BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Demi kesempurnaan penelitian, maka penulis perlu melakukan perbandingan untuk mengetahui gambaran dari penelitian terdahulu. Menurut Venny Karamoy (2013 : 3), dalam jurnalnya yang berjudul "Analisis Piutang pada PT. Sucofindo (Persero) Cabang Manado", Ditengah persaingan bsinis yang ketat, perusahaan dituntut untuk mampu menguasai pasar, sehingga perusahaan perlu melakukan strategi penjualan secara kredit agar jumlah penjualan meningkat. Secara umum, piutang timbul karena adanya transaksi penjualan barang atau jasa secara kredit. Investasi yang terlalu besar dalam piutang bisa menimbulkan lambatnya perputaran modal kerja, sehingga semakin kecil pula kemampuan perusahaan dalam meningkatkan volume penjualan. Akibatnya semakin kecil kesempatan yang dimiliki perusahaan untuk menghasilkan keuntungan atau laba. Tujuan yang diharapkan dapat dicapai dari penelitian ini adalah untuk menganalisis piutang tak tertagih yang dilakukan oleh PT. SUCOFINDO (Persero) Cabang Manado. Salah satu pemanfaatan dari sistem ini adalah menggunakan pencatatan piutang dapat diterapkan lebik baik lagi.

Menurut Imanuella Fensi da (2015 : 699), dalam jurnalnya yang berjudul "Analisis Kerugian Piutang Tak Tertagih Pada PT. Metta Karuna Jaya Makassar", Penelitian ini akan dilakukan Dengan Tujuan

Tujuan dari penelitian ini adalah untuk (1) mengetahui beban penyisihan piutang dan laba operasional PD Putra Madani Ciamis (2) mengetahui pengaruh antara beban penyisihan piutang tak tertagih terhadap laba operasional pada PD. Putra Madani Ciamis. Metode Penelitian yang dipergunakan dalam penelitian ini adalah deskriptif dengan pendekatan studi kasus. Hasil penelitian yang diperoleh dari penelitian ini adalah terdapat pengaruh antara beban penyisihan piutang tidak tertagih terhadap laba operasional.

II.2. Landasan Teori

II.2.1. Definisi 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.2.2. 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 menunjukan 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 dimasukan 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.2.3. 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 pada saat ia mendukung keputusan alokasi sumberdaya, sehingga dengan demikian mendukung sistem untuk mencapai tujuan. (Mujilan : 2012 : 1)

II.2.4. 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 informasia kuntansi 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 stratejik manajemen tingkat puncak.
6. Sistem Informasi Akuntansi – sistem berbasis komputer yang dirancang untuk mengubah data akuntansi menjadi informasi. (Agustinus Mujilan : 2012)

II.2.5. 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. (Mujilan : 2012 : 3).

II.2.6. Piutang

Piutang merupakan pos yang penting bagi kebanyakan perusahaan, karena merupakan bagian aktiva lancar perusahaan dan cukup berperan dalam laporan keuangan perusahaan. Kurangnya pemahaman dan pengendalian piutang akan mengakibatkan kerugian yang cukup besar. Oleh karena itu diperlukan sistem informasi, pengendalian yang memadai, dan didukung sumber daya manusia yang potensial, akan menghindarkan perusahaan dari kerugian sehingga tujuan perusahaan akan tercapai sesuai rencana. Semakin besar volume penjualan kredit semakin besar juga resiko tidak tertagihnya piutang tersebut. Kemampuan piutang untuk dapat dikonversikan kedalam uang tunai dikenal dengan kolektibilitas atau penagihan piutang. Ada beberapa kendala dalam penagihan piutang, baik intern maupun ekstern. Faktor intern berasalnya dari pemeriksaan intern penjualan kredit

yang kurang baik. sedangkan faktor ekstern dapat disebabkan oleh keadaan pelanggan. (Ali Nurdin Siregar : 2016)

II.3. Piutang Dicatat Kotor (Gross Method)

Metode kotor mengakui jumlah piutang sebesar penjualan tanpa dikurangi dengan potongan yang akan diberikan. Apabila ternyata debitur mengambil potongan maka akan diakui sebagai pengurang jumlah penjualan bukan pengurangjumlah piutang. Dengan metode ini Prosedur penjurnalan dan pembukuannya adalah sebagai berikut :

1. Pada saat terjadi penjualan kredit barang dagangan
2. Pada saat diterima pelunasan piutang.
    a. Bila pelunasan piutang dagang telah melebihi masa potongan, yaitu lebih 10 hari maka perlu diperhitungkan potongan dan perusahaan akan menerima seluruh piutang.
    b. Bila pelunasan piutang masih dalam batas masa potongan, maka perlu diperhitungkan dan memberi potongan penjualan, yaitu sebesar 2 % dari piutang dan perusahaan akan menerima uang sebesar 98%. (Leno dan Sambodo, 2013).

II.4. Normalisasi

Agar model database relasional bisa digunakan secara efektif, maka pengelompokkan yang rumit harus disederhanakan sehingga data berlebih dan

relasi yang salah bisa dieliminasi. Proses membuat struktur data yang lebih kecil dan stabil dari sekelompok data yang rumit disebut Normalisasi.

Normalisasi merupakan sebuah teknik dalam logical desain sebuah basis data / database, teknik pengelompokkan atribut dari suatu relasi sehingga membentuk struktur relasi yang baik tanpa redudansi. Tujuan normalisasi adalah mengorganisasikan data kedalam tabel-tabel untuk memenuhi kebutuhan pemakai, menghilangkan kerangkapan data, mengurangi kompleksitas, mempermudah modifikasi data. (Mukhlisulfatih Latief : 2016)

1. Proses Normalisasi
   a. Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu kebeberapa tingkat.
   b. Apabila tabel yang diuji belum memenuhi persyaratan tertentu maka tabel tersebut pelu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk yang optimal.
2. Tahapan Normalisasi :
   1) Bentuk tidak normal : Menghilangkan perulangan grup.

Tabel II.1. Contoh bentuk tidak normal (Unnormal)

No-Mhs	Nama Mhs	Jurusan	Kode-MK	Nama-MK	Kode Dosen	Nama Dosen	Nilai
2683	Welli	MI	M1350	Manajemen DB	B104	Ati	A
5432	Bakli	Ak.	M1465 M1350 Akn201	Analisis Perc.Sistim Manajemen DB Akuntansi	B317 B104 D310	Dita Ati Lia	B C B
				Keuangan			
			MKT300	Dasar Pemasaran	B212	Dita	A

Sumber : Mukhlisulfatih Latief : 2016

2) Bentuk Normal pertama (1NF) : Menghilangkan ketergantungan sebagian.

Yaitu : suatu relasi dikatakan sudah memenuhi bentuk normal kesatu bila setiap data bersifat atomik yaitu setiap irisan baris dan kolom hanya mempunyai satu nilai data.

Tabel II.2. Contoh Bentuk Normal Pertama (1NF)

No-Mhs	Nama Mhs	Jurusan	Kode MK	Nama MK	Kode Dosen	Nama Dosen	Nilai
2683	Welli	MI	M1350	Manajemen DB	B104	Ati	A
2600	Welli	MI	M1465	Analisis Perc.Sistim	B317	Dita	B
5432	Bakti	Ak.	M1350	Manajemen DB	B104	Ati	C
5432	Bakti	Ak.	Akn201	Akuntansi	D310	Lia	B
5432	Bakti	Ak.		Keuangan			
5432	Bakti	Ak.	MKT300	Dasar Pemasaran	B212	Lola	A

Sumber : Mukhlisulfatih Latief : 2016

3) Bentuk Normal kedua (2NF) : Menghilangkan ketergantungan transitif.

Yaitu : suatu relasi dikatakan sudah memenuhi bentuk normal kedua bila relasi tersebut sudah memenuhi bentuk normal kesatu dan atribut yang bukan key sudah tergantung penuh terhadap key-nya.

Tabel II.3. Contoh Bentuk Normal Kedua (2NF)

Kode-MK	Nama-MK	Kode Dosen	Nama Dosen
M1350	Manajemen DB	B104	Ati
M1465	Analisis Perc.Sistim	B317	Dita
M1350	Manajemen DB	B104	Ati
Akn201	Akuntansi	D310	Lia
	Keuangan		
MKT300	Dasar Pemasaran	B212	Lola

Sumber : Mukhlisulfatih Latief : 2016

4) Bentuk Normal ketiga (3NF) : Menghilangkan anomali-anomali hasil dari ketergantungan fungsional. Yaitu : suatu relasi dikatakan sudah memenuhi bentuk normal ketiga bila relasi tersebut sudah memenuhi bentuk normal kedua dan atribut yang bukan key tidak tergantung transitif terhadap key-nya.

Tabel II.4. Contoh Tabel Mahasiswa Dan Tabel Kuliah (3NF)

No-Mhs	Nama Mhs	Jurusan
2683	Welli	MI
5432	Bakti	Ak.

Sumber : Mukhlisulfatih Latief : 2016

II.5. 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.5.1. Model Database

Model database yang saat ini banyak digunakan adalah model database relational. Imam (2008) menyebutkan “Model database ini disusun dalam bentuk tabel dua dimensi yang terdiri dari baris (*record*) dan (*field*), pertemuan antara baris dengan kolom disebut item data (*data value*). Tabel-tabel yang ada

dihubungkan (*relationship*) sedemikian rupa menggunakan *field-field* kunci (*key field*) sehingga dapat meminimalkan duplikasi data."

Model database relational ini dapat kita kenal konsepnya mulai dari yang paling sederhana misalnya dengan penerapan program aplikasi *excel*. Meskipun untuk pengelolaan database secara luas *excel* jarang digunakan dan kurang mencukupi, namun untuk melihat konsep database dan konsep membangunnya program ini dapat dimanfaatkan. *Excel* mempunyai baris yang disebut *raw* dan mempunyai kolom. Kemudian item data merupakan sel atau pertemuan antara baris dan kolom. Tabel-tabel dapat diumpakan apabila kita menggunakan tabel dalam suatu *sheet* tertentu. Data dari berbagai tabel dapat diambil dari tabel lain menggunakan perintah *look up* yang berdasarkan kode kunci tertentu. Kode kunci tersebut berada pada suatu kolom tertentu, yang dalam konsep database relational disebut sebagai *key field* tadi. (Agustinus Mujilan : 2012:24)

II..5.2. Struktur *Database*

Untuk memahami konteks *database* kita perlu memahami istilah dan hal-hal yang terkait dengan database. Dalam berbagai program aplikasi database terdapat kesamaan ataupun sedikit perbedaan di dalamnya. Seseorang yang mempelajari database dengan program aplikasi tertentu harus memperhatikan struktur dan karakteristik sesuai dengan bahasa dalam aplikasi tersebut. Namun demikian, secara umum terdapat karakteristik sebagai berikut:

1. Nama *file*

Nama *file* adalah nama yang digunakan untuk mengidentifikasi adanya data yang disimpan dalam komputer dan digunakan untuk pemanggilan data. *File* yang dikelola akan muncul dalam komputer dengan ekstensi sesuai dengan program aplikasinya. *File* tersebut dapat digunakan untuk menandakan adanya *file* database, ataupun *file table*. Database dan table akan saling terkait, meskipun cara menyimpan dalam komputer akan mengalami sedikit perbedaan pada beberapa aplikasi. Misalnya Ms. *Access* akan menyimpan dengan *file* yang dapat kita lihat adalah *file* databasenya. Di program *MySql* nama database ini akan menjadi *folder*. Sementara di *FoxPro* nama database dapat menjadi *file* tersendiri. *Table* cara menyimpannya juga berbeda, dalam *Ms. Access* mungkin kita tidak melihat nama *table* secara kasat mata karena akan dikelola di dalam *file* database. Di dalam *MySql* kita bisa melihat beberapa nama *file* terkait dengan pengelolaan *table*. Dan di dalam *FoxPro table* ini dapat menjadi nama *file* terpisah dan dapat dikenali pula sebagai *free table*.

2. Database

Database sebenarnya merupakan nama untuk menampung berbagai *table* di dalamnya. Konsep ini akan sama dalam berbagai program aplikasi. Misalnya kita membangun database akuntansi dengan nama database "*akun_base*". Di dalam *akun_base* akan diorganisasi berbagai table yang terkait dengan kegiatan akuntansi misalnya tabel: rekening, pelanggan, jurnal, buku induk, dan administrator program, dan sebagainya. Setiap data yang masuk tidaklah dicatat dalam database, namun di dalam masing-masing *table* yang sesuai.

3. *Table*

Table merupakan tempat untuk menyimpan data sesuai dengan kelompok data. Setiap isi table mengandung data yang mempunyai karakteristik dalam penggunaannya. Untuk mempermudah pengolahan biasanya pembangun database mengkategorikan *table* sesuai dengan data isinya sebagai berikut :

a. *Master table*

Master table berisi data tentang hal-hal utama dalam kegiatan database. Table ini berisi record yang relatif permanen atau seringkali menjadi acuan ketika mengoperasikan transaksi. Dalam master tabel identitas record menjadi penting dan diusahakan merupakan data atau kode yang bersifat unik. Unik dapat diartikan bahwa tidak ada dalam satu table berisi kode yang sama. Disain kode menjadi penting di sini. Misalnya dalam mendisain nama akun dalam database akuntansi, maka kode akun menjadi sangat penting artinya. Dalam *table* berisi nama barang, maka kode barang menjadi hal penting. Contoh lain dalam database akademik, tabel *master* dapat berupa : mahasiswa, daftar dosen, daftar kurikulum.

b. *Transaction table*

Tabel transaksi digunakan untuk menyimpan data dalam menjalankan suatu kegiatan atau bisnis. Data ini seringkali akan bertambah dalam kesehariannya ketika terjadi transaksi yang sesuai dengannya. Secara lebih mudah dapat dipahami dalam akuntansi seringkali mencatat transaksi dalam jurnal. Terkait hal tersebut, transaksi ini dicatat dalam tabel jurnal. Dalam

mencatat transaksi ini, kita harus menyesuaikan kode data tertentu dengan kode yang terdapat dalam *master table.*

c. *Tabulation table*

Tabulasi data dapat digunakan untuk menyimpan data seperti halnya master data namun bersifat sebagai data pembantu ketika menginput formulir baik untuk data master maupun transaksi. Misalnya untuk memetakan keterangan hobi, jenis kelamin, nama golongan, nama level manajemen, dan sebagainya. Dengan konsep penamaan field yang baik mungkin saja table tabulasi ini dapat digunakan untuk memuat berbagai kelompok data. Misalnya fieldnya berupa kode dan keterangan. Contoh kelompok gender dengan L = laki-laki; P = perempuan. Kelompok level dengan M = Manajer, O = operator, S = seller

d. *Temporary table*

Temporary adalah data sementara yang digunakan untuk membantu ketika terjadi proses transaksi. Data ini dapat saja langsung dihapus ketika transaksi selesai terproses. Misalnya digunakan untuk mempermudah perhitungan, penyimpanan data sementara sebelum diproses setuju ke database. Misalnya: ketika terjadi transaksi di depan kasir, data-data pertama akan ditangkap dan dimasukan dalam file temporary sebelum akhirnya kasir melakukan perintah “ok” yang menandakan data transaksi siap untuk disimpan atau diproses dalam komputer. Ketika masa tunggu ini, data masih dapat diedit, dibatalkan, ataupun ditambah. Sementara ketika sudah masuk ke sistem, edit atau penambahan akan membutuhkan prosedur tertentu.

Seandainya dianalogikan dengan sistem akuntansi maka proses edit data yang telah masuk ke sistem dapat digunakan prosedur seperti halnya melakukan jurnal koreksi.

4. *Field*

Field adalah penanda untuk kolom data. Jika dalam *excel* penanda tersebut adalah kolom A, B, dan seterusnya, sementara dalam konsep *table* dalam database maka nama *field* memegang peranan penting. Dalam konsep table dalam database, ketika memanggil dengan nama *field* tertentu maka data-data di dalamnya akan muncul. Pengolahan dapat dilakukan dengan membuat *filter*, misalnya berdasarkan kode tertentu, berdasarkan *record* tertentu. Misalnya kita ingin memanggil record terkait nama karyawan Andi. Dalam database Andi ini diberi ID: 11001. Sehingga kita bisa menggunakan konsep filtrasi untuk memanggil personalia dengan kode ID 11001. Apabila data ketemu, maka kita dapat menggunakan data berdasarkan *field-field* yang ada, misalnya nama, tempat lahir, tanggal lahir, alamat, dan sebagainya yang mengacu pada ID 11001.

Dalam mengatur setting *field*, biasanya akan terkait hal-hal sebagai berikut :

a. *Field type* : tipe *field* ini dapat terkait apakah *field* tersebut akan berisi data berupa *key field* (*primary key*, *secondary key*), atau *descriptor*. *Primary key* akan berisi ID atau kode pokok yang akan digunakan dalam mengidentifikasi *record*, sehingga data di dalam *field* tersebut tidak diijinkan untuk memiliki lebih dari satu data yang sama. *Secondary* adalah subset dari *key* utama. Misalnya saja kode mata kuliah dalam satu semester tidak boleh terdapat lebih dari satu pada ID atau NIM yang sama. *Descriptor* adalah *field*

berisi data yang akan merupakan satu kesatuan dengan yang lainnya sebagai penjelasan akan adanya *record* atau ID tertentu.

b. Data *type*: tipe data merupakan jenis data yang dapat dimasukkan dalam *field*. Hal ini dapat dibagi secara umum sebagai karakter/ *text*, numerik, tanggal dan sebagainya.

c. *Field Size*. Penting untuk memahami ukuran field yang akan digunakan dalam menampung data. Dalam pengembangan sistem harus dapat memperkirakan berapa lebar ukuran *field* yang efektif. Apabila terlalu lebar akan terjadi banyak spasi kosong dan berpengaruh pada ukuran *file* yang disimpan. Sementara apabila terlalu sempit akan terdapat data yang tidak tersimpan. Misalnya ketika kita menghitung bahwa nama

menggunakan ukuran 40 karakter sudah memenuhi untuk *field* data kita. Konsekuensinya, apabila terdapat nama di atas 40 karakter maka akan terpotong menjadi 40 karakter. Konsekuensi lain adalah nama tersebut diinput hingga memuat maksimal 40 karakter yaitu dengan mengadakan singkatan nama.

5. *Records*

Records merupakan baris data. Karena satu baris data biasanya mengindikasikan satu kesatuan data tertentu, maka satu *record* ada yang menyebut satu data. Misalnya keterangan mengenai biodata Andi disimpan dalam satu record beridentitas ID 11001, maka untuk menyebutkan satu kesatuan data seputar Andi dalam baris tertentu ada yang menyebutkan sebagai *record* data

Andi. Dalam konsep database masing-masing *record* memiliki nomor identitas tersendiri baik itu identitas yang diberikan komputer

ataupun yang diinputkan secara manual. Sehingga dalam konteks tertentu dapat digunakan konsep nomor *record*, ID otomatis, ID *primary key*. (Agustinus Mujilan : 2012:31)

II.6. Visual Studio 2010

Visual Basic 2010 adalah inkarnasi dari bahasa visual basic yang sangat populer dan telah dilengkapi dengan fitur serta fungsi yang setara dengan bahasa tingkat lainnya sseperti 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 ataau terkadang serangkaian program yang terkait dengan satu aplikasi. (Christopher Lee:2014:1)

Tabel II.5. 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 sepertin 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 bassic solution, yaitu nama yang ddiberikan kepada program visual bassic dan

		item lainnya yang dihasilkan oleh visual basic 2010 sehingga program akan mengeksekusi dengan benar.
7	Properties Window	Setiap Objek dalm 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.1. sebagai berikut :

Gambar II.1. Tampilan Utama Visual Basic 2010
(Sumber : Christopher Lee ; 2014 ; 2)

II.7. 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 didefenisikan 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 Accsess*

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 mengggunakan database besar.

II.8. *UML* (*Unified Modelling Language*)

Menurut Windu Gata (2013) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (*UML*). *UML*

adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

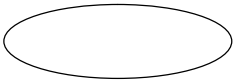
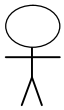
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. *UML* saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem. (Gellysa Urva dan Helmi Fauzi Siregar, 2015, : 93).

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan *UML* adalah sebagai berikut:

1. *Use case* Diagram

Use case diagram merupakan pemodelan untuk kelakukan (*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 dapat dilihat pada tabel II.6 dibawah ini :

Tabel II.6. Simbol *Use Case*

Gambar	Keterangan
	*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*.
	Aktor adalah *abstraction* dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasikan aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan

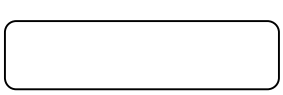
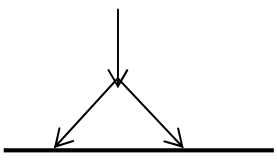
	dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan *use case*, tetapi tidak memiliki control terhadap *use case*.
	Asosiasi antara aktor dan *use case*, digambarkan dengan garis tanpa panah yang mengindikasikan siapa 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.
	Include, merupakan di dalam *use case* lain (*required*) atau pemanggilan *use case* oleh *use case* lain, contohnya adalah pemanggilan sebuah fungsi program.
	Extend, merupakan perluasan dari *use case* lain jika kondisi atau syarat terpenuhi.

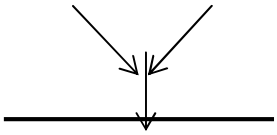
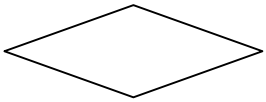
(Sumber : Gellysa Urva dan Helmi Fauzi Siregar : 2015: 94)

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* dapat dilihat pada tabel II.7 dibawah ini:

Tabel II.7. Simbol *Activity Diagram*

Gambar	Keterangan
	*Start point*, diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	*End point*, akhir aktifitas.
	*Activites*, menggambarkan suatu proses/kegiatan bisnis.
	*Fork* (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.

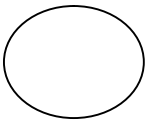
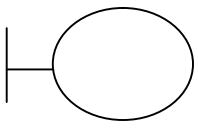
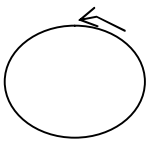
	*Join* (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	*Decision Points*, menggambarkan pilihan untuk pengambilan keputusan, *true, false*.
	*Swimlane*, pembagian *activity* diagram untuk menunjukkan siapa melakukan apa.

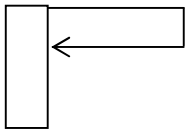
(Sumber : Gellysa Urva dan Helmi Fauzi Siregar : 2015: 94)

3. Diagram Urutan (*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* dapat dilihat pada tabel II.8 dibawah ini :

Tabel II.8. Simbol *Sequence Diagram*

Gambar	Keterangan
	*Entity Class*, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	*Boundary Class*, berisi kumpulan kelas yang menjadi *interface* atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan *form* cetak.
	*Control class*, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	Message, simbol mengirim pesan antar *class*.

	*Recursive,* menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	*Activation, activation* mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	Lifeline, garis titik-titik yang terhubung dengan objek, sepanjang *lifeline* terdapat *activation*.

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar:2015: 95)

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 tanggng 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 yang dapat dilihat pada tabel II.9 dibawah ini :

Tabel II.9. ***Multiplicity Class Diagram***

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

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar : 2015 : 95)