BAB 2
TINJAUAN PUSTAKA

2.1 Tinjauan Perusahaan

Tahap tinjauan perusahaan ini merupakan peninjauan terhadap tempat penelitian studi kasus yang dilakukan di PT. Nusa Pratama Mandiri.

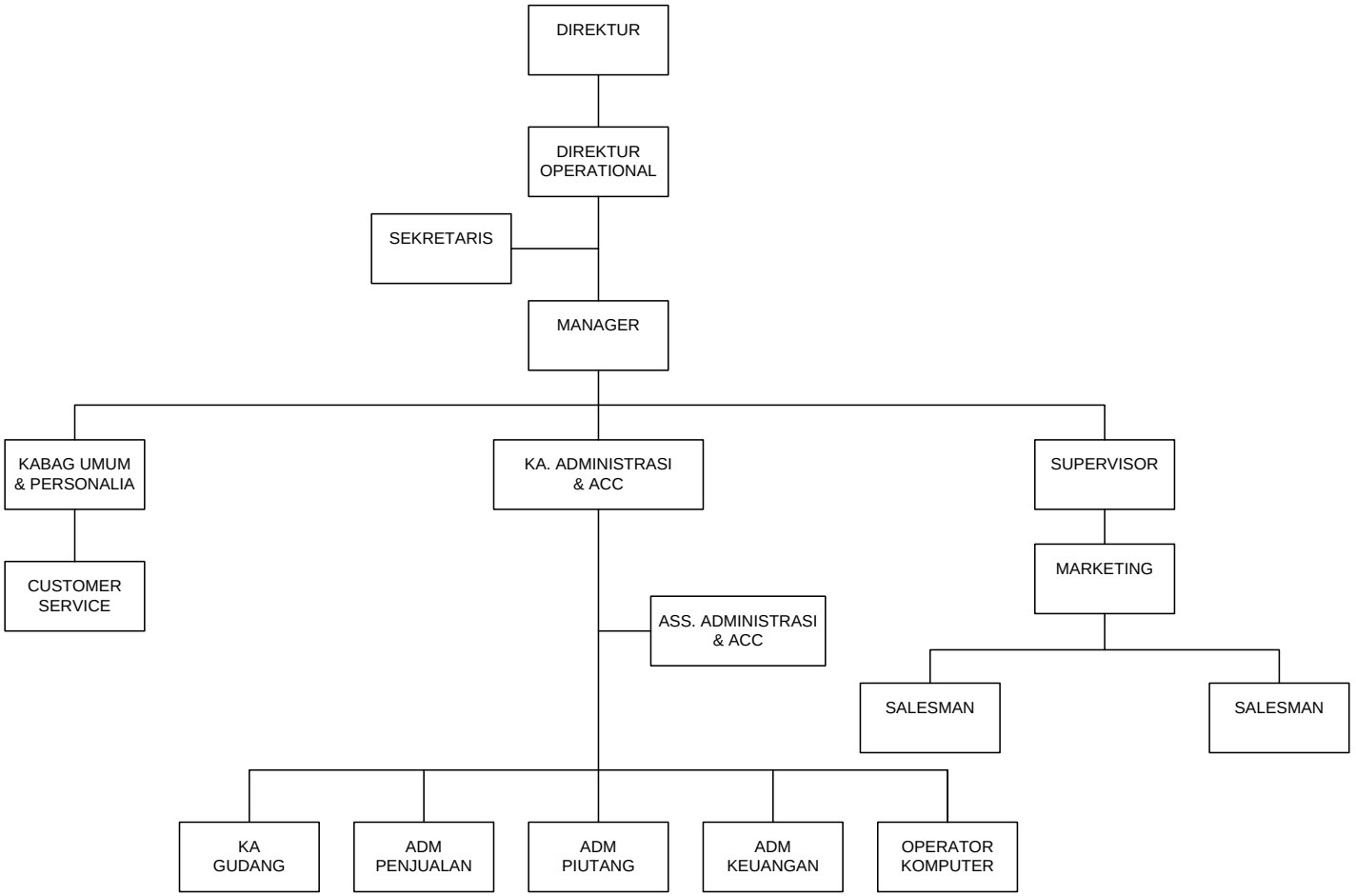
2.1.1 Profil PT. Nusa Pratama Mandiri

Suatu perusahaan yang mengembangkan perusahaannya khususnya dalam bidang penjualan maka diperlukan suatu strategi-strategi yang baik, begitu pula dengan perusahaan ini untuk menembus penawaran yang lebih luas yaitu pemasaran ke daerah-daerah yang menjadi sasarannya. PT. Nusa Pratama Mandiri berlokasi di Komp. Bukit Indah B-7, Jl. Kol. Masturi KM 2,2 Cipageran, Cimahi Utara, Kota Cimahi. Perusahaan ini didirikan pada tahun 2009 dan dipegang oleh Nugroho Tri Widarto sekaligus sebagai pendiri PT. Nusa Pratama Mandiri.

PT. Nusa Pratama Mandiri adalah *water & waste water treatment trading* yang bergerak di bidang perdagangan umum atau perdagangan pengolahan air limbah pabrik. Produk yang dijual oleh PT. Nusa Pratama Mandiri adalah pasir active, pasir silica, *caporit*, *zeolit* (batu kapur), serta *carbon active* atau yang lebih dikenal sebagai media penjernih air dan bahan-bahan kimia. Produk yang dipasarkan didalam dan diluar kota terutama pabrik tekstile dan farmasi. PT. Nusa Pratama Mandiri memiliki visi dan misi sebagai berikut : Visi perusahaan adalah menjadi perusahaan yang global yang berwawasan lingkungan dalam usaha pengelolaan air dan limbah industri. Sedangkan misi perusahaan adalah emenuhi kebutuhan seluruh lapisan masyarakat akan air bersih dan menjadikan lingkungan yang sehat dan bersih dengan menerapkan teknologi pengolahan air yang efisien dan tepat guna.

2.1.2 Struktur Organisasi

Struktur Organisasi merupakan kerangka yang mewujudkan pola tetap dari hubungan-hubungan diantara bidang-bidang kerja, maupun orang-orang yang menunjukkan kedudukan dan peranan masing-masing dalam kebulatan kerjasama. Struktur organisasi PT. Nusa Pratama Mandiri dilakukan secara terarah dan terpimpin untuk mencapai tujuan perusahaan. Oleh karena itu, struktur organisasi PT. Nusa Pratama Mandiri dapat dilihat pada Gambar 2.1.



Gambar 2. 1 Struktur Organisasi PT. Nusa Pratama Mandiri

Adapun tugas dan wewenang dari struktur organisasi PT. Nusa Pratama Mandiri adalah sebagai berikut :

a. Direktur

   Tugas dari direktur PT. Nusa Pratama Mandiri adalah sebagai berikut :

   1. Sebagai pemilik perusahaan sekaligus pemegang saham.
   2. Memantau perkembangan dan kinerja perusahaan.
   3. Menetukan visi dan misi perusahaan untuk mencapai tujuan perusahaan.
   4. Merumuskan dan menetapkan kebijakan umum perusahaan.
   5. Bertanggung jawab terhadap terhadap aktivitas perusahaan secara umum.
   6. Mengawasi dan memonitor kegiatan seluruh bagian secara berkala.
   7. Mengevaluasi hasil kerja yang dicapai dan mengambil tindakan yang diperlukan.
   8. Bertanggung jawab terhadap kelangsungan hidup perusahaan.

b. Direktur Operasional

   Tugas dan tanggung jawab dari direktur operasional PT. Nusa Pratama Mandiri adalah membantu direktur dalam malaksanakan tugas dan bertanggung jawab.

c. Sekretaris

   Tugas dan tanggung jawab dari sekretaris PT. Nusa Pratama Mandiri adalah berrtanggung jawab atas administrasi efisien perusahaan, khususnya berkaitan dengan memastikan kepatuhan terhadap persyaratan hukum dan peraturan, serta memastikan bahwa keputusan direktur dilaksanakan.

d. Manager

Tugas dan tanggung jawab dari manager PT. Nusa Pratama Mandiri adalah sebagai berikut :

1. Managerial cycle atau siklus pengambilan keputusan, membuat rencana, menyusun organisasi, pengarahan organisasi, pengendalian, penilaian dan pelaporan.
2. Memotivasi, artinya seorang manager harus dapat mendorong para bawahannya untuk bekerja giat dan membina para bawahan dengan baik.
3. Berusaha memenuhi kebutuhan para bawahannya.
4. Dapat menciptakan kondisi yang akan membantu bawahannya mendapat kepuasan dalam pekerjaannya.
5. Berusaha agar para bawahannya bersedia memikul tanggung jawab.
6. Membenahi fungsi-fungsi manajemen secara baik.
7. Mewakili dan membina hubungan yang harmonis dengan pihak luar.

e. Kabag Umum dan Personalia

Pada umumnya kegiatan personalia dan umum untuk membantu manager dalam hal ketenagakerjaan. Adapun tugas dan tanggung jawab adalah sebagai berikut :

1. Pembinaan hubungan yang baik antara seluruh karyawan melalui kerjasama sebagai suatu kesatuan.
2. Merekrut pegawai baru, termasuk mewawancarai sebagai saringan awal san syarat pertama seleksi pegawai, mengadakan panggilan dan lain-lain yang berhubungan dengan recruitment.
3. Menegaskan kedisiplinan pegawai dan menilai serta pelanggani teguran serta peringatan maupun pemecatan jika perlu.
4. Mengolah data seluruh pegawai, baik yang ada maupun yang keluar agar ada pengaturan arsip-arsipnya.
5. Mengurus surat menyurat untuk kepentingan umum.
6. Bertanggung jawab penuh masalah ketenagakerjaan.

7. Melaporkan baik lisan maupun tulisan kepada pemimpin mengenai semua jenis kegiatan yang berhubungan dengan personalia.
8. Mewakili pimpinan untuk mengadakan pertemuan-pertemuan maupun aktivitas lainnya jika berhalangan hadir.

f. Customer Service

Tugas dan tanggung jawab customer service PT. Nusa Pratama Mandiri adalah untuk melayani dan mencatat pemesanan dan menerima pengaduan keluhan dari pelanggan.

g. KA. Administrasi dan ACC

Kepala administrasi dan accounting mempunyai tugas dan tanggung jawab mengurus kinerja perusahaan dan bertanggung jawab terhadap berbagai masalah keuangan dan kepengurusan pegawai.

h. Ass. Administrasi dan ACC

Assisten administrasi dan accounting mempunyai tugas dan tanggung jawab membantu kepala administrasi dan accounting untuk mengurus kinerja perusahaan dan bertanggung jawab terhadap berbagai masalah keuangan dan kepengurusan pegawai.

i. KA. Gudang

Kepala gudang memiliki tugas dan tanggung sebagai berikut :

1. Membuat surat jalan atau surat lainnya yang diperlukan untuk proses pengeluaran barang dari gudang.
2. Pelangganikan informasi mengenai persediaan barang yang dibutuhkan.
3. Penyimpanan semua barang yang siap dikirim beserta memelihara dokumen-dokumen yang berkaitan.
4. Melakukan kontrol atas jadwal pengeluaran barang.

j. Administrasi Penjualan

Administrasi penjualan mempunyai tugas dan tanggung jawab merencanakan, mengelola, melaksanakan serta mempersiapkan strategi-strategi yang diperlukan untuk meningkatkan penjualan produk yang dihasilkan serta merencanakan dan melaksanakan kebijakan-kebijakan

untuk kegiatan pelayanan pelanggan dan melakukan pencatatan dan pembukuan semua transaksi penjualan harian disertai bukti penjualan, bisa berupa nota penjualan atau yang lainnya.

k. Administrasi Piutang

Administrasi piutang memiliki tugas dan tanggung jawab sebagai berikut :

1. Melakukan pencatatan administrasi piutang sesuai dengan tanggal transaksi.
2. Memastikan bahwa semua faktur sudah tertagih sesuai dengan jatuh tempo faktur.
3. Melakukan koordinasi dengan pihak sales apabila ada kendala dalam penagihan.

l. Administrasi Keuangan

Administrasi keuangan memiliki tugas dan tanggung jawab sebagai berikut :

1. Mengelola dan mencatat semua penerimaan dan pengeluaran uang perusahaan.
2. Bertanggung jawab langsung kepada kepala administrasi dan accounting sehubung dengan setiap kegiatan finansial perusahaan.
3. Melaporkan serta membuat pembukuan atas semua kegiatan keuangan.

m. Supervisor

Dalam kedudukan sebagai supervisor, mempunyai tugas dan tanggung jawab sebagai berikut :

1. Menyampaikan kebijakan yang disampaikan oleh jabatan diatasnya kepada seluruh bawahannya.
2. Melaksanakan tugas, proyek dan pekerjaan secara langsung.
3. Menegakkan aturan yang telah ditentukan oleh perusahaan dan pelangganikan info pada management mengenai kondisi bawahan, atau menjadi perantara antara pekerja dengan management.
4. Mengontrol dan mengevaluasi kinerja bawahannya.
5. Bertanggung jawab merencanakan, mengordinasikan, memastikan, mengarahkan dan melakukan kegiatan dapat berjalan dengan lancar.

n. Marketing

Marketing mempunyai tugas dan tanggung jawab merencanakan dan melaksanakan konsepsi, penetapan harga, promosi, distribusi atas ide, barang dan jasa. Bagian marketing juga bertanggung jawab dalam proses pengiriman barang.

o. Salesman

Salesman mempunyai tugas dan tanggung jawab mempromosikan dan memperkenalkan produk kepada pelanggan dan calon pelanggan.

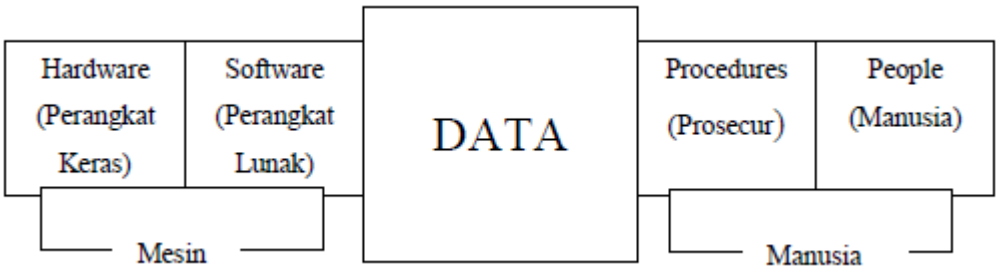
2.2 Landasan Teori

Landasan teori menjelaskan beberapa gambaran sumber dan kajian dari teori-teori yang berkaitan dengan permasalahan yang dibahas sebagai dasar pemahaman dalam sebuah sistem serta metode yang dipakai untuk kegiatan pembangunan sistem, landasan teori yang akan dibahas mengenai sistem informasi, *Customer Relationship Management* (CRM), *Framework of Dynamic* CRM, metode *Fuzzy* RFM dan pemrograman berorientasi objek.

2.2.1 Sistem Informasi

Sistem informasi adalah sekumpulan prosedur organisasi yang pada saat dilaksanakan akan pelangganikan informasi bagi pengambil keputusan dan untuk mengendalikan organisasi. Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang di perlukan. Sistem informasi terdiri dari 5 komponen , kelima komponen tersebut dapat diklasifikasikan sebagai berikut :

a. *Hadware* dan *Software* yang berfungsi sebagai mesin.
b. *People* dan *Procedures* yang merupakan manusia dan tata cara menggunakan mesin.
c. Data merupakan jembatan penghubung antara manusia dan mesin agar terjadi suatu proses pengolahan data.



Gambar 2. 2 Lima Komponen Sistem Informasi

Kegiatan–kegiatan dalam utama yang masuk dalam klasifikasi sistem informasi adalah :

a. Input

   Menggambarkan suatu kegiatan untuk menyediakan data untuk diproses.

b. Proses

   Menggambarkan bagaimana suatu data diproses untuk menghasilkan suatu informasi yang bernilai tambah.

c. Output

   Suatu kegiatan untuk menghasilkan laporan dari proses diatas tersebut.

d. Penyimpanan

   Suatu kegiatan untuk memelihara dan menyimpan data.

e. Control

   Suatu aktivitas untuk menjamin bahwa system informasi tersebut berjalan sesuai dengan yang diharapkan.

Sistem informasi dapat membuat keuntungan kompetitif dalam perusahaan. Meningkatkan keuntungan saat ini, bagaimanapun juga adalah sulit untuk diestimasi. Kenaikan profit sering kali tidak terjadi secara tiba-tiba, tetapi terjadi beberapa tahun sesudah implementasi dari sistem baru. Ketika dua alternatif sistem mendekati kesamaan perbandingan, pintu terbuka untuk mempertimbangkan faktor strategi perusahaan. Terdapat faktor-faktor strategi perusahaan tersebut adalah sebagai berikut.

a. Kepuasan konsumen (*customer satisfacation*).
b. Meningkatkan penjualan (*increase sales*).
c. Komitmen konsumen dan vendor (*customer and vendor commitments*).

d. Informasi produk pemasaran (*information product marketing*).

Sistem Informasi Manajemen (SIM) adalah kumpulan dari sistem manajemen atau sistem yang menyediakan informasi yang bertujuan mendukung operasi manajement dan pengambilan keputusan dalam suatu organisasi yang cenderung berhubungan dengan pengolahan informasi yang berbasis pada komponen dengan mempertimbangkan informasi apa, untuk siapa dan kapan harus disajikan.

2.2.2 *Customer Relationship Management*

Customer Relationship Management (CRM) didefinisikan sebagai *core business strategy* yang mengintegrasikan proses internal serta fungsi-fungsi yang ada dalam perusahaan dengan jejaring eksternal untuk menciptakan dan men-*deliver value* (manfaat) kepada target pelanggan yang memberikan keuntungan.

Berdasarkan definisi CRM disimpulkan CRM adalah strategi bisnis yang terdiri dari *software* dan layanan yang didesain untuk meningkatkan keuntungan (*profit*), pendapatan (*revenue*) dan kepuasan pelanggan (*customer satisfaction*). Caranya adalah dengan membantu berbagai bentuk perusahaan untuk mengidentifikasi pelanggannya dengan tepat, memperoleh lebih banyak pelanggan dengan lebih cepat, dan mempertahankan kesetiaan pelanggannya.

Konsep CRM menekankan pada pendekatan untuk membangun *portfolio* melalui jalinan bisnis yang terbentuk dengan pelanggan melalui interaksi, memberdayakan hubungan yang terbentuk, pembentukan jaringan, dan meningkatkan komunikasi dalam rangka menumbuhkan loyalitas pelanggan. CRM mengedepankan kolaborasi antara perusahaan dengan pelanggan, dimana system ini tidak dimiliki oleh system pemasaran secara umum yang menekankan pada penggunaan media periklanan.

2.2.2.1 Kebutuhan *Customer Relationship Management*

Kebutuhan CRM menegaskan bahwa CRM memungkinkan perusahaan mengidentifikasi dan membidik pelanggan terbaiknya, yaitu mereka yang menguntungkan perusahaan, sehingga mereka dapat dipertahankan menjadi

pelanggan yang setia untuk jangka panjang. Mereka diharapkan akan memberikan keuntungan yang lebih besar bagi perusahaan.

CRM memungkinkan realtime customization and personalization atas produk dan jasa berdasarkan keinginan, kebutuhan, kebiasaan pembelian dan siklus kehidupan pelanggannya.

CRM juga memantau data pelanggannya setiap kali dia melakukan kontak dengan perusahaan. CRM memungkinkan perusahaan untuk menyediakan layanan dan dukungan yang konsisten dan prima terhadap semua kontak yang diinginkan pelanggannya sehingga pelanggan memperoleh pengalaman yang menyenangkan. Semua keuntungan ini memberikan nilai bisnis yang strategis bagi perusahaan sekaligus perceived value bagi pelanggannya.

Sebuah sistem CRM harus bisa menjalankan fungsi:

1. Mengidentifikasi faktor-faktor yang penting bagi pelanggan.
2. Mengusung falsafah customer-oriented (*customer centric*).
3. Mengadopsi pengukuran berdasarkan sudut pandang pelanggan.
4. Membangun proses ujung ke ujung dalam melayani pelanggan.
5. Menyediakan dukungan pelanggan yang sempurna.
6. Menangani keluhan/komplain pelanggan.
7. Mencatat dan mengikuti semua aspek dalam penjualan.
8. Membuat informasi holistik tentang informasi layanan dan penjualan dari pelanggan.

2.2.2.2 Jenis *Customer Relationship Management*

Cunstomer Relationship Management terbagi menjadi 3 jenis CRM, yang dapat dilihat pada Tabel 2.1.

Tabel 2. 1 Jenis *Customer Relationship Management* (Francis Buttle (2007:4))

Jenis CRM	Ciri yang Dominan
Operational	Pandangan tentang CRM yang berfokus pada proyek-proyek otomatisasi seperti otomatisasi layanan, otomatisasi armada penjualan, dan otomatisasi pemasaran.
Analytic	Pandangan "*buttom-up*" tentang CRM yang terfokus pada

	kegiatan penggalian data konsumen untuk tujuan-tujuan strategis dan taktis.
Collaborate	Pandangan tentang CRM yang terfokus mengoptimalkan perusahaan, mitra dan nilai pelanggan

Untuk lebih jelasnya, jenis CRM dapat dijelaskan sebagai berikut :

1. *Operasional* CRM

   CRM operasional lebih terfokus pada otomatisasi cara-cara perusahaan dalam berhubungan dengan para konsumen. Berbagai aplikasi perangkat lunak CRM memungkinkan fungsi-fungsi pemasaran, penjualan dan pelayanan dapat berjalan secara otomatis.

   1. Otomatisasi pemasaran
      a. Segmentasi pasar
      b. Manajemen kampanye komunitas
      c. *Event-based marketing*
   2. Otomatisasi armada penjualan
      a. *Opportunity management* termasuk *lead management*
      b. Pembuatan proposal
      c. Konfigurasi produk
   3. Otomatisasi layanan
      a. Operasi *contact-center* dan *call-center*
      b. Layanan berbasis website
      c. Layanan dilapangan

2. *Analytical* CRM

   CRM analitis di/gunakan untuk mengeksploitasi data konsumen demi meningkatkan nilai perusahaan.

   Sementara menurut Sheth, Parvatiyar dan Shainesh (2002) dimensi CRM dapat dibagi menjadi tiga yaitu :

   1. *Continuity Marketing* adalaha pemberian pelayanan yang berkelanjutan untuk mengikat konsumen.
   2. *One to One Marketing* adalah pendekatan secara individual berdasarkan pemenuhan kebutuhan dan keinginan konsumen.

3. *Partnering Program* adalah hubungan kerjasama perusahaan dengan perusahaan penyedia produk atau jasa lain dalam mengikat konsumen baik yang berada diluar maupun didalam perusahaan.

3. *Collaborate* CRM

   CRM kolaborasi adalah istilah yang digunakan untuk menggambarkan strategi dan taktik keselarasan perusahaan, biasanya terpisah dalam rantai pasokan untuk tabel profi lebih teridentifikasi, daya tarik, retensi dan pengembangan pelanggan. Sebagai contoh produsen, barang-barang konsumen dan pengecer dapat menyelaraskan proses teknologi untuk melayani pembeli lebih efisien dan efektif. Mereka menggunakan praktek-praktek seperti *co-marketing*, kategori manajemen, kolaboratif peramalan, baru bersama pengembangan produk dan riset pasar bersama. CRM kolaborasi menggunakan teknologi CRM untuk berkomunikasi dan bertransaksi dibatas organisasi. Meskipun teknologi tradisional seperti surat pos biasa, telepon dan faks memungkinkan hal ini terjadi, istilah biasanya diterapkan dengan teknologi yang lebih terbaru seperti pertukaran data elektronik (*EDI*), portal, *e-bisnis*, *voice over internet protocol* (*VoIP*), konferensi, *chat room*, forum web dan *email*. CRM kolaborasi memungkinkan organisasi terpisah untuk menyelaraskan upaya mereka untuk layanan pelanggan yang lebih efektif. Hal ini memungkinkan berharganya informasi untuk dibagikan sepanjang *supply chain* (rantai pasok).

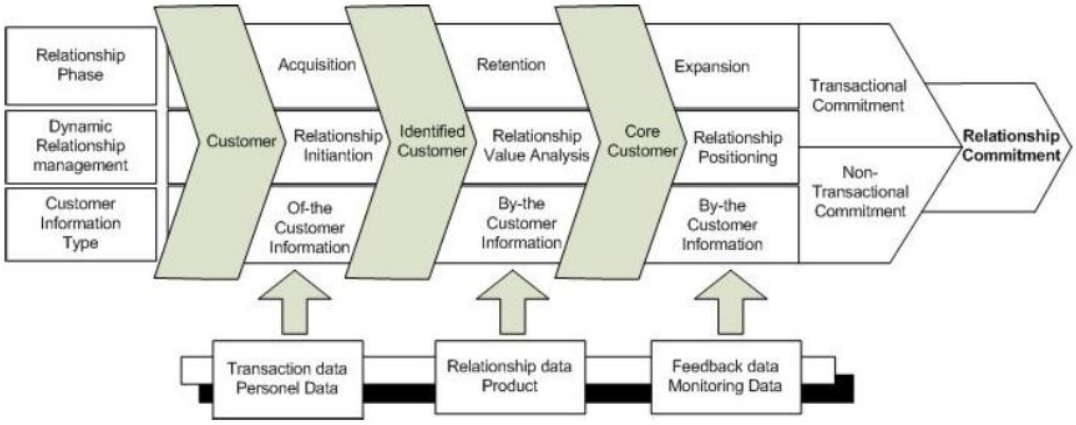
2.2.3 *Framework of Dynamic CRM*

Ada banyak rujukan yang mengartikan CRM dengan berbeda-beda, dan diantaranya yang sesuai dengan penulisan ini ialah :

"*CRM is business strategy to select and manage customers to optimize long-term value. CRM requires customer-centric business philosophy and culture to support marketing, sales and services process. CRM applications can enable effective Customer Relationship Management, Provided that enterprise has the right leadership, strategy and culture* (greaterchinacrm.org 2003)."

Secara sederhanan *Customer Relationship Management* (CRM) merupakan perpaduan antara 3 aspek yang saling berkaitan. Jika membentuk suatu segitiga dengan simpul yang disebut *Customer - Relation – Management*. Adapun masing masing aspek dapat dijelaskan sebagai berikut:

1. ***Customer***, dapat diartikan sebagai pengguna layanan barang dan jasa dari suatu organisasi bisnis yang menghasilkan profit (manfaat) bagi organisasi tersebut.
2. ***Relationship***, merupakan hubungan antara *customer* dengan organisasi bisnis yang dapat berupa komunikasi maupun interaksi dan dapat terjadi dalam jangka pendek, jangka panjang, secara terus-menerus, atau hanya sekali. Biasanya customer yang melakukan secara berulang bila telah memiliki rasa percaya/puas terhadap bentuk pelayanan suatu organisasi.
3. ***Management***, semua jenis informasi / data yang diperoleh dari *customer* menyebabkan organisasi bisnis dapat memetakan bagaimana perilaku dan kebiasaan dari pelanggan, sehingga dapat digunakan sebagai nilai tambah bagi perusahaan, untuk meningkatkan pelayanan serta bersaing dengan kompetitor.



Gambar 2. 3 *Framework of Dynamic* CRM

Berikut Pendekatan Dalam Menetapkan Fitur CRM :

Siklus CRM yang terdiri dari proses *Acquire–Retain–Expansion*, masing-masing fase dalam siklus tersebut dapat dijelaskan sebagai berikut:

1. *Acquire*, proses *acquiring a new relationship* adalah untuk mendapatkan pelanggan baru yang dapat diwujudkan dalam berbagai strategi. Kata kunci

yang dikaitkan dengan fase ini adalah diferensiasi, inovasi, dan kenyamanan (*convenience*).

2. *Retain, retaining customer relationship* adalah fase dimana organisasi melakukan strategi untuk mempertahankan pelanggan dengan cara peningkatan pelayanan. *Customer retaintion* menjadi strategi yang penting bagi perusahaan karena pilihan bagi pelanggan semakin banyak yang di ciptakan oleh banyak organisasi penyedia layanan (*competitor*). Kata kunci yang diasosiasikan dengan fase ini adalah *adaptabiliy*, *listening*, dan *Responship*.
3. *Expansion*, adalah masa untuk membina hubungan baik yang telah ada dengan pelanggan, dengan selalu mendengar keinginan pelanggan dan melayaninya dengan baik, terciptanya pelanggan yang loyal terhadap produk/layanan organisasi. Beberapa kata kunci yang di asosiasikan dengan fase ini adalah *loyality, reduce cost*, dan *customer service*.

Dari beberapa definisi tentang CRM dapat disimpulkan bahwa *Customer Relationship Management* (CRM) terdiri beberapa pihak yang terlibat dalam penerapannya secara langsung mulai dari *customer* hingga organisasi penyedia layanan (*service*) dan menjadi sebuah siklus yang dapat di management sesuai dengan proses bisnis organisasi tersebut. Untuk menetapkan fitur CRM secara tepat, sebagai landasan untuk menyusun *Software* Requirement *Specification* (SRS), diperlukan *framework* sebagai acuan. CH Park & YG Kim mengusulkan sebuah CRM *Framework* yang dinamakan "*A framework of Dynamic CRM*".

Framework ini menjelaskan serangkaian tahapan pada pembangunan/penerapan CRM. Substantif terpentingnya adalah informasi yang didapat dari *customer* sehingga diperoleh outputnya yang berupa *Relationship Commitment*, model tersebut dapat dilihat pada Informasi yang menjadi fokus perhatian adalah sebagai berikut:

a) Informasi mengenai pelanggan
b) Informasi untuk pelanggan
c) Informasi oleh pelanggan

Didasari dari ketiganya di atas jelas informasi amat penting untuk mewujudkan *Customer Relation Management* (CRM) Organisasi untuk melakukan persaingan, dengan melakukan pengolahan informasi tersebut akan didapat *customer behavior*, karena *customer behavior* pada saat ini lebih cenderung menginkan bagaimana mereka merasa terperhatikan, dilayani serta ketanggapan dari organisasi sebagai penyedia layanan kepada mereka dan biasanya kecenderungan tersebut tidak melihat jumlah harga yang mereka keluarkan lagi. Dengan semakin meningkatnya pelayanan terhadap *customer* suatu organisasi seharusnya harus sudah dapat memisahkan seperti yang penulis bahas di awal yang mana fasilitas dan yang mana suatu *service.*

2.2.4 Kalsifikasi Fuzzy RFM

Klasifikasi *fuzzy* RFM adalah proses menganalisis prilaku pelanggan. Hal ini umumnya digunakan dalam pemasaran database dan pemasaran langsung. RFM singkatan dari *Recency, Frequency, Monetary*. Menggunakan informasi tentang perilaku pelanggan pada masa lalu yang mudah dilacak dan tersedia. Tujuan dari RFM adalah untuk meramalkan prilaku konsumen di masa yang depan (mengarahkan keputusan segmentasi yang lebih baik). Oleh karena itu, perlu menterjemahkan prilaku konsumen dalam ‘angka’ sehingga dapat digunakan sepanjang waktu.

Analisa RFM terdiri dari tiga dimensi, yaitu:

1. *Recency*, yaitu rentang waktu (dalam satuan hari, bulan, tahun) dari transaksi terakhir yang dilakukan oleh konsumen sampai saat ini.
2. *Frequency,* yaitu jumlah total transaksi atau jumlah rata-rata transaksi dalam satu periode.
3. *Monetary*, yaitu jumlah rata-rata nilai pembelian konsumen dalam suatu satuan waktu.

Metode yang digunakan dalam segmentasi pelanggan ini adalah metode RFM. Metode RFM mendeskripsikan atribut *recency, frequency,* dan *monetary.* Konteks dari masing-masing atribut didefinisikan sebagai berikut :

1. Domain atribut *recency* didefinisikan dengan rumus:

$$\boldsymbol{K = \frac{1}{R}\, x\, 10}$$

Ket :

K = Skor Recency

R = Recency

Contoh :

Jika ada konsumen yang memiliki transaksi terakhir 4 hari yang lalu, maka skor *recency* nya adalah $\boldsymbol{R = \frac{1}{4}\, x\, 10 = 2,5}$, maka skor *recency* nya 2,5 .

2. Domain atribut *frequency* didefinisikan dengan rata-rata jumlah transaksi konsumen dalam waktu 1 tahun terakhir.
3. Domain atribut *monetary* didefinisikan dengan rata-rata nilai pembelian konsumen dalam 1 tahun terakhir, kemudian dibagi 10000.

$$skor\ monetary = \frac{nilai\ monetary}{10000}$$

Contoh :

Jika seorang konsumen memiliki nilai *monetary* 220.000, maka skor *monetary* nya adalah $skor\ monetary = \frac{220000}{10000} = 22\ skornya.$

Pengelompokkan konsumen berdasarkan hasil skor RFM menurut kategori *recency*, *frequency*, dan *monetary*. Terdapat 4 kelompok konsumen dimana pembagiannya berdasarkan total hasil skor RFM yang diperoleh. Pengelompokkan dapat dilihat pada tabel 2.2

Tabel 2. 2 Kelompok Pelanggan

No.	Kategori Pelanggan	Skor
1	*Most valuable customer*	Skor > 200
2	*Most growable customer*	200 > skor < 100
3	*Migrator*	100 > skor < 50
4	*Below zeros*	Skor < 50

2.2.5 Pemrograman Berorientasi Objek

Metodologi berorientasi objek adalah suatu strategi pembangunan peralngkan lunak yang mengorganisasikan perangkat lunak sebagai kumpulan

objek yang berisi data dan operasi yang diberlakukan terhadapnya . Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Metode berorientas objek didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas.

Metode berorientasi objek meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek, dan pengujian berorientasi objek.

Pada saat ini, metode berorientasi objek banyak dipilih karena metodologi lama banyak menimbulkan masalah seperti adanya kesulitan pada saat mentransformasi hasil dari satu tahap pengembangan ke tahap berikutnya, misalnya pada metode pendekata terstruktur, jenis aplikasi yang dikembangkan saat ini berbeda dengan masa lalu. Aplikasi yang dikembangkan pada saat ini beragam dengan *platform* yang berbeda-beda, sehingga menimbulkan tuntutan kebutuhan metodologi pengembangan yang dapat mengakomodasi ke semua jenis aplikasi tersebut.

Keuntungan dalam menggunakan metodologi berorientasi objek adalah sebagai berikut :

1. Meningkatkan produktivitas

   Karena kelas dan objek yang ditemukan dalam suatu masalah masih dapat dipakai ulang untuk masalah lainnya yang melibatkan objek tersebut (*reusable*).

2. Kecepatan pengembangan

   Karena sistem yang dibangun dengan baik dan benar pada saat analisis dan perancangan akan menyebabkan berkurangnya kesalahan pada saat pengodean.

3. Kemudahan pemeliharaan

   Karena dengan model objek, pola-pola yang cenderung tetap dan stabil dapat dipisahkan dan pola-pola yang mungkin sering berubah-ubah.

4. Adanya konsistensi

   Karena sifat pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan maupun pengkodean.

5. Meningkatkan kualitas perangkat lunak

   Karena pendekatan pengembangan lebih dekat dengan dunai nyata dan adanya konsistensi pada saat pengembangannya, perangkat lunak yang dihasilkan akan mampu memenuhi kebutuhan pemakai serta mempunyai sedikit kesalahan.

2.2.5.1 Konsep Dasar Berorientasi Objek

Pendekatan berorientasi objek merupakan suatu teknik atau cara pendekatan dalam melihat permasalahan dan sistem (sistem perangkat lunak, sistem informasi, atau sistem lainnya). Pendekatan berorientasi objek akan memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata.

Ada banyak cara untuk mengabstraksikan dan memodelkan objek-objek tersebut, mulai dan abstraksi objek, kelas, hubungan antar kelas sampai abstraksi sistem. Saat mengabstraksikan dan memodelkan objek, data dan proses-proses yang dipunyai oleh objek akan dienkapsulasi (dihubungkus) menjadi suatu kesatuan.

Dalam rekayasa perangkat lunak, konsep pendekatan berorientasi objek dapat diterapkan pada tahap analisis, perancanggan, pemrograman, dan pengujian perangkat lunak. Ada berbagai teknik yang dapat digunakan pada masing-masing tahap tersebut, dengan aturan dan alat bantu pemodelan tertentu.

Sistem berorientasi objek merupakan sebuah sistem yang dibangun dengan berdasarkan metode berorientasi objek adalah sebuah sistem yang komponennya dibungkus (dienkapsulasi) menjadi kelompok daata dan fungsi. Setiap komponen dalam sistem tersebut dapat mewarisi atribut dan sifat dan komponen lainnya, dan dapat berinteraksi satu sama lain.

Berikut ini adalah beberapa konsep dasar yang harus dipahami tentang metodologi berorientasi objek:

1. Kelas (*Class*)

   Kelas adalah sekumpulan objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statis dan himpunan objek yang sama yang mungkin lahir

atau diciptakan dan kelas tersebut. Sebuah kelas akan mempunyai sifat (atribut), kelakuan (metode/operasi), hubungan (*relationship*) dan arti. Suatu kelas dapat diturunkan dan kelas yang lain, dimana atribut dan kelas semula dapat diwariskan ke kelas yang baru. Secara tekniks kelas adalah sebuah struktur dalam pembuatan perangkat lunak. Kelas merupakan bentuk struktur pada kode program yang menggunakan metodologi berorientasi objek.

2. Objek (*object*)

   Objek adalah abstraksi dari sesuatu yang mewakili dunia nyata benda, manusia, satua organisasi, tempat, kejadian, struktur, status, atau hal-hal lain yang bersifat abstra. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya. Objek mempunyai siklus hidup yaitu diciptakan, dimanipulasi, dan dihancurkan.

   Secara teknis, sebuah kelas saat program dieksekusi makan akan dibuat sebuah objek. Objek dilihat darisegi teknis adalah elemen pada saat *runtime* yang akan diciptakan, dimanipulasi, dan dihancurkan saat eksekusi sehinga sebuah objek hanya ada saat sebuah program dieksekusi. Jika masih dalam bentuk kode, disebut sebagai kelas jadi pada saat *runtime* (saat sebuah program dieksekusi), yang kita punya adalah objek, di dalam teks program yang kita lihat hanyalah kelas.

3. Metode (*method*)

   Operasi atau metode pada sebuah kelas hampir sama dengan fungsi atau prosedur pada metodologi struktural. Sebuah kelas boleh memiliki lebih dari satu metode atau operasi. Metode atau operasi yang berfungsi untuk memanipulasi objek itu sendiri. Operasi atau metode merupakan fungsi atau transformasi yang dapat dilakukan terhadap objek atau dilakukan oleh objek.

   Metode atau operasi dapat berasal dari *event*, aktifitas atau aksi keadaan, fungsi, atau kelakuan dunnia nyata. Contoh metode atau operasi misalnya *Read, Write, Move, Copy,* dan sebagainya.

4. Atribut (*attribute*)

   Atribut dari sebuah kelas adalah variabel global yang dimiliki sebuah kelas. Atribut dapat beruoa nilai atau elemen-elemen data yang dimiliki oleh objek dalam kelas objek. Atribut dipunyai secara individual oleh sebuah objek, misalnya berat, jenis, nama, dan sebagainya.

5. Abstraksi (*abstraction*)

   Prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

6. Enkapsulasi (*encapsulation*)

   Pembungkusa atribut data dan layanan (operasi-operasi) yang dipunyai objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerja-nya.

7. Pewarisan (*inheritance*)

   Mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dan dirinya.

8. Antarmuka (*interface*)

   Antarmuka sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa isi. Deklarasi metode pada sebuah *interface* dapat diimplementasikan oleh kelas lain.

9. *Reusability*

   Pemanfaatan kembali objek yang sudah didefinisikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek tersebut.

10. Generalisasi dan Spesialisasi

    Menunjukkan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus. Misalnya kelas yang lebih umum (generalisasi) adalah kendaraan darat dan kelas khususnya (spesialisasi) adalah mobil, motor, dan kereta.

11. Komunikasi Antarobjek

    Komunikasi antarobjek dilakukan lewat pesan (*message*) yang dikirim dari satu objek ke objek lainnya.

12. Polimorfisme (*polymorphism*)

    Kemampuan suatu objek digunakan di banyak tujuan yang berbeda dengan nama yang sehingga menghemat baris program.

13. *Package*

    Package adalah sebuah kontainer atau kemasan yang dapat digunakan untuk mengelompokkan kelas-kelas sehingga memungkinkan beberapa kelas yang bernama sama disimpan dalam *package* yang berbeda.

2.2.5.2 Unified Model Language (UML)

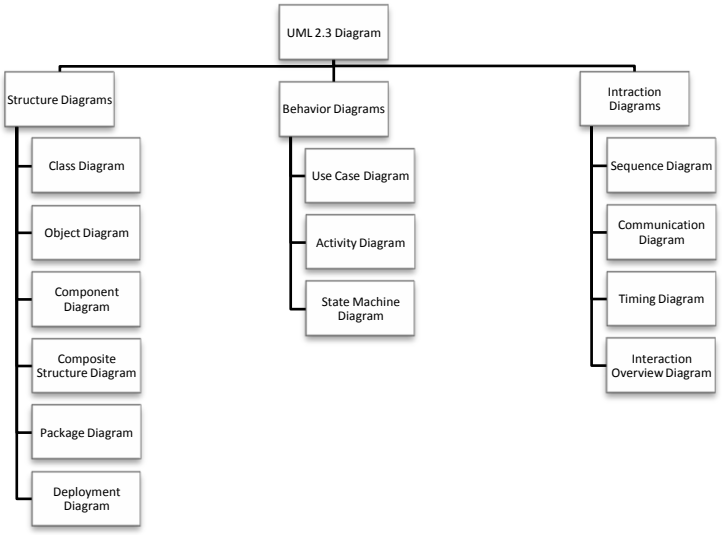
Pada perkembangan teknologi perangkat lunak, diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak. Seperti yang kita ketahui bahwa menyatukan banyak kepala untuk menceritakan sebuah ide dengan tujuan untuk memahami hal yang tidaklah mudah, oleh karena itu diperlukan sebuah bahasa pemodelan perangkat lunak yang dapat dimengerti oleh banyak orang. Banyak orang yang telah membuat bahasa pemodelan pembangunan perangkat lunak sesuai dengan teknologi pemrograman yang berkembang pada saat itu, misalnya yang sempat berkembang dan digunakan banyak pihak adalah *Data Flow Diagram* (DFD) untuk memodelkan perangkat lunak yang menggunakan pemrograman prosedural atau struktural, kemudian juga ada *State Transition Diagram* (STD) yang digunakan untuk memodelkan sistem *real time* (waktu nyata).

Pada perkembangan teknil pemrograman berorientasi objek, munculah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML). UML muncul karena adanya kebutuhan pemodelan voisual untu menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan digaram dan teks-teks pendukung.UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu,

meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek. Seperti yang kita ketahui di dunia sistem informasi yang tidak dapat dibakukan, semua tergantung kebutuhan, lingkunan dan konteksnya. Begitu juga dengan perkembangan penggunaan UML bergantung pada level abstraksi penggunaannya. Jadi, belum tentu pandangan yang perbeda dalam penggunaan UML adalah suatu yang salah, tapi perlu ditelaah dimanakah UML digunakan dan hal apa yang ingin digambarkan. Secara analogi jika dengan bahasa yang digunakan sehari-hari, belum tentu penyampaian bahasa dengan puisi adalahhal yang salah. Sistem informasi bukanlah ilmu pasti, maka jika ada banyak perbedaan dan interpretasi di dalam bidang sistem informasi merupakan hal yang sangat wajar.

2.2.5.2 Diagram UML

Pada UML terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada Gambar 2.5.



Gambar 2. 4 Contoh Diagram UML

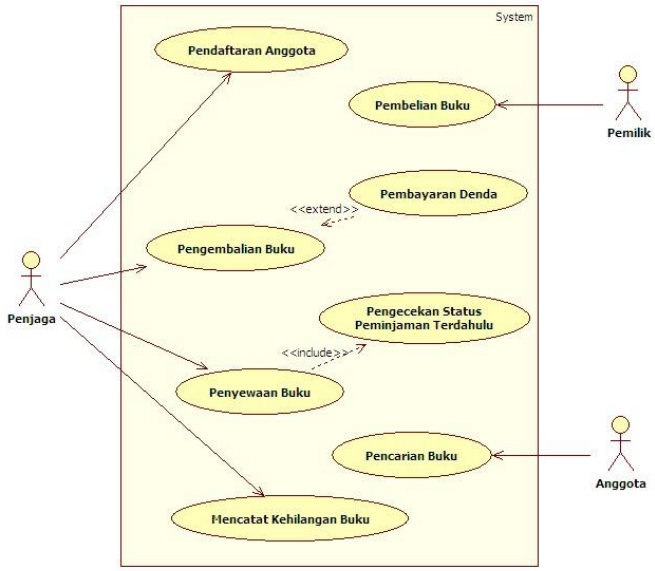
Berikut ini penjelasan singkat dari pembagian kategori tersebut :

1. *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antarsubsistem pada suatu system.

2.2.5.2.2 Use Case Diagram

Use Case atau diagram *Use Case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Syarat penamaan pada *Use Case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *Use Case* yaitu pendefinisian apa yang disebut aktor dan *Use Case.*



Gambar 2. 5 Contoh Use Case Diagram

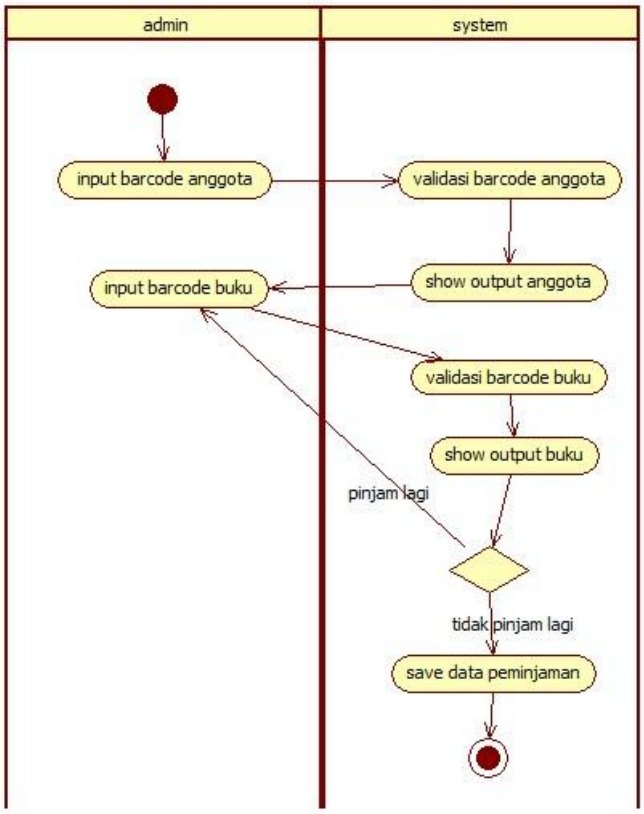
2.2.5.2.3 Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut :

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
2. Urutan atau pengelompokan tampilan dari sistem/*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan

Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.



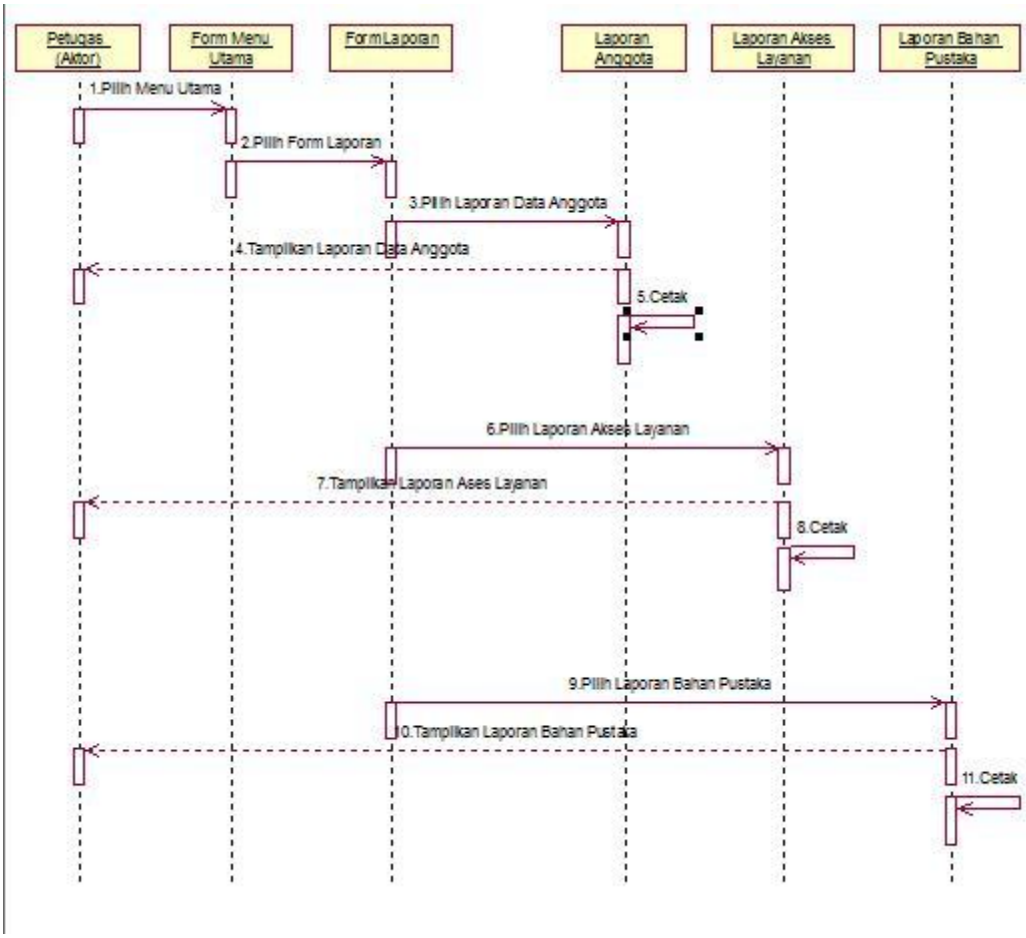
Gambar 2. 6 Contoh Activity Diagram

2.2.5.2.4 Sequence Diagram

Diagram sequence menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sequence maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Banyaknya diagram sequence yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sequence sehingga semakin banyak *use case* yang didefinisikan maka diagram sequence yang harus dibuat juga semakin banyak.

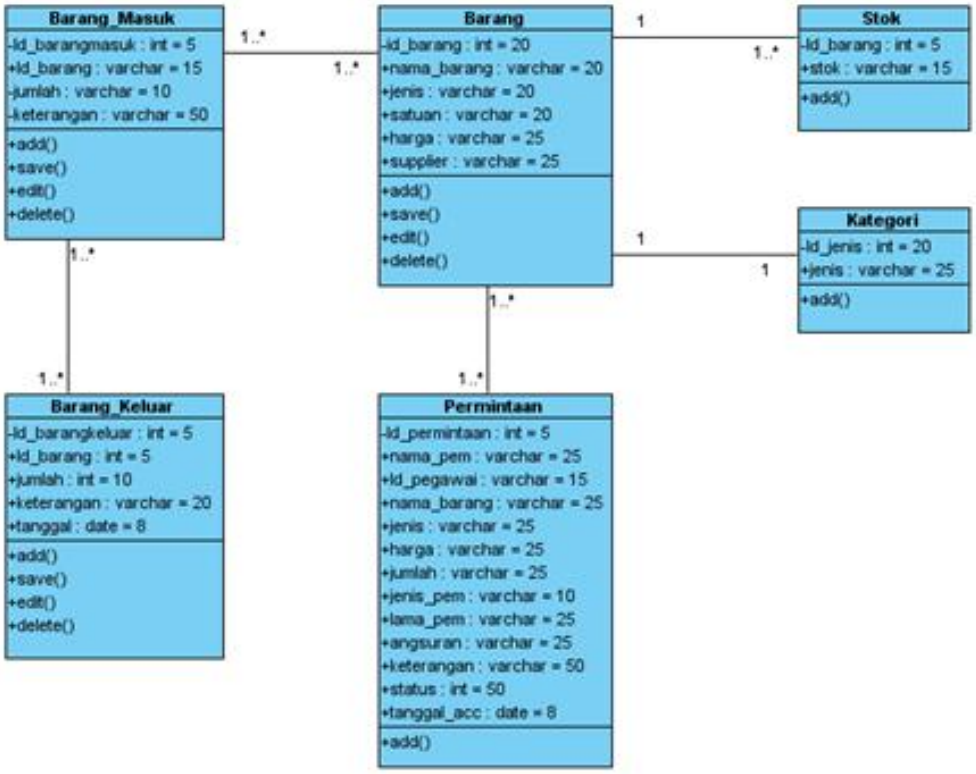
Penomoran pesan berdasarkan urutan interaksi pesan. Penggambaran letak pesan harus berurutan, pesan yang lebih atas dari lainnya adalah pesan yang berjalan terlebih dahulu.



Gambar 2. 7 Contoh Sequence Diagram

2.2.5.2.5 Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.



Gambar 2. 8 Contoh Class Diagram

Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik pada diagram kelas ebaiknya memiliki jenis-jenis kelas berikut :

1) Kelas Main

   Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

2) Kelas yang menangani tampilan sistem

   Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

3) Kelas yang diambil dari pendefinisian *use case*

   Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*.

4) Kelas yang diambil dari pendefinisian data

   Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Jenis-jenis kelas diatas juga dapat digabungkan satu sama lain sesuai dengan pertimbangan yang dianggap baik asalkan fungsi-fungsi yang sebaiknya ada pada struktur kelas tetap ada. Susunan kelas juga dapat ditambahkan kelas utilitas seperti koneksi ke basis data, membaca *file* teks, dan lain sebagainya sesuai kebutuhan.

Dalam mendefinisikan metode yang ada di dalam kelas perlu memperhatikan apa yang disebut dengan *cohesion* dan coupling. *Cohesion* adalah ukuran seberapa dekat keterkaitan instruksi di dalam sebuah metode terkait satu sama lain sedangkan *coupling* adalah ukuran seberapa dekat keterkaitan instruksi antara metode yang satu dengan metode yang lain dalam sebuah kelas. Sebagai aturan secara umum maka sebuah metode yang dibuat harus memiliki kadar *choesion* yang kuat dan kadar *coupling* yang lemah.

Dalam class diagram terdapat beberapa relasi (hubungan antar class) yaitu:

1. *Generalization* dan *Inheritence*

   Diperlukan untuk memperlihatkan hubungan pewarisan (inheritance) antar unsur dalam diagram kelas. Pewarisan memungkinkan suatu kelas mewarisi semua atribut, operasi, relasi, dari kelas yang berada dalam hirarki pewarisannya.

2. *Associations*

   Hubungan statis antar class. Umumnya menggambarkan class yang memiliki atribut berupa class lain, atau class yang harus mengetahui ekstensi class lain. Dalam notasi UML kita mengenal asosiasi 2 arah (bidirectional) dan 1 arah (undirectional).

3. *Aggregation*

   Hubungan antar-class dimana class yang satu (part class) adalah bagian dari class lainnya (whole class).

4. *Composition*

   Aggregation dengan ikatan yang lebih kuat. Di dalam composite aggregation, siklus hidup part class sangat bergantung pada whole class sehingga bila objek instance dari whole class dihapus maka objek instance dari part calss juga akan terhapus.

5. *Depedency*

   Hubungan antar-class dimana sebuah class memiliki ketergantungan pada class lainnya tetapi tidak sebaliknya.

6. *Realization*

   Hubungan antar-class dimana sebuah class memiliki keharusan untuk mengikuti aturan yang ditetapkan class lainnya. Biasanya realization digunakan untk menspesifikasikan hubungan antara sebuah interface dengan class yang mengimplementasikan interface tersebut.