

## BAB II LANDASAN TEORI

### 2.1 Tinjauan Pustaka

Tinjauan pustaka atau *literatur review* berisikan penelitian-penelitian yang serupa dan berkaitan dengan penelitian yang akan dilakukan bertujuan untuk keaslian penelitian, untuk tinjauan pustaka dapat dilihat sebagaiberikut:

**Tabel 2.1 Penelitian Terdahulu**

No	Penulis	Tahun	Metode	Judul Penelitian
1.	Siti Anisah	2021	Prototype	Customer Relationship Management Menggunakan Metode Prototyping Untuk Produk Pembiayaan BPRS Al-Salam
2.	Merti Warsela, Agung Deni Wahyudi dan Ari Sulistiyawati	2021	XP	Penerapan Customer Relationship Management Untuk Mendukung Marketing Credit Executive (Studi Kasus: Pt Fif Group)
3.	Johar Saputra Irsandi, Iskandar Fitri dan Novi Dian Nathasia	2021	Waterfall & Agile	Sistem Informasi Pemasaran dengan Penerapan CRM (Customer Relationship Management) Berbasis Website menggunakan Metode Waterfall dan Agile
4.	Seni Asria dan Asmal	2021	Waterfall	Implementasi Crm Dan Metode Usg Pada Perancangan Sistem Informasi Penjualan Berbasis Web
5.	Ririn Dayanti, Nofriadi dan Pristiyanilicia Putri	2021	XP	Penerapan Crm Dalam Sistem Informasi Penjualan Dan Pemasaran Berbasis Web Pada Toko Bintang Kaca Tanjungbalai

### 2.1.1 Literatur 1

Anisah (2021) melakukan penelitian tentang penerapan konsep manajemen hubungan pelanggan atau *Customer Relationship Management (CRM)* pada produk pembiayaan BPRS Al-Salam. Permasalahan dalam penelitian ini adalah dalam kegiatan pengajuan pembiayaan masih manual yaitu nasabah harus datang ke BPRS Al-Salaam hal ini sangat merepotkan nasabah yang ingin mengajukan pembiayaan. Media pemasaran jenis pembiayaan pun masih menggunakan brosur, belum adanya media yang mudah diakses untuk mengetahui informasi data pembiayaan nasabah seperti status pembiayaan, jatuh tempo pembayaran dari transaksi pembiayaan dan belum adanya wadah khusus untuk berkomunikasi antara pihak BPRS dengan nasabah. Tujuan dari penelitian ini adalah merancang CRM untuk produk pembiayaan agar dapat meningkatkan pelayanan kepada nasabah. Metode pengembangan sistem menggunakan metode *Prototyping*. Tools dalam perancangan sistem menggunakan UML yang diantaranya *usecase diagram*, *usecase narrative*, *activity diagram*, *sequence diagram* dan *class diagram*. Sedangkan pemrograman sistem menggunakan PHP dan *database mysql*, pengujian sistem dengan metode *User Acceptance Test (UAT)*. Hasil penelitian ini adalah pengembangan *Customer Relationship Management System* dengan menggunakan metode prototyping telah berhasil dikembangkan dengan fitur testimony kritik dan saran, dan fitur pesan agar mampu berkomunikasi dengan customer service. Dan cukup memenuhi kepuasan pengguna berdasarkan hasil dari pengujian menggunakan *User Acceptance Test (UAT)*.

### 2.1.2 Literatur 2

Warsela, Wahyudi dan Sulistiyawati (2021) melakukan penelitian tentang Penerapan *Customer Relationship Management* untuk mendukung marketing *Credit Executive* di PT. FIF Group. Permasalahan dalam penelitian ini adalah keterbatasan sarana dalam memperoleh informasi dan promosi mengenai peminjaman dana serta melakukan pendaftaran. Tujuan penelitian ini adalah untuk Membangun dan mengimplementasikan sistem dengan konsep *Customer Relationship Management (CRM)* untuk mendukung kegiatan *Marketing Credit Executive (MCE)* pada PT. FIF Group agar proses pengajuan data nasabah dapat dilakukan kapanpun dan dimanapun serta dapat menyajikan informasi kepada nasabah tentang produk dan promosi mengenai peminjaman dana. Metodologi penelitian yang digunakan adalah metode *Extreme Programming (XP)*. Perancangan perangkat lunak dalam penelitian ini menggunakan *Unified Modeling Language (UML)*. Bahasa pemrograman menggunakan *PHP* dengan *database MySQL*. Hasil Sistem Penerapan *Customer Relationship Management* Untuk Mendukung *Marketing Credit Executive* pada PT. FIF Group memperoleh hasil yang sangat baik untuk digunakan memperoleh hasil 88,44%. Maka dapat disimpulkan bahwa perancangan sistem ini dinyatakan sangat baik untuk digunakan dan layak untuk di implementasikan.

### 2.1.3 Literatur 3

Irsandi, Fitri dan Nathasia (2021) melakukan penelitian tentang penerapan CRM pada sistem informasi pemasaran berbasis website pada Toko Bella Frame ART. Masalah dalam penelitian ini adalah metode promosi masih menggunakan brosur, poster dan pamflet, sehingga pemilik toko memerlukan biaya yang tidak sedikit untuk menyampaikan informasi ketarget pasar secara luas. Tujuan penelitian

ini adalah membuat sistem yang dapat melakukan penyampaian informasi produk ke pelanggan dapat dilakukan dengan lebih mudah dan lebih cepat, serta diharapkan dapat membantu pihak toko dalam melakukan pengambilan keputusan dalam pemasaran produk dan pelayanan pelanggan untuk kedepannya. Dalam penelitian ini penulis menggunakan metode pengembangan aplikasi yaitu metode waterfall dan agile. Hasil dari penelitian ini adalah sistem informasi berbasis web dengan metode *Customer Relationship Management* (CRM) yang dikembangkan telah memenuhi kebutuhan dan mempunyai keunggulan yaitu terintegrasi dengan sistem yang berkaitan dengan sistem informasi pemasaran.

#### **2.1.4 Literatur 4**

Asria dan Asmal (2021) melakukan penelitian tentang pembangunan sistem *Customer Relationship Management* pada PT. Central Georgette Nusantara. Masalah dalam penelitian ini adalah banyaknya distributor dan konveksi yang bekerja sama, perusahaan semakin sulit dalam berinteraksi dengan pihak distributor dan konveksi, karena tidak dapat mengatur waktu ketika ada lebih dari satu *customer* yang ingin berinteraksi dengan perusahaan. Tujuan dari penelitian ini adalah merancang sistem informasi pemesanan arang berbasis web dengan menerapkan strategi CRM dan metode USG, mengimplementasikan sistem informasi pemesanan arang berbasis web dengan menerapkan strategi CRM dan metode USG. Penelitian ini dilakukan secara sistematis dengan menggunakan metode penelitian *waterfall*. Perancangan perangkat lunak dalam penelitian ini menggunakan *UML (Unified Modeling Language)* yang terdiri dari *usecase diagram*, *activity diagram* dan *class diagram*. Bahasa pemrograman menggunakan *PHP* dengan *database MySQL*. Hasil penelitian ini adalah dalam pembangunan

sistem *Customer Relationship Management* dapat disimpulkan bahwa dengan tersedianya sistem *Customer Relationship Management* perusahaan dapat mempertahankan *customer* yang ada dengan adanya bantuan penyajian informasi seperti informasi kritik dan saran *customer*.

### **2.1.5 Literatur 5**

Dayanti, Nofriadi dan Putri (2021) melakukan penelitian tentang mengukur kesiapan dari penerapan dalam perancangan *Customer Relationship Management* (CRM) berbasis web pada Toko Bintang Kaca Tanjungbalai. Permasalahan dalam penelitian ini adalah proses penjualan masih dilakukan dengan memasang banner di depan toko dan rekomendasi dari orang ke orang. Sehingga mengakibatkan kurangnya informasi yang diketahui oleh pelanggan mengenai produk pada Toko Bintang Kaca Tanjungbalai. Tujuan dalam penelitian ini adalah membangun sistem menggunakan *Customer Relationship Management* (CRM) untuk mengolah sistem informasi penjualan pada Toko Bintang Kaca Tanjungbalai berbasis website. Metode pengembangan sistem dalam penelitian ini menggunakan *waterfall*, sedangkan bahasa pemrograman yang digunakan *PHP* dan *MySQL* sebagai basis data. Hasil penelitian ini adalah hasil penerapan aplikasi *Customer Relationship Management* (CRM) berbasis web yang mempermudah Toko Bintang Kaca Tanjungbalai dalam meningkatkan pelayanan dan minat terhadap pelanggan, pelanggan dan pengguna dapat dengan mudah memperoleh informasi mengenai Toko Bintang Kaca Tanjungbalai dan melakukan pemesanan secara online dimana saja dan kapan saja tanpa harus datang langsung ke toko.

## **2.2 Teori *Customer Relationship Management* (CRM)**

### **2.2.1 Pengertian *Customer Relationship Management* (CRM)**

Secara umum *Customer Relationship Management* (CRM) atau manajemen hubungan pelanggan dapat diartikan sebagai proses dalam mengidentifikasi, menarik, dan mempertahankan pelanggan guna untuk menciptakan nilai pelanggan. *Customer Relationship Management* (CRM) merupakan kegiatan yang dilakukan untuk menangani hubungan antara lembaga dengan anggota yang bertujuan untuk meningkatkan nilai perusahaan di mata anggota. Hubungan dengan pelanggan sangatlah penting bagi sebuah lembaga. Sebab, tanpa pelanggan lembaga tersebut tidak akan bisa menjalankan tujuan mereka (Putri, 2018).

*Customer Relationship Management* (CRM) dapat disimpulkan sebagai proses mendapatkan, memelihara dan meningkatkan hubungan pelanggan yang menguntungkan, yang bertujuan untuk menciptakan nilai pelanggan, sehingga memuaskan pelanggan dan mendatangkan keuntungan maksimal bagi perusahaan. Fokus dari *Customer Relationship Management* (CRM) adalah tingkat kepuasan pelanggan, meningkatkan loyalitas pelanggan dan meningkatkan pendapatan dari pelanggan yang ada, dalam menghadapi tingginya tingkat persaingan, globalisasi dan perputaran pelanggan serta perkembangan biaya pengakuisisian pelanggan. Tujuan utama yang ingin dicapai oleh semua strategi *Customer Relationship Management* (CRM) adalah untuk mengembangkan hubungan yang menguntungkan dengan pelanggan (Yulianti, Sjahrudin dan Tahir, 2017).

### **2.2.2 Tahapan Dalam CRM (*Customer Relationship Management*)**

Menurut (Nugraha, Witanti dan Komarudin, 2017) implementasi konsep *CRM* yang baik adalah mengorganisasikan proses *CRM* di sekitar member dan tidak hanya pada fungsi internal perusahaan. Terdapat tiga tahapan dalam *CRM*, yaitu:

1. Mendapatkan pelanggan baru dengan cara memberikan kemudahan pengaksesan informasi, inovasi baru, dan pelayanan menarik.
2. Meningkatkan hubungan dengan pelanggan yang telah ada. Perusahaan berusaha menjalin hubungan dengan pelanggan melalui pemberian pelanggan yang baik terhadap pelanggannya.
3. Mempertahankan pelanggan, tahap ini merupakan usaha mendapatkan loyalitas pelanggan dengan mendengarkan pelanggan dan berusaha memenuhi keinginan pelanggan.

### **2.2.3 Faktor-Faktor Yang Mempengaruhi CRM**

Menurut Robinette dan Claire (2001) ada empat faktor yang mempengaruhi *Customer Relationship Management* (CRM), yaitu keuntungan bersama, komitmen, komunikasi, dan kualitas pelayanan. Dalam penelitian ini faktor-faktor yang akan diteliti adalah faktor komitmen, komunikasi, dan kualitas pelayanan. Pengertian dan penjelasan masing-masing faktor yang diteliti adalah:

1. Komitmen

Komitmen merupakan orientasi jangka panjang pelanggan terhadap hubungan yang didasarkan pada ikatan emosional dan itikad untuk tetap menjaga hubungan baik agar dapat menghasilkan keuntungan yang lebih tinggi. Loyalitas pelanggan adalah komitmen pelanggan terhadap suatu

merek, toko, atau pemasok berdasarkan sikap yang sangat positif dan tercermin dalam pembelian ulang yang konsisten.

## 2. Komunikasi

Komunikasi merupakan salah satu karakteristik fundamental dari sebuah hubungan yang berjalan dengan baik. Pengumpulan informasi dari pelanggan dapat dilakukan saat terjadi percakapan atau tanya jawab secara langsung, berfokus pada kelompok pelanggan tertentu, dan mengobservasi atau mencari tahu mengenai pembelian yang dilakukan oleh pelanggan.

## 3. Kualitas Pelayanan

Kualitas merupakan kunci untuk menciptakan nilai dan kepuasan pelanggan. Kualitas pelayanan terdiri dari 5 faktor, yaitu frekuensi, pelayanan baku, kenyamanan, ketepatan, keamanan, dan keselamatan.

### **2.2.4 Manfaat Strategi *Customer Relationship Management* (CRM)**

Penerapan strategi *Customer Relationship Management* (CRM) dalam bisnis memiliki beberapa manfaat yang menguntungkan perusahaan. Adapun manfaat penerapan strategi *Customer Relationship Management* (CRM) menurut Mekari (2019) adalah sebagai berikut:

#### 1. Mendorong Loyalitas Pelanggan

*Customer Relationship Management* (CRM) memungkinkan perusahaan untuk memanfaatkan informasi dari semua titik kontak dengan pelanggan, baik melalui *web*, *call center* atau melalui staf pemasaran dan pelayanan di lapangan. Dengan adanya konsistensi dan kemudahan dalam mengakses dan menerima informasi, maka bagian pelayanan dapat memberikan



layanan yang lebih baik lagi kepada pelanggan dengan memanfaatkan berbagai informasi penting mengenai pelanggan tersebut.

## 2. Mengurangi Biaya

Dengan penerapan *Customer Relationship Management* (CRM), penjualan dan pelayanan terhadap pelanggan dapat memiliki skema yang spesifik dan fokus serta dapat menargetkan pelayanan pada pelanggan yang tepat di saat yang tepat. Dengan demikian, biaya yang dikeluarkan akan terperinci dengan baik dan tidak terbuang percuma yang berujung pada pengurangan biaya.

## 3. Meningkatkan Efisiensi Operasional

Kemudahan proses penjualan dan layanan akan dapat mengurangi risiko turunnya kualitas pelayanan dan mengurangi beban *cash flow*.

## 4. Peningkatan *Time to Market*

Penerapan *Customer Relationship Management* (CRM) akan memungkinkan perusahaan mendapatkan informasi mengenai pelanggan seperti data tren pembelian oleh pelanggan yang dapat dimanfaatkan perusahaan dalam menentukan waktu yang tepat dalam memasarkan suatu produk.

## 5. Peningkatan Pendapatan

Penerapan *Customer Relationship Management* (CRM) yang tepat akan meningkatkan loyalitas pelanggan, mengurangi biaya, meningkatkan efisiensi operasional dan peningkatan *time to market* yang pada akhirnya akan berujung pada peningkatan pendapatan perusahaan.

### **2.3 Pelanggan/Nasabah**

Pelanggan adalah salah satu aset paling berharga yang dimiliki oleh perusahaan. Tanpa adanya pelanggan, kelangsungan hidup perusahaan tidak akan terjamin. Pelanggan adalah pihak yang memaksimalkan nilai. Jadi pelanggan adalah seseorang yang melakukan pembelian suatu produk atau jasa yang dilakukan secara berulang dan mereka juga ingin mendapatkan yang terbaik dari perusahaan agar kebutuhan mereka terpenuhi selain itu mereka memiliki *track record* hubungan yang kuat dan pembelian berulang (Carissa, Fauzi dan Kumadji, 2014).

### **2.4 Pembiayaan/Peminjaman Dana**

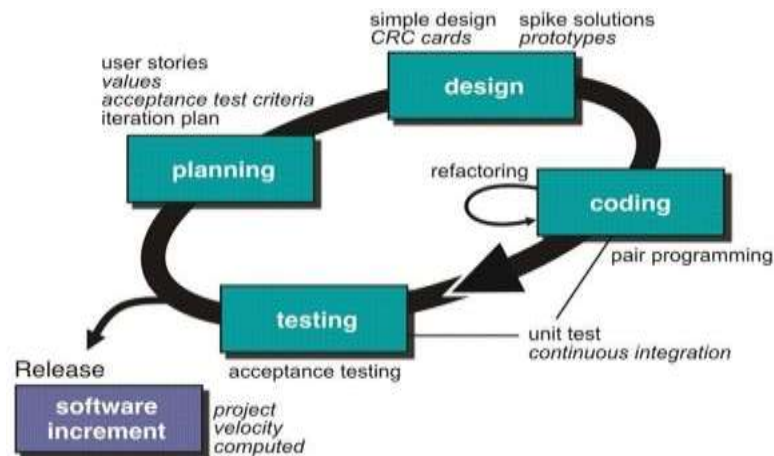
Pembiayaan atau *financing* yaitu pendanaan yang diberikan oleh suatu pihak kepada pihak lain untuk mendukung investasi yang telah direncanakan, baik dilakukan sendiri maupun lembaga. Dengan kata lain, pembiayaan adalah pendanaan yang dikeluarkan untuk mendukung investasi yang telah direncanakan. Peminjaman dana merupakan penyediaan uang atau tagihan yang dapat dipersamakan dengan itu, berdasarkan tujuan atau kesepakatan pinjam meminjam antara suatu instansi dengan pihak lain yang mewajibkan pihak peminjam untuk melunasi hutangnya setelah jangka waktu tertentu dengan disertai pembayaran sejumlah imbalan (Sanwani, Herwanti dan Jufri, 2017) .

### **2.5 Metode Pengembangan Sistem**

#### **2.5.1 *Extreme Programming (XP)***

*Extreme Programming (XP)* adalah metodologi pengembangan perangkat lunak yang ditujukan untuk meningkatkan kualitas perangkat lunak dan tanggap terhadap perubahan kebutuhan pelanggan. Jenis pengembangan perangkat lunak semacam ini dimaksudkan untuk meningkatkan produktivitas dan memperkenalkan

pos pemeriksaan dimana persyaratan pelanggan baru dapat diadopsi. Tahapan-tahapan dari *Extreme Programming* terdiri dari *planning* seperti memahami kriteria pengguna dan perencanaan pengembangan, *designing* seperti perancangan *prototype* dan tampilan, *coding* termasuk pengintegrasian, dan yang terakhir adalah *testing* (Pressman, 2012).



**Gambar 2.1 Model *Extreme Programming* (XP)**

**Sumber : (Pressman, 2012)**

### 2.5.2 Tahapan *Extreme Programming* (XP)

Terdapat empat tahap proses yang dilakukan dalam *Extreme Programming* menurut Pressman (2012), yaitu :

#### 1. *Planning* (Perencanaan)

Kegiatan Perencanaan (disebut juga *planning game*) biasanya dimulai dengan mendengarkan suatu kegiatan yang bertujuan mengumpulkan kebutuhan-kebutuhan untuk memahami konteks bisnis dan perlunya keluaran-keluaran (*output*), fungsi utama, dan *fungsi*alitas. Pada perencanaan terdapat *user stories values* yaitu story dengan value tertinggi akan dipindahkan dari jadwal dan diimplementasikan pertama, *acceptance test criteria iteration plan* melakukan perhitungan kecepatan project

selama *development*, *customer* dapat menambah *story*, merubah *value*, membagi *story* atau menghapusnya.

2. *Design* (Perancangan)

Perancangan yang simple, menarik, dan sederhana selalu memberikan hasil yang lebih disukai daripada gambaran-gambaran yang lebih kompleks. Perancangan XP memberikan panduan implementasi untuk suatu cerita ketika ditulis, tidak kurang, tidak lebih. Terdapat *simple design* *CRC Cards* untuk mengenali dan mengatur *object oriented class* sesuai dengan *software increment* dan *spike solutions prototypes* melakukan spesifikasi solusi dari *object oriented class*.

3. *Coding* (Pengkodean)

Pengkodean ini dilanjutkan setelah cerita yang telah dikembangkan dan rancangan yang telah dilakukan oleh tim perangkat lunak. Pengkodean ini tidak langsung mengarah ke kode-kode program. Tim akan mengembangkan serangkaian unit pengujian lalu beralih ke pengkodean. Pada tahapan *pair programming* melakukan kerja sama untuk membuat kode dari satu story. Dan *refactoring* adalah proses restrukturisasi kode program komputer yang ada tanpa mengubah perilaku eksternalnya.

4. *Testing* (Pengujian)

Unit pengujian yang harus dibuat dan kemudian dijalankan menggunakan kerangka kerja yang memungkinkan mereka untuk diotomatisasi sehingga dapat dijalankan dengan mudah dan dapat dijalankan berulang kali. Pada tahapan pengujian yaitu *unit test*

*continuous integration* yaitu tahapan pengujian code yang diintegrasikan dengan kerja lainnya dengan pengujian yang dilakukan oleh customer dan focus pada keseluruhan dan fungsional sistem, dan *acceptance testing* yaitu pengujian yang dilakukan *customer stories* yang akan diimplementasikan sebagai bagian dari *software release*. Selanjutnya terdapat tahapan *software increment* *project velocity computed* yaitu tahapan yang telah diimplementasikan dari *software release* yang nantinya akan diterapkan dalam suatu sistem.




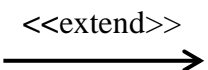
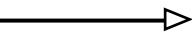
## **2.6 Unified Modeling Language (UML)**

Alat pengembangan sistem yang digunakan pada penelitian ini yaitu *Unified Modeling Language (UML)*. *UML* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (A.S and Shalahudin, 2018). Berikut ini merupakan penjelasan tentang masing-masing diagram yang ada pada *UML (Unified Modelling Language)*.

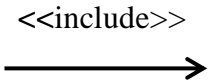
### **2.6.1 Use Case Diagram**

A.S dan Shalahudin (2018) *Use Case Diagram* merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara salah satu lebih aktor dengan sistem informasi yang akan dibuat. Simbol-simbol pada *use case diagram* dapat dilihat pada tabel 2.2 dibawah ini:

Tabel 2.2 Simbol *Use Case Diagram*

No	Simbol	Deskripsi
1	<i>Use case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor.
2	Aktor / <i>actor</i> 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
3	Asosiasi / <i>association</i> 	Komunikasi antar aktor dan <i>usecase</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4	Ekstensi / <i>extend</i> 	Relasi <i>use case</i> tambahan kesebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
5	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.
6	<i>Include</i>	Relasi <i>usecase</i> tambahan ke <i>usecase</i> dimana <i>usecase</i> yang ditambahkan

**Tabel 2.2 Simbol *Use Case Diagram* (Lanjutan)**


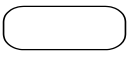
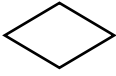


		<p>memerlukan <i>usecase</i> ini menjalankan fungsionalnya atau sebagai syarat dijalankan <i>use case</i> ini.</p>
--	---	--

Sumber: (A.S dan Shalahudin, 2018)

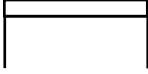
### 2.6.2 *Activity Diagram*

*Activity Diagram* adalah diagram yang menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor jadi aktivitas yang dapat dilakukan oleh sistem (A.S dan Shalahudin, 2018). Simbol-simbol yang terdapat dalam *Activity Diagram* dapat dilihat pada tabel 2.3 dibawah ini:

**Tabel 2.3 Simbol *Activity Diagram***

No	Simbol	Deskripsi
1	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3	Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4	Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5	Status Akhir 	Status akhir yang dilakukan sistem sebuah diagram aktivitas memiliki sebuah status

**Tabel 2.3 Simbol Activity Diagram (Lanjutan)**

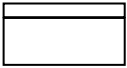


		akhir.
6	Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

**Sumber: (A.S dan Shalahudin, 2018)**

### 2.6.3 Class Diagram


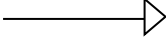
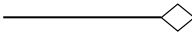
*Class diagram* adalah sebuah *class* yang menggambarkan struktur sistem dari segi pendefinisian kelas kelas yang akan dibuat untuk membangun sistem (A.S dan Shalahudin, 2018). *Class diagram* juga menjelaskan hubungan antar class dalam sebuah sistem yang sedang dibuat dan bagaimana caranya agar mereka saling berkolaborasi untuk mencapai sebuah tujuan. Simbol-simbol yang terdapat dalam *Activity Diagram* dapat dilihat pada tabel 2.4 dibawah ini:

**Tabel 2.4 Simbol Class Diagram**

No	Simbol	Deskripsi
1	Kelas / <i>class</i> 	Kelas pada struktur sistem.
2	Antarmuka / <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3	Asosiasi / <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4	Asosiasi berarah / <i>directed association</i>	Relasi antar kelas dengan makna kelas kelas



**Tabel 2.4 Simbol Class Diagram (Lanjutan)**

		yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisai (umum-khusus).
6	Agresi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua bagian ( <i>whole-part</i> ).

Sumber: (A.S and Shalahudin, 2018)

## 2.7 Teori Program Aplikasi

### 2.7.1 Website

*Website* adalah kumpulan halaman yang menampilkan informasi data *teks*, data gambar diam atau gerak, data animasi, suara, *video* dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman Destiningrum dan Adrian (2017). Untuk menyediakan sebuah *website*, maka harus menyediakan unsur-unsur penunjangnya, yaitu:

1. Nama *Domain* (*URL*)

Nama *domain* atau atau *URL* adalah alamat unik di dunia *internet* yang digunakan untuk mengidentifikasi sebuah *website*.

2. Rumah Tempat *Website* (*Web hosting*)

Pengertian *web hosting* dapat diartikan sebagai ruangan yang terdapat dalam *harddisk* tempat menyimpan berbagai data, *file-file*, gambar dan lain sebagainya yang akan ditampilkan di *website*. Besarnya data yang bisa

dimasukkan tergantung dari besarnya *web hosting* yang disewa atau dipunyai.

### 3. Bahasa Program (*Scripts Program*)

Bahasa yang digunakan untuk menerjemahkan setiap perintah dalam *website* yang pada saat diakses.

### 4. Desain *Website*

Setelah melakukan penyewaan *domain name* dan *web hosting* serta penguasaan bahasa program (*scripts program*), unsur *website* yang penting dan utama adalah desain. Desain *website* menentukan kualitas dan keindahan sebuah *website*.

### 5. Publikasi *Website*

Keberadaan situs tidak ada gunanya dibangun tanpa dikunjungi atau dikenal oleh masyarakat atau pengunjung *internet*. Karena efektif tidaknya situs sangat tergantung dari besarnya pengunjung dan komentar yang masuk. Untuk mengenalkan situs kepada masyarakat memerlukan apa yang disebut publikasi atau promosi *website*.

## **2.9.2 Web Server**

Web server adalah sebuah program atau aplikasi yang menggunakan model klien atau server dan world (HTTP), melayani file yang berbentuk halaman web untuk pengguna web (merespon permintaan komputer klien berisi permintaan HTTP) (Yuhefizar 2011).

## **2.9.3 PHP**

PHP adalah sekumpulan skrip atau bahasa program memiliki fungsi utama yaitu mampu mengumpulkan dan mengevaluasi hasil survey atau bentuk apapun

keserver database dan pada tahap selanjutnya akan menciptakan efek beruntun, suatu bahasa *pemrograman* yang digunakan untuk menerjemahkan baris kode program dan mesin yang dapat dimengerti oleh komputer yang bersifat *server-side* yang dapat ditambahkan ke dalam HTML, juga merupakan bahasa pemrograman berbasis server-side yang dapat parsing *script php* menjadi *script web* sehingga dari sisi client menghasilkan suatu tampilan yang menarik (Rochman, Hakim dan Riswanto, 2018).

#### 2.9.4 *Framework Codeigniter (CI)*

*Codeigniter* adalah sebuah *web application framework* yang bersifat *open source* digunakan untuk membangun aplikasi *web*. Tujuan utama pengembangan *codeigniter* adalah untuk membantu *developer* dalam pengerjaan aplikasi yang lebih cepat daripada menulis semua *code* dari awal dan *codeigniter* merupakan salah satu *framework* PHP tercepat yang ada saat ini. *Codeigniter* memiliki banyak fitur yang membantu para pengembang PHP untuk dapat membuat aplikasi *web* secara mudah dan cepat. *Codeigniter* mengizinkan para pengembang untuk menggunakan *framework* secara parsial maupun secara keseluruhan Romadhon dan Desmulyati (2019). Adapun keunggulan dari *codeigniter* adalah sebagai berikut:

1. *Codeigniter* adalah *framework* yang bersifat *free* dan *open-source*.
2. *Codeigniter* memiliki ukuran kecil dibandingkan dengan *framework* lain.
3. Aplikasi yang dibuat menggunakan *Codeigniter* bisa berjalan cepat.
4. *Codeigniter* menggunakan pola design *Model-View-Controller (MVC)* sehingga satu file tidak terlalu berisi banyak kode.
5. *Codeigniter* dapat diperluas sesuai dengan kebutuhan .

6. *Codeigniter* terdokumentasi dengan baik. Informasi tentang pustaka kelas dan fungsi yang disediakan oleh *Codeigniter* dapat diproses melalui dokumentasi yang disertai di dalam paket distribusinya .

### 2.9.5 *Sublime Text*

*Sublime Text* adalah aplikasi editor untuk kode dan teks yang dapat berjalan di berbagai *platform operating system* dengan menggunakan teknologi *Python API*. Aplikasi ini sangatlah fleksibel dan *powerfull*. Fungsionalitas dari aplikasi ini dapat dikembangkan dengan menggunakan *sublime packages*. *Sublime text* bukanlah aplikasi *open source*, yang artinya aplikasi ini membutuhkan lisensi yang harus dibeli. Akan tetapi beberapa fitur pengembangan fungsionalitas (*packages*) dari aplikasi ini merupakan hasil dari temuan dan mendapat dukungan penuh dari komunitas serta memiliki lisensi aplikasi gratis. *Sublime Text* mendukung berbagai bahasa pemrograman dan mampu menyajikan fitur *syntax highlight* hampir di semua bahasa pemrograman yang didukung ataupun dikembangkan oleh komunitas seperti *C, C++, C#, CSS, D, Dylan, Erlang, HTML, Groovy, Haskell, Java, JavaScript, LaTeX, Lisp, Lua, Markdown, MATLAB, OCaml, Perl, PHP, Python, R, Ruby, SQL, TCL, Textile dan XML*. Bahasa pemrograman yang didukung atau belum didukung secara *default* dapat lebih dimaksimalkan atau didukung dengan menggunakan *add ons* yang bisa di *download* sesuai kebutuhan *user* (Suprianto dan Matsea, 2018).

### 2.9.6 **MySQL**

*Structured Query Language* adalah suatu sistem basis data relation atau *Relational Database managemnt System (RDBMS)* yang mampu bekerja secara cepat dan mudah digunakan *MySQL* juga merupakan program pengakses *database*

yang bersifat jaringan, sehingga sapat digunakan untuk aplikasi multi user (banyak pengguna). *MySQL* didistribusikan gratis dibawah lisensi GPL (*General Public License*). Dimana setiap program bebas menggunakan *MySQL* namun tidak bisa dijadikan produk turunan yang dijadikan *closed source* atau komersial (Destiningrum dan Adrian, 2017).

### **2.9.7 XAMPP**

XAMPP adalah sebuah aplikasi yang dapat menjadikan komputer kita menjadi sebuah server. Kegunaan XAMPP ini untuk membuat jaringan local sendiri dalam artian kita dapat membuat *website* secara *offline* untuk masa coba-coba di komputer sendiri. Jadi fungsi dari XAMPP server itu sendiri merupakan server *website* kita untuk cara memakainya. Disebut server karena dalam hal ini komputer yang akan kita pakai harus memberikan pelayanan untuk mengakses web, untuk itu komputer kita harus menjadi server. Dapat disimpulkan xampp adalah aplikasi tools untuk menyediakan paket lunak yang berisi konfigurasi *Web Server*, *Apache*, *PHP*, *MySQL* untuk membantu dalam proses pembuatan aplikasi web yang menyatu menjadi satu sehingga memudahkan kita dalam membuat program web (Hariyanto, 2012).

### **2.9.8 Mockup Plus**

*Mockup Plus* adalah sebuah media visual atau *preview* dari sebuah konsep desain yang diberikan efek visual sehingga hasilnya sangat tampak atau menyerupai wujud nyata. *Mockup Plus* dapat memberikan gamabaran nyata dari sebuah konsep desain bagaimana konsep itu terlihat nantinya (Sabila, Rosali and Nugroho, 2018).

## 2.10 Teori Pengujian

### 2.10.1 ISO 25010

ISO/IEC 25010 merupakan standar terbaru dari *International Organization for Standardization and International Electrotechnical Commission* disingkat menjadi ISO/IEC yang dikembangkan untuk menggantikan versi sebelumnya yaitu ISO/IEC 9126 dan sejak tahun 2011 menjadi standar tolak ukur analisis kualitas perangkat lunak yang relevan untuk mengevaluasi sistem informasi (Tangkudung, Dako dan Dako, 2019). Gambar ISO/IEC 25010 dapat dilihat pada gambar 2.2 berikut:



**Gambar 2.2 Model Pengujian Perangkat Lunak ISO 25010**  
**Sumber: (Tangkudung, Dako dan Dako, 2019)**

### 2.10.2 Karakteristik ISO 25010

Terdapat enam karakteristik kualitas perangkat lunak utama *ISO 25010* menurut Rahayuda (2017), yaitu:

1. *Functional Suitability*

Produk aplikasi yang memberikan fungsional untuk memenuhi kebutuhan saat menggunakan produk dalam keadaan tertentu.

2. *Performance Efficiency*

Tingkat dimana produk aplikasi menyediakan performa yang baik dengan jumlah *resource* yang digunakan.

3. *Usability*

Tingkat dimana produk aplikasi mudah dimengerti, dipakai dan menarik untuk digunakan.

4. *Reliability*

Tingkat dimana produk aplikasi dapat mempertahankan kinerja pada level tertentu ketika digunakan dalam keadaan tertentu.

5. *Maintainability*

Tingkat dimana produk aplikasi dapat dimodifikasi. Modifikasi yang dilakukan dapat meliputi perbaikan, pengembangan atau adaptasi perangkat lunak untuk menyesuaikan dengan lingkungan, serta modifikasi pada kriteria dan spesifikasi fungsi.

6. *Portability*

Tingkat dimana produk aplikasi dapat dipindahkan dari satu ruang ke ruang lain.

Masing-masing karakteristik kualitas perangkat lunak model ISO/IEC 25010 dibagi menjadi beberapa sub-karakteristik kualitas. Tabel karakteristik kualitas perangkat lunak model ISO/IEC 25010 dapat dilihat pada tabel 2.5 berikut:

**Tabel 2.5 Karakteristik ISO 25010**

<b>Karakteristik</b>	<b>Sub Karakteristik</b>	<b>Deskripsi</b>
<i>Functional Suitability</i>	<i>Functional Completeness</i>	Kemampuan perangkat lunak dalam menyediakan fungsi dapat

Tabel 2.5 Karakteristik *ISO 25010* (Lanjutan)

		mencakup semua tugas dan tujuan pengguna secara spesifik.
	<i>Functional Correctness</i>	Kemampuan perangkat lunak sejauh mana produk atau sistem mampu menyediakan hasil yang tepat sesuai dengan kebutuhan.
	<i>Functional Appropriateness</i>	Kemampuan perangkat lunak dalam fungsi yang disediakan mampu menyelesaikan tugas dan tujuan secara spesifik.
<i>Performance Efficiency</i>	<i>Resource Utilization</i>	Kemampuan perangkat lunak sejauh mana jumlah dan jenis sumber daya yang digunakan produk atau sistem saat menjalankan fungsinya sesuai dengan kriteria
	<i>Capacity</i>	Kemampuan perangkat lunak sejauh mana batas maksimal sebuah produk atau sistem mampu memenuhi kriteria.
	<i>Time Behaviour</i>	Kemampuan perangkat lunak sejauh mana respon dan lama proses sebuah produk atau sistem saat



Tabel 2.5 Karakteristik *ISO 25010* (Lanjutan)

		menjalankan fungsinya sesuai dengan kriteria.
<i>Usability</i>	<i>Appropriateness</i> <i>Recognizability</i>	Kemampuan perangkat lunak sejauh mana pengguna mengetahui sebuah produk atau sistem sesuai dengan kebutuhan mereka.
	<i>Learnability</i>	Kemampuan perangkat lunak sejauh mana produk atau sistem dapat digunakan pengguna tertentu untuk mencapai tujuan mempelajari sebuah produk atau sistem secara efektif, efisien dan bebas dari resiko serta memenuhi kepuasan dalam konteks penggunaan.
	<i>Operability</i>	Kemampuan perangkat lunak sejauh mana produk atau sistem sistem mampu memenuhi kriteria dioperasikan dan dikontrol.
	<i>User Error</i> <i>Protection</i>	Kemampuan perangkat lunak sejauh mana produk atau sistem melindungi pengguna dalam melakukan kesalahan.

**Tabel 2.5 Karakteristik ISO 25010 (Lanjutan)**

	<i>User Interface</i> <i>Aesthetics</i>	Kemampuan perangkat lunak sejauh mana tampilan antarmuka memenuhi kesenangan dan kepuasan pengguna.
	<i>Accessibility</i>	Kemampuan perangkat lunak sejauh mana produk atau sistem dapat digunakan oleh pengguna secara luas untuk mencapai tujuan tertentu dalam konteks penggunaan
<i>Reliability</i>	<i>Maturity</i>	Kemampuan perangkat lunak sejauh mana sistem, produk, atau komponen memenuhi kriteria reliabilitas dibawah kondisi normal.
	<i>Availability</i>	Kemampuan perangkat lunak sejauh mana sistem, produk, atau komponen dapat beroperasi ketika diperlukan untuk digunakan.
	<i>Fault Tolerance</i>	Kemampuan perangkat lunak sejauh mana sistem, produk atau komponen.
	<i>Recoverability</i>	Kemampuan perangkat lunak ketika produk atau sistem mengalami

Tabel 2.5 Karakteristik *ISO 25010* (Lanjutan)

		kejadian atau kegagalan dapat mengembalikan data dan menjalankan kondisi sistem yang diharapkan.
<i>Maintainability</i>	<i>Modularity</i>	Kemampuan perangkat lunak dimana sistem atau program terdiri dari komponen yang berlainan sehingga perubahan terhadap satu komponen minimal memiliki pengaruh terhadap komponen lain
	<i>Reusability</i>	Kemampuan perangkat lunak dimana sebuah asset dapat digunakan pada lebih dari satu sistem perangkat lunak atau pada pembangunan asset lainnya.
	<i>Analyzability</i>	Kemampuan perangkat lunak dimana perangkat lunak dapat dianalisis untuk mengetahui apa yang menyebabkan kegagalan pada perangkat lunak atau untuk mengidentifikasi bagian yang dapat dimodifikasi.

**Tabel 2.5 Karakteristik ISO 25010 (Lanjutan)**

	<i>Modifiability</i>	Kemampuan perangkat lunak dimana perangkat lunak dapat menghindari efek yang tidak diharapkan dari modifikasi yang dilakukan terhadap perangkat lunak.
	<i>Testability</i>	Kemampuan perangkat lunak dimana perangkat lunak memungkinkan modifikasi perangkat lunak untuk dilakukan validasi.
<i>Portability</i>	<i>Adaptability</i>	Kemampuan perangkat lunak dapat beradaptasi dengan perubahan lingkungan atau sistem yang berbeda.
	<i>Installability</i>	Kemampuan perangkat lunak dapat digunakan dalam lingkungan atau sistem tertentu.
	<i>Replaceability</i>	Kemampuan perangkat lunak dapat menggantikan perangkat lunak lain apakah ada ketergantungan kepada perangkat lunak lain digunakan.

**Sumber: (Rahayuda, 2017)**