

BAB I. PENDAHULUAN

1.1 Latar Belakang

Usaha Mikro, Kecil, dan Menengah atau biasa disebut UMKM, saat ini menjadi banyak diminati di Indonesia. Harganya yang terjangkau membuat usaha kecil menjadi favorit di kalangan semua orang. Produk dengan kualitas dan harga yang bersahabat selalu diminati oleh semua kalangan. Selain sebagai tempat penjualan produk, pusat UMKM juga berfungsi sebagai sumber informasi mengenai asal-usul setiap produk yang ada di lokasi tersebut. Ini memungkinkan pembeli untuk mengunjungi langsung tempat asal usaha produk yang mereka beli.

Namun, di tengah era digital saat ini, usaha-usaha tersebut masih memiliki banyak hambatan yang menyebabkan mereka merasa kesulitan. Hal ini disebabkan karena banyaknya produk-produk dari perusahaan besar yang sudah melakukan penjualan online. Saat ini Upaya pemasaran produk UMKM yang didukung oleh pemerintah masih sangat terbatas karena hanya ada pada pusat-pusat UMKM di setiap kota. Situasi ini menghambat potensi pemasaran produk UMKM yang seharusnya dapat mencapai pasar di luar kota bahkan hingga tingkat ekspor.

Dengan latar belakang permasalahan di atas, akan dibuat sebuah aplikasi berbasis web yang berfungsi sebagai platform pemasaran dan penjualan produk UMKM. Aplikasi yang sedang dikembangkan diharapkan mampu mengelola semua data terkait usaha mikro, kecil, dan menengah yang terdaftar. Data-data yang dimaksud mencakup informasi tentang usaha, detail produk, dan juga lokasi usaha. Aplikasi tersebut juga didesain memiliki kemampuan untuk menyimpan data para pelanggan saat mereka melakukan transaksi pembelian dengan para pengusaha UMKM yang terdaftar (Setiawan, 2018).

Istana *Gordyn* Lampung adalah salah satu UMKM yang menjual kebutuhan interior dan dekorasi yang berdiri sejak tahun 2014. UMKM ini didirikan oleh seorang pengusaha bernama Bapak Asrizal. Istana *Gordyn* Lampung beralamatkan di Bambu Kuning *Trade Center* Lantai 2 Blok F No. 17 - 18, Jalan Imam Bonjol, Klp. Tiga, Tanjung Karang - Bandar Lampung, Kota Bandar Lampung, Lampung 35111.

Istana *Gordyn* Lampung saat berdiri hingga tahun 2019 mendapat pendapatan yang stabil, hingga tahun 2020 Istana *Gordyn* Lampung terkena dampak dari *pandemi covid-19* sehingga pendapatannya turun drastis. Hingga saat ini, pemilik usaha masih menjalankan transaksi usahanya secara manual dengan berinteraksi langsung atau melalui telepon. Namun, sering kali muncul kendala seperti kesulitan dalam menjaga janji pertemuan atau ketidakterediaan saat dihubungi. Dari sisi penjual, mereka diharuskan selalu hadir di lokasi untuk memenuhi pesanan dari pelanggan (Nugraha & Yulianeu, 2018).

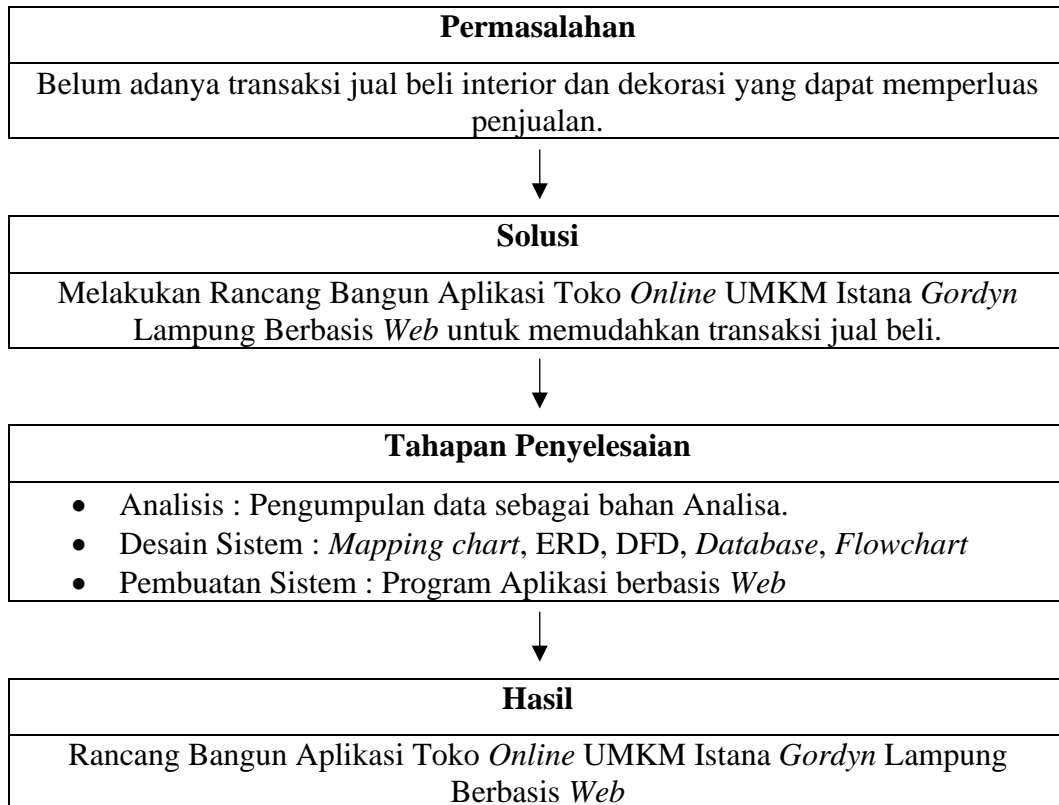
Saat ini, segala aktivitas manusia tidak terpisahkan dari kemajuan teknologi yang telah mengakibatkan perubahan yang penting di berbagai aspek, termasuk sosial, ekonomi, dan budaya. Pemanfaatan teknologi komunikasi dan informasi telah mempercepat perkembangan ini. Demikian juga, dalam hal peminjaman dan pengelolaan data, teknologi yang sedang berkembang dimanfaatkan untuk memberikan kemudahan kepada pengguna (Supriyanto & Ismawati, 2019).

1.2 Tujuan

Membuat Rancang Bangun Aplikasi Toko Online UMKM Istana *Gordyn* Lampung Berbasis Web adalah tujuan utama dalam pembuatan tugas akhir ini.

1.3 Kerangka Pemikiran

Berdasarkan permasalahan diatas Istana *Gordyn* Lampung memerlukan sebuah Aplikasi Toko *Online* berbasis *Web* untuk mengatasi masalah tersebut. “Rancang Bangun Aplikasi Toko *Online* UMKM Istana *Gordyn* Lampung Berbasis *Web*” dapat berguna untuk melakukan transaksi jual beli interior dan dekorasi yang memudahkan pembeli untuk memilih dan membeli produk yang diinginkan, serta melakukan komunikasi dari pembeli kepada penjual.



Gambar 1. Kerangka Pemikiran

1.4 Kontribusi

Dengan dibuatnya Rancang Bangun Aplikasi Toko *Online* UMKM Istana *Gordyn* Lampung Berbasis *Web* untuk Istana *Gordyn* Lampung. Diharapkan dapat memberikan kontribusi sebagai berikut :

- 1) Istana *Gordyn* Lampung (Penjual):
 - a. Memudahkan untuk transaksi jual beli interior dan dekorasi yang lebih luas.
 - b. Mampu mengelola produk yang akan dijual. Proses kelola produk dimulai dari menambah produk baru, mengubah informasi produk, dan menambah ketersediaan produk.
- 2) Pembeli
 - a. Mempermudah pembeli untuk menemukan produk yang sesuai dengan keinginan pembeli.
 - b. Dapat melakukan proses transaksi sehingga diharapkan akan membangun kepercayaan pelanggan dalam melakukan transaksi aman dalam aplikasi.

BAB II. TINJAUAN PUSTAKA

2.1 Aplikasi

Aplikasi adalah suatu program yang disiapkan dan siap digunakan untuk menjalankan perintah yang diberikan oleh pengguna aplikasi. Tujuannya adalah untuk menghasilkan hasil yang lebih akurat berdasarkan tujuan pembuatan aplikasi. Aplikasi augmented reality (AR) adalah solusi masalah yang menggunakan salah satu dari beberapa teknik pemrosesan data aplikasi, biasanya berdasarkan perhitungan yang diharapkan atau pemrosesan data yang diinginkan. Aplikasi ini bekerja dengan sangat baik dan terintegrasi semaksimal mungkin. Aplikasi adalah alat IT yang siap digunakan oleh pengguna (Wahyuni & Irawan, 2020)

2.1.1 Aplikasi Berbasis Web

Aplikasi berbasis *Web* menyatakan pada perangkat lunak yang dapat dijangkau melalui pencari (*browser*) dengan bantuan jaringan internet atau intranet. Aplikasi Web ini merupakan perangkat lunak yang dikembangkan menggunakan bahasa pemrograman seperti HTML, *JavaScript*, CSS, PHP, dan bahasa pemrograman lainnya. Salah satu keunggulan utama dari aplikasi berbasis *Web* adalah kemampuannya untuk diakses melalui berbagai *platform*, serta kemampuan untuk menyajikan informasi melalui pencari *web* pada berbagai sistem operasi yang berbeda. Aplikasi *Web* ini dapat digunakan untuk berbagai tujuan atau keperluan yang beragam (Mandang, 2020).

2.2 Komponen Pendukung Aplikasi

CodeIgniter merupakan salah satu *framework* yang biasa digunakan oleh para pengembang PHP. *Framework Codeigniter* ini sangat populer karena terbukti membantu mengembangkan berbagai jenis *website* berbasis PHP dengan lebih mudah, cepat dan efektif. Salah satu alasan banyaknya *codeigniter* dipakai karena memiliki *library* yang lebih beragam dari sebagian *framework* lainnya. Selain itu, *framework* ini dikenal karena ringan dan tidak mengonsumsi *resource* banyak di situs *web*. *Framework* ini menerapkan model MVC (*model*, *view*, dan *controller*) yang cocok digunakan untuk membangun situs *web* dinamis dengan lebih cepat dan

efisien. Konsep MVC melibatkan pemisahan komponen utama menjadi tiga bagian yang membentuk pola dalam sebuah aplikasi (Safira, 2021).

XAMPP merupakan penyedia perangkat lunak dalam satu paket aplikasi. Setelah menginstal XAMPP, pengguna tidak perlu lagi menginstalasi dan mengkonfigurasi *Web Server Apache*, PHP, dan MySQL secara manual. XAMPP akan secara otomatis melakukan instalasi dan konfigurasi ini atau akan dikonfigurasi secara otomatis. XAMPP merupakan paket perangkat lunak PHP yang bersumber terbuka dan dikembangkan oleh komunitas *open source*. Penggunaan XAMPP tidak memerlukan instalasi program tambahan, karena XAMPP telah menyediakan semua yang dibutuhkan. Komponen yang disertakan dalam XAMPP meliputi *Apache*, MySQL, PHP, FileZilla FTP Server, PHPMyAdmin, dan lain-lain (Santoso & Nurmalina, 2017).

MySQL merupakan basis data yang sering digunakan dalam pengembangan aplikasi *web* dinamis. MySQL merupakan salah satu *Relational Database Management System (RDBMS)*. MySQL dapat digunakan bersamaan dengan bahasa pemrograman seperti PHP. MySQL juga menggunakan bahasa SQL (*Structured Query Language*) yang sederhana dan menggunakan *escape character* yang sama dengan PHP. RDBMS sendiri adalah perangkat lunak yang memungkinkan pengguna basis data untuk membuat, mengelola, dan mengakses data dalam model relasional, di mana tabel-tabel dalam basis data memiliki hubungan satu sama lain (Hermiati, 2021).

2.3 Model View Controller (MVC)

Model View Controller (MVC) merupakan sebuah pola desain arsitektur dalam pengembangan sistem secara terstruktur yang terdiri dari tiga komponen utama didalamnya. Komponen pertama adalah yang bertanggung jawab untuk mengelola dan berinteraksi langsung dengan *database*, yang disebut sebagai "*model*." Komponen kedua adalah yang akan menampilkan informasi kepada pengguna, yang dikenal sebagai "*view*." Sementara komponen ketiga adalah yang menghubungkan antara *model* dan *view* dalam proses permintaan yang diberikan oleh pengguna, yang disebut sebagai "*controller*". (Putri Aprilia, 2021). Pengertian dari MVC dibagi menjadi 3, yaitu:

2.3.1 Model

Model ini terkait dengan aspek *database* dan interaksi. Biasanya, *model* digunakan untuk menggambarkan struktur data aplikasi, yang dapat berupa *database*, *file teks*, *file XML*, atau layanan *web*. *Model* ini bertugas untuk mendefinisikan informasi apa yang perlu ada dalam aplikasi. Ketika kondisi aplikasi berubah, *model* mengirimkan informasi kepada *view* dan *controller*.

2.3.2 View

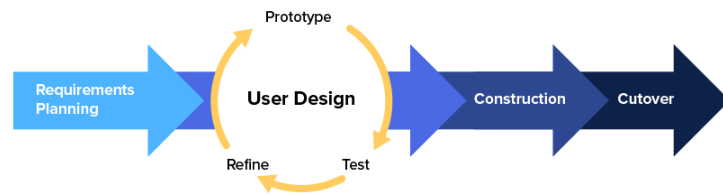
View merupakan bagian yang terkait secara langsung dengan apa yang dilihat oleh pengguna atau yang terkait dengan logika presentasi. Dalam konteks aplikasi *web*, *view* adalah representasi tampilan dalam bentuk *file template* HTML yang diatur oleh sebuah *controller*. *View* bertanggung jawab dalam menentukan tampilan yang akan ditampilkan kepada pengguna, dan juga menerima informasi tampilan dari *model*. Namun, *view* tidak memiliki izin untuk mengakses *model* secara langsung.

2.3.3 Controller

Pada tahap akhir dalam konsep MVC, terdapat bagian yang disebut sebagai *controller*. *Controller* ini mengandung logika yang bertanggung jawab atas pembaruan yang dapat dilakukan terhadap *model* atau *view* sebagai tanggapan terhadap tindakan yang dilakukan oleh pengguna aplikasi. Peran *controller* melibatkan berbagai tugas, seperti menyediakan variabel yang akan ditampilkan di tampilan, mengatasi kesalahan, mengelola proses logika aplikasi, melakukan validasi, dan memanggil *model* untuk mengakses *database*. Namun, dalam beberapa situasi, *controller* juga dapat langsung mengupdate tampilan tanpa melibatkan *model*. (Mauditalani, 2022).

2.4 Rapid Application Development (RAD)

Rapid Application Development (RAD) adalah suatu model proses pengembangan perangkat lunak yang dilakukan secara bertahap dalam jangka waktu yang singkat. RAD menggunakan metode iteratif, yang berarti model kerja sistem dibangun dari awal, selama tahap pengembangan untuk mengidentifikasi kebutuhan pengguna (Sagala, 2018).



Gambar 2. Model RAD
Sumber : (Dirjen, 2018)

2.4.1 *Requirements Planning*

Dalam tahap ini, pengguna dan pengembang berinteraksi langsung untuk memahami dan mengatasi masalah yang ada, serta menentukan persyaratan yang dibutuhkan untuk membuat sebuah sistem aplikasi. Langkah awal ini bertujuan untuk mencegah terjadinya kesalahan komunikasi antara pengguna dan pengembang.

2.4.2 *RAD Design Workshop*

Tahap *design* dirancang untuk menghasilkan suatu rencana yang akan diajukan agar dapat memenuhi kebutuhan, berjalan sesuai rencana, dan diharapkan mampu mengatasi masalah yang muncul. Perancangan sistem yang dijelaskan dalam penelitian ini menggunakan alat bantu seperti *Mapping Chart*, *Flowchart*, ERD, dan DFD.

2.4.3 *Construction*

Pada tahap ini, proses dimulai dengan pembuatan sistem sesuai rencana yang telah disusun. Ini melibatkan penulisan kode program, yang sering dikenal sebagai *coding*, untuk mengubah desain sistem menjadi sebuah aplikasi yang telah direncanakan dan siap untuk digunakan.





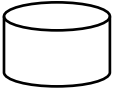
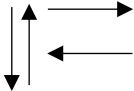


2.4.4 *Cutover*

Tahap ini melibatkan pengujian menyeluruh dari keseluruhan sistem yang telah dibangun. Semua komponen sistem harus diuji dengan cermat menggunakan metode *Black Box Testing* guna mengurangi risiko kesalahan dalam sistem. (Nurman Hidayat & Kusuma Hati, 2021).

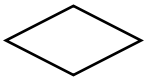
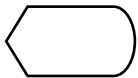
2.5 Mapping Chart

Mapping Chart adalah bagan alir yang menggambarkan alur dokumen dan formulir, termasuk duplikatnya. Diagram alur dokumen ini menggunakan notasi yang identik dengan aliran sistem yang digunakan dalam diagram alur dokumen (Verawati & Liksha, 2018). Simbol *Mapping Chart* adalah sebagai berikut:

Tabel 1. Simbol-simbol *Mapping Chart*

Simbol	Nama	Fungsi
	Dokumen	Simbol ini mencerminkan makna untuk dokumen yang berfungsi sebagai input dan output.
	Manual	Menggambarkan tindakan atau pekerjaan yang dilakukan secara manual.
	Proses	Mengindikasikan aktivitas proses dalam perangkat lunak komputer.
	Keyboard	Menggambarkan masukan yang memanfaatkan papan ketik daring.
	<i>Harddisk</i>	Menggambarkan penggunaan harddisk dalam proses masukan maupun keluaran.
	Garis Alur	Menggambarkan aliran dalam setiap tahapan proses.
	Penghubung	Mengindikasikan tautan ke halaman yang sama atau ke halaman lain.
	Arsip	Menggambarkan penyimpanan dokumen tanpa melibatkan komputer.

Tabel 1 (Lanjutan)


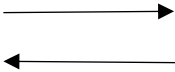
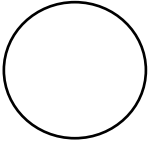

	Keputusan	Digunakan untuk suatu penyeleksian kondisi di dalam program
	Display	Menyatakan peralatan <i>output</i> yang digunakan.

Sumber : (Agus Suratna, 2021)

2.6 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) seringkali digunakan untuk mengilustrasikan sistem yang sedang berjalan atau sistem yang direncanakan secara logis, tanpa melibatkan serta mempertimbangkan lingkungan fisik tempat aliran data atau tempat penyimpanan data (Dewi, 2020). Dalam membuat *data flow diagram* (DFD) digunakan beberapa simbol, antara lain:

Tabel 2. Simbol-simbol DFD


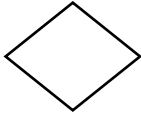
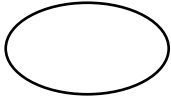

Simbol	Nama	Fungsi
	<i>External Entity</i>	Simbol ini merepresentasikan orang, organisasi diluar sistem yang berhubungan dengan sistem secara langsung.
	<i>Data Flow</i>	Menggambarkan aliran data dari satu bagian ke bagian lain secara sistematis.
	<i>Process</i>	Menandakan sebuah proses dengan tahapan-tahapan bisnis dengan bantuan komputerisasi.
	<i>Data Store</i>	Menggambarkan sebuah kumpulan dari beberapa data yang tersimpan dari sebuah proses.

Sumber : (Pranata, 2020)

2.7 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) merupakan langkah mengorganisir data bisnis dengan menentukan jenis entitas beserta hubungannya dengan memanfaatkan data yang tersimpan di sistem. ERD juga mengilustrasikan koneksi antara suatu entitas dengan sejumlah atribut dan entitas lainnya dalam kerangka sistem terintegrasi. Perancang sistem menggunakan model data ini sebagai landasan yang akan di implementasikan sebagai sebuah *database* (Siregar, 2019). Dalam membuat *Entity Relationship Diagram* (ERD) digunakan beberapa simbol, antara lain:

Tabel 3. Simbol-simbol ERD



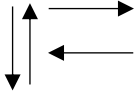
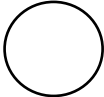

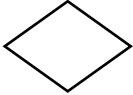
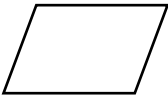





Simbol	Nama	Fungsi
	Entitas	Dijabarkan sebagai objek-objek unik dalam sebuah sistem.
	Relasi	Merepresentasikan hubungan yaitu <i>one to one</i> , <i>one to many</i> , <i>many to many</i>
	Atribut	Penjelasan secara mendetail dari sebuah entitas.
	Garis	Merupakan simbol penghubung antar entitas, proses, dan relasinya.

Sumber : (Siregar, 2019)

2.8 Flowchart

Flowchart adalah urutan kegiatan yang digambarkan dalam bentuk simbol-simbol. Flowchart juga didefinisikan sebagai diagram yang menggambarkan urutan suatu proses dengan memanfaatkan anotasi seperti kotak, panah, oval, berlian, dan elemen lainnya (Siregar, 2019). Dalam membuat *Flowchart* digunakan beberapa simbol, antara lain:

Tabel 4. Simbol-simbol Flowchart

Simbol	Nama	Fungsi
	<i>Terminator</i>	Simbol pernyataan sebuah awal atau akhir dalam sebuah program.
	Proses	Menggambarkan sebuah proses terkomputerisasi
	<i>Flow Line</i>	Didefinisikan sebagai garis penghubung antar simbol.
	<i>On-page Reference</i>	Simbol yang menyatakan lanjutan proses dalam lembar yang sama.
	<i>Off-page Reference</i>	Simbol yang menyatakan lanjutan proses dalam lembar yang berbeda.
	<i>Decision</i>	Simbol yang menyatakan atau menghasilkan pilihan ya atau tidak.
	<i>Input/output</i>	Menggambarkan sebuah masukan atau keluaran dengan atau sebuah peralatan.
	<i>Manual Operation</i>	Menyatakan sebuah operasi atau proses yang dilakukan manual.
	<i>Document</i>	Inputan atau keluaran yang berbentuk fisik/di cetak ke dalam media kertas atau lainnya.
	<i>Predefine Proses</i>	Simbol yang menggambarkan sub-program.
	<i>Display</i>	Menyatakan hasil keluaran yang ditampilkan di sebuah monitor.
	<i>Preparation</i>	Simbol yang menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberikan nilai awal.

Sumber : (Rony Setiawan, 2021)

2.9 Pengujian *Black Box Testing*

Dalam tahap pengujian, diterapkannya metode pengujian *black box testing* yang berfokus pada pengujian fungsional sebuah sistem. *Black box testing* adalah jenis pengujian sistem tanpa memperhatikan struktur logika internal perangkat lunak. Pada dasarnya, tujuan dari *black box testing* adalah untuk memastikan bahwa aplikasi berjalan dengan benar. Setiap fungsi dalam sistem informasi diuji untuk memeriksa apakah berjalan dengan baik atau tidak (Melinda, 2018).