

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Konsep Sistem Informasi**

##### **II.1.1. Definisi Sistem**

Jogiyanto (2008: 34) mendefinisikan 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.

Ardhian Agung (2009 : 2) Sistem ialah interaksi dari elemen-elemen yang saling berkaitan bekerja sama untuk mencapai tujuan. Elemen-elemen tersebut ialah elemen sistem konvensional (data, manusia dan prosedur) dan elemen sistem modern (data, manusia, prosedur, *hardware* dan *software*).

Herlambang dan Haryanto Tanuwijaya (2005: 116) menyatakan 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.

Dari definisi sistem di atas, dapat disimpulkan sistem adalah suatu jaringan kerja yang saling memiliki keterkaitan antar bagian dan prosedur-prosedur yang ada.

### II.1.2. Definisi Informasi

Jogiyanto (2008: 36) mendefinisikan informasi sebagai data yang diolah menjadi bentuk yang berguna bagi para pemakaiannya.

Herlambang dan Haryanto Tanuwijaya (2005: 122) Pengertian mengenai data dan informasi sering rancu. Ada yang mengatakan bahwa data itu sama dengan informasi, atau data itu berisi informasi, data sebenarnya merupakan fakta-fakta atau kejadian yang dapat berupa angka-angka atau kode-kode tertentu. Data berupa angka atau wujud yang lain belum mempunyai arti bagi penggunanya. Sedangkan informasi adalah data yang telah diolah dan mempunyai arti bagi penggunanya.

Edhy Sutanta (2011: 13) Informasi merupakan hasil pengolahan data sehingga menjadi bentuk yang penting bagi penerimanya dan mempunyai kegunaan sebagai dasar dalam pengambilan keputusan yang dapat dirasakan akibatnya secara langsung saat itu juga atau secara tidak langsung pada saat mendatang. Informasi diperlukan oleh pemakai (manajemen) pada seluruh level manajemen dalam seluruh fungsi *organisatoris*. Informasi tersebut dapat mempunyai beberapa fungsi, antara lain :

#### 1. Menambah Pengetahuan

Adanya informasi akan menambah pengetahuan bagi penerimanya yang dapat digunakan sebagai bahan pertimbangan yang mendukung proses pengambilan keputusan.

#### 2. Mengurangi Ketidakpastian

Adanya informasi akan mengurangi ketidakpastian karena apa yang akan

terjadi dapat diantisipasi dengan baik sehingga menghindari keraguan pada saat pengambilan keputusan.

### 3. Mengurangi Risiko Kegagalan

Adanya informasi akan mengurangi risiko kegagalan karena apa yang akan terjadi dapat diantisipasi dengan baik sehingga kemungkinan terjadinya kegagalan akan dapat dikurangi dengan pengambilan keputusan yang tepat.

### 4. Mengurangi Keanekaragaman Yang Tidak diperlukan

Adanya informasi akan mengurangi keanekaragaman yang tidak diperlukan karena keputusan yang diambil lebih terarah.

## II.1.3. Sistem Informasi

Ardhian Agung (2009 : 2-3) Sistem Informasi ialah interaksi antara data, manusia dan prosedur (yang didukung oleh *hardware* dan *software*) untuk memberikan suatu penyelesaian berupa informasi yang dapat dipakai untuk mengambil suatu tindakan keputusan selanjutnya baik untuk jangka pendek, menengah atau panjang dengan kata lain sistem informasi juga adalah suatu kumpulan dari komponen-komponen yang saling berinteraksi untuk mengelola informasi .

Tujuan utama sistem informasi pada saat itu adalah untuk melakukan otomatisasi proses bisnis yang berjalan pada organisasi. Pada tahun 2000 an, sistem informasi berkembang semakin pesat. Perkembangan ini didorong dengan semakin berkembang teknologi internet, dengan kapasitas semakin besar dan

harga yang semakin murah. Sudah banyak organisasi yang telah mengintegrasikan sistem informasi mereka dengan sistem informasi organisasi lain untuk mendukung kegiatan organisasi tersebut

Pada masa mendatang, sistem informasi akan semakin berkembang lagi. Perkembangan teknologi dan perubahan dunia usaha yang sangat cepat, mendorong organisasi untuk mengembangkan suatu sistem informasi yang mampu beradaptasi dengan cepat menghadapi perubahan tersebut. Sistem informasi tersebut juga harus dapat diintegrasikan dengan bermacam-macam sistem yang lain agar kinerja organisasi menjadi lebih efisien

## **II.2. Penyusutan Aktiva Tetap**

Warren Reeve Fess (2006: 509-511) Perusahaan menggunakan berbagai macam aktiva tetap, seperti peralatan, perabot, alat-alat, mesin-mesin, bangunan, dan tanah. Aktiva tetap (*fixed assets*) merupakan aktiva jangka panjang atau aktiva relatif permanen. Mereka merupakan aktiva berwujud (*tangible assets*) karena terlihat secara fisik. Aktiva tersebut dimiliki dan digunakan oleh perusahaan serta tidak dimaksudkan untuk dijual sebagai bagian dari operasional normal. Nama-nama deskriptif lain bagi aktiva-aktiva ini adalah aktiva pabrik (*plant assets*), atau properti, pabrik, dan peralatan (*property, plant assets, and equipment*).

Tiga faktor harus dipertimbangkan dalam menentukan jumlah beban penyusutan yang diakui setiap periode. Ketiga faktor itu adalah:

- a. Biaya awal aktiva tetap
- b. Umur manfaat yang diperkirakan
- c. Estimasi nilai pada akhir umur manfaat

Salah satu metode yang biasa digunakan adalah metode saldo menurun. Metode saldo menurun (*declining-balance method*) menghasilkan beban periodik yang terus menurun sepanjang estimasi umur manfaat aktiva. Untuk merapkan metode ini, tarif penyusutan garis lurus tahunan terlebih dahulu harus digandakan. Sebagai contoh, tarif penyusutan saldo menurun atas suatu aktiva yang memiliki estimasi umur manfaat 5 tahun adalah 40%, yaitu dua kali tarif garis lurus sebesar 20% ( $100\% / 5$ ).

Untuk tahun pertama, biaya aktiva dikalikan dengan tarif saldo menurun. Setelah tahun pertama, nilai buku (*book value*) yang menurun (biaya dikurangi akumulasi penyusutan) dikalikan dengan tarif yang dimaksud. Sebagai ilustrasi, penyusutan saldo menurun tahunan atas suatu aktiva yang memiliki umur manfaat 5 tahunan dan biaya \$24.000 diperlihatkan berikut ini:

Tabel II.1: Penyusutan Saldo Menurun

Tahun	Harga Perolehan	Akumulasi Penyusutan Awal Tahun	Nilai Buku Awal Tahun	Tarif	Penyusutan Tahunan	Nilai Buku Akhir Tahun
1	24.000		24.000	40	9.600	14.400
2	24.000	9.600	14.400	40	5.760	8.640
3	24.000	15.360	8.640	40	3.456	5.184
4	24.000	18.816	5.184	40	2.073	3.110
5	24.000	20.889	3.110	40	1.110	2.000

Sumber : Warren Reeve Fess (2006: 512)

Penggunaan metode saldo menurun, estimasi nilai sisa tidak diperhitungkan dalam penentuan tarif penyusutan. Nilai sisa juga diabaikan dalam penghitungan periode penyusutan. Namun aktiva tidak boleh disusutkan melampaui estimasi nilai sisa. Dalam contoh di atas, estimasi nilai sisa adalah \$2.000. Jadi, Penyusutan untuk tahun-5 adalah \$1.110,40 ( $\$3.110,40 - \$2.000,00$ ), bukan \$1.244,16 ( $40\% \times \$3.110,40$ ).

Dalam contoh di atas, diasumsikan bahwa permulaan pemakaian aktiva terjadi pada awal tahun fiskal. Hal ini jarang terjadi dalam praktik dan karenanya, penyusutan untuk sebagian tahun pertama penggunaan harus dihitung. Sebagai contoh, asumsikan bahwa aktiva diatas mulai digunakan pada akhir bulan ketiga tahun fiskal. Dalam hal ini, hanya sebagian (yaitu, 9/12) dari seluruh penyusutan tahun pertama sebesar, \$7.200 ( $9/12 \times \$9.600$ ) dialokasikan ke sebagian dari tahun pertama. Penyusutan untuk tahun fiskal kedua akan menjadi \$6.720 [ $40\% \times (\$24.000 - \$7.200)$ ].

### **II.3. Konsep Basis Data**

Pengatahuan kosep basis data muncul dan mulai berkembang seiring dengan adanya kebutuhan pengolahan dan penyimpanan data untuk memenuhi kebutuhan informasi. Perkembangan basis data telah berkembang ke arah aplikasi-aplikasi basis data untuk sistem kecerdasan buatan (*Artificial Intelligent/AI*).

Edhy Sutanta (2011: 29) menyatakan basis data sebagai suatu kumpulan data terhubung (*interrelated data*) yang disimpan secara bersama-sama pada suatu

media, tidak perlu suatu kerangkapan data. Data disimpan dengan cara-cara tertentu sehingga mudah digunakan/atau ditampilkan kembali.

Janner Simarmata (2007: 4-5) menyatakan bahwa basis data merupakan Koleksi data yang bisa mencari secara menyeluruh dan secara sistematis memelihara me-retrieve informasi.

Sistem basis data terus dikembangkan oleh para ahli agar dapat diperoleh cara pengorganisasian data yang efisien dan efektif. Hal ini diperlukan karena sekarang ini berbagai bidang usaha telah menjadikannya sebagai tumpuan manajemen informasi perusahaannya. Adapun penerapan sistem *database* ini antara lain untuk membangun sebuah sistem informasi seperti persediaan, pegawai, akuntansi, pemasaran, produksi, reservasi, layanan pelanggan yang digunakan dalam perusahaan retail, perbankan, perhotelan, pariwisata, rumah sakit, institusi pendidikan dan lain sebagainya. Data dalam *database* disusun berdasarkan sistem hierarki yang unik yaitu :

1. *Database*, merupakan kumpulan *file* yang saling terkait satu sama lain. Kumpulan *file* yang tidak saling terkait antara satu dengan lainnya tidak dapat disebut basis data.
2. *File* yaitu kumpulan dari *record* yang saling terkait dan memiliki *format field* yang sama dan sejenis.
3. *Record* yaitu kumpulan *field* yang menggambarkan suatu unit data individu tertentu.
4. *Field* yaitu atribut dari *record* yang menunjukkan suatu *item* dari data seperti nama, alamat dan lain sebagainya.

5. *Byte* yaitu atribut dari *field* yang berupa huruf yang membentuk nilai dari sebuah *field*. Huruf tersebut dapat berupa numerik maupun abjad atau karakter khusus.

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

1. Sistem *database* tunggal

Pada arsitektur ini, *database* dan aplikasinya diletakkan pada komputer yang sama yang tidak berada dalam lingkungan jaringan, sehingga *database* itu hanya dapat diakses oleh aplikasi tunggal. Sistem ini biasanya digunakan oleh perusahaan kecil.

2. Sistem *database* terpusat

Pada arsitektur ini, lokasi *database* secara fisik berada dalam komputer pusat dalam suatu lingkungan jaringan. Meskipun pemasukan dan akses data dapat dilakukan dari berbagai terminal yang terhubung dengan komputer tersebut, namun proses pengolahan data hanya berlangsung di komputer pusat. Dengan sistem ini, komputer pusat menjadi titik kritis dari proses pengolahan *database*. Bila komputer pusat terganggu, maka secara keseluruhan sistem informasi akan terganggu.

3. Sistem *database* terdistribusi

Pada arsitektur ini salinan *database*, baik sebagian maupun secara keseluruhan terdistribusi di beberapa lokasi. Pada model ini, titik kritis pada sistem terpusat dapat dihindari. Namun pada sistem ini, tantangan terbesar

yang dihadapi adalah proses pengintegrasian untuk menjaga konsistensi data.

#### II.4. Normalisasi

Janner Simarmata (2007: 77-84) Normalisasi merupakan suatu proses pengelompokan elemen data menjadi table-table yang menunjukkan entitas dan 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

1. Bentuk Normal Kesatu (1 NF).

Skema tabel disebut dalam bentuk normal pertama jika nilai atribut tidak terpisahkan.

2. Bentuk Normal Kedua (2 NF)

Skema tabel disebut bentuk normal kedua jika semua atribut informasi (atribut yang tidak memiliki kunci manapun) 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. *Boyce Codd Normal Form (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*.

## **II.5. *Unified Modeling Language***

Ardhian Agung (2009 : 65-66) menyatakan UML merupakan diagram yang saling terkait oleh karena itu perlu adanya kekonsistenan rancangan diagram yang satu dengan lainnya, bukan asal menggambar. Berikut adalah keterkaitan diagram-diagram pada UML beserta urutan pembuatannya. UML sendiri juga memberikan standar penulisan sebuah sistem *blue print*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen-komponen yang diperlukan dalam sistem software.

Bahasa pemrograman berorientasi objek yang pertama dikembangkan dikenal dengan nama Simula-67 yang dikembangkan pada tahun 1967. Bahasa pemrograman ini kurang berkembang dan dikembangkan lebih lanjut, namun dengan kemunculannya telah memberikan sumbangan yang besar pada *developer* pengembang bahasa pemrograman berorientasi objek selanjutnya. Perkembangan aktif dari pemrograman berorientasi objek mulai menggeliat ketika berkembangnya bahasa pemrograman Smalltalk pada awal 1980-an yang kemudian diikuti dengan perkembangan bahasa pemrograman berorientasi objek yang lainnya seperti C objek, C++, Eiffel, dan CLOS. Secara aktual, penggunaan bahasa pemrograman berorientasi objek pada saat itu

masih terbatas, namun telah banyak menarik perhatian di saat itu. Sekitar lima tahun setelah Smalltalk berkembang, maka berkembang pula metode pengembangan berorientasi objek.

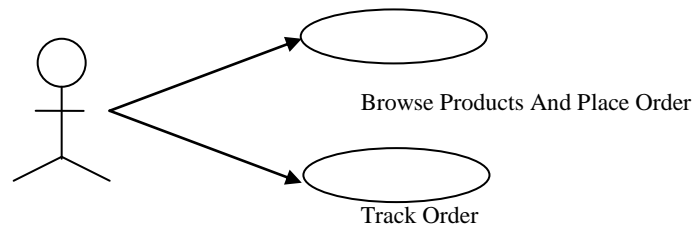
Ardhian Agung (2009: 65-66) menyatakan keterkaitan UML beserta urutan pembuatannya adalah sebagai berikut :

1. Diagram *Use Case*
2. Diagram Kelas
3. Diagram Objek
4. Diagram Interaksi
5. Diagram Status
6. Diagram Aktivitas
7. Diagram Komponen
8. Diagram *Deployment*

Berikut ini adalah keterangan dari masing-masing diagram pada *Unified Modelling Language (UML)*.

### **II.5.1 Diagram *Use Case***

Ardhian Agung (2009 : 65-66) menyatakan *Use Case* Diagram atau diagram *use case* merupakan pemodelan untuk menggambarkan kelakuan (*behavior*) sistem yang akan dibuat. Diagram *use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Dengan pengertian yang cepat, diagram *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

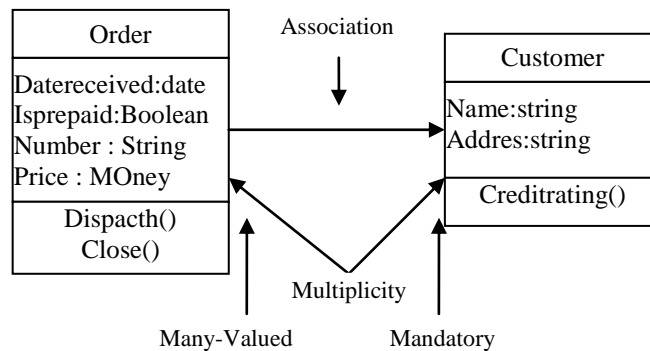


**Gambar II.1: Contoh Diagram Use Case**

Sumber : Ardhian Agung (2009 : 78)

### II.5.2. Diagram Kelas

Ardhian Agung (2009 : 89) menyatakan Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

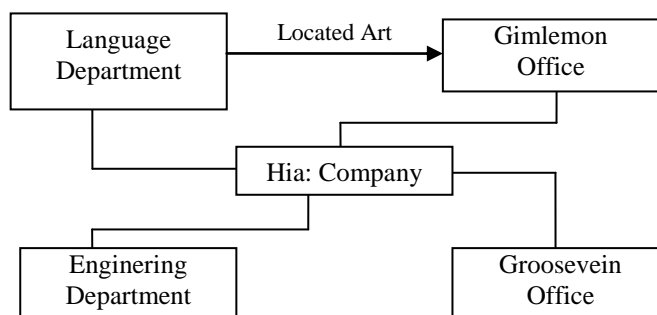


**Gambar II.2: Contoh Diagram Kelas**

Sumber : Ardhian Agung (2009 : 92)

### II.5.3. Diagram Objek

Ardhian Agung (2009: 98) menyatakan Diagram objek menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem. Pada diagram objek harus dipastikan semua kelas yang sudah didefinisikan pada diagram kelas harus dipakai objeknya, karena jika tidak, pendefinisian kelas itu tidak dapat dipertanggungjawabkan.



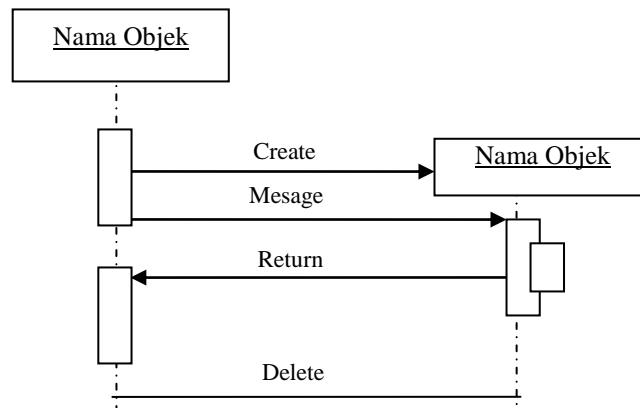
**Gambar II.3 : Contoh Diagram Objek**

Sumber : Ardhian Agung (2009 : 99)

#### II.5.4. Diagram Interaksi

Ardhian Agung (2009: 103) Diagram interaksi merupakan diagram perilaku dari sebuah use case ketika antar objek saling berinteraksi dalam melengkapinya tugas-tugasnya dan menggambarkan aliran message atau pesan. Dua jenis diagram interaksi adalah Diagram Sekuen dan Diagram Kolaborasi Diagram interaksi atau interaction diagram digunakan untuk memodelkan interaksi objek di dalam sebuah use case (proses). Diagram interaksi memperlihatkan interaksi yang memuat himpunan dari objek dan relasi yang terjadi antar objek tersebut, termasuk juga bagaimana message (pesan) mengalir diantar objek. Diagram interaksi terdiri dari beberapa diagram antara lain diagram sekuen dan diagram kolaborasi.

Diagram sekuen menggambarkan kelakuan/perilaku objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang di instansiasi menjadi objek itu. Banyaknya diagram sekuen yang harus digambar adalah sebanyak pendefinisian use case yang memiliki proses sendiri.

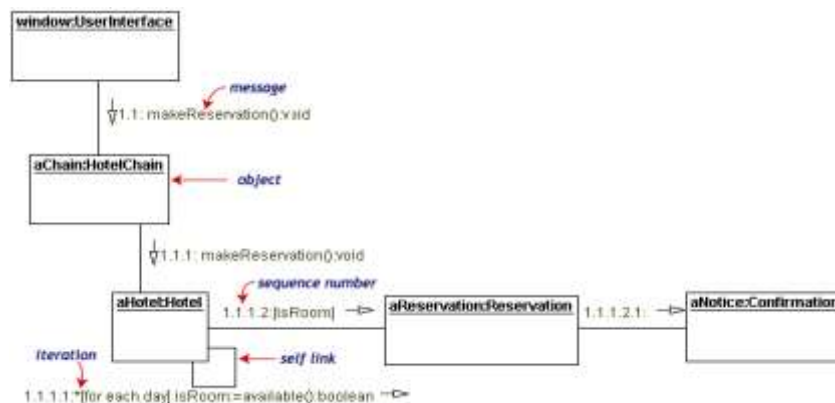


**Gambar II.4: Contoh Diagram Interaksi**

Sumber : Ardhan Agung (2009 : 106)

### II.5.5. Diagram Collaboration

Ardhan Agung (2009: 118) diagram kolaborasi mengelompokkan message pada kumpulan diagramsekuen menjadi sebuah diagram. Dalam diagram kolaborasi yang dituliskanadalah operasi/metode yang dijalankan antara objek yang satu dan objek lainnya secara keseluruhan, oleh karena itu dapat diambil dari jalannyainteraksi pada semua diagram sekuen. Penomoran metode dapat dilakukanberdasarkan urutan dijalankannya metode/operasi diantara objek yang satudengan objek lainnya atau objek itu sendiri.Untuk menunjukkan sebuah pesan/message.

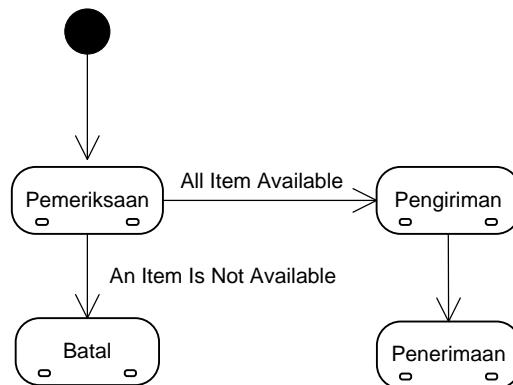


**Gambar II.5 : Contoh Diagram Collaboration**

Sumber : Ardhian Agung (2009 : 120)

### II.5.6. Diagram Status

Ardhian Agung (2009: 126) diagram status atau state diagram atau statechart diagram menunjukkan kondisi yang dapat dialami atau terjadi pada sebuah objek sehingga setiap objek memiliki sebuah diagram status. Diagram status diadopsi dari penggambaran kondisi mesin status (state machine) yang menggambarkan status apa saja yang dialami oleh mesin, misalnya mesin pembelian kopi dengan uang koin. Diagram Status menggambarkan seluruh state/status yang memungkinkan obyek-obyek dalam Class dapat dimiliki dan kejadian-kejadian yang menyebabkan status berubah. Perubahan dalam suatu state disebut juga transisi(transition). Suatu transisi juga dapat memiliki sebuah aksi yang dihubungkan pada status, lebih spesifik apa yang harus dilakukan dalam hubungannya dengan transisi status. Pada diagram ini, perilaku sistem ditunjukkan. Sebuah status adalah kondisi selama hidup objek atau interaksi selama memenuhi suatu kondisi, melaksanakan suatu aksi, atau menunggu suatu kejadian.

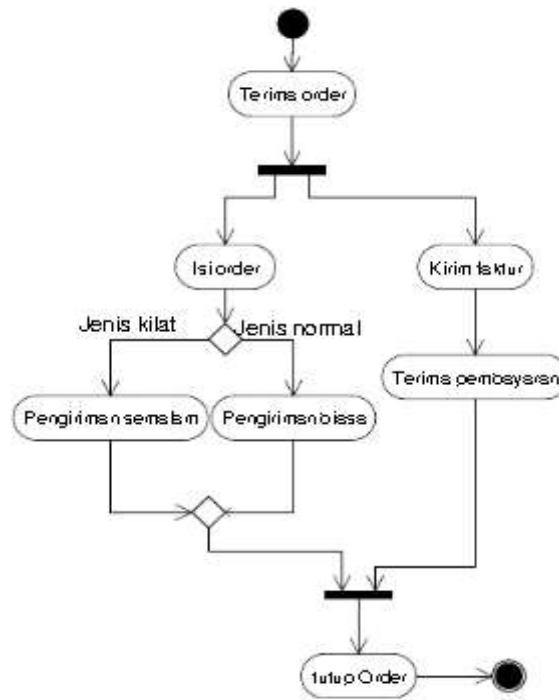


**Gambar II.6 : Contoh Diagram Status**

Sumber : Ardhian Agung (2009 : 127)

### II.5.7. Diagram Aktivitas

Ardhian Agung (2009: 140) diagram aktivitas mendeskripsikan aliran kerja dari perilaku sistem. Diagram ini hampir sama dengan diagram status karena kegiatan-kegiatannya merupakan status suatu pekerjaan dengan menunjukkan kegiatan yang dilakukan secara berurutan. Sebaiknya diagram aktivitas digunakan untuk melengkapi diagram lain seperti diagram interaksi dan diagram status, karena diagram aktivitas dapat mengetahui aliran sistem yang akan dirancang. Selain itu diagram aktivitas bermanfaat untuk menganalisis use case melalui penggambaran aksi-aksi yang dibutuhkan, penggambaran algoritma berurutan yang kompleks, dan pemodelan aplikasi dengan proses paralel. Tetapi diagram aktivitas tidak menunjukkan bagaimana objek berperilaku atau objek berkolaborasi secara detail.



**Gambar II.7: Contoh Diagram Aktivitas**

Sumber : Ardhian Agung (2009 : 141)

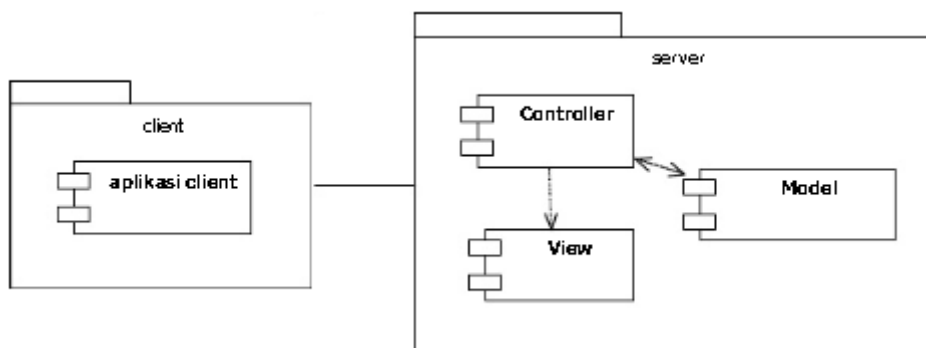
### II.5.8. Diagram Komponen

Ardhian Agung (2009: 150) diagram komponen atau component diagram dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. Pengertian komponen sendiri dalam UML adalah hal-hal fisik dari sistem yang akan dimodelkan dan ada ketika sistem dieksekusi. Diantara contoh komponen dasar pada sebuah sistem yaitu :

- a. Komponen *user interface* yang menangani tampilanKomponen
- b. *bussiness processing* yang menangani fungsi-fungsi proses bisnis.
- c. Komponen data yang menangani manipulasi data.
- d. Komponen *security* yang menangani keamanan sistem

Contoh lain komponen dalam perangkat lunak yaitu operating sistem, bahasapemrograman, obyek-obyek *library*, *file executable*, COM+. Termasuk

jugadapat dimodelkan sebagai komponen adalah tabel, *file (source code)* dan dokumen



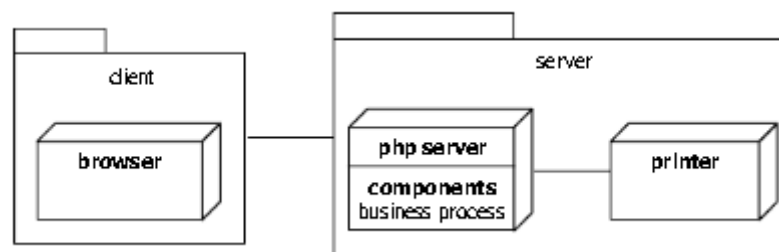
**Gambar II.8 : Contoh Diagram Komponen**

Sumber : Ardhian Agung (2009 : 151)

### II.5.9. Diagram *Deployment*

Diagram deployment atau deployment diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram deployment mewakili pandangan pengembangan sistem sehinggahanya akan ada satu diagram deployment untuk satu sistem. Diagram deployment terdiri dari node dan node merupakan perangkat keras fisik yang digunakan untuk menyebarkan aplikasi. Diagram deployment banyak digunakan oleh system engineer .

Tiap node pada diagram deployment mewakili satu unit komputasi sistem yang dalam banyak hal merupakan bagian dari perangkat keras. Diagram deployment umumnya memiliki node dan hubungan kebergantungan.



**Gambar II.9: Contoh Diagram Deployment**

## II.6. Microsoft Visual Studio 2008

Muhammad Sadeli (2009: 2) *Visual Studio 2008* merupakan suatu perangkat lunak yang dapat digunakan untuk pengembangan berbagai macam aplikasi yang memiliki berbagai macam tipe antara lain aplikasi *desktop (windows form, command line (console)*, aplikasi *web, windows mobile (pocket pc)*. *Visual studio 2008* diluncurkan microsoft pada tanggal 19 november 2007 dan memiliki lebih dari satu kompiler, *SDK (Software Development Kit)* dan dokumentasi tutorial (*MSDN Library*).

Untuk memulai program *Visual Studio 2008* dapat dilakukan dengan langkah-langkah di bawah ini:

1. Klik tombol *start* pada *windows taskbar*.
2. Klik menu *All Program > Microsoft Visual Studio 2008 > Microsoft Visual Studio 2008*.

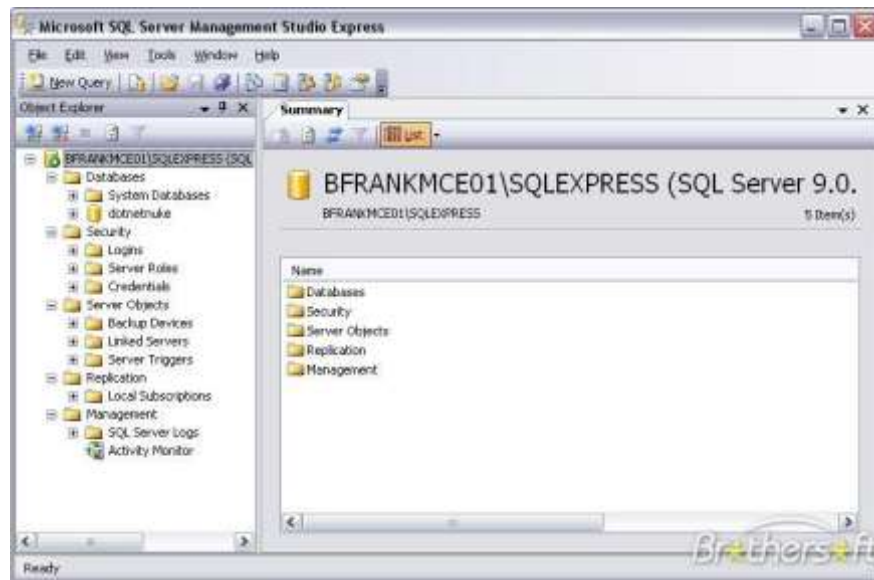


tabel atau lebih dan memiliki nama yang berbeda untuk masing-masing tabel. Masing-masing tabel memiliki satu kolom (*field*) atau lebih dan memiliki baris (*record*). *Query* digunakan untuk mengakses dan mengolah database. SQL terdiri dari 5 bagian utama, yaitu:

1. *Retrieving data*: perintah untuk menampilkan data dari database (SELECT).
2. *Data Definition Language* (DDL): bahasa yang digunakan untuk membuat dan menghapus tabel atau database itu sendiri (CREATE, DROP, ALTER)
3. *Data Manipulation Language*(DML): merupakan bahasa untuk memanipulasi/ mengubah isi tabel (INSERT, DELETE, UPDATE)
4. *Data Control Language* (DCL): bahasa yang berhubungan dengan pengendalian akses ke database (GRANT, REVOKE)
5. *Data Transaction Language* (DTL): bahasa yang digunakan untuk transaksi database (COMMIT, ROLLBACK).

Untuk memulai program *SQL Server 2005* dapat dilakukan dengan langkah-langkah di bawah ini:

1. Klik tombol *start* pada *windows taskbar*.
2. Klik menu *All Program > Microsoft SQL Server 2005 > SQL Server Management Studio*.



**Gambar II.12 : Jendela Microsoft SQL Server 2005**