

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Penelitian terdahulu yang berhubungan penulisan skripsi dapat dilihat pada berikut ini

1. Berdasarkan penelitian yang dilakukan oleh Abdul Rahman Ariga, Ahmad Supaidi, Izhah Aslamiah, Ali Ibrahim (2018) dengan judul “Implementasi *Customer Relationship Management* (CRM) Pelayanan Pelanggan (*Corporate*) Pada PT. Telkom Witel Sumsel”. Tujuan penelitian yang dilakukan adalah menjaga hubungan baik dengan pelanggan. Karena dari hasil penelitian mempertahankan pelanggan yang sudah ada jauh lebih mudah dibandingkan dengan mendapatkan pelanggan baru. Persaingan bisnis yang sangat ketat akan sangat memerlukan langkah-langkah strategis untuk menghadapinya. Kehadiran *customer relationship management* (CRM) akan mampu mengatasi masalah yang ada salah satu langkah yang juga dapat ditempuh untuk menghadapi masalah tersebut adalah dengan memanfaatkan teknologi yang ada yaitu dengan pemanfaatan *customer relationship management* (CRM) berbasis *web*.
2. Berdasarkan penelitian yang dilakukan oleh Mira Afrina, Ali Ibrahim, (2013), dengan judul “Rancang Bangun *Electronic Customer Relationship Management* (E-CRM) Sebagai Sistem Informasi Dalam Peningkatan Layanan Perpustakaan Digital Fakultas Ilmu Komputer Unsri”. Untuk

menangani kendala yang dihadapi hal ini harus dilakukan untuk memenuhi tuntutan terhadap mutu layanan perpustakaan, *resource sharing*, mengefektifkan SDM, efisiensi waktu dan keragaman informasi yang dikelola. Perpustakaan yang menerapkan konsep E-CRM. *Customer relationship management* (CRM) merupakan strategi untuk memperoleh, mengkonsolidasi, menganalisa data, untuk kemudian digunakan untuk berinteraksi dengan customer (pengguna perpustakaan: siswa, mahasiswa, guru, dosen, dan masyarakat), dengan demikian akan tercipta suatu pandangan yang komprehensif terhadap *customer* dan hubungan yang lebih baik dengan *customer*.

3. Berdasarkan penelitian yang dilakukan oleh Sandy Kosasi (2015), dengan judul “Perancangan Sistem *Electronic Customer Relationship Management* Untuk Mempertahankan Loyalitas Pelanggan”. Tujuan dari penelitian menghasilkan sistem E-CRM yang dapat membantu pihak manajemen distributor sparepart sepeda motor mengetahui dan mengelola semua pengetahuan dan kebutuhan pelanggan secara lebih personal sehingga dapat memberikan pelayanan yang lebih cepat dan berkualitas dalam meningkatkan kepuasan dan upaya mempertahankan loyalitas pelanggan. Hasil penelitian memperlihatkan sistem E-CRM dapat memberikan kemudahan dalam mengelola semua data pelanggannya dan menarik calon pelanggan baru melalui fasilitas yang komunikatif dan interaktif secara *online*.
4. Berdasarkan penelitian yang dilakukan oleh Hendra Darmawan (2018), dengan judul “Analisis dan Perancangan *Electronic Customer Relationship*

Management pada Cetta Mom and Baby Spa Pontianak”. Tujuan dari penelitian ini untuk merancang aplikasi E-CRM pada perusahaan Cetta Mom and Baby Spa Pontianak dengan memberikan layanan informasi secara *online* dalam hal reservasi jasa treatment yang *up-date*. Metode penelitian yang digunakan dalam penelitian ini adalah metode penelitian dan pengembangan atau yang biasa dikenal dengan *research and development*. Hasil penelitian memperlihatkan adanya *website* E-CRM lebih mempermudah konsumen untuk melakukan transaksi dan berkomunikasi dengan pihak perusahaan serta mempertahankan konsumen lamanya dan juga untuk memperoleh konsumen baru dengan memasarkan barang kepada masyarakat luas.

5. Berdasarkan penelitian yang dilakukan oleh Evri Ekadiansyah (2015), dengan judul “Perancangan E-CRM (*Customer Relationship Management*) Pada PT. Sonya FM Medan”. Tujuannya untuk mempererat hubungan antara pihak radio, pihak luar, maupun penggemar radio yang menggunakan sistem informasi berbasis internet yang dapat di akses kapanpun dimana saja berada. Sehingga akan terjadi interaktif antar pengguna *website* dan memberikan efek timbal balik antara pengguna, dengan pihak radio demi meningkatkan perkembangan radio saat ini juga dimasa yang akan datang. Hasil dari perancangan ini menjadi acuan bagi pengembangan untuk masuk ke tahap berikutnya dalam pengembangan aplikasi.

II.2. Landasan Teori

II.2.1. Sistem Informasi

Sistem informasi adalah sekumpulan komponen-komponen sistem yang berada didalam suatu ruang lingkup organisasi, saling berinteraksi untuk menghasilkan sebuah informasi yang bertujuan untuk pihak manajemen tertentu dan untuk mencapai tujuan tertentu. Faktor-faktor yang menentukan kehandalan dari suatu sistem informasi atau informasi dapat dikatakan baik jika memenuhi kriteria-kriteria sebagai berikut :

1. Keunggulan (*Usefulness*) yaitu suatu sistem yang harus dapat menghasilkan informasi yang tepat dan relevan untuk mengambil keputusan manajemen dan personil operasi dalam organisasi.
2. Ekonomis yaitu kemampuan sistem yang mempengaruhi sistem harus bernilai manfaat minimal, sebesar biayanya.
3. Kehandalan (*Reliability*) yaitu keluaran dari sistem harus mempunyai tingkat ketelitian tinggi dan sistem tersebut harus beroperasi secara efektif.
4. Pelayanan (*Customer Service*) yakni suatu sistem memberikan pelayanan yang baik dan efisien kepada para pengguna sistem pada saat berhubungan dengan organisasi.
5. Kapasitas (*Capacity*) yaitu setiap sistem harus mempunyai kapasitas yang memadai untuk menangani setiap periode sesuai yang dibutuhkan.

6. Sederhana dalam kemudahan (*Simplicity*) yaitu sistem tersebut lebih sederhana (umum) sehingga struktur dan operasinya dapat dengan mudah dimengerti dan *prosedure* mudah diikuti.
7. Fleksibel (*Fleksibility*) yaitu sistem informasi ini harus dapat digunakan dalam kondisi yang bagaimana yang diinginkan oleh organisasi tersebut atau pengguna tertentu.
8. Komponen sistem informasi yaitu 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 (*Technologi 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 mengapses data (*storege device*), alat untuk menghasilkan dan mengirimkan keluaran (*output divice*) 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 : 2016)

II.2.2. E-Customer Relationship Management

Electronic Customer Relationahip Management (E-CRM) merupakan sebuah strategi bisnis dimana terjadi sebuah proses yang dibangun untuk mempertahankan hubungan dengan konsumen guna menciptakan suasana sama-

sama untuk baik bagi karyawan maupun bagi konsumen, dengan demikian kesuksesan penjualan pun akan lebih maksimal. (Hendra Darmawan : 2018)

Customer Relationship Management (CRM) adalah merupakan salah satu sarana untuk menjalin hubungan yang berkelanjutan antara perusahaan dengan pelanggan, dengan memanfaatkan CRM perusahaan akan mengetahui apa yang diharapkan dan diperlukan pelanggannya, strategi pengelolaan pelanggan, mulai dari proses pemasaran, penjualan sampai dengan pelayanan setelah penjualan, yang bertujuan untuk meningkatkan kepuasan pelanggan, yang berujung pada loyalitas pelanggan tersebut (Ali, 2018 : 2)

Ada tiga dalam kerangka yang digunakan dalam customer relationship management yaitu (Ali, 2018 : 4) :

- a. Operasional *Customer Relationship Management* merupakan proses otomatisasi yang terintegrasi dari keseluruhan proses bisnis yang berjalan di dalam perusahaan, baik penjualan maupun pemasaran. Operasional *customer relationship management* sering disebut *front office* perusahaan.
- b. *Analytical Customer Relationship Management* adalah proses analisis dari data-data yang dihasilkan melalui proses operational. Pada *fase* ini meliputi data mining, analisis kebutuhan, *trend* pasar dan perilaku pelanggan.
- c. *Collaborative Customer Relationship Management* adalah proses aplikasi kolaboratif pelayanan meliputi email, *personalized*, *e-communities*, forum diskusi dan sejenisnya yang menyediakan fasilitas interaksi antara perusahaan dengan pelanggannya.

Manfaat *Customer Relationship Management* (Ali, 2018 : 5) :

- a. Mendorong loyalitas pelanggan
- b. Mengurangi biaya yang dikeluarkan
- c. Meningkatkan efisiensi operasional
- d. Meningkatkan pendapatan

II.2.3. Normalisasi

Normalisasi merupakan cara pendekatan dalam membangun desain logika basis data relasional yang tidak secara langsung berkaitan dengan model data, tetapi dengan menerapkan sejumlah aturan dan kriteria standar untuk menghasilkan struktur tabel yang normal.

Salah satu topik yang cukup kompleks dalam dunia manajemen *database* adalah proses untuk menormalisasi tabel-tabel dalam *database relasional*. Dengan normalisasi kita ingin mendesain *database relasional* yang terdiri dari tabel-tabel berikut :

1. Berisi data yang diperlukan.
2. Memiliki sesedikit mungkin redundansi.
3. Mengakomodasi banyak nilai untuk tipe data yang diperlukan.
4. Mengefisienkan *update*.
5. Menghindari kemungkinan kehilangan data secara tidak disengaja/tidak diketahui.

Alasan utama dari normalisasi *database* minimal sampai dengan bentuk normal ketiga adalah menghilangkan kemungkinan adanya "*insertion anomalies*",

“*deletion anomalies*”, dan “*update anomalies*”. Tipe-tipe kesalahan tersebut sangat mungkin terjadi pada *database* yang tidak normal. (Kusrini : 2012)

II.2.3.1. Bentuk-bentuk Normalisasi

1. Bentuk tidak normal

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti format tertentu, dapat saja tidak lengkap dan terduplikasi. Data dikumpulkan apa adanya sesuai keadaanya.

Tabel II.1. Tabel Tidak Normal

Kode	Kode Suplier	Nama Suplier	Kode Brg	Nama brg	Tgl
004	G01	Gobel Nustra	A01 A02	AC Model 1 AC Model 2	07/10/04
006	A03	Angkasa	B01 A03	Meja, Kursi	09/10/05

(Sumber : Kusrini : 2012)

2. Bentuk normal tahap pertama (1” Normal Form)

Definisi :

Sebuah table disebut 1NF jika :

- a. Tidak ada baris yang duplikat dalam tabel tersebut.
- b. Masing-masing *cell* bernilai tunggal

Catatan: Permintaan yang menyatakan tidak ada baris yang duplikat dalam sebuah tabel berarti tabel tersebut memiliki sebuah kunci, meskipun kunci tersebut dibuat dari kombinasi lebih dari satu kolom atau bahkan kunci tersebut merupakan kombinasi dari semua kolom.

Tabel II.2. Tabel Normal 1

Kode	Kode Suplier	Nama Suplier	Kode Brg	Nama brg	Tgl
004	G01	Gobel Nustra	A01	AC Model 1	07/10/04
004	G01	Gobel Nustra	A02	AC Model 2	07/10/04
006	A03	Angkasa	B01	Meja	09/10/05
006	A03	Angkasa	A03	Kursi	09/10/05

(Sumber : Kusri : 2012)

3. Bentuk normal tahap kedua (2nd normal form)

Bentuk normal kedua (2NF) terpenuhi jika pada sebuah tabel semua atribut yang tidak termasuk dalam *primary key* memiliki ketergantungan fungsional pada primary key secara utuh.

KdFaktur → Tgl, Jtempo.

KodeSup, NamaSup

KdFaktur, Kodebrg → namaBrg, Qty, Harga

4. Bentuk normal tahap ketiga (3rd normal form)

Sebuah tabel dikatakan memenuhi bentuk normal ketiga (3NF), jika untuk setiap ketergantungan fungsional dengan notasi $X \rightarrow A$, dimana A mewakili semua atribut tunggal di dalam tabel yang tidak ada di dalam X, maka :

a. X haruslah superkey pada tabel tersebut.

b. Atau A merupakan bagian dari primary key pada tabel tersebut.

5. Bentuk Normal Tahap Keempat dan Kelima

Penerapan aturan normalisasi sampai bentuk normal ketiga sudah memadai untuk menghasilkan tabel berkualitas baik. Namun demikian, terdapat pula bentuk normal keempat (4NF) dan kelima (5NF). Bentuk Normal keempat berkaitan dengan sifat ketergantungan banyak nilai (*multivalued dependency*) pada suatu tabel yang merupakan pengembangan dari

ketergantungan fungsional. Adapun bentuk normal tahap kelima merupakan nama lain dari *project join normal form* (PJNF).

6. Boyce Code Normal Form (BCNF)

- a. Memenuhi 1st NF
- b. Relasi harus bergantung fungsi pada atribut *superkey* [14]

KdFaktur → Tgl, Jtempo, KodeSup

KdSup → NamaSup

KdFaktur, Kodebrg → Qty, Harga

KodeBrg → NamaBrg (Kusrini : 2012)

II.2.4. PHP

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah *web* dan bisa digunakan pada dokumen *html*. *PHP* dirancang untuk dapat bekerja sama dengan *database* server dan dibuat sedemikian rupa sehingga pembuatan dokumen *html* yang dapat mengakses *database* menjadi begitu mudah. Tujuan dari bahasa *scripting* ini adalah untuk membuat aplikasi dimana aplikasi tersebut yang dibangun oleh *php* pada umumnya akan memberikan hasil pada *web browser*, tetapi proses secara keseluruhan dijalankan di server. (Saipul Anwar : 2016)

II.2.5. *MySQL*

MySQL adalah sebuah *system* manajemen *database* relasi (*relational database management system*) yang bersifat “terbuka” (*open source*). *MySQL* termasuk *RDMS* (*Relational Database Management System*). Pada *MySQL* sebuah *database* terdiri atas tabel-tabel. Sebuah tabel terdiri atas baris dan kolom.

MySQL dapat didefinisikan sebagai :

- a) *MySQL* merupakan *system* manajemen *database*. *Database* merupakan struktur penyimpanan data. Untuk menambah, mengakses, dan memproses data yang disimpan dalam sebuah *database* komputer, diperlukan sistem manajemen *database* seperti *MySQL Server*.
- b) *MySQL* merupakan sistem manajemen *database* atau basis data terhubung (*relational database manajemen sistem*). *Database* terhubung menyimpan data pada tabel-tabel terpisah. Hal tersebut akan menambah kecepatan dan fleksibilitasnya. Kata *SQL* pada *MySQL* merupakan singkatan dari “*Structured Query Language*”. *SQL* merupakan bahasa standar yang digunakan untuk mengakses *database* dan ditetapkan oleh *ANSI/ISO SQL Standard*.
- c) *MySQL* merupakan *software open source*. *Open source* berarti semua orang diizinkan menggunakan menggunakan dan memodifikasi *software*. Semua orang dapat mendownload *software MySQL* dari internet dan menggunakannya tanpa membayar. Anda dapat mempelajari *source code* dan akan menggunakannya sesuai kebutuhan .

- d) *Server database MySQL* mempunyai kecepatan akses tinggi, mudah digunakan, dan handal. *MySQL* dikembangkan untuk menangani *database* yang besar secara cepat dan telah sukses digunakan selama bertahun-tahun. Konektifitas, kecepatan, dan keamanannya memuat *server MySQL* cocok untuk mengakses *database* di internet.
- e) *MySQL server* bekerja di *client / server* atau *system embedded*. *Software database MySQL* merupakan sistem *client / server* yang terdiri atas *multithread SQL server* yang mendukung *software client* dan *library* yang berbeda, *tool administrative*, dan sejumlah *application programming interface (APIs)*.
- f) *MySQL* tersedia dalam beberapa macam bahasa. (Doro Edi : 2012).

II.2.6. Unified Modeling Language (UML)

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 (Yosua P. W Simaremare, et al., 2013 : 471).

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 lebih cocok untuk penulisan peranti


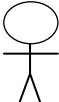


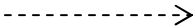
lunak dalam bahasa-bahasa berorientasi objek seperti C++, *Java*, C#, atau *VB.NET*. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C (I Made Budi Adnyana, 2016 : 51-52).

II.2.6.1. 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 (I Made Budi Adnyana, 2016 : 51).

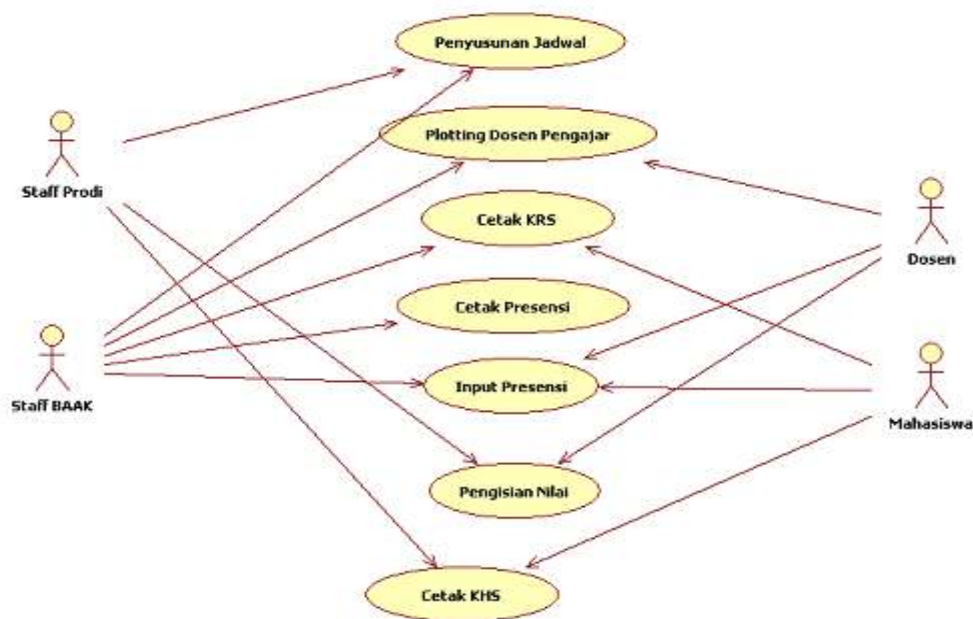
Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.3 dibawah ini :

Tabel II.3. 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 mengindikasikan aliran data.
	Asosiasi	Penghubung antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan 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, 2016)



Gambar II.1. Use Case Diagram

(Sumber : I Made Budi Adnyana, 2016 : 53)

II.2.6.2. Class Diagram

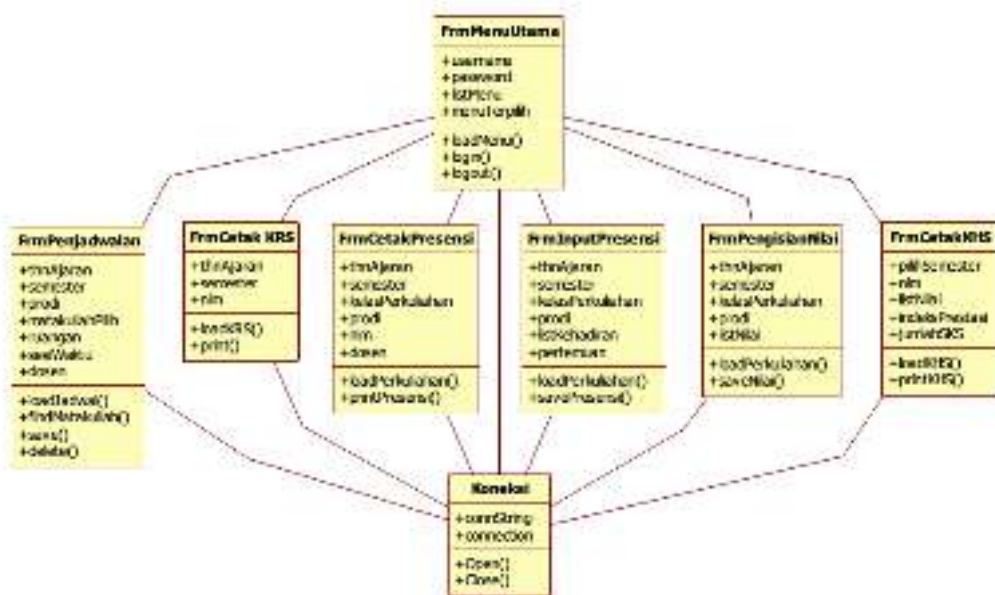
Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan *object* beserta hubungan satu sama lain seperti pewarisan, asosiasi, dan lain-lain (I Made Budi Adnyana, 2016 : 52).

Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau *cardinality* (Ade Hendini, 2016 : 111).

Tabel II.4. *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, 2016 : 110)

Gambar II.2. *Class Diagram*

(Sumber : I Made Budi Adnyana, 2016 : 54)




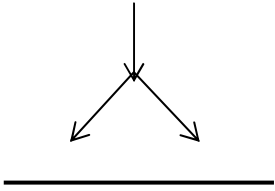
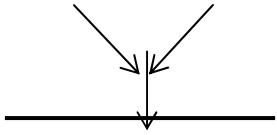
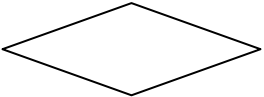

II.2.6.3. *Activity Diagram*

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

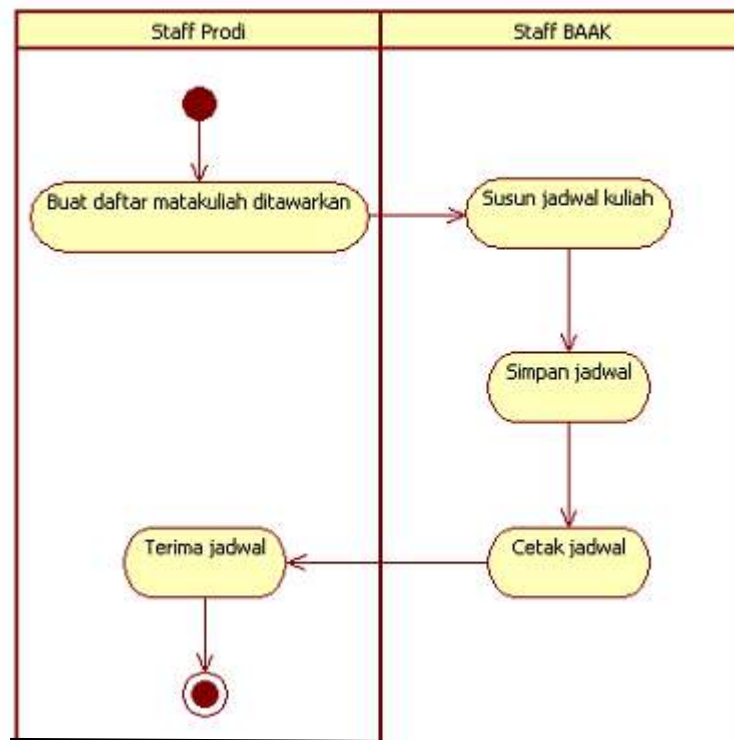
behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses *parallel* (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertical. *Activity* diagram dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktifitas tertentu (I Made Budi Adnyana, 2016 : 52).

Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.5 dibawah ini:

Tabel II.5. 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, 2016 : 109)



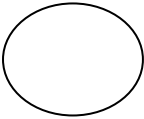
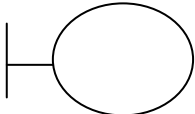
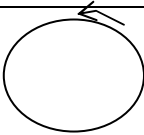

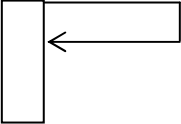


Gambar II.3. Activity Diagram
(Sumber : I Made Budi Adnyana, 2016 : 53)

II.2.6.4. Sequence Diagram

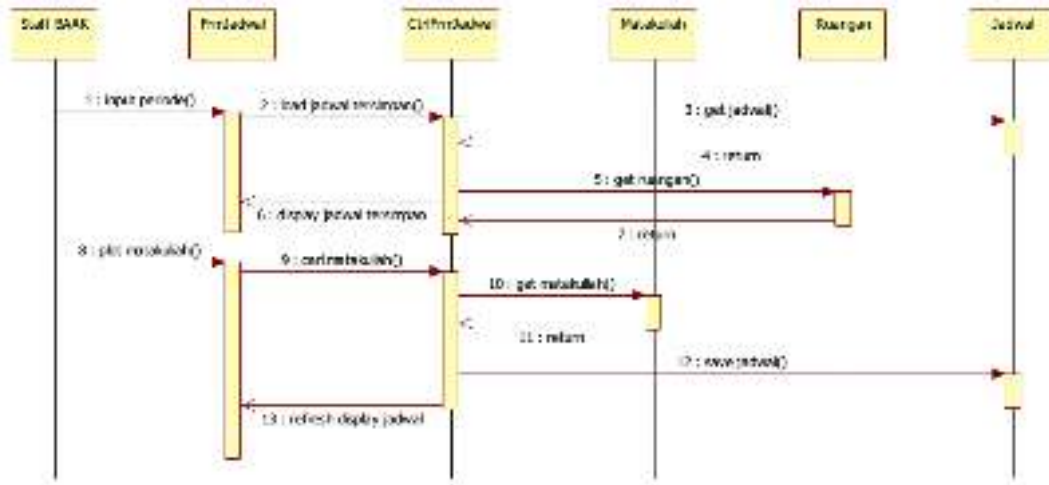
Sequence Diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence Diagram* terdiri atas dimensi *vertical* (waktu) dan dimensi *horizontal* (obyek-obyek yang terkait). *Sequence 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, 2016 : 52).

Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.6 dibawah ini :

Tabel II.6. 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, 2016 : 110)



Gambar II.4. Sequence Diagram

(Sumber : I Made Budi Adnyana, 2016 : 54)