

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Perancangan suatu program aplikasi terdiri dari satu kesatuan sistem. Terdapat dua kelompok pendekatan di dalam mendefinisikan sistem, yaitu yang menekankan pada prosedur dan yang menekankan pada komponen. Pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan sistem sebagai berikut.

“Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu”. (B.Romney, Dkk; 2006: 473).

Prosedur di definisikan oleh Ricard F. Neuschel sebagai suatu urutan operasi klerikal (tulis-menulis), biasanya melibatkan beberapa orang di dalam satu atau lebih departemen, yang di terapkan untuk menjamin penanganan yang seragam dari transaksi-transaksi bisnis yang terjadi.

Pendekatan yang lebih menekankan pada elemen atau komponennya didefinisikan sistem sebagai kumpulan dari elemen-elemen yang berinteraksi untuk mencapai tujuan tertentu. Suatu sistem yang baik harus mempunyai tujuan dan sasaran yang tepat karena hal ini akan sangat menentukan dalam mendefinisikan masukan yang dibutuhkan sistem dan juga keluaran yang dihasilkan.

Dapat di simpulkan sistem adalah kegiatan-kegiatan yang saling berhubungan antara satu sama yang lainnya yang terdiri dari objek-objek, unsur-unsur atau komponen-komponen sehingga membentuk suatu kesatuan pemrosesan untuk mencapai tujuan tertentu.

II.1.1. Karakteristik Sistem

Model umum sebuah sistem adalah input, proses dan output. Hal ini merupakan konsep sebuah sistem yang sangat sederhana sebab sebuah sistem dapat mempunyai beberapa masukan dan keluaran. Seperti yang telah di sebutkan dari beberapa pengertian diatas, maka sistem terdiri dari elemen-elemen yang berkaitan. Dan elemen-elemen sistem yang terdiri dari :

1. Komponen-komponen (*components*) suatu sistem yang saling berinteraksi, yang artinya saling bekerjasama membentuk satu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa satu subsistem atau bagian-bagian dari sistem.
2. Batas sistem (*Boundry*) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas suatu sistem ini memungkinkan suatu sistem di pandang sebagai satu kesatuan.
3. Lingkungan luar sistem (*Enviroments*) adalah apapun diluar sistem batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem tersebut.

Sedangkan lingkungan luar yang merugikan harus di tahan dan di kendalikan, kalau tidak akan mengganggu kelangsungan hidup dari sistem .

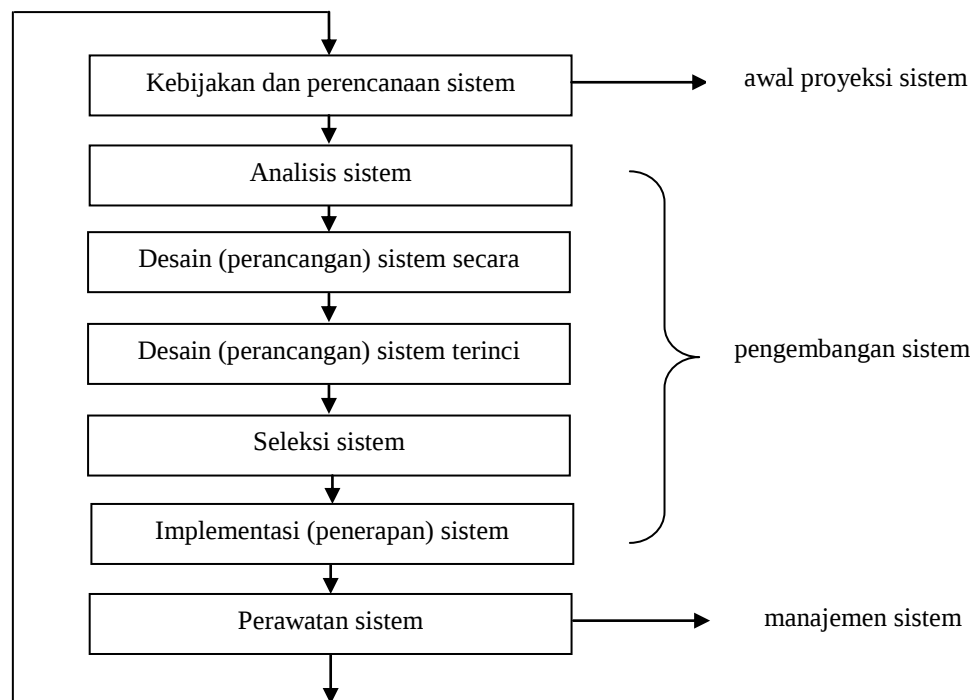
4. Penghubung (*interface*) merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem yang lainnya.
5. Masukan (*Input*) adalah energi yang dimasukkan kedalam sistem masukan dapat berupa masukan perawatan dan masukan sinyal. Maintenance input adalah energi yang di masukkan supaya sistem tersebut dapat beroperasi. Signal input adalah energi yang di proses untuk di dapatkan keluaran. Sebagai contoh di dalam sistem komputer.
6. Keluaran (*output*) adalah hasil dari energi yang diolah dan di klasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada sistem. Misalnya untuk sistem komputer, panas yang di hasilkan adalah keluaran yang tidak berguna dan merupakan hasil sisa pembuangan, sedangkan informasi adalah keluaran yang di butuhkan.

II.1.2. Siklus Hidup Pengembangan Sistem

Pengembangan sistem informasi yang berbasis komputer dapat merupakan tugas kompleks yang membutuhkan banyak sumber daya dan dapat memakan waktu berbulan-bulan bahkan bertahun-tahun untuk menyelesaikannya. Proses pengembangan sistem melewati beberapa tahapan. Tahapan utama siklus hidup pengembangan sistem dapat terdiri dari tahapan perencanaan sistem (*systems planning*), analisis sistem (*systems analysis*), desain sistem (*systems design*),

seleksi sistem (*systems selection*), implementasi sistem (*systems implementation*) dan perawatan sistem (*systems maintenance*).

Adapun Siklus hidup pengembangan sistem dapat dilihat pada gambar II.1 dibawah ini :



Gambar II.1. Siklus Hidup Pengembangan Sistem
Sumber : B.Romney, Dkk; 2006

II.2. Informasi

Informasi adalah data yang telah di klasifikasikan atau diolah atau diinterpretasi untuk di gunakan dalam proses pengambilan keputusan. Atau Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.

“Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan berarti bagi yang menerimanya” (B.Romney, Dkk; 2006).

Kualitas dari suatu sistem informasi (*quality of information*) di tentukan oleh tiga faktor yaitu sebagai berikut.

1. Relevan

Relevan berarti informasi mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya adalah berbeda. Misalnya informasi mengenai harga produksi untuk ahli teknik perusahaan merupakan informasi yang kurang relevan jika di tujukan kepada akuntan.

2. Tepat Waktu

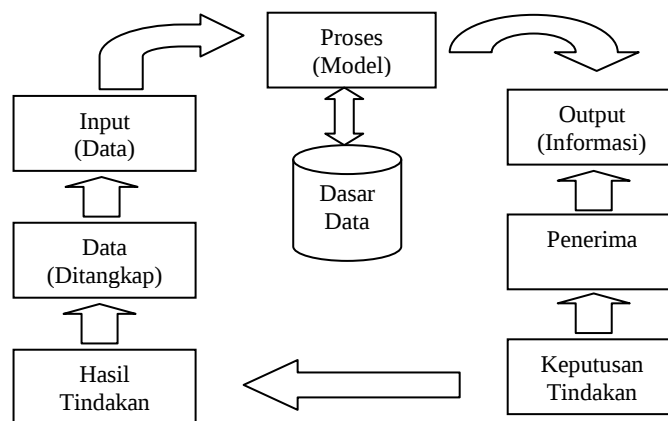
Tepat waktu berarti bahwa informasi datang pada saat di butuhkan sehingga bermanfaat untuk pengambilan keputusan. Berarti informasi yang datang pada penerima tidak boleh terlambat, informasi yang sudah usang tidak akan mempunyai nilai lagi sebab informasi merupakan landasan dalam mengambil keputusan.

3. Akurat

Akurat berarti bahwa informasi bebas dari kesalahan. Berarti informasi harus bebas dari kesalahan dan tidak menyesatkan harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke penerima informasi kemungkinan banyak terjadi gangguan yang dapat merubah informasi.

Informasi itu mempunyai kandungan ”makna”, maksudnya adalah karena berdasarkan maknalah si penerima dapat memahami informasi tersebut dan secara lebih jauh dapat menggunakannya untuk menarik suatu kesimpulan.

Pertama-tama data dimasukkan ke dalam model yang umumnya memiliki urutan proses tertentu dan pasti. Setelah diproses akan di hasilkan informasi tertentu yang bermanfaat bagi penerima (level management) sebagai dasar dalam membuat suatu keputusan atau melakukan tindakan tertentu. Dari keputusan atau tindakan tersebut akan menghasilkan atau diperoleh kejadian-kejadian tertentu yang akan di gunakan kembali sebagai data yang nantinya akan di masukkan ke dalam model (proses), begitu seterusnya. Dengan demikian akan membentuk suatu siklus informasi (*information cycle*) atau siklus pengolahan data (*data processing cycles*), adapun siklus informasi dapat dilihat pada gambar II.2 dibawah ini :



Gambar II.2. Siklus Informasi
Sumber : B.Romney, Dkk; 2006

II.3. Konsep Dasar Sistem Informasi

Sebuah sistem informasi merupakan kumpulan dari perangkat keras dan perangkat lunak komputer serta perangkat manusia yang akan mengolah data menggunakan perangkat keras dan perangkat lunak tersebut.

Sistem Informasi adalah sekumpulan hardware, software, brainware, prosedur atau aturan yang diorganisasikan secara integral untuk mengolah data menjadi informasi yang bermanfaat guna memecahkan masalah dan pengambilan keputusan.

Sistem Informasi adalah cara teratur untuk mengumpulkan, memproses, mengelola, dan melaporkan informasi agar organisasi dapat mencapai tujuan dan sasarannya (B.Romney, Dkk; 2006: 473).

Informasi merupakan hal yang sangat penting dalam pengambilan keputusan, permasalahannya adalah dimana informasi tersebut didapat. Informasi dapat diperoleh dari sistem informasi. Robert A Leitch dan K. Roscoe Davis mendefinisikan sistem informasi sebagai berikut:

Komponen-komponen sistem informasi adalah sebagai berikut:

1. Perangkat Keras (*hardware*)

Perangkat keras (*hardware*) terdiri dari monitor, CPU, keyboard, mouse dan hard disk.

2. Perangkat Lunak (*software*)

Perangkat lunak (*software*) berupa program-program aplikasi yang akan digunakan, yaitu merupakan kumpulan dari perintah atau fungsi yang ditulis

dengan aturan tertentu untuk memerintahkan komputer melaksanakan tugas tertentu.

3. Data

Data merupakan komponen dasar dari informasi yang akan di proses lebih lanjut untuk menghasilkan informasi.

4. Prosedur

Prosedur merupakan dokumentasi prosedur atau proses sistem, tata cara atau penuntun operasional (aplikasi) dan teknis

5. Manusia

Manusia adalah pengguna dari sistem informasi.

II.4. Sistem Informasi Akuntansi (SIA)

II.4.1. Pengenalan SIA

Mempelajari SIA adalah hal yang penting dalam akuntansi, Dalam *Statement Of Financial Accounting Concepts no.2, Financial Accounting Standards Board* mendefenisikan akuntansi sebagai sistem informasi. Di dalam standar akuntansi keuangan tersebut juga disebutkan bahwa tujuan utama akuntansi adalah untuk menyediakan informasi yang berguna bagi para pengambil keputusan. Oleh sebab itu, bukanlah hal yang mengherankan apabila *accounting education change commision* merekomendasikan bahwa kurikulum akuntansi harus menekankan bahwa akuntansi adalah suatu proses identifikasi, pengembangan, pengukuran dan komunikasi informasi. Komisi tersebut menyarankan agar kurikulum akuntansi harus dirancang untuk memberi para mahasiswa sebuah pemahaman yang kuat atas tiga komponen dasar berikut:

1. Pemakaian informasi didalam pengambilan keputusan
2. Sifat, desain, pemakaian dan implementasi SIA
3. Pelaporan informasi keuangan

II.4.2. Defenisi SIA

Sistem Informasi Akuntansi adalah sekumpulan hardware, software, brainware, prosedur atau aturan yang di organisasikan secara integral untuk mengolah data menjadi informasi yang bermanfaat guna memecahkan masalah dan pengambilan keputusan(B.Romney,Dkk:2006,473).

II.4.3. Komponen-Komponen SIA

Sistem Informasi Akuntansi (SIA) terdiri dari lima (5) komponen yaitu:

1. Orang-orang yang mengoperasikan sistem tersebut dan melaksanakan berbagai fungsi.
2. Prosedur-prosedur baik manual maupun yang terotomatisasi, yang dilibatkan dalam mengumpulkan, memproses dan menyimpan data tentang aktifitas-aktivitas organisasi.
3. Data tentang proses-proses bisnis organisasi.
4. *Software* yang dipakai untuk memproses dan organisasi.
5. *Infrastruktur teknologi informasi*, termasuk komputer, peralatan pendukung(*Peripheral device*) dan peralatan untuk komunikasi jaringan.

Kelima komponen ini secara bersama-sama memungkinkan suatu SIA memenuhi tiga fungsi pentingnya dalam organisasi, yaitu:

1. Mengumpulkan dan menyimpan data tentang aktivitas-aktivitas yang dilaksanakan oleh organisasi, sumber daya yang dipengaruhi oleh aktivitas-aktivitas tersebut, dan para pelaku yang terlibat dalam berbagai aktivitas tersebut, agar pihak manajemen, dan pihak-pihak luar yang berkepentingan dapat meninjau ulang (*review*) hal-hal yang telah terjadi.
2. Mengubah data menjadi informasi yang berguna bagi pihak manajemen untuk membuat keputusan dalam aktivitas perencanaan, pelaksanaan, dan pengawasan.
3. Menyediakan pengendalian yang memadai untuk menjaga aset-aset organisasi, termasuk data organisasi, untuk memastikan bahwa data tersebut tersedia dibutuhkan, akurat dan andal.

II.5. Pengolahan Dana Pensiun

Menurut Floyd A.Beams, dkk Dana adalah suatu entitas fiskal dan akuntansi dengan sekumpulan akun yang dapat menyeimbangkan diri yang mencatat kas dan sumber daya keuangan lainnya, beserta seluruh kewajiban terkait dan saldo atau ekuitas, dan perubahannya, yang dipisahkan untuk tujuan menyelenggarakan aktivitas tertentu sesuai dengan peraturan, larangan, atau pembatasan khusus.

Pemerintah mencatat dan melaporkan sistem pensiun pegawai umum (*public employee retirement system / PERS*) melalui dana pensiun. Beberapa program pensiun pemerintah didanai pemerintah sepenuhnya, lainnya didanai sebagian, dan selebihnya melakukan seluruh pembayaran pensiun dari

pendapatan operasional. Dana kepemilikan dan perwalian yang sejenis dan entitas lain yang menggunakan akuntansi dana kepemilikan harus mengakui biaya pensiun atas basis akrual dalam laporan keuangan pemerintah maupun dana. Kewajiban atau aktiva bersih harus diakui sebagai kewajiban atau aktiva dana.

II.6. *Unified Modelling Language (UML)*

II.6.1. Pengenalan UML

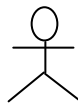
Unified Modelling Language (UML) adalah ‘bahasa’ pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek’. Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami. (B.Romney, Dkk; 2006)

UML dikembangkan oleh 3 pendekar ‘berorientasi objek’, yaitu Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. UML menjadi bahasa yang bisa digunakan untuk berkomunikasi dalam perspektif obyek antara user dengan developer, antara developer dengan developer, antara developer analisis dengan developer disain, dan antara developer disain dengan developer pemrograman. UML memungkinkan developer melakukan pemodelan secara visual, yaitu penekanan pada penggambaran, bukan didominasi oleh narasi. Pemodelan visual membantu untuk menangkap struktur dan kelakuan dari obyek, mempermudah penggambaran interaksi antara elemen dalam sistem, dan mempertahankan konsistensi antara disain dan implementasi dalam pemrograman.

UML menyediakan standar pada notasi dan artifak (diagram) yang bisa digunakan untuk memodelkan suatu sistem. Berikut ini adalah notasi yang ada pada UML, yaitu sebagai berikut.

1. *Actor*

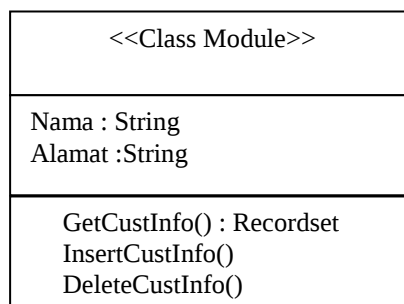
Actor adalah segala sesuatu yang berinteraksi dengan sistem aplikasi komputer.



Gambar II.3. Notasi Actor
Sumber : (B.Romney, Dkk; 2006)

2. *Class*

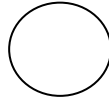
Class merupakan pembentuk utama dari sistem berorientasi objek karena class menunjukkan kumpulan objek yang memiliki atribut dan operasi yang sama. Notasi *class* berbentuk persegi panjang berisi 3 bagian : persegi paling atas untuk nama *class*, persegi panjang tengah untuk atribut, dan persegi panjang paling bawah untuk operasi.



Gambar II.4. Notasi Class
Sumber : (B.Romney, Dkk; 2006)

3. *Interface*

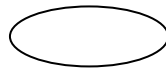
Interface merupakan kumpulan operasi tanpa implementasi dari suatu *class*.



Gambar II.5. Notasi Interface
Sumber : (B.Romney, Dkk; 2006).

4. *Use Case*

Use case menjelaskan urutan kegiatan yang dilakukan *actor* dan sistem untuk mencapai suatu tujuan tertentu. Walaupun menjelaskan kegiatan namun *use*

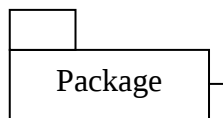


Gambar II.6. Notasi Use Case
Sumber : (B.Romney, Dkk; 2006).

case hanya menjelaskan apa yang dilakukan oleh *actor* dan sistem, bukan bagaimana *actor* dan sistem melakukan kegiatan tersebut.

5. *Package*

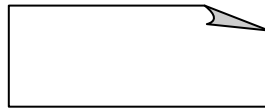
Package adalah container atau wadah konseptual yang digunakan untuk mengelompokkan elemen-elemen dari sistem yang sedang dibangun, sehingga bisa dibuat model yang lebih sederhana. Tujuannya adalah untuk mempermudah penglihatan (*visibility*) dari model yang sedang dibangun.



Gambar II.7. Notasi Package
Sumber : (B.Romney, Dkk; 2006).

6. Note

Note di gunakan untuk memberikan keterangan dan komentar tambahan dari suatu elemen sehingga bisa langsung terlampir dalam model.



Gambar II.8. Notasi Note
Sumber : (B.Romney, Dkk; 2006).

Relasi-relasi antarpengklasifikasi (*classifier*) yang di kenali UML (*Unified Modelling Language*) adalah asosiasi (*association*), generalisasi (*generalization*), aliran (*flow*), dan berbagai jenis kebergantungan (*dependency*), termasuk didalamnya realisasi (*realization*) dan penggunaan (*usage*).

Tabel II.1. Relasi-relasi dalam UML

Relasi	Fungsi	Notasi
Asosiasi (<i>association</i>)	Mendeskripsikan hubungan antar- <i>instance</i> suatu kelas.	—————
Kebergantungan (<i>dependency</i>)	Relasi antardua elemen model.	----->
Aliran (<i>flow</i>)	Relasi antardua versi suatu objek.	----->
Generalisasi (<i>generalization</i>)	Relasi antara pengklasifikasi yang memiliki deskripsi yang bersifat lebih umum dengan berbagai pengklasifikasi yang lebih spesifik, digunakan dalam struktur pewarisan.	—>
Realisasi (<i>realization</i>)	Relasi antara spesifikasi dan implementasinya.	----->
Penggunaan (<i>usage</i>)	Situasi dimana salah satu elemen membutuhkan elemen yang lainnya agar dapat berfungsi dengan baik.	----->

Sumber : (B.Romney, Dkk; 2006).

II.6.2 Diagram UML

Untuk membuat suatu model, UML memiliki diagram grafis yaitu sebagai berikut:

1. *Use Case Diagram*

Use case diagram menggambarkan fungsionalitas yang di harapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang di perbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem.

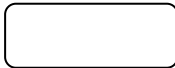
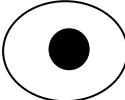
2. *Class Diagram*

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* memiliki tiga area pokok , yaitu: Nama (dan stereotype), Atribut, serta Metoda.

3. *Statechart Diagram*

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari *stimuli* yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*).

Tabel II.2. Jenis-Jenis State

Relasi	Fungsi	Notasi
<i>State sederhana</i>	<i>State</i> tanpa struktur apapun di dalamnya.	
<i>State komposit konkuren</i>	<i>State</i> yang dibagi menjadi 2 atau lebih <i>substate</i> konkuren.	
<i>Initial state</i>	<i>State</i> mengindikasikan awal rangkaian <i>state</i> dalam diagram <i>state</i> .	
<i>Final state</i>	<i>State</i> mengindikasikan akhir rangkaian <i>state</i> dalam diagram <i>state</i> .	

Sumber : (B.Romney, Dkk; 2006).

4. Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

5. Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

6. *Collaboration Diagram*

Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*.

7. *Component Diagram*

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya. Komponen piranti lunak adalah modul berisi *code*. Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil.

8. *Deployment Diagram*

Deployment diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisik.

II.7. Pengertian Visual Basic.Net 2008

Visual Basic berawal dari bahasa BASIC yang di kembangkan mulai dari tahun 1963. BASIC adalah singkatan dari *Beginner's All Purpose Symbolic Instruction Code*. Sesuai namanya, bahasa BASIC dibuat untuk tujuan memudahkan pengguna agar dapat dengan mudah mempelajari, membuat dan mengembangkan program komputer.

Visual basic 6.0 sangat populer dan masih banyak di pakai hingga saat ini. Sayangnya dukungan terhadap Visual Basic 6 telah di hentikan oleh microsoft mulai bulan Maret 2008. Namun, program yang dibuat dengan Visual Basic 6 masih dapat di jalankan pada sistem operasi terbaru, seperti windows server 2008 maupun Windows Vista. Visual Basic .Net diluncurkan Februari 2002, merupakan penerus dari visual basic 6.0 dan menggunakan platform .Net yang berbeda dengan Visual Basic sebelumnya.

Visual Basic, merupakan bahasa dan aturan pemrograman yang harus ditaati dalam menuliskan perintah-perintah agar program dapat dikompilasi.

Visual.Basic 2008 adalah Aplikasi IDE (Integrated Development Environment) yang digunakan untuk mengembangkan software. Di dalam aplikasi IDE inilah tersedia berbagai fitur yang memudahkan pemrograman, seperti kompilasi debugging, pengaturan proyek, mengedit antarmuka secara visual, dan lain-lain (Rachmad Hakim S: 2009,2).

Visual Studio 2008 hadir dengan beberapa versi, yaitu:

- **Team System**, didesain untuk pemrograman dilingkungan korporasi dengan jumlah programmer yang besar.
- **Professional Edition**, di desain untuk pemrograman yang melibatkan sedikit programmer.
- **Standard Edition**, di desain untuk pemrograman standar yang bukan enterprise.

- **Express Edition**, di desain untuk pemula yang baru belajar hobi dengan fasilitas yang sangatterbatas.

Pemrograman dengan Visual Basic dibuat dengan beberapa tahap berikut:

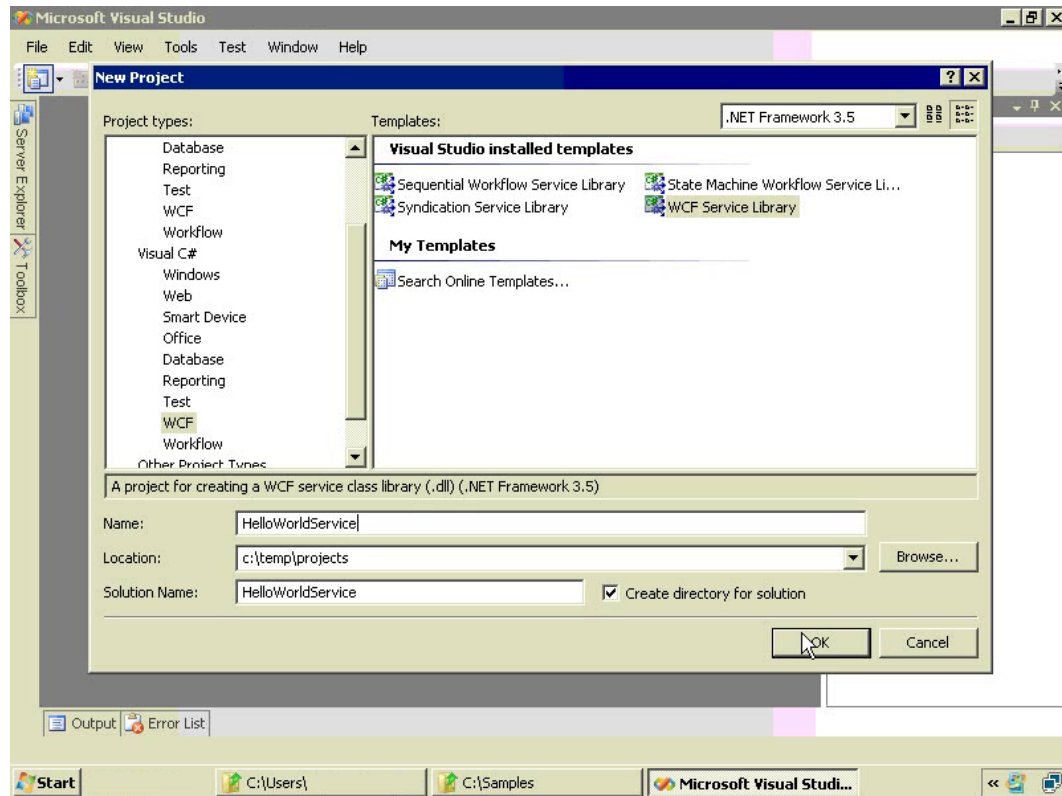
- Menuliskan kode program dengan bantuan aplikasi IDE. Artinya, Anda menuliskan program menggunakan aplikasi Microsoft Visual Basic 2008 Express Edition.
- Mengompilasi kode program tersebut menjadi program yang dapatdijalankan/dieksekusi. Hasil kompilasi adalah intruksi CIL (*Common Intermediate Language*) atau MSIL (*Microsoft Internediate Language*) yang hanya dimengerti oleh kompiler JIT(*Just In Time*) dan tidak dapat dieksekusi langsung padda komputer.
- Penyebaran (distribusi) program di komputer dengan .Net Framework.

.NET Framework, merupakan *library* dan *virtual machine* yang terus berkembang mengikuti teknologi terbaru. Versi .NET Framework dimulai dari versi 1.0, 1.1, 2.0, 3.0 dan 3.5. Versi .NET yang biasanya dirilis dengan perbaikanserta dukungan terhadap teknologi baru sehinggasemakin memudahkan pengembangan software.

.NET Framework, merupakan software kerangka kerja yang menghubungkan antara aplikasi .NET dengan sistem operasi, yang secara garis besar terdiri atas:

- *Library*, berisi kode-kode siap pakai dan banyak dibutuhkan oleh programmer.

- *Virtual machine*, berupa aplikasi yang digunakan untuk menjalankan program hasil kompilasi.



Gambar II.9. Microsoft Visual Studio

Sumber : Rachmad Hakim S (2009 : 10)

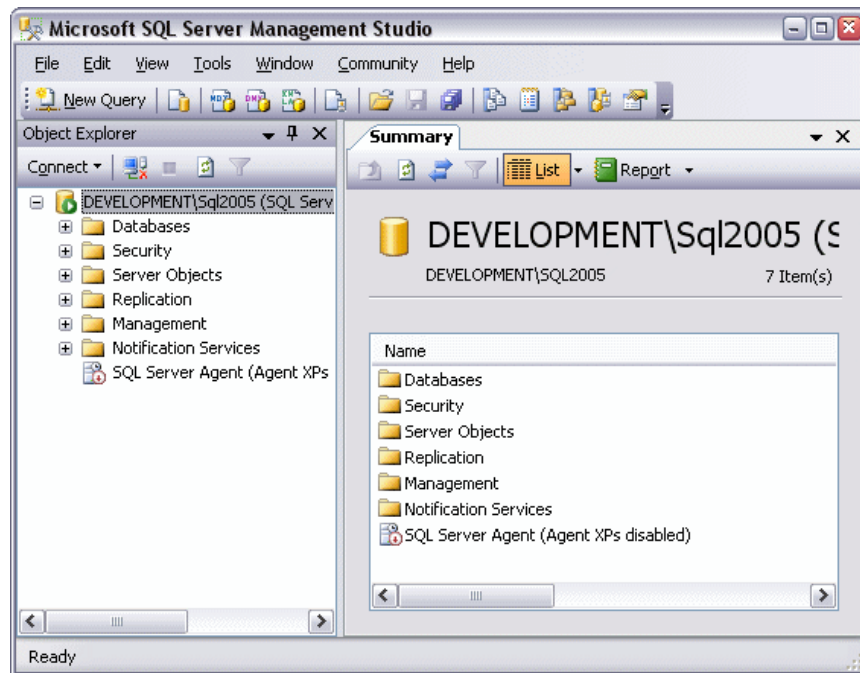
II.8. Pengertian SQL Server 2005

SQL Server 2005 adalah Server basis data yang secara fungsional adalah proses menyediakan layanan basis data. Klien berinteraksi dengan layanan basis data melalui antar muka komunikasi tertentu yang bertujuan untuk pengendalian dan keamanan. Klient tidak mempunyai akses langsung ke data tetapi selalu berkomunikasi dengan server basis data (William R. Stanek : 2009, 4)

Dengan menggunakan SQL Server Setup, kita dapat membuat instance-instance baru dari SQL Server, menambahkan komponen-komponen, membangun kembali registry SQL Server, meng-uninstall SQLServer 2005 akan digunakan dilingkungan Anda. Ketika Anda sudah memutuskan mengenai role atau peran yang akan dimiliki oleh SQL Server, Anda bisa membuat rencana peluncuran dan kemudian meluncurkan SQL Server.

SQL server dirancang sebagai sebuah platform Business Intelligence yang lengkap yang bisa digunakan untuk:

- Ekstraksi, Transformasi, dan Loading (*Extraction, Transformation, and Loading- ETL*)
- *Data warehouse* (gudang data) relasional
- Database-database multidimensi dan data mining (penambangan data)
- Analysis services di SQL Server 2005
- Pelaporan terkelola (*managed Reporting*)



Gambar II.10. Microsoft SQL Server Management Studio

Sumber : William R. Stanek (2009 : 16)