

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Commanditaire vennootschap (CV) atau persekutuan komanditer adalah salah satu bentuk badan usaha yang peserta persekutuan dagang tersebut dengan memberikan Sebagian modal saja tanpa turut menjalankan perusahaan itu atau bisa juga ikut dalam kepengurusan. CV didirikan oleh satu orang atau beberapa orang yang bertanggung jawab secara pribadi untuk keseluruhan CV bersama dengan satu atau beberapa orang yang melepas modal[1].

Banyak perusahaan di Kecamatan Cililin, Kabupaten Bandung Barat yang belum menerapkan teknologi *komputerisasi* untuk menjalankan proses bisnisnya. Salah satunya adalah CV Medal Rasa yang terletak di Jalan Kaum, Kecamatan Cililin, Kabupaten Bandung Barat. Perusahaan ini sudah berdiri sejak 1990 yang didirikan oleh almarhum Bapak Ismail, pada awalnya perusahaan ini merupakan UMKM yang berada di Desa Cililin yang dinamai Medal Rasa. Perubahan nama ke CV mulai bulan Juli tahun 2022. Secara umum perusahaan ini merupakan perusahaan yang bergerak di bidang manufaktur. CV Medal Rasa menjual makanan khas oleh-oleh Cililin yaitu wajit.

CV Medal Rasa hanya melakukan produksi wajit ketika ada yang memesan. Pada saat covid-19 penjualan mengalami penurunan dan berhenti memproduksi wajit karena tidak ada pesanan. Dibutuhkan *platform Marketplace* digital berbasis web dapat meningkat penjualan dan menambah tempat untuk berjualan. Marketplace adalah platform yang menjadi perantara antara penjual dan pembeli di internet. Website marketplace bertindak sebagai pihak ketiga dalam transaksi online dengan menyediakan tempat berjualan dan fasilitas pembayaran[2]. Setiap transaksi penjualan yang ada di CV Medal Rasa masih menggunakan pencatatan akuntansi manual yang beresiko kesalahan dalam pencatatan dan hilang atau tercecer catatan penjualannya.

Berdasarkan masalah tersebut maka solusi yang ditawarkan adalah membuat “Aplikasi *Marketplace* untuk Penjualan dan Pembayaran Menggunakan *Payment Gateway*”. Dengan adanya aplikasi ini CV Medal Rasa dapat mengolah data jurnal umum, buku besar, laporan penjualan secara otomatis. Kemudian dapat membuat data produk, dan data pelanggan secara rapih di aplikasi.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, maka dapat diperoleh rumusan masalah sebagai berikut.

- a. Bagaimana membuat aplikasi pembayaran berbasis *web* yang terintegrasi dengan *payment gateway*?
- b. Bagaimana menangani penjualan produk menggunakan Aplikasi *Marketplace*?
- c. Bagaimana membuat aplikasi berbasis *web* yang menghasilkan laporan penjualan *online*?

1.3 Tujuan

Adapun tujuan yang ingin dicapai berdasarkan rumusan masalah diatas sebagai berikut.

- a. Dapat membuat aplikasi pembayaran berbasis *web* yang terintegrasi dengan *payment gateway*.
- b. Dapat menangani penjualan produk menggunakan aplikasi *marketplace* berbasis web.
- c. Dapat membuat aplikasi berbasis *web* menghasilkan laporan penjualan *online*.

1.4 Batasan Masalah

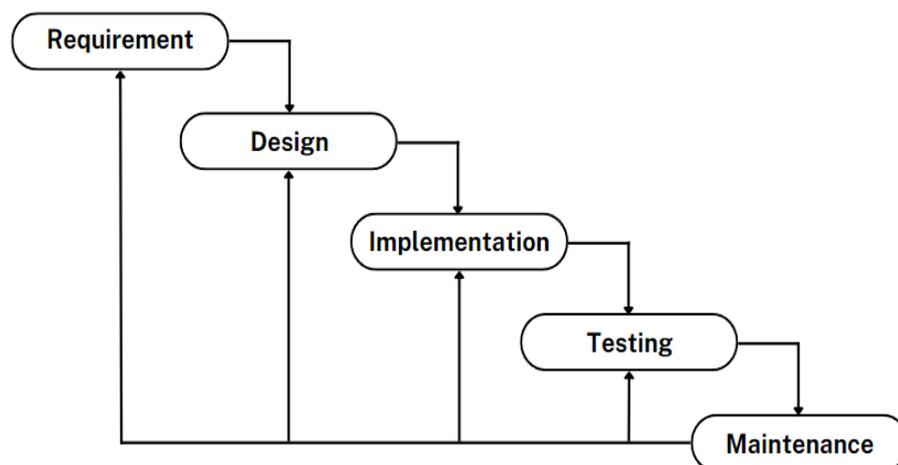
Batasan masalah dari aplikasi ini adalah sebagai berikut.

- a. Aplikasi ini tidak menangani *kredit* atau utang.

- b. Aplikasi ini tidak menangani pembatalan pembelian.
- c. Aplikasi ini tidak menangani perhitungan pajak.
- d. Aplikasi ini tidak menangani perhitungan ongkos kirim.
- e. Aplikasi ini tidak menangani *tracking* pengiriman.
- f. Aplikasi ini diuji menggunakan metode black box testing.

1.5 Metode Pengerjaan

Metode yang digunakan yaitu *Software Development Life Cycle* (SDLC) dengan model *waterwall*. *Software Development Life Cycle* (SDLC) adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan tahapan-tahapan yang harus dikerjakan. Model *Waterfall* disebut model sekuensial liner atau alur hidup klasik dimana kemajuan teknologi terus mengalir kebawah. Metode ini berurutan dari satu tahap ke tahap berikutnya dimulai dari analisis, desain, implementasi, pengujian, dan pemeliharaan.



Gambar 1- 1 Metode Waterfall

a. Requirement

Dalam tahapan requirement melakukan pencarian informasi yang penting Pada kepada CV Medal Rasa agar dapat mengidentifikasi sistem yang akan dibuat sesuai dengan kebutuhan.

b. Design

Pada tahap desain dilakukan adaptasi dari analisis sebelumnya. Desain Sistem membantu dalam menentukan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan. Mulai dari mendefinisikan proses bisnis menggunakan BPMN lalu penggambaran aplikasi ini menggunakan *Use Case Diagram, Activity Diagram, Sequence Diagram, dan Class Diagram*. Kemudian *Entity Relationship Diagram* (ERD) untuk gambaran grafis databasenya.

c. Implementation

Pada tahap ini sistem pertama kali dikembangkan di program kecil yang disebut unit, yang terintegrasi dalam tahap selanjutnya. Pengkodean pada aplikasi ini menggunakan bahasa *Hypertext PreProcessor* (PHP) sebagai bahasa pemrograman untuk membuat web menjadi dinamis, MySQL untuk database, *CodeIgniter* sebagai *framework* PHP untuk kerangka kerja PHP dengan model MVC (*Model, View, Controller*) untuk membangun situs web dinamis dengan menggunakan PHP, Bootstrap sebagai *framework* CSS untuk kebutuhan pembuatan tampilan desain visual dari aplikasi web, HTML dan CSS untuk halaman web dan mengatur tampilannya serta Midtrans alat pembantu payment gatewaynya. Sehingga semua fungsi dapat dijalankan oleh pengguna

d. Testing

Testing merupakan tahap untuk memeriksa apakah aplikasi ini terdapat kesalahan dalam desain maupun pengkodean serta apakah semua fungsi pada aplikasi ini berjalan dengan baik. *Testing* menggunakan metode *Black Box Testing*.

e. *Maintenance*

Maintenance merupakan tahap terakhir. *Software* yang dijalankan serta perlunya pemeliharaan secara berkala seperti mencadangkan database, dan update fitur pada sistem. Tetapi pada tahap ini penulis tidak melakukan tahap terakhir.

1.6 Jadwal Pengerjaan

Berikut adalah jadwal pembuatan proyek akhir:

Tabel 1-1 Jadwal Pengerjaan Proyek Akhir

Kegiatan	2022			2023						
	Oktober	November	Desember	Januari	Februari	Maret	April	Mei	Juni	Juli
Requirement										
Design										
Implementation										
Testing										
Maintenance										