

Pengembangan Sistem Informasi Pemesanan Layanan Jasa Cleaning Service Berbasis Web Dan Mobile Di Liochita Cleaning Semarang

Aulio Romadho Agung¹⁾, Rinta Kridalukmana²⁾, Ike Pertiwi Windasari²⁾
Program Studi Sistem Komputer, Fakultas Teknik, Universitas Diponegoro
Jalan Prof. Sudharto, Tembalang, Semarang, Indonesia

Abstract—Liochita cleaning is a company engaged in the field of cleaning services which are located in the city of Semarang. Until now, the existing of information systems on Liochita Cleaning were not sufficiently able to manage the company and thus to make this company as a company that developed and developing its field and can compete with other companies is not possible. Start from recording customer data and order data, which is the became one as income data, so this companies are less aware in detail of the customer data. On the other hand, customers must make a call in advance to order the services that it requires no small cost. Lack of marketing facilities makes this company unable to include all of costumers in the city of Semarang. Data recording is still using paper, so this company were at risk of paper lost which is containing data that has been recorded.

Development of an information system in this study using the method of the waterfall. Waterfall model consists of: requirements, design, implementation, testing and maintenance. For the start of the design, developer need to observations or interviews to determine the needs of the system that being developed. Applications developed using the framework CodeIgniter.

The results of this final project is an information system that can meet the needs of Liochita Cleaning in the management of these services, customers and orders in accordance with the business processes that Liochita Cleaning have and allow customers to book services.

Keywords: *information systems, Android, CodeIgniter, Bootstrap*

I PENDAHULUAN

Liochita Cleaning merupakan perusahaan yang bergerak di bidang jasa kebersihan yang berlokasi di wilayah Semarang. Perusahaan ini melayani jasa pencucian karpet dan pembersihan kaca gedung.

Pada Liochita Cleaning pencatatan data pelanggan dan data pesanan yang masih menjadi satu yaitu data pemasukan sehingga perusahaan kurang mengetahui data pelanggan secara terperinci, serta pencatatan semua data yang ada masih menggunakan kertas hal ini memiliki resiko kertas dan data yang telah tercatat tersebut hilang.

Ilmu pengetahuan serta teknologi informasi saat ini telah mengalami perkembangan. Kemudahan dalam mengakses informasi yang dapat diakses dimanapun dan

kapanpun inilah yang menjadikan pemanfaatan teknologi diterapkan hampir diberbagai bidang termasuk pada bidang wirausaha. Ketersediaan dan Kemudahan yang ditawarkan mampu menjaring lebih banyak konsumen bagi perusahaan yang bergerak di bidang usaha jasa kebersihan (*cleaning service*) serta memudahkan konsumen untuk melakukan pemesanan jasa, selain itu bagi perusahaan itu sendiri sistem informasi sangat membantu dalam penyimpanan dan pengolahan data yang masuk dan keluar serta mampu meningkatkan produktifitas perusahaan tersebut dan juga mampu meminimalisir terjadinya kehilangan data pada saat proses pengolahan data terjadi.

Sistem informasi dapat digunakan untuk membantu proses pemasaran dan pemesanan di Liochita Cleaning. Sebagai contoh, pemesanan jasa pencucian karpet atau pembersihan kaca gedung dapat dilakukan di dalam sebuah perangkat lunak serta pembuatan laporan transaksi juga dapat dilakukan berdasarkan data pemesanan.

Salah satu alternatif solusi dari permasalahan tersebut adalah sistem informasi berbasis *web* dan *mobile* yang diharapkan mampu memudahkan pelanggan dalam melakukan pemesanan, memudahkan karyawan untuk melakukan pelayanan dan pemasaran serta dapat digunakan untuk memonitor perusahaan bagi pemilik. Perangkat bergerak yang akan digunakan adalah jenis ponsel pintar dengan sistem operasi *Android*, maka sistem informasi berbasis *mobile* tersebut harus dirancang agar dapat dijalankan pada platform *Android*.

II LANDASAN TEORI

A Penelitian Sebelumnya

Sebelum melakukan pengembangan sistem diperlukan beberapa studi literatur untuk mengetahui efektivitas penggunaan sistem informasi pada pemesanan layanan.

Rakasiwi Syarief Hidayat^[3] berpendapat bahwa Sistem informasi pelayanan merupakan aplikasi komputer yang digunakan untuk bisnis yang berkaitan dengan jasa pelayanan, terutama dalam sistem penyimpanan dan pengolahan data maupun informasi agar dapat menjalani bisnis jasa dengan mudah dan berjalan dengan baik. Keadaan tersebut menyebabkan banyaknya perusahaan yang

meningkatkan pengembangan di bidang pelayanan jasa untuk meningkatkan pelayanan yang lebih baik lagi.

Reza Milady Fauzan [2] dari hasil penelitian yang dilakukan bahwa sistem informasi pemesanan produk berbasis web mampu meningkatkan kinerja perusahaan dengan semakin luasnya jangkauan pemasaran produk kepada konsumen, memudahkan konsumen untuk mendapatkan informasi secara lengkap dan cepat dan meminimalisir kesalahan data serta kesalahan perhitungan keuangan hasil transaksi terhadap konsumen.

Adriana Winina [8] dalam penelitiannya berpendapat bahwa dengan pengelolaan pemesanan produk dan layanan jasa berbasis web dapat memudahkan admin dalam hal mencatat, menyimpan, dan menampilkan data untuk melakukan transaksi pemesanan produk maupun jasa. Menggunakan metode *model-view-controller* atau MVC mampu memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, antarmuka pengguna, dan bagian yang menjadi kontrol dalam sebuah aplikasi web.

B Sistem Informasi

Sistem Sistem adalah sekumpulan unsur atau elemen yang saling berkaitan dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai tujuan bersama. Suatu sistem harus mempunyai sasaran, tujuan, komponen-komponen yang saling berinteraksi satu sama lainnya sehingga dapat mencapai tujuan yang telah ditetapkan. Informasi merupakan fakta yang memiliki arti sehingga memungkinkan untuk dimanfaatkan. Sedangkan semua fakta tersebut yang tercatat, terekam atau terdokumentasikan disebut data.

Burch dan Grudnitski [4] pada tahun 1989 berpendapat bahwa Informasi memiliki tingkat kualitas yang ditentukan. Informasi haruslah akurat yaitu informasi harus bebas dari kesalahan-kesalahan dan harus jelas penyampaian maksudnya. Informasi harus tepat waktu yaitu informasi yang datang tidak boleh terlambat kepada penerimanya. Informasi harus relevan yaitu informasi harus mempunyai manfaat bagi pengguna.

C Kerangka Kerja Codeigniter

Codeigniter adalah aplikasi *open source* yang berupa *framework* untuk membangun website dinamis dengan menggunakan bahasa pemrograman PHP. Codeigniter menawarkan kemudahan serta standarisasi dalam proses pengembangan website dan aplikasi berbasis web. Dengan Codeigniter proses pengembangan website menjadi lebih cepat dan terstandar. Standar ini sangat berguna dalam pengembangan website yang dikerjakan oleh tim.

Laravel Codeigniter juga telah menyediakan *library* dan *helper* yang berguna dan mempermudah proses *development*. Terdapat beberapa kelebihan penggunaan Codeigniter dalam pengembangan suatu proyek, yaitu :

1. Menghemat Waktu
2. Code Reuse
3. Dukungan Komunitas
4. Kumpulan Best Practice

Codeigniter dibangun menggunakan konsep *Model-View-Controller development pattern*, yaitu sebuah teknik pemrograman yang memisahkan logika bisnis, logika data, dan logika presentasi [7].

D Twitter Bootstrap

Menurut Spurlock [6] Twitter Bootstrap merupakan sebuah *framework* yang dikembangkan oleh Twitter guna menciptakan sebuah tampilan web yang responsif. Sejak peluncurannya pada tahun 2011 Twitter Bootstrap telah digunakan oleh banyak pengembang web karena

kemampuannya untuk menyesuaikan bentuk dan ukuran tampilan aplikasi sesuai perangkat pengguna.

Twitter Bootstrap terdiri dari CSS, Javascript dan berkas gambar yang telah di gabungkan sehingga dapat digunakan oleh pengembang web. Untuk penggunaan Twitter Bootstrap dalam sebuah modul pengembangan web file CSS dan Javascript direferensikan di dalam file HTML

Penggunaan *Twitter Bootstrap* dapat mempermudah *developer* dalam mempercantik *web* yang dibuatnya dan mengurangi waktu pengerjaan dalam pembuatan sebuah *website*.

E MySQL Server

MySQL adalah salah satu jenis basis data server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan basis data sebagai sumber dan pengelolaan datanya. Kepopuleran MySQL antara lain karena MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses basis data-nya sehingga mudah untuk digunakan, kinerja *query* cepat, dan mencukupi untuk kebutuhan basis data perusahaan-perusahaan skala menengah-kecil. MySQL juga bersifat *open source* dan gratis pada berbagai platform. MySQL didistribusikan dengan lisensi *open source GPL (General Public License)* mulai versi 3.23, pada bulan Juni 2000 [1].

Sistem basis data *MySQL* didukung oleh bahasa operasional SQL. SQL singkatan dari *Structured Query Language*. SQL merupakan bahasa *query* standar yang digunakan untuk mengakses basis data relasional.

Kemampuan SQL tidak hanya untuk *query* (memperoleh data) saja, tetapi juga mencakup kemampuan lain, seperti pendefinisian struktur data, perubahan data, pengaturan keamanan, dan lain-lain. SQL belum dapat dikatakan sebagai bahasa tetapi sebagai subbahasa. Hal ini karena SQL tidak mendukung persyaratan bahasa yang lengkap seperti pernyataan pengujian kondisi dan pengulangan atau iterasi. Sebagai *server* basis data, MySQL dapat dikatakan lebih unggul dibandingkan *server* basis data lainnya dalam *query* data [5].

III PERANCANGAN SISTEM

A Analisis Kebutuhan

Penyusunan analisis kebutuhan dilakukan untuk mengurangi risiko sistem gagal dikembangkan. Kegagalan pengembangan sistem umumnya terjadi karena sistem tidak memenuhi kebutuhan pengguna atau sistem tidak sesuai dengan proses bisnis yang berjalan. Studi kelayakan perlu dilakukan dalam penyusunan analisis kebutuhan untuk menghindari risiko tersebut. Studi kelayakan dapat dilakukan dengan cara wawancara, observasi dan membaca dokumen. Hasil analisis kebutuhan berupa dokumen *System Requirements Specifications (SRS)*. Pengembangan ini menggunakan struktur SRS tersebut untuk menyampaikan analisis kebutuhan.

1) Kondisi Awal

Proses selama wawancara dan observasi memberikan gambaran kondisi yang terjadi dilapangan. Pada proses pemesanan, terdapat beberapa aktivitas yang harus dilakukan, antara lain:

1. Admin menerima pesanan yang masuk.
2. Admin memeriksa tanggal yang kosong untuk mengerjakan pesanan pelanggan agar pesanan yang masuk dapat segera diproses.
3. Admin memberikan konfirmasi kepada pelanggan yang memesan.

Jika ketiga proses tersebut telah dilalui, proses dapat berlanjut menuju persetujuan. Apabila persetujuan telah

dilakukan, aktivitas dalam proses tersebut akan bertambah menjadi:

1. Admin mengecek kembali pesanan dan informasi harga yang harus di bayar.
2. Admin mengubah status pesanan pelanggan.
3. Admin mencatat bahan baku yang digunakan untuk memenuhi kebutuhan pesanan pelanggan.
4. Admin memberi informasi kepada pelanggan bahwa pekerjaan telah selesai.
5. Admin mengubah status pembayaran dan menambahkan transaksi tersebut ke dalam laporan pendapatan.
6. Admin memberikan laporan pendapatan kepada pemilik.

Dari proses yang ada maka dapat dianalisis data yang diolah sebelum sistem dikembangkan. Data tersebut yaitu:

- a. Nama pelanggan
- b. Alamat pelanggan (untuk pengiriman)
- c. Jenis jasa (cuci karpet atau cuci kaca gedung)
- d. Harga layanan
- e. Bahan baku
- f. Detail pesanan

Pembuatan SRS dapat dilakukan setelah data yang dikumpulkan pada kondisi awal sudah cukup menjelaskan keadaan sebelum sistem dikembangkan.

2) Deskripsi Umum Sistem

Deskripsi umum sistem diperlukan untuk menjelaskan perspektif produk akhir hasil pengembangan, fungsi produk yang diharapkan untuk berjalan, kelompok pengguna, lingkungan pengoperasian serta lingkungan pengguna yang akan menggunakan sistem.

1. Perspektif Produk Akhir

Produk pengembangan adalah sistem informasi yang dapat digunakan oleh Liochita Cleaning dalam melakukan pelayanan. Sistem informasi tersebut digunakan untuk mengelola data aktivitas pemesanan jasa.

2. Fungsi Produk

Sistem informasi dapat menyimpan data pemesanan. Pengolahan transaksi dapat dilakukan lebih cepat menggunakan komputer. Data yang disimpan dalam sistem dapat diolah untuk menghasilkan laporan.

3. Kelompok Pengguna

Pengguna yang menggunakan sistem ini terdiri dari 3 jenis, yaitu: pengguna admin, pengguna pelanggan dan pengguna owner atau pemilik. ketiga pengguna tersebut memiliki aktivitas yang berbeda dalam menggunakan sistem.

- a. Pengguna admin adalah pengguna yang bertugas melakukan pengelolaan terhadap pengguna pelanggan, data bahan baku dan data transaksi pemesanan. Pengguna admin berperan sebagai staf Liochita Cleaning.
- b. Pengguna pelanggan adalah pengguna yang melakukan aktivitas pemesanan. Pengguna pelanggan merupakan pelanggan dari Liochita Cleaning yang akan melakukan pemesanan layanan jasa.
- c. Pengguna owner adalah pengguna yang melakukan pengelolaan terhadap pengguna admin dan memonitor data transaksi pemesanan.

4. Lingkungan Pengoperasian

Sistem dioperasikan di Liochita Cleaning. Liochita Cleaning menggunakan koneksi internet dalam mengoperasikan sistem. Sistem bisa digunakan

sepanjang hari selama *server* dan jaringan tidak mengalami gangguan.

5. Lingkungan Pengguna

Pada sisi Butik Goddies Fashion aplikasi berupa aplikasi berbasis *web*. Pengguna bisa mengakses sistem melalui *web browser* dari perangkat komputer atau tablet, sedangkan pada sisi pelanggan Butik Goddies Fashion aplikasi berupa aplikasi ponsel pintar yang telah dipasang sebelumnya.

3) Kebutuhan Fungsional

Sistem yang dikembangkan diharapkan dapat memenuhi kebutuhan fungsional agar dapat digunakan sesuai dengan proses bisnis yang telah berjalan pada sisi admin Liochita Cleaning. Berikut adalah kebutuhan fungsional yang harus dipenuhi oleh sistem.

1. Admin dapat menyimpan, mengubah dan menghapus data pelanggan. Fungsi tersebut bertujuan untuk memastikan bahwa pelanggan yang terdaftar benar-benar pelanggan yang ingin menggunakan jasa Liochita Cleaning.
2. Fitur order yang digunakan untuk menambah, mengubah, membatalkan dan menghapus pesanan yang masuk hanya dapat dikelola oleh Admin.
3. Fitur bahan baku digunakan untuk mengelola bahan baku yang keluar saat memenuhi pengerjaan pesanan dan dan digunakan untuk pembelian bahan baku apabila stok bahan baku kurang.
4. Fitur keuangan digunakan oleh Admin dan Pemilik untuk melihat laporan pemasukan dan laporan pengeluaran secara periodik maupun tahunan.
5. Fitur kelola data digunakan untuk mengelola data wilayah kecamatan dan ongkos kirim yang berisi tarif harga yang dikenakan pelanggan saat melakukan pemesanan berdasarkan kecamatan tempat tinggal pelanggan, data supplier dan nonsupplier berisi rekanan perusahaan yang digunakan untuk memenuhi kebutuhan bahan baku Liochita Cleaning, data produk yang berisi jasa yang ditawarkan Liochita Cleaning.
6. Pemilik dapat menambahkan, menyimpan, mengubah dan menghapus data admin. Fungsi tersebut bertujuan agar Admin yang tidak lagi bekerja pada Liochita Cleaning tidak bisa mengakses kedalam sistem.
7. Pelanggan dapat menambahkan pesanan layanan berupa jasa cuci karpet atau cuci kaca gedung setelah melakukan pendaftaran kedalam sistem.
8. Fitur profil saya digunakan Pelanggan untuk mengelola informasi profil tentang data diri Pelanggan dan mengelola akun yang digunakan Pelanggan untuk masuk kedalam sistem.
9. Fitur order terakhir digunakan Pelanggan untuk melihat status pesanan yang telah di pesan.
10. Fitur histori pesanan digunakan Pelanggan untuk melihat pesanan yang pernah dilakukan oleh Pelanggan.
11. Sistem memiliki fasilitas untuk Pelanggan guna melihat daftar dan detail jasa yang ditawarkan oleh Liochita Cleaning.

4) Kebutuhan Non-Fungsional

Selain kebutuhan fungsional sistem juga harus memenuhi kebutuhan non-fungsional agar sistem yang berjalan dapat digunakan dengan baik oleh pengguna pada sisi admin Liochita Cleaning maupun sisi pelanggan Liochita Cleaning. Berikut merupakan kebutuhan fungsional yang harus dipenuhi oleh sistem.

1. Operasional
 - a. Sistem yang bekerja pada admin dan owner Liochita Cleaning dapat diakses dari perangkat komputer dan tablet.
 - b. Sistem memiliki fitur pemberitahuan pesan kesalahan muncul jika pengguna melakukan kesalahan pemakaian sistem.
 - c. Sistem dapat memperbarui data secara otomatis.
2. Keamanan
 - a. Sistem memiliki fungsi *login* untuk memberikan akses ke pengguna.
 - b. Sistem memiliki fungsi *role* untuk membedakan hak pengguna.
 - c. Sistem mengimplementasikan metode *soft delete* untuk menjaga validasi pengolahan data

B Desain Sistem

SRS yang dihasilkan dari analisis kebutuhan digunakan sebagai bahan perancangan sistem. Kebutuhan fungsional di dalam SRS menentukan bagaimana sistem harus dibangun sehingga mampu memenuhi kebutuhan yang diharapkan. Spesifikasi teknis dibuat pertama kali dalam perancangan untuk menetapkan lingkup pengembangan. Pemodelan dilakukan setelah spesifikasi sistem selesai ditentukan. Penulisan kode belum bisa dimulai sebelum pemodelan selesai dibuat. Penulisan kode tanpa memperhatikan pemodelan akan menimbulkan risiko kegagalan pengembangan. Desain antarmuka dilakukan setelah pemodelan selesai dibuat sehingga sistem mampu dioperasikan pengguna dengan mudah. Pengembangan aplikasi yang mengabaikan desain antarmuka *user friendly* cenderung susah digunakan.

1) Spesifikasi Teknis

1. Arsitektur Sistem

Sistem yang dikembangkan berupa aplikasi berbasis *web* dan aplikasi *mobile*. *Web server* dibutuhkan untuk menjalankan aplikasi tersebut. Data yang diolah sistem disimpan pada basis data. Server basis data juga dibutuhkan untuk menjalankan basis data yang digunakan sistem. Pengguna dapat mengakses sistem melalui *web browser* dengan cara memanggil URL aplikasi.

2. Arsitektur Aplikasi

Deskripsi umum SRS menjelaskan bahwa sistem berupa aplikasi *web* pada sisi admin Liochita Cleaning dan aplikasi perangkat bergerak serta aplikasi web pada pelanggan Liochita Cleaning. Bahasa yang dipakai adalah PHP dengan kerangka kerja *Coderigniter 3.0*.

3. Kebutuhan Perangkat Keras

Kebutuhan perangkat keras bagi pengguna sistem ini memerlukan PC yang dapat menjalankan *web browser*, dari berbagai macam kebutuhan minimal *web browser*, maka perangkat keras yang dibutuhkan pada sisi pengguna adalah sebagai berikut:

- Prosesor : Intel Pentium 4
- RAM : 512 MB
- HDD : 4 GB

Untuk sistem yang berjalan pada perangkat bergerak bagi pengguna pelanggan Liochita Cleaning memerlukan perangkat bergerak dengan spesifikasi minimal RAM 512 MB.

4. Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak merupakan kebutuhan yang diperlukan untuk mendukung implementasi sistem yang dibangun. Kebutuhan aplikasi dibagi menjadi tiga kategori, yaitu :

1. Operasional

Kebutuhan minimum yang diperlukan untuk menjalankan sistem informasi yang akan dibangun ini adalah sebagai berikut :

a. Sistem Operasi

Sistem operasi yang dapat digunakan untuk menjalankan aplikasi ini antara lain *Windows, Mac OS, Linux* dan dibutuhkan sistem operasi *Android* versi 4.1 bagi pengguna di sisi Pelanggan.

b. PHP (PHP : *Hypertext Preprocessor*)

Spesifikasi versi PHP minimum yang dapat digunakan yaitu versi 5.1.6.

c. *Web Browser*

Untuk dapat menjalankan sistem informasi yang akan dibangun, dibutuhkan *web browser* pada sisi Admin maupun Pelanggan. *Web Browser* yang dapat digunakan antara lain *Internet Explorer* versi 7+, *Mozilla Firefox*, *Google Chrome*, *Safari*, dan *Opera*.

d. Basis Data

Basis data yang digunakan untuk menyimpan data menggunakan *MySQL 4.1+*.

2. Antarmuka

Antarmuka merupakan tampilan yang disajikan dari aplikasi yang dibuat. Antarmuka menentukan bagaimana pengguna dapat memahami dan menggunakan sebuah aplikasi. Antarmuka yang baik adalah antarmuka yang mampu memberikan tujuan dengan baik kepada pengguna, dan pewarnaan serta tatanan *layout* membuat pengguna nyaman dalam menggunakan aplikasi. Pada antarmuka aplikasi ini menggunakan kerangka kerja *Twitter Bootstrap*.

3. Keamanan

Keamanan data merupakan salah satu unsur penting dalam pembuatan sebuah sistem terutama dalam menjaga data dari pengguna yang tidak berwenang untuk mengolah data tersebut. Dalam sistem ini, pengamanan dilakukan dengan mengenkripsi *password* pengguna dengan menggunakan metode MD5..

2) Desain ERD

Sistem yang dikembangkan merupakan sistem yang berfungsi untuk mengolah data. Pemodelan perlu dilakukan untuk menentukan data yang diolah. Pemodelan tersebut dilakukan setelah spesifikasi teknis ditentukan. ERD digunakan untuk memodelkan data yang akan disimpan dalam basis data. Langkah yang dilakukan dalam merancang ERD adalah menentukan entitas yang ada, lalu entitas tersebut dihubungkan dalam notasi diagram.

1. Menentukan Entitas

Entitas dapat ditentukan dari SRS. Kebutuhan yang dijelaskan memberikan gambaran data-data yang diolah. Data tersebut merupakan entitas yang akan digunakan dalam merancang ERD. Tabel 1 menunjukkan entitas yang telah ditentukan.

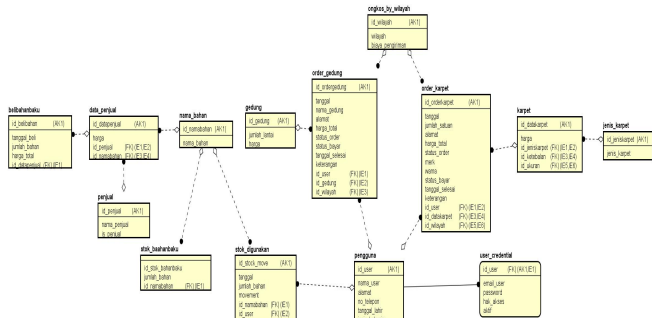
Tabel 1 Tabel Entitas ERD

No	Nama	Keterangan
1.	belibahanbaku	Data pembelian bahan baku
2.	karpel	Data tarif karpel
3.	data penjual	Data keseluruhan penjual
4.	jenis karpel	Data jenis-jenis karpel
5.	gedung	Data tarif gedung
6.	nama bahan	Data nama-nama bahan baku
7.	order gedung	Data pesanan gedung
8.	order karpel	Data pesanan karpel
9.	pengguna	Data pengguna
10.	stok digunakan	Data stok bahan baku yang

		digunakan
11.	stok_bahanbaku	Data stok bahan baku yang tersisa
12.	penjual	Data nama penjual
13.	user_credential	Data akun pengguna
14.	ongkos_by_wilayah	Data wilayah kecamatan di kota semarang dan ongkos kirim

2. Menggambar Diagram

Diagram dapat digambarkan setelah seluruh entitas didefinisikan beserta atributnya. Tujuan dari diagram adalah menjelaskan hubungan antara entitas yang telah ditentukan. Relasi yang digambarkan dapat diperjelas dengan menambahkan informasi kardinalitas antara entitas.



Gambar 1 Diagram ERD Keseluruhan Sistem

Diagram yang menggambarkan keseluruhan sistem tersebut dapat menjelaskan semua data yang diolah sistem. Gambar 1 Menunjukkan ERD keseluruhan sistem.

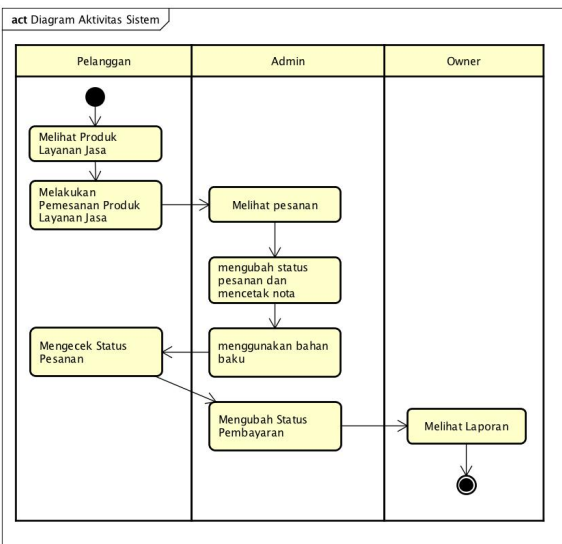
3) Desain UML

Fungsi yang ditunjukkan pada SRS belum memiliki detail yang cukup untuk dikembangkan. Pemodelan UML memberikan ruang untuk mengembangkan detail fungsi yang dibutuhkan oleh sistem sebelum penulisan kode dimulai. Kesalahan yang dilakukan pada pemodelan akan lebih mudah dianalisis dan diperbaiki daripada saat sistem sudah ditulis dengan bahasa pemrograman.

1. Diagram Aktivitas Sistem

Diagram aktivitas dibuat pada awal pemodelan UML untuk memberikan perspektif mengenai proses bisnis yang dilakukan.

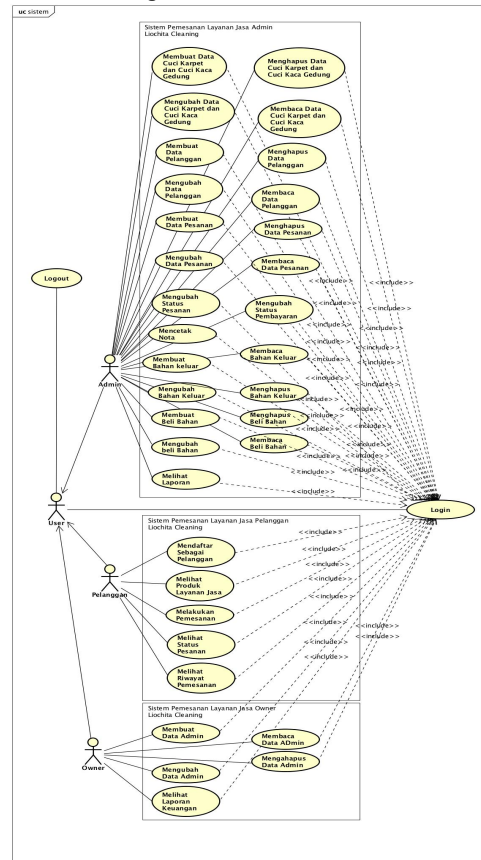
Gambar 2 menunjukkan diagram aktivitas sistem yang dikembangkan. Proses bisnis sistem ini dimulai ketika pelanggan melihat produk layanan jasa dan berakhir hingga admin mengubah status pembayaran.



Gambar 2 Diagram Aktivitas Sistem

2. Diagram Use Case Sistem

Bagaimana pengguna menggunakan sistem dapat digambarkan dengan diagram use case. Diagram use case dibuat berdasarkan skenario yang dilakukan pengguna ketika menggunakan fungsi sistem. Diagram use case terlihat pada Gambar 3.

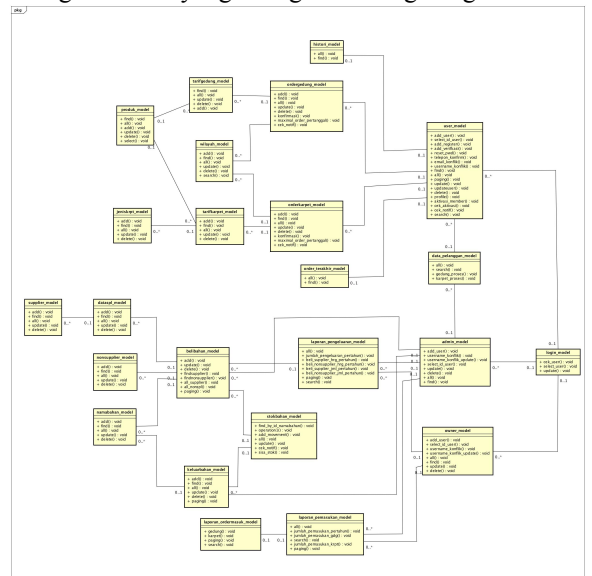


Gambar 3 Diagram Use Case Sistem

Pemodelan diagram use case menunjukkan bahwa sistem terdapat 3 aktor yang menggunakan sistem. Sistem memiliki tiga puluh lima skenario umum.

3. Diagram kelas

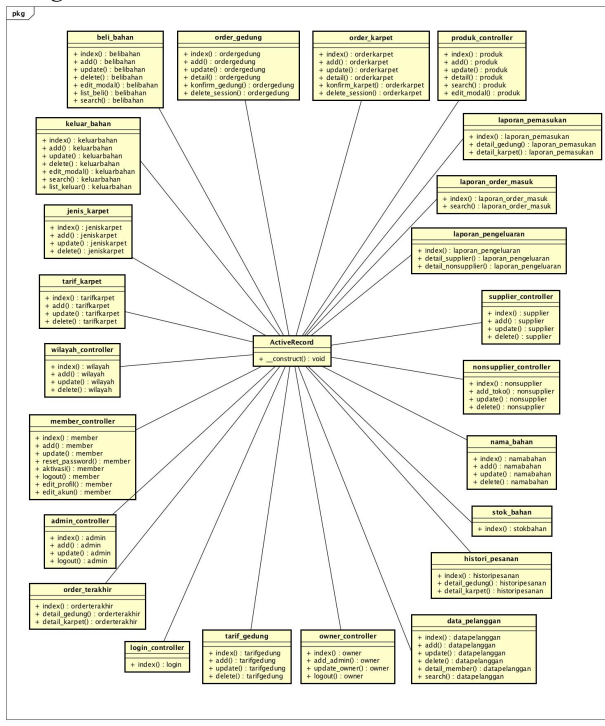
Diagram kelas pada gambar 4 menunjukkan kelas dengan atribut yang saling terhubung dengan kelas lain.



Gambar 4 Diagram kelas tipe model keseluruhan sistem

Kelas yang ada pada Gambar 3.48 tersebut merupakan kelas dengan tipe model yang berfungsi sebagai abstraksi basis data. Semua model yang ada merupakan

extend dari kelas *active record* yang berfungsi sebagai penghubung dengan kelas tipe *controller*. Diagram kelas pada gambar 5 menunjukkan kelas *controller* yang berfungsi sebagai pengolah *Business logic*.



Gambar 5 Diagram kelas tipe *controller* keseluruhan sistem

Kelas *controller* dapat berinteraksi dengan data di dalam basis data dengan *Active Record*. Deskripsi masing – masing kelas ditunjukkan pada tabel 2.

Tabel 2 Deskripsi diagram kelas

N o	Kelas	Deskripsi
1	supplier_model	Model data supplier
2	nonsupplier_model	Model data non supplier
3	keluarbahan_model	Model data bahan baku digunakan
4	namabahan_model	Model data nama bahan
5	belibahan_model	Model data pembelian bahan baku
6	dataspl_model	Model data keseluruhan supplier
7	laporan_pengeluaran_model	Model data laporan pengeluaran
8	laporan_ordermasuk_model	Model data pesanan yang masuk
9	laporan_pemasukan_model	Model data laporan pemasukan
10	stokbahan_model	Model data stok bahan
11	owner_model	Model data pengguna owner
12	admin_model	Model data pengguna admin
13	login_model	Model data login pengguna
14	jeniskrpt_model	Model data jenis karpet
15	tarifkarpet_model	Model data jenis dan tarif cuci karpet
16	orderkarpet_model	Model data pesanan cuci karpet
17	order_terakhir_model	Model data proses pesanan pengguna pelanggan
18	ordergedung_model	Model data pesanan cuci kaca gedung

19	user_model	Model data pengguna pelanggan
20	wilayah_model	Model data biaya pengiriman berdasar wilayah kecamatan pengguna pelanggan
21	tarifgedung_model	Model data jenis dan tarif cuci kaca gedung
22	produk_model	model data cuci karpet dan cuci kaca gedung
23	data_pelanggan_mode l	Model data keseluruhan pelanggan
24	histori_model	Model data histori pesanan pelanggan
25	supplier_controller	<i>Controller</i> yang mengelola data supplier
26	nonsupplier_controller	<i>Controller</i> yang mengelola data nonsupplier
27	keluar_bahan	<i>Controller</i> yang mengelola data bahan baku digunakan
28	nama_bahan	<i>Controller</i> yang mengelola data bahan baku
29	beli_bahan	<i>Controller</i> yang mengelola data pembelian bahan baku
30	laporan_order_masuk	<i>Controller</i> yang mengelola data laporan pesanan masuk
31	laporan_pengeluaran	<i>Controller</i> yang mengelola data laporan pengeluaran
32	laporan_pemasukan	<i>Controller</i> yang mengelola data laporan pemasukan
33	stok_bahan	<i>Controller</i> yang mengelola data stok bahan baku
34	data_pelanggan	<i>Controller</i> yang mengelola data keseluruhan pelanggan
35	owner_controller	<i>Controller</i> yang mengelola data owner
36	admin_controller	<i>Controller</i> yang mengelola data admin
37	login_controller	<i>Controller</i> yang mengelola data login
38	jenis_karpet	<i>Controller</i> yang mengelola data jenis karpet
39	tarif_karpet	<i>Controller</i> yang mengelola data tarif cuci karpet
40	order_karpet	<i>Controller</i> yang mengelola data pesanan cuci karpet
41	order_gedung	<i>Controller</i> yang mengelola data pesanan cuci kaca gedung
42	order_terakhir	<i>Controller</i> yang mengelola data proses pesanan pelanggan
43	member_controller	<i>Controller</i> yang mengelola data pelanggan
44	wilayah_controller	<i>Controller</i> yang mengelola data biaya pengiriman berdasar wilayah kecamatan
45	tarif_gedung	<i>Controller</i> yang mengelola data tarif cuci kaca gedung
46	produk_controller	<i>Controller</i> yang mengelola data cuci karpet dan cuci kaca gedung
47	histori_pesanan	<i>Controller</i> yang mengelola data riwayat pemesanan pelanggan

48	ActiveRecord	Kelas yang digunakan oleh <i>controller</i> untuk mengakses model
----	--------------	---

Pemodelan yang dibuat sudah cukup untuk digunakan menjelaskan bagaimana sistem akan memenuhi kebutuhan sistem. Diagram aktivitas memberikan gambaran utama bagaimana sistem akan digunakan dalam proses bisnis. Diagram *use case* memberikan informasi bagaimana pengguna menggunakan sistem. Diagram *sequence* menampilkan perilaku sistem ketika digunakan pengguna. Diagram kelas menunjukkan struktur sistem.

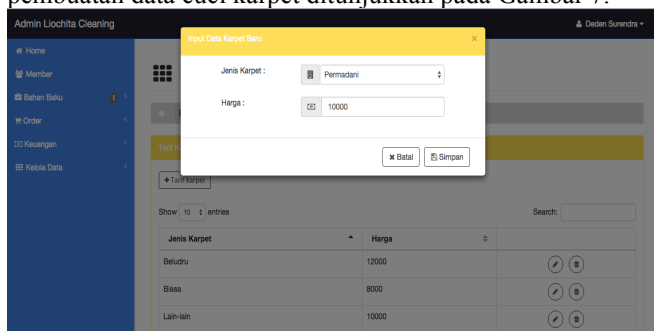
Kode yang digunakan untuk membangun aplikasi bisa dikerjakan setelah tahap ini. Spesifikasi teknis dan pemodelan dapat memberikan gambaran dan arahan bagaimana kode ditulis. Penulisan kode kemudian dikerjakan hingga selesai.

IV PENGUJIAN SISTEM

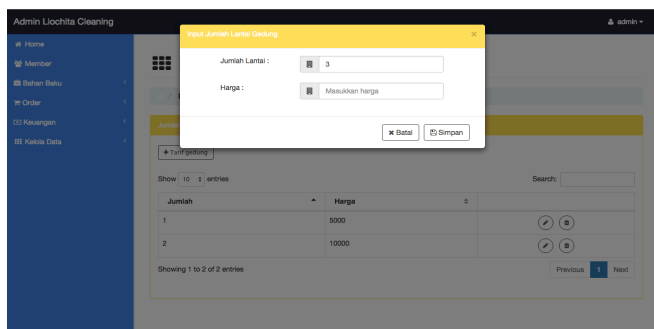
Pengujian integrasi dilakukan untuk mengetahui apakah setiap fungsi yang dimiliki sistem mampu saling terhubung untuk melakukan pekerjaan utama. Pengujian dilakukan karena kesalahan dalam sistem mungkin terjadi walaupun masing – masing fungsi telah diuji. Kesalahan seperti itu umumnya terjadi karena fungsi yang ada tidak terhubung atau tidak memiliki batasan yang sama dengan fungsi lainnya. Pengguna mengeksekusi banyak fungsi dalam melakukan pekerjaannya. Hubungan antar fungsi tersebut adalah bentuk integrasi unit. Skenario buatan atau simulasi pekerjaan dilakukan untuk menjalankan pengujian integrasi. Skenario yang dilakukan pada pengujian ini adalah sebagai

1. Liochita Cleaning memiliki beberapa data cuci karpet dan cuci kaca gedung

Admin Liochita Cleaning membuat data cuci karpet dengan jenis karpet permadani, ketebalan karpet tipis dan ukuran karpet kecil, serta membuat data cuci kaca gedung dengan jumlah lantai 3. Tampilan pembuatan data cuci karpet ditunjukkan pada Gambar 6 dan tampilan untuk pembuatan data cuci karpet ditunjukkan pada Gambar 7.



Gambar 6 Pengujian integrasi membuat data cuci karpet

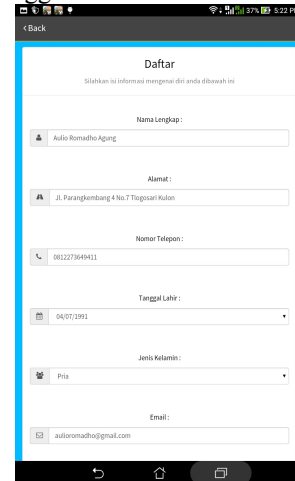


Gambar 7 Pengujian integrasi membuat data cuci kaca gedung

Data cuci karpet dan cuci kaca gedung yang dibuat oleh Admin Liochita Cleaning akan ditampilkan kepada pelanggan saat akan melakukan pemesanan.

2. Pelanggan mendaftar

Pelanggan harus melakukan pendaftaran terlebih dahulu untuk melakukan kegiatan pemesanan layanan jasa, melihat status pesanan, melihat riwayat pemesanan. Pada pengujian ini pelanggan dengan nama Aulio Romadho Agung melakukan pendaftaran menggunakan email aulioromadho@gmail.com email ini digunakan untuk melakukan aktivitas *login*. Gambar 8 menunjukkan proses pendaftaran pelanggan.

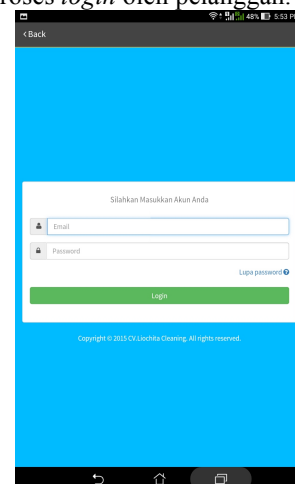


Gambar 8 Pengujian integrasi mendaftar sebagai pelanggan

Pelanggan yang mendaftar pertama kali diharuskan mengaktifkan akun melalui email yang telah didaftarkan kedalam sistem, sistem akan mengirim pesan kepada pelanggan melalui email yang berisi link aktivasi.

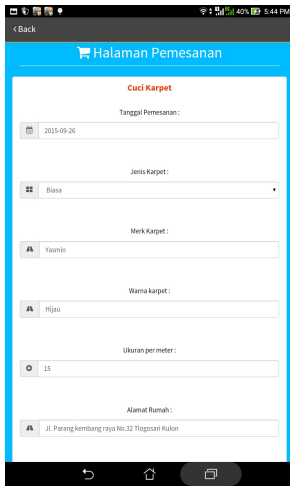
3. Pelanggan melakukan pemesanan layanan jasa

Akun yang telah diaktifkan oleh pelanggan dapat digunakan untuk melakukan kegiatan *login* untuk bisa mengakses fitur pemesanan layanan jasa, melihat status pesanan, melihat riwayat pemesanan. Gambar 9 menunjukkan proses *login* oleh pelanggan.

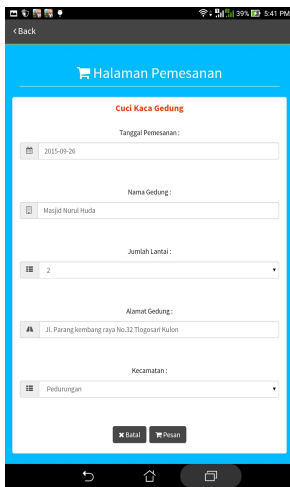


Gambar 9 Pengujian integrasi aktivitas *login*

Email aulioromadho@gmail.com digunakan untuk melakukan *login*. Untuk melakukan kegiatan pemesanan layanan jasa, pelanggan mengisi form data pesanan sesuai jasa yang akan digunakan. Gambar 10 menunjukkan pemesanan layanan jasa cuci kaca karpet dan Gambar 11 menunjukkan pemesanan layanan jasa cuci kaca gedung



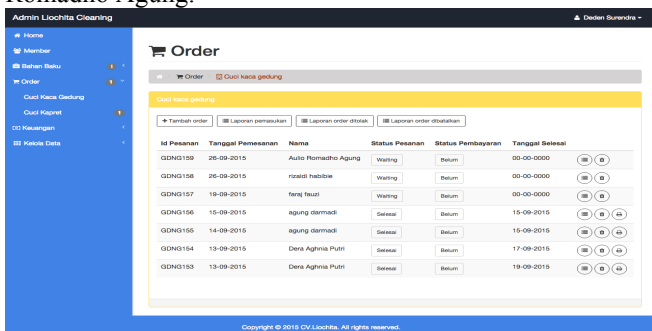
Gambar 10 Pengujian integrasi melakukan pemesanan jasa cuci karpet



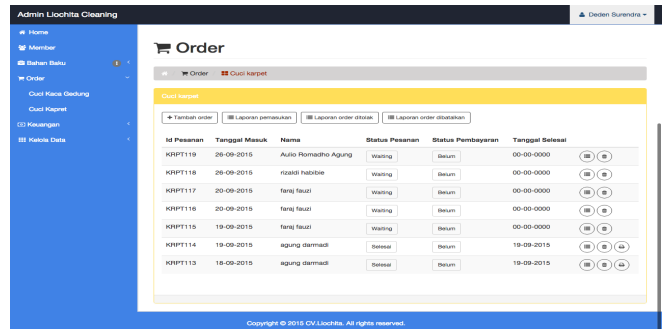
Gambar 11 Pengujian integrasi melakukan pemesanan jasa cuci kaca gedung

4. Admin mengelola pesan yang telah dilakukan oleh pelanggan

Kegiatan Pemesanan yang dilakukan pelanggan akan dilihat oleh admin terlebih dahulu. Gambar 12 menunjukkan pemesanan jasa cuci karpet dan Gambar 13 menunjukkan pemesanan jasa cuci kaca gedung dengan nama Aulio Romadho Agung.

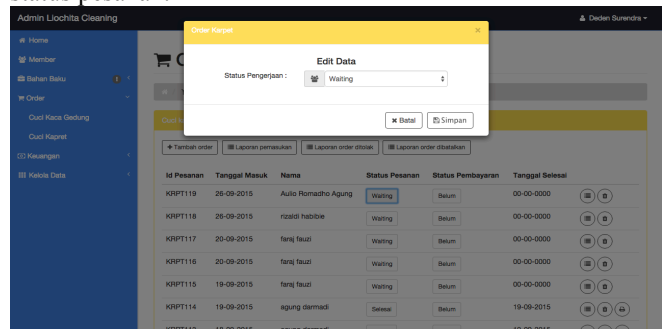


Gambar 12 Pengujian integrasi melihat pesan cuci karpet



Gambar 13 Pengujian integrasi melihat pesan cuci kaca gedung

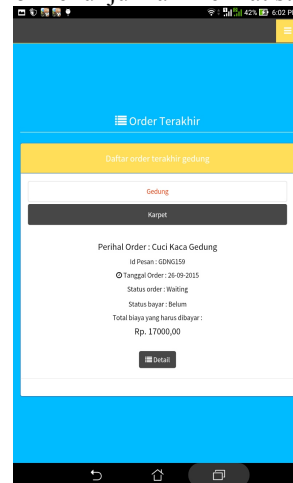
Setelah admin melihat pesan yang masuk, admin mengubah status pesan. Status pesanan akan ditampilkan pada pengguna pelanggan untuk mengetahui status pesan yang telah dipesan. Gambar 14 menunjukkan mengubah status pesan.



Gambar 14 Pengujian integrasi mengubah status pesan

5. Pelanggan melihat status pesan

Pelanggan yang telah melakukan pemesanan dapat melihat status pesan, pelanggan dapat membayar kepada Liochita Cleaning apabila status pesan telah selesai. Gambar 15 menunjukkan melihat status pesan.



Gambar 15 Pengujian integrasi melihat status pesan

6. Admin Liochita Cleaning mencetak nota

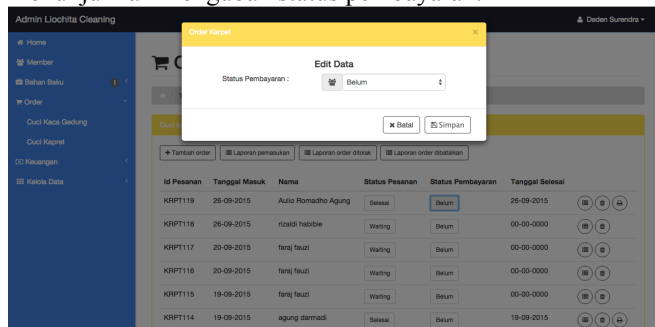
Admin dapat mencetak nota setelah status pesan pelanggan yang telah selesai dikerjakan. Gambar 16 menunjukkan mencetak nota.



Gambar 16 Pengujian integrasi mengubah status pembayaran

7. Admin Liochita Cleaning melakukan perubahan status pembayaran

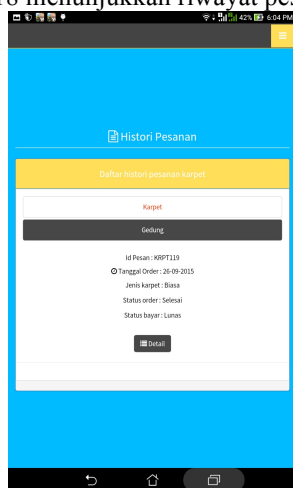
Admin dapat melakukan perubahan status pembayaran setelah pelanggan melakukan pembayaran. Gambar 17 menunjukkan mengubah status pembayaran.



Gambar 17 Pengujian integrasi mengubah status pembayaran

8. Pelanggan melihat riwayat pemesanan

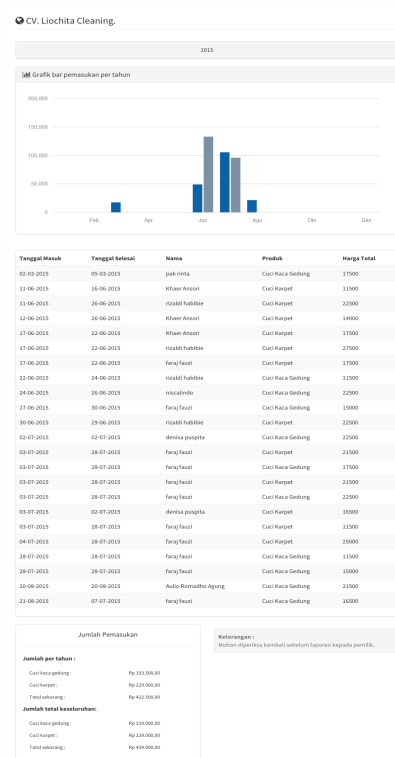
Pesanan yang telah dilakukan dapat dilihat riwayatnya oleh pelanggan melalui menu histori pesanan pada menu profil. Gambar 18 menunjukkan riwayat pesanannya.



Gambar 18 Pengujian integrasi membaca riwayat pemesanan

9. Laporan

Transaksi yang dilakukan oleh pelanggan dan Admin Liochita Cleaning tercatat dalam sistem. Laporan dapat dilihat per tahun. Laporan pada tahun 2015 dapat dilihat pada Gambar 19.



Gambar 19 Pengujian integrasi membaca laporan

Delapan skenario yang dijalankan menjelaskan bahwa sistem dapat mengintegrasikan unit dalam sistem untuk melakukan pekerjaan. Skenario yang diuji ditampilkan detail dengan fungsi terkait pada tabel 3.

Tabel 3 Hasil skenario pengujian integrasi

Skenario	Fungsi Terkait	Analisis
Liochita Cleaning memiliki beberapa data cuci karpet dan kaca gedung	Login, membuat data cuci karpet dan kaca gedung, membuat data supplier, membuat data nonsupplier, membuat data wilayah dan ongkos kirim, membuat data beli bahan nonsupplier, membuat data beli bahan supplier, membuat data beli bahan keluar, membuat data pesanan cuci karpet, membuat data pesanan cuci kaca gedung	Berhasil
Pelanggan mendaftarkan	Membuat data pelanggan	Berhasil
Pelanggan melakukan pemesanan layanan jasa	Login, melakukan pemesanan layanan jasa cuci kaca gedung atau cuci karpet	Berhasil
Admin mengelola pesanan yang telah dilakukan oleh pelanggan	Login, melihat status pesanan, mengubah status pesanan, mencetak nota	Berhasil
Pelanggan melihat status pesanan	Login, melihat status pesanan	Berhasil
Admin Liochita Cleaning melakukan	Login, mengubah status pembayaran	Berhasil

perubahan status pembayaran		
Pelanggan melihat riwayat pemesanan	Membaca riwayat pemesanan	Berhasil
Laporan	Login, membaca data laporan	Berhasil

Tahap pengujian yang dilakukan menunjukkan sistem memiliki fungsi yang bekerja dengan benar.

V PENUTUP

A Kesimpulan

Selama pengembangan sistem ini terdapat beberapa hal yang bisa disimpulkan. Kesimpulan yang didapatkan antara lain:

1. Sistem informasi dapat mengolah data pelanggan sesuai kebutuhan dengan lebih baik dibanding metode sebelumnya yang dalam pencatatan data pelanggan dan data pesanan masih menjadi satu yaitu data pemasukan sehingga perusahaan kurang mengetahui data pelanggan secara terperinci.
2. Sistem informasi dapat mengolah data riwayat pesanan sesuai kebutuhan dengan risiko data hilang atau rusak lebih kecil daripada metode sebelumnya yang dalam melakukan pencatatan semua data masih menggunakan kertas hal ini memiliki resiko kertas dan data yang telah tercatat tersebut hilang.
3. Sistem dapat digunakan oleh pengguna dengan mudah karena sistem dirancang untuk bekerja sesuai dengan bisnis proses yang ada sehingga mempermudah proses adaptasi.
4. Sistem yang dikembangkan dapat memfasilitasi kegiatan pemesanan jasa oleh pelanggan dan pelayanan oleh karyawan sesuai kebutuhan perusahaan Liochita Cleaning.
5. Sistem yang dikembangkan dapat menampilkan data pesanan per tahun untuk kemudian dibuat menjadi laporan pemasukan.

B Saran

1. Sistem informasi ini memiliki banyak aspek untuk dikembangkan seperti integrasi dengan sistem manajemen karyawan.
2. Fitur absensi karyawan dan gaji karyawan pada sistem manajemen karyawan belum bisa diimplementasikan karena dibutuhkan perancangan dan pengembangan sistem yang berbeda.

DAFTAR PUSTAKA

- [1] Bell, C. (2012). *Expert MySQL*. Apress.
- [2] Fauzan, Milady Reza. "Perancangan Sistem Pemesanan Produk Berbasis Web Pada CV. Hanif Niaga Group," 2008.
- [3] Hidayat, Rakasiwi Syarief. "Perancangan Sistem Informasi Pelayanan Jasa Laundry Pada Hello Laundry," 2014.
- [4] Jhon Burch, Gary Grudnitski. 1989. *Information System Theory and Practice*. New York: John Wiley & Sons.
- [5] M. Huda dan B. Komputer, *Membuat Aplikasi Database Dengan Java, MySql dan NetBeans*, Jakarta: Elex Media Komputindo, 2010.
- [6] Spurlock Jack, *Bootstrap Responsive Web Development*, California: O'Reilly Media, Inc, 2013.

- [7] Wardana. 2000. *Menjadi Master PHP dengan Framework Codeigniter*. Jakarta: PT Elex Media komputindo.
- [8] Winina, Ardiana. "Sistem Informasi Pelayanan Salon Kecantikan Berbasis Web Dengan Menggunakan Framework Codeigniter," 2011.