

## **BAB II**

### **TINJAUAN PUSTAKA**

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

Terdapat dua kelompok pendekatan didalam pendefinisian sistem, yaitu yang menekankan pada prosedur dan yang menekankan pada elemen atau komponen (Tata Sutabri, 2012:2). Pendekatan yang menekankan pada prosedur mendefinisikan sistem sebagai suatu jaringan kerja yang saling berhubungan, berkumpul bersama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Sedangkan Pendekatan sistem yang lebih menekankan pada komponen mendefinisikan sistem sebagai kumpulan elemen yang berinteraksi untuk mencapai tujuan tertentu.

Menurut Anastasia Diana, Lilis Setiawati sistem merupakan serangkaian bagian yang saling tergantung dan bekerja sama untuk mencapai tujuan tertentu. Suatu sistem pasti tersusun dari sub – sub sistem yang lebih kecil yang juga saling tergantung dan bekerja sama untuk mencapai tujuan. (2011 : 3). Secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variable yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu. Teori sistem pertama kali diuraikan oleh Kenneth boulding, terutama menekankan pentingnya perhatian terhadap setiap bagian yang membentuk sebuah sistem, (Tata Sutabri, 2012:3).

### **II.1.1 Karakteristik Sistem**

Model umum sebuah sistem terdiri dari input, proses, dan output. Hal ini merupakan konsep sebuah sistem yang sangat sederhana mengingat sebuah sistem dapat mempunyai beberapa masukan dan keluaran sekaligus, (Tata Sutabri, 2012:13). Selain itu sebuah sistem mempunyai karakteristik atau sifat-sifat yang tertentu, yaitu:

#### **1. Komponen Sistem**

Suatu sistem terdiri dari sebuah komponen yang saling bekerja sama untuk membentuk satu kesatuan. Komponen-komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Walaupun sistem tersebut sangat kecil, selalu mengandung komponen-komponen atau subsistem-subsistem

#### **2. Batas Sistem**

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batas suatu sistem menunjukkan ruang lingkup dari sistem tersebut.

#### **3. Lingkungan Luar Sistem**

Lingkungan luar dari suatu sistem adalah apapun diluar batas suatu sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan merugikan sistem itu sendiri. Lingkungan yang menguntungkan merupakan energi dari sistem yang harus dijaga dan dipelihara, sedangkan yang merugikan sistem harus ditahan dan dikendalikan, kalau tidak maka akan mengganggu kelangsungan hidup dari sistem tersebut

#### 4. Penghubung Sistem

Penghubung merupakan media penghubung antara subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari suatu subsistem ke subsistem yang lainnya. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk suatu kesatuan

#### 5. Masukan Sistem

Masukan adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa perawatan dan masukan sinyal. Masukan perawatan adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Masukan sinyal adalah energi yang diproses untuk didapatkan keluaran

#### 6. Keluaran Sistem

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

#### 7. Pengolahan Sistem

Suatu sistem dapat mempunyai suatu bagian pengolahan yang akan merubah masukan menjadi keluaran

#### 8. Sasaran dan Tujuan

Suatu sistem pasti mempunyai tujuan dan sasaran. Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran

yang dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenal sasaran dan tujuannya

### **II.1.2 Pengertian Informasi**

Informasi merupakan proses lebih lanjut dari data yang memiliki nilai tambah. Informasi dapat dikelompokkan menjadi 3 bagian, (Tata Sutabri, 2012:21), yaitu:

1. Informasi Strategis. Informasi ini digunakan untuk mengambil keputusan jangka panjang, yang mencakup informasi eksternal, rencana perluasan perusahaan dan sebagainya.
2. Informasi taktis. Informasi ini dibutuhkan untuk mengambil keputusan jangka menengah, seperti informasi tren penjualan yang dapat dimanfaatkan untuk menyusun rencana penjualan.
3. Informasi teknis. Informasi ini dibutuhkan untuk keperluan operasional sehari-hari seperti informasi persediaan stock, retur penjualan dan laporan kas harian.

Data adalah fakta atau apapun yang dapat digunakan sebagai input untuk menghasilkan informasi. Data dibentuk dari cabang grafis, alfabatis, numerik atau lambang khusus. Jadi informasi adalah data yang telah diolah yang mempunyai nilai guna atau manfaat bagi si pemakai dalam proses pengambilan keputusan atau informasi atau *output* dari proses transformasi dimana data tersebut berfungsi sebagai *input*. Menurut Tata Sutabri, (2012:33), Kualitas dari informasi tergantung dari beberapa hal, yaitu:

1. Keakuratan, dapat didefinisikan sebagai perbandingan dari informasi yang benar dengan jumlah seluruh informasi yang dihasilkan pada satu proses pengolahan data tertentu. Berarti informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan.
2. Tepat waktu, berarti informasi yang datang pada penerima tidak boleh terlambat. Karena informasi yang sudah usang tidak akan bernilai lagi. Informasi merupakan landasan bagi pengambilan keputusan maka jika terjadi keterlambatan dapat berakibat fatal bagi perusahaan atau organisasi itu sendiri.
3. Relevan, berarti informasi tersebut mempunyai manfaat untuk pemakainya. Tingkat relevansi informasi untuk tiap berbeda-beda.

### **II.1.3 Sistem Informasi**

Sistem Informasi, yang kadangkala disebut sebagai sistem pemrosesan data, merupakan sistem buatan manusia yang biasanya terdiri dari sekumpulan komponen baik manual ataupun berbasis komputer yang terintegrasi untuk mengumpulkan, menyimpan, dan mengelola data serta menyediakan informasi kepada pihak – pihak yang berkepentingan sebagai pemakai informasi tersebut.

*Input* dalam sistem informasi adalah data – data yang relevan untuk menghasilkan informasi yang diinginkan. Proses adalah langkah – langkah yang perlu dilakukan untuk mengolah data menjadi informasi. Sedangkan *output* adalah berupa informasi yang merupakan hasil dari pemrosesan data. ( Anastasia Diana, Lilis Setiawati ; 2011 : 4)



**Gambar II.1. Konsep *Input-Proses-Output***  
**Sumber : ( Anastasia Diana, Lilis Setiawati ; 2011 : 4)**

## **II.2 Akuntansi**

Akuntansi adalah suatu disiplin ilmu yang menyediakan informasi penting sehingga memungkinkan adanya pelaksanaan dan penilaian jalannya perusahaan secara efisien. Akuntansi dapat juga didefinisikan sebagai proses mengidentifikasi, mengukur dan melaporkan informasi ekonomi untuk memungkinkan adanya penilaian dan keputusan yang jelas dan tegas bagi mereka yang menggunakan informasi tersebut ( Soemarso S.R ; 2010:14).

### **II.2.1 Akuntansi Biaya**

Akuntansi Biaya adalah Bidang Akuntansi yang berhubungan dengan pengumpulan, analisis, dan pengontrolan atas biaya, Dimana Akuntansi Biaya merupakan proses pencatatan, penggolongan peringkasan dan penyajian biaya pembuatan dan penjualan produk atau jasa. ( Soemarso S.R ; 2010:14).

### **II.2.2 Sistem Akuntansi**

Sistem akuntansi adalah organisasi formulir, catatan, dan laporan yang dikoordinasi sedemikian rupa untuk menyediakan informasi keuangan yang dibutuhkan oleh manajemen guna memudahkan pengelolaan perusahaan. Dari definisi sistem akuntansi tersebut, unsur suatu sistem akuntansi pokok adalah

formulir, catatan yang terdiri dari jurnal, buku besar dan buku pembantu, serta laporan. ( Mulyadi ; 2008 : 3)

### **II.2.3 Sistem Informasi Akuntansi**

Sistem Informasi Akuntansi menurut Anastasia Diana dan lilies setiawati dalam buku yang berjudul Sistem Informasi Akuntansi adalah sebagai berikut: “Sistem yang bertujuan untuk mengumpulkan dan memproses data serta melaporkan yang berkaitan dengan transaksi keuangan. Misalnya, salah satu input dari sistem informasi akuntansi pada sebuah toko baju adalah transaksi penjualan, kemudian akan menghasilkan output berupa laporan keuangan yang terdiri dari neraca dan laporan laba rugi”. (2011: 4).

Output dari sistem informasi akuntansi tersebutlah yang akan digunakan sebagai bahan analisis dalam pengambilan keputusan untuk menentukan kebijakan atau membuat keputusan yang berkaitan dengan usaha tersebut.

Sistem Informasi Akuntansi (SIA) juga merupakan sebuah sistem informasi yang menangani segala sesuatu yang berkenaan dengan akuntansi. Akuntansi sendiri sebenarnya adalah sebuah sistem informasi (Rochmawati Daud dan Valeria Mimosa Windana ; 2014:3). Adapun Faktor–faktor yang dipertimbangkan dalam penyusunan sistem informasi akuntansi yaitu :

1. Sistem informasi akuntansi yang disusun harus memenuhi prinsip.
2. cepat yaitu sistem informasi akuntansi harus menyediakan informasi yang diperlukan dengan cepat dan tepat waktu serta dapat memenuhi kebutuhan dan kualitas yang sesuai.

3. aman yaitu sistem informasi harus dapat membantu menjaga keamanan harta milik perusahaan.
4. murah yang berarti bahwa biaya untuk menyelenggarakan sistem informasi akuntansi tersebut harus dapat ditekan sehingga relatif tidak mahal.

Tujuan sistem informasi akuntansi menurut Rochmawati Daud dan Valeria Mimosa Windana dalam Jurnal Manajemen dan Bisnis Sriwijaya Vol.12 diantaranya adalah:

1. Untuk mendukung operasi-operasi sehari-hari (*to Support the –day-to-day operations*).
2. Mendukung pengambilan keputusan manajemen (*to support decision making by internal decisionmakers*),
3. Untuk memenuhi kewajiban yang berhubungan denganpertanggung jawaban (*to fulfill obligations relating to stewardship*) (2014 : 3).

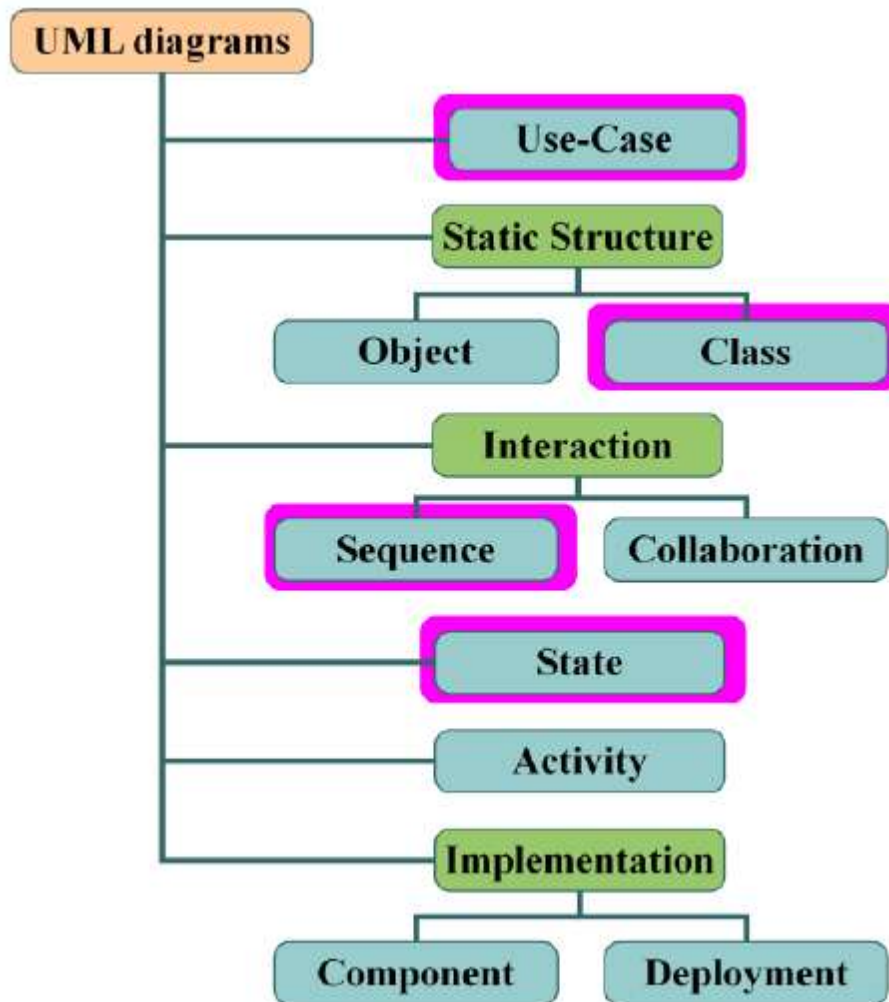
## **II.3 Alat Bantu Perancangan**

Untuk menjelaskan perancangan dari penelitian ini, maka dibutuhkan alat bantu yang akan digunakan sebagai penjelasan model perancangan sebuah sistem.

### **II.3.1 UML (Unified Modeling Language)**

Unified Modelling Language merupakan alat perancangan sistem yang berorientasi pada objek. Secara filosofi kemunculan UML diilhami oleh konsep yang telah ada yaitu konsep permodelan Object Oriented (OO), karena konsep ini menganalogikan sistem seperti kehidupan nyata yang didominasi oleh obyek dan

digambarkan atau dinotasikan dalam simbol-simbol yang cukup spesifik maka OO memiliki proses standard dan bersifat independen, (Haviluddin, 2011:1).



**Gambar II.2 Diagram UML**

**Sumber : ( Haviluddin ; 2011 : 2)**

Berikut tujuan utama dalam desain UML adalah (Sugrue J. 2009) :

1. Menyediakan bagi pengguna (analisis dan desain sistem) suatu bahasa visual yang ekspresif sehingga mereka dapat mengembangkan dan melakukan pertukaran model data yang bermakna.

2. Menyediakan mekanisme yang spesialisasi untuk memperluas konsep inti.
3. Karena merupakan bahasa pemodelan *visual* dalam proses pembangunannya maka UML bersifat independen terhadap bahasa pemrograman tertentu.
4. Memberikan dasar formal untuk pemahaman bahasa pemodelan.
5. Mendorong pertumbuhan pasar terhadap penggunaan alat desain sistem yang berorientasi objek (OO).
6. Mendukung konsep pembangunan tingkat yang lebih tinggi seperti kolaborasi, kerangka, pola dan komponen terhadap suatu sistem.
7. Memiliki integrasi praktik terbaik. ( Havaluddin ; 2011 : 2)

Pada UML versi 2 terdiri atas tiga kategori dan memiliki 13 jenis diagram yaitu :

### **1. Struktur Diagram**

Menggambarkan elemen dari spesifikasi dimulai dengan kelas, *obyek*, dan hubungan mereka, dan beralih ke dokumen arsitektur logis dari suatu sistem.

Struktur diagram dalam UML terdiri atas :

#### 1) *Class* diagram

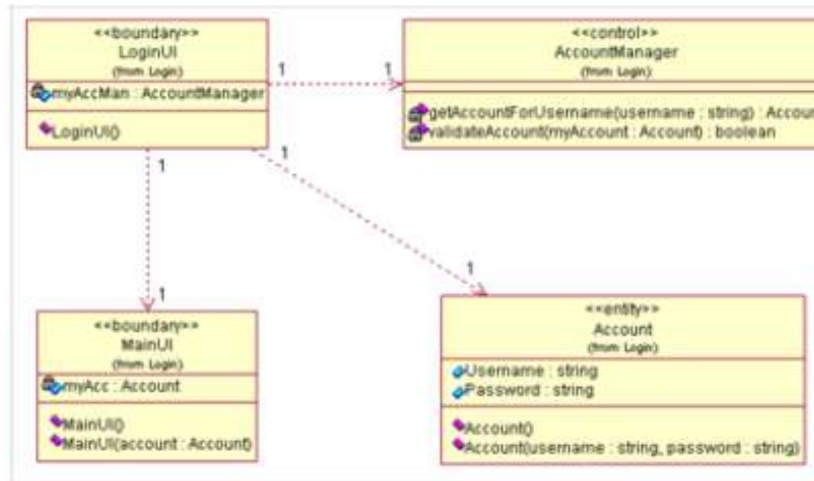
*Class diagram* menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas. *Class diagram* membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama tahap *desain*, *class diagram* berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat. ( Havaluddin ; 2011 : 3)

*Class* memiliki tiga area pokok :

#### 1. Nama (dan *stereotype*)

2. Atribut

3. Metoda (Haviluddin ; 2011 : 3)



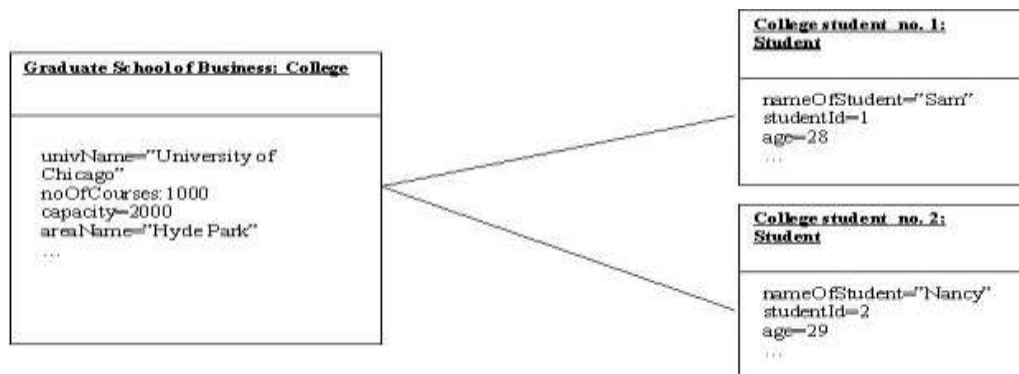
Gambar II.3 Contoh Notasi Class diagram

Sumber : ( Haviluddin ; 2011 : 3)

2) Object diagram

Object diagram menggambarkan kejelasan kelas dan warisan dan kadang-kadang diambil ketika merencanakan kelas, atau untuk membantu pemangku kepentingan non-program yang mungkin menemukan diagram kelas terlalu abstrak.

Berikut notasi *object diagram*. (Haviluddin ; 2011 : 3 )

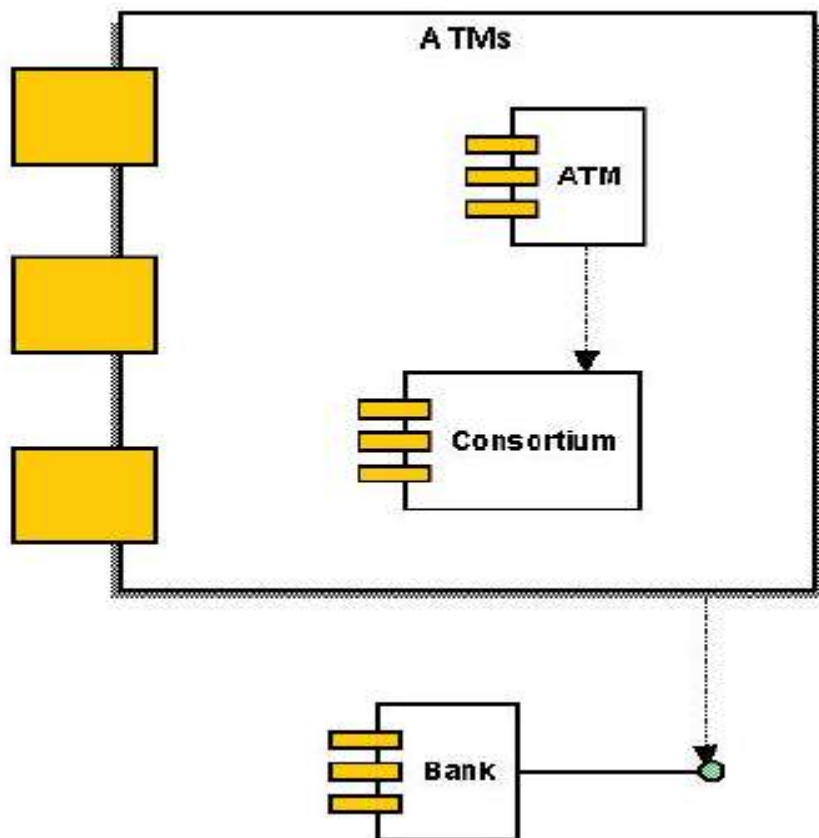


Gambar II.4 Contoh Notasi Object Diagram

Sumber : ( Haviluddin ; 2011 : 3)

### 3) *Component diagram*

*Component diagram* menggambarkan struktur fisik dari kode, pemetaan pandangan logis dari kelas proyek untuk kode aktual di mana logika ini dilaksanakan. (Haviluddin ; 2011 : 3 )



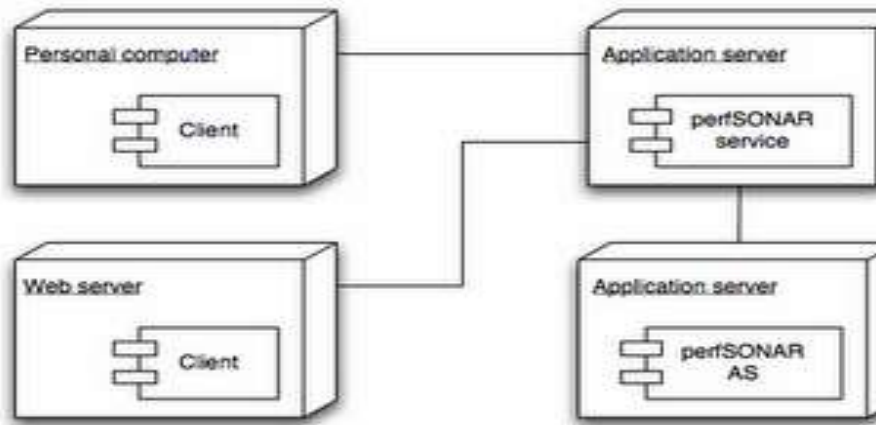
**Gambar II.5 Contoh Notasi *Component Diagram***

**Sumber : ( Haviluddin ; 2011 : 3)**

### 4) *Deployment diagram (Collaboration diagram in version 1.x)*

*Deployment diagram* memberikan gambaran dari arsitektur fisik perangkat lunak, perangkat keras, dan artefak dari sistem. *Deployment diagram* dapat dianggap sebagai ujung spektrum dari kasus penggunaan, menggambarkan bentuk fisik dari

sistem yang bertentangan dengan gambar konseptual dari pengguna dan perangkat berinteraksi dengan sistem. (Haviluddin ; 2011 : 3 )

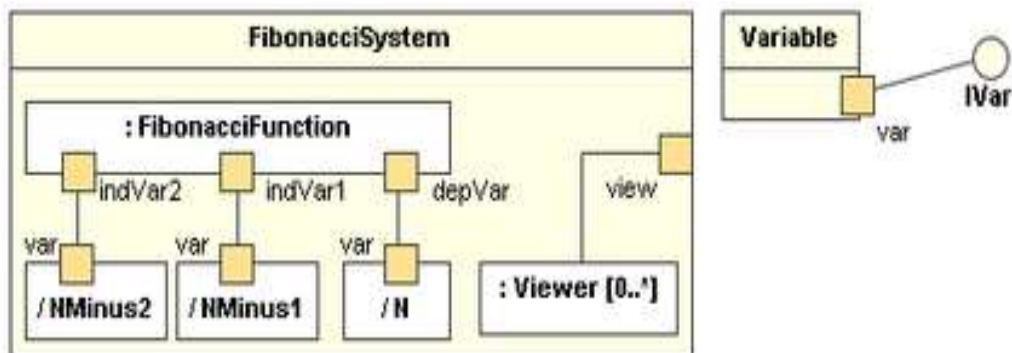


**Gambar II.6** Notasi *Deployment Diagram*

Sumber : ( Haviluddin ; 2011 : 4)

#### 5) *Composite Structure* diagram

Sebuah diagram struktur komposit mirip dengan diagram kelas, tetapi menggambarkan bagian individu, bukan seluruh kelas. Kita dapat menambahkan konektor untuk menghubungkan dua atau lebih bagian dalam atau ketergantungan hubungan asosiasi. (Haviluddin ; 2011 : 4 )

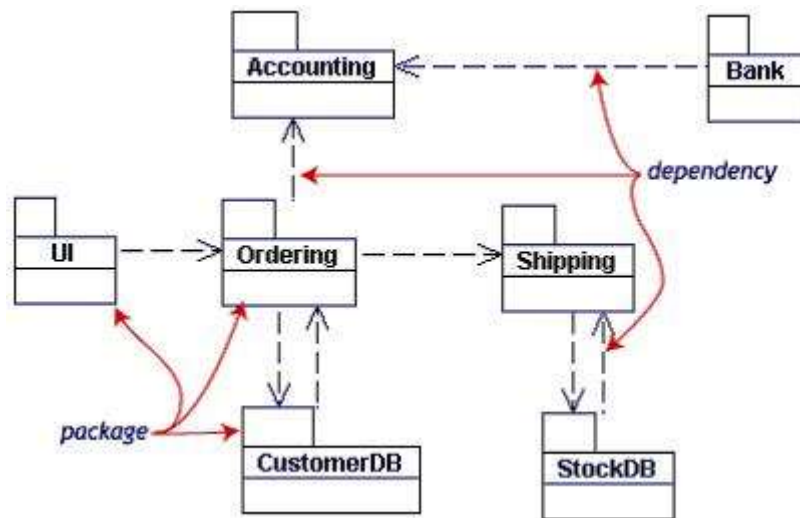


**Gambar II.7** Notasi *Composite Diagram*

Sumber : ( Haviluddin ; 2011 : 4)

## 6) *Package* diagram

Paket diagram biasanya digunakan untuk menggambarkan tingkat organisasi yang tinggi dari suatu proyek *software*. Atau dengan kata lain untuk menghasilkan diagram ketergantungan paket untuk setiap paket dalam Pohon Model. (Haviluddin ; 2011 : 4 )



**Gambar II.8 Notasi *Package Diagram***

Sumber : ( Tgl : 25-6-2014 ; Jam : 21.50 ; <http://rezakusuma15.blogspot.com>)

## 2. *Behavior Diagram*

Menggambarkan ciri-ciri *behavior* atau metode maupun fungsi dari sebuah sistem atau *business process*. *Behavior* diagram dalam UML terdiri atas :

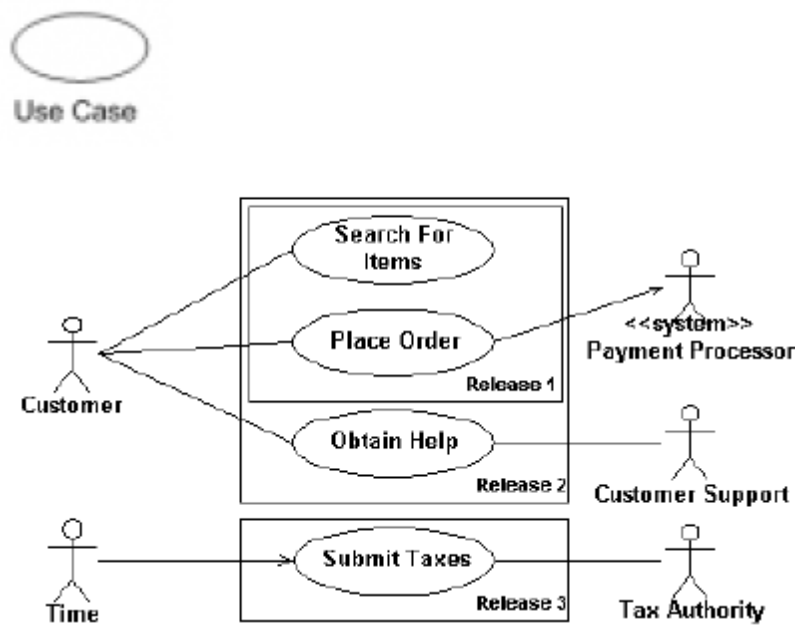
### 7) *Use case* diagram

Diagram yang menggambarkan *actor*, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah *use case* digambarkan sebagai *elips horizontal* dalam suatu diagram UML use case.

*Use Case* memiliki dua istilah

1. *System use case*; interaksi dengan sistem.
2. *Business use case*; interaksi bisnis dengan konsumen atau kejadian nyata.

(Haviluddin ; 2011 : 4 )

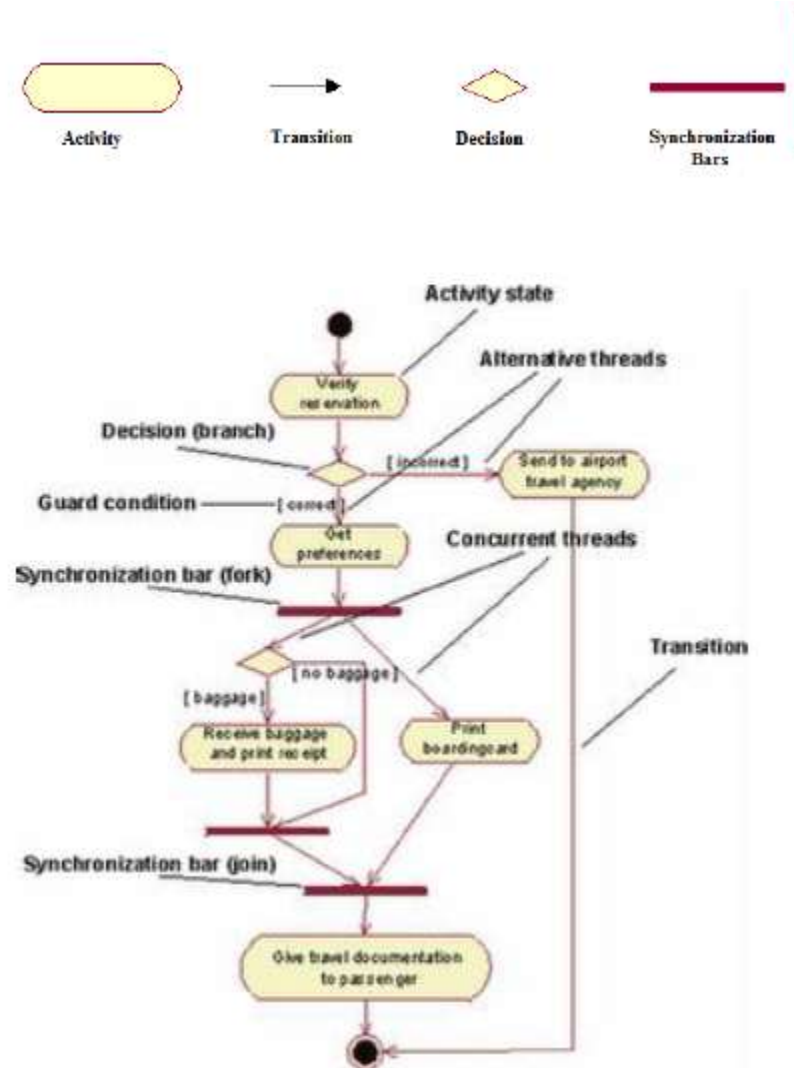


**Gambar II.9 Use Case Diagram**

**Sumber : ( Haviluddin ; 2011 : 4 )**

#### 8) Activity diagram

Menggambarkan aktifitas-aktifitas, objek, *state*, transisi *state* dan *event*. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas. (Haviluddin ; 2011 : 4 )

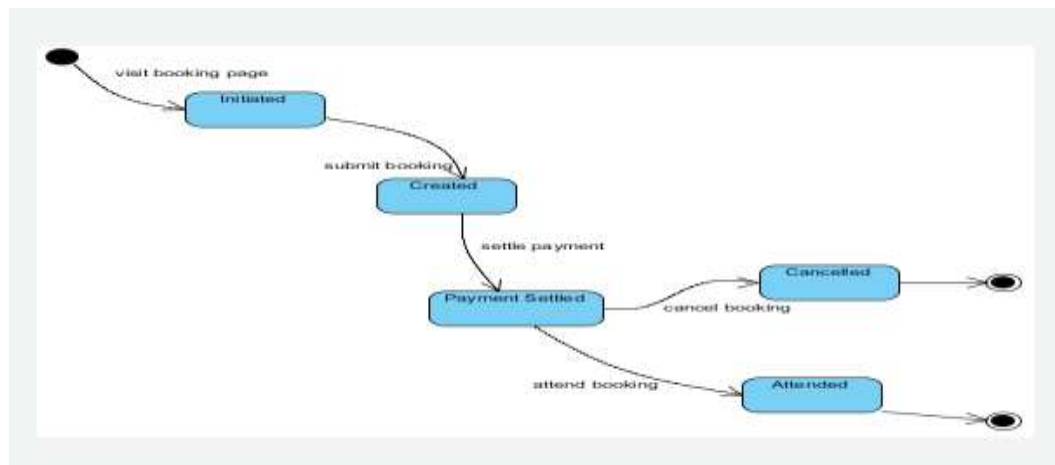


**Gambar II.10 Activity Diagram**

Sumber : ( Havaluddin ; 2011 : 4)

9) *State Machine* diagram (*State chart* diagram in version 1.x)

Menggambarkan *state*, transisi *state* dan *event*. (Havaluddin ; 2011 : 4 )



**Gambar II.11 Notasi State Machine Diagram**

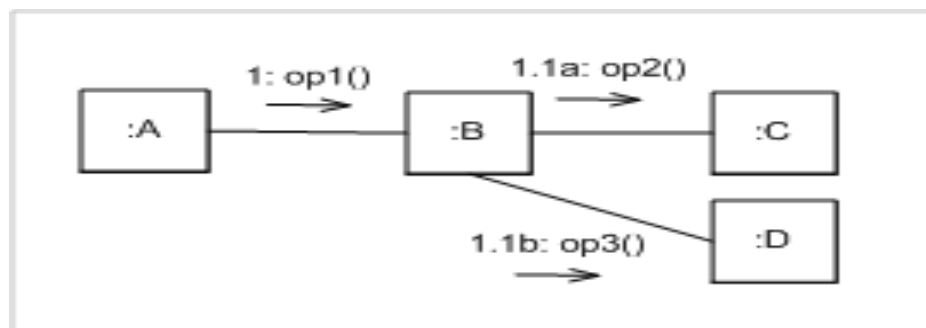
Sumber : ( Havaluddin ; 2011 : 5)

### 3. Interaction diagram

Bagian dari *behavior* diagram yang menggambarkan interaksi objek. *Interaction* diagram dalam UML terdiri atas :

#### 10) Communication diagram

Serupa dengan *sequence* diagram, tetapi diagram komunikasi juga digunakan untuk memodelkan perilaku dinamis dari *use case*. Bila dibandingkan dengan *Sequence* diagram, diagram komunikasi lebih terfokus pada menampilkan kolaborasi benda daripada urutan waktu. (Havaluddin ; 2011 : 5)

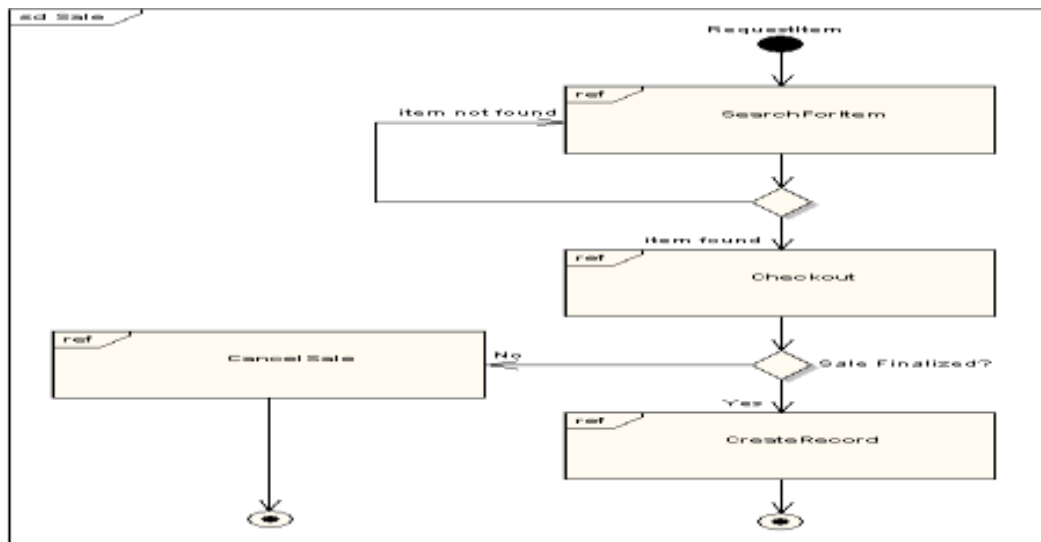


**Gambar II.12 Notasi Communication Diagram**

Sumber : ( Havaluddin ; 2011 : 5)

### 11) *Interaction Overview* diagram

Interaksi *overview* diagram berfokus pada gambaran aliran kendali interaksi dimana node adalah interaksi atau kejadian interaksi. (Haviluddin ; 2011 : 5)

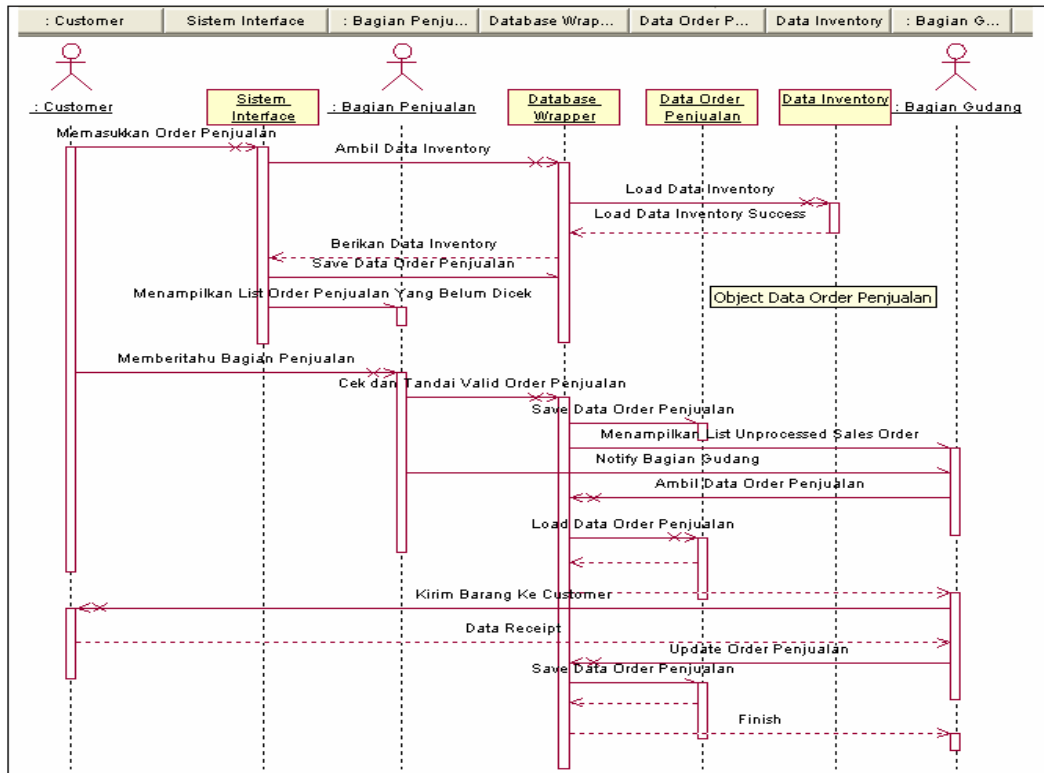


**Gambar II.13 Notasi *Overview Diagram***

**Sumber : ( Haviluddin ; 2011 : 5)**

### 12) *Sequence* diagram

*Sequence* diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *sequence* diagram adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case* diagram. (Haviluddin ; 2011 : 5)

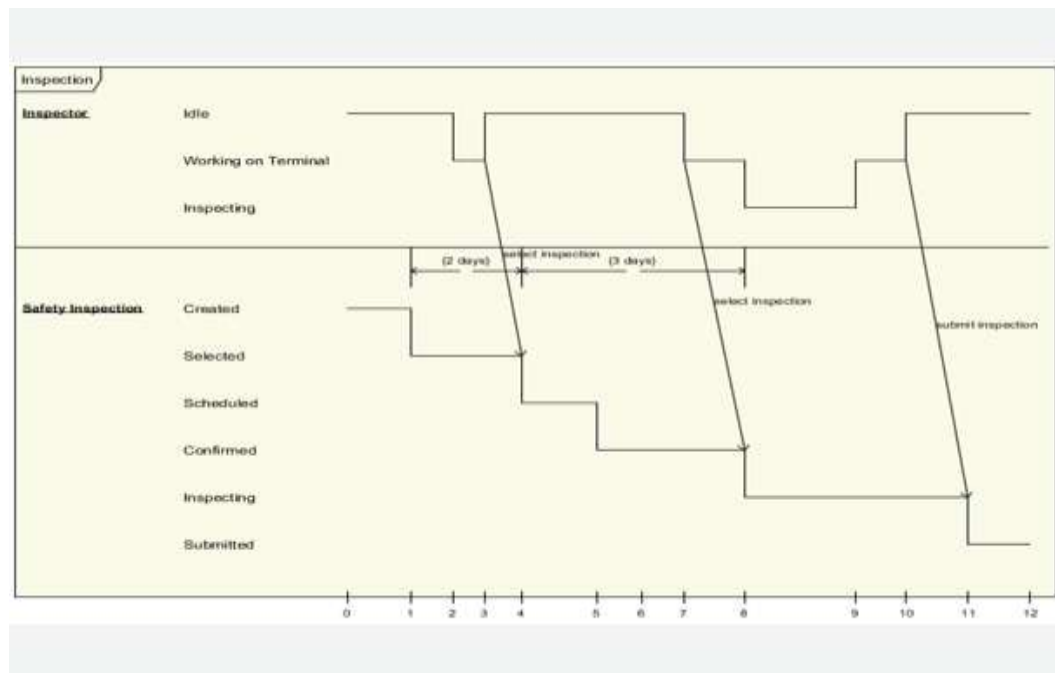


**Gambar II.14 Contoh Notasi *Sequence Diagram***

**Sumber : ( Joan Nugroho ; 2006 : 162)**

### 13) *Timing diagram*

*Timing diagram* di UML didasarkan pada diagram waktu *hardware* awalnya dikembangkan oleh para insinyur listrik. (Haviluddin ; 2011 : 5)



**Gambar II.15 Notasi *Timing Diagram***

**Sumber : ( Haviluddin ; 2011 : 6)**

Berikut akan dijelaskan 4 macam diagram yang paling sering digunakan dalam pembangunan aplikasi berorientasi object, yaitu use case diagram, sequence diagram, collaboration diagram, dan class diagram.

1. Use Case Diagram. Use case diagram digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. Use case diagram terdiri atas diagram untuk use case dan actor. Actor merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi. Use case merepresentasikan operasi-operasi yang dilakukan oleh actor. Use case digambarkan berbentuk elips dengan nama operasi dituliskan di dalamnya. Actor yang melakukan operasi dihubungkan dengan garis lurus ke use case.
2. Sequence Diagram. Diagram Class dan diagram Object merupakan suatu gambaran model statis. Namun ada juga yang bersifat dinamis, seperti

Diagram Interaction. Diagram sequence merupakan salah satu diagram Interaction yang menjelaskan bagaimana suatu operasi itu dilakukan; message (pesan) apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu. Obyek-obyek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut.

3. Collaboration Diagram. Collaboration diagram dipakai untuk memodelkan interaksi antar object di dalam sistem. Berbeda dengan sequence diagram yang lebih menonjolkan kronologis dari operasi-operasi yang dilakukan, collaboration diagram lebih fokus pada pemahaman atas keseluruhan operasi yang dilakukan oleh object.
4. Class Diagram. Class diagram menggambarkan struktur statis class di dalam sistem. class merepresentasikan sesuatu yang ditangani oleh sistem. class dapat berhubungan dengan yang lain melalui berbagai cara: associated (terhubung satu sama lain), dependent (satu class tergantung / menggunakan class yang lain), specialied (satu class merupakan spesialisasi dari class lainnya), atau package (group bersama sebagai satu unit). sebuah sistem biasanya mempunyai beberapa class diagram.

### **II.3.2 Perancangan Sistem ERD**

*Entity relationship* (ER) data model di dasarka pada persepsi terhadap dunia nyata yang tersusun atas kumpulan objek-objek dasar yang disebut entitas dan hubungan antar objek . Entitas adalah sesuatu objek dalam dunia nyata yang

dapat dibedakan dari objek lain. Sebagai contoh, masing-masing mahasiswa adalah entitas dan mata kuliah dapat pula dianggap sebagai entitas.

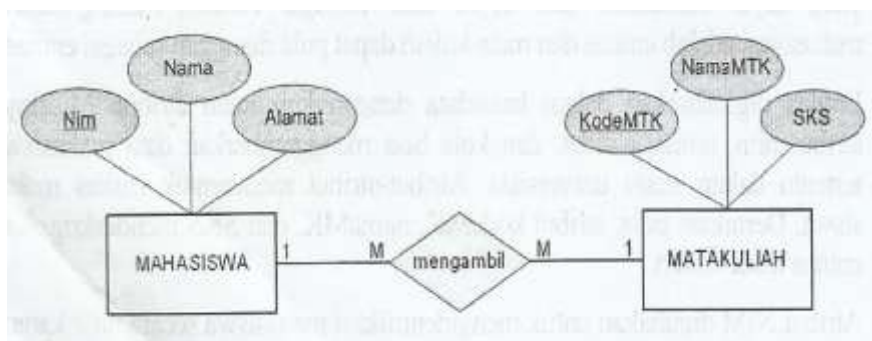
Entitas digambarkan dalam basisdata dengan kumpulan atribut. Misalnya atribut nim, nama, alamat, dan kota bias menggambarkan data mahasiswa tertentu dalam suatu universitas. Atribut-atribut membentuk entitas mahasiswa. Demikian pula, atribut kodeMK, namaMK, dan SKS mendeskripsikan entitas mata kuliah.

Atribut NIM digunakan untuk mengidentifikasi mahasiswa secara unik karena dimungkinkan terdapat dua mahasiswa dengan nama, alamat, dan kota yang sama. Pengenal unik harus diberikan pada masing-masing mahasiswa. Relasi adalah hubungan antara beberapa entitas. Sebagai contoh, relasi menghubungkan mahasiswa dengan mata kuliah yang di ambilnya. Kumpulan semua entitas bertipe sama disebut kumpulan entitas (*entity set*), sedangkan kumpulan semua relasi bertipe sama disebut kumpulan relasi (*relation ship set*). (Janner Simarmata, Iman Paryudi; 201 0 : 60 ). Struktur logis (skema *database*) dapat ditunjukkan secara grafis dengan diagram ER yang dibentuk dari komponen-komponen berikut :

**Tabel II.1. Atribut *Entity Relationship* (ER)**

<div style="border: 1px solid black; width: 60px; height: 30px; margin: 0 auto; display: flex; align-items: center; justify-content: center;">Entitas</div>	Persegi panjang mewakili kumpulan entitas
<div style="border: 1px solid black; border-radius: 50%; width: 60px; height: 30px; margin: 0 auto; display: flex; align-items: center; justify-content: center;">Atribut</div>	Elips mewakili atribut
<div style="border: 1px solid black; width: 60px; height: 30px; margin: 0 auto; transform: rotate(45deg); display: flex; align-items: center; justify-content: center;">Rela</div>	Belah ketupat mewakili relasi
<div style="border-top: 1px solid black; width: 60px; height: 5px; margin: 0 auto;"></div>	Garis menghubungkan atribut dengan kumpulan entitas dan kumpulan entitas dengan relasi.

Masing-masing komponen diberi nama entitas atau relasi yang diwakilinya. Sebagai ilustrasi, bayangkan anda mengambil bagian *system* basis data universitas yang terdiri atas mahasiswa dan mata kuliah. Gambar II.4 menunjukkan diagram ER dari contoh. Diagram menunjukkan bahwa ada dua kumpulan entitas yaitu mahasiswa dan mata kuliah, dan bahwa relasi mengambil mahasiswa dan mata kuliah.

**Gambar II.16. Diagram ER**

(Sumber: Jannerr Simarmata & Iman Paryudi, 2010: 60)

### II.3.3. Membuat ERD

*Entity Relationship* Diagram adalah alat pemodelan data utama dan akan membantu dan mengorganisasi data dalam suatu proyek ke dalam entitas-entitas dan menentukan hubungan antar entitas. Proses memungkinkan analisis menghasilkan struktur basis data yang baik sehingga data dapat disimpan dan diambil secara efisien.

#### 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. Contoh suatu entitas disebut instansi, misalnya pegawai Adi, pembayaran Joko dan lain sebagainya.

#### 2. Relasi (*Relationship*)

Relasi adalah hubungan alamiah yang terjadi antara satu atau lebih entitas, misalnya proses pembayaran pegawai. Kardinalitas menentukan kejadian suatu entitas untuk satu kejadian pada entitas yang berhubungan. Misalnya, mahasiswa bisa mengambil banyak mata kuliah.

#### 3. Atribut (*Attribute*)

Atribut adalah ciri 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. Misalnya, nomor pegawai adalah kunci utama untuk pegawai.( Janner Simarmata, Iman Paryudi; 2010 : 67 ).

## **II.4 Perangkat Lunak Yang digunakan**

### **II.4.1 Visual Basic 2010**

*Visual Basic* 2010 merupakan salah satu bagian dari produk pemrograman terbaru yang dikeluarkan oleh *Microsoft*, yaitu *Microsoft Visual Studio* 2010. Sebagai produk lingkungan pengembangan terintegrasi atau IDE andalan yang dikeluarkan oleh *Microsoft*, *Visual Studio* 2010 menambahkan perbaikan – perbaikan fitur dan fitur baru yang lebih lengkap dibandingkan versi *Visual Studio* 2008. (Wahana Komputer ; 2010 : 2 )

Supaya aplikasi *Visual Basic* 2010 dapat berjalan dengan optimal, perlu dilakukan pengecekan apakah perangkat yang akan digunakan nantinya sudah memenuhi kriteria yang ditetapkan oleh *Microsoft* selaku pengembang aplikasi *Visual Studio* 2010. Persyaratan awal yang harus dipenuhi sebelum instalasi dapat dilakukan adalah bahwa komputer yang nantinya akan diinstal *Visual Studio* 2010 minimal harus terdapat sistem operasi sebagai berikut :

1. *Windows 7*
2. *Windows Vista*
3. *Windows XP Service Pack 3*
  - *Internet Explorer 6 Service Pack 1*
  - *Windows Installer 3.1, minimum*
4. *Windows 2008*

#### 5. *Windows Server 2003 Service Pack 2*

Selain Persyaratan sistem operasi minimal yang bisa digunakan, terdapat pula persyaratan awal *library* aplikasi atau *prerequisites* yang harus dipenuhi, diantaranya :

1. *Visual C Runtime 9.0 \**
2. *Visual C Runtime 10.0\**
3. *.NET Framework 4\**
4. *Visual Studio Macro Tools\**
5. *TFS Object Model\** ( Wahana Komputer ; 2010 : 2 )

*Visual Basic 2010* merupakan lingkungan pengembangan terintegrasi atau biasa disebut IDE yang dikembangkan berbasis bahasa pemrograman *BASIC*. Bahasa *BASIC* sendiri sebenarnya sudah lama dibuat dan dikembangkan oleh *Microsoft Corporation* dengan nama *Microsoft Quick Basic*. ( Wahana Komputer ; 2010 : 36 )

Dalam bahasa *BASIC*, variabel dapat diartikan sebagai suatu tampungan yang digunakan sebagai penampung sebuah nilai atau *value*. ( Wahana Komputer ; 2010 : 36 )

*BASIC* menyediakan beberapa jenis variabel yang dapat digunakan oleh pengembang dalam membangun aplikasi. Adapaun jenis – jenis variabel tersebut dicantumkan pada tabel II.1.

**Tabel II.2 Jenis – jenis variabel pada BASIC**  
**Sumber : ( Wahana Komputer ; 2010 : 37 )**

No	Nama Variabel	Keterangan
1.	String	Terdiri dari beberapa huruf dan angka
2.	Char	Terdiri dari 1 huruf/ angka (1karakter)
3.	Integer	Range = -214783648..2147483647 Format = signed 32 bit
4.	Long	Range = -9,223,372,036,854,775,808.. 9,223,372,036,854,775,807 Format = signed 64 bit
5.	Short	Range = -32,768..32,767 Format = signed 64 bit
6.	Ulong	Range = 0.. 18,446,744,073,709,551,615 Format = signed 64 bit
7.	Byte	Range = 0..255 Format = unsigned 8-bit
8.	Smallint	Range = -32768..32767 Format = signed 16-bit
9.	Longword	Range = 0..4294967295 Format = unsigned 32-bit
10.	Boolean	True and False
11.	Date	0:00:00(midnight) on January 1,0001 through 11:59PM on December 31,9999

#### **II.4.2 Database (Basis Data)**

*Database* atau basis data adalah sekumpulan data yang memiliki hubungan secara logika dan diatur dengan susunan tertentu serta disimpan dalam media penyimpanan komputer. Data itu sendiri adalah representasi dari semua fakta yang ada pada dunia nyata. ( Wahana Komputer ; 2010 : 24 )

Basis data adalah kumpulan data yang saling berelasi. Data sendiri merupakan fakta mengenai obyek, orang, dan lain – lain. Data dinyatakan dengan nilai (angka, deretan karakter, atau *symbol*). ( Kusrini ; 2007 : 2 )

Basis data dapat didefinisikan dalam berbagai sudut pandang seperti berikut :

1. Himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa sehingga kelak dapat dimanfaatkan dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa pengulangan (*redundancy* ) yang tidak perlu, untuk memenuhi kebutuhan.
3. Kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik. ( Kusrini ; 2007 : 2)

Basis data bertujuan untuk mengatur data sehingga diperoleh kemudahan, ketepatan, dan kecepatan dalam pengambilan kembali. (Kusrini ; 2007 : 2)

### **II.4.3 SQL Server 2008**

*SQL Server* 2008 adalah sebuah terobosan baru dari *Microsoft* dalam bidang *database*. *SQL Server* adalah sebuah DBMS (*Database Management System*) yang dibuat oleh *Microsoft* untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti IBM dan *Oracle*. *SQL Server* 2008 dibuat pada saat kemajuan dalam bidang *hardware* sedemikian pesat. Oleh karena itu sudah dapat dipastikan bahwa *SQL Server* 2008 membawa beberapa terobosan dalam bidang pengolahan dan penyimpanan data ( Wahana Komputer ; 2010 : 2)

SQL *Server* 2008 termasuk salah satu mesin *database* relasional. IDE tentang sistem *database* relasional pertama kali dicetuskan oleh seorang yang bernama E.F.Codd lewat sebuah artikel yang berjudul “*A Relational Model of Data For Large Shared Data Banks* “. Artikel tersebut ditulis pada tahun 1970. Sistem *database relational* berdasarkan pada metode lama yang berdasarkan *network* dan hirarki. Metode relasional ini bekerja berdasarkan keterkaitan antartabel. (Wahana Komputer ; 2010 : 25)