

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

1. Penelitian dari Risa Ferossa (2016) yang berjudul “ Aplikasi Perhitungan Harga Pokok Produksi Dengan Metode *Full Costing* Pada Mommyindo Bandung”. Mommyindo adalah salah satu perusahaan manufaktur dibidang kuliner yang mengolah bahan berupa kerupuk mentah menjadi bahan siap saji yang dinamakan seblak yang dirintis oleh pengusaha muda yang bernama Lofty Rainidi. Mommyindo juga merupakan perusahaan kreatif, hal ini dikarenakan perusahaan tersebut membuat seblak instan berbentuk cup sehingga bisa langsung dijual atau dipasarkan. Berkaitan dengan pengelolaan perusahaan, Mommyindo masih melakukan perhitungan penjualan, pembelian, persediaan dan laporan secara manual. Hal tersebut seringkali memicu terjadinya ketidakcocokan antara laporan dengan biaya yang dikeluarkan. Maka dari itu, penelitian ini dibuat untuk menerangkan aplikasi yang dapat menghitung berapa biaya produksi yang dibutuhkan, menghitung beban-beban yang ada saat produksi, serta menentukan harga jual setiap *cups* dengan menggunakan metode *Full Costing*. Oleh karena itu disusunlah proyek akhir ini yang berjudul “Aplikasi Perhitungan Harga Pokok Produksi Dengan Metode *Full Costing* Pada Mommyindo Bandung”.
2. Penelitian dari Prasetyo Budi Jatmiko (2014) yang berjudul “Sistem dan Aplikasi Penentuan Harga Jual Berdasarkan Biaya Total Pada CV. Terbit

Terang”. Sejak 1997, CV Terbit Terang telah berdiri berproduksi di percetakan dan penerbitan. Selama perusahaan beroperasi, dalam menentukan harga jual perusahaan yang menggunakan estimasi dasar. Perkiraan harga jual adalah empat atau lima kali dari perkiraan biaya produksi per unit. Hal ini bisa terjadi karena perusahaan memperkirakan penentuan harga jual untuk menutupi biaya lain yang tidak teridentifikasi dalam pembuatan produk. Berdasarkan penentuan taksiran harga jual, maka akan terpengaruh CV. Penjualan Terbit Terang dan membuat angka penjualan turun 40%. Masalah ini bisa dilihat pada periode 2008-2009. Penurunan penjualan di perusahaan dapat diatasi dengan penerapan penentuan harga jual. Pendekatannya diambil dengan berdirinya sistem dan aplikasi untuk menentukan harga jual berdasarkan total biaya. Implementasi metode dalam aplikasi ini dapat memberikan informasi keuntungan yang diinginkan. Ini akan diperoleh berdasarkan target pengembalian dana yang telah diinvestasikan dari markup perhitungan. Berdasarkan hasil pengujian, aplikasi tersebut menggunakan data dari CV. Terbit Terang, yang membuat sistem bisa menentukan harga jual dengan total metode biaya. Dengan harga jual yang ditentukan, maka pada periode selanjutnya CV. Terbit Terang memperoleh harga jual berdasarkan laba perusahaan yang ditargetkan biaya investasi.

3. Penelitian dari Sofyan Affandy Yusuf (2010) yang berjudul “ Rancang Bangun Sistem Informasi Akuntansi Dengan Metode *Full Costing* Untuk Menentukan Harga Pokok Produksi Pada PT. Kasa Husada Wira Jatim”. PT. Kasa Husada Wira Jatim adalah perusahaan yang bergerak dalam bidang

produksi alat kesehatan. Adapun alat kesehatan tersebut adalah kasa, kapas untuk kebutuhan medis dengan berbagai jenis dan ukuran. Proses kegiatan pencatatan akuntansi di kasa husada dimulai dari proses pembelian bahan baku, bahan penolong dan lain-lain. Kemudian dilanjutkan dengan proses pencatatan yang berkenaan dengan kegiatan produksi. Proses produksi di PT. Kasa Husada Wira Jatim dibagi menjadi dua yaitu : proses produksi kasa dan proses produksi kapas. Kemudian proses pencatatan penjualan barang jadi berupa hasil dari produksi yaitu : kasa pembalut, kasa hidروفil, kapas bola dan kapas pembalut dengan berbagai ukuran per masing-masing jenis. Manajemen PT. Kasa Husada Wira Jatim sering kali mengalami kesulitan didalam menghitung biaya produksi dan penyusunan laporan produksi, yang diakibatkan oleh proses yang masih dilakukan secara manual. Sedangkan untuk proses pembelian dan penjualan sudah dilakukan secara komputerisasi, tetapi untuk proses pembelian dan penjualan tidak terintegrasi dengan proses atau modul akuntansi, sehingga menyebabkan tidak efektif dan efisien dalam penyusunan laporan keuangan. Oleh karena itu untuk meningkatkan kerja yang efektif dan efisien maka PT. Kasa Husada Wira Jatim menghendaki suatu sistem baru yang saling terintegrasi antara proses penjualan, pembelian, produksi dan proses akuntansi, sehingga dapat menghasilkan laporan biaya produksi dan laporan keuangan dengan cepat dan tepat. Dengan adanya sistem informasi akuntansi ini diharapkan dapat membantu permasalahan perusahaan tersebut mulai dari transaksi pembelian bahan baku dan bahan penolong, perhitungan biaya-biaya yang terjadi pada saat proses produksi barang, sampai

dengan proses penjualan barang jadi serta pembuatan laporannya, tidak menutup kemungkinan adanya tambahan untuk memperbaiki beberapa kekurangan yang terdapat pada sistem ini sehingga berguna dalam proses pengambilan keputusan perusahaan.

II.2. Uraian Teoritis

II.2.1. Pengertian Sistem

Sistem merupakan kumpulan dari unsur atau elemen-elemen yang saling berkaitan/berinteraksi dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu. (Asbon Hendra : 2012 : 157 - 158).

Berikut ini merupakan pengertian sistem dari beberapa ahli:

1. Jerry FithGerald, "Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan dan berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu.
2. Ludwig Von Bartalanfy, "Sistem merupakan seperangkat unsur yang saling terikat dalam suatu antar relasi di antara unsur-unsur tersebut dengan lingkungan.
3. Anatol Raporot, "Sistem adalah suatu kumpulan kesatuan dan perangkat hubungan satu sama lain.

4. L. Ackof, "Sistem adalah setiap kesatuan secara konseptual atau fisik yang terdiri dari bagian-bagian dalam keadaan saling tergantung satu sama lainnya.

Dari uraian di atas, sehingga dapat disimpulkan bahwa sistem adalah sekumpulan elemen yang saling terkait atau terpadu untuk mencapai tujuan tertentu.

1. Karakteristik Sistem

Model umum sebuah sistem terdiri dari *input*, proses dan *output*. Hal ini merupakan konsep sebuah sistem yang sangat sederhana mengingat sebuah sistem dapat mempunyai beberapa masukan dan keluaran sekaligus. Selain itu sebuah sistem juga memiliki karakteristik atau sifat-sifat tertentu, yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem.

Adapun karakteristik yang dimaksud adalah sebagai berikut:

a. Komponen Sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang berinteraksi, bekerja sama membentuk satu kesatuan. Komponen-komponen sistem dapat berupa suatu bentuk subsistem atau bagian-bagian dari sistem. Setiap sistem mempunyai sifat-sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai sistem yang lebih besar yang disebut dengan supra sistem.

b. Batasan Sistem (*Boundary*)

Batas sistem merupakan daerah yang membatasi antara sistem dengan sistem lainnya atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan, karena dengan batas sistem ini fungsi dan tugas dari subsistem yang satu dengan lainnya berbeda tetapi tetap saling berinteraksi. Batas suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

c. Lingkungan Luar Sistem (*Environment*)

Environment merupakan segala sesuatu di luar batas sistem yang memengaruhi operasi dari suatu sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan atau merugikan. Lingkungan luar yang menguntungkan harus dipelihara dan dijaga agar tidak hilang pengaruhnya, sedangkan lingkungan luar yang merugikan harus dimusnahkan atau dikendalikan agar tidak mengganggu operasi sistem.

d. Penghubung Sistem (*Interface*)

Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya untuk membentuk satu kesatuan sehingga yang lainnya untuk membentuk satu kesatuan sehingga sumber-sumber daya mengalir dari subsistem yang satu ke subsistem yang lainnya. Dengan kata lain, *output* dari suatu subsistem akan menjadi *input* dari subsistem yang lainnya.

e. Masukan Sistem (*Input*)

Merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*Maintenance Input*) adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Masukan sinyal (*Signal Input*) adalah energi yang diproses untuk didapatkan keluaran. Sebagai contoh, di dalam sistem komputer, program adalah *maintenance input* yang digunakan untuk mengoperasikan komputernya dan data adalah *signal input* untuk diolah menjadi informasi.

f. Keluaran Sistem (*Output*)

Merupakan hasil dari energi yang diolah oleh sistem, meliputi *output* yang berguna, contohnya informasi yang dikeluarkan oleh komputer. Dan *output* yang tidak berguna dikenal sebagai sisa pembuangan, contohnya panas yang dikeluarkan oleh komputer.

g. Pengolahan Sistem (*Procces*)

Merupakan bagian yang memproses masukan untuk menjadi keluaran yang diinginkan.

h. Tujuan Sistem (*Goal*)

Setiap sistem pasti mempunyai tujuan ataupun sasaran yang memengaruhi *input* yang dibutuhkan dan *output* yang dihasilkan. Dengan kata lain, suatu sistem akan dikatakan berhasil kalau pengoperasian sistem itu mengenai sasaran atau tujuannya. Jika sistem

tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya (Asbon Hendra : 2012 : 158 - 160).

II.2.2. Informasi

Informasi merupakan hasil atau keluaran dari sistem informasi. Informasi yang berkualitas adalah informasi yang didalamnya bukan saja diperlukan adanya keakurasian dan kecepatan akan tetapi juga diperlukan adanya kesesuaian dengan kebutuhan manajemen dan kelengkapan dari informasi yang dihasilkan. (Djajun Djuhara, Januariska : 2014 : 113).

II.2.3. Sistem Informasi

Sistem informasi merupakan komponen-komponen yang saling berhubungan dan bekerja sama untuk mengumpulkan, memproses, menyimpan dan mendistribusikan informasi tersebut untuk mendukung proses pengambilan keputusan, kordinasi dan pengendalian. (Djajun Djuhara, Januariska : 2014 : 114).

II.3. Sistem Informasi Akuntansi

II.3.1. Pengertian Akuntansi

Akuntansi dapat didefinisikan dari dua sudut pandang, yaitu dari sudut pandang pemakai jasa akuntansi dan dari sudut pandang proses kegiatannya, ditinjau dari sudut pandang pemakaiannya, akuntansi adalah suatu disiplin ilmu yang menyediakan informasi yang diperlukan untuk melaksanakan kegiatan

secara efisien dan mengevaluasi kegiatan-kegiatan suatu organisasi, jika ditinjau dari sudut kegiatannya, akuntansi didefinisikan sebagai proses pencatatan, penggolongan, peringkasan, pelaporan dan penganalisisan data keuangan suatu entitas. (Cathry Septiani Putri et, al : 2016 : 104).

Ahmad dan Abdullah (2012:17) menyatakan bahwa akuntansi (*accounting*) merupakan suatu kegiatan atau jasa yang berfungsi menyediakan informasi kuantitatif terutama yang bersifat keuangan mengenai kesatuan-kesatuan ekonomi tertentu kepada pihak-pihak yang berkepentingan, untuk digunakan sebagai bahan pertimbangan dalam pengambilan keputusan-keputusan ekonomi. Akuntansi adalah sebuah kegiatan jasa, fungsinya adalah menyediakan informasi kuantitatif, terutama yang bersifat keuangan, tentang entitas ekonomi yang dimaksudkan agar berguna dalam mengambil keputusan ekonomi dalam membuat pilihan-pilihan nalar di antara berbagai alternatif arah tindakan. (Juvita Pratiwi : 2013 : 1619).

II.3.2. Sistem Informasi Akuntansi

Sistem informasi akuntansi adalah sebuah sistem yang memproses data transaksi guna menghasilkan informasi yang bermanfaat untuk merencanakan, mengendalikan dan mengoperasikan bisnis. Sedangkan sistem informasi akuntansi menurut La Midjan dan Azhar Susanto yaitu, suatu sistem pengolah dokumen akuntansi yang merupakan koordinasi manajemen, alat dan metode berinteraksi secara harmonis dalam wadah organisasi yang berstruktur untuk menghasilkan

informasi keuangan dan informasi akuntansi manajemen yang terstruktur pula. (Djajun Djuhara, Januariska : 2014 : 114).

Adapun beberapa penelitian sebelumnya yang pernah dilakukan penelitian adalah sebagai berikut:

1. Penelitian yang telah dilakukan oleh Lintang Arini (2017) yang berjudul “Analisis Penentuan Harga Pokok Produksi Untuk Menetapkan Harga Jual Menggunakan Metode *Ful Costing* Pada CV. Purnama Onix Periode 2011-2015 (Studi Kasus CV. Purnama Onix). Perbedaan signifikan pada harga jual karena perusahaan tidak membebankan biaya tenaga kerja, biaya *overhead* pabrik, biaya bahan baku dan laba secara proporsional sehingga terjadi perbedaan harga jual yang ditetapkan perusahaan dengan dilakukan oleh penelitian ini yaitu menggunakan metode *Full Costing*. Dengan menggunakan metode *Full Costing* maka harga jual bisa maksimal. Hal ini disebabkan pembebanan biaya bahan baku, biaya *overhead* pabrik, biaya tenaga kerja dan laba dialokasikan secara proporsional dengan jumlah produksi agar mendapatkan keuntungan maksimal. Namun kelemahan dari penelitian ini adalah tidak adanya aplikasi yang dirancang untuk mempermudah dalam pencatatan biaya bahan baku, biaya *overhead*, biaya tenaga kerja dan perhitungan laba secara terkomputerisasi.
2. Penelitian yang telah dilakukan oleh Akmal Abdullah (2016) yang berjudul ” Analisis Perhitungan Biaya Produksi Berdasarkan Metode *Full Costing* Dalam Menentukan Harga Jual Produk Lemari (Studi pada Usaha Meubel Sumber Riski, Kabupaten Konawe Selatan)”. Perhitungan harga pokok produksi

dengan menggunakan metode *Full Costing* yang diperhitungkan oleh perusahaan lebih

rendah dibandingkan perhitungan dengan menggunakan metode *Full Costing* yang sebenarnya. Perhitungan yang dipaparkan dalam jurnal ini cukup jelas dan akurat namun tidak secara terkomputerisasi.

3. Penelitian yang telah dilakukan oleh Endra Setianingsih (2013) dengan judul ” Analisis Penerapan Metode *Full Costing* Dalam Perhitungan Harga Pokok Produksi Untuk Penetapan Harga Jual (Studi Kasus Pada Pabrik Tahu Lestari)”. Dari hasil metode yang digunakan tersebut ada banyak perbedaan yang terjadi dalam biaya produksi antara perhitungan pabrik dengan perhitungan yang sebenarnya. Lebih terperinci dan realistis biaya-biaya yang dikeluarkan pabrik untuk selama proses produksi.
4. Penelitian yang telah dilakukan oleh Zulia Hanum (2013) dengan judul ”Sistem Informasi Akuntansi Penuh Dalam Penentuan Harga Jual Pada PT. Coca Cola Bottling Indonesia Medan”. Penerapan Metode *Full Costing* dalam penentuan harga jual sangat berpengaruh. Karena dengan metode ini maka akan menunjukkan biaya-biaya yang digunakan dalam proses produksi baik itu biaya langsung maupun biaya tidak langsung.
5. Penelitian yang telah dilakukan oleh Ronaldo Mardi Putra, Magdalena Karismariyanti, Eti Suprihatin (2013) dengan judul “Aplikasi Penentuan Harga Jual Menggunakan Metode *Full Costing* Pada CV. Optik Lanamai Makassar”. Aplikasi ini mampu mengelola pembelian, penjualan setelah ditambah biaya bahan baku, biaya tenaga kerja, biaya *overhead* pabrik *fix* dan

biaya *overhead* variabel serta menghasilkan keluaran sesuai dengan yang diharapkan. Mampu mengelola kegiatan produksi, mampu menampilkan jurnal transaksi yang diinginkan dan sesuai dengan yang diharapkan. Namun aplikasi ini tidak dapat menghitung selisih biaya *overhead* pabrik dan belum bisa mencatat biaya yang rusak akibat kesalahan produksi dan perhitungan aktiva yang rusak.

II.4. Biaya Produksi

Pengertian dari biaya produksi adalah biaya yang diperlukan untuk memperoleh bahan baku (mentah) dari pemasok dan mengubahnya menjadi produk selesai yang siap dijual.

Adapun elemen – elemen pada biaya produksi antara lain:

1. **Bahan baku.** Bahan yang digunakan untuk membuat produk selesai. Bahan baku dapat diidentifikasi ke produk dan merupakan bagian integral dari produk tersebut.
2. **Tenaga kerja langsung.** Tenaga yang langsung menangani proses produksi. Mereka menangani secara langsung proses produksi dan oleh karena itu dapat diidentifikasi ke produk. Gaji atau upah tenaga kerja langsung merupakan elemen biaya produksi.
3. **Overhead pabrik.** Biaya – biaya selain bahan baku dan tenaga kerja langsung yang diperlukan untuk memproduksi barang disebut biaya *overhead* pabrik (*factory overhead* atau *factory burden*). *Overhead* pabrik termasuk biaya

langsung produk sebab biaya tersebut dapat diidentifikasi ke biaya yang diproduksi tadi. (Slamet Sugiri Sodikin : 2015 : 22).

II.5. Metode *Full Costing*

Menurut *Full Costing (absorption costing)*, biaya produk meliputi seluruh komponen biaya untuk membuat produk. Biaya produk menurut metode ini meliputi bahan baku, biaya tenaga kerja langsung, biaya *overhead* variabel dan biaya *overhead* tetap. (Slamet Sugiri Sodikin : 2015: 83).

Full Costing adalah metode untuk menentukan harga pokok produksi, dengan membebankan semua biaya produksi tetap maupun variabel pada produk yang dihasilkan. Metode *Full Costing* disebut juga *absortion* atau *conventional costing*.

Menurut R.A. Supriyono (2002), menerangkan *Full Costing Method* adalah konsep penentuan harga pokok penuh, membebankan semua elemen biaya produksi, baik biaya tetap maupun biaya variabel, ke dalam harga pokok produk. Oleh karena itu elemen biaya produksi pada konsep penentuan harga pokok penuh meliputi biaya bahan baku, biaya tenaga kerja langsung dan biaya *overhead* pabrik.

Sedangkan Mulyadi (2009) mendefinisikan *Full Costing Method* merupakan metode penentuan harga pokok produksi yang memperhitungkan semua unsur biaya produksi ke dalam harga pokok produksi, yang terdiri dari biaya bahan baku, biaya tenaga kerja langsung dan biaya *overhead* pabrik, baik

yang berperilaku variabel maupun tetap. (W. Wiratna Sujarweni : 2015 : 148-149).

Contoh Penyajian Laporan Harga Pokok Produksi Metode Full Costing

Harga Pokok Produksi :

Biaya Bahan Baku		Rp. xxx.xxx
Biaya Tenaga Kerja Langsung	Rp. xxx.xxx	
Biaya Overhead Pabrik Tetap	Rp. xxx.xxx	
Biaya Overhead Pabrik Variabel	Rp. xxx.xxx	
Harga Pokok Produk	Rp. xxx.xxx	

(Sumber : W. Wiratna Sujarweni :2015 :148-149).

II.6. Pengertian Database

Database merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan untuk memanipulasinya. *Database* merupakan salah satu komponen yang penting dalam sistem informasi karena merupakan basis dalam menyediakan informasi bagi para

pemakai. Penerapan *database* dalam sistem informasi disebut dengan *database system*. Sistem basis data (*database system*) adalah suatu sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan yang lainnya dan membuatnya tersedia untuk beberapa aplikasi yang bermacam-macam di dalam suatu organisasi. Dengan sistem dasar data ini tiap-tiap orang atau bagian dapat memandang *database* dari beberapa sudut pandangan yang berbeda. Bagian kredit dapat memandangnya sebagai data piutang. Bagian penjualan dapat memandangnya sebagai data penjualan. Bagian personalia dapat memandangnya sebagai data relawan. Semua terintegrasi dalam sebuah data yang umum. Berbeda dengan sistem pengolahan data tradisional (*traditional processing system*), sumber data ditangani sendiri untuk tiap-tiap aplikasi. (Samsuri Ridwan, et al. : 2014 : 45).

II.7. Kamus Data

Kamus data adalah katalog fakta tentang data dan kebutuhan informasi suatu sistem informasi. Kamus data terdapat pada tahapan analisis dan perancangan. Pada tahap analisis, kamus data berfungsi untuk mendefinisikan data yang mengalir pada sistem. Sedangkan pada tahap perancangan, kamus data ini digunakan untuk merancang masukan dan keluaran seperti laporan serta basis data. Pada DFD aliran data memiliki sifat global, sedangkan pada kamus data dibuat berdasarkan aliran data yang terdapat pada DFD. (Indrajani : 2015 : 30).

Tabel II.1. Notasi Kamus Data

Notasi	Keterangan
=	<i>Is Composed Of</i>
+	<i>And</i>
()	<i>Optional (May be present or absent)</i>
{ }	<i>Iteration</i>
[]	<i>Select one of several alternative choices</i>
**	<i>Comment</i>
@	<i>Identifier (key field) for a store</i>
	<i>Separates alternative choices in the construct</i>

(Sumber : Indrajani : 31)

Contoh kamus data, antara lain:

name = courtesy-title + first-name +(middle-name) + last-name

courtesy-title = [Mr. | Miss | Mrs. | Ms. | Dr. | Profesor]

first-name = {legal-character}

middle-name = {legal-character}

last-name = {legal-character}

legal-character = [A-Z|a-z|0-9|'|_|]

II.8. Normalisasi

Normalisasi diartikan sebagai suatu teknik yang menstrukturkan mendekomposisi data dalam cara-cara tertentu untuk mencegah timbulnya permasalahan pengolahan data dalam basis data. Permasalahan yang dimaksud

adalah berkaitan dengan penyimpangan-penyimpangan (*anomallies*) yang terjadi akibat adanya kerangkapan data dalam relasi dan in-efisiensi pengolahan. (Martin, 1975) : (Edy Sutanta : 2011 : 174-175).

Proses normalisasi menghasilkan relasi yang optimal, yaitu:

1. Memiliki struktur *record* yang konsisten secara logik;
2. Memiliki struktur *record* yang mudah untuk dimengerti;
3. Memiliki struktur *record* yang sederhana dalam pemeliharaan;
4. Memiliki struktur *record* yang mudah ditampilkan kembali untuk memenuhi kebutuhan pengguna;
5. Minimalisasi kerangkapan data guna meningkatkan kinerja sistem.

Secara berturut-turut masing-masing level normal tersebut dibahas berikut ini, dimulai dari bentuk tidak normal. (Edy Sutanta : 2011 : 176-179)

1. Relasi bentuk tidak normal (*Un Normalized Form* / UNF)

Relasi-relasi yang dirancang tanpa mengindahkan batasan dalam definisi basis data dan karakteristik *Relational Database Management System* (RDBM) menghasilkan relasi *Un Normalized Form* (UNF). Bentuk ini harus di hindari dalam perancangan relasi dalam basis data. Relasi *Un Normalized Form* (UNF) mempunyai kriteria sebagai berikut.

- a. Jika relasi mempunyai bentuk *non flat file* (dapat terjadi akibat data disimpan sesuai dengan kedatangannya, tidak memiliki struktur tertentu, terjadi duplikasi atau tidak lengkap).
- b. Jika relasi membuat *set atribut* berulang (*non single values*).

c. Jika relasi membuat *atribut non atomic value*.

2. Relasi bentuk normal pertama (*First Norm Form / 1NF*)

Relasi disebut juga *First Norm Form (1NF)* jika memenuhi kriteria sebagai berikut:

- a. Jika seluruh atribut dalam relasi bernilai *atomic (atomic value)*.
- b. Jika seluruh atribut dalam relasi bernilai tunggal (*single value*).
- c. Jika relasi tidak memuat set atribut berulang.
- d. Jika semua record mempunyai sejumlah atribut yang sama.

Permasalahan dalam *First Norm Form (1NF)* adalah sebagai berikut.

- a. Tidak dapat menyisipkan informasi parsial.
- b. Terhapusnya informasi ketika menghapus sebuah *record*.

3. Bentuk normal kedua (*Second Normal Form / 2NF*)

Relasi disebut sebagai *Second Normal Form (2NF)* jika memenuhi kriteria sebagai berikut:

- a. Jika memenuhi kriteria *First Norm Form (1NF)*.
- b. Jika semua atribut nonkunci *Functional Dependence (FD)* pada *Primary Key (PK)*.

Permasalahan dalam *Second Normal Form / 2NF* adalah sebagai berikut:

- a. Kerangkapan data (*data redundancy*).
- b. Pembaharuan yang tidak benar dapat menimbulkan inkonsistensi data (*data inconsistency*).
- c. Proses pembaharuan data tidak efisien.

Kriteria tersebut mengidentifikasi bahwa antara atribut dalam *Second Normal Form* masih mungkin mengalami *Third Norm Form*. Selain itu, relasi *Second Normal Form* (2NF) menuntut telah didefinisikan atribut *Primary Key* (PK) dalam relasi. Mengubah relasi *First Norm Form* (1NF) menjadi bentuk *Second Normal Form* (2NF) dapat dilakukan dengan mengubah struktur relasi dengan cara:

- a. Identifikasikan *Functional Dependence* (FD) relasi *First Norm Form* (1NF).
 - b. Berdasarkan informasi tersebut, dekomposisi relasi *First Norm Form* (1NF) menjadi relasi-relasi baru sesuai *Functional Dependence* nya. Jika menggunakan diagram maka simpul-simpul yang berada pada puncak diagram ketergantungan data bertindak *Primary Key* (PK) pada relasi baru.
4. Bentuk normal ketiga (*Third Norm Form* / 3NF)

Suatu relasi disebut sebagai *Third Norm Form* jika memenuhi kriteria sebagai berikut:

- a. Jika memenuhi kriteria *Second Normal Form* (2NF).
- b. Jika setiap atribut nonkunci tidak (*TDF*) (*Non Transitive Dependency*) terhadap *Primary Key* (PK).

Permasalahan dalam *Third Norm Form* (3NF) adalah keberadaan penentu yang tidak merupakan bagian dari *Primary Key* (PK) menghasilkan duplikasi rinci data pada atribut yang berfungsi sebagai *Foreign Key* (FK) (duplikasi berbeda dengan keterangan data).

Mengubah relasi *Second Normal Form* (2NF) menjadi bentuk *Third Norm Form* (3NF) dapat dilakukan dengan mengubah struktur relasi dengan cara :

- a. Identifikasi TDF relasi *Second Normal Form* (2NF).
- b. Berdasarkan informasi tersebut, dekomposisi relasi *Second Normal Form* (2NF) menjadi relasi-relasi baru sesuai TDF-nya.

5. Bentuk normal *Boyce-Codd* (*Boyce-Codd Norm Form* / BCNF)

Bentuk normal *Boyce-Codd Norm Form* (BCNF) dikemukakan oleh R.F. Boyce dan E.F. Codd. Suatu relasi disebut sebagai *Boyce-Codd Norm Form* (BCNF) jika memenuhi kriteria sebagai berikut:

- a. Jika memenuhi kriteria *Third Norm Form* (3NF).
- b. Jika semua atribut penentu (*determinan*) merupakan CK.

6. Bentuk normal keempat (*Forth Norm Form* / 4NF)

Relasi disebut sebagai *Forth Norm Form* (4NF) jika memenuhi kriteria sebagai berikut:

- a. Jika memenuhi kriteria *Boyce-Codd Norm Form*.
- b. Jika setiap atribut didalamnya tidak mengalami ketergantungan pada banyak nilai.

7. Bentuk normal kelima (*Fifth Norm Form* / 5NF)

Suatu relasi memenuhi kriteria *Fifth Norm Form* (5NF) jika kerelasiaan antar data dalam relasi tersebut tidak dapat direkonstruksi dari struktur relasi yang sederhana.

8. Bentuk normal kunci domain (*Domain Key Norm Form* / DKNF)

Relasi disebut sebagai *Domain Key Norm Form* (DKNF) jika setiap batasan dapat disimpulkan secara sederhana dengan mengetahui sekumpulan nama atribut dan domainnya selama menggunakan sekumpulan atribut pada kuncinya.

II.9. *Unified Modeling Language* (UML)

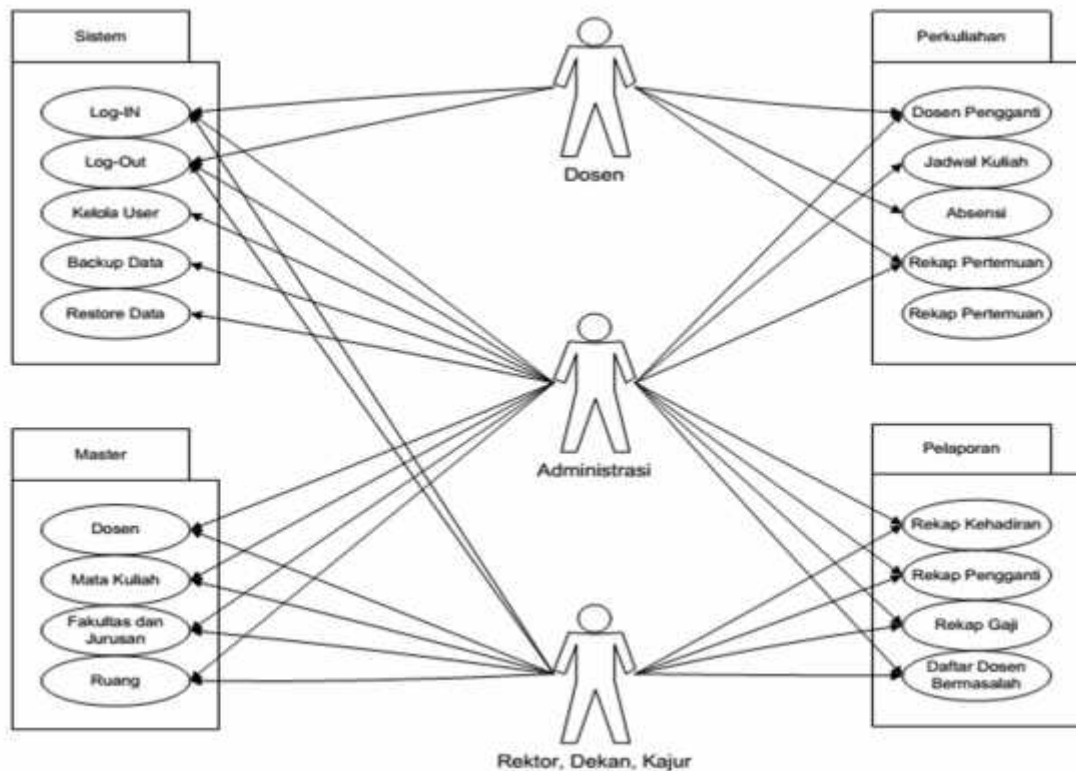
UML adalah bahasa untuk mengspesifikasi, memvisualisasi, membangun dan mendokumentasikan *artefact* (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya). UML juga menggunakan *class* dan *operation* 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 : 2016 : 51-52).

Menurut Windu Gata, Grace (2013:4), *Unified Modeling Language* (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga

merupakan alat untuk mendukung pengembangan sistem. (Ade Hendini : 2016 : 108).

II.9.1. Use Case Diagram

Use case diagram digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. *Use case* diagram terdiri atas diagram untuk *use case* dan *actor*. *Actor* merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi. *Use case* merepresentasikan operasi-operasi yang dilakukan oleh *actor*. *Use case* digambarkan berbentuk elips dengan nama operasi dituliskan di dalamnya. *Actor* yang melakukan operasi dihubungkan dengan garis lurus ke *use case*. (Arif Yulianto : 2015 : 218).

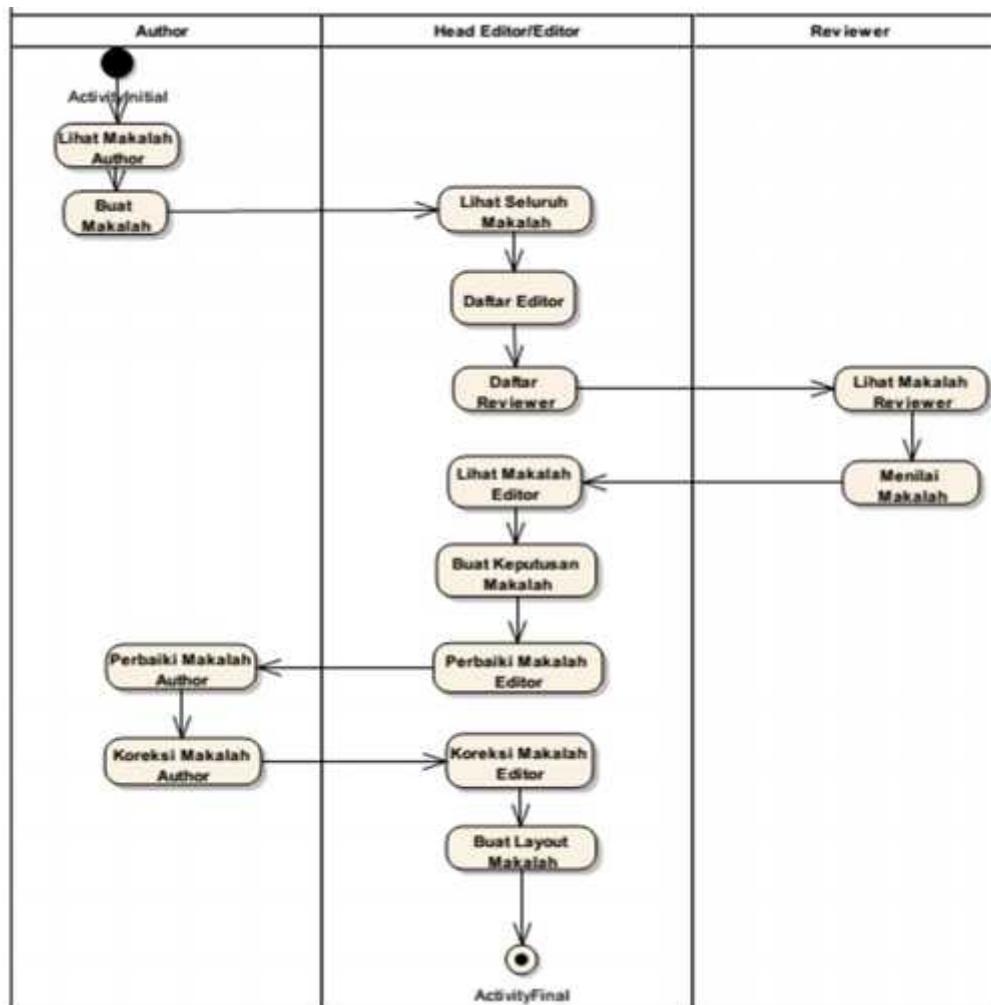


Gambar II.1. Use Case Diagram
(Sumber : Arif Yulianto : 2015 : 219)

II.9.2. Activity Diagram

Activity diagram adalah diagram yang menggambarkan sifat dinamis secara alamiah sebuah sistem dalam bentuk model aliran dan kontrol dari aktivitas ke aktivitas lainnya. (Yosua P.W Simeremere, et al. : 2013 : 471).

Activity diagram menggambarkan berbagai aliran aktivitas (*work flow*) dalam sistem yang sedang dirancang, bagaimana masing-masing aliran berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. (Abulwafa Muhammad, et al. : 2013 : 57-58).

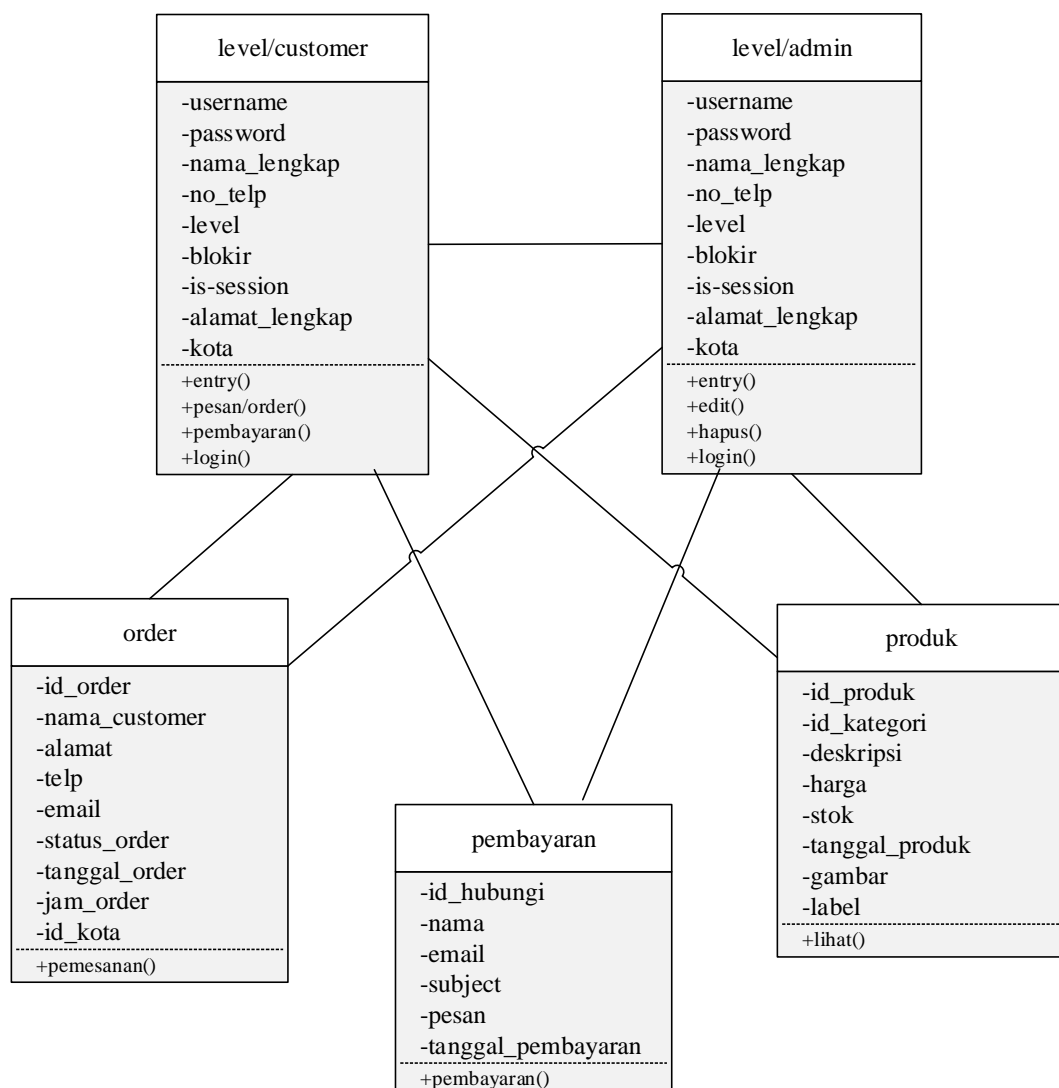


Gambar II.2. Activity Diagram

(Sumber : Yosua P.W Simeremere, et al. : 2013 : 472)

II.9.3. Class Diagram

Class Diagram menggambarkan *class* dalam sebuah sistem dan hubungannya antara satu dengan yang lain, serta dimasukkan pula atribut dan operasi. Umumnya *class* diagram dari suatu sistem akan menggambarkan juga bagaimana struktur *database* yang dibutuhkan untuk membangun sistem tersebut. (Abulwafa Muhammad, et al.: 2013 : 57).

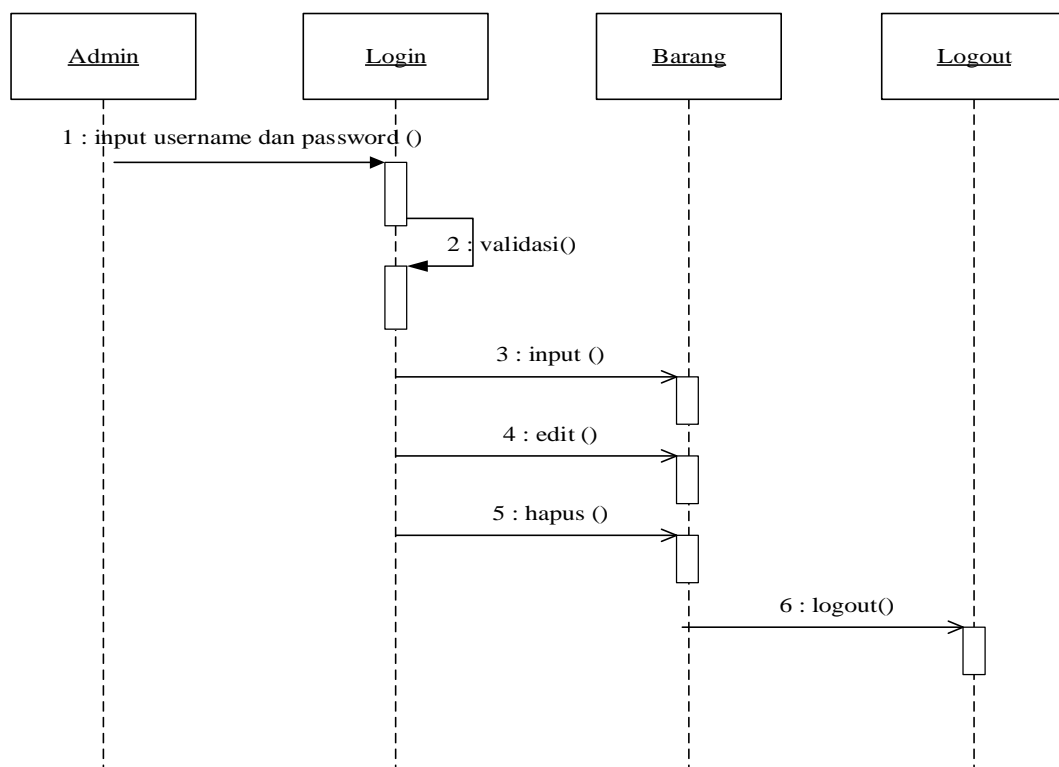


Gambar II.3. Class Diagram
(Sumber : Abulwafa Muhammad, et al. : 2013 : 57)

II.9.4. Sequence Diagram

Sequence diagram adalah suatu diagram yang memperlihatkan/ menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, *display*, dan sebagainya berupa pesan/*message*. (Yosua P.W Simeremere, et al. : 2013 : 471).

Sequence diagram digunakan untuk menggambarkan perilaku aktor pada sebuah sistem secara detail menurut waktu. Diagram ini menunjukkan sejumlah contoh objek dan *message* (pesan) yang diletakkan diantara objek-objek di dalam *use case*. (Abulwafa Muhammad, et al. :2013 : 58).



Gambar II.4. Sequence Diagram
(Sumber : Abulwafa Muhammad, et al. : 2013 : 58)

II. 10. Microsoft Visual Basic 2010

Microsoft Visual Basic adalah bahasa pemrograman yang digunakan untuk membuat aplikasi *Windows* yang berbasis grafis (GUI–*Graphical User Interface*). Saat itu masih *Windows* 3.11. sebagai bahasa pemrograman yang tergolong RAD (*Rapid Application Development*) tool dan perkembangan *Visual Basic* mendapatkan sambutan yang sangat baik di kalangan *programmer*. *Visual Basic* merupakan bahasa pemrograman berbasis visual (*Windows*). Program ini sangat diminati di seluruh dunia oleh para pengguna jasa komputer karena bahasanya yang mudah dan fitur-fiturnya yang familiar. Ada banyak hal yang mendasari kenapa pemrograman berbasis *Visual Basic* digemari banyak orang, yang pertama karena mudah dan yang kedua cepat. (Iman Saufik Suasana dan Jatu Suryantari : 2016 : 37).

Microsoft Visual Basic .NET adalah sebuah alat untuk mengembangkan dan membangun aplikasi yang bergerak di atas sistem *.NET Framework*, dengan menggunakan bahasa BASIC. Dengan menggunakan alat ini, para *programmer* dapat membangun aplikasi *Windows Forms*, Aplikasi web berbasis ASP.NET, dan juga aplikasi *command line*. Alat ini dapat diperoleh secara terpisah dari beberapa produk lainnya (seperti *Microsoft Visual C++*, *Visual C#*, atau *Visual J#*), atau juga dapat diperoleh secara terpadu dalam *Microsoft Visual Studio .NET*. Bahasa *Visual Basic .NET* sendiri menganut paradigma bahasa pemrograman berorientasi objek yang dapat dilihat sebagai evolusi dari *Microsoft Visual Basic* versi sebelumnya yang diimplementasikan di atas *.NET Framework*. Peluncurannya mengundang kontroversi, mengingat banyak sekali perubahan yang dilakukan

oleh *Microsoft* dan versi baru ini tidak kompatibel dengan versi terdahulu. Teknologi *.NET* muncul karena adanya beberapa alasan pada teknologi aplikasi *Microsoft* yang lama. (Ahmad Rais Ruli : 2017 : 11).

VB.NET mempunyai beberapa fitur baru yang membuat bahasa VB menjadi lebih kuat sehingga dapat mematahkan mitos bahwa VB hanya bahasa mainan (*toy language*) bila dibandingkan dengan bahasa lain seperti C++ dan Java. Fitur-fitur tersebut antara lain :

1. Dukungan *Object Oriented Programming*

VB.NET adalah bahasa pemrograman yang penuh *Object Oriented*. Jadi *VB.NET* mendukung fitur-fitur OOP seperti *inheritance*, *interface*, *method overloading* dan *polymorphism*.

2. *Structure Exception Handling*

Untuk menggantikan perintah *On Error Goto* pada VB6, *VB.NET* menyediakan *error handling* yaitu *Try..Catch..Finally*. *Error handling* pada *VB.NET* ini lebih mudah digunakan karena hanya cukup menaruh kode yang akan dicek dibagian *Try*, dan menyiapkan *exception handling* nya dibagian *catch*.

3. GDI+

GDI+ adalah *library* grafik yang digunakan untuk mengembangkan aplikasi *windows form*.

4. *Web Services* dan *Web Form*

Dengan penggunaan *VB.NET* kita dapat membuat aplikasi berbasis web dengan menggunakan *Web Form (ASP.NET)*. Kita juga dapat membuat aplikasi *web service* untuk membuat *three tier application*.

5. *Cross-Language Interoperability*

Oleh karena disetiap program yang berjalan di *.NET* dikompilasi menjadi *assembly* maka dapat dibuat aplikasi dengan bahasa pemrograman yang berbeda yang berjalan diatas *platform .NET* seperti C# dan C++. Jadi dapat juga menggunakan komponen yang dibuat menggunakan C# atau C++ untuk digunakan di VB.

6. *Multithreading*

Fitur ini sangat berguna apabila terdapat aplikasi yang proses komputasinya memakan waktu lama.

7. *Type Safe Collection*

Fitur ini mulai ada di *.NET 2.0 (VB 8 atau VB 2005)*. Dengan fitur ini dapat dibuat *object collection* yang *type safe*.

8. *LINQ (Language Integrated Query)*

Fitur ini mulai ada pada *.NET 3.5 (VB 9 atau VB 2008)*. Fitur ini digunakan untuk dilakukan *query* data yang ditambahkan kedalam bahasa VB dan C# sehingga kedua bahasa tersebut dapat melakukan *query* ke objek *database*. (Ahmad Rais Ruli : 2017 : 11-12).

II.11. Microsoft SQL Server 2008

Microsoft SQL Server adalah sebuah sistem manajemen basis data relasional (RDBMS) produk *Microsoft*. Bahasa query utamanya adalah *Transact-SQL* yang merupakan implementasi dari *SQL* standar ANSI/ISO yang digunakan oleh *Microsoft* dan *Sybase*. Umumnya *SQL Server* digunakan di dunia bisnis yang memiliki basis data berskala kecil sampai dengan menengah tetapi kemudian berkembang dengan digunakannya *SQL Server* pada basis data besar.

SQL Server 2008 menyimpan data dengan konsep *Relational Database*. Selain itu penyajiannya merupakan penyajian pada level fisik karena kita akan langsung menyimpan data pada *database* dengan kondisi yang sebenarnya, yaitu disimpan pada tabel apa, kolom nama, dan menggunakan tipe data apa saat penyimpanan. (Ilham Akbar, et al. : 2013 : 90).

SQL Server 2008 adalah sebuah terobosan baru dari *Microsoft* dalam bidang *database*. *SQL Server* adalah DBMS (*Database Management System*) yang dibuat oleh *Microsoft* untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti IBM dan *Oracle*. *SQL Server 2008* dibuat pada saat kemajuan dalam bidang *hardware* sedemikian pesat. Oleh karena itu sudah dapat dipastikan bahwa *SQL Server 2008* membawa beberapa terobosan dalam bidang pengolahan dan penyimpanan data. *Microsoft* merilis *SQL Server 2008* dalam beberapa versi yang disesuaikan dengan segment-segment pasar yang dituju.

Menurut cara pemrosesan data pada prosesor maka *Microsoft* mengelompokkan produk ini berdasarkan 2 jenis yaitu :

1. Versi 32-bit (x86), yang biasanya digunakan untuk komputer dengan *single* prosesor (Pentium 4) atau lebih tepatnya prosesor 32 bit dan sistem operasi Windows XP.
2. Versi 64-bit (x64), yang biasanya digunakan untuk komputer dengan lebih dari satu prosesor (Misalnya *Core 2 Duo*) dan sistem operasi 64 bit seperti Windows XP 64, Vista, dan Windows 7.

Sedangkan secara keseluruhan terdapat versi-versi seperti berikut ini :

1. Versi *Compact*, ini adalah versi “Tipis” dari semua versi yang ada. Versi ini seperti versi *desktop* pada *SQL Server 2000*. Versi ini juga digunakan pada *handheld device* seperti Pocket PC, PDA, *SmartPhone*, Tablet PC.
2. Versi *Express*, ini adalah versi “Ringan” dari semua versi yang ada (tetapi versi ini berbeda dengan versi *compact*) dan paling cocok untuk latihan. *Express Manager* standar, integrasi dengan CLR dan XML (Agus Tinus Setiawan dan Rika Putri Permadani :2016 : 54-55).