

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Penelitian terdahulu yang dilakukan oleh Dilla Khairiah 2017, Analisis Pembiayaan Konsumtif Pada PT. Bank Sumut Syariah Cabang Pembantu Karya Medan. Berkesimpulan bahwa Faktor-faktor yang mempengaruhi permintaan pembiayaan konsumtif pada PT. Bank Sumut Syariah Cabang Pembantu Karya Medan ditinjau dari beberapa aspek yaitu, kemampuan keuangan, kemampuan daya beli wilayah, tingkatan pendapatan atau gaji berdasarkan level industri, tingkatan pendapatan atau gaji berdasarkan sektor ekonomi, dan tingkatan pendapatan bonus atau jasa produksi berdasarkan level industri.

Penelitian terdahulu yang dilakukan oleh Sri Utami 2020, Perancangan Aplikasi Sistem Peminjaman Online Berbasis Android Desa Bugis. Aplikasi peminjaman berbasis android mudah digunakan dengan tampilan menarik sehingga petani nyaman dalam menggunakannya. Dengan tata letak yang rapi memberikannilai efektif secara fungsional.

Penelitian terdahulu yang dilakukan oleh Penda Sudarto Hasugian (2018), Perancangan Website Sebagai Media Promosi Dan Informasi. Dengan adanya website sebagai media promosi dan informasi pada ves boutique, dapat membantu

memasarkan dan mempublikasikan berita, produk, kegiatan, diskon dan sebagainya yang akan diselenggarakan dalam penyampaian informasi dan promosi *Ves Boutique*.

Penelitian Berdasarkan Penelitian yang dilakukan oleh Bagus Dwi Prasetyo (2017), Aplikasi ini dapat memberikan kemudahan dalam perhitungan kredit secara akurat dan cepat karena perhitungan biaya pokok, biaya angsuran, dan biaya bunga dapat dilakukan hanya dengan menginput harga atau pokok pinjaman, uang muka dan lama jangka waktu kreditnya.

Penelitian terdahulu yang dilakukan Aris Kurniawan (2019), Dalam prasyarat kredit konsumtif yang berlaku pada PT BPR Intidana mewajibkan calon nasabah memenuhi syarat utama, seperti: jaminan BPKB Mobil/Motor (asli), sertifikat rumah, mempunyai penghasilan tetap. Sedangkan syarat administrasi, seperti FC BPKB, faktur, STNK, FC SHM/ SHGB dan PBB terakhir, FC KTPsuami/istri KK dan surat nikah, FC rekening listrik dan telepon, FC SlipGaji/keterangan penghasilan, serta FC rekening tabungan.

II.2. Landasan Teori

Beberapa landasan teori yang dikutip dari beberapa referensi penelitian yang berkaitan dengan penelitian ini yaitu :

II.2.1. Implementasi

Proses implementasi ada proses penterjemahan dari pemodelan ke dalam pengkodean atau pembentukan antarmuka. Implementasi merupakan proses

penterjemahan dari pemodelan menjadi bentuk aplikasi sesuai dengan kebutuhan informasi pengguna. (Oktaviani dan Sauda, 2019 : 182).

II.2.1.1. Fungsi Implementasi

Fungsi implementasi adalah mentransformasikan tujuan kebijakan ke dalam bentuk-bentuk kegiatan operasional yang dibutuhkan agar kebijakan mencapai tujuannya. Fungsi implementasi tidak akan berubah, sekali-pun kebijakan yang diimplementasikan berbeda, yang berbeda adalah hasil akhirnya. (Purwanti, dkk, 2019 : 13).

II.2.1.2. Tujuan Implementasi

Tujuan implementasi itu sendiri adalah menghasilkan perubahan sebagaimana yang dikehendaki oleh kebijakan. Oleh karenanya jika dikatakan bahwa sebuah kebijakan dibuat untuk mengintervensi peri kehidupan publik, maka implementasi adalah bentuk nyata dari tindakan intervensi itu sendiri. (Purwanti, dkk, 2019 : 13).

II.2.1.3. Jenis-Jenis Implementasi

Berikut ini terdapat beberapa jenis-jenis implementasi, antara lain:

1. Implementasi Kebijakan

Publik Implementasi kebijakan Publik adalah suatu tahapan administrasi yang dilakukan setelah kebijakan atau aturan hukum ditetapkan/disetujui melalui proses politik.

2. Implementasi Sistem

Implementasi sistem adalah suatu proses untuk menempatkan serta menempatkan informasi baru ke dalam operasi.

3. Implementasi Strategi

Proses mewujudkan dan menerapkan strategi yang sudah dibuat ke dalam bentuk tindakan melalui berbagai prosedur, program, dan anggaran.

4. Implementasi Kebijakan

Implementasi kebijakan adalah suatu proses dalam melaksanakan suatu kebijakan tertentu kemudian mengembangkannya dengan tujuan untuk menyempurnakan suatu program.

5. Implementasi Keperawatan

Suatu tahapan kegiatan yang dilakukan oleh seorang perawat dalam membantu klien dari status kesehatan yang bermasalah menuju status kesehatan yang lebih baik lagi. (Purwanti, dkk, 2019 : 13).

II.2.2 Pinjaman Konsumtif

kredit konsumtif merupakan kredit yang diberikan kepada nasabah untuk membeli barang dan jasa untuk keperluan pribadi dan tidak untuk digunakan keperluan usaha. Kredit konsumtif adalah kredit yang digunakan untuk membiayai pembelian barang-barang atau jasa-jasa yang dapat memberi kepuasan langsung terhadap kebutuhan manusia. Berdasarkan penjelasan diatas dapat disimpulkan bahwa

kredit konsumtif merupakan kredit yang disalurkan kepada nasabah untuk memenuhi kebutuhan-kebutuhan konsumsinya (Putri, dkk, 2015)

II.2.3 Bunga Flat

Bunga *flat* ialah Pembebanan bunga setiap bulan tetap dari jumlah pinjamannya, demikian pula pokok pinjaman setiap bulan juga dibayar sama, sehingga setiap bulan juga sama sampai kredit tersebut lunas. Jenis bunga ini diberikan kepada kredit yang berjangka pendek atau bersifat konsumtif seperti pembelian rumah tinggal, pembelian mobil pribadi atau kredit konsumtif lainnya yang merupakan kredit yang memiliki jangka waktu kurang dari 1 tahun atau paling lama 1 tahun. (Raymond, 2022)

II.2.4 Menentukan Cicilan Perbulan dan Rasio Kredit

Saat melakukan pinjaman konsumtif pengguna akan diberitahukan bahwa pembayaran pinjaman menggunakan bunga flat. Angsuran per bulannya harus disertai dengan nominal bunga yang telah ditentukan oleh pihak bank. Bunga merupakan bentuk kompensasi, karena menjadi penyedia dana untuk dipinjamkan, tidak hanya bunga biasanya pihak penyedia layanan juga menambahkan biaya administrasi yang dibebankan dalam cicilan ataupun potongan langsung. Dalam menghitung cicilan perbulan harus diketahui jumlah bunga yang dihitung dengan rumus :

$$\text{Bunga} = p\% \times \frac{b}{12} \times (\text{modal} + \text{admin})$$

Setelah nilai bunga di ketahui maka dapat menghitung cicilan per bulan dengan rumus :

$$\text{Angsuran} = (\text{bunga} + \text{modal} + \text{admin})/b$$

Ket : p = Bunga per tahun

b = lama Pinjaman

Dalam setiap pinjaman yang diberikan oleh bank tentunya terdapat beberapa resiko seperti telat membayar cicilan dan tidak membayar cicilan yang akan merugikan pihak bank, untuk mengurangi resiko tersebut pihak bank menerapkan rasio kredit. Rasio kredit dibawah 20% adalah rasio paling ideal Sebab besaranutangnya dianggap masih layak dan sehat, sedangkan rasio kredit antara 20-36% masih dipertimbangkan oleh pihak penyedia layanan dengan pengecekan riwayatpinjaman, sementara untuk rasio kredit diatas 36% akan sulit menerima pinjaman. Untuk menentukan rasio kredit ditentukan dengan rumus :

$$\text{Rasio Kredit} = (a/c) \times 100$$

Ket : a = hutang per bulan

c = penghasilan per bulan

II.2.5 Android

Sistem operasi adalah *software* utama yang melakukan manajemen dan kontrol terhadap *hardware* secara langsung serta manajemen dan mengontrol *software-software* lain sehingga *software-software* lain tersebut dapat bekerja. Salah satu sistem operasi khususnya untuk perangkat *mobile* adalah *Android. Operating System (OS)* *Android* dibangun berbasis platform *Linux* yang bersifat *opensource*, senada dengan *Linux*, *Android* juga bersifat *Open Source*. Dengan nama besar *Google*

dan konsep *opensource* pada OS *Android*, tidak membutuhkan waktu lama bagi *android* untuk bersaing dan menyisihkan *Mobile OS* lainnya seperti *Symbian*, *Windos Mobile*, *Blackberry* dan *iOS*. Sistem operasi *Android* sendiri mempunyai beberapa versi, yang terbaru adalah versi 9.0 dengan kode name *Pie*.. (Akmal, 2019)

II.2.6. *Android Studio*

Android Studio adalah sebuah IDE untuk *Android Development* yang diperkenalkan *google* pada acara *GoogleI/O 2013*. *Android Studio* merupakan pengembangan dari *Eclipse IDE*, dan dibuat berdasarkan IDE *Java* populer, yaitu *IntelliJ IDEA*. *Android Studio* merupakan IDE resmi untuk pengembangan aplikasi *Android*.

Sebagai pengembangan dari *Eclipse*, *Android Studio* mempunyai banyak fitur-fitur baru dibandingkan dengan *Eclipse IDE*. Berbeda dengan *Eclipse* yang menggunakan *Ant*, *Android Studio* menggunakan *Gradle* sebagai *build environment*. Fitur-fitur lainnya adalah sebagai berikut :

1. Menggunakan *Gradle-based build system* yang *fleksibel*.
2. Bisa *mem-build multiple AP*
3. *Template support* untuk *Google Services* dan bermacam tipe perangkat.

4. *Layout editor yang lebih bagus*
5. *Built-in support untuk Google Cloud Platform, sehingga mudah untuk integrasi dengan Google Cloud Messaging dan App Engine.*
6. *Import library langsung dari Maven repository dan masih banyak lagi lainnya.*(Salman, 2019)

II.2.7. Unified Modeling Language (UML)

UML yaitu satu kumpulan konvensi permodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem perangkat lunak yang terkait dengan objek. UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan. (Andikos, 2019 : 39).

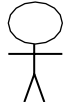
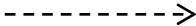
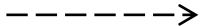
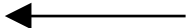
Alat bantu yang digunakan dalam perancangan berorientasi objek berdasarkan UML adalah sebagai berikut :

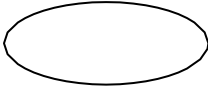
1. Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor

dengan sistem. *Use case diagram* dapat digambarkan dengan sumber-sumber pada Tabel II.1.

Tabel II. 1 Simbol *Use Case*

Gambar	Nama	Keterangan
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang mnghubungkan antara objek satu dengan objek lainnya.
	<i>System</i>	Menspesifikan paket yang menampilkan sistem secara terbatas.

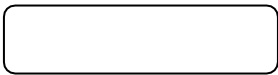
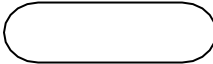
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemennya (sinergi).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

(Sumber : Andikos, 2019 : 39)

2. Diagram Aktivitas (*Activity Diagram*)

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* dapat digambarkan dengan simbol-simbol seperti pada tabel II.2.

Tabel II. 2 Simbol *Activity Diagram*

Gambar	Nama	Keterangan
	<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	<i>Action</i>	<i>State</i> dari sistem yang mencerminkan eksekusi dari suatu

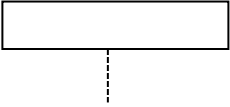
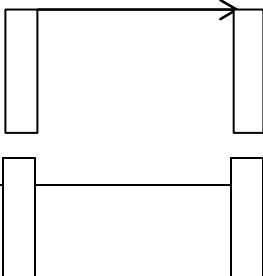
		aksi.
	<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
	<i>Activity Final</i>	Bagaimana objek dibentuk dan dihancurkan
	<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

(Sumber : Andikos, 2019 : 39)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence Diagram* dapat digambarkan dengan simbol-simbol seperti pada Tabel II.3.

Tabel II. 3 Simbol *Sequence Diagram*

Gambar	Nama	Keterangan
	<i>Lifeline</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

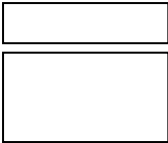
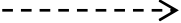
(Sumber : Andikos, 2019 : 39)

4. *Class Diagram* (Diagram Kelas)

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class diagram* dapat digambarkan dengan simbol-simbol seperti pada Tabel II.4.

Tabel II. 4 Simbol *Class Diagram*

Gambar	Nama	Keterangan
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.

	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Depedency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
	<i>Assocation</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

(Sumber : Andikos, 2019 : 39)