

## BAB II LANDASAN TEORI

### 2.1. Tinjauan Pustaka

Berikut adalah beberapa penelitian yang diperlukan tinjauan pustaka yang diambil dari beberapa jurnal penelitian yang berkaitan dengan judul penelitian ini dan pokok bahasan, berikut ini merupakan tinjauan pustaka yang diambil dapat dilihat pada tabel 2.1.

**Tabel 2. 1** Tinjauan Pustaka

No	Nama (Tahun)	Judul	Metode	Hasil
1	(Saharna and Rukun, 2019)	Perancangan Sistem E-Commerce Berbasis Web Pada Toko Indah Surya Furniture	Metode pengujian Black Box Testing	Sistem e-commerce berbasis web yang dihasilkan dapat memasarkan produk dalam cakupan area yang lebih luas serta dapat memberikan informasi mengenai produk dengan cepat, dan dengan adanya sistem e-commerce ini konsumen akan lebih mudah dalam melakukan pemesanan produk.
2	(Achyar and Pratama, 2021)	Sistem Informasi E-Commerce Furniture Berbasis Web	metode <i>waterfall</i>	menghasilkan sistem informasi e-commerce furniture

		Pada Toko Mebel Ubaidillah Kamal Jepara		berbasis web pada toko mebel Ubaidillah Kamal Jepara untuk memudahkan pembeli dalam melakukan pemesanan
3	(Handayani, 2018)	Perancangan Sistem Informasi Penjualan Berbasis E-Commerce Studi Kasus Toko Kun Jakarta	Metode SDLC	Hasil dan kesimpulan dari penelitian ini adalah website penjualan berbasis ecommerce dapat dijadikan sebagai media promosi, mempermudah proses penjualan yang dilakukan secara online, serta dapat mempermudah dalam pengolahan data dan pencetakan laporan yang tidak lagi memerlukan pencatatan dalam bentuk arsip.
4	(Syahriani, 2017)	Penerapan Sistem Informasi E-Commerce Pakaian Dan Perlengkapan Bayi	Model <i>waterfall</i>	Sistem informasi penjualan pakaian dan perlengkapan bayi berbasis website ini

				mempermudah pelanggan sebagai media alternatif untuk melakukan transaksi penjualan. Dengan
5	(Hasanudin <i>et al.</i> , 2019)	Aplikasi E-Commerce Sistem Informasi Penjualan Rolling Door Berbasis Rapid Application Development	Metode RAD	aplikasi penjualan Rolling Door memenuhi kebutuhan pengguna karena dapat menghemat waktu dalam merancang dan membangun aplikasi tersebut. Hasil pengujian menampilkan sistem memiliki fitur navigasi dalam memberikan kemudahan bagi konsumen saat mengunjungi eCommerce tersebut. konsumen dapat melakukan proses pemesanan barang secara online tanpa dibatasi ruang dan waktu.

## **2.2. Rancang Bangun**

Rancang Bangun adalah penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam suatu kesatuan yang utuh dan berfungsi. Dengan demikian pengertian rancang bangun merupakan kegiatan menerjemahkan hasil analisa ke dalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut atau memperbaiki sistem yang sudah ada (Pressman, 2015)

## **2.3. Sistem**

Sistem adalah kumpulan atau himpunan dari unsur atau variable-variabel yang saling terkait, saling berinteraksi, dan saling tergantung satu sama lain untuk mencapai tujuan (Tohari, 2017).

Sistem dapat didefinisikan sebagai kumpulan dari elemen-elemen berupa data, jaringan kerja dari prosedur-prosedur yang saling berhubungan, sumber daya manusia, teknologi baik hardware dan software yang saling berinteraksi sebagai kesatuan untuk mencapai tujuan atau sasaran tertentu yang sama (Maniah and Haminidin, 2017).

Dari beberapa kutipan di atas maka penulis dapat menyimpulkan bahwa sistem informasi adalah sistem di dalam suatu instansi atau organisasi perusahaan yang mempertemukan kebutuhan pengolahan transaksi harian dan memberikan laporan-laporan atau informasi yang dibutuhkan.

## **2.4. Penjualan**

Penjualan adalah kegiatan transaksi yang sesungguhnya untuk mencapai volume penjualan yang dikehendaki (Tjendera, 2016).

Tujuan tersebut akan menjadi kenyataan apabila dilaksanakan dengan kemauan dan kemampuan yang memadai. Selain itu, harus diperhatikan pula faktor-faktor berikut ini.

1. Modal yang diperlukan
2. Kemampuan perencanaan dan membuat produk
3. Kemampuan menentukan tingkat harga yang tepat
4. Kemampuan memilih penyalur yang tepat
5. Kemampuan dalam menggunakan cara-cara promosi yang tepat

Pada umumnya, para pengusaha mempunyai tujuan mendapatkan laba tertentu atau semaksimal mungkin, dan mempertahankan atau bahkan berusaha meningkatkannya untuk jangka panjang. Tujuan tersebut dapat direalisasikan apabila penjualan dapat dilaksanakan seperti yang direncanakan, dengan demikian tidak berarti bahwa barang atau jasa yang terjual selalu akan menghasilkan laba, disinilah faktor-faktor diatas harus mendapat perhatian sepenuhnya oleh pihak perusahaan. Bagi perusahaan, pada umumnya mempunyai tujuan umum, sebagai berikut.

1. Mencapai volume penjualan tertentu.
2. Mendapatkan laba tertentu.

#### **2.4.1. Model Transaksi**

Jenis transaksi dalam menerapkan sistemnya pada penelitian yang dilakukan oleh peneliti menggunakan *Business to Consumers (B2C)*. Berikut ini Menurut (Laudon and Traver, 2017) Jenis-jenis transaksi di antaranya sebagai berikut:

1. *Business to Consumers (B2C)* yaitu penjual adalah suatu organisasi dan pembeli adalah individu. *Business to Consumers* memiliki karakteristik sebagai berikut:
  - a. Terbuka untuk umum, dimana informasi disebarakan ke umum.
  - b. *Service* yang diberikan bersifat umum (*generic*).
  - c. *Service* diberikan berdasarkan permohonan. *Consumer* melakukan inisiatif dan produsen harus memberikan respon sesuai permohonan.
2. *Collaborative Commerce (C-Commerce)* yaitu kerjasama secara elektronik antara rekanbisnis. Kerjasama ini biasanya terjadi antara rekan bisnis yang berada pada jalur penyediaan barang (*Supply Chain*).
3. *Business to Business (B2B)* tipe ini meliputi transaksi antar organisasi yang dilakukan di *electronicmarket*. *Business to Business* memiliki karakteristik:
  - a. *Trading partners* yang sudah diketahui dan umumnya memiliki hubungan yang cukup lama. Informasi hanya dipertukarkan dengan *partner* tersebut. Dikarenakan sudah mengenal rekan komunikasi, jenis informasi yang dikirimkan dapat disusun sesuai dengan kebutuhan dan kepercayaan.
  - b. Pertukaran data berlangsung berulang-ulang dan secara berkala, misalnya setiap hari dengan format data yang sudah disepakati bersama. Dengan kata lain, servis yang digunakan sudah tertentu. Hal ini memudahkan pertukaran data untuk dua *entity* yang menggunakan standar yang sama.

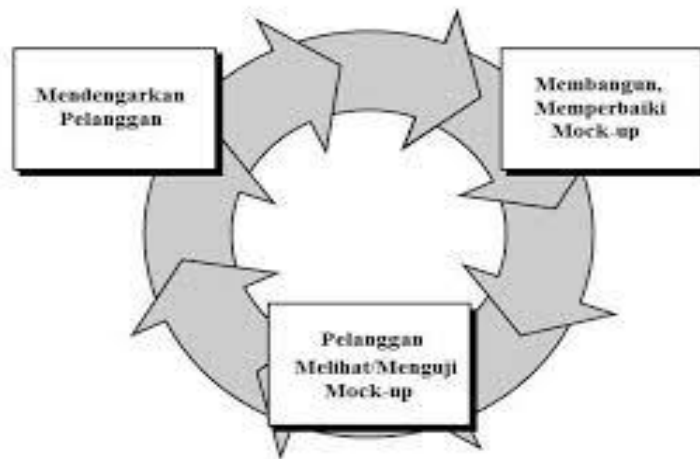
- c. Salah satu pelaku dapat melakukan inisiatif untuk mengirimkan data, tidak harus menunggu partnernya.
  - d. Model yang umum digunakan adalah *peer to peer*, dimana *processing intelligence* dapat didistribusikan pada pelaku bisnis.
4. *Consumer to Business (C2B)* yaitu *Consumerto Business* memberitahukan kebutuhan atas suatu produk atau jasa tertentu, dan para pemasok bersaing untuk menyediakan produk atau jasa tersebut ke konsumen.
  5. *Costumer to Customer (C2C)* yaitu konsumen menjual secara langsung kekonsumen lain atau mengiklankan jasa pribadi di internet. Dalam *Costumer to Costumer costumer* menjual produk atau jasa ke costumer lain. Dapat juga disebut sebagai pelanggan ke pelanggan, yaitu orang yang menjual produk dan jasa ke satu sama lain.

## **2.5. E-Commerce**

*Electronik Commerce* dikenal jugadengan sebutan E-commerce merupakan suatu proses transaksi jual beli antara penjual dengan pembeli untuk menyediakan barang dan jasa atau pengambilan hak yang terjadi menggunakan bantuan dengan media elektronik yang hadir secara fisik. Medium ini muncul di dalam sebuah jaringan dengan menggunakan sistem terbuka yaitu dengan menggunakan. Transaksi ini akan terjadi terlepas dari batas suatu wilayah dan syarat nasional E-Commerce yang mempunyai dampak bagi proses jual beli (Achyar and Pratama, 2021).

## 2.6. Metode Pengembangan *Prototype*

Menurut Alfiah and Damayanti (2020) yang dikutip dari buku Rosa and Shalahudin (2018) menyatakan bahwa : Model prototipe dapat digunakan untuk menyambung ketidakpahaman pelanggan mengenai hal teknis dan memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak. Adapun tahapan – tahapan dalam metode *Prototype* dapat dilihat pada gambar 2.1:



**Gambar 2. 1** Tahapan – Tahapan Metode *Prototype*

Sumber : Alfiah and Damayanti (2020)

Menurut Rosa dan Shalahuddin (2018) menyatakan bahwa : Pada metode *prototype* terdapat tiga tahap yaitu :

### 1. Mendengarkan Pelanggan

Pada tahap ini, dilakukan pengumpulan kebutuhan dari sistem dengan cara mendengar kebutuhan pelanggan sebagai pengguna sistem perangkat lunak untuk menganalisis serta mengembangkan kebutuhan pengguna.



## 2. Merancang dan Membuat *Prototype*

Pada tahap ini, dilakukan perancangan dan pembuatan *prototype* sistem yang disesuaikan dengan kebutuhan pengguna.

## 3. Uji Coba

Pada tahap ini, dilakukan pengujian *prototype* sistem oleh pengguna kemudian dilakukan evaluasi sesuai dengan kekurangan-kekurangan dari kebutuhan pelanggan. Jika sistem sudah sesuai dengan *prototype*, maka sistem akan diselesaikan sepenuhnya.

## 2.7. Perancangan Sistem UML (*Unified Modeling Language*)




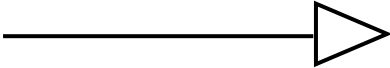

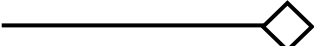
UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Rosa and Shalahudin, 2018)

### 1. *Class Diagram*

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Simbol-simbol yang ada pada diagram kelas pada tabel 2.2 di bawah ini:

**Tabel 2. 2** Simbol *Class Diagram*

Simbol	Deskripsi
Kelas	Kelas pada struktur sistem
<b>nama_kelas</b>	
+atribut	
+operasi()	

Antarmuka/ <i>Interface</i>  <b>nama_interface</b>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi/ <i>asociation</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah/ <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan <i>multiplicity</i>
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan/ <i>dependecy</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi/ <i>agregation</i> 	Relasi antar kelas dengan makna semua bagian ( <i>whole-part</i> )

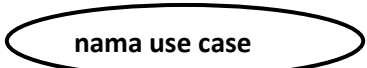
## 2. Use Case Diagram

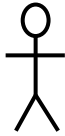



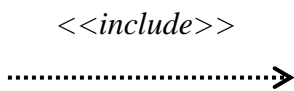
Menurut Rosa dan Shalahuddin (2018)

“*Use case diagram* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat”.

*Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Simbol-simbol yang ada pada diagram *use case* dapat dilihat pada gambar 2.3 di bawah ini:

**Tabel 2.3** Simbol diagram *use case*





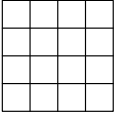


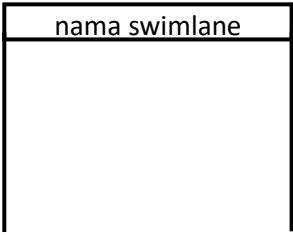
Simbol	Deskripsi
<i>Use Case</i>  <b>nama use case</b>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata

	kerja di awal frase nama <i>use case</i>
Aktor/ <i>actor</i> 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase
Asosiasi/ <i>association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
Ekstensi/ <i>extend</i>  	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek biasanya <i>use case</i> tambahan memiliki nama depan
Generalisasi/ <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
Menggunakan/ <i>Include/uses</i>  	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini

### 3. Activity Diagram

Activity diagram atau Diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. simbol-simbol yang ada pada *activity diagram* dapat dilihat pada tabel 2.4 di bawah ini :

Tabel 2. 4 Simbol *Activity Diagram*

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Tabel 	Suatu file komputer dari mana data bisa dibaca atau direkam selama kejadian bisnis
Dokumen 	Menunjukkan dokumen sumber atau laporan
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber: Rosa dan Shalahuddin (2018)

## 2.8. Pengertian MySQL

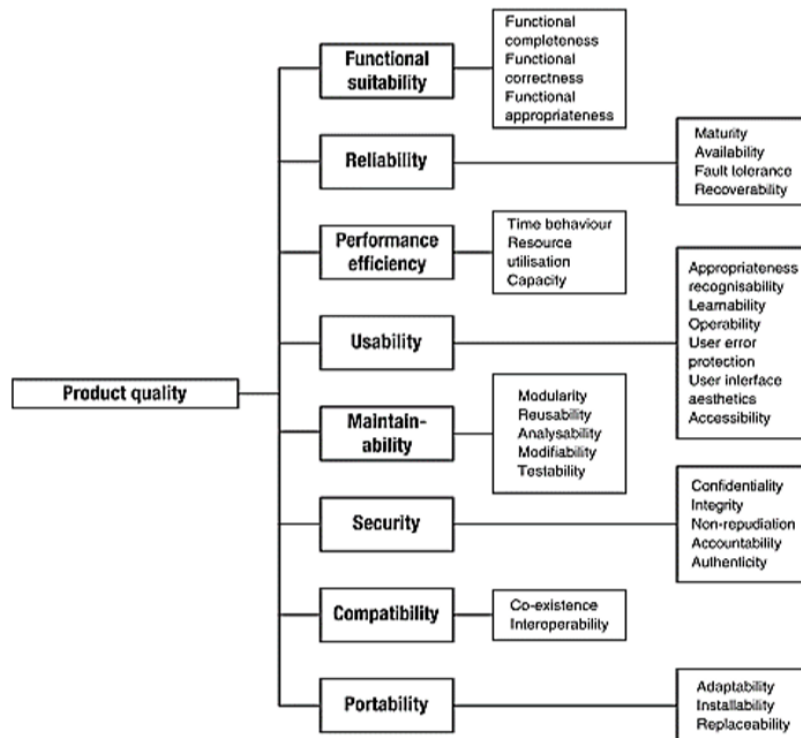
Menurut Gilmore (2015), *MySQL* adalah sebuah relasional *database server* yang menawarkan berbagai macam mekanisme dalam pengolahan data yang dikenal dengan *storage engines*.

Menurut Sheldon dan Moes (2015) *MySQL* sama seperti sistem *DBMS* lainnya, seperti *Oracle*, *DB2*, dan *SQL Server*. Memungkinkan untuk mengakses, memanipulasi, melindungi, dan memelihara metadata yang diperlukan untuk mendefinisikan data yang disimpan.

## 2.9. Pengujian iso 25010

ISO/IEC 25010 merupakan model kualitas sistem dan perangkat lunak yang menggantikan ISO/IEC 9126 tentang *software engineering* (Iqbal, 2016). Product quality ini juga digunakan untuk tiga model kualitas yang berbeda untuk produk perangkat lunak antara lain:

1. Kualitas dalam model penggunaan
2. Model kualitas produk
3. Data model kualitas



**Gambar 2. 2 Model kualitas produk ISO/IEC 25010**

Sumber : (Iqbal, 2016).

Adapun dimensi yang pertama terdapat beberapa faktor elemen diantaranya :

- 1) *Functionality* (Fungsionalitas). Kemampuan perangkat lunak Merupakan tingkatan dimana perangkat lunak dapat menyediakan fungsionalitas yang dibutuhkan ketika perangkat lunak digunakan pada kondisi spesifik tertentu dalam hal ini perangkat lunak dapat memenuhi kelayakan dari sebuah fungsi untuk melakukan pekerjaan yang spesifik bagi pengguna dan dapat memberikan hasil yang tepat dan ketelitian terhadap tingkat kebutuhan pengguna.
- 2) *Reliability* Merupakan tingkatan dimana perangkat lunak dapat bertahan pada tingkatan tertentu ketika digunakan oleh pengguna pada kondisi yang spesifik dalam hal ini perangkat lunak dapat beroperasi dan siap ketika dibutuhkan untuk digunakan dan juga dapat bertahan pada tingkat

kemampuan tertentu terhadap kegagalan, kesalahan serta perangkat lunak kembali pada tingkat tertentu dalam mengembalikan pengembalian data yang disebabkan kegagalan atau kesalahan pada perangkat lunak.

- 3) *Performance efficiency* Merupakan tingkatan dimana perangkat lunak dapat memberikan kinerja terhadap sejumlah sumber daya yang digunakan pada kondisi tertentu dalam hal ini *performance efficiency* dapat memberikan reaksi dan waktu yang dibutuhkan ketika melakukan aksi dari sebuah fungsi dan perangkat lunak dapat menggunakan sejumlah sumber daya ketika melakukan aksi dari sebuah fungsi.
- 4) *Usability* Perangkat lunak dapat dimengerti, dipelajari, digunakan dan menarik pengguna ketika digunakan dalam hal ini perangkat lunak mudah dipelajari oleh pengguna, perangkat lunak dapat digunakan dan dioperasikan oleh pengguna.
- 5) *Security* Merupakan perlindungan terhadap perangkat lunak dari berbagai ancaman atau keganjalan dalam hal ini perangkat lunak memiliki perlindungan terhadap data atau informasi dari pengguna dan merupakan dari kelengkapan, ketepatan dari sejumlah *asset* yang telah dijaga sehingga aksi atau tindakan yang dilakukan telah terbukti dan hal tersebut tidak dapat ditolak.
- 6) *Compability* Faktor ini merupakan kemampuan dari dua atau lebih komponen perangkat lunak dapat melakukan pertukaran informasi dan melakukan fungsi yang dibutuhkan ketika digunakan pada *hardware* atau lingkungan perangkat lunak yang sama.

- 7) *Maintainability* Merupakan tingkat dimana sebuah perangkat lunak dapat dimodifikasi. Dalam hal ini modifikasi adalah perbaikan, perubahan atau penyesuaian perangkat lunak untuk dapat berubah pada lingkungan, kebutuhan dan fungsionalitas yang spesifik. Selain itu perangkat lunak dapat di analisis untuk mengetahui apa yang menyebabkan kegagalan pada perangkat lunak untuk mengidentifikasi bagian yang dapat dimodifikasi.
- 8) *Transferability* Merupakan kemudahan dimana sistem atau komponen dapat berpindah dari lingkungan satu ke lingkungan yang lain dalam hal ini perangkat lunak dapat beradaptasi dengan cepat pada spesifikasi lingkungan yang berbeda tanpa menerapkan aksi atau cara lain dari pada memberikan tujuan tertentu terhadap perangkat lunak yang telah ada.