

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Adapun penelitian terkait yang akan digunakan sebagai sumber acuan yang relevan dan terkini yaitu:

Menurut Andri Anto Tri S (2018) dalam jurnal yang berjudul "Sistem Informasi Pembayaran Kredit Pada Prisma Motor Lubuklinggau Menggunakan Metode Flat Rate" Dengan sistem seperti itu menyebabkan sering terjadinya kesalahan yang dapat berdampak pada Pembayaran angsuran secara keseluruhan. Dengan dibangunnya aplikasi ini akan membantu proses pengelolaan data angsuran agar lebih efisien. Dalam membangun aplikasi menggunakan Visual Basic.Net 2010. Apabila aplikasi ini digunakan maka diharapkan dapat mempermudah memberikan informasi kepada pimpinan dan laporan yang dihasilkan merupakan data yang efektif.

Berdasarkan penelitian dari Imam Triyuniadi (2019) dengan judul "Sistem Informasi Penjualan Kredit Sepeda Motor Pada CV. Jaya Abadi Motor Semarang Menggunakan Metode Flat Rate" Dalam hal ini, pegawai maupun pihak yang mengolah data-data mengalami banyak kendala, seperti yang terjadi pada CV. Jaya Abadi Motor Semarang kendala yang ada adalah : kesulitan dalam pencarian data konsumen, kesulitan dalam proses pencatatan transaksi penjualan kredit sepeda motor dan angsuran, kesulitan dalam pembuatan laporan-laporan penjualan kredit sepeda motor dan pembayaran angsuran dari konsumen seperti

halnya laporan pembayaran angsuran dari konsumen, belum ada koneksi antar data antar bagian sehingga menyebabkan pengontrolan data sangat sulit dilakukan

Menurut Siagian Maruli (2018) yang berjudul “Sistem Informasi Analisis Komparatif Kredit Dengan Perhitungan Bunga Flat Dan Sliding Pada PT. BPR Permata Hati Jaya Samarinda” Penerapan metode *flat rate* dan *sliding* memiliki perbedaan yang signifikan pada PT. BPR Permata Hati Jaya Samarinda dimana perusahaan melakukan perhitungan suku bunga dengan nilai bersih laba. Dan nilai laba per tahun lebih tinggi untuk semua jangka waktu kredit yaitu metode flat 22% dan metode *sliding* 36%.

Menurut Swasnita (2018), yang berjudul “Sistem Informasi Perhitungan Suku Bunga Sliding Rate Untuk Penentuan Alternatif Pembiayaan Kendaraan Motor Pada *Leasing* Dan Bank Dengan Metode Interpolasi Linier (Studi Kasus Harga Sepeda Motor Honda Beat Injeksi Terdaftar Bulan September 2014)” Dalam penentuan suku bunga dengan menggunakan *sliding* diterapkan dengan baik dan dalam pembuatan laporan menghasilkan laporan efektif dan relevan. Namun, dalam penentuan cicilan tiap bulan kurang efisien dikarenakan sistem yang diterapkan belum terkomputerisasi. Selain itu, yang menganalisis pengadaan kendaraan operasional berupa mobil menunjukkan bahwa kredit bank lebih menguntungkan daripada menggunakan *leasing* karena keuntungan yang diperoleh berupa *present value* kredit bank lebih kecil.

Menurut Hedwigis Esti (2018) dalam jurnal yang berjudul “Sistem Informasi Analisis Pengaruh Tingkat Suku Bunga Kredit Dan Simpanan Nasabah Terhadap Jumlah Penyaluran Kredit Investasi Pada Bank Persero” Penelitian ini

bertujuan untuk menganalisis pengaruh tingkat suku bunga kredit dan simpanan nasabah terhadap kredit investasi pada Bank Persero secara parsial maupun secara simultan. Data yang digunakan adalah data sekunder berupa data kredit investasi, tingkat suku bunga kredit, dan posisi simpanan nasabah pada Bank Persero. Sampel yang digunakan dalam penelitian ini diambil dari data series kredit investasi, tingkat suku bunga kredit, dan simpanan nasabah yang dipublikasikan oleh Bank Indonesia. Metode analisis data yang digunakan dalam penelitian ini adalah uji hipotesis t Test dan F Test, dan uji koefisien determinasi berganda (R^2). Hasil penelitian menunjukkan bahwa secara parsial tingkat suku bunga kredit memiliki pengaruh negatif signifikan, sedangkan simpanan nasabah memiliki pengaruh positif signifikan terhadap kredit investasi. Secara simultan, tingkat suku bunga kredit dan simpanan nasabah memiliki pengaruh yang signifikan terhadap kredit investasi.

II.2. Landasan Teori

II.2.1. Sistem

Sistem dapat terdiri dari sistem-sistem bagian (*subsystem*). Sebagai misal, sistem komputer dapat terdiri dari subsistem perangkat keras dan subsistem perangkat lunak. Masing-masing subsistem dapat terdiri dari subsistem-subsistem yang lebih kecil atau terdiri dari komponen-komponen. Subsistem perangkat keras (*hardware*) dapat terdiri dari alat masukan, alat pemroses, alat keluaran dan simpanan luar, dan kemudian subsistem-subsistem tersebut akan berinteraksi sedemikian rupa sehingga dapat mencapai satu kesatuan yang terpadu. Dalam

buku Analisa dan *Design* Sistem Informasi Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis. Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu (Linda, 2018).

II.2.1.1. Karakteristik Sistem

Untuk memahami atau mengembangkan suatu sistem, maka perlu membedakan unsur-unsur dari sistem yang membentuknya. Berikut adalah karakteristik sistem yang dapat membedakan suatu sistem dengan sistem lainnya yaitu:

1. Batasan (*Boundary*) adalah Penggambaran dari suatu elemen atau unsur mana yang termasuk di dalam sistem dan mana yang di luar sistem.
2. Lingkungan (*Environment*) adalah Segala sesuatu di luar sistem, lingkungan yang menyediakan asumsi, kendala, dan input terhadap suatu sistem.
3. Masukan (*Input*) adalah Sumber daya (data, bahan baku, peralatan, energi) dari lingkungan yang dikonsumsi dan dimanipulasi oleh suatu sistem.
4. Keluaran (*Output*) adalah Sumber daya atau produk (informasi, laporan, dokumen, tampilan layar Komputer, barang jadi) yang disediakan untuk lingkungan sistem oleh kegiatan dalam suatu sistem.
5. Komponen (*Component*) adalah Kegiatan-kegiatan atau proses dalam sistem yang mentransformasikan input menjadi bentuk setengah jadi (*output*). Komponen ini bisa merupakan subsistem dari sebuah sistem.

6. Penghubung (*Interface*) adalah Tempat di mana komponen atau sistem dan lingkungannya bertemu atau berinteraksi.
7. Penyimpanan (*Storage*) adalah Area yang dikuasai dan digunakan untuk penyimpanan sementara dan tetap dari informasi, energi, bahan baku, dan sebagainya. Penyimpanan merupakan suatu media penyangga di antara komponen tersebut bekerja dengan berbagai tingkatan yang ada dan memungkinkan komponen yang berbeda dari berbagai data yang sama (Aris : 2017).

II.2.2. Informasi

Informasi dapat diibaratkan sebagai darah yang mengalir didalam tubuh manusia, seperti halnya informasi di dalam sebuah perusahaan yang sangat penting untuk mendukung kelangsungan perkembangannya, Sehingga terdapat alasan bahwa informasi sangat dibutuhkan bagi sebuah perusahaan. Akibat bila kurang mendapatkan informasi, dalam waktu tertentu perusahaan akan mengalami ketidakmampuan mengontrol sumber daya, sehingga dala mengambil keputusan-keputusan strategis sangat terganggu, yang pada akhirnya akan mengalami kekalahan dalam bersaing dengan lingkungan pesaingnya. Disamping itu, sistem informasi yang dimiliki seringkali tidak dapat bekerja dengan baik. Masalah utamanya adalah bahwa sistem informasi tersebut terlalu banyak informasi yang tidak bermanfaat atau berarti sistem terlalu banyak data. (Andri Kristanto : 2018 : 7)

II.2.3. Sistem Informasi

Sebuah sistem mempunyai tujuan atau sasaran berpendapat sistem adalah sekelompok elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan, mendefinisikan sistem sebagai seperangkat elemen-elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan bersama. Sistem informasi merupakan kumpulan dari sub-sub sistem baik fisik maupun non fisik yang berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan yaitu mengolah data menjadi informasi yang berarti dan berguna. Berdasarkan dari kedua definisi di atas, dapat disimpulkan bahwa sistem informasi adalah kumpulan dari sub-sub sistem dalam organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan (Ula, 2018).

II.2.3.1. Faktor - faktor Sistem Informasi

Faktor-faktor yang menentukan kehandalan dari suatu sistem informasi atau informasi dapat dikatakan baik jika memenuhi kriteria-kriteria sebagai berikut :

a) Keunggulan (*usefulness*)

Yaitu suatu sistem yang harus dapat menghasilkan informasi yang tepat dan relevan untuk mengambil keputusan manajemen dan personil operasi dalam organisasi.

b) Ekonomis

Kemampuan sistem yang mempengaruhi sistem harus bernilai manfaat minimal, sebesar biayanya.

c) Keandalan (*Reliability*)

Keluaran dari sistem harus mempunyai tingkat ketelitian tinggi dan sistem tersebut harus beroperasi secara efektif.

d) Pelayanan (*Customer Service*)

Yakni suatu sistem memberikan pelayanan yang baik dan efisien kepada para pengguna sistem pada saat berhubungan dengan organisasi.

e) Kapasitas (*Capacity*)

Setiap sistem harus mempunyai kapasitas yang memadai untuk menangani setiap periode sesuai yang dibutuhkan.

f) Sederhana dalam kemudahan (*Simplicity*)

Sistem tersebut lebih sederhana (umum) sehingga struktur dan operasinya dapat dengan mudah dimengerti dan *prosedure* mudah diikuti.

g) Fleksibel (*Flexibility*)

Sistem informasi ini harus dapat digunakan dalam kondisi yang bagaimana yang diinginkan oleh organisasi tersebut atau pengguna tertentu.

h) Komponen Sistem Informasi

Istilah dalam komponen sistem informasi adalah blok bangunan (*building block*) yang dapat di bagi menjadi enam blok yaitu :

a. Blok masukan (*input block*)

Blok input merupakan data–data yang masuk ke dalam sistem informasi, yang dapat berupa *document-document* dasar yang dapat diolah menjadi suatu informasi tertentu.

b. Blok model (*model block*)

Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan mengolah data *input* untuk menghasilkan suatu informasi yang dibutuhkan.

c. Blok keluaran (*output block*)

Merupakan informasi yang menghasilkan sekumpulan data yang nantinya akan disimpan berupa data cetak laporan.

d. Blok teknologi (*technology block*)

Blok teknologi merupakan penunjang utama dalam berlangsungnya sistem informasi. Yang memiliki beberapa komponen yaitu diantaranya alat memasukan data (*input device*), alat untuk menyimpan dan mengakses data (*storage device*), alat untuk menghasilkan dan mengirimkan keluaran (*output device*) dan alat untuk membantuk pengendalian sistem secara keseluruhan (*control device*). Teknologi informasi terdiri dari 3 (tiga) bagian utama, yaitu teknisi (*humanware* atau *brainare*), perangkat lunak (*software*), dan perangkat keras (*hardware*).

e. Blok basis data (*database block*)

Basis data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan

digunakan perangkat lunak untuk memanipulasinya. Data perlu di simpan dan perlu di organisasi sedemikian rupa, supaya informasi yang dihasilkan berkualitas.

f. Blok kendali (*control block*)

Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat di cegah bila terlanjur terjadi. (I Made Budi Adnyana : 2019)

II.2.4. Kredit

Kredit sudah menjadi sebuah jenis produk keuangan yang umum di tengah masyarakat. Sebagai contoh Kredit Pemilikan Rumah (KPR), Kredit Kendaraan Bermotor (KKB), dan lain-lain. Kredit tentunya memiliki banyak definisi, misalnya saja jika diambil dari bahasa Yunani, kredit adalah “*credere*” yang artinya kepercayaan akan kebenaran atau “*credo*” yang berarti saya percaya. Penyediaan uang atau tagihan yang dapat disamakan dengan itu, berdasarkan kesepakatan pinjam meminjam antara pihak bank dengan pihak lain yang mewajibkan pihak peminjam untuk melaksanakan dengan jumlah bunga sebagai imbalan (Hedwigis Esti : 2017).

II.2.5. Metode Flat Rate

Flat rate adalah pembebanan suku bunga tetap setiap periode, sehingga jumlah angsuran yang harus dibayarkan debitur setiap bulan tetap, sampai kredit tersebut lunas. Suku bunga dihitung dengan mengalikan persen bunga per periode

dikali dengan pinjaman. Bunga flat adalah bunga yang jumlah pembayaran pokok dan bunga kredit yang besarnya sama setiap bulan. biasanya diperuntukkan untuk kredit jangka pendek seperti kredit kendaraan bermotor dan kredit tanpa agunan. Bunga flat ini menggunakan metode perhitungannya yang jelas dimana besaran angsuran pokok dan angsuran bunga tak berubah setiap bulannya. Semuanya dihitung secara proporsional dan disesuaikan dengan tenor (jangka waktu pinjaman yang ditetapkan atas kesepakatan debitur dan kreditur). Si A mendapatkan pokok pinjaman dari bank senilai Rp36 juta. Dari pinjaman tersebut Si A diberikan bunga flat sebesar 5% per tahun. Dan disepakati jangka waktu pengembalian utang selama 24 bulan (2 tahun). (Mamahit : 2018) Maka perhitungan bunganya dapat diketahui dengan menggunakan rumus:

$$\text{Bunga per bulan} = (P \times I) / j_b$$

P = pokok pinjaman

I = suku bunga per tahun

tjb = jumlah bulan dalam jangka waktu kredit (Mamahit : 2018)

contoh kasus :

Contoh kasus:

Andi mengajukan pinjaman sebesar Rp120 juta dengan bunga 2 persen per bulan. Andi memilih tenor cicilan selama 24 bulan. Berapa jumlah uang yang harus Andi bayarkan setiap bulan dengan perhitungan bunga *flat*?

Jadi,

Jumlah pinjaman = Rp120.000.000

Bunga = 2 persen per bulan

Jangka waktu = 24 bulan

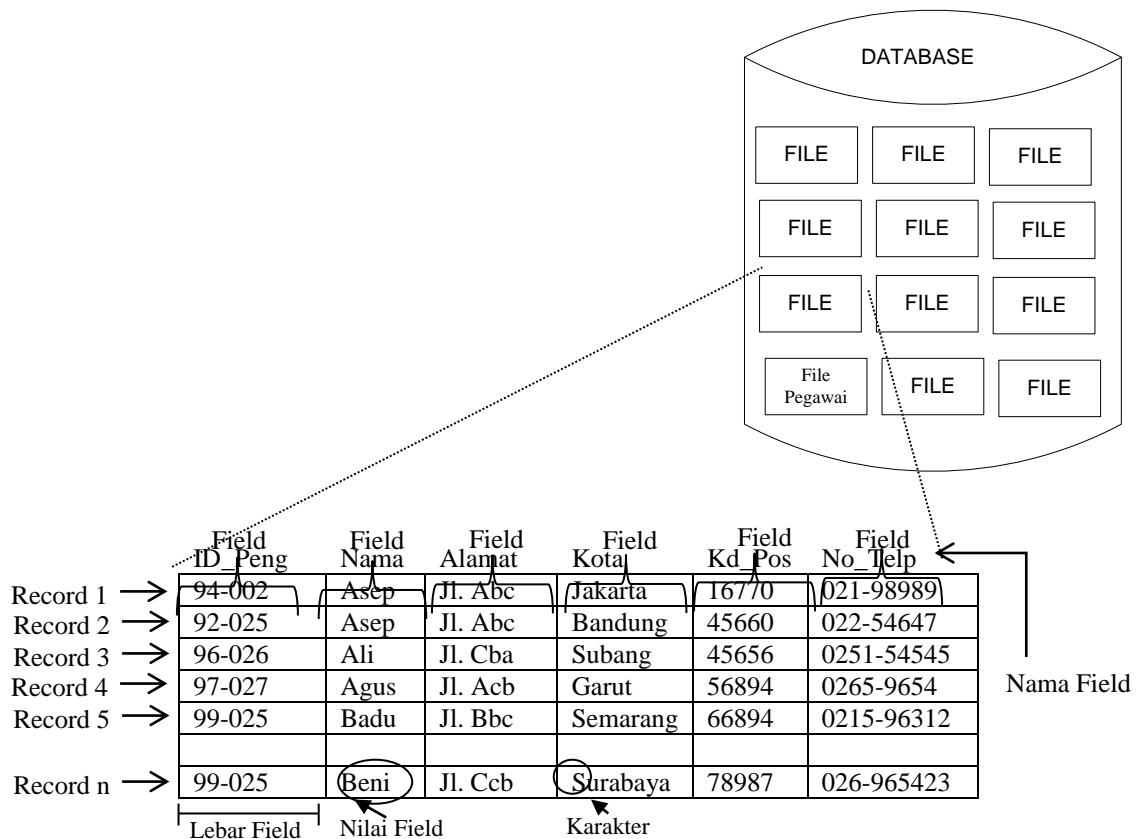
Perhitungan:

- Angsuran pokok per bulan = $\text{Rp}120.000.000 / 24 \text{ bulan} = \text{Rp}5.000.000$.
- Angsuran bunga per bulan = $2\% \times \text{Rp}120.000.000 = \text{Rp}2.400.000$.
- Total angsuran per bulan = $\text{Rp}5.000.000 + \text{Rp}2.400.000 = \text{Rp}7.400.000$.

Dari perhitungan di atas, maka Andi harus membayar angsuran per bulan sebesar Rp7.400.000 selama 24 bulan hingga utang lunas.

II.2.6. Database

Database merupakan kumpulan dari item data yang saling berhubungan satu dengan yang lainnya yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, tersimpan di *hardware* komputer dan dengan *software* untuk melakukan manipulasi untuk kegunaan tertentu *Database* dapat juga diartikan Koleksi data yang terorganisasi untuk melayani beragam aplikasi secara efisien dengan mensentralisasi data dan meminimalisasi data yang berlebih. Sebagian besar aplikasi *database* memiliki bahasa yang khas yang disebut bahasa manipulasi data yang digunakan dalam hubungannya dengan beberapa bahasa pemrograman aplikasi konvensional untuk memanipulasi data dalam *database*. Bahasa ini mengandung perintah-perintah yang memungkinkan pengguna akhir dan para ahli pemrograman untuk mengekstrak data dari *database* untuk memenuhi kebutuhan informasi dan mengembangkan aplikasi. Bahasa manipulasi yang paling banyak digunakan dewasa ini adalah *Structured Query Language* (SQL).



Gambar II.1. Hirarki Database
Sumber : Mukhlisulfatih Latief : 2019

Bahasa SQL tersusun atas 3 kelompok pernyataan berdasarkan fungsi dari pernyataan tersebut yaitu:

1. *Data Definition Language (DDL)* : Mendefinisikan jenis data yang akan dibuat (dapat berupa angka atau huruf), cara relasi data, validasi data dan lainnya. Contoh : *create, drop, alter table*.
2. *Data Manipulation Language (DML)* : Data yang telah dibuat dan didefinisikan tersebut akan dilakukan beberapa pengerjaan, seperti menyaring data, melakukan proses *query*. Contoh : *select, update, insert*.

3. *Data Control Language* (DCL) : Bagian ini berkenaan dengan cara mengendalikan data, seperti siapa saja yang bisa melihat isi data, bagaimana data bisa digunakan oleh banyak *user*.

II.2.7. Normalisasi

Normalisasi adalah proses pembentukan struktur basis data sehingga sebagian besar ambiguity bisa dihilangkan. Normalisasi merupakan sebuah teknik dalam logical desain sebuah basis data relasional yang mengelompokkan atribut dari suatu tabel sehingga membentuk struktur tabel yang normal. Adapun kriteria tabel dikatakan normal adalah ketika tidak ada kerangkapan data (redundansi data).

Tujuan dari normalisasi adalah untuk :

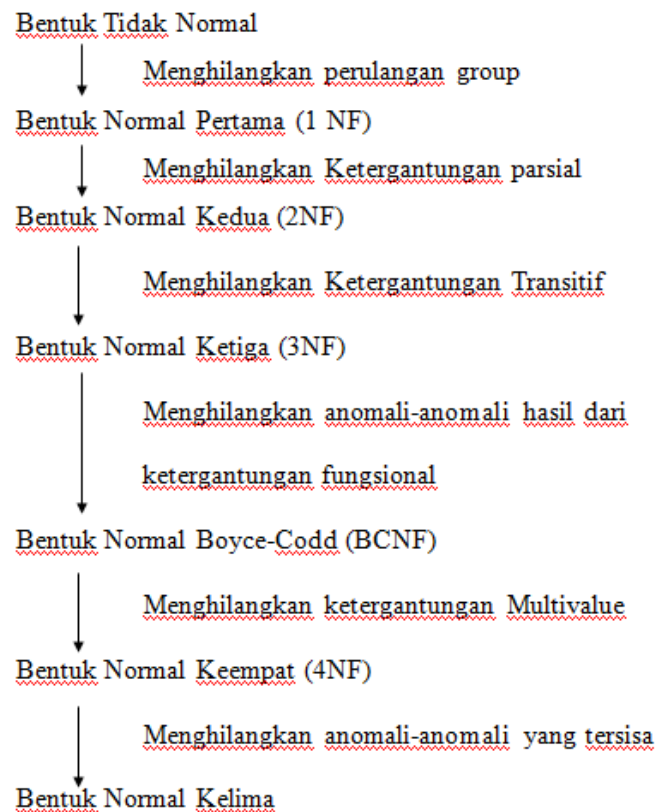
1. Untuk menghilangkan kerangkapan data sehingga meminimumkan pemakaian *storage* yang dipakai oleh *base relations* (file)
2. Untuk mengurangi kompleksitas
3. Untuk mempermudah pemodifikasian data

Gambaran proses normalisasi adalah :

1. Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu ke beberapa tingkat, kemudian
2. Apabila tabel yang diuji belum memenuhi persyaratan tertentu, maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk yang optimal.

Untuk melakukan proses tersebut dibutuhkan beberapa tahapan. Tahapan dalam normalisasi dimulai dari tahap paling ringan (1NF) hingga paling ketat

(5NF). Biasanya hanya sampai pada tingkat 3NF atau BCNF karena sudah cukup memadai untuk menghasilkan tabel-tabel yang berkualitas baik. Urutan tahapan normalisasi tampak seperti gambar II.2.



Gambar II.2 Tahapan Pada Normalisasi

Adapun aturan dalam normalisasi adalah suatu tabel dikatakan baik (efisien) atau normal jika memenuhi 3 kriteria sbb:

1. Jika ada dekomposisi (penguraian) tabel, maka dekomposisinya harus dijamin aman (*Lossless-Join Decomposition*). Artinya, setelah tabel tersebut diuraikan / didekomposisi menjadi tabel-tabel baru, tabel-tabel baru tersebut bisa menghasilkan tabel semula dengan sama persis.
2. Terpeliharanya ketergantungan fungsional pada saat perubahan data (*Dependency Preservation*).

3. Tidak melanggar Boyce-Codd Normal Form (BCNF)

Jika kriteria ketiga (BCNF) tidak dapat terpenuhi, maka paling tidak tabel tersebut tidak melanggar Bentuk Normal tahap ketiga (3rd Normal Form / 3NF). Pada penelitian ini formula yang dibuat sampai memenuhi bentuk normal ke 3 yaitu 3NF. (Dwi Puspitasari, 2019)

II.2.8. PHP

PHP atau kependekan dari *Hypertext Preprocessor* adalah salah satu bahasa pemrograman *open source* yang sangat cocok atau dikhususkan untuk pengembangan *web* dan dapat ditanamkan pada sebuah skripsi HTML. Bahasa PHP dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti C, *Java*, dan *Perl* serta mudah untuk dipelajari. PHP merupakan bahasa *scripting server – side*, dimana pemrosesan datanya dilakukan pada sisi *server*. Sederhananya, *server*lah yang akan menerjemahkan skrip program, baru kemudian hasilnya akan dikirim kepada *client* yang melakukan permintaan. Adapun pengertian lain PHP adalah akronim dari *Hypertext Preprocessor*, yaitu suatu bahasa pemrograman berbasis kode – kode (*script*) yang digunakan untuk mengolah suatu data dan mengirimkannya kembali ke *web browser* menjadi kode HTML” (Astria Firman, 2019 : 30).

II.2.9. MySQL

MySQL (*My Structure Query Language*) merupakan sebuah program pembuat database yang bersifat *open source*, artinya semua orang dapat

menggunakannya dan dapat dijalankan pada semua *platform* baik *windows* maupun *linux*. *MySQL* juga merupakan sebuah perangkat lunak sistem manajemen basis data *SQL* yang bersifat jaringan sehingga dapat digunakan untuk aplikasi multi *user*. *MySQL* juga sering dikenal dengan nama sistem manajemen *database* relasional. Suatu *database* relasional menyimpan data dalam table yang terpisah. Tabel -table tersebut terhubung oleh suatu relasi terdefinisi yang memungkinkan memperoleh kombinasi data dari beberapa table dalam suatu permintaan. Untuk administrasi *database*, seperti pembuatan *database*, pembuatan tabel, dan sebagainya dapat digunakan aplikasi berbasis web seperti *phpMyAdmin* dengan aplikasi *XAMPP* (Saipul Anwar, 2019).

II.2.10. UML (*Unified Modelling Language*)

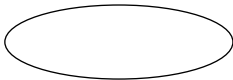
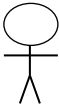


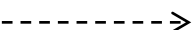
UML merupakan bahasa visual dalam pemodelan yang memungkinkan pengembang sistem membuat sebuah *blueprint* yang dapat menggambarkan visi mereka tentang sebuah sistem dalam format yang standar, mudah dimengerti, dan menyediakan mekanisme untuk mudah dikomunikasikan dengan pihak lain . UML adalah bahasa untuk mengspesifikasi, memvisualisasi, membangun dan mendokumentasikan *artefact* (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya). UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan peranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C#, atau

VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C (I Made Budi Adnyana, 2019).

II.2.10.1. Use Case Diagram

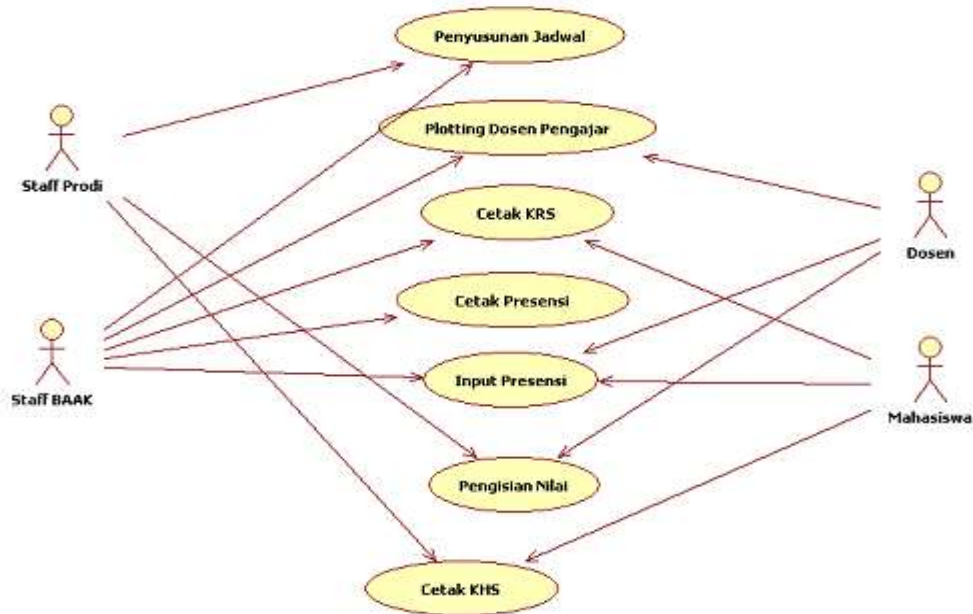
Use Case Diagram adalah gambaran fungsionalitas dari suatu sistem, sehingga *customer* atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun. Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.1 dibawah ini :

Tabel II.1. Simbol Use Case

Gambar	Keterangan	Deskripsi
	<i>Use case</i>	Menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor	Sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem.
	Asosiasi	Penghubung antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi	Penghubung antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i>	Merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i>

		oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
←-----	<i>Extend</i>	Merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Ade Hendini, 2019)



Gambar II.3. Use Case Diagram

(Sumber : I Made Budi Adnyana, 2019)

II.2.10.2. Class Diagram

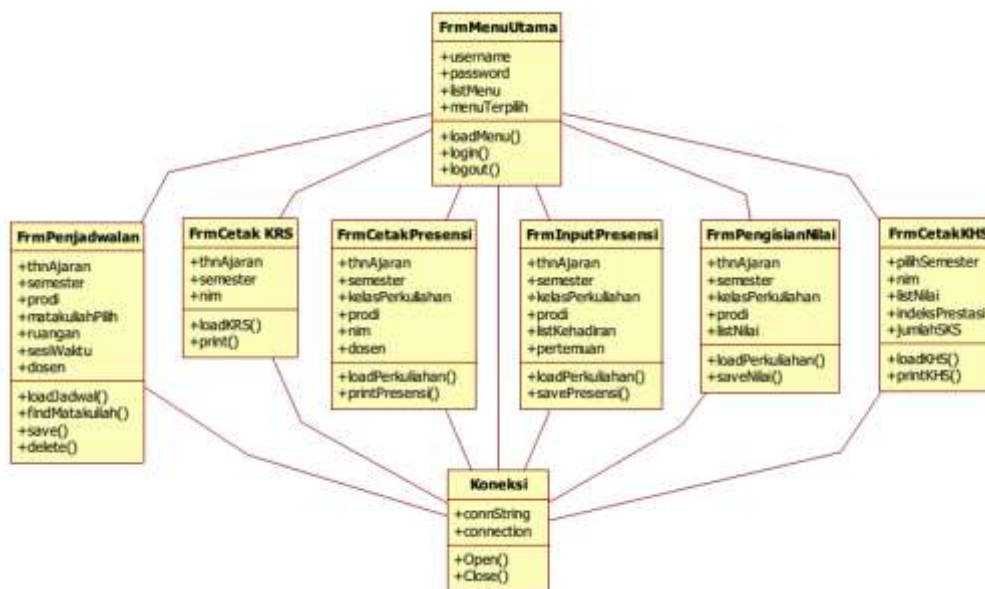
Class Diagram menggambarkan struktur dan deskripsi *class*, *package* dan *object* beserta hubungan satu sama lain seperti pewarisan, asosiasi, dan lain-lain (I Made Budi Adnyana, 2019).

Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality* (Ade Hendini, 2019).

Tabel II.2. *Multiplicity Class Diagram*

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Ade Hendini, 2019)

Gambar II.4. *Class Diagram*

(Sumber : I Made Budi Adnyana, 2019)




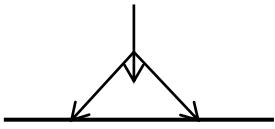
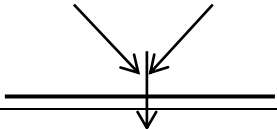
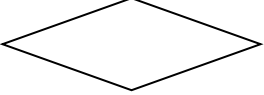

II.2.10.3. Activity Diagram

Activity Diagram menggambarkan sebagai alur aktifitas dalam sistem yang sedang dirancang. Bagaimana masing-masing alur berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. Sebuah aktifitas dapat direalisasikan oleh satu *use case* atau lebih. Aktifitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana *actor* menggunakan sistem untuk melakukan aktivitas. *Decision* digunakan untuk menggambarkan

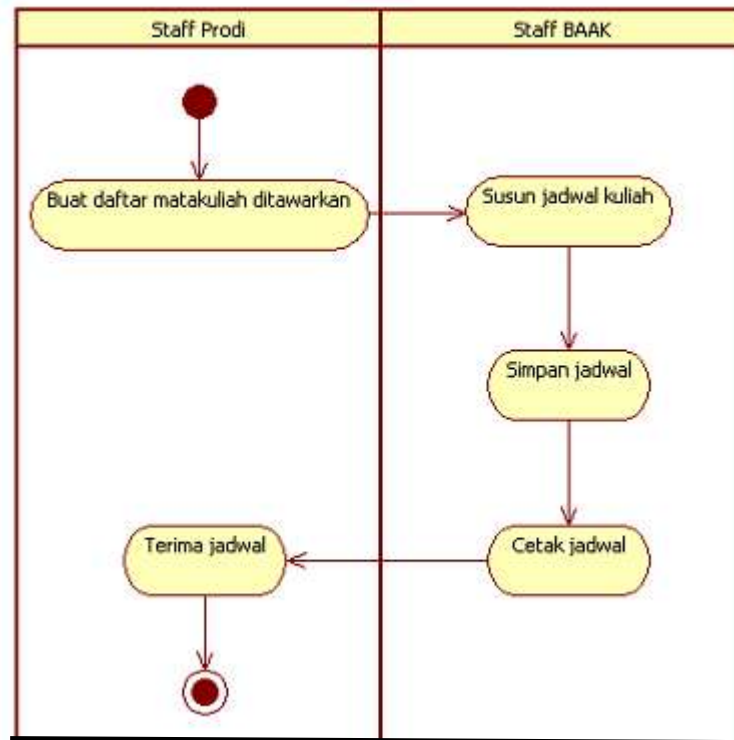
behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses *parallel* (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertical. *Activity* diagram dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktifitas tertentu (I Made Budi Adnyana, 2019).

Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.3 dibawah ini:

Tabel II.3. Simbol Activity Diagram

Gambar	Keterangan	Deskripsi
	<i>Start point</i>	Diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i>	Akhir aktifitas.
	<i>Activites</i>	Menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan).	Digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	<i>Join</i> (penggabungan)	Digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i>	Menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i>	Pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.

(Sumber : Ade Hendini, 2019)



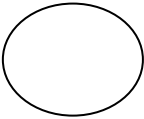
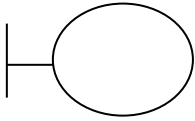
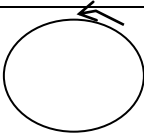

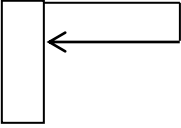


Gambar II.5. Activity Diagram
(Sumber : I Made Budi Adnyana, 2019)

II.2.10.4. Sequence Diagram

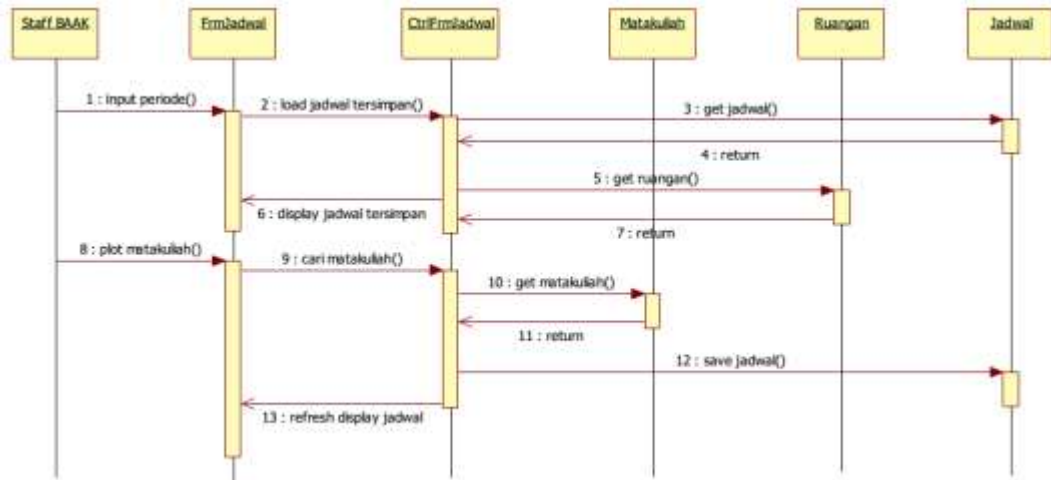
Sequence Diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence Diagram* terdiri atas dimensi vertical (waktu) dan dimensi horizontal (obyek–obyek yang terkait). *Sequence Diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu (I Made Budi Adnyana, 2019).

Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.4 dibawah ini :

Tabel II.4. Simbol *Sequence Diagram*

Gambar	Keterangan	Deskripsi
	<i>Entity Class</i>	Merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i>	Berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<i>Control class</i>	Suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i>	Simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i>	Menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i>	Mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i>	Garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Ade Hendini, 2019)



Gambar II.6. Sequence Diagram

(Sumber : I Made Budi Adnyana, 2019)