

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Penelitian Terdahulu**

Demi kesempurnaan penelitian, maka penulis perlu melakukan perbandingan untuk mengetahui gambaran dari penelitian terdahulu.

1. Menurut Synthia Marcella (2016), dalam jurnalnya yang berjudul “Analisis Pengakuan Dan Pengukuran Pendapatan Berdasarkan Psak No.23 Pada PT. Pandu Siwi Sentosa Palembang”, Pendapatan adalah peningkatan manfaat ekonomi selama periode akuntansi tertentu dalam bentuk pemasukkan atau penambahan aktiva atau penurunan kewajiban yang mengakibatkan kenaikan ekuitas yang tidak berasal dari kontribusi penanaman modal. manajemen secara cepat dalam penyampaian data, tepat waktu dalam pelaporan, laporan juga dapat dipahami oleh sipemakai, efisien dan ekonomis dengan menghemat pengeluaran dari transaksi manual ke komputersasi sehingga dapat membantu pengambilan dan pengelolaan keputusan dalam pengembangan usaha. Dengan Aplikasi ini dapat menghasilkan informasi akuntansi yaitu laporan keuangan yang dapat digunakan oleh manajemen untuk mengetahui perkembangan usaha. Selain itu juga dapat mengurangi tingkat kesalahan dalam proses pencatatan maupun laporan keuangan
2. Menurut Valen Abraham Lumingkewas (2013), dalam jurnalnya yang berjudul Pengakuan Pendapatan Dan Beban Atas Laporan Keuangan Pada PT. Bank Sulut”, Penelitian ini akan dilakukan pada PT. Bank Sulut yang merupakan

Bank Pembangunan Daerah (BPD) . Memilih penelitian di PT. Bank Sulut karena merupakan Bank daerah yang sudah maju sehingga memungkinkan untuk dianalisis penyajian laporan keuangan pada praktik akuntansi PT. Bank Sulut terutama pada pengakuan pendapatan dan beban. Saat ini sistem informasi akuntansi yang dimiliki estimasi pendapatan yang cukup tinggi, khususnya pada PT. Bank Sulut masih sangat sederhana dan tidak sistematis, dalam artian belum memadai, sehingga tidak dapat memberikan informasi bagi manajer untuk mengambil keputusan. Manajer tidak dapat melihat dengan baik perkembangan dan kinerja usahanya dengan baik dari informasi ataupun catatan yang dimiliki saat ini.

## **II.2. Landasan Teori**

### **II.2.1. Sistem**

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

- a. 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 akuntansi paling dasar dalam setiap organisasi. Sehubungan dengan perkembangan teknologi komputer, istilah pengolahan data mulai dikenal dan mempunyai arti yang sama dengan istilah EDP.

- b. Sistem Informasi Manajemen (SIM), menguraikan penggunaan teknologi komputer untuk menyediakan informasi bagi pengambilan keputusan para manajer.
- c. Sistem Pendukung Keputusan – *Decision Support Systems* (DSS). DSS diarahkan untuk melayani permintaan informasi tertentu, khusus, dan tidak rutin dari manajemen. Contoh adalah penggunaan *spreadsheet* untuk melakukan analisis “*what if*” dari data operasi atau anggaran. (Agustinus Mujilan, 2012).

### **II.2.2. 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 pada saat ia mendukung keputusan alokasi sumberdaya, sehingga dengan demikian mendukung sistem untuk mencapai tujuan. (Agustinus Mujilan, 2012 : 1)

### II.2.3. Sistem Informasi

Berikut ini adalah defenisi sistem informasi menurut beberapa ahli:

**Tabel II.1. Defenisi Sistem Informasi**

Sumber	Defenisi
Alter (1992)	Sistem informasi adalah kombinasi antar prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi
Bodnar Dan Hopwood (1993)	Sistem informasi adalah sekumpulan perangkat keras dan perangkat lunak yang dirancang untuk mentransformasikan data kedalam bentuk informasi yang berguna
Gelinas, Oram Dan Wiggins (1990)	Sistem informasi adalah suatu sistem buatan manusia yang secara umum terdiri atas kumpulan komponen berbasis komputer dan manual yang dibuat untuk menghimpun menyimpan, dan mengelola data ke serta menyediakan informasi keluaran pada para pemakai
Hall (2001)	Sistem informasi adalah sebuah rangkaian prosedur formal dimana data dikelompokkan, diproses menjadi informasi dan didistribusikan kepada pemakai
Turban , Mclean dan Wetherbe (1999)	Sebuah sistem informasi mengumpulkan, memproses, menyimpan, menganalisis dan menyebarkan informasi untuk tujuan spesifik
Wilkinson (1992)	Sebuah sistem informasi adalah kerangka kerja yang mengkoordinasikan suber daya (manusia, komputer) untuk mengubah masukan (input) menjadi keluaran (informasi ), guna mencapai sasaran perusahaan

(Sumber : Abdul Kadir, 2014 : 9)

Berikut ini adalah contoh Sistem Informasi :

1. *System* reservasi pesawat terbang digunakan dalam biro perjalanan untuk melayani pemesanan/ pembelian tiket online.

2. *System* untuk menangani penjualan kredit kendaraan bermotor, sehingga dapat di gunakan untuk memantau piutang para pelanggan, dan system seperti ini sangat mempermudah petugas dalam melayani pembayaran para pelanggan.
3. *System biometric* dapat mencegah orrang yang tidak berwenang memasuki fasilitas-fasilitas rahasia atau mengakses informasi yang bersifat rahasia, dengan cara menganalisis sidik jari atau retina mata.
4. *System point of sale/ POS* yang diterapkan pada kebanyakan pasar swalayan dengan dukungan pembaca Barcode ditujukan mempercepat layanan kepada pelanggan dan memungkinkan persediaan barang bias diketahui oleh petugas dengan cepat. (Abdul Kadir, 2014:2)

#### **II.2.4. 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 keuanga 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.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. (Agustinus Mujilan, 2012 : 3).

### **II.2.6. Pendapatan**

Pendapatan adalah jumlah uang yang diterima oleh perusahaan dari aktivitasnya, Setiap perusahaan dalam melaksanakan kegiatannya mempunyai suatu tujuan untuk menghasilkan laba sesuai yang diinginkan, maka untuk menghasilkan suatu laba yang maksimal tidak terlepas dari masalah pengakuan pendapatan yang diperoleh oleh perusahaan dalam melakukan usahanya. Pendapatan sangat penting bagi kelangsungan hidup perusahaan untuk membiayai segala pengeluaran dan aktivitas perusahaan. Tidak semua kenaikan dari aset dapat disebut sebagai pendapatan, dimana kegiatan utama yang berlanjut merupakan karakteristik yang membatasi kenaikan aset sebagai pendapatan. Menurut karakteristik operasi utama berlanjut, pendapatan merupakan produk perusahaan yang dihasilkan sebagai upaya produktif. Produk yang dihasilkan oleh perusahaan bisa diklasifikasikan sebagai pendapatan operasi dan non operasi. Menurut PSAK nomor 23 paragraf 6 adalah pengertian pendapatan adalah arus masuk bruto dari manfaat ekonomi yang timbul dari aktivitas normal perusahaan selama suatu periode bila arus masuk itu mengakibatkan kenaikan ekuitas yang tidak berasal dari kontribusi penanaman modal. (Sumayyah Yani, 2015 : 6-7)

### **II.2.7. Defenisi Beban**

Beban merupakan konsep arus yang menggambarkan perubahan yang tidak menguntungkan sumber daya perusahaan. Beban timbul akibat adanya aktivitas perusahaan yang biasanya terbentuk pada arus kas atau berkurangnya asset

seperti kas dan setara kas, persediaan serta aset tetap. Definisi beban berkaitan dengan kerugian, karena beban merupakan arus keluar harta atau timbulnya hutang atau keduanya selama satu periode sebagai akibat dari penyerahan atau produksi barang, pemberian jasa, aktivitas yang mendatangkan keuntungan lainnya yang merupakan operasi utama satu entitas. Karakteristik utama beban terjadi di dalam proses pembentukan pendapatan. (Sumayyah Yani, 2015 : 13).

#### **II.2.8. Normalisasi**

Normalisasi dapat dipahami sebagai tahapan-tahapan yang masing-masing berhubungan dengan bentuk normal. Bentuk normal adalah keadaan relasi yang dihasilkan dengan menerapkan aturan sederhana berkaitan dengan konsep kebergantungan fungsional pada relasi yang bersangkutan. (Putu Manik, 2012 : 90). Kita akan menggambarannya secara garis besar sebagai berikut :

##### **1. Bentuk Normal Pertama (1NF/ *First Normal Form*)**

Bentuk normal pertama adalah suatu bentuk relasi dimana atribut bernilai banyak (*multivalued attribute*) telah dihilangkan sehingga kita akan menjumpai nilai tunggal (mungkin saja nilai *null*) pada perpotongan setiap baris dan kolom.

##### **2. Bentuk Normal Kedua (2NF/ *Second Normal Form*)**

Semua kebergantungan fungsional yang bersifat sebagian (*partial functional dependency*) telah dihilangkan.

3. Bentuk Normal Ketiga (3RD/ *Thrid Normal Form*)

Semua kebergantungan transitif (*transitive dependency*) telah dihilangkan.

4. Bentuk Normal *Boyce-Codd* (BCNF/ *Boyce-Codd Normal Form*)

Semua anomaly yang tersisa dari hasil penyempurnaan kebergantungan fungsional sebelumnya telah dihilangkan.

5. Bentuk Normal Keempat (4NF/ *Fourth Normal Form*)

Semua kebergantungan bernilai banyak telah dihilangkan.

6. Bentuk Normal Kelima (5NF/ *Fifth Normal Form*)

Semua anomaly yang tertinggi telah dihilangkan.

### **II.2.9. 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 manipulasi data seperti menambah serta menghapus data. Dengan memanfaatkan komputer, data dapat disimpan dalam media penganget 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.2.9.1. 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.2.9.2. 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 dimasukkan 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.2.10. 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 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.5. Jendela Aplikasi Visual Studio 2010**

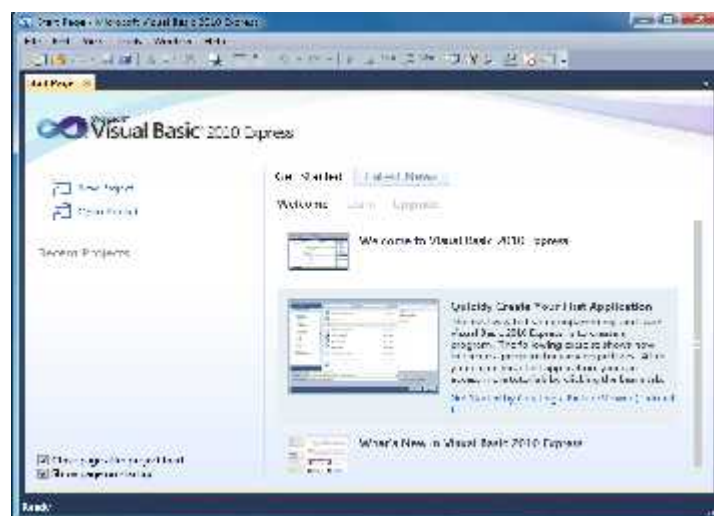
No	Bagian	Keterangan
1	Title Bar	Menampilkan nama aplikasi yang sedang terbuka.
2	Mneu Bar	Menampilkan daftar perintah yang memungkinkan anda dapat menulis, mengedit, menyimpan, mencetak, menguji dan menjalankan program visual basic.
3	Standard Toollbar	Berisi Tombol yang menjalankan perintah yang sering digunakan 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.1.

sebagai berikut :



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

### II.2.11. SQL Server 2008

SQL Server 2008 adalah sebuah RDBMS (*Relational Database Management System*) yang sangat *powerful* dan telah terbukti kekuatannya dalam mengolah data. Dalam versi terbarunya ini, *SQL Server* 2008 memiliki banyak fitur yang bisa diandalkan untuk meningkatkan *performa database*. *SQL Server* 2008 memiliki suatu GUI (*Graphic User Interface*) yang kita gunakan untuk melakukan aktivitas sehari-hari berkaitan dengan *database*, seperti menulis T-SQL, melakukan *backup* dan *restore database*, melakukan *security database* terhadap aplikasi, dan sebagainya. Pada GUI (*Graphic User Interface*) tersebut kita bisa melakukan settingan terhadap *SQL Server* untuk berkerja lebih optimal. Settingan juga bisa dilakukan menggunakan *script* untuk memudahkan *developer* mengubah *Setting Options* pada *SQL Server* 2008. (Ruslan, 2013 : 9)

### II.2.12. UML (*Unified Modelling Language*)


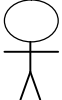
Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga 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. (Windu Gata, 2013)



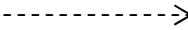
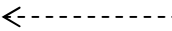
### 1. *Diagram Use Case (Use Case Diagram)*

*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.6. 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 mengindikasikan aliran data.
	Asosiasi antara aktor dan use case yang menggunakan panah terbuka untuk mengindikasikan 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.


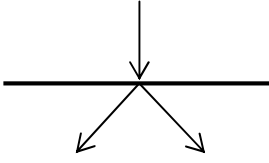
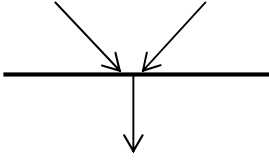
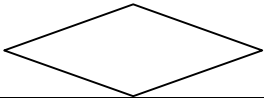

(Sumber : Windu Gata, 2013:5)

## 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.7. Simbol Diagram Aktivitas**

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.

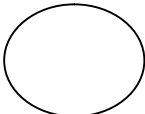
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<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.

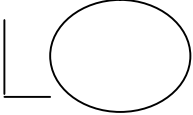
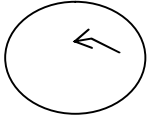

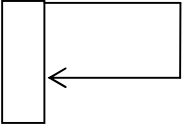
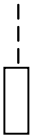

(Sumber : Windu Gata, 2014:6)

### 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.8. Simbol Sequence Diagram**

<b>Gambar</b>	<b>Keterangan</b>
	<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.9. Simbol Class Diagram**

<b>Multiplicity</b>	<b>Penjelasan</b>
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*)