

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Sistem**

Sistem dapat dilakukan dengan 2 pendekatan komponen yaitu dengan pendekatan prosedur, sistem didefinisikan sebagai kumpulan dari beberapa prosedur yang mempunyai tujuan tertentu, sebagai contoh sistem informasi penjualan barang. (Herlambang dan Haryanto Tanuwijaya:2005:116)

Sistem sebagai pendekatan dan prosedur dan dengan pendekatan komponen. Dengan pendekatan prosedur, sistem dapat didefinisikan sebagai kumpulan dan prosedur-prosedur tertentu yang mempunyai tujuan tertentu. Dengan pendekatan komponen sistem dapat didefinisikan sebagai kumpulan komponen yang saling berhubungan satu dengan yang lainnya membentuk satu kesatuan untuk mencapai tujuan tertentu. (Jogiyanto:2009:34)

#### **II.2. Informasi**

Informasi merupakan hasil pengolahan data atau fakta yang dikumpulkan dengan cara tertentu. Informasi disajikan dalam bentuk yang mudah dipahami dan merupakan pengetahuan yang relevan yang dibutuhkan untuk menambah wawasan bagi pemakainya guna mencapai suatu tujuan. (Budi Sutedjo Dharma Oetomo;2006:12)

### **II.2.1. Kualitas Informasi**

Kualitas dari suatu informasi ditentukan oleh beberapa faktor yaitu:

1. Keakuratan dan Teruji Kebenarannya, yaitu informasi harus bebas dari kesalahan-kesalahan, tidak bias, dan tidak menyesatkan. Kesalahan-kesalahan itu dapat berupa kesalahan perhitungan maupun akibat gangguan (noise) yang dapat mengubah informasi tersebut.
2. Kesempurnaan Informasi, yaitu untuk mendukung faktor pertama diatas, maka kesempurnaan informasi menjadi faktor penting, dimana informasi disajikan lengkap tanpa pengurangan, penambahan, atau perubahan.
3. Tepat Waktu, yaitu informasi harus disajikan secara tepat waktu, mengingat informasi akan menjadi dasar dalam pengambilan keputusan. Keterlambatan informasi akan mengakibatkan kekeliruan dalam pengambilan keputusan.
4. Relevansi, yaitu informasi akan memiliki nilai manfaat yang tinggi, jika informasi tersebut diterima oleh mereka yang membutuhkan, dan menjadi tidak berguna jika diberikan kepada mereka yang tidak membutuhkan.
5. Mudah dan Murah, yaitu kini, cara dan biaya untuk memperoleh informasi juga menjadi bahan pertimbangan tersendiri. Bilamana cara dan biaya untuk memperoleh informasi sulit dan mahal, maka orang menjadi tidak berminat untuk memperolehnya, atau menjadi alternative substitusinya. (Budi Sutedjo Dharma Oetomo;2006:16-17)

### **II.3. Sistem Informasi**

Sistem Informasi dapat didefinisikan sebagai kumpulan elemen yang saling berhubungan satu sama lain yang membentuk satu kesatuan untuk mengintegrasikan data, memproses dan menyimpan serta mendistribusikan informasi. Dengan kata lain Sistem Informasi kesatuan elemen – elemen yang saling berinteraksi secara sistematis dan teratur untuk menciptakan dan membentuk aliran informasi yang akan mendukung pembuatan keputusan dan melakukan control terhadap jalannya perusahaan. (Budi Sutedjo Dharma Oetomo; 2006: 11).

### **II.4. Sistem Informasi Akuntansi**

Sistem informasi akuntansi adalah sistem yang bertujuan untuk mengumpulkan dan memproses data serta melaporkan informasi yang berkaitan dengan transaksi keuangan. Misalnya, salah satu input dari sistem informasi akuntansi pada sebuah toko baju, seperti pada contoh sebelumnya, adalah transaksi penjualan. Kita memproses transaksi dengan mencatat penjualan tersebut ke dalam jurnal penjualan, mengklasifikasikan transaksi dengan menggunakan kode rekening, dan memposting transaksi ke dalam jurnal. Kemudian, secara periodik sistem informasi akuntansi akan menghasilkan output berupa laporan keuangan yang terdiri dari neraca dan laporan laba rugi. (Anastasia;2011:4-5).

#### **II.4.1. Tujuan Sistem Informasi Akuntansi**

Sistem informasi akuntansi dapat dijelaskan dari mamfaat yang didapat dari informasi akuntansi. Mamfaat atau tujuan sistem informasi akuntansi tersebut adalah:

1. Mengamankan harta/kekayaan perusahaan. Harta/kekayaan di sini meliputi kas perusahaan, persediaan barang dagangan: termasuk aset tetap perusahaan.
2. Menghasilkan beragam informasi untuk pengambilan keputusan. Misal, pengelola toko swalayan memerlukan informasi mengenai barang apa yang diminati oleh konsumen. Membeli barang dagangan yang kurang laku berarti kas akan terjebak dalam persediaan (yang sulit laku tersebut) dan berarti kehilangan kesempatan untuk membeli barang dagangan yang laku.
3. Menghasilkan informasi untuk pihak eksternal. Setiap pengelola usaha memiliki kewajiban untuk membayar pajak. Besarnya pajak yang dibayar tergantung pada omset penjualan (jika pengelola memiliki menggunakan norma dalam perhitungan pajaknya) atau tergantung pada laba rugi usaha (Anastasia;2011: 5-7)

## **II.5. Hutang**

Hutang adalah kewajiban membayar kepada pihak lain sejumlah uang, barang atau jasa di masa mendatang akibat transaksi di masa lalu. (Rudianto; 2009: 17)

Hutang adalah kewajiban perusahaan untuk membayar kepada pihak lain sejumlah uang/barang/jasa di masa mendatang akibat transaksi di masa lalu. (Rudianto; 2009: 25)

Untuk dapat dikelompokkan sebagai hutang, suatu kewajiban harus memiliki kriteria:

1. Jumlah nominalnya jelas

Berarti hutang tidak dapat didasarkan pada taksiran besarnya kewajiban yang harus dibayar di masa mendatang.

2. Pihak penerimanya jelas

Pihak penerima uang barang jasa yang akan diberikan perusahaan di masa mendatang harus diketahui dengan pasti. Kewajiban yang tidak diketahui pihak penerimanya, seperti garansi purnajual, tidak dapat dikelompokkan sebagai hutang.

3. Berdasarkan transaksi yang telah terjadi di masa lalu

Hutang tersebut timbul akibat transaksi dan kesepakatan legal yang telah terjadi. Bukan karena suatu niat baik atau kewajiban sosial. (Rudianto; 2009: 292)

### II.5.1. Pengelompokan Hutang

Hutang dapat dikelompokkan ke dalam beberapa jenis hutang berdasarkan kategori yang diciptakan, seperti:

1. Berdasarkan Jenis Aktivitas Transaksi yang menjadi penyebab munculnya hutang, maka hutang dapat dikelompokkan menjadi:
  - a. Hutang Usaha adalah hutang yang berasal dari transaksi pembelian barang dan jasa dalam rangka memperoleh pendapatan usaha perusahaan.
  - b. Hutang Bank adalah hutang yang timbul dari transaksi pemberian pinjaman bank kepada perusahaan. Hutang bank biasanya mencakup persyaratan pembayaran, jangka waktu pinjaman dan bunga pinjaman yang dibebankan.
  - c. Wesel Bayar adalah hutang yang disertai dengan janji tertulis kepada pihak kreditor, untuk membayar sejumlah uang di masa mendatang dalam jumlah yang telah disepakati beserta bunga yang ditentukan.
  - d. Hutang Pajak adalah kewajiban yang timbul akibat perusahaan belum membayar pajak yang dikenakan sesuai dengan perundangan yang berlaku, misalnya pajak pertambahan nilai, pajak penghasilan dan sebagainya.

2. Berdasarkan Jangka Waktu Jatuh Temponya, maka hutang dapat dikelompokkan ke dalam kelompok:

- a. Hutang Jangka Pendek yaitu hutang yang harus dilunasi dalam tempo satu tahun. Termasuk dalam kelompok ini adalah: hutang dagang, hutang deviden, hutang jangka panjang yang segera jatuh tempo.
- b. Hutang Jangka Panjang adalah hutang yang jatuh temponya lebih dari satu tahun atau satu periode akuntansi. Jatuh temponya dapat terjadi dalam 1,5 tahun atau 2 tahun atau lima tahun atau lebih dari itu.  
(Rudianto; 2009: 292-295)

## II.6. Pajak

Pajak adalah iuran kepada Negara (yang dipaksakan) yang terutang oleh wajib membayar menurut peraturan-peraturan, dengan tidak mendapat prestasi kembali yang langsung dapat ditunjukkan, dan yang gunanya adalah untuk membiayai pengeluaran-pengeluaran umum berhubungan dengan tugas Negara yang menyelenggarakan pemerintahan. (Dr.Waluyo, M.Sc.,Ak; 2011: 2)

Kutipan beberapa pengertian pajak yang dikemukakan para ahli lainnya adalah sebagai berikut:

1. Pengertian pajak menurut Mr. Dr. NJ. Feldmann dalam buku *De Overheidsmiddelen Van Indonesia* (terjemahan): Pajak adalah prestasi yang dipaksakan sepihak oleh dan terutang kepada pengusaha (menurut

norma-norma yang ditetapkannya secara umum), tanpa adanya kontraprestasi, dan semata-mata digunakan untuk menutup pengeluaran-pengeluaran umum.

2. Pengertian pajak menurut Prof. Dr. MJH. Smeets dalam buku *De Economische Betekenis Belastingen* (terjemahan): Pajak adalah prestasi kepada pemerintah yang terutang melalui norma-norma umum dan yang dapat dipaksakannya, tanpa adanya kontraprestasi yang dapat ditunjukkan dalam hal yang individual, dimaksudkan untuk membiayai pengeluaran pemerintah.
3. Pengertian pajak menurut Dr. Soeparman Soemahamidjaja dalam disertasinya yang berjudul “Pajak Berdasarkan Asas Gotong Royong” menyatakan: “Pajak adalah iuran wajib berupa uang atau barang yang dipungut oleh penguasa berdasarkan norma-norma hukum, guna menutup biaya produksi barang-barang dan jasa-jasa kolektif dalam mencapai kesejahteraan umum”. Dari definisi di atas tidak tampak istilah “dipaksa karena bertitik tolak pada istilah “iuran wajib”. Sisi lainnya yang berhubungan dengan kontraprestasi menekankan pada mewujudkan kontraprestasi itu diperlukan pajak.
4. Prof. Dr. Rochmat. Soemitro, S.H. dalam bukunya *Dasar-dasar Hukum Pajak dan Pajak Pendapatan* (1990: 5) menyatakan: “Pajak adalah iuran kepada kas Negara berdasarkan undang-undang (yang dapat dipaksa) dengan tidak mendapat jasa timbal (kontraprestasi), yang langsung dapat

ditunjukkan dan digunakan untuk membayar pengeluaran umum.

(Dr.Waluyo, M.Sc.,Ak; 2011: 2-3)

Dari pengertian-pengertian tersebut dapat disimpulkan bahwa cirri-ciri yang melekat pada pengertian pajak, adalah sebagai berikut:

1. Pajak dipungut berdasarkan undang-undang serta aturan pelaksanaannya yang sifatnya dapat dipaksakan.
2. Dalam pembayaran pajak tidak dapat ditunjukkan adanya kontraprestasi individual oleh pemerintah.
3. Pajak dipungut oleh Negara baik pemerintah pusat maupaun pemerintah daerah.
4. Pajak diperuntukkan bagi pengeluaran-pengeluaran pemerintah, yang bila dari pemasukkannya masih terdapat surplus, dipergunakan untuk membiayai *public investment*.
5. Pajak dapat pula mempunyai tujuan selain *budgeter*, yaitu mengatur.

(Dr.Waluyo, M.Sc.,Ak; 2011: 3)

## **II.7. Unified Modelling Language**

UML adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem berorientasi objek. Hal ini dikarenakan UML menyediakan bahasa pemodelan visual yang memungkinkan pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti dan dilengkapi dengan mekanisme yang efektif untuk berbagi dan

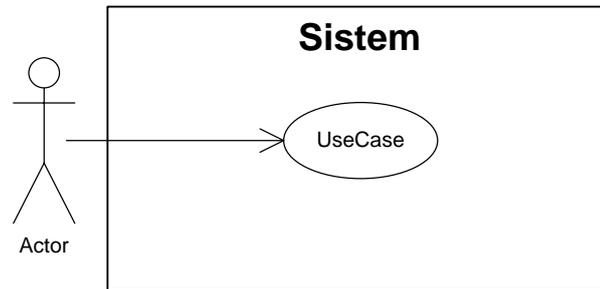
mengkomunikasikan rancangan sistem dengan pengembang lainnya (Munawar;2005:17).

### **II.7.1. Diagram *Usecase***

Menurut Munawar (2005:63), diagram *usecase* adalah deskripsi fungsi dari sebuah sistem dengan perspektif pengguna. *Usecase* bekerja dengan mendeskripsikan tipikal interaksi antar pengguna sistem dengan sistem itu sendiri melalui skenario bagaimana sistem itu digunakan. Setiap skenario mendeskripsikan urutan kejadian dan setiap kejadian diinisialisasikan oleh pengguna, sistem, mesin atau urutan waktu. Dengan demikian secara singkat dapat dikatakan diagram *usecase* adalah serangkaian skenario yang digabungkan bersama-sama oleh tujuan umum pengguna.

Dalam pembicaraan tentang *usecase*, pengguna disebut dengan aktor, dimana aktor adalah sebuah peran yang dimainkan oleh pengguna dalam interaksinya dengan sistem.

Diagram *usecase* menunjukkan tiga aspek dari sistem yaitu aktor, *usecase* dan sistem/sub sistem *boundary*. Aktor mewakili peran orang, *usecase* mewakili aktivitas aktor dan sistem mewakili alat berkomunikasi antara aktor dengan *usecase*. Notasi dan contoh diagram *usecase* dapat dilihat pada gambar II.1 dibawah ini:



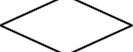
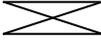
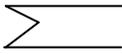
**Gambar II.1 Usecase Diagram**

Sumber: Munawar, 2005:64

### **II.8.2. Diagram Activity**

*Activity Diagram* adalah teknik untuk mendiskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity Diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity* diagram bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa. Adapun simbol *activity diagram* dapat dilihat pada table II.1:

Tabel II.1. Simbol *Activity Diagram*

Notasi	Keterangan
	Titik Awal
	Titik Akhir
	<i>Activity</i>
	Pilihan untuk pengambilan keputusan
	<i>Fork</i> digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu
	<i>Rake</i> menunjukkan adanya dekomposisi
	Tanda waktu
	Tanda pengiriman
	Tanda penerimaan
	Aliran Akhir ( <i>Flow Final</i> )

Sumber : Munawar, 2005:110

### II.8.3. Diagram *Class*

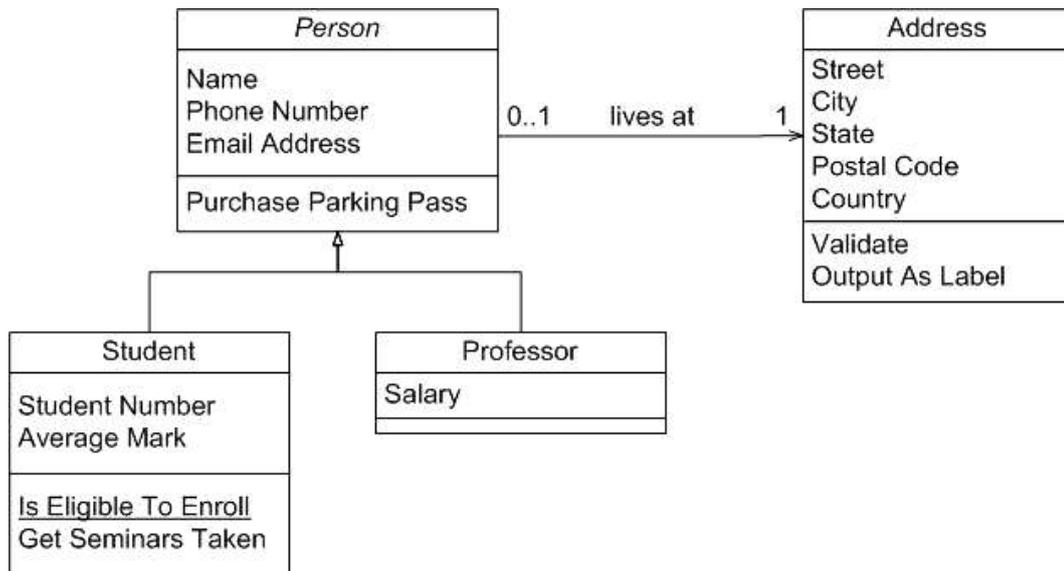
*Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (*atribut*/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (*metoda*/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* memiliki tiga area pokok :

1. Nama kelas
2. Atribut
3. Metode

*Atribut* dan metode dapat memiliki salah satu sifat berikut :

1. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan.
2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan.
3. *Public*, dapat dipanggil oleh siapa saja.

*Class* dapat merupakan implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metode. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class*. Contoh diagram *class* dapat dilihat pada gambar II.2 dibawah ini:



**Gambar II.2 Class Diagram**

Sumber: Munawar, 2005:220

#### II.8.4. Diagram Sequence

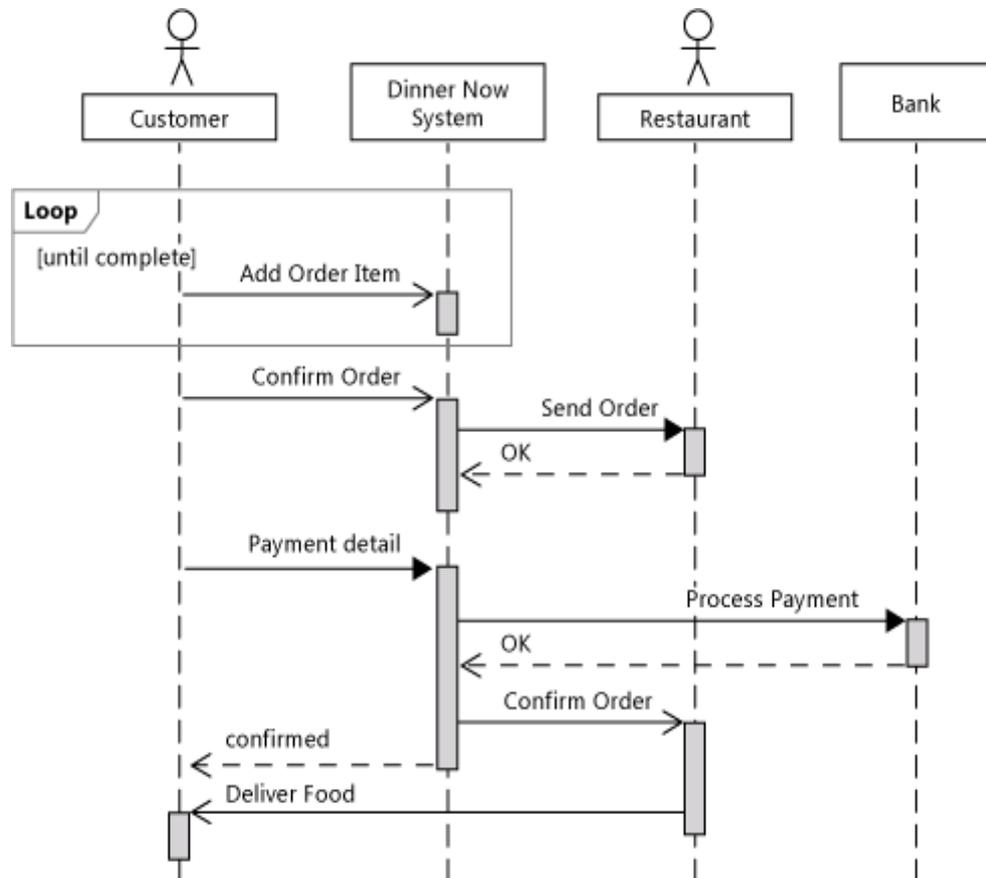
Pesan diawali oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical*.

Untuk *Sequence* adalah diagram yang digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh objek dan pesan yang diletakkan diantara objek-objek ini di dalam *use case*.

Komponen utama *sequence* diagram terdiri dari objek yang dituliskan dengan kotak berbentuk segiempat dan memiliki nama. membentuk diagram *sequence*, perlu diketahui notasi-notasi yang termasuk kedalam diagram *sequence*. Peraturan dalam membuat diagram *sequence* yaitu:

1. Objek atau partisipan diletakkan di dekat bagian atas diagram dengan urutan dari kiri ke kanan.
2. Setiap partisipan terhubung dengan garis titik-titik yang disebut *lifeline*. Pada *lifeline* terdapat kotak yang disebut *activation*. *Activation* mewakili sebuah eksekusi operasi dari partisipan. Panjang kotak ini berbanding lurus dengan durasi *activation*.
3. Sebuah message bergerak dari satu partisipan ke partisipan yang lain dan dari satu *lifeline* ke *lifeline* yang lain. Sebuah partisipan bisa mengirim sebuah pesan kepada dirinya sendiri.
4. *Time* adalah diagram yang mewakkili waktu pada arah vertikal waktu dimulai dari atas akan dijalankan terlebih dahulu.

Contoh diagram *sequence* dapat dilihat pada gambar II.3 dibawah ini:



**Gambar II.3 Sequence Diagram**

Sumber: Munawar, 2005:89

## II.9. Basis Data (*Database*)

Kutipan beberapa pengertian Basisdata yang dikemukakan para ahli lainnya adalah sebagai berikut:

1. Basisdata, menurut Stephens dan plew (2000), adalah mekanisme yang digunakan untuk menyimpan informasi atau data. Informasi adalah sesuatu yang kita gunakan sehari-hari untuk berbagai alasan. Dengan basisdata, pengguna dapat menyimpan data secara terorganisasi. Setelah

data disimpan, informasi harus mudah diambil. Kriteria dapat digunakan untuk mengambil informasi. Cara data disimpan dalam basisdata menentukan seberapa mudah mencari informasi berdasarkan banyak kriteria. Data pun harus mudah ditambahkan ke dalam basisdata, dimodifikasi, dan dihapus.

2. Silberschatz, dkk., (2002) mendefinisikan basisdata sebagai kumpulan data berisi informasi yang sesuai untuk sebuah perusahaan. Sistem manajemen basisdata (DBMS) adalah kumpulan data yang saling berhubungan dan kumpulan program untuk mengakses data. Tujuan utama sistem manajemen basisdata adalah menyediakan cara menyimpan dan mengambil informasi basisdata secara mudah dan efisien.
3. Ramakrishnan dan Gehrke (2003) menyatakan basisdata sebagai kumpulan data, umumnya mendeskripsikan aktivitas satu organisasi atau lebih yang berhubungan.
4. Definisi basisdata, menurut McLeod, dkk., (2001) adalah kumpulan seluruh sumber daya berbasis computer memiliki organisasi. Sistem manajemen basisdata adalah aplikasi perangkat lunak yang menyimpan struktur basisdata, hubungan antardata dalam basisdata, serta berbagai formulir dan laporan yang berkaitan dengan basisdata. Basisdata yang dikendalikan oleh sistem manajemen basisdata adalah satu set catatan data yang berhubungan dan saling menjelaskan.
5. Basisdata warisan (*legacy database*) adalah basisdata yang sedang digunakan oleh sebuah perusahaan. Istilah warisan menyatakan bahwa

basisdata telah dipakai selama beberapa tahun dan basisdata yang ada tidak sesuai dengan teknologi masa kini. Ketika sebuah perusahaan telah menentukan untuk merancang sebuah basisdata, basisdata yang ada dianggap sebagai basisdata warisan. (Janner Simarmata dan Imam Prayudi; 2006: 1-2)

Dalam pembangunan *database*, analisis sistem harus dapat menentukan dalam model arsitektur mana *database* itu akan diletakkan. Dinilai dari penempatannya, arsitektur *database* dapat dikategorikan dalam beberapa bagian yaitu:

1. Terpusat dan Berbagi Data

Pada arsitektur ini, mengurangi penggunaan kertas, *file*, *folder*, dan juga kemungkinan kehilangan atau kesalahan saat menempatkan data. Sekali data disimpan dalam komputer, banyak pengguna bias mengaksesnya via jaringan komputer. Fisik pengguna atau lokasi geografis tidak lagi menjadi batasan.

2. Current Data

Pada arsitektur ini, pengguna bias cepat memperbarui data dan data yang tersedia siap untuk pengguna.

3. Kecepatan dan Produktivitas

Pada arsitektur ini, mengurutkan, *re-retrieve*, membuat perubahan, dan mencetak data, seperti halnya perhitungan jumlah total yang akan secara cepat dilakukan daripada jika melakukannya secara manual.

#### 4. Akurasi Dan Konsistensi

Pada arsitektur ini, merancang basis data untuk memvalidasi masukan data, serta memastikan bahwa data yang dimasukkan adalah valid dan konsisten.

#### 5. Analisis

Pada arsitektur ini, basis data bias menyimpan, menjejaki, dan memproses isi data yang besar dari sumber yang berbeda. Kumpulan data yang bervariasi untuk menjejaki performa dari area untuk menganalisis.

#### 6. Keamanan

Pada arsitektur ini, memproteksi basis data dengan penetapan daftar kata sandi (*password*) dan identifikasi pengguna diotorisasi.

#### 7. Pemulihan

Pada arsitektur ini, kegagalan sistem tidak bisa diremehkan/diacuhkan. Penggunaan basis data memungkinkan integritas data mengalami kegagalan.

#### 8. Tansaksi

Pada arsitektur ini, transaksi menyediakan suatu mekanisme pemulihan kesalahan yang disamaratakan untuk memberikan konsekuensi atas kesalahan yang tidak terduga. Tansaksi memastikan bahwa suatu kelompok perubahan basis data yang terkait selalu terjadi sebagai unit.

(Janner Simarmata; 2007: 4-5)

## II.10. *Entity Relationship Diagram (ERD)*

*Entity Relationship Diagram (ERD)* adalah alat pemodelan data utama dan akan membuat mengorganisasi data dalam suatu proyek ke dalam entitas-entitas menghasilkan struktur basisdata yang baik sehingga data dapat disimpan dan diambil secara efisien. (Janner Simarmata dan Imam Prayudi; 2006: 67)

*Entity Relationship Diagram (ERD)* terbagi atas beberapa yaitu:

### 1. Entitas (*Entity*)

Entitas adalah sesuatu yang nyata atau abstrak di mana kita akan menyimpan data. Ada 4 kelas entitas, yaitu misalnya pegawai, pembayaran, kampus, dan buku.

### 2. Relasi (*Relationship*)

Relasi adalah hubungan alamiah yang terjadi antara atau lebih entitas, misalnya proses pembayaran pegawai. Kardinalitas menentukan kejadian suatu entitas untuk satu kejadian pada entitas yang perhubungan.

### 3. Atribut (*Attribute*)

Atribut adalah cirri umum semua atau sebagian besar instansi pada entitas tertentu. Sebutan lain atribut adalah property, elemen data, dan field. Misalnya nama, alamat, nomor pegawai, dan gaji adalah atribut entitas pegawai. Sebuah atribut atau kombinasi atribut yang mengidentifikasi satu dan hanya satu instansi suatu entitas disebut

kunci utama atau pengenal. (Janner Simarmata dan Imam Prayudi; 2006: 67)

Sruktur logis (skema *database*) dapat ditunjukkan secara grafis dengan diagram ER yang dibentuk dari komponen-komponen berikut:



Persegi panjang mewakili kumpulan entitas.



Elips mewakili atribut.



Belah ketupat mewakili relasi.



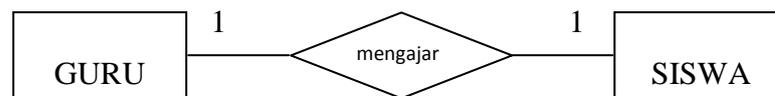
Garis menghubungkan atribut dengan kumpulan entitas dan kumpulan entitas dengan relasi. (Janner Simarmata dan Imam Prayudi; 2006: 60).

### II.10.1. Pemetaan Kardinalitas

Pemetaan kardinalitas menyatakan jumlah entitas di mana entitas lain dapat dihubungkan ke entitas tersebut melalui sebuah himpunan relasi biner meskipun pemetaan dalam deskripsi himpunan relasi yang melibatkan lebih dari dua himpunan entitas.

Untuk suatu himpunan relasi biner  $R$  antara himpunan entitas  $A$  dan  $B$ , pemetaan kardinalitas harus salah satu dari berikut:

1. *One-to-One*, sebuah entitas pada A berhubungan dengan paling banyak satu entitas pada B dan sebuah entitas pada B berhubungan dengan paling banyak satu entitas pada A. Misalnya pada pengajar privat, satu guru satu siswa, seorang guru mengajar seorang siswa, seorang siswa diajar oleh seorang guru.



**Gambar II.4 Hubungan *one-to-one***

Sumber: Janner Simarmata dan Imam Prayudi; 2006: 64

2. *One-to-Many*, sebuah entitas pada A berhubungan dengan nol atau lebih entitas pada B. Sebuah entitas pada B dapat dihubungkan dengan paling banyak satu entitas pada A. Misalnya dalam suatu perusahaan, satu bagian memperkerjakan banyak pegawai, satu bagian memperkerjakan banyak pegawai, satu pegawai kerja dalam satu bagian.



**Gambar II.5 Hubungan *one-to-many***

Sumber: Janner Simarmata dan Imam Prayudi; 2006: 65

3. *Many-to-One*, sebuah entitas pada A berhubungan dengan paling banyak satu entitas pada B. Sebuah entitas pada B dapat dihubungkan dengan nol atau lebih entitas pada A. Misalnya dalam suatu perusahaan, banyak

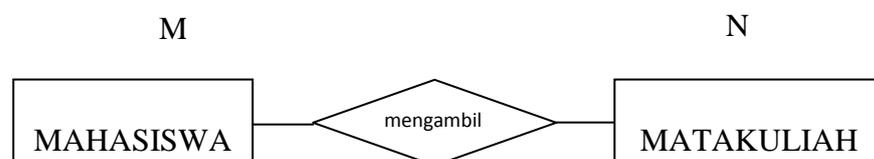
pegawai dipekerjakan pada satu bagian, banyak pegawai dipekerjakan pada satu bagian, satu pegawai bekerja dalam satu bagian.



**Gambar II.6 Hubungan *many-to-one***

Sumber: Janner Simarmata dan Imam Prayudi; 2006: 65

4. *Many-to-Many*, sebuah entitas pada A berhubungan dengan nol atau lebih entitas pada B dan sebuah entitas pada B dapat dihubungkan nol atau lebih entitas pada A. Misalnya dalam universitas, seorang mahasiswa dapat mengambil banyak mata kuliah, satu mahasiswa mengambil banyak mata kuliah dan satu mata kuliah diambil banyak mahasiswa.



**Gambar II.7 Hubungan *many-to-many***

Sumber: Janner Simarmata dan Imam Prayudi; 2006: 66

## II.11. Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basisdata 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. (Janner Simarmata dan Iman Prayudi; 2006: 77)

Normalisasi merupakan proses mengubah perancangan basis data untuk menghasilkan skema table dalam bentuk normal. (Janner Simarmata; 2007: 77)

Normalisasi bisa dipakai oleh para perancang *database* untuk melakukan verifikasi terhadap tabel-tabel yang telah dibuat sehingga tidak menimbulkan suatu permasalahan saat data diperbaharui maupun saat data dihapus. Aturan-aturan normalisasi dinyatakan dalam istilah bentuk normal. Terbagi dalam beberapa bentuk yaitu:

1. Bentuk Normal Pertama (1 NF)

Skema tabel disebut dalam bentuk normal pertama jika nilai atribut tidak terpisahkan. Untuk mengilustrasikannya, semua penulisan buku di dalam atribut tunggal disebut penulis.

2. Bentuk Normal Kedua (2 NF)

Berdasarkan skema, table T memiliki bentuk normal kedua jika semua atribut informasi (atribut yang tidak memiliki kunci mana pun) adalah atribut dari entitas lain di dalam skema tabel dan bukan dari kelas entitas lainnya.

### 3. Bentuk Normal Ketiga (3 NF)

Untuk bentuk normal ketiga, maka relasi haruslah dalam bentuk normal kedua dan semua atribut bukan primer dan tidak mempunyai hubungan yang transitif, dengan kata lain setiap atribut tergantung pada *subset* yang sesuai dengan kunci.

### 4. Bentuk Normal *Boyce Codd* (BCNF)

BCNF merupakan bentuk normal sebagai perbaikan terhadap 3NF. Suatu relasi BCNF selalu memenuhi 3NF, tetapi tidak sebaliknya. Suatu relasi yang memenuhi 3NF belum tentu memenuhi BCNF. BCNF adalah perbaikan dari 3NF karena bentuk normal ketiga pun masih memungkinkan mengandung anomali sehingga masih perlu dinormalisasikan lebih lanjut. Skema tabel disebut bentuk ketiga apabila semua atribut harus bergantung pada *superkey*. (Janner Simarmata; 2007: 77-84)

## **II.12. Microsoft Visual Basic.NET**

Visual Basic merupakan bahasa pemrograman terbaru keluaran Microsoft yang merupakan kelanjutan dari Visual Basic 6.0. Seperti halnya pada Visual Basic 6.0, aplikasi yang dapat dikembangkan oleh Visual Basic.NET antara lain adalah aplikasi database.(Eko Priyo Utomo, ST;2006:11)

Untuk pembuatan aplikasi database, Visual Basic.NET memiliki komponen pendukung yaitu ADO.NET. Sedangkan untuk membuat laporan, Visual

Studio.NET memiliki sebuah alat yaitu Crystal Report. Aplikasi lain yang bisa didukung oleh Visual Basic.NET antara lain aplikasi mobile, web ASP, dan layanan web XML. Aplikasi yang dihasilkan oleh Visual Basic.NET akan berjalan di lingkungan GUI (Graphical User Interface). (Eko Priyo Utomo, ST;2006:11-12)

### **13. My SQL**

MySQL diciptakan di negara swedia oleh perusahaan MSQl AB. Adapun masing-masing nama yang berjasa dalam mmenciptakan MSQl adalah david axmark, allan larsson, dan michael "monty" widenius. Perangkat lunak ini terbesar luas secara gratis karena memiliki lisensi general public license. Sampai sekarang, tercatat ada beberapa bahasa pemograman yang cukup populer yang bisa bersinkronisasi dengan MSQl, Seperti C, C++, C#, bahasa pemograman Eiffel, bahasa pemograman smalltalk, bahasa pemograman java, bahasa pemogramanlips, perl, PHP, bahasa pemograman python, ruby, REARbasic, dan Tcl. ( Andre Adelheid dan khairil Nst 2010:3).