

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

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

Menurut (Jogiyanto, 2011) dalam buku Analisa dan Design Sistem Informasi Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis. “Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu. (Deppi Linda, 2016 : 62).

II.1.1. Karakteristik Sistem

Sistem mempunyai karakteristik atau sifat-sifat tertentu, diantaranya adalah sebagai berikut :

1. Komponen Sistem (*Components*)

Sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk suatu kesatuan. Komponen-komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

2. Batas Sistem (*Boundary*)

Batas sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batas sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem (*Environments*)

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga merugikan suatu sistem.

4. Penghubung Sistem (*Interface*)

Penghubung merupakan media penghubung antara suatu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari suatu subsistem ke subsistem lainnya.

5. Masukan Sistem (*Input*)

Masukan adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*).

6. Keluaran sistem (*Output*)

Keluaran adalah hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran dapat berupa masukan untuk subsistem yang lain.

7. Pengolah Sistem (*Process*)

Sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolah. Pengolah yang akan merubah masukan menjadi keluaran.

8. Sasaran Sistem (*Objectives*)

Sistem mempunyai tujuan atau sasaran yang akan menentukan masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem (Deppi Linda, 2016 : 62).

II.2. Informasi

Informasi adalah data yang berguna yang telah diolah sehingga dapat dijadikan dasar untuk mengambil keputusan yang tepat. Informasi sangat penting bagi organisasi. Pada dasarnya informasi adalah penting seperti sumber daya yang lain, misalnya peralatan, bahan, tenaga, dan sebagainya. Informasi yang berkualitas dapat mendukung keunggulan kompetitif suatu organisasi. Dalam sistem informasi akuntansi, kualitas dari informasi yang disediakan merupakan hal penting dalam kesuksesan sistem. Secara konseptual seluruh sistem organisasional mencapai tujuannya melalui proses alokasi sumberdaya, yang diwujudkan melalui proses pengambilan keputusan manajerial. Informasi memiliki nilai ekonomik yang mendukung keputusan alokasi sumberdaya, sehingga dengan demikian mendukung sistem untuk mencapai tujuan. (Agustinus Mujilan : 2012 : 1)

II.3. Sistem Informasi

Sistem informasi berbasis komputer merupakan sekelompok perangkat keras dan perangkat lunak yang dirancang untuk mengubah data menjadi informasi yang bermanfaat. Jenis sistem informasi berbasis komputer.

1. Pengolahan Data. Pengolahan data elektronik – *electronic data processing (EDP)* adalah pemanfaatan teknologi komputer untuk melakukan pengolahan data transaksi-transaksi dalam suatu organisasi. EDP adalah aplikasi sistem informasi kuantitas paling dasar dalam setiap organisasi. Sehubungan dengan perkembangan teknologi komputer, istilah pengolahan data mulai dikenal dan mempunyai arti yang sama dengan istilah EDP.
2. Sistem Informasi Manajemen (SIM), menguraikan penggunaan teknologi komputer untuk menyediakan informasi bagi pengambilan keputusan para manajer.
3. Sistem Pendukung Keputusan – *Decision Support Systems (DSS)*. DSS diarahkan untuk melayani permintaan informasi tertentu, khusus, dan tidak rutin dari manajemen.
4. Sistem Pakar – *expert systems (ES)* adalah sistem informasi berbasis pengetahuan yang memanfaatkan pengetahuannya tentang bidang aplikasi tertentu untuk bertindak seperti seorang konsultan ahli bagi pemakainya.
5. Sistem Informasi Eksekutif – *executive information systems (EIS)*. EIS dibuat bagi kebutuhan informasi strategis manajemen tingkat puncak.
6. Sistem Informasi Akuntansi – sistem berbasis komputer yang dirancang untuk mengubah data akuntansi menjadi informasi. (Agustinus Mujilan : 2012)

II.4. Akuntansi

Akuntansi berasal dari bahasa Inggris yaitu "*To account*" yang artinya menghitung atau mempertanggungjawabkan sesuatu yang ada kaitannya dengan pengelolaan bidang keuangan dari suatu perusahaan kepada pemiliknya atas kepercayaan yang telah diberikan kepada pengelola tersebut untuk menjalankan kegiatan perusahaan. Pengertian lain akuntansi merupakan kumpulan prosedur berupa kegiatan mencatat, mengikhtisarkan, mengklasifikasikan dan melaporkan keuangan dalam bentuk laporan keuangan dalam dalam periode waktu. Laporan keuangan yang dihasilkan harus dapat di pertanggungjawabkan kepada pihak-pihak yang berkepentingan. Akuntansi adalah proses dari transaksi yang dibuktikan dengan faktur, lalu dari transaksi dibuat jurnal, buku besar, neraca lajur kemudian akan menghasilkan Informasi dalam bentuk laporan keuangan yang digunakan pihak-pihak tertentu. (Wiratna Sujarweni : 2016 : 1)

II.5. Sistem Informasi Akuntansi

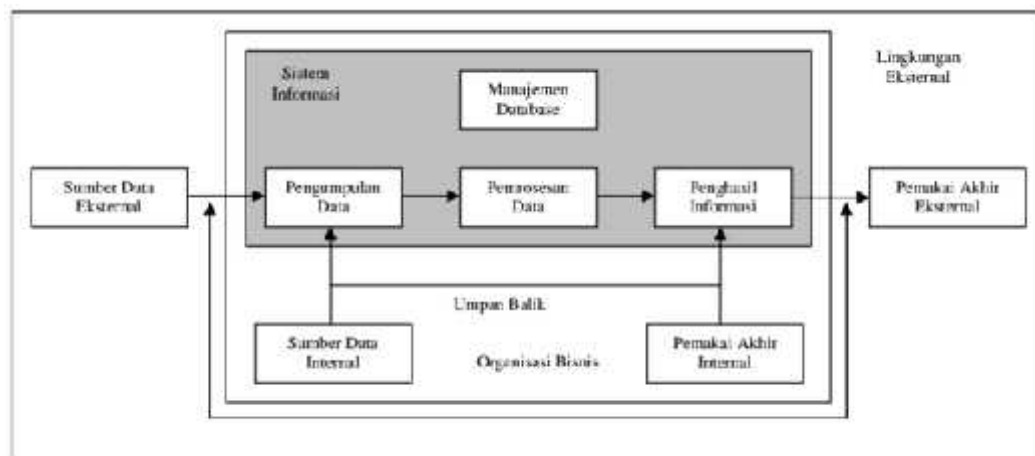
Sistem informasi akuntansi adalah kumpulan sumberdaya, seperti manusia dan peralatan, yang diatur untuk mengubah data menjadi informasi. Informasi ini dikomunikasikan kepada beragam pengambil keputusan. Sistem Informasi Akuntansi mewujudkan perubahan ini secara manual atau terkomputerisasi. Sistem Informasi Akuntansi juga merupakan sistem yang paling penting di organisasi dan merubah cara menangkap, memproses, menyimpan, dan mendistribusikan informasi. Saat ini, digital dan informasi *online* semakin digunakan dalam sistem informasi akuntansi. Organisasi perlu menempatkan

sistem di lini depan, dan mempertimbangkan baik segi sistem ataupun manusia sebagai factor yang terkait ketika mengatur Sistem Informasi Akuntansi. Sistem Informasi Akuntansi pada umumnya meliputi beberapa siklus pemrosesan transaksi :

1. Siklus pendapatan. Berkaitan dengan pendistribusian barang dan jasa ke entitas lain dan pengumpulan pembayaran-pembayaran yang berkaitan.
2. Siklus pengeluaran. Berkaitan dengan perolehan barang jasa dari entitas lain dan pelunasan kewajiban yang berkaitan.
3. Siklus produksi. Berkaitan dengan pengubahan sumber daya menjadi barang dan jasa.
4. Siklus keuangan. Kejadian-kejadian yang berkaitan dengan perolehan dan manajemen dana-dana modal, termasuk kas. (Agustinus Mujilan : 2012 : 3).

II.5.1. Arsitektur Sistem Informasi Akuntansi

Arsitektur Sistem Informasi Akuntansi merupakan kerangka model umum yang menggambarkan semua sistem yang digunakan dalam pembelajaran SIA dari mulai sumber data, proses pengumpulan, pemrosesan dan penyajian hasil informasi baik di lingkungan internal maupun eksternal perusahaan, yang dapat kami paparkan sebagai berikut :



Gambar II.1. Arsitektur Sistem Informasi Akuntansi
(Sumber : Ebook aris koerniawan ; 2012 : 8)

II.5.2. Kualitas Data Sistem Informasi Akuntansi

Seperti telah disebutkan di atas bahwa kualitas informasi merupakan hal yang sangat penting dalam sistem informasi (Xu:2009). Informasi yang baik akan tergantung pula dengan data yang baik. Untuk membentuk sistem yang mampu menyediakan data dan informasi yang baik dan dapat dipercaya membutuhkan suatu usaha yang tidak mudah.

(Xu:2009) menggaris bawahi hal penting dalam kaitannya dengan kualitas data.

1. Personil yang kompeten (competent personnel) agar sistem sesuai atau cocok digunakan.
2. Kontrol input (input control) merupakan hal bagian dari kontrol yang penting, apalagi dalam kasus transaksi *online*.
3. Jika perlu sediakan manajer kualitas data (DQ manager) yang bertanggung jawab pada kualitas data dalam Sistem Informasi Akuntansi.

Berikut ini merupakan gambaran tentang pentingnya kualitas data dalam sistem informasi akuntansi dari hasil wawancara yang dilakukan oleh Xu (2009) pada para manajer.

We have to monitor our cash balances fairly closely and it [data quality] is definitely one of the highest priorities. We have forecasts that need to be met, so we need to give ourselves early warning signals if a part of the business looks like it is not performing. The numbers will tell us that hopefully, so we can address the issue.

Kami harus memonitor saldo kas secara jelas dan kualitas data merupakan salah satu prioritas penting. Kami memperkirakan kebutuhan itu untuk dipenuhi, sehingga kami perlu mendapat tanda peringatan awal jika salah satu bagian dari bagian bisnis tidak berjalan sebagaimana mestinya. Suatu nomor aka memberitahukannya, jadi kita dapat mengenali permasalahannya. (Agustinus Mujilan, 2012:4).

II.5.3. Tujuan Sistem Informasi Akuntansi

Lingkup Sistem Informasi Akuntansi dapat dijelaskan dari manfaat sistem informasi akuntansi. Manfaat atau tujuan sistem informasi akuntansi adalah :

1. Mengamankan harta/ kekayaan perusahaan.

Harta kekayaan meliputi kas perusahaan, persediaan barang dagang, termasuk aset tetap perusahaan. Tidak ada pemilik yang senang dicuri orang (entah itu karyawan maupun orang asing).

2. Menghasilkan beragam informasi untuk menghasilkan keputusan.

Misal pengelola swalayan mengelola informasi mengenai barang apa saja yang diminati oleh konsumen. Membeli barang dagang yang kurang laku berarti kas akan terebak akan persediaan dan kehilangan kesempatan dalam membeli persediaan yang diinginkan oleh konsumen. (Agustinus Mujilan, 2012:4).

II.6. Persediaan

Persediaan barang merupakan banyaknya barang yang digunakan untuk memfasilitasi produksi atau untuk memuaskan permintaan konsumen. Persediaan barang meliputi barang yang dibeli dan disimpan untuk dijual kembali. Dalam hal ini, untuk memenuhi persediaan barang sebuah perusahaan dagang melakukan pembelian dan selanjutnya barang yang dibeli akan dijual kembali kepada konsumen. Persediaan berpengaruh terhadap neraca dan laporan Laba/ Rugi dalam neraca perusahaan dagang atau perusahaan Industri, persediaan seringkali merupakan bagian yang sangat besar dari keseluruhan Aktiva lancar yang dimiliki perusahaan ini merupakan bukti betapa pentingnya persediaan barang dagangan untuk kegiatan pembelian dan penjualan dalam operasi perusahaan semacam itu. Pencatatan persediaan yang efektif seringkali merupakan kunci keberhasilan operasi perusahaan disamping pimpinan perusahaan berusaha untuk mempertahankan kuantitas dan jenis persediaan yang cukup untuk memenuhi permintaan konsumen, pimpinan juga harus menjaga keseimbangan persediaan agar tidak terlalu tinggi dan juga tidak terlalu rendah.

Persediaan yang terlalu kecil akan menimbulkan kekecewaan konsumen sebaliknya persediaan yang terlalu tinggi akan menyebabkan biaya penyimpanan dan pemeliharaan persediaan akan membengkak . (Ruslan ; 2013 : 38).

II.7. Metode *Economic Order Quantity*

Economic Order Quantity (EOQ) adalah jumlah unit (kuantitas) barang yang dapat dibeli dengan biaya minimal. Tujuan metode persediaan ini adalah menentukan jumlah pesanan yang dapat meminimumkan biaya penyimpanan dan biaya pemesanan persediaan. Dengan menggunakan EOQ, maka persediaan yang ada di dalam gudang tidak terlalu banyak, tetapi juga tidak akan terlalu sedikit, sehingga aktivitas perusahaan tidak akan terganggu karenanya. Salah satu masalah dalam menentukan analisis EOQ adalah bahwa sulit bagi kita untuk dapat menentukan titik pemesanan kembali. Perlu diingat bahwa titik pemesanan kembali diperlukan untuk mencegah terjadinya kehabisan/kekurangan stok selama waktu antara melakukan pemesanan dan penerimaan pesanan tersebut. Titik pemesanan kembali adalah suatu tingkat persediaan yang tetap ada dalam stok yang jumlahnya sama dengan permintaan selama masa waktu yang dibutuhkan untuk menerima pesanan (lead time). Ketika permintaan bersifat pasti, persediaan ini akan berkurang/dihabiskan pada tingkat yang diketahui, sehingga pesanan akan sampai tepat pada saat tingkat persediaan mencapai titik nol. (Andy Wijaya : 2013)

$$EOQ = Q^* = \sqrt{\frac{2CR}{H}}$$

Keterangan:

Q^* = jumlah/nilai EOQ (unit).

C = biaya pemesanan per pesanan.

R = permintaan per periode (unit).

H = biaya penyimpanan.

II.8. Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basis data relasional. Pada dasarnya normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel relasional

1. Bentuk Normal Pertama (1 NF)

Contoh yang kita gunakan di sini adalah sebuah perancangan yang mendapatkan barang dari sejumlah pemasok. Masing-masing pemasok berada pada satu kota. Sebuah kota dapat mempunyai lebih dari satu pemasok dan masing-masing kota mempunyai kode status tersendiri. Masing-masing pemasok bisa menyediakan banyak barang. Tabel relasionalnya dapat dituliskan sebagai berikut :

PEMASOK (P#, Status, Kota, b#, qty) di mana

p# : kode pemasok (kunci utama)

status : kode status kota

Kota : nama kota

b# : barang yang dipasok

qty : jumlah barang yang dipasok.

Sebuah tabel relasional secara defenisi selalu berada dalam bentuk normal pertama. Semua nilai pada kolom-kolomnya adalah atomi. Ini berarti kolom-kolom tidak mempunyai nilai berulang. Tabel II.1. menunjukkan tabel pemasok dalam 1 NF.

Tabel II.1. Normalisasi Pertama Pemasok

P#	Status	Kota	B#	Qty
P1	20	Yogyakarta	B1	300
P1	20	Yogyakarta	B2	200
P1	20	Yogyakarta	B3	400
P1	20	Yogyakarta	B4	200
P1	20	Yogyakarta	B5	100
P1	20	Yogyakarta	B6	100
P2	10	Medan	B1	300
P2	10	Medan	B2	400
P3	10	Medan	B2	200
P4	20	Yogyakarta	B2	200
P4	20	Yogyakarta	B4	300
P4	20	Yogyakarta	B5	400

(Sumber : Janner Simarmata, dkk ; 2010:80)

2. Bentuk Normal Kedua (2 NF)

Defenisi bentuk normal kedua menyatakan bahwa tabel dengan kunci utama gabungan hanya dapat berada pada 1 NF, tetapi tidak pada 2 NF, sebuah tabel relasional berada pada bentuk normal kedua jika dia berada pada 1 NF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada kunci utama. Ini berarti bahwa setiap kolom bukan kunci harus tergantung pada seluruh kolom yang membentuk kunci utama. Tabel pemasok berada pada 1 NF, tetapi tidak pada 2 NF karena status dan kota tergantung secara fungsional hanya pada kolom p# dari

kunci gabungan (p#, b#). Ini dapat digambarkan dengan membuat daftar ketergantungan fungsional.

P# → Kota, Status

Kota → Status

(P#, B#) → qty

Proses mengubah tabel 1 NF ke 2 NF adalah :

- a. Tentukan sembarang kolom penentu selain kunci gabungan dan kolom-kolom yang ditentukannya.
- b. Buat dan beri nama tabel baru untuk masing-masing penentu dan kolom-kolom yang ditentukan.
- c. Pindahkan kolom-kolom yang ditentukan dari tabel asal ke tabel baru penentu akan menjadi kunci utama pada tabel baru
- d. Hapus kolom yang baru dipindahkan dari tabel asal, kecuali penentu yang akan berfungsi sebagai kunci tamu.
- e. Tabel asal bisa diberi nama baru.

Pada contoh, kita memindahkan kolom p#, status, dan kota ke tabel baru yang disebut pemasok2. Kolom p# menjadi kunci utama tabel ini. Tabel

II.3. menunjukkan hasilnya.

Tabel II.2. Tabel Bentuk Normal Kedua

Pemasok2			Barang		
P#	Status	Kota	P#	B#	Qty
P1	20	Yogyakarta	P1	B1	300
P2	10	Medan	P1	B2	200
P3	10	Medan	P1	B3	400
P4	20	Yogyakarta	P1	B4	200
P5	30	Bandung	P1	B5	100
			P1	B6	100
			P2	B1	300
			P2	B2	400
			P3	B2	200
			P4	B2	200
			P4	B4	300
			P4	B5	400

(Sumber : Janner Simarmata, dkk ; 2010:82)

3. Bentuk Normal Ketiga (3 NF)

Bentuk normal ketiga mengharuskan semua kolom pada tabel relasional hanya pada kunci utama. Secara defenisi, sebuah tabel berada pada bentuk normal ketiga (3 NF) jika tabel sudah berada pada 2 NF dan setiap kolom yang bukan kunci tidak tergantung secara transitif pada kunci utamanya. Dengan kata lain, semua atribut bukan kunci tergantung secara fungsional hanya pada kunci utama. Tabel barang sudah dalam bentuk normal ketiga. Kolom bukan kunci, qty, tergantung sepenuhnya pada kunci utama (p#, b#). Pemasok masih berada pada 2 NF, tetapi belum berada pada 3 NF karena dia mengandung ketergantungan transitif. Ketergantungan transitif terjadi ketika sebuah kolom bukan kunci, yang ditentukan oleh kunci utama, menentukan kolom lainnya. Konsep ketergantungan

transistif dapat digambarkan dengan menunjukkan ketergantungan fungsional pada pemasok2, yaitu :

Pemasok2. p# \longrightarrow Pemasok2, status

Pemasok2. p# \longrightarrow Pemasok2, kota

Pemasok2. kota \longrightarrow Pemasok2, status

Perlu dicatat bahwa pemasok2, status ditentukan, baik oleh kunci utama p#, maupun kolom bukan kunci, kota

Proses mengubah tabel menjadi 3 NF adalah :

- a. Tentukan semua penentu selain kunci utama dan kolom yang ditentukannya.
- b. Buat dan beri nama tabel baru untuk masing-masing penentu dan kolom yang ditentukannya.
- c. Pindahkan kolom yang ditentukan dari tabel asal ke tabel baru. Penentu menjadi kunci utama tabel baru.
- d. Hapus kolom yang baru saja dipindahkan dari tabel asal, kecuali penentu yang akan berfungsi sebagai kunci tamu.
- e. Tabel asal bisa diberi nama baru.

Untuk mengubah PEMASOK2 menjadi 3 NF, kita membuat tabel baru yang disebut KOTA_STATUS dan memindahkan kolom kota dan status ke tabel baru. Status dihapus dari tabel diberi nama baru PEMASOK_KOTA. Tabel II.4 menunjukkan hasilnya.

Tabel II.3. Tabel Bentuk Normal Ketiga

PEMASOK_KOTA

P#	Kota
P1	Yogyakarta
P2	Medan
P3	Medan
P4	Yogyakarta
P5	Bandung

KOTA_STATUS

Kota	Status
Yogyakarta	20
Medan	10
Bandung	30
Semarang	40

(Sumber : Janner Simarmata, dkk ; 2010:83)

4. Bentuk Normal Boyce Code (BCNF)

Setelah 3 NF, semua masalah normalisasi hanya melibatkan tabel yang mempunyai tiga kolom atau lebih dan semua kolom adalah kunci. Banyak praktisi berpendapat bahwa menempatkan entitas pada 3 NF sudah cukup karena sangat jarang entitas yang berada pada 3 NF bukan merupakan 4 NF dan 5 NF. Lebih lanjut, mereka berpendapat bahwa keuntungan yang didapat mengubah entitas ke 4 NF dan 5 NF sangat kecil sehingga tidak perlu dikerjakan. Bentuk Normal Boyce- Code (BCNF) adalah versi 3 NF lebih teliti dan berhubungan dengan tabel relasional yang mempunyai (a) banyak kunci kandidat (b) kunci kandidat gabungan, dan (c) kunci kandidat yang saling tumpang tindih. BCNF didasarkan pada konsep penentu. Sebuah kolom penentu adalah kolom di mana kolom-kolom lain sepenuhnya tergantung secara fungsional. Sebuah tabel relasional berada pada BCNF jika dan hanya setiap penentu adalah kunci kandidat

5. Bentuk Normal Keempat (4 NF)

Sebuah tabel relasional berada pada bentuk normal keempat (4 NF) jika dia dalam BCNF dan semua ketergantungan multivalued merupakan ketergantungan

fungsional. Bentuk normal keempat (4 NF) didasarkan pada konsep ketergantungan multivalued (MVD). Sebuah ketergantungan multivalued terjadi ketika dalam sebuah tabel relasional yang mengandung setidaknya tiga kolom, satu kolom mempunyai banyak baris bernilai sama, tetapi kolom lain bernilai berbeda. Definisi secara formal diberikan oleh C.J. Date, yaitu :

Misalnya, ada sebuah tabel relasional R dengan kolom A, B dan C, Maka R.A \longrightarrow R.B (kolom A menentukan kolom B). Adalah benar jika dan hanya jika himpunan nilai B yang cocok dengan pasangan nilai A dan nilai C pada R hanya tergantung pada nilai A dan tidak tergantung pada nilai C. MVD selalu terjadi dalam pasangan, yaitu R.A \longrightarrow R.B dipenuhi jika dan hanya jika R.A R.C dipenuhi pula.

6. Bentuk Normal Kelima (5 NF)

Sebuah tabel berada pada bentuk normal kelima jika dia tidak dapat mempunyai dekomposisi lossless menjadi sejumlah tabel lebih kecil. Empat bentuk normal pertama berdasarkan pada konsep ketergantungan fungsional, sedangkan bentuk normal kelima berdasarkan pada konsep ketergantungan gabungan (*join dependence*). Ketergantungan gabungan berarti sebuah tabel, setelah deskomposisi menjadi tiga atau lebih tabel yang lebih kecil, harus dapat digabungkan kembali untuk membentuk tabel asal. Dengan kata lain 5 NF menunjukkan ketika sebuah tabel tidak dapat dideskomposisi lagi (Janner Simarmata : 2012:86).

II.9. Basis Data (*Database*)

Secara sederhana database (basis data/ pangkalan data) dapat diungkapkan sebagai suatu pengorganisasian data dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah dan cepat (Kadir, 2004). Pengertian akses dapat mencakup pemerolehan data maupun pemanipulasian data seperti menambah serta menghapus data. Dengan memanfaatkan komputer, data dapat disimpan dalam media pengingat yang disebut *hard disk*. Dengan menggunakan media ini, keperluan kertas untuk menyimpan data dapat dikurangi. Selain itu, data menjadi lebih cepat untuk diakses terutama jika dikemas dalam bentuk database.

Pengaplikasian database dapat kita lihat dan rasakan dalam keseharian kita. Database ini menjadi penting untuk mengelola data dari berbagai kegiatan. Misalnya, kita bisa menggunakan mesin ATM (*anjungan tunai mandiri/ automatic teller machine*) bank karena bank telah mempunyai database tentang nasabah dan rekening nasabah. Kemudian data tersebut dapat diakses melalui mesin ATM ketika bertransaksi melalui ATM. Pada saat melakukan transaksi, dalam konteks database sebenarnya kita sudah melakukan perubahan (*update*) data pada database di bank.

Ketika kita menyimpan alamat dan nomor telepon di HP, sebenarnya juga telah menggunakan konsep database. Data yang kita simpan di HP juga mempunyai struktur yang diisi melalui formulir (*form*) yang disediakan. Pengguna dimungkinkan menambahkan nomor HP, nama pemegang, bahkan kemudian dapat ditambah dengan alamat *email*, alamat *web*, nama kantor, dan sebagainya.

Pemahaman tentang database ini dapat didekatkan pada konsep akuntansi. Kita bisa umpamakan bahwa ketika kita melakukan proses akuntansi secara manual, kita menuliskan suatu catatan ke dalam lajur dan kolom buku. Mulai dari jurnal, buku besar, buku pembantu kita memasukkan catatan satu demi satu. Melihat buku akuntansi tersebut, sebenarnya kita sudah melihat konsep database, yang jika dikelola dengan komputer masih diperlukan penyesuaian dalam membentuk kolom-kolomnya. (Agustinus Mujilan : 2012:23)

II.10. Visual Studio 2010

Visual Studio 2010 adalah inkarnasi dari bahasa visual basic yang sangat populer dan telah dilengkapi dengan fitur serta fungsi yang setara dengan bahasa tingkat lainnya seperti C++. Anda dapat menggunakan visual basic 2010 untuk membuat aplikasi windows, mobile, web, dan office atau kode yang telah ditulis oleh orang lain dan kemudian dimasukkan ke program lainnya. Visual basic menyediakan berbagai tools dan fitur canggih yang memungkinkan dapat menulis kode, menguji dan menjalankan program tunggal atau terkadang serangkaian program yang terkait dengan satu aplikasi. (Christopher Lee:2014:1)

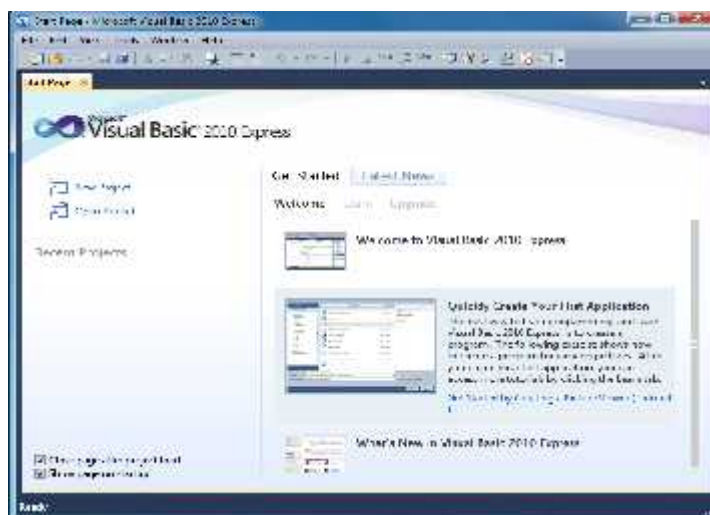
Tabel II.4. Jendela Aplikasi Visual Studio 2010

No	Bagian	Keterangan
1	Title Bar	Menampilkan nama aplikasi yang sedang terbuka.
2	Mneu Bar	Menampilkan daftar perintah yang memungkinkan anda dapat menulis, mengedit, menyimpan, mencetak, menguji dan menjalankan program visual basic.
3	Standard Toollbar	Berisi Tombol yang menjalankan perintah yang sering digunakan seperti open project, new project, save, cut, copy, paste, dan undo
4	Toolbox	Berisi komponen NET yang dapat anda gunakan

		untuk mengembangkan antarmuka pengguna grafis untuk program visual studio 2010.
5	Area Kerja Utama	Menampilkan item yang sedang di kerjakan
6	Solution Explorer	Menampilkan elemen dari visual basic solution, yaitu nama yang ddiberikan kepada program visual basic dan item lainnya yang dihasilkan oleh visual basic 2010 sehingga program akan mengeksekusi dengan benar.
7	Properties Window	Setiap Objek dalam program visual basic memiliki seperangkat karakteristik yang disebut sifat-sifat objek.

(Sumber : Christopher Lee ; 2014 : 2)

Untuk melihat tampilan visual basic 2010 dapat dilihat pada gambar II.2. sebagai berikut :



Gambar II.2. Tampilan Utama Visual Basic 2010
(Sumber : Christopher Lee ; 2014 ; 2)

II.11. SQL Server 2008

SQL (*Structured Query Language*) adalah bahasa non procedural untuk mengakses data pada database relasional. SQL adalah bahasa database yang dipergunakan dalam menyelesaikan permasalahan dalam database serta mempunyai kelebihan dalam mengolah data. Standar SQL mula-mula

didefinisikan oleh ISO (*International Standards Organization*) dan ANSI (*the American National Standards Institute*) yang dikenal dengan sebutan SQL86. (Eka Iswandy : 2015 : 73)

Dengan menggunakan SQL, kita dapat melakukan hal-hal berikut:

1. Memodifikasi struktur database .
2. Mengubah, mengisi, menghapus isi database.
3. Mentransfer data antara database yang berbeda. SQL ada yang dikembangkan untuk PC dan ada juga yang dikembangkan untuk dapat mengakomodasi database yang sangat besar.

Beberapa contohnya antara lain:

1. *Microsoft Access*

Digunakan untuk PC, sangat mudah dipakai dimana perintah SQL dapat langsung dimasukkan atau melalui fasilitas yang telah digunakan.

2. *Microsoft Query*

SQL yang dipaket dengan produk lain dari Microsoft Windows, yaitu Microsoft Visual Studio seperti Visual Basic dan Visual C++. Untuk terhubung dengan database lain menggunakan ODBC.

3. *Oracle*

Digunakan untuk perusahaan yang menggunakan database besar.

II.12. *Unified Modelling Language (UML)*

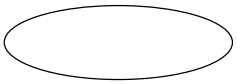
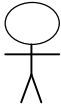
Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language (UML)*. *Unified Modeling Language (UML)* adalah bahasa



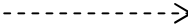

spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. *Unified Modeling Language* (UML) merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. *Unified Modeling Language* (UML) saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem. (Windu Gata : 2013)

1. *Diagram Use Case (Use Case Diagram)*

Use case diagram 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. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram, yaitu :

Tabel II.5. Simbol *Use Case Diagram*

Gambar	Keterangan
	<p>Use case menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama use case.</p>
	<p>Aktor adalah <i>abstraction</i> dari orang atau 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. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan use case, tetapi tidak memiliki control terhadap use case.</p>




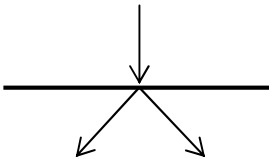
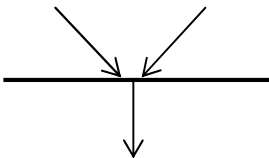
	Asosiasi antara aktor dan use case, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara aktor dan use case yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam use case lain (<i>required</i>) atau pemanggilan use case oleh use case lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari use case lain jika kondisi atau syarat terpenuhi.

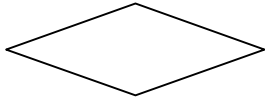
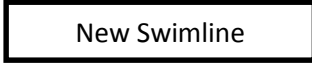
(Sumber : Windu Gata ; 2013:5)

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram*, yaitu :

Tabel II.6. Simbol Diagram Aktivitas

Gambar	Keterangan
	<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) atau rake, digunakan untuk menunjukkan adanya dekomposisi.

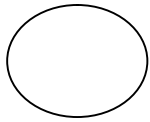
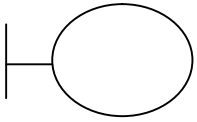
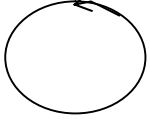
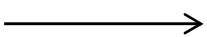
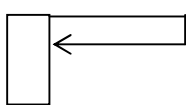

	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.


(Sumber : Windu Gata ; 2014:6)

3. Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :

Tabel II.7. Simbol Sequence Diagram

Gambar	Keterangan
	<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> , <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 : Windu Gata ; 2013:7)

4. Class Diagram (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

Class diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan.

Class diagram secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

Tabel II.8. Simbol 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 : Windu Gata ; 2013:9)