

## BAB II

### KAJIAN TEORI

#### 2.1. Sistem Informasi

Sistem informasi adalah suatu kerangka kerja atau prosedur kerja untuk mengkoordinasikan sumber daya yang memproses masukan( input) menjadi keluaran (informasi) untuk mencapai tujuan dan sasaran-sasaran sebuah organisasi. [1]

Sistem Informasi merupakan integrasi dari teknologi informasi yang digunakan pada suatu kelompok maupun organisasi dalam membantu pengoperasian yang terdapat pada lingkungan tersebut, keberadaan sistem informasi sangat berpengaruh dalam pencapaian dalam konteks penelitian ini, kelebihan sistem informasi dalam organisasi dapat menjadi faktor yang mempengaruhi pengembangan dan peningkatan kualitas penilaian nasabah dan proses pengambilan keputusan dalam pengajuan pinjaman.[2]

#### 2.2. Metode Prototype

Metode *Prototype* metode ini merupakan suatu paradigma baru dalam pengembangan perangkat lunak. dan juga salah satu metode Prototype dalam pengembangan sistem yang menggunakan pendekatan untuk membuat sesuatu program dengan cepat dan bertahap sehingga segera dapat dievaluasi oleh pemakai [3]

Metode penelitian yang digunakan adalah *metode prototype*, model proses yang diterapkan saat menjalankan komunikasi dengan *client* untuk membuat sebuah aplikasi, prototype tidak menyajikan bentuk asli sistem secara lengkap akan tetapi metode prototype berperan penting dalam penelitian untuk memberikan gambaran aplikasi yang akurat terhadap client [2]

Dari definisi di atas dapat diambil kesimpulan dengan cara membuat dalam penelitian ini, model fisik kerja sistem yang berfungsi sebagai versi awal dapat menjadi komponen penting dalam pengembangan solusi yang lebih efektif untuk menghadapi masalah kredit macet dan mengatasi kesulitan nasabah dalam melakukan pembayaran angsuran. awal dari sebuah sistem yang dapat dikembangkan sampai perangkat lunak tersebut memenuhi kebutuhan pengguna.

### **2.3. Bank**

Bank adalah badan usaha yang menghimpun dana dari masyarakat dalam bentuk simpanan dan menyalurkan kepada masyarakat dalam bentuk kredit atau bentuk lainnya dengan rangka meningkatkan taraf hidup rakyat banyak (Undang-Undang Nomor 10 Tahun 1998 Perubahan Undang-Undang Nomor 7 Tahun 1992 tentang perbankan).

Dalam konteks penelitian ini, bank yang menyalurkan modal dari sumber yang tidak langsung dapat memberikan peluang bagi nasabah yang menderita kesulitan dalam melakukan pembayaran angsuran atau memperoleh pinjaman. [4]

### **2.4. Nasabah**

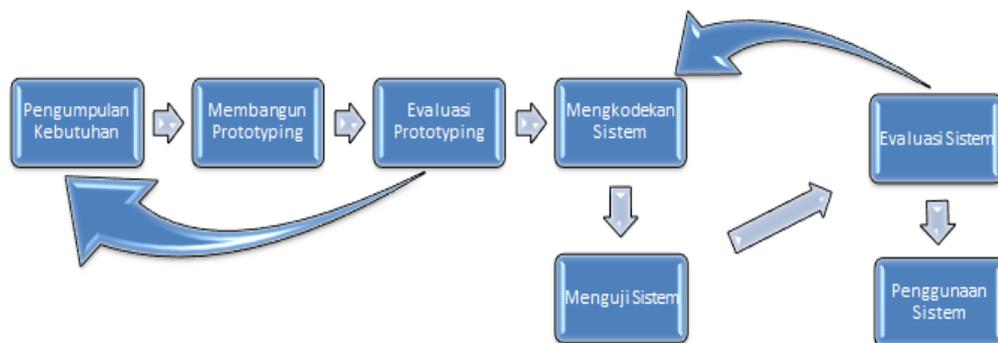
Nasabah adalah konsumen (customer) yaitu perseorangan yang memperoleh keuntungan meliputi aktivitas pembelian, penyewaan dan layanan jasa). Nasabah memiliki posisi yang berguna dalam dunia perbankan, menjadi dana yang pertama melaksanakan usahanya. Bank harus bisa meyakinkan nasabah untuk menyimpan dananya, agar bersedia menyimpan dana di bank.

Hingga pihak bank memberikan kepercayaan berbentuk balas jasa atas si penyimpan (nasabah). Balas jasa tersebut berbentuk memberi hasil, hadiah, bunga, maka akan bertambah minat masyarakat atau nasabah untuk menyimpan dananya. Oleh sebab itu, pihak bank perlu memberikan kepercayaan maka para nasabah berkeinginan untuk menyimpan dananya. [5]

### **2.5 Metode Pengembangan Sistem**

Metode *Prototype* metode ini merupakan suatu paradigm dalam pengembangan sistem yang menggunakan *metode prototype*, salah satunya adalah pendekatan yang digunakan untuk membuat sesuatu program dengan cepat dan bertahap. Hal ini memungkinkan sistem tersebut dapat dievaluasi oleh pemakai segera. [3]

*Prototyping* merupakan teknik pengembangan sistem yang menggunakan Prototype dalam menggambarkan sistem, sehingga dapat diperbaiki dan dioptimalisasi secara bertahap, dapat membantu pengembang dalam memahami dan mengembangkan sistem yang lebih efektif dan efisien pengguna atau pemilik mempunyai gambaran tentang sistem yang akan dilakukan. Dari definisi di atas dapat diambil kesimpulan bahwa dengan cara membuat model fisik kerja sistem dan berfungsi sebagai versi awal dari sebuah sistem yang dapat dikembangkan sampai perangkat lunak tersebut memenuhi kebutuhan pengguna.



**Gambar 2. 1 Metode Prototype**

Berikut adalah tahapan-tahapan dalam Prototyping dengan penjelasan setiap tahap:

1. Pengumpulan kebutuhan pelanggan: Dalam tahap ini, pengembang dan pelanggan bekerja bersama untuk mengumpulkan kebutuhan dari sisi pelanggan. Diharuskan untuk mendefinisikan format seluruh perangkat lunak, mengidentifikasi semua kebutuhan, dan menentukan garis besar sistem yang akan dibuat.
2. Membangun prototyping: Pada tahap ini, membangun sebuah prototyping dengan membuat perancangan sementara yang fokus pada penyajian kepada pelanggan. Misalnya, membuat input dan format output.
3. Evaluasi prototyping: Evaluasi dilakukan oleh pelanggan untuk memeriksa apakah prototyping yang sudah dibangun sesuai dengan keinginan pelanggan. Jika sudah sesuai, langkah 4 dilanjutkan. Jika tidak, prototyping direvisi dengan mengulangi langkah 1, 2, dan 3.
4. Mengkodekan sistem: Dalam tahap ini, prototyping yang sudah di sepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai.

5. Menguji sistem: Setelah sistem menjadi perangkat lunak yang siap pakai, dilakukan pengujian sebelum digunakan. Pengujian meliputi White Box, Black Box, Basis Path, pengujian arsitektur, dan lain-lain.

6. Evaluasi sistem: Pelanggan mengevaluasi apakah sistem yang sudah jadi sesuai dengan yang diharapkan. Jika sudah, langkah 7 dilanjutkan; jika tidak, ulangi langkah 4 dan 5.

7. Menggunakan sistem: Perangkat lunak yang telah diuji dan diterima pelanggan siap untuk digunakan.

Dalam proses ini, perhatian khusus diberikan pada komunikasi dan kerjasama yang baik antara pengembang dan pelanggan untuk memastikan sistem yang dihasilkan memenuhi kebutuhan dan expectasi dari sisi pelanggan.

## **2.6 Bahasa Pemodelan Pengembangan Sistem (*UML*)**

Unified Modelling Language (*UML*) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak, *UML* menawarkan sebuah standar untuk merancang model sebuah sistem. Tujuan Penggunaan *UML* yaitu untuk memodelkan suatu sistem yang menggunakan konsep berorientasi objek dan menciptakan bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin (Purwati, Halimah and Rahardi, 2019)

### **2.7.1 Use Case Diagram**

Use case merupakan cara paling mudah dan sederhana dalam menggambarkan fungsi sistem dan kegiatan berbagai jenis dari interaksi antar pengguna sistem. Use case merupakan sistem yang dikembangkan yang terdapat fungsionalitas atau persyaratan-persyaratan sistem yang harus dipenuhi menurut pandangan pemakai sistem. Use case dapat diartikan prosedur sistem yang secara tindakan saling terkait (skenario), baik terotomatisasi maupun secara manual, untuk tujuan melengkapi satu tugas bisnis tunggal. Use case diagram dapat disimpulkan berdasarkan pendapat para ahli adalah diagram yang dirancang yang dapat menggambarkan keseluruhan sistem yang dirancang yang terdapat interaksi

antara aktor dan sistem. [7]. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat menjelaskan simbol-simbol yang ada pada diagram *use case* dapat dilihat pada gambar 2.1 di bawah ini:

Simbol	Deskripsi
<p data-bbox="312 557 443 584"><i>Use Case</i></p> 	<p data-bbox="823 557 1359 808">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></p>
<p data-bbox="312 835 472 862"><i>Aktor/actor</i></p>	<p data-bbox="823 835 1359 1579">dalam proses pengembangan sistem informasi atau aplikasi, terdapat banyak proses atau sistem lain yang berinteraksi dengannya. Aktor yang berada dalam sistem informasi atau aplikasi dapat berupa orang, tetapi juga dapat berupa benda atau entitas lainnya yang terkait dengan tujuan sistem tersebut. Dalam penggambaran simbolik, aktor yang bukan orang sering kali dituliskan menggunakan kata benda di awal. Misalnya, aktor seperti "File", "Database", "Printer", "Koneksi Jaringan", dan lain-lalu.</p> <p data-bbox="823 1659 1359 1966">Interaksi antara sistem informasi dengan proses atau sistem lainnya dapat melibatkan pengiriman dan penerimaan data, perintah, atau informasi lainnya. Hal ini penting untuk memahami dan mengatur hubungan antara sistem</p>

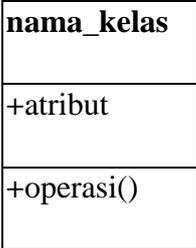
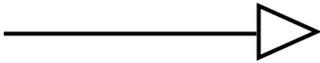
	tersebut dengan baik supaya sistem dapat berfungsi dengan efisiensi tinggi dan memenuhi kebutuhan yang diinginkan akurat dan efektif. frase nama <i>actor</i>
Asosiasi/ <i>association</i> 	berpartisipasi pada <i>use case</i> memiliki interaksi dengan <i>actor</i>
Ekstensi/ <i>extend</i>	Relasi use case tambahan ke sebuah use case artinya menambahkan sebuah skenario atau bentuk penggunaan sistem baru ke dalam use case yang sudah ada. Use case yang ditambahkan dapat berasal dari perbaikan, perubahan, ataupun ekspansi kebutuhan pengguna. Dalam konteks ini, use case yang ditambahkan disebut sebagai sub-use case atau turunan use case.
<< <i>extend</i> >> 	Anda benar, prinsip inheritance pada pemrograman berorientasi objek dapat dilihat sebagai sebuah contoh yang dapat digunakan untuk memahami cara penggunaan relasi include (termasuk) antara use case utama dan sub-use case.
Generalisasi/ <i>generalization</i> 	Hubungan generalisasi Spesialisasi (umum-khusus) antara dua buah use case dapat dilihat dari segi hubungan antara fungsi atau tujuan dari salah satu use case dengan use case lainnya, dimana fungsi atau tujuan dari salah satu lebih umum daripada yang lain. Hal ini berguna untuk memudahkan

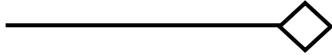
	<p>pengertian dan pemahaman antara use case yang berkaitan, serta memungkinkan kita mengembangkan sistem yang lebih efisien dan efektif.</p>
<p>Menggunakan/<i>Include/uses</i></p> <p>&lt;&lt;<i>include</i>&gt;&gt;</p> 	<p>Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan, dapat dikatakan sebagai relasi "dependen" atau "dependensi" (dependency). Dalam konteks use case, relasi ini menunjukkan bahwa salah satu use case tidak dapat dijalankan secara efektif tanpa penggunaan atau interaksi dengan use case lainnya.</p>

**Tabel 2. 1 Simbol Diagram Use Case**

### 2.7.2 Class Diagram

Diagram kelas atau *class diagram* merupakan diagram utama pada rancangan sebuah sistem. Diagram kelas digunakan untuk menampilkan kelas-kelas atau paket-paket dalam sistem dan relasi antar mereka, dari pendapat para ahli dapat disimpulkan bahwa diagram kelas merupakan antar kelas pada sebuah sistem yang dirancang [7]. pada tabel *class diagram* pada gambar 2.2. dibawah ini :

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka/ <i>Interface</i>  <b>nama_interface</b>	Sama dengan konsep interface dalam pemrograman berorientasi objek, relasi dependensi antara use case dapat dibandingkan dengan hubungan antara interface dan kelas implementasi di dunia pemrograman berorientasi objek.
Asosiasi/ <i>asociation</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity, adalah sebuah konsep penting dalam pemrograman berorientasi objek.
Asosiasi berarah/ <i>directed association</i> 	dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya menunjukkan hubungan antara kelas yang saling bergantung untuk melaksanakan tugas atau menghasilkan interaksi yang lebih baik.
Generalisasi 	makna generalisasi-spesialisasi (umum khusus)
Kebergantungan/ <i>dependecy</i> 	kebergantungan antar kelas

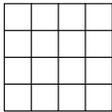
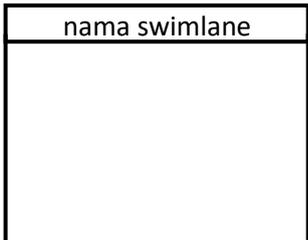
Agregasi/agregation 	dengan makna semua bagian ( <i>whole-part</i> )
--	---

**Tabel 2. 2 Simbol Class Diagram**

### 2.7.3 Activity Diagram

Activity diagram menggambarkan model dari kebiasaan dalam berbagai kegiatan dalam objek. Diagram aktivitas adalah cara untuk memodelkan alur kerja (workflow) dari use case bisnis dalam bentuk grafik. Sehingga dapat disimpulkan bahwa *activity diagram* yaitu sistem pada rancangan sistem yang dirancang oleh penulis [7]. Simbol *activity diagram* dapat dilihat pada tabel 2.3 di bawah ini :

Simbol	Deskripsi
Status awal 	sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	aktivitas biasanya diawali dengan kata kerja.
<i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu adalah konsep yang digunakan dalam pemrograman berorientasi objek untuk menunjukkan hubungan antara kelas yang memiliki beberapa opsi dalam melakukan suatu tindakan.
<i>join</i> 	Penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu dapat diartikan sebagai proses kombinasi atau kemerasan beberapa tugas atau tindakan menjadi satu entitas yang lebih besar atau

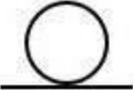
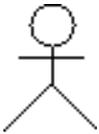
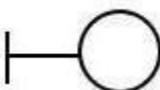
	lebih efektif.
<p>Tabel</p> 	<p>Suatu file komputer dari mana data bisa dibaca atau Direkam selama kejadian bisnis artinya, data atau informasi mengenai asosiasi percabangan dengan pilihan aktivitas lebih dari satu akan disimpan dan diberikan perhatian khusus dalam proses analisis dan pemantauan kegiatan bisnis.</p>
<p>Dokumen</p> 	<p>Menunjukkan dokumen sumber atau laporan status akhir artinya, dalam konteks bisnis atau proyek, memberikan informasi terbaru dan akurat tentang kondisi dan hasil yang telah dicapai.</p>
<p>Status akhir</p> 	<p>dilakukan sistem dalam sebuah diagram aktivitas merupakan bagian yang sangat penting dalam menganalisis proses kerja.</p>
<p>Swimlane</p> 	<p>Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi</p>

**Tabel 2. 3 Simbol Activity Diagram**

#### 2.7.4 Sequence Diagram

Sequence diagram atau diagram urutan adalah sebuah diagram yang digunakan untuk menjelaskan dan menampilkan interaksi antar objek-objek dalam sebuah system secara terperinci. Selain itu *sequence diagram* juga akan menampilkan pesan atau perintah yang dikirim, beserta waktu pelaksanaannya.

objek-objek yang berhubungan dengan berjalan proses operasi biasanya diurutkan dari kiri ke kanan. Berikut symbol-simbol yang menggambarkan *sequence diagram* :

No	Simbol	Namasimbol	Deskripsi
1		<i>Entity Class</i>	Menggambarkan hubungan yang akan dilakukan
2		Aktor	Menggambarkan orang atau sistem yang menyediakan atau menerima informasi dari sistem atau menggambarkan pengguna <i>software</i> aplikasi ( <i>user</i> )
3		<i>Boundary Class</i>	Menggambarkan sebuah gambar anda rifoem
4		<i>Control Class</i>	Control Class merupakan konsep dalam pemrograman yang digunakan untuk menggambarkan hubungan antara kelas ( <i>class</i> ) dengan boundary (batasan) dan tabel ( <i>table</i> ).
5		<i>A focus of control &amp; a life line</i>	Menggambarkan tempat mulai dan berakhirnya message
6		<i>A message</i>	Menggambarkan pengiriman pesan

**Tabel 2. 4 Simbol Sequence Diagram**

## **2.8 Alat Implementasi**

### **2.8.1 Xampp**

“XAMPP merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis dan dapat diakses secara lokal menggunakan web server local (localhost)”. [8]

### **2.8.2 MySQL**

MySQL (My Structure Query Language) adalah salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan database sebagai sumber dan pengelolaan datanya [9]

### **2.8.3 UML**

Unified Modelling Language (UML) adalah. merupakan suatu bahasa. Suatu bahasa terdiri dari kata-kata, dan memiliki aturan untuk menggabungkan kata-kata tersebut, sehingga tercipta komunikasi. Sebuah permodelan bahasa adalah suatu bahasa dimana kata-kata dan aturannya fisik

### **2.8.4 HTML**

HTML adalah Sebuah protokol yang digunakan untuk membuat format suatu dokumen web yang mampu dibaca dalam browser dari berbagai platform computer”. [8]

### **2.8.5 PHP**

PHP adalah script yang digunakan untuk membuat halaman website yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh client. Mekanisme ini menyebabkan informasi yang diterima client selalu yang terbaru/up to date. Semua script PHP dieksekusi pada server dimana script tersebut dijalankan. Dengan menggunakan program PHP, sebuah website akan lebih interaktif dan dinamis. [10]

PHP adalah scripting language yang dapat ditanamkan atau disisipkan dalam HTML. PHP digunakan secara luas untuk membuat situs web dinamis, dimana konten website akan berubah secara otomatis berdasarkan interaksi pengguna atau input lainnya. Dengan PHP, kita dapat membuat situs web yang lebih interaktif, menarik, dan berfungsi lebih baik. Beberapa contoh penggunaan PHP dalam pembuatan situs web antara lain adalah pengambilan input dari formulir, pengecekan kelayakan pengguna, dan pengelolaan database.. PHP atau kependekan dari Hypertext Preprocessor adalah salah satu bahasa pemrograman open source yang sangat cocok atau dikhususkan untuk pengembangan web dan dapat di tanamkan pada sebuah skrip HTML. [11].

### **2.8.5 Website**

“Website merupakan kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masingmasing masing dihubungkan dengan jaringan-jaringan halaman”. [9]

### **2.9 Pengujian Black Box**

Pengujian yang dilakukan untuk mengamati hasil input dan output dari perangkat lunak tanpa mengetahui struktur kode dari perangkat lunak. Pengujian di akhir pembuatan perangkat lunak, atau yang sering disebut dengan Testing in the Software Development Life Cycle (SDLC), dilakukan untuk mengetahui apakah perangkat lunak (software) dapat berfungsi dengan baik dan memenuhi spesifikasi yang ditetapkan. Proses pengujian ini memiliki tujuan untuk mengevaluasi kualitas, kehandalan, dan keamanan perangkat lunak sebelumnya diperbaiki dan diperbaharui. Berikut contoh pengujian Black Box:

<b>Data Masukkan</b>	<b>Hasil Yang Diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
----------------------	------------------------------	-------------------	-------------------

Memasukkan Username dan password, (Benar)	Masuk ke-Halaman menu <i>admin</i>	Masuk menu halaman <i>admin</i>	Diterima [ <input checked="" type="checkbox"/> ] Ditolak [ <input type="checkbox"/> ]
Memasukkan Username, (Salah)	Muncul pesan “Password Salah”	Tidak dapat masuk menu <i>admin</i>	Diterima [ <input checked="" type="checkbox"/> ] Ditolak [ <input type="checkbox"/> ]

**Tabel 2. 5 Pengujian Black Box**

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 atau kehandalan merupakan tingkat kepastian bahwa 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 waktu. Kemampuan perangkat lunak untuk mengatasi kegagalan, kesalahan, dan kembali pada tingkat tertentu dalam waktu mempengaruhi pengalaman pengguna dan efisiensi kerja mengembalikan pengembalian data yang disebabkan kegagalan atau kesalahan pada perangkat lunak.
- 3) Efisiensi kinerja atau performance efficiency merupakan tingkat kepastian bahwa perangkat lunak dapat memberikan kinerja yang maksimal dengan menggunakan sejumlah sumber daya yang tepat pada kondisi tertentu. Performance efficiency penting karena mempengaruhi efisiensi kerja, kecepatan, dan kualitas hasil dari perangkat lunak. tertentu dalam hal ini *performance efficiency*
- 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) Faktor kompatibilitas merupakan kepentingan penting dalam pengembangan perangkat lunak, karena ia memastikan bahwa dua atau lebih komponen perangkat lunak dapat berinteraksi dan berfungsi sebagaimana harusnya ketika digunakan pada hardware atau lingkungan perangkat lunak yang sama. Kompatibilitas membantu menjamin bahwa perangkat lunak dapat berjalan secara harmonis dan efektif di dalam sistem yang sudah ada.
- 7) Pemeliharaan atau *maintainability* adalah konsep penting dalam pengembangan perangkat lunak, yang membahas tingkat kehandalan suatu sistem atau aplikasi. Hal ini memastikan bahwa perangkat lunak dapat diubah, diperbaiki, atau disesuaikan untuk mengadaptasi dengan lingkungan, kebutuhan, dan fungsionalitas yang spesifik.
- 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

## **2.10 Tinjauan Pustaka**

beberapa penerapan terhadap penelitian berdasarkan dengan jurnal dibawah ini:

PHP adalah script yang digunakan untuk membuat halaman website yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh client. Mekanisme ini menyebabkan informasi yang diterima client selalu yang terbaru/up to date. Semua script PHP dieksekusi pada server dimana

script tersebut dijalankan. Dengan menggunakan program PHP, sebuah website akan lebih interaktif dan dinamis. [10]

Sistem Informasi merupakan integrasi dari teknologi informasi yang digunakan pada suatu kelompok maupun organisasi dalam membantu pengoperasian yang terdapat pada lingkungan tersebut, keberadaan sistem informasi sangat berpengaruh dalam pencapaian organisasi dikarenakan sistem informasi memiliki kelebihan untuk terus berkembang [2]

Unified Modelling Language (UML) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak, UML menawarkan sebuah standar untuk merancang model sebuah sistem. Tujuan Penggunaan UML yaitu untuk memodelkan suatu sistem yang menggunakan konsep berorientasi objek dan menciptakan bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin (Purwati, Halimah and Rahardi, 2019)

Saya mengerti bahwa Anda ingin tahu lebih dalam tentang Metode Prototype dalam pengembangan sistem yang menggunakan pendekatan untuk membuat sebuah program dengan cepat dan bertahap. Metode Prototype adalah salah satu teknik pengembangan perangkat lunak yang membantu mengubah ide menjadi produk yang konkret dan dapat diperbaiki. Dalam proses ini, pengembang membuat sebuah "prototipe" atau versi awal dari program yang dapat diuji dan diperbaiki. [3]

Metode penelitian yang digunakan adalah sebuah model proses yang diterapkan saat menjalankan komunikasi dengan *client* untuk membuat sebuah aplikasi, prototype tidak menyajikan bentuk asli sistem secara lengkap akan tetapi metode prototype berperan penting dalam penelitian untuk memberikan gambaran aplikasi yang akurat terhadap client [2]

Bank adalah lembaga bisnis yang bertujuan untuk mengumpulkan dana dari masyarakat dalam bentuk simpanan atau deposit. Kemudian, bank ini memanfaatkan dana tersebut untuk memberikan pinjaman atau investasi kepada

masyarakat. Proses ini membantu memenuhi kebutuhan finansial dari masyarakat serta memperkuat ekonomi secara keseluruhan. Bank juga menyediakan jasa-jasa lainnya seperti transfer uang, pengelolaan investasi, dan layanan lain yang berkaitan dengan kebutuhan finansial individu dan perusahaan dalam bentuk kredit atau bentuk lainnya dengan (UU No 10 Tahun 1998 Perubahan UU No 7 Tahun 1992 tentang perbankan).

Bank memanfaatkan dana yang diumpangkan dalam bentuk simpanan atau deposit oleh mereka yang tidak dapat atau tidak memerlukan sepenuhnya untuk memberikan pinjaman atau investasi kepada masyarakat. Hal ini membantu memenuhi kebutuhan finansial dari individu dan perusahaan yang memerlukan modal untuk kegiatan bisnis, investasi, atau pengembangan. Proses ini menjadi salah satu cara untuk mengoptimalkan penggunaan sumber daya ekonomi dan membantu pengembangan ekonomi secara keseluruhan.[4]