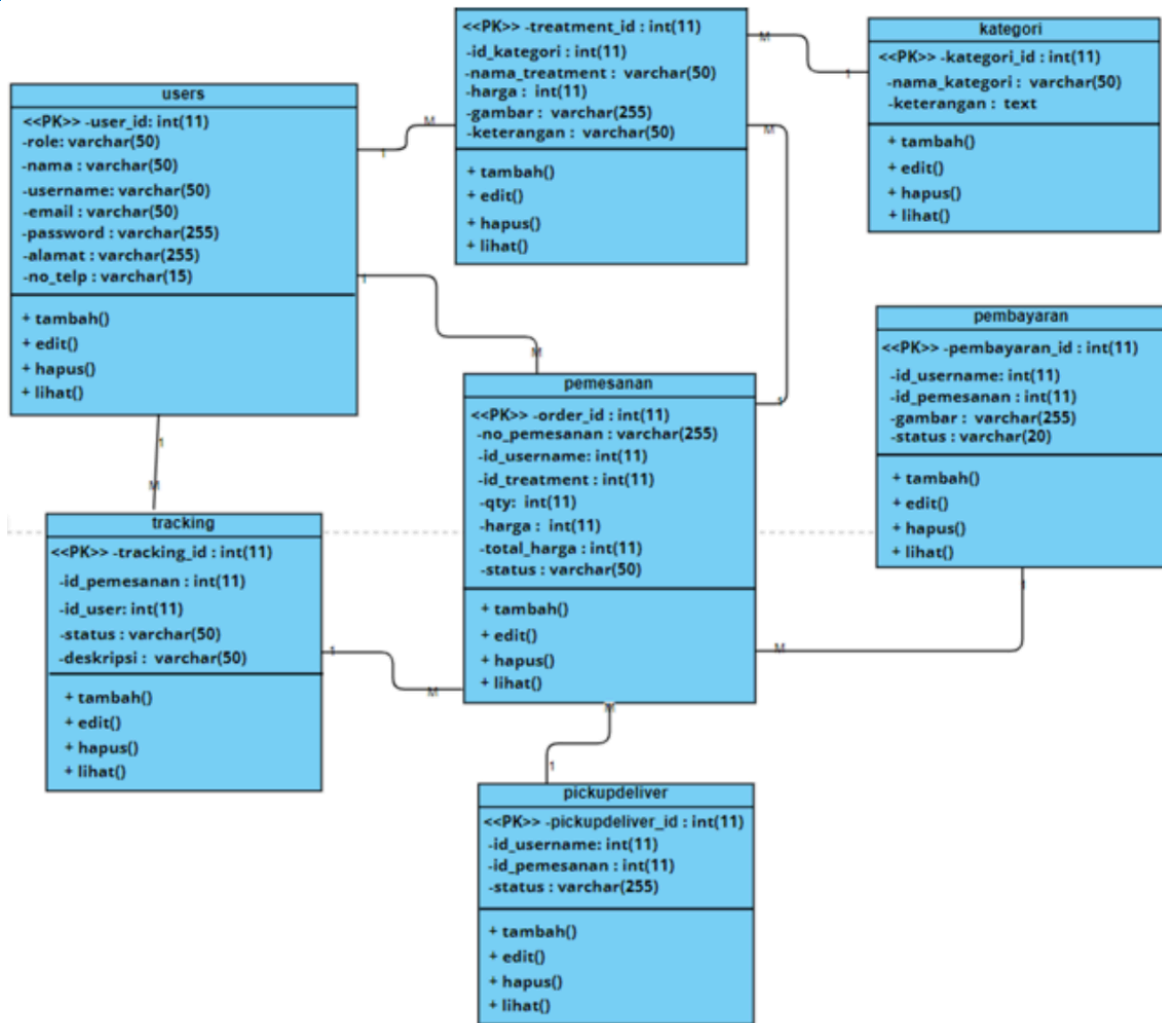


Gambar 2. Rancangan Use Case Diagram Vist.Clean

Gambar 2 merupakan rancangan *Use Case Diagram* Vist Clean yang memiliki tiga aktor Admin, Pelanggan, dan *Guest*. Aktor Admin memiliki banyak fitur untuk mengelolanya, salah satu contoh diantaranya adalah fitur *Treatment* yaitu admin dapat menambah, mengubah, serta menghapus *Treatment* dengan syarat melakukan *login* terlebih dahulu. Contoh lainnya Admin dapat mengelola fitur Kategori *Treatment*, *Tracking*, bukti pembayaran, dan profile. Selanjutnya aktor Pelanggan dapat mengakses fitur Register, *Treatment*, Pemesanan, *Tracking*, *Profil*, sedangkan untuk aktor *Guest* dapat melihat Fitur Tentang Vist Clean, Galeri Vist Clean, *Testimoni* Vist Clean, *Treatment*, *Treatment* dengan hak akses terbatas dengan tidak melakukan *login*.

c. Development

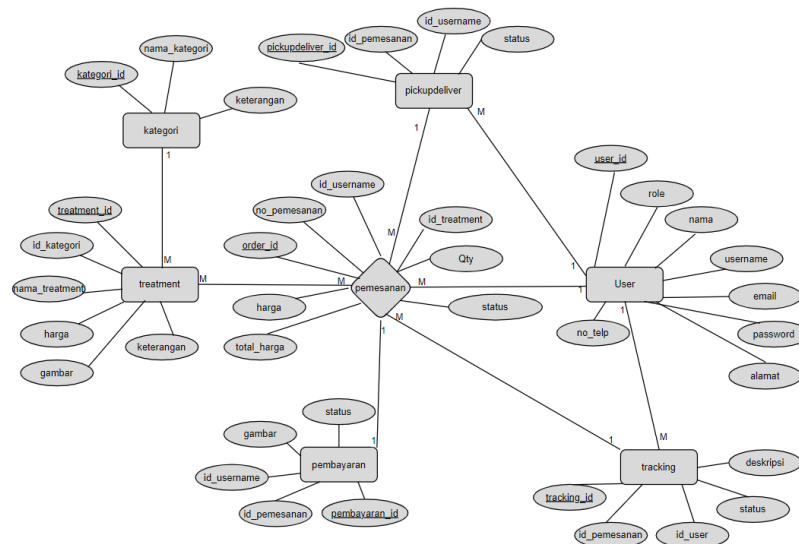
Tahap ketiga adalah pengembangan sistem, di mana desain yang telah disetujui diimplementasikan menjadi website melalui proses pemrograman. Pada tahap ini, website akan dikembangkan menjadi bentuk yang dapat diuji dan digunakan. Pengembangan memiliki dua acuan yaitu *Class Diagram* untuk dasar pembuatan *coding*, dan ERD untuk acuan pembuatan *Database* [10].



Gambar 3. Class Diagram (CD) Vist.Clean

Gambar 3 merupakan *Class Diagram* (CD) untuk sistem informasi data penjualan pada Vist.Clean sebagai berikut: *Class User* dua role yaitu Admin dan Pelanggan, pada bagian Admin berfungsi sebagai menambah, melihat, mengubah, dan menghapus *Treatment*, *Kategori*, *Pesanan*, *Bukti Pembayaran*, *Data Pickup Delivery*. serta dapat melakukan *Login* dan *Logout*. Pada bagian Pelanggan berfungsi sebagai sebagai menambah, melihat, mengubah, dan menghapus *Treatment* yang di pesan, melihat pemesanan dan *Tracking*. *Class Kategori* pada *Class kategori* berfungsi sebagai tambah, edit, hapus dan lihat data kategori yang dibuat oleh *Class user role admin*. *Class kategori* memiliki relasi *one to many* dengan *Class Treatment*. Yang artinya satu kategori dapat terisi banyak *Treatment*. *Class Treatment* pada *class Treatment* berfungsi sebagai tambah, edit, hapus dan lihat Data *Treatment* yang dibuat oleh *Class user role admin*. *Class Treatment* memiliki relasi *many to many* dengan *Class Pemesanan* Yang artinya banyak *Treatment* dapat dilakukan banyak *Pesanan*. *Class Tracking* pada *Class Tracking* berfungsi sebagai edit, hapus dan lihat Data *Tracking*. *Class Tracking* memiliki relasi *one to many* dengan *Class user*, yang artinya satu *user* dapat melakukan banyak *Tracking*. *Class Pickup Delivery* pada *Class Pickup Delivery* berfungsi sebagai edit, hapus dan lihat Data *Pickup Delivery*. *Class Pickup Delivery* memiliki relasi *one to many* dengan *Class pemesanan*,

yang artinya satu *Pickup Delivery* dapat dilakukan untuk banyak Pemesanan. *Class* Bukti Pembayaran pada *Class* Bukti Pembayaran berfungsi sebagai tambah, edit, hapus dan lihat Data Bukti Pembayaran. *Class* Bukti Pembayaran *one to one* dengan *Class* pemesanan, yang artinya satu bukti pembayaran hanya dapat dilakukan untuk satu pemesanan. *Class* Pemesanan pada *Class* Pemesanan berfungsi tambah, tambah, edit, hapus dan lihat data Pemesanan. *Class* Pemesanan *one to one* dengan *Class* pembayaran, yang artinya satu pemesanan hanya dapat dilakukan satu kali pembayaran.



Gambar 4. Entity Relational Diagram (ERD) Vist.Clean

Gambar 4 merupakan rancangan *Entity Relational Diagram* (ERD) untuk membangun aplikasi Vist.Clean yang memiliki 7 entitas yang terdiri dari: Entitas Kategori mempunyai relasi *one to many* dengan entitas *Treatment*, yang artinya satu kategori terdapat banyak jenis *Treatment*, dilanjutkan dengan entitas *Treatment*, pemesanan, User, pemesanan, *Tracking*, *Pickup Delivery*, dan Bukti Pembayaran [11].

d. Testing

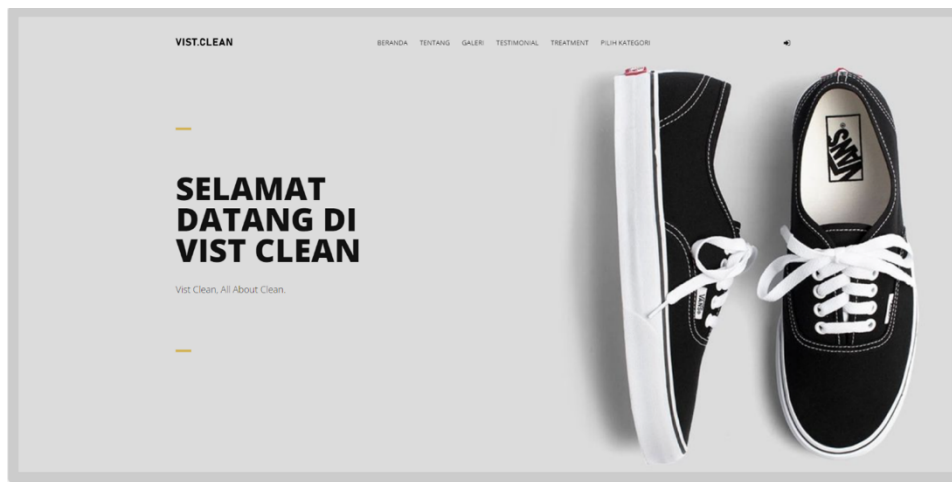
Tahap keempat adalah pengujian sistem. Pengujian ini mencakup pengintegrasian sistem dan uji kelayakan website, baik dari sisi fungsionalitas maupun *user acceptance testing* (UAT). Tujuan dari tahap ini adalah untuk memastikan bahwa website berjalan sesuai dengan yang diharapkan. Kemudian untuk mengukur kegunaan situs web dilakukan dengan metode *System Usability Scale* (SUS) [12].

e. Maintenance

Tahap terakhir adalah pemeliharaan. Pada tahap ini, dilakukan penginstalan perangkat lunak pada website dan pengujian aplikasi. Pemeliharaan bertujuan untuk memastikan website tetap berjalan dengan lancar dan dapat digunakan oleh pengguna dalam jangka waktu yang ditentukan.

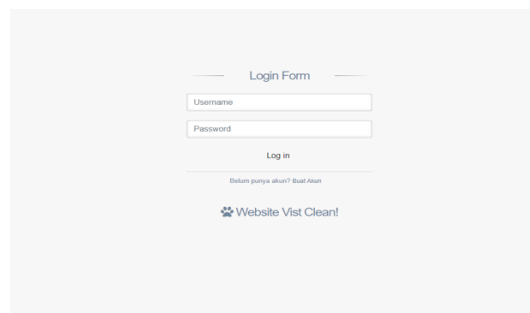
C. Hasil dan Pembahasan

Hasil implementasi hasil dari perancangan sistem laundry sepatu berbasis web pada Vist Clean menggunakan *framework* Laravel sudah berhasil diimplementasikan dengan proses pengujian tampilan baik dari sisi *login* sebagai Admin, Pelanggan, dan *Guest*. Berikut ini beberapa hasil tampilan halaman web. Halaman *Beranda* adalah halaman utama ketika website vist clean dibuka, Halaman *Beranda* dapat menampilkan halaman tentang vist clean, galeri, testimonial, *treatment*, kategori, tidak diperlukan login user untuk melihat halaman ini seperti Gambar 5.



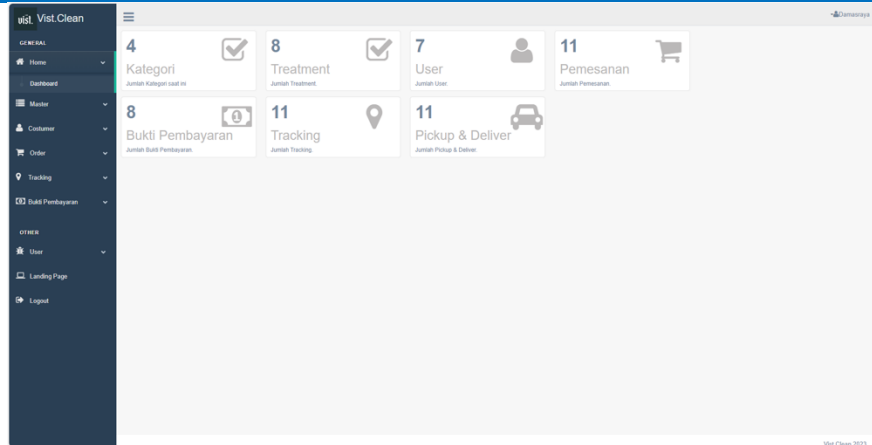
Gambar 5. Halaman Beranda

Halaman *login* setiap user harus memasukkan *username* yang benar dan juga *password* yang benar. Jika salah maka akan dikembalikan Kembali untuk memasukan *username* dan *password* seperti Gambar 6.



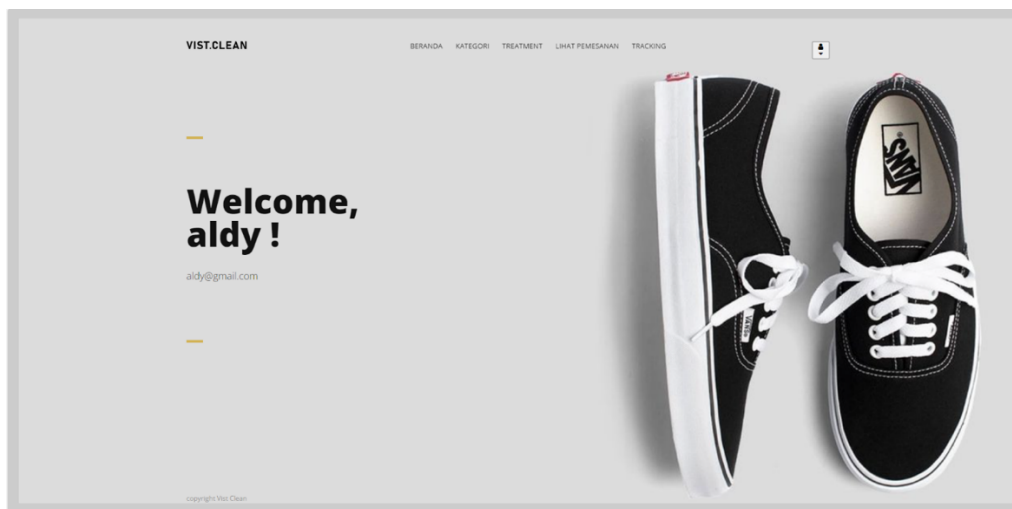
Gambar 6. Halaman Login

Halaman *dashboard admin* ketika berhasil melakukan *Login admin*, maka akan otomatis memasuki halaman *Dashboard admin*. Tampilan *Dashboard admin* dapat dilihat pada Gambar 7.



Gambar 7. Halaman Dashboard Admin

Halaman *Dashboard* pelanggan ketika berhasil melakukan *Login* pelanggan, maka akan otomatis memasuki halaman *Dashboard* pelanggan. Tampilan *Dashboard* pelanggan dapat dilihat pada Gambar 8.



Gambar 8. Halaman Dashboard Pelanggan

Pengujian berikutnya adalah terkait pengujian sistem di mana perangkat lunak dijalankan dan dievaluasi untuk memastikan kesesuaiannya dengan spesifikasi sistem yang telah ditentukan dan beroperasi dalam lingkungan yang diinginkan. Pengujian sistem seringkali berkaitan dengan identifikasi dan perbaikan bug, kesalahan, serta kekurangan dalam program, sehingga dapat menghindari kegagalan dalam pelaksanaan perangkat lunak tersebut. Proses pengujian ini melibatkan pengujian setiap tahap dan potensi kesalahan yang dapat terjadi dalam setiap tahapan. Pengujian ini mengadopsi metode pengujian kotak hitam (*black box testing*), yang berfokus pada aspek fungsional dari perangkat lunak. Pengujian ini melibatkan definisi serangkaian kondisi input dan melaksanakan pengujian terhadap spesifikasi fungsional program. Berikut adalah komponen yang diperlukan dalam pelaksanaan proses pengujian ini: Menyiapkan komputer atau laptop sesuai spesifikasi yang ditentukan, Menjalankan aplikasi

yang dibutuhkan, Melakukan proses pengujian, dan Mencatat hasil pengujian yang telah dilakukan

Tabel 1. Pengujian *Login*

No	Rencana Pengujian	Yang di Uji	Hasil Yang diharapkan	Status
1.	<i>Login tanpa username dan password</i>	Kosongkan <i>username</i> dan <i>password</i> kemudian klik ' <i>login</i> '	Tidak dapat melakukan ' <i>login</i> ' dan diharuskan memasukan <i>username</i> dan <i>password</i> .	<i>Valid</i>
2.	<i>Login tanpa username.</i>	Memasukan <i>password</i> tanpa memasukan <i>username</i> .	Tidak dapat ' <i>login</i> ' dan dikembalikan lagi untuk memasukan <i>username</i> dan <i>password</i> .	<i>Valid</i>
3.	<i>Login tanpa password.</i>	Memasukan <i>username</i> tanpa <i>password</i> , lalu klik ' <i>login</i> '	Tidak dapat ' <i>login</i> ' dan dikembalikan lagi untuk memasukan <i>username</i> dan <i>password</i>	<i>Valid</i>
4.	<i>Login sukses</i>	Memasukan <i>username</i> dan memasukan <i>password</i> yang benar dan klik ' <i>login</i> '	Dapat mengakses ke dalam sistem.	<i>Valid</i>
5.	<i>Login Admin.</i>	Memasukan <i>username</i> dan memasukan <i>password</i> yang benar dan klik ' <i>login</i> '.	Dapat mengakses ke dalam sistem web khusus Admin	<i>Valid</i>
6.	<i>Login Pelanggan.</i>	Memasukan <i>username</i> dan memasukan <i>password</i> yang benar dan klik ' <i>login</i> '.	Dapat mengakses ke dalam sistem web khusus Pelanggan.	<i>Valid</i>

Tabel 1 menunjukkan hasil masuk ke sistem. Ada beberapa cara berbeda untuk masuk, seperti tidak memasukkan nama pengguna atau kata sandi, masuk hanya dengan satu dari dua opsi, atau masuk dengan akun yang benar (admin atau pengguna). Hasil yang diharapkan mencerminkan cara sistem harus merespons pada satu situasi. Semua kiriman ditandai sebagai *Valid*, yang berarti sistem bekerja seperti yang diharapkan.

Tabel 2. Pengujian Sistem untuk Admin

No	Rencana Pengujian	Yang Di Uji	Hasil Yang Diharapkan	Status
1.	<i>Login</i>	Memasukan <i>username</i> dan <i>password</i> yang benar	Dapat memasuki halaman utama <i>website</i> Admin.	<i>Valid</i>
2.	Membuat <i>User</i>	Membuat <i>User</i> untuk pelanggan yang dilakukan Admin.	Dapat membuat <i>user</i> untuk pelanggan agar dapat mengakses halaman aplikasi <i>web</i>	<i>Valid</i>
3.	<i>Input Data</i>	Melakukan proses <i>input</i> Data	Dapat melakukan dan memproses <i>input</i> data.	<i>Valid</i>
4.	Hapus Data	Melakukan proses hapus Data	Dapat melakukan dan memproses hapus data	<i>Valid</i>
5.	Edit Data	Melakukan proses edit Data.	Dapat melakukan dan memproses hapus data.	<i>Valid</i>
6.	Ekspor Data	Mendownload Laporan.	Dapat melakukan dan	<i>Valid</i>

No	Rencana Pengujian	Yang Di Uji	Hasil Yang Diharapkan	Status
			memproses ekspor data.	
7.	Logout	Keluar dari halaman web khusus Admin.	Dapat keluar dari halaman web Admin	Valid

Tabel 2 menunjukkan hasil masuk ke sistem untuk Admin, semua kiriman ditandai sebagai *Valid*, yang berarti sistem bekerja seperti yang diharapkan

Tabel 3. Pengujian Sistem untuk Pelanggan

No	Rencana Pengujian	Yang Di Uji	Hasil Yang Diharapkan	Status
1.	Register	Membuat User untuk pelanggan yang dilakukan pelanggan.	Dapat membuat <i>user</i> untuk pelanggan agar dapat mengakses halaman aplikasi web	Valid
2.	Login	Memasukan <i>username</i> dan <i>password</i> yang benar	Dapat memasuki halaman utama <i>website</i> Admin.	Valid
3.	Melakukan pemesanan	Melakukan pemesanan pada <i>treatment</i> yang tersedia, Upload bukti pembayaran. Melihat pesanan yang sudah berhasil	Dapat melakukan pemesanan, Dapat melakukan upload bukti pembayaran, Dapat Melihat pesanan berhasil dibuat.	Valid
4.	Tracking Pesanan	Melakukan proses pengecekan <i>tracking</i> pesanan.	Dapat melakukan proses pengecekan status pesanan.	Valid
5.	Edit Profile	Melakukan proses edit <i>profile</i> .	Dapat melakukan dan memproses edit <i>profile</i> .	Valid
6.	Logout	Keluar dari halaman web khusus pelanggan	Dapat keluar dari halaman web pelanggan.	Valid

Tabel 3 menunjukkan hasil masuk ke sistem untuk Pelanggan, semua kiriman ditandai sebagai *Valid*, yang berarti sistem bekerja seperti yang diharapkan. Pengujian selanjutnya untuk mengukur kebergunaan web dengan melakukan pengumpulan data, peneliti menggunakan *System Usability Scale (SUS) Score*. Peneliti akan melakukan *testing* website kepada *user* admin maupun pelanggan Vist Clean dengan cara melakukan penjelasan terlebih dahulu mengenai website tersebut dan kemudian mempersilahkan *user* untuk melakukan *testing*. Setelah melakukan *testing website*, peneliti akan memberikan kuisisioner tentang kebergunaan website kepada *user*. Kebergunaan produk akan dapat diukur melalui kuisisioner tersebut setelah diolah dengan rumus *System Usability Scale (SUS) Score*. Penilaian skor pada kuisisioner berisi pertanyaan dengan paramater sangat setuju (5), setuju (4), netral (3), tidak setuju (2), sangat tidak setuju (1). Pertanyaan kuesioner *system usability scale* pun perlu disusun secara berurutan. Tabel 4 menunjukkan data pengisian terhadap *user*.

Tabel 4. Data Pengisian Pertanyaan Kuisisioner *System Usability Scale (SUS) Score*

No	Pertanyaan	Skor				
		STS	TS	N	S	SS
1	Saya pikir bahwa saya akan menginginkan lebih sering menggunakan website ini untuk	0	0	7	9	5

No	Pertanyaan	Skor				
		STS	TS	N	S	SS
	melakukan pemesanan.					
2	Saya menemukan bahwa website ini, tidak harus dibuat serumit ini.	3	13	4	1	
3	Saya pikir dengan adanya website ini akan mudah untuk digunakan dalam pencatan pemesanan.	0	0	2	10	9
4	Saya pikir bahwa saya akan membutuhkan bantuan dari orang teknis untuk dapat menggunakan website ini.	5	12	4	0	0
5	Saya menemukan berbagai fungsi seperti tracking pemesanan sangat berguna untuk mengetahui status pengerjaan sepatu.	0	0	3	14	4
6	Saya pikir ada terlalu banyak ketidakkonsistenan dalam website ini.	2	11	8	0	0
7	Saya bayangkan bahwa kebanyakan orang akan mudah untuk mempelajari website ini dengan sangat cepat.	0	0	1	19	1
8	Saya menemukan, website ini sangat rumit untuk digunakan.	0	13	5	2	1
9	Saya merasa sangat percaya diri untuk menggunakan website ini.	0	0	6	12	3
10	Saya perlu belajar banyak hal sebelum saya bisa memulai menggunakan website.	4	6	10	0	1

Proses selanjutnya dengan cara menghitung hasil pengukuran *system usability scale* yaitu: Setiap pertanyaan pada urutan ganjil kurangi dengan nilai satu. Contoh pertanyaan nomor 1 memiliki skor 4, maka kurangi 4 dengan 1 sehingga skor pertanyaan 1 adalah 3. Untuk setiap pertanyaan pada urutan genap kurangi nilainya dari lima. Contoh pertanyaan 2 memiliki skor 1. Maka kurangi 5 dengan 1 sehingga skor pertanyaan 2 adalah 4. Tambahkan nilai-nilai dari pernyataan bernomor genap dan ganjil. Kemudian hasil penjumlahan tersebut dikalikan dengan 2,5. Adapun Kategori penilaian seperti pada Tabel 5.

Tabel 5. Kategori penilaian *System Usability Scale (SUS) Score*

Sus Score	Grade	Adjectiv Rating
>80.3	A	Excellent
68-80.3	B	Good
68	C	Okay
51-68	D	Poor
<51	F	Awful

Tabel 6. Skor Hasil Hitung Data

Skor Hasil Hitung Data										Nilai Jumlah (Jumlah x 2,5)
Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	
3	3	3	3	3	2	3	1	4	4	29
										72,5

Skor Hasil Hitung Data										Jumlah	Nilai (Jumlah x 2,5)
Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10		
4	4	4	3	3	2	3	3	3	3	32	80
2	3	3	4	4	2	3	1	2	3	27	67,5
3	3	4	4	2	3	3	3	3	4	32	80
3	3	4	4	4	3	3	2	3	2	31	77,5
2	3	4	3	2	2	3	3	3	2	27	67,5
2	3	3	3	3	3	3	3	3	2	28	70
4	4	3	3	3	4	3	3	2	3	32	80
4	3	2	3	3	3	3	3	2	2	28	70
2	3	3	3	3	4	3	3	2	3	29	72,5
4	3	3	3	3	3	3	3	3	4	32	80
4	2	4	2	4	2	4	3	4	2	31	77,5
3	1	4	3	3	3	3	3	3	2	28	70
2	3	3	2	3	3	2	2	3	0	23	57,5
3	3	3	3	3	3	3	2	3	3	29	72,5
2	3	4	3	3	3	3	3	3	2	29	72,5
3	3	2	4	4	3	3	3	2	2	29	72,5
3	2	3	2	3	2	3	2	4	2	26	65
2	2	4	2	2	2	3	0	2	3	22	55
3	4	4	3	3	3	3	3	3	4	33	82,5
3	2	3	4	3	2	3	2	3	2	27	67,5
Jumlah										1.510	

Tahapan berikutnya dari Tabel 6 kategori penilaian di atas peneliti mendapatkan hasil bahwa website memudahkan dalam melakukan proses pencatatan pemesanan, pengecekan status laundry sepatu, dan melakukan pemesanan. Tabel 6 mendapat *grade B* atau **Good**, karena *SUS Score* diantara 72 yang berada di nilai 68-80.3. Dengan demikian web yang dibangun sudah baik untuk dipergunakan dan perlu peningkatan untuk meningkatkan ke level berikutnya.

D. Simpulan

Berdasarkan hasil dari penelitian yang telah dilakukan oleh penulis dapat disimpulkan bahwa dengan sudah dibuatnya sistem informasi layanan perawatan sepatu berbasis website pada Vist Clean menggunakan *framework* Laravel menunjukkan proses pemesanan secara *online* dapat memudahkan admin dalam melakukan proses pengelolaan data dan laporan-

laporan pemesanan dari pelanggan. Meminimalisir kesalahan data atau kehilangan data pemesanan yang terjadi karena proses pendataan dilakukan secara *online* dan tersimpan di *database* guna mengurangi resiko kehilangan data pemesanan. Fitur *tracking* pemesanan dapat memudahkan admin dalam melakukan proses update status pengerjaan sepatu pelanggan. Sehingga pelanggan sudah bisa melacak secara *online* status pengerjaan sepatunya secara mandiri.

Acknowledgment

Penulis menyampaikan ucapan terima kasih kepada Universitas Muhammadiyah Yogyakarta atas biaya penelitian yang dilakukan.

Daftar Pustaka

- [1] D. Karmila and D. Rusda, "E-marketplace penjualan dan pemasaran barang furniture pada toko mebel menggunakan php dan mysql server," *J. Penelit. Dosen Fikom UNDA*, vol. 10, no. 1, 2019.
- [2] F. Saragih, Y. A. Dalimunthe, and H. Lubis, "Rancang Bangun Sistem Tracking Jasa Laundry Sepatu Di Clinix Shoes Care Berbasis Website," *METHOMIKA J. Manaj. Inform. Komputerisasi Akunt.*, vol. 5, no. 1, pp. 73–76, 2021.
- [3] T. B. Tahir, M. Rais, and M. A. H. Sirad, "Aplikasi Point OF Sales Menggunakan Framework Laravel," *JIKO J. Inform. Dan Komput.*, vol. 2, no. 2, pp. 55–59, 2019.
- [4] D. Nurcahyono, A. Topadang, and W. Mutiarasari, "Sistem Informasi Manajemen Shoes Care Berbasis Web," *J. Inform. Comput.*, vol. 2, no. 1, pp. 1–10, 2023.
- [5] S. N. Rakhmah and I. Rizki, "Sistem Informasi Pelayanan Jasa Laundry Sepatu Pada Queen Shoes Cleaning," *J. Kaji. Ilm.*, vol. 22, no. 1, pp. 21–34, 2022.
- [6] M. J. Budiman and N. Yudianthi, "Perancangan Sistem Informasi Pelayanan Jasa Usaha Laundry Sepatu Berbasis Progressive Web APP (PWA)," presented at the Prosiding Seminar Nasional Penelitian LPPM UMJ, 2022.
- [7] Y. Anggraini, R. Fadillah, and N. T. Suban, "Perancangan Sistem Informasi Persediaan Obat Pada Klinik Medika Prima Berbasis Web Menggunakan Metode Waterfall," *BINER J. Ilmu Komput. Tek. Dan Multimed.*, vol. 1, no. 2, pp. 87–98, 2023.
- [8] M. Z. Abdullah, M. Astiningrum, Y. Ariyanto, D. Puspitasari, and A. N. Asri, "Rancang Bangun Sistem Informasi Akuntansi Berbasis Website menggunakan Framework Laravel," *SITEKIN J. Sains Teknol. Dan Ind.*, vol. 18, no. 1, pp. 49–56, 2020.
- [9] A. F. Prasetya, S. Sintia, and U. L. D. Putri, "Perancangan Aplikasi Rental Mobil Menggunakan Diagram UML (Unified Modelling Language)," *J. Ilm. Komput. Terap. Dan Inf.*, vol. 1, no. 1, pp. 14–18, 2022.
- [10] G. B. Putra, "Rancang Bangun Sistem Penelitian Dan Pengabdian Masyarakat Universitas Bangka Belitung Berbasis Web Server," *J. Ecotipe Electron. Control Telecommun. Inf. Power Eng.*, vol. 4, no. 1, pp. 17–22, 2017.
- [11] T. Wulandari and S. Nurmiati, "Rancang Bangun Sistem Pemesanan Wedding Organizer Menggunakan Metode Rad di Shofia Ahmad Wedding," *J. Rekayasa Inf.*, vol. 11, no. 1, pp. 79–85, 2022.
- [12] R. Hermiati, A. Asnawati, and I. Kanedi, "Pembuatan E-Commerce Pada Raja Komputer Menggunakan Bahasa Pemrograman Php Dan Database Mysql," *J. Media Infotama*, vol. 17, no. 1, 2021.