

Perancangan *Website E-commerce* Soemardjan Florist Menggunakan *Framework* Laravel

Richardus Sapta Anggara¹, Magdalena A. Ineke Pakereng²

^{1,2}Program Studi Teknik Informatika, Fakultas Teknologi Informasi,
Universitas Kristen Satya Wacana

e-mail: 1672020715@student.uksw.edu, ²ineke.pakereng@uksw.edu

Abstract

Soemardjan Florist is one of the biggest florists in Salatiga, Central Java. Nearly 70% of Soemardjan Florist customers come from outside Salatiga City. Frequent found case is that customers come outside Salatiga but they sent the wreaths to weddings which is located in the Salatiga area. This proves that Soemardjan Florist has the potential to expand the market outside Salatiga city. In this research, the Soemardjan Florist website was developed to expand the Soemardjan Florist business. The website is built in PHP programming language using Laravel 5.5 framework. User test results indicate that Soemardjan Florist website can useful as a promotional media and online store as it should.

Keywords: *E-commerce, website, online shop, Laravel framework, Soemardjan Florist.*

Abstrak

Soemardjan Florist merupakan salah satu toko bunga terbesar di Kota Salatiga, Jawa Tengah. Hampir 70% pelanggan Soemardjan Florist berasal dari luar Kota Salatiga. Salah satu kasus yang sering dijumpai adalah pelanggan memesan rangkaian bunga dari luar Kota Salatiga melalui media telpon untuk dikirim ke acara pernikahan yang berada di daerah Salatiga. Hal ini membuktikan bahwa Soemardjan Florist berpotensi untuk melebarkan pasar hingga ke luar Kota Salatiga. Pada penelitian ini dilakukan pembangunan website Soemardjan Florist untuk mengembangkan usaha toko bunga Soemardjan Florist. Website dibangun dengan bahasa pemrograman PHP menggunakan framework Laravel versi 5.5. Hasil dari uji pengguna menunjukkan bahwa website Soemardjan Florist dapat berfungsi sebagai media promosi dan toko online sebagaimana mestinya.

Kata Kunci: *E-commerce, website, toko online, framework Laravel, Soemardjan Florist.*

1. PENDAHULUAN

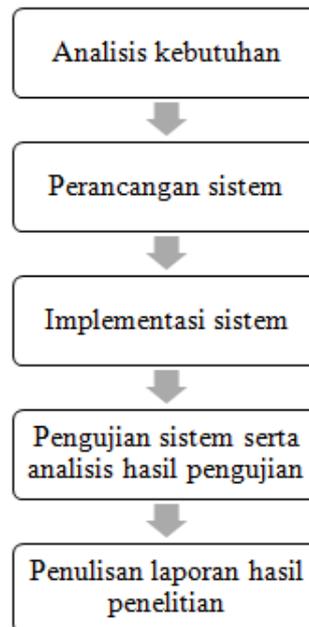
Soemardjan Florist merupakan salah satu toko bunga terbesar di Kota Salatiga, Jawa Tengah. Berdasarkan data yang didapat dari wawancara dengan pemilik Soemardjan Florist hampir 70% pelanggan Soemardjan Florist berasal dari luar Kota Salatiga. Salah satu kasus yang sering dijumpai adalah pelanggan memesan rangkaian bunga dari luar Kota Salatiga melalui media telepon untuk dikirim ke acara pernikahan yang berada di daerah Salatiga. Hal ini membuat proses transaksi antara pelanggan dan Soemardjan Florist kurang efisien, karena dalam perhitungan ongkos kirim dan pembukuannya masih dilakukan dengan cara manual. Pelanggan yang datang dari luar Kota Salatiga membuktikan bahwa Soemardjan Florist memiliki peluang dan berpotensi untuk melebarkan pasar hingga ke luar kota dengan cara melayani transaksi dari semua daerah di pulau Jawa. Pada era informasi dan globalisasi yang kian berkembang pesat, teknologi makin banyak diterapkan dalam berbagai aspek kehidupan. Pemanfaatan teknologi yang tepat dapat meningkatkan produktivitas dalam dunia bisnis [1]. Salah satu

teknologi yang dapat meningkatkan produktivitas bisnis adalah *e-commerce*. *E-commerce* adalah kegiatan membeli atau menjual produk pada layanan *online* atau melalui Internet. Selain meningkatkan produktivitas *e-commerce* dapat membantu dalam memperluas pemasaran [2].

Berdasarkan permasalahan yang ada maka diperlukan sebuah *website e-commerce* untuk mengembangkan pasar hingga ke luar Kota Salatiga guna memajukan usaha toko bunga Soemardjan Florist. *Website* tersebut nantinya akan digunakan sebagai sarana promosi dan penjualan untuk melayani transaksi dari daerah luar kota, sehingga diharapkan pelanggan Soemardjan Florist tidak hanya datang dari Salatiga. Berdasarkan latar belakang yang sudah dijelaskan, penelitian yang dilakukan yaitu membangun *website e-commerce* Soemardjan Florist memanfaatkan *framework* Laravel versi 5.5. Laravel dipilih sebagai *framework* PHP karena memiliki banyak *library* dan bersifat *opensource* yang diharapkan dapat mempermudah pembangunan *website* Soemardjan Florist [3].

2. METODOLOGI PENELITIAN

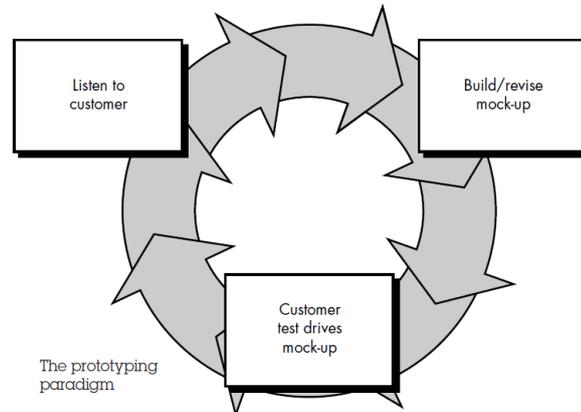
Tahapan dilakukan dalam penelitian ini ditunjukkan pada Gambar 1.



Gambar 1. Tahapan Penelitian

Berdasarkan Gambar 1 dapat dijelaskan bahwa tahap pertama penelitian yaitu analisis kebutuhan dan pengumpulan data. Tahap ini merupakan tahap awal penelitian. Analisis kebutuhan dilakukan melalui observasi dan wawancara kepada pemilik Soemardjan Florist, menunjukkan bahwa Soemardjan Florist belum memiliki *website* profil maupun toko *online*. Tahapan kedua, ketiga dan keempat adalah perancangan sistem, pembuatan aplikasi, dan pengujian aplikasi dengan menggunakan metode *prototyping*. Pengembangan *website* dengan metode *prototyping* dilakukan secara bertahap, yaitu dengan membuat *prototype* awal, kemudian *prototype*

tersebut akan dikembangkan menjadi *website* yang akan diperbarui menyesuaikan kebutuhan dari pengguna dari waktu ke waktu [11]. Tahapan atau langkah-langkah yang dilakukan pada metode *prototyping* ditunjukkan pada Gambar 2.



Gambar 2. Tahapan Metode Prototype [11]

Proses *prototyping* pada penelitian ini menghasilkan dua siklus. Berdasarkan Gambar 2, tahapan *prototyping* dapat dijelaskan sebagai berikut: Tahap pertama dari fase 1 *prototyping* adalah *listen to customer*, yaitu melakukan *briefing* dengan klien (pemilik Soemardjan Florist) tentang *website* yang akan dibangun. Beberapa informasi diperoleh dari hasil *briefing*, yaitu: (a) Toko bunga Soemardjan Florist membutuhkan *website* sebagai sarana promosi sekaligus *e-commerce* untuk menunjang penjualan bunga; (b) *Website* dapat menampilkan halaman profil dan artikel atau *blog* yang dapat menunjang promosi toko bunga Soemardjan Florist; (c) *Website* dapat menampilkan seluruh produk beserta harga dan detailnya; (d) Administrator dapat menambah, mengubah dan menghapus produk dan artikel atau *blog*; dan (e) *User* tidak perlu mendaftar agar dapat melakukan transaksi dalam *website*.

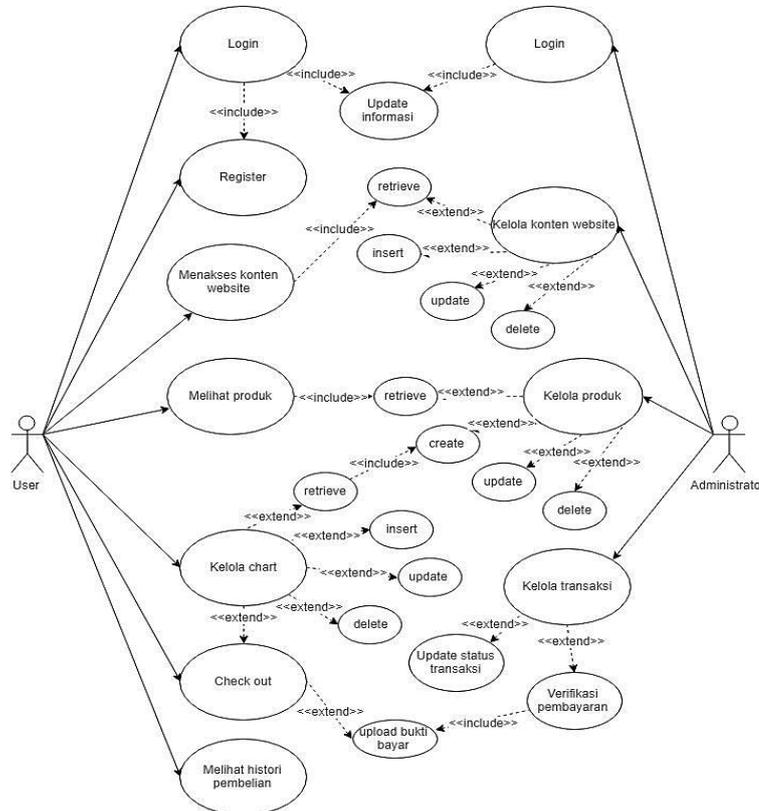
Tahap kedua adalah *build / revise mock-up*. Tahap ini diawali dengan membuat *UML (Unified Modelling Language)* berdasarkan kebutuhan yang telah diperoleh dari *briefing*. Diagram yang pertama dibuat adalah *use case diagram*. *Use case diagram* merupakan pemodelan untuk mengetahui kelakuan (*behaviour*) sistem informasi yang akan dibuat. *Use case diagram* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat dan berfungsi untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi itu [12].

3. HASIL DAN PEMBAHASAN

3.1. Use Case diagram

Gambar 3 menjelaskan bahwa dalam *website* Soemardjan Florist melibatkan tiga aktor, di antaranya adalah *user*, administrator dan bagian pengiriman. Administrator dapat menjalankan seluruh fungsi penuh dalam *website* yang meliputi; *login*, mengelola produk, mengelola artikel serta

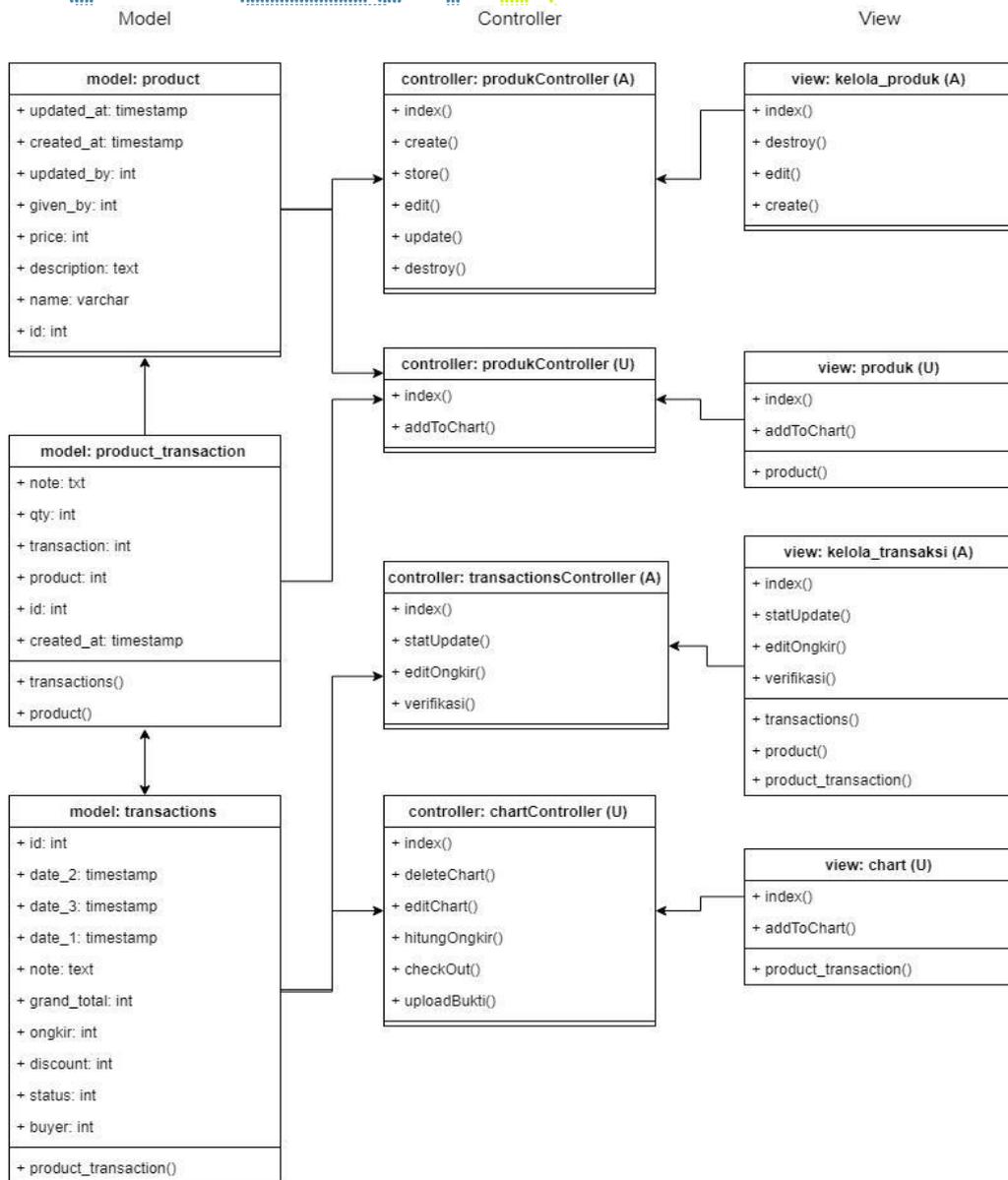
memverifikasi pembayaran. *User* dapat *login*, mendaftar, melihat konten *website* serta melakukan transaksi. Setelah *user* melakukan *check out* transaksi, pembayaran akan diverifikasi oleh administrator, yang selanjutnya *order* akan diteruskan pada bagian pengiriman.



Gambar 3. Use Case Diagram Sistem

3.2. Class Diagram

Class diagram dalam *website* Soemardjan Florist ditunjukkan pada Gambar 4. Gambar 4 menunjukkan keterkaitan kelas *MVC* (*Model*, *View*, *Controller*) dalam *framework* Laravel yang digunakan pada *website* Soemardjan Florist. Struktur kelas dibagi dalam tiga bagian, yaitu *model*, *controller* dan *view*. Semua konten yang terdapat pada kelas-kelas *view* merupakan tampilan antar muka berisi data yang telah diolah oleh fungsi yang berada di dalam kelas-kelas *controller*. Kelas-kelas *controller* tersebut memanipulasi data melalui kelas-kelas *model* yang bertugas mengolah dan berkomunikasi dengan basis data.



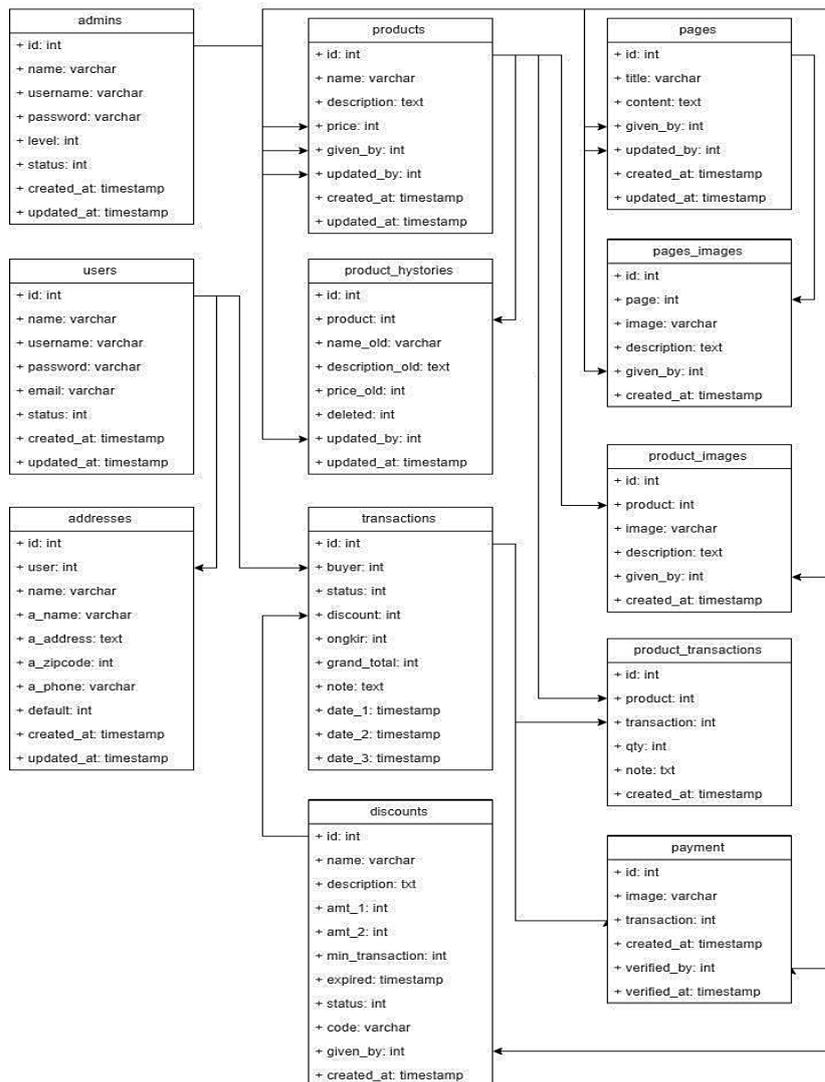
Gambar 4. Class Diagram Sistem

3.3. Relasi Basis Data Sistem

Relasi basis data dalam *website* Soemardjan Florist ditunjukkan pada Gambar 5. Gambar 5 menunjukkan relasi basis data yang digunakan dalam *website* Soemardjan Florist. Pada *website* Soemardjan Florist menggunakan 12 tabel yang terikat satu sama lainnya. Langkah terakhir dalam siklus ini adalah *building process*, yaitu membuat *prototype* dari UML yang telah dirancang.

Tahap ketiga dari fase pertama adalah *customer test drives mock-up*. Pada tahap ini dilakukan evaluasi dari *prototype* yang telah dibuat. Selanjutnya klien (pemilik Soemardjan Florist) melakukan *test drive* untuk meninjau *prototype* berdasarkan *briefing* yang dilakukan sebelumnya. Hasil *test drive* ditunjukkan pada *prototyping* 2 tahap *listen to customer*.

Tahap pertama dari fase 2 *prototyping* adalah *listen to customer*. Pada tahap ini dilakukan *briefing* kedua dengan klien. Berdasarkan hasil tes didapatkan beberapa revisi, sebagai berikut. Tahap kedua dari fase ke dua adalah *build / revise mock-up*. Pada tahap ini dilakukan peninjauan *prototype* yang telah dibuat sebelumnya dengan merevisi fungsi sesuai dengan *briefing* klien. Tahap terakhir pada fase kedua adalah *customer test drives mock-up*. Hasil evaluasi oleh klien pada *prototype 2* menunjukkan bahwa *prototype website* Soemardjan Florist sudah sesuai dan siap untuk dikembangkan lebih lanjut.

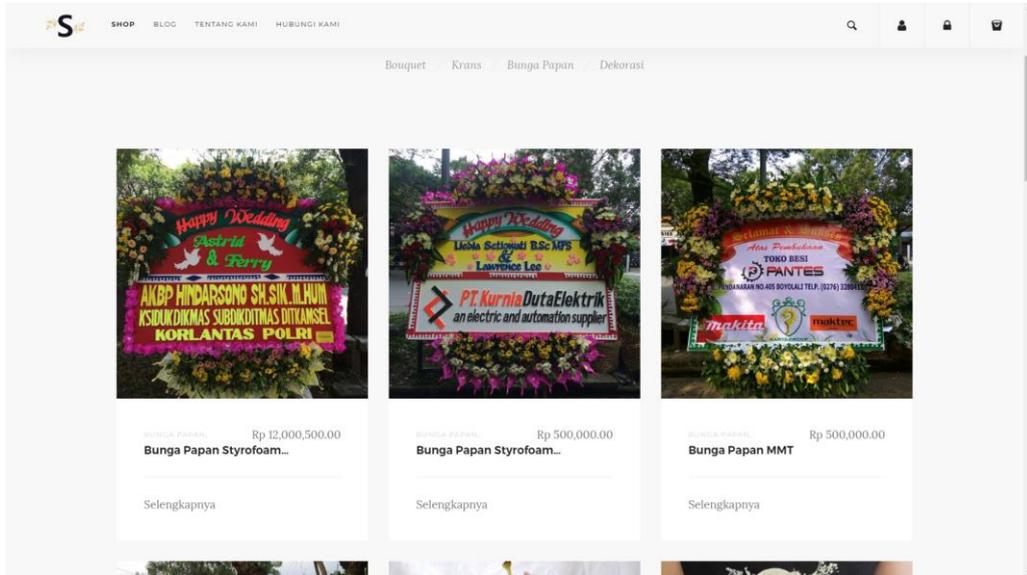


Gambar 5. Relasi Basis Data Sistem

3.4. Hasil penelitian

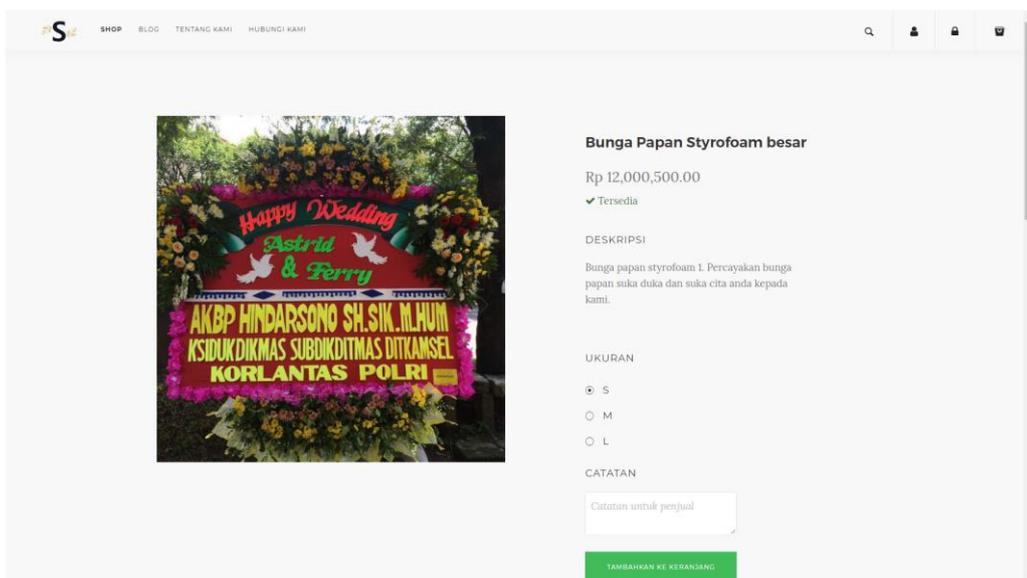
Website Soemardjan Florist dibuat dengan dua fungsi utama, yaitu sebagai *e-commerce* dan *website* profil. Dalam *website* ini terdapat dua modul yaitu administrator dan modul publik. Pada modul administrator, administrator yang telah terdaftar harus melewati sistem *login* sebelum

dapat mengaksesnya. Dalam modul administrator terdapat beberapa menu untuk mengelola *website*, di antaranya adalah pengelolaan *blog*, produk, pesanan dan kategori. Pada modul publik, pengunjung dapat mengakses seluruh konten yang berada pada *public directory*. Tampilan modul publik halaman *shop* dapat dilihat pada Gambar 6.



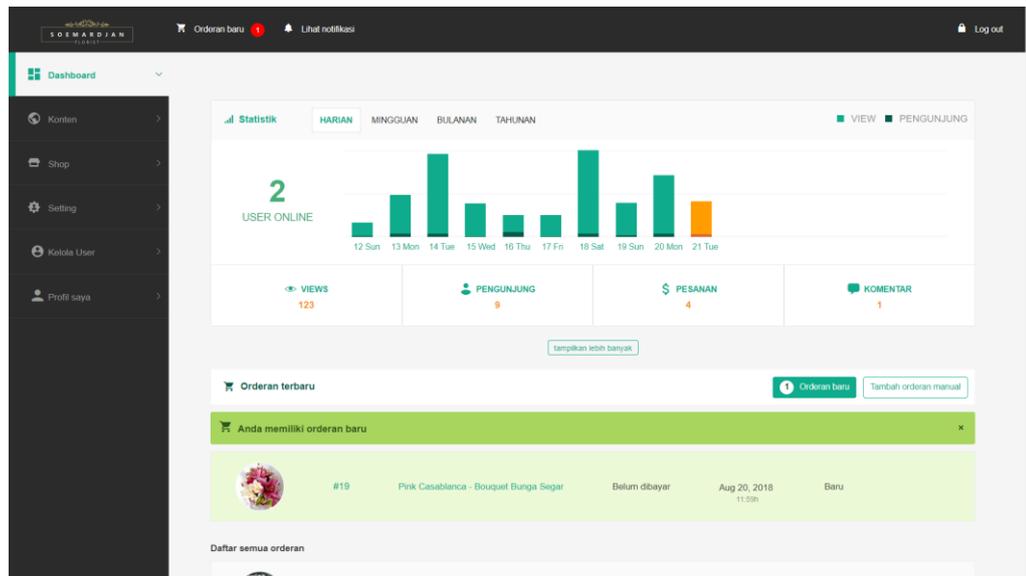
Gambar 6. Halaman Shop

Gambar 6 menunjukkan halaman *shop* yang berisi daftar produk dan kategori produk. Dalam halaman ini pengunjung dapat memilih produk sebelum memasukkannya ke dalam keranjang belanja. Dalam daftar produk terdapat elemen *thumbnail*, kategori, harga, judul dan tautan menuju detail produk. Tampilan halaman detail produk dapat dilihat pada Gambar 7.



Gambar 7. Halaman Detail Produk

Gambar 7 menunjukkan halaman detail produk. Pada halaman ini terdapat beberapa informasi mengenai produk yang dipilih pengunjung, meliputi gambar, judul, harga, deskripsi dan beberapa detail lainnya. Pengunjung dapat menambahkan produk ke keranjang belanja jika variabel harga tidak nol atau kosong dan status stok tersedia. Tampilan modul administrator dapat dilihat pada Gambar 8.



Gambar 8. Tampilan Modul Administrator

Gambar 8 menunjukkan tampilan dari modul administrator yang berisi beberapa menu yaitu: 1) *Dashboard*, menu ini berisi beberapa ringkasan mengenai pengelolaan *website*, seperti statistik, pesanan baru dan komentar baru; 2) *Konten*, menu ini berisi tentang pengelolaan konten dalam modul publik, meliputi halaman, *blog*, kategori dan produk. Pada menu ini administrator dapat menambah, mengubah dan menghapus konten; 3) *Shop*, pada menu ini administrator dapat mengelola pesanan, meliputi menerima pesanan, menolak pesanan serta mengatur pesanan 4) *Setting*, menu ini berisikan pengaturan yang berhubungan dengan *website*, seperti mengatur tautan media sosial dan mengatur *metadata website*; 5) *Kelola user*. Dalam menu ini administrator dapat mengelola seluruh administrator yang terdaftar, termasuk menambah, mengubah dan menghapus data administrator; dan 6) *Profil Saya*, menu ini berisikan pengaturan administrator yang sedang *login*. Dalam menu Profil, Saya administrator dapat mengubah *avatar*, informasi pribadi dan level administrator. Perintah untuk mengelola produk ditunjukkan pada Kode Program 1.

Kode Program 1 Perintah untuk Mengelola Produk

```
<?php
namespace App\Http\Controllers;
class ProdukController extends Controller
{
    public function store(Request $request)
    {
        $produk = new Produk;
        $produk->nama = $request->nama;
        //variabel lainnya
        $produk->save();
        return redirect('admin/produk')->with('success', 'Tersimpan');
    }
    public function edit($id)
    {
        $produk = Produk::find($id);
        session()->put('edit', url()->previous());
        return view('admin.edit_produk',compact('produk','id'));
    }
    public function update(Request $request, $id)
    {
        $produk = Produk::find($id);
        $produk->judul = $request->judul;
        $produk->isi = $request->isi;
        //variabel lainnya
        $produk->save();
        return redirect(session()->pull('edit'))->with('success',
        'Terupdate');
    }
    public function delete($id)
    {
        $produk = Produk::find($id);
        if (!$produk)
            abort(404);
        $produk->withTrashed();
        return back()->with('success', 'Terhapus');
    }
}
```

Kode Program 1 merupakan file ProdukController dari MVC yang bertugas mengelola produk seperti membuat, mengubah dan menghapus. Perintah pada baris 5-12 adalah fungsi store(). Fungsi store() bertugas untuk menambah atau menyimpan produk baru. Perintah pada baris 13-18 adalah fungsi edit(), berfungsi untuk mengambil data dari variabel id dan meneruskannya ke dalam view edit_produk untuk diubah administrator. Perintah pada baris 19-18 adalah fungsi update(). Data dari view edit_produk yang telah diubah dan disimpan administrator akan dikembalikan ke controller dan akan di-update melalui fungsi update(). Perintah pada baris 29-36 adalah fungsi delete yang berfungsi untuk menghapus produk dengan metode *soft delete*. *Soft delete* tidak benar-benar menghapus produk dari database melainkan hanya menambahkan variabel terhapus ke dalam baris produk berdasarkan id produk, sehingga website tidak akan menampilkan produk tersebut.

Website Soemardjan Florist yang telah dirancang kemudian diuji menggunakan *alfa* dan *beta testing*. Pengujian *alpha* dilakukan dengan metode *blackbox* yaitu pengujian langsung tanpa memperhatikan alur eksekusi program [13]. Pengujian ini dilakukan untuk mengetahui apakah fungsi yang diuji telah berjalan sesuai yang diharapkan. Hasil pengujian *alfa* dengan *blackbox* dapat dilihat pada Tabel 1.

Tabel 1. Hasil Pengujian *Black-box*

Fungsi yang diuji	Kondisi	Output yang diharapkan	Output yang dihasilkan sistem	Status Pengujian
Menampilkan halaman website	Membuka semua halaman publik	Sukses tampilkan semua halaman	Sukses tampilkan semua halaman	<i>Valid</i>
Log in	<i>Username</i> dan <i>password</i> benar <i>Username</i> dan <i>password</i> kosong atau salah	Berhasil <i>log in</i>	Berhasil <i>log in</i>	<i>Valid</i>
		Gagal <i>log in</i>	Gagal <i>log in</i>	<i>Valid</i>
Menampilkan daftar produk	Membuka halaman <i>shop</i>	Menampilkan seluruh produk yang <i>di-publish</i>	Menampilkan seluruh produk yang <i>di-publish</i>	<i>Valid</i>
Menampilkan detail produk	Membuka halaman detail produk	Sukses tampilkan halaman detail produk	Sukses tampilkan halaman detail produk	<i>Valid</i>
Menampilkan daftar blog	Membuka halaman daftar blog	Menampilkan seluruh blog yang <i>di-publish</i>	Menampilkan seluruh blog yang <i>di-publish</i>	<i>Valid</i>
Menampilkan detail blog	Membuka halaman detail blog	Sukses tampilkan halaman detail blog	Sukses tampilkan halaman detail blog	<i>Valid</i>
Menambahkan produk ke keranjang belanja	Menekan tombol Tambahkan ke Keranjang	Sukses menambah produk ke keranjang belanja	Sukses menambah produk ke keranjang belanja	<i>Valid</i>
Check out belanja	Menekan tombol <i>check out</i>	Sukses <i>check out</i> dan notifikasi muncul di modul administrator	Sukses <i>check out</i> dan notifikasi muncul di modul administrator	<i>Valid</i>
Menampilkan halaman status	Membuka halaman Tentang Kami dan Hubungi Kami	Sukses menampilkan halaman Tentang Kami dan Hubungi Kami beserta isinya	Sukses menampilkan halaman Tentang Kami dan Hubungi Kami beserta isinya	<i>Valid</i>
Menampilkan halaman hubungi kami	Membuka halaman hubungi kami	Sukses tampilkan halaman hubungi kami	Sukses tampilkan halaman hubungi kami	<i>Valid</i>
Mengelola produk dan blog	Administrator menambah, mengubah dan menghapus produk	Administrator sukses menambah, mengubah dan menghapus produk	Administrator sukses menambah, mengubah dan menghapus produk	<i>Valid</i>
Tampilan responsif	Membuka <i>website</i> pada beberapa perangkat dengan ukuran layar yang berbeda	Tampilan menyesuaikan layar	Tampilan menyesuaikan layar	<i>Valid</i>

Tabel 1 menunjukkan bahwa fungsi dalam *website* Soemardjan Florist berjalan seperti yang diharapkan. Berdasarkan pengujian *alpha* dapat disimpulkan bahwa *website* Soemardjan Florist telah memenuhi seluruh kondisi yang disyaratkan dengan *output* yang diharapkan. Setelah pengujian *alpha* dilakukan, pengujian *beta* diperlukan untuk mendapat evaluasi dari pengguna. Pengujian *beta* dilakukan melalui pembagian kuesioner kepada 40 responden yang terdiri dari pelanggan Soemardjan Florist dan masyarakat

umum. Responden mengakses rancangan *website* Soemardjan Florist pada *device* yang berbeda. Hasil rekapitulasi kuesioner dari pengujian *beta* dapat dilihat pada Tabel 2.

Tabel 2. Rekapitulasi Kuisisioner Pengujian *Beta*

	Pertanyaan	STS	TS	CS	S	SS
	Apakah <i>website</i> Soemardjan Florist mudah digunakan di <i>device</i> Anda?	1	4	11	20	4
	Apakah navigasi pada <i>website</i> Soemardjan Florist mudah dipahami/tidak membingungkan?	0	2	20	15	3
	Apakah tampilan, huruf dan gambar dari aplikasi ini dapat dilihat dengan jelas di <i>device</i> Anda?	0	7	14	17	2
	Apakah <i>website</i> Soemardjan Florist membantu dalam memperoleh informasi seputar Soemardjan Florist?	0	1	8	21	10
	Apakah Anda mudah menemukan produk di <i>website</i> Soemardjan Florist?	2	10	24	4	0
	Apakah <i>website</i> Soemardjan Florist efektif sebagai situs belanja <i>online</i> ?	2	12	20	5	1

Berdasarkan data pada Tabel 2, hasil rekapitulasi kuisisioner untuk pertanyaan 1 menunjukkan sebanyak 3% responden menjawab sangat tidak setuju, 10% responden menjawab tidak setuju, 28% responden menjawab cukup setuju, 50% responden menjawab setuju dan 10% responden menjawab sangat setuju. Jadi disimpulkan bahwa *website* Soemardjan Florist mudah untuk digunakan.

Hasil rekapitulasi kuisisioner untuk pertanyaan 2 menunjukkan sebanyak 0% responden menjawab sangat tidak setuju, 5% responden menjawab tidak setuju, 50% responden menjawab cukup setuju, 38% responden menjawab setuju dan 8% responden menjawab sangat setuju. Jadi disimpulkan bahwa *website* Soemardjan Florist mudah untuk dipahami.

Hasil rekapitulasi kuisisioner untuk pertanyaan 3 menunjukkan sebanyak 0% responden menjawab sangat tidak setuju, 18% responden menjawab tidak setuju, 35% responden menjawab cukup setuju, 43% responden menjawab setuju dan 5% responden menjawab sangat setuju. Jadi disimpulkan bahwa *website* Soemardjan Florist dapat dilihat dengan jelas melalui *device*.

Hasil rekapitulasi kuisisioner untuk pertanyaan 4 menunjukkan sebanyak 0% responden menjawab sangat tidak setuju, 3% responden menjawab tidak setuju, 20% responden menjawab cukup setuju, 53% responden menjawab setuju dan 25% responden menjawab sangat setuju. Jadi disimpulkan bahwa *website* Soemardjan Florist membantu dalam memperoleh informasi seputar Soemardjan Florist.

Hasil rekapitulasi kuisisioner untuk pertanyaan 5 menunjukkan sebanyak 5% responden menjawab sangat tidak setuju, 25% responden menjawab tidak setuju, 60% responden menjawab cukup setuju, 10% responden menjawab setuju dan 0% responden menjawab sangat setuju. Jadi

disimpulkan bahwa *website* Soemardjan Florist mempermudah untuk menemukan produk di Soemardjan Florist.

Hasil rekapitulasi kuisioner untuk pertanyaan 6 menunjukkan sebanyak 5% responden menjawab sangat tidak setuju, 30% responden menjawab tidak setuju, 50% responden menjawab cukup setuju, 13% responden menjawab setuju dan 3% responden menjawab sangat setuju. Jadi disimpulkan bahwa *website* Soemardjan Florist efektif sebagai situs belanja *online*.

4. SIMPULAN

Berdasarkan penelitian yang telah dilakukan maka dapat disimpulkan bahwa *website* Soemardjan Florist berhasil dibangun dengan menggunakan *framework* Laravel versi 5.5. Penggunaan *framework* Laravel dapat membantu pembangunan *website* dengan cepat karena memiliki fitur pihak ke tiga berupa *vendor* yang bersifat *opensource*. Proses penulisan kode juga lebih terorganisir berkat struktur *MVC* yang dimiliki Laravel. Sedangkan *Distance Matrix* membuat penghitungan ongkos kirim menjadi lebih mudah. *Website* Soemardjan Florist ini mempunyai fitur *blog* dan *shop*. Dimana pengunjung dapat mengakses informasi berupa *blog* yang dibuat oleh administrator sekaligus berbelanja produk yang ditawarkan Soemardjan Florist. Untuk penelitian selanjutnya diharapkan dapat dibangun aplikasi berbasis *mobile* untuk mempermudah pengguna *smartphone*.

DAFTAR PUSTAKA

- [1] Hidayatullah, Deny. 2005. "Dampak Teknologi Informasi dan Internet terhadap Pendidikan, Bisnis, dan Pemerintahan Indonesia." *Majalah Ekonomi dan Komputer No.1, Universitas Gunadarma*.
- [2] Kuncoro, Cahyono Wantosusilo, Aristarchus Pranayama Kuntjara, dan Ryan Pratama Sutanto. 2016. "Perancangan Website E-Commerce Ud. Sardono." *Jurnal DKV Adiwarna, Universitas Kristen Petra*.
- [3] Suharsana, I Ketut Suharsana, IGP Wirarama Wedashwara Wirawan, dan Ni Luh Ayu Kartika Yuniastari S.. 2000. "Implementasi Model View Controller Dengan Framework Codeigniter Pada E-Commerce Penjualan Kerajinan Bali." *Jurnal Sistem dan Informatika (JSI), STIKOM Bali*.
- [4] Rohman. 2014. "Mengenal Framework Laravel (Best PHP Frameworks For 2014)." *ILMUTI*.
- [5] Prasetya, K.J. 2016. Pengembangan Jurnal Elektronik FTI UKSW Berbasis Web menggunakan Framework CodeIgniter. Salatiga: Universitas Kristen Satya Wacana.