

BAB II

TINJAUAN PUSTAKA

II.1. Sistem Informasi

Sistem informasi dapat diartikan sebagai penggunaan teknologi komputer yang meliputi perangkat hardware dan software yang telah dirancang untuk mengubah data menjadi suatu informasi yang berguna kepada pengguna informasi di dalam sebuah organisasi atau perusahaan (Bodnar, 2000:4).

Definisi sistem informasi menurut Henry C. Lucas adalah suatu kegiatan dari prosedur-prosedur yang diorganisasikan, bilamana dieksekusi akan menyediakan suatu informasi yang digunakan untuk mendukung pengambilan keputusan dan pengendalian di dalam organisasi. Sedangkan John F. Nash dan Martin B. Roberts menyebutkan bahwa sistem informasi adalah suatu kombinasi orang-orang, fasilitas, teknologi, media, prosedur-prosedur dan pengendalian yang ditujukan untuk mendapatkan suatu jalur komunikasi yang penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada para manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan suatu dasar dengan tujuan untuk pengambilan keputusan yang cerdas (Jogianto H. M., 1996:16)

Sistem Informasi (information system) adalah serangkaian prosedur secara formal dimana data dikumpulkan, kemudian diproses menjadi suatu informasi dan didistribusikan kepada para pengguna (James A. Hall, 2006:9).

Berdasarkan beberapa pendapat tentang definisi sistem informasi, maka dapat ditarik kesimpulan bahwa sistem informasi merupakan suatu rangkaian kegiatan pengumpulan dan pengolahan data untuk menghasilkan output berupa informasi yang dapat diterima dan digunakan oleh pihak yang berkepentingan. Sistem Informasi dalam sebuah organisasi atau perusahaan pada dasarnya dikelompokkan menjadi dua, yaitu : sistem informasi manajemen (SIM) dan sistem informasi akuntansi (SIA). Sistem informasi manajemen (SIM) sifatnya lebih terperinci dan biasanya digunakan untuk kepentingan internal perusahaan/pengelola bisnis, sedangkan sistem informasi akuntansi (SIA) biasanya lebih digunakan untuk kepentingan eksternal.

II.2. Sistem Informasi Akuntansi

Terdapat dua kelompok pendekatan di dalam mendefinisikan sistem, yaitu yang menekankan pada prosedurnya dan yang menekankan pada komponen atau elemennya. Pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan sistem sebagai suatu jaringan kerja dari prosedur – prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Sedangkan pendekatan sistem yang lebih menekankan pada elemen atau komponennya mendefinisikan sistem sebagai kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu (Jogiyanto, 1993:2). Menurut Romney dan Steinbart, sistem merupakan kumpulan dari dua atau lebih komponen yang saling terkait dan berinteraksi untuk mencapai tujuan (Romney dan Steinbart, 2003:2).

Suatu sistem akan menghasilkan informasi yang berguna bagi penggunanya. Informasi tersebut diperoleh dari data yang telah diproses. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Sedangkan informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya (Jogiyanto, 1993:8). Data merupakan bentuk yang masih mentah yang belum dapat bercerita banyak, sehingga perlu diolah lebih lanjut. Data akan ditangkap sebagai input, diproses, dan akan menghasilkan output berupa informasi. Proses pengolahan data menjadi informasi disebut dengan siklus informasi. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya. Informasi merupakan hal yang sangat penting bagi manajemen di dalam pengambilan keputusan. Suatu informasi dikatakan berkualitas bila informasi tersebut akurat, tepat waktu dan relevan atau bermanfaat (Jogiyanto, 1993:10).

Sistem informasi didefinisikan oleh Robert A. Leitch dan K. Roscoe Davis dalam buku Jogiyanto sebagai suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan (Jogiyanto, 1993:11). John Burch dan Gary Grudnitski dalam buku Jogiyanto mengemukakan bahwa sistem informasi terdiri dari komponen-komponen yang disebutnya dengan istilah blok bangunan (*building block*), yaitu: blok masukan (*input block*), blok model (*model block*), blok keluaran (*output block*), blok teknologi (*technology block*), blok basis data (*database block*) dan blok kendali (*control block*). Sebagai suatu

sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sasarannya (Jogiyanto, 1993:12).

Setiap perusahaan menerapkan akuntansi sebagai alat komunikasi bisnis. Akuntansi merupakan proses pencatatan (*recording*), pengelompokan (*classifying*), perangkuman (*summarizing*), dan pelaporan (*reporting*) dari kegiatan transaksi perusahaan. Tujuan akhir dari kegiatan akuntansi adalah penerbitan laporan-laporan keuangan (Jogiyanto, 1993:17). Sistem Informasi Akuntansi (SIA) didefinisikan oleh Stephen A. Moscovice dan Mark G. Simkin dalam buku Jogiyanto sebagai suatu komponen organisasi yang mengumpulkan, mengklasifikasikan, memproses, menganalisis, mengkomunikasikan informasi pengambilan keputusan dengan orientasi finansial yang relevan bagi pihak-pihak dalam perusahaan. Sedangkan SIA menurut Romney dan Steinbart adalah sistem yang mengumpulkan, merekam, menyimpan, dan memproses data akuntansi dan data lainnya untuk menghasilkan informasi bagi para pengambil keputusan.

II.3. Metodologi Pengembangan Sistem (SDLC)

Dalam rekayasa perangkat lunak, *metodologi pengembangan sistem* adalah suatu kerangka kerja yang digunakan untuk menstrukturkan, merencanakan, dan mengendalikan proses pengembangan suatu sistem informasi. Banyak ragam kerangka kerja yang telah dikembangkan selama ini, yang masing-masing memiliki kekuatan dan kelemahan sendirisendiri. Beberapa contoh metodologi pengembangan perangkat lunak yang tersedia, antara lain *waterfall*,

prototyping, incremental, spiral, RAD. Suatu metodologi pengembangan sistem tidak selamanya cocok untuk digunakan pada semua proyek pengembangan sistem. Masing-masing metodologi mungkin cocok diterapkan untuk suatu proyek tertentu, berdasarkan berbagai pertimbangan teknis, organisasi, proyek, serta tim

SDLC dalam rekayasa sistem dan rekayasa perangkat lunak, adalah proses pembuatan dan perubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem-sistem tersebut. Konsep ini umumnya merujuk pada sistem komputer atau informasi. SDLC juga merupakan pola yang diambil untuk mengembangkan sistem perangkat lunak, yang terdiri dari tahap-tahap :

- Analisis (*analysis*) dan perencanaan (*planning*)
Dalam tahap ini, dibutuhkan informasi untuk melakukan pembelian, pengembangan dan memodifikasi sistem.
- Desain (*design*).
Dalam tahap ini, kebutuhan pengguna diidentifikasi dan dievaluasi melalui alternative rancangan dan diterjemahkan ke dalam spesifikasi menggunakan kode dan program komputer, rancangan dokumen input dan output, membuat file dan database, mengembangkan prosedur, dan membangun pengendalian dalam sistem baru.
- Implementasi (*implementation*).
Dalam tahap ini sistem mulai diterapkan. Standar dan pengendalian sistem baru mulai dibangun dan dokumentasi sistem telah lengkap.

- Uji coba (*testing*).

Dalam tahap ini analis melakukan uji coba sistem yang mulai diterapkan dan melakukan evaluasi.

- Operasi dan pemeliharaan (*maintenance*).

Selama tahap ini, review dan modifikasi sistem jika terjadi masalah yang timbul dari sistem yang baru (Romney dan Steinbart, 2003).

Dalam rekayasa perangkat lunak, konsep SDLC mendasari berbagai jenis metodologi pengembangan perangkat lunak. Metodologi-metodologi ini membentuk suatu kerangka kerja untuk perencanaan dan pengendalian pembuatan sistem informasi, yaitu proses pengembangan perangkat lunak.

II.4.1. Siklus Hidup Perusahaan

Siklus hidup perusahaan yaitu suatu grafik yang menggambarkan riwayat perusahaan sejak perusahaan itu berdiri sampai dengan ditarik dari pasaran atau bangkrut. Siklus hidup perusahaan sebagai suatu konsep mengenai dinamika bersaing suatu perusahaan. Menurut Gup dan Agrawal (1996) dalam Gumanti dan Puspitasari (2008) siklus hidup perusahaan dianggap sebagai nilai strategik bagi suatu perusahaan, maka seorang manajer harus dapat menentukan di mana posisi perusahaan pada tahapan siklus hidup perusahaan. Lindanaty (2011) membagi tahapan siklus kehidupan perusahaan sebagai berikut:

- Tahap pendirian (*establishment or start-up*)

Tahap ini adalah tahap permulaan bagi setiap perusahaan baru. Segala sesuatu yang mendukung operasi perusahaan bersifat baru, misalnya

tenaga kerja, lokasi, dan fasilitas lainnya. Kebutuhan modalnya dipenuhi oleh pemilik ditambah dengan dana pinjaman dari bank.

- Tahap ekspansi

Pada tahap ini perusahaan sudah memiliki pelanggan dan cukup mampu memposisikan keberadaannya di pasar untuk itu dibutuhkan dana yang tidak sedikit. Pada tahap ini kebutuhan dana eksternal sangat tinggi karena aliran kas masuk relatif kecil.

- Tahap kedewasaan (*maturity*)

Perusahaan yang memasuki tahap ini mempunyai dua ciri yaitu: pertama, peningkatan laba dan aliran kas yang cepat sebagai cermin dari keberhasilan investasi masa lalu. Dan kedua, kebutuhan dana untuk investasi ada produk dan proyek baru akan mulai menurun.

- Tahap penurunan (*declining*)

Pada tahap ini ciri utama yang dapat diketahui adalah penurunan yang stabil terhadap pendapatan dan laba sebagai konsekuensi dari kedewasaan perusahaan dan masuknya pesaing-pesaing baru. Pada tahap ini kebutuhan dana eksternal menurun drastis karena proyek-proyek atau investasi baru juga menurun dan jumlah dana internal yang tersedia di perusahaan sangat besar.

Penetapan siklus kehidupan perusahaan menurut Gup dan Agrawal (1996) dalam Gumanti dan Puspitasari (2008) didasarkan pada pertumbuhan penjualan yang dihitung dengan rumus:

Setelah pertumbuhan penjualan diketahui dari rumus tersebut, maka perusahaan yang menjadi sampel penelitian dikelompokkan pertumbuhan penjualannya ke dalam tiap tahapan siklus kehidupan dengan mengikuti kriteria seperti yang digunakan oleh Anthony dan Ramesh (1998) serta Gup dan Agrawal (1996) dalam Gumanti dan Puspitasari (2008) sebagai berikut:

Tabel 2.1 Tahap Siklus Hidup dan Rata-rata Pertumbuhan Penj. 5

No	Tahap Siklus Hidup	Rata-rata Pertumbuhan Penj. 5 tahun
1	<i>Start-up</i>	>50%
2	Ekspansi awal	20-50%
3	Ekspansi akhir	10-20%
4	<i>Maturity</i>	1-10%
5	<i>Decline</i>	<1%

Sumber : Gup dan Agrawal (1996); dalam Gumanti dan Puspitasari (2008)

II.5. Penentuan Harga Pokok

Penentuan harga pokok produksi adalah proses pembebanan biaya produksi, yang terdiri dari biaya bahan baku, biaya tenaga kerja dan biaya *overhead* pabrik, terhadap produk yang dihasilkan. Pemahaman terhadap proses tersebut merupakan hal yang sangat penting bagi tiap manajer, karena cara-cara yang digunakan dalam penentuan harga pokok akan mempunyai pengaruh yang besar terhadap laba rugi yang dilaporkan, serta terhadap aktiva mempunyai pengaruh yang besar terhadap laba rugi yang dilaporkan, serta aktiva lancar dalam neraca.

Ada dua sistem akuntansi biaya, yaitu sistem akuntansi biaya sesungguhnya (*historical cost*) dan sistem biaya standar dan sistem biaya taksiran (*estimated cost*). Pendekatan yang paling umum digunakan dalam penentuan harga pokok produksi adalah pendekatan *full costing* atau *absorption cost* yang dalam menghitung harga pokok produksi memperhitungkan semua biaya produksi baik yang bersifat variabel maupun yang bersifat tetap. Alternatif pendekatan yang lain adalah *direct costing* atau *variable costing* maupun yang dalam menghitung harga pokok produksi hanya mempertimbangkan biaya produksi variabel sedangkan seluruh biaya tetap dianggap sebagai periode (*period cost*) Sistem akuntansi manajemen harus dirancang sedemikian rupa sehingga manajemen dapat mengetahui jumlah biaya yang terjadi untuk membuat sesuatu atau membuat produk tertentu. Informasi biaya untuk membuat sesuatu merupakan informasi yang sangat penting bagi akuntan manajemen maupun akuntan keuangan.

II. 5. 1. Variable Costing

Pembahasan sistem penentuan harga pokok produk selama ini difokuskan pada *absorption cost* yang dalam menentukan harga pokok produk memasukan seluruh biaya produksi yang terjadi dipabrik, yang meliputi bahan baku, tenaga kerja langsung dan *overhead* pabrik baik variabel maupun tetap. Sistem tersebut terutama ditujukan untuk pelaporan keuangan eksternal dan pajak.

Variable costing mempunyai beberapa keunggulan untuk pelaporan internal, karena menyajikan informasi dalam bentuk yang banyak manfaatnya

bagi manajemen, bentuk laporan yang dibutuhkan untuk analisis biaya – volume – laba atau analisis *break even*. Manajer tidak akan dapat memprediksi laba masa depan dengan menggunakan *absorption costing* dengan alasan bahwa manajer tidak hanya memprediksi penjualan masa depan tetapi juga produksi, dan biaya harus dipisahkan antara komponen biaya tetap dan variabel sehingga dapat ditentukan individu yang harus bertanggung jawab untuk mengendalikannya.

Dengan *variable costing* produksi tidak ada pengaruhnya terhadap laba sehingga manajer dapat mengkonsentrasikan pada pengaruh penjualan tanpa harus memperhatikan pada produksi. Jika penjualan turun dari suatu periode ke periode yang lain maka laba juga turun. Dengan menggunakan *absorption costing*, laba mungkin menunjukkan kenaikan dalam periode terjadi penurunan penjualan jika produksi lebih besar dari pada penjualan. Sebagian manajer dalam aktivitasnya memperoleh laba perusahaan membiasakan untuk memikirkan penjualan bukan produksi.

II. 6. Visual Basic

Microsoft Visual Basic merupakan sebuah bahasa pemrograman yang menawarkan *Integrated Development Environment (IDE)* visual untuk membuat program perangkat lunak berbasis sistem operasi *Microsoft Windows* menggunakan model pemrograman (COM).

Visual Basic merupakan turunan bahasa pemrograman BASIC dan menawarkan pengembangan perangkat lunak komputer berbasis grafik dengan cepat. Beberapa bahasa skrip seperti *Visual Basic for Application (VBA)* dan

Visual Basic Scripting Edition (VB-Script) mirip seperti halnya *Visual Basic*, tetapi cara kerjanya yang berbeda.

Microsoft Visual Basic.NET adalah sebuah alat untuk mengembangkan dan membangun aplikasi yang bergerak diatas sistem *.NET Framework* dengan menggunakan bahasa BASIC. Dengan alat ini , para programmer dapat membangun aplikasi *Windows Form*, aplikasi web berbasis ASP.NET dan aplikasi *command-line*. Alat ini dapat diperoleh secara terpisah dari beberapa produk lainnya (seperti *Microsoft Visual C++*, *Visual C#* atau *Visual J#*) atau dapat juga diperoleh secara terpadu dalam *Microsoft Visual Studio.NET*.

Bahasa *Visual Basic.NET* sendiri menganut paradigma bahasa pemrograman berorientasi objek yang dapat dilihat sebagai evolusi dari *Microsoft Visual Basic* versi sebelumnya yang diimplementasikan diatas *.NET Framework*.

II. 7. SQL-Server

Microsoft SQL Server adalah sebuah sistem manajemen basis data relasional (RDBMS) produk *Microsoft*. Bahasa kueri utamanya adalah *transact-SQL* yang merupakan implementasi dari SQL standar ANSI/ISO yang digunakan oleh *Microsoft* dan *Sybase*. Umumnya SQL Server digunakan didunia bisnis yang memiliki basis data berskala kecil sampai menengah, tetapi kemudian berkembang dengan digunakannya SQL erver pada basis data besar.

Microsoft SQL Server juga mendukung ODBC (*Open Database Connectivity*) dan mempunyai driver JDBC untuk bahasa pemrograman Java. Fitur

lain dari SQL Server adalah kemampuannya membuat basis data *mirroring* dan *clustering*

II.8. Client Server

Client merupakan sistem atau proses yang melakukan suatu permintaan data atau layanan ke server, sedangkan *server* adalah sistem atau proses yang menyediakan data atau layanan yang diminta oleh *client*. *Client* memiliki fungsi lainnya seperti mengatur *user interface*, menerima dan memeriksa *syntax* input dari pemakai, menyediakan akses basis data secara bersamaan, serta menyediakan control *recovery*. *Server* juga memiliki fungsi lainnya seperti menerima dan memproses basis data yang diminta dari *client*, memeriksa otorisasi, menjamin tidak terjadi pelanggaran terhadap *integrity constraint*, melakukan *query*/pemrosesan *update* dan memindahkan *response* ke *client*, serta memelihara data *dictionary*.

Client server adalah pembagian kerja antara *server* dan *client* yang mengakses *server* dalam suatu jaringan. Maka arsitektur *client-server* adalah desain sebuah aplikasi terdiri dari *client* dan *server* yang saling berkomunikasi ketika mengakses *server* dalam suatu jaringan.

Dalam konteks basis data, *client* mengatur *interface* yang berfungsi sebagai *workstation* tempat menjalankan aplikasi basis data. *Client* menerima permintaan pemakai, memeriksa *syntax* dan *generate* kebutuhan basis data dalam SQL atau bahasa yang lain. Kemudian meneruskan pesan ke *server*, menunggu *response* dan bentuk *response* untuk pemakai akhir. *Server*

menerima dan memproses permintaan basis data kemudian mengembalikan hasil ke *client*.

Penggunaan aplikasi berbasis *client-server* memiliki keuntungan seperti memungkinkan akses basis data yang besar, menaikkan performa kinerja, dapat memproses aplikasi secara paralel, biaya untuk *hardware* dapat dikurangi, dari dua atau lebih komputer hanya *server* yang membutuhkan ruang penyimpanan besar, biaya komunikasi berkurang, serta meningkatkan kekonsistenan. (*Frizal Luthfi Hadyan, Doro Edi, 2014:65*)

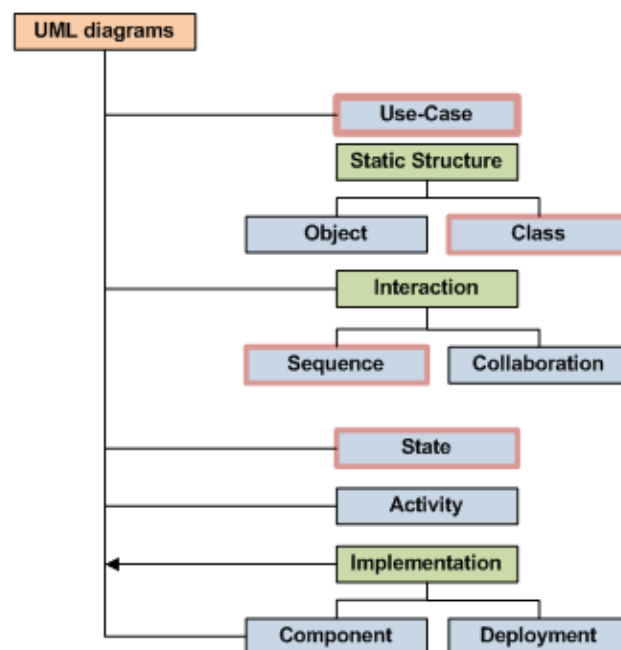
II.9. UML (*Unified Modelling Language*)

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.

UML diagram memiliki tujuan utama untuk membantu tim pengembangan proyek berkomunikasi, mengeksplorasi potensi desain, dan memvalidasi desain arsitektur perangkat lunak atau pembuat program. Komponen atau notasi UML diturunkan dari 3 (tiga) notasi yang telah ada sebelumnya yaitu Grady Booch, OOD (*Object-Oriented Design*), Jim Rumbaugh, OMT (*Object Modelling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).

UML mempunyai tiga kategori utama yaitu *struktur diagram*, *behaviour diagram* dan *interaction diagram*. Dimana masing-masing kategori tersebut memiliki diagram yang menjelaskan arsitektur sistem dan saling terintegrasi. (Haviluddin , 2011 ; 1)

Menurut Haviluddin (2011) Secara filosofi UML diilhami oleh konsep yang telah ada yaitu konsep permodelan *Object Oriented* karena konsep ini menganalogikan sistem seperti kehidupan nyata yang didominasi oleh obyek dan digambarkan atau dinotasikan dalam simbol-simbol yang cukup spesifik. Berikut gambar dari diagram UML



Gambar II. 3. Diagram UML
(Haviluddin , 2011 ; 2)

Komponen-komponen UML

Sejauh ini para pakar merasa lebih mudah dalam menganalisa dan mendesain atau memodelkan suatu sistem karena UML memiliki seperangkat aturan dan notasi dalam bentuk grafis yang cukup spesifik (Sugrue J. 2009).

Komponen atau notasi UML diturunkan dari 3 (tiga) notasi yang telah ada sebelumnya yaitu Grady Booch, OOD (*Object-Oriented Design*), Jim Rumbaugh, OMT (*Object Modelling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*). (Haviluddin, 2011 ; 3)

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

A. 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 :

- *Class Diagram*

Class diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas.

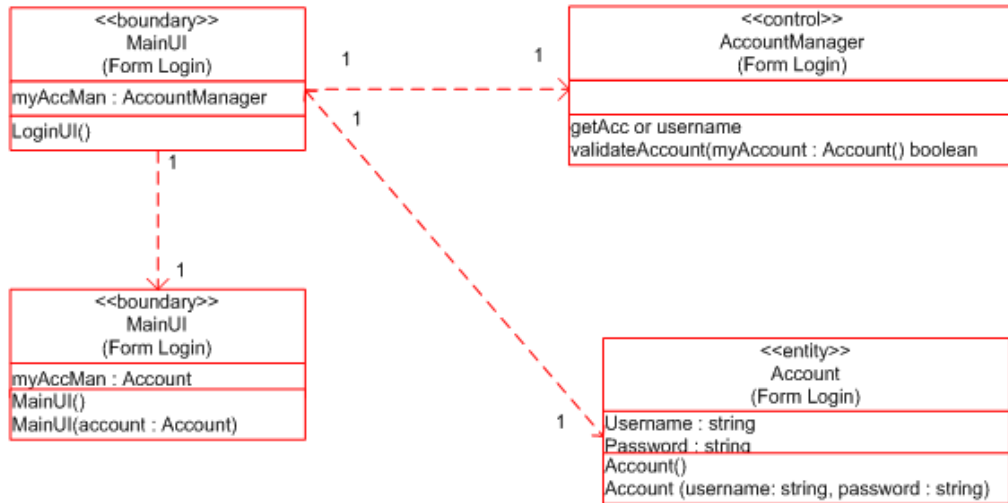
Class diagram membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai.

Selama tahap desain, class diagram berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat.

Class memiliki tiga area pokok :

- a. Nama (dan stereotype)
- b. Atribut

c. Metoda

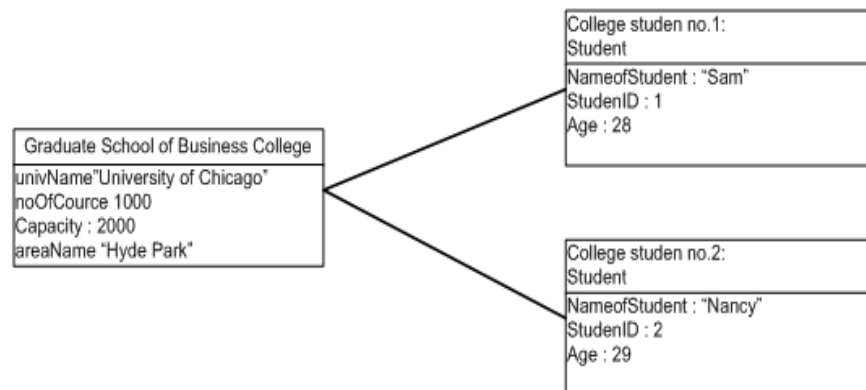


II.4. Contoh Notasi Class Diagram
 (Sumber : Haciluddin , 2011 ; 3)

- *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.

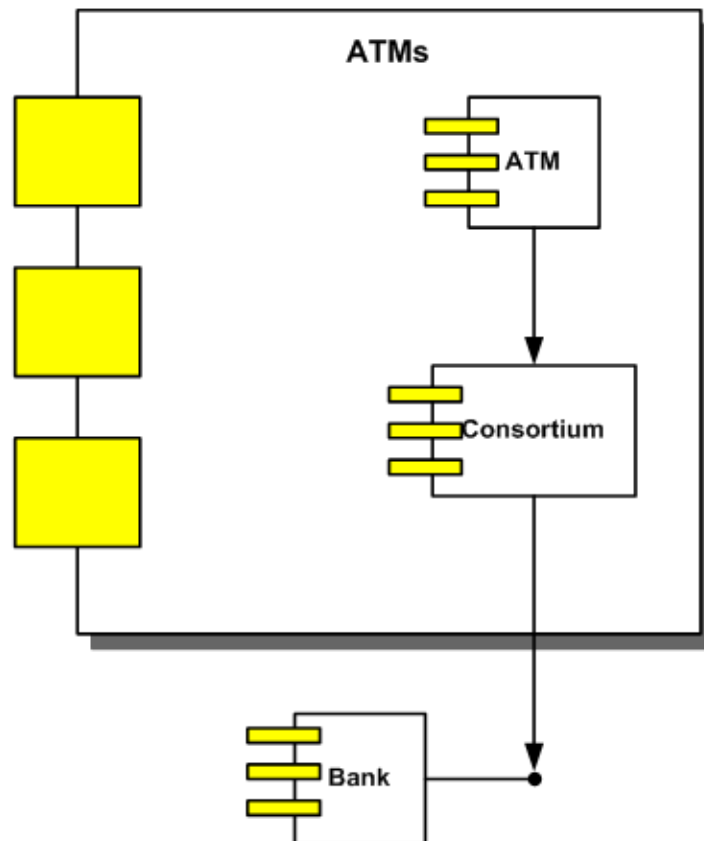
Berikut notasi *object diagram* :



II.5. Contoh Notasi Object Diagram
 (Sumber : Haviluddin , 2011 ; 3)

- *Component diagram*

Component diagram menggambarkan struktur fisik dari kode, pemetaan pandangan logis dari kelas proyek untuk kode aktual di mana logika ini dilaksanakan.

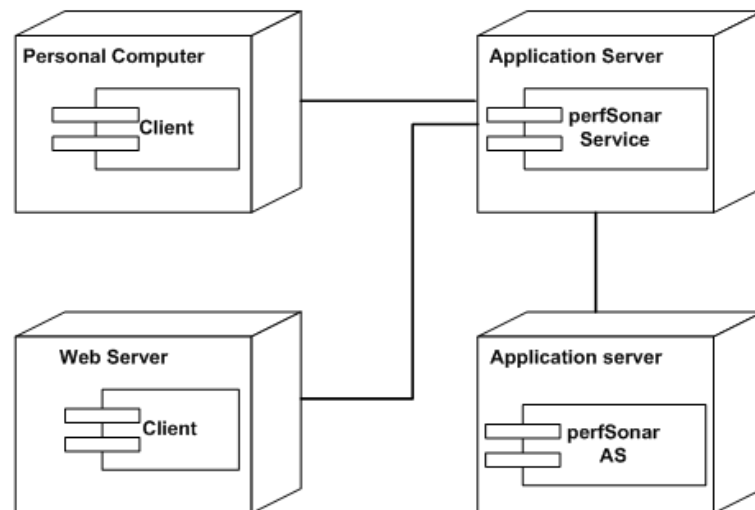


II.6. Contoh Notasi *Object Diagram*
(Sumber : Haviluddin , 2011 ; 3)

- *Deployment diagram*

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.



II.7. Contoh Notasi *Deployment Diagram*
(Sumber : *Haviluddin , 2011 ; 4*)

- *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.

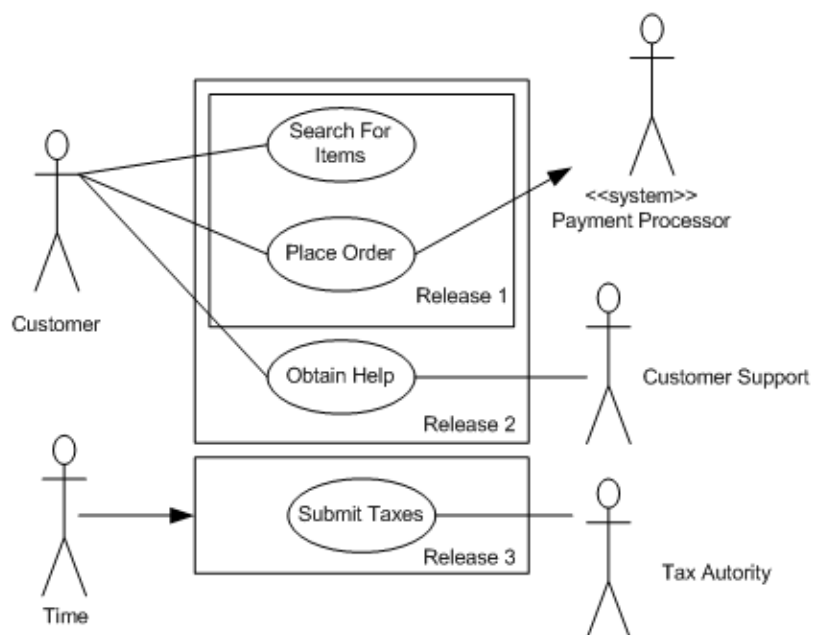
- *Package diagram*

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.

B. Behavior Diagram

- *Usecase 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.



II.8. Contoh Notasi Usecase Diagram

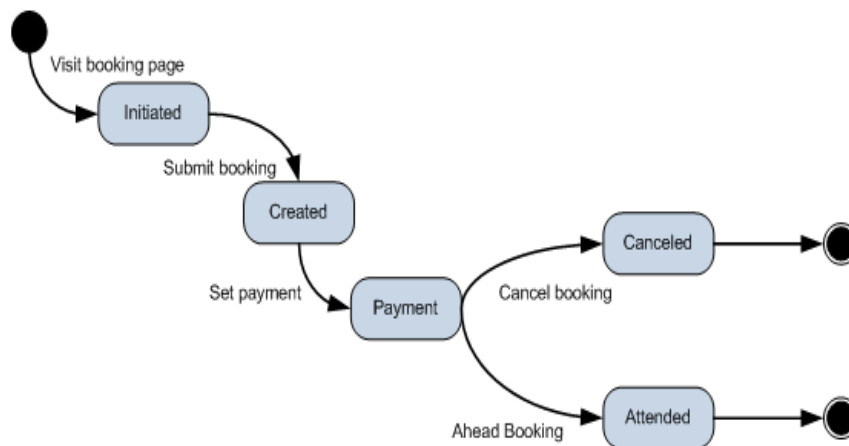
(Sumber : Havaluddin , 2011 ; 4)

- *Activity diagram*

Menggambarkan aktifitas-aktifitas, objek, state, transisi state dan event. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas

- *State Machine diagram*

Menggambarkan state, transisi state dan event.



II.9. Contoh Notasi *State Machine Diagram*
 (Sumber : Haviluddin , 2011 ; 4)

B. Interaction diagram

- *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.

- *Interaction Overview diagram*

Interaksi *overview* diagram berfokus pada gambaran aliran kendali interaksi dimana node adalah interaksi atau kejadian interaksi.

- *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*

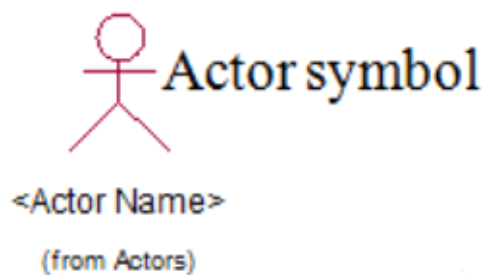
- *Timing diagram*

Timing diagram di UML didasarkan pada diagram waktu hardware awalnya dikembangkan oleh para insinyur listrik.

(Haviluddin , 2011 ; 3-5)

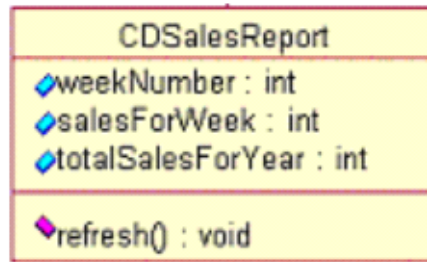
Untuk menggambarkan analisa dan desain diagram, UML memiliki seperangkat notasi yang akan digunakan ke dalam tiga kategori diatas yaitu struktur diagram, behaviour diagram dan interaction diagram. Berikut beberapa notasi dalam UML diantaranya :

- *Actor*, menentukan peran yang dimainkan oleh user atau sistem lain yang berinteraksi dengan subjek. *Actor* adalah segala sesuatu yang berinteraksi langsung dengan sistem aplikasi komputer, seperti orang, benda atau lainnya. Tugas actor adalah memberikan informasi kepada sistem dan dapat memerintahkan sistem untuk melakukan sesuatu tugas.



II.10. Notasi Actor (Sumber : Haviluddin , 2011 ; 6)

- *Class diagram* Notasi utama dan yang paling mendasar pada diagram UML adalah notasi untuk mempresentasikan suatu *class* beserta dengan atribut dan operasinya. *Class* adalah pembentuk utama dari sistem berorientasi objek.



II.11. Notasi Class (Sumber : Haviluddin , 2011 ; 6)

- *Use Case* dan *use case specification*, *Use case* adalah deskripsi fungsi dari sebuah sistem perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut skenario.
- Realization, Realization menunjukkan hubungan bahwa elemen yang ada di bagian tanpa panah akan merealisasikan apa yang dinyatakan oleh elemen yang ada di bagian dengan panah.
- Interaction, Interaction digunakan untuk menunjukkan baik aliran pesan atau informasi antar obyek maupun hubungan antar obyek.
- Dependency, Dependency merupakan relasi yang menunjukkan bahwa perubahan pada salah satu elemen memberi pengaruh pada elemen lain. Terdapat 2 stereotype dari dependency, yaitu include dan extend. Include menunjukkan bahwa suatu bagian dari elemen (yang ada digaris tanpa panah) memicu eksekusi bagian dari elemen lain (yang ada di garis dengan

panah). Extend menunjukkan bahwa suatu bagian dari elemen di garis tanpa panah bisa disisipkan ke dalam elemen yang ada di garis dengan panah. (Haviluddin , 2011 ; 6-7)