

BAB 2

LANDASAN TOERI

2.1 2 Teori-teori yang menjadi dasar penulisan adalah sebagai berikut:

2.1.1 Sistem

Menurut Hall (2011,p.5) “*A system is a group of two or more interrelated components or subsystem that seve a common purpose.*”Yang artinya sistem adalah serangkaian dari dua atau lebih komponen atau subsistem yang saling terkait yang melayani tujuan yang sama.

Menurut O’Brien diterjemahkan oleh Dewi Fitriyani dan Deny Arnos Kwary (2006,p.29), sistem adalah sekelompok komponen yang saling berhubungan bekerja bersama untuk mencapai tujuan bersama dengan menerima input serta menghasilkan output dalam prosestransformasi yang teratur.

Dengan demikian sistem merupakan kumpulan dari beberapa bagian yang memiliki keterkaitan dan saling bekerja sama serta membentuk suatu kesatuan untuk mencapai suatu tujuan dari sistem tersebut. Maksud dari sistem tersebut adalah untuk mencapai suatu tujaun dan sasaran dalam ruang lingkup yang sempit.

2.1.2 Informasi

Menurut O’Brien diterjemahkan oleh Dewi Fitriyani dan Deny Arnos Kwary (2006,p.38), informasi adalah data yang telah diubah menjadi konteks yang berarti dan berguna bagi para pemakai akhir tertentu.

Menurut Hall (2011,p.780) *“Information is fact that cause the user to take an action that he or she otherwise could not, would not, have taken”*. yang artinya informasi adalah fakta yang menyebabkan pengguna untuk mengambil tindakan bahwa ia dinyatakan tidak bisa, atau tidak mau, telah diambil.

Dengan demikian informasi merupakan sekumpulan data yang diproses dan dirubah menjadi sautu bentuk yang mempunyai arti dan berguna bagi penerima.

2.1.3 Karakteristik Informasi

Karakteristik informasi yang berguna menurut (Hall,2011,p.13) adalah sebagai berikut :

a. Relavan

isi sebuah laporan atau dokumen harus melayani suatu tujuan yaitu memenuhi kebutuhan pengguna informasi. Dengan demikian laporan atau dokumen yang bersangkutan dapat mendukung keputusan manajer.

b. Tepat Waktu

Umur informasi merupakan faktor yang kritikal dalam menentukan kegunaannya. Informasi harus tidak lebih tua dari periode waktu tindakan yang didukungnya.

c. Akurat

Informasi harus bebas dari kesalahan yang sifatnya material. Materialistis merupakan suatu konsep yang sulit dikualifikasi dan tidak memiliki nilai yang absolut.

d. Lengkap

Tidak boleh ada bagian informasi yang sesnsial bagi pengambilan keputusan atau pelaksanaan tugas yang hilang.

2.1.4 Sistem Informasi

Menurut O'Brien (2006:5), Sistem informasi merupakan kombinasi teratur apapun dari orang-orang, hardware, software, jaringan komunikasi dan sumber daya data yang mengumpulkan, mengubah dan menyebarkan informasi dalam sebuah perusahaan. Manusia bergantung pada sistem informasi untuk berkomunikasi antara satu sama lain dengan menggunakan berbagai jenis perangkat keras, piranti lunak, perintah, jaringan dan sumber daya data sejak permulaan peradaban

Menurut Hall (2011: 780), "*Information system is set of formal procedures by which data are collected, processed into information, and distributed to users*". Yang artinya Sistem informasi adalah sekumpulan prosedur-prosedur yang sah dimana data dikumpulkan, diproses menjadi informasi, dan disalurkan kepada pengguna“

Secara umum sistem informasi dapat didefinisikan sebagai suatu sistem di dalam suatu organisasi yang merupakan kombinasi dari orang-orang fasilitas, teknologi, media prosedur-prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputusan.

2.1.5 Komponen Sistem Informasi

Komponen Sistem Informasi Menurut Reynolds dan Stair (2010: 10), terdapat beberapa komponen dalam sistem informasi:

- a. Input, aktivitas mengumpulkan dan memperoleh data mentah.
- b. Processing, mengkonversi atau mentransformasikan data menjadi output yang bermanfaat.

- c. Output, produksi dari informasi yang bermanfaat, biasanya dalam bentuk dokumen dan laporan.
- d. Feedback, output yang digunakan untuk membuat perubahan pada input atau aktivitas pemrosesan.

2.2 Sistem Informasi Akuntansi

2.2.1 Pengertian Sistem Informasi Akuntansi

Menurut Hall (2011: 773), "*Accounting information system is specialized subset of information system that processes financial transaction*". Yang diterjemahkan; " Sistem informasi akuntansi adalah kumpulan dari sebagian informasi sistem yang khusus yang dapat mengolah transaksi keuangan."

Menurut Gelinas dan Dull (2012: 667), "*Accounting information system is a specialized subsystem of the IS that collects, processes, and reports information related to the financial aspects of business events*". Yang diterjemahkan; " Sistem informasi akuntansi adalah sebuah subsistem khusus dari sistem informasi yang mampu mengoleksi, memproses, dan melaporkan informasi yang berhubungan dengan aspek keuangan dalam suatu peristiwa bisnis"

Berdasarkan pendapat ahli di atas, maka dapat disimpulkan bahwa sistem informasi akuntansi adalah Subsistem dari sistem informasi dan teknologi yang dirancang secara khusus untuk menjalankan fungsi dalam mengumpulkan, mengolah, dan melaporkan informasi keuangan yang terdapat didalam organisasi.

2.2.2 Komponen Sistem Informasi Akuntansi

Menurut Hall (2007:27) karakteristik kualitas informasi akuntansi terdiri dari 5, sebagai berikut :

a. Relevan

Informasi yang relevan merupakan informasi yang perlu diketahui untuk memberikan pemahaman yang baru. Laporan yang hanya bersifat sementara, dan selanjutnya tidak relevan harus dihentikan pembuatannya.

b. Tepat Waktu

Umur informasi merupakan faktor yang kritikal dalam menentukan kegunaannya. Informasi harus tidak lebih tua dari periode waktu tindakan yang didukungnya.

c. Akurat

Informasi harus bebas dari kesalahan yang sifatnya material. Kesalahan-kesalahan material ada ketika jumlah informasi yang tidak akurat menyebabkan pemakaiannya melakukan keputusan yang buruk atau gagal melakukan keputusan yang diperlukan.

d. Lengkap

Tidak ada bagian informasi yang esensial bagi pengambilan keputusan atau pelaksanaan tugas yang hilang. Informasi yang tidak lengkap bisa menimbulkan kesulitan, karena informasi yang tidak disertakan itu akan menjadi unsur ketidakpastian yang besar.

e. Rangkuman

Informasi harus digregasi agar sesuai dengan kebutuhan pemakai. Informasi yang ringkas dan menghitiskan data relevan yang menunjukkan bidang-bidang penyimpanan terhadap tingkat normal, standar atau yang direncanakan merupakan bentuk informasi yang banyak diperlukan oleh para pemakai informasi.

Dapat disimpulkan karakteristik dari sistem informasi akuntansi adalah melaksanakan tugas yang diperlukan, berpegang pada prosedur yang relatif standar, menangani data rinci, berfokus historis, dan menyediakan informasi pemecahan minimal.

2.2.3 Tujuan dan Kegunaan Sistem Informasi Akuntansi

Menurut Jones dan Rama (2009:7), kegunaan sistem informasi akuntansi adalah:

1. Menghasilkan laporan-laporan eksternal

Perusahaan menggunakan Sistem Informasi Akuntansi untuk menghasilkan laporan khusus untuk memuaskan informasi yang dibutuhkan oleh para investor, kreditor, pemungut pajak, dan yang lainnya.

2. Mendukung aktivitas rutin

Manajer membutuhkan Sistem Informasi Akuntansi untuk menangani aktivitas operasi rutin dalam siklus operasi perusahaan. Contoh dalam hal ini termasuk dalam hal mengambil pesanan pelanggan, menyampaikan barang dan jasa, membebaskan piutang pada konsumen, dan mengumpulkan kas. Sistem terkomputerisasi ahli dalam menangani transaksi yang berulang, dan banyak paket perangkat lunak mendukung fungsi rutin tersebut.

3. Mendukung pengambilan keputusan

Informasi juga dibutuhkan untuk mendukung pengambilan keputusan non-rutin pada seluruh tingkat organisasi, seperti mengetahui produk mana yang terjual dengan baik dan mana yang paling banyak dibeli oleh konsumen. Informasi ini penting bagi perencanaan produk baru, memutuskan produk mana yang harus selalu tersedia, dan memasarkan produk pada konsumen.

4. Perencanaan dan pengendalian

Sistem informasi dibutuhkan pula bagi aktivitas perencanaan dan pengendalian informasi mengenai anggaran dan biaya standar simpan oleh sistem informasi, dan laporan-laporan dirancang untuk membandingkan anggaran dengan jumlah yang sesungguhnya.

5. Mengimplementasikan pengendalian internal

Pengendalian internal termasuk kebijakan, prosedur, dan sistem informasi yang digunakan untuk melindungi harta

perusahaan dari kehilangan atau kekacauan dan untuk memelihara akurasi data keuangan. Membangun pengendalian ke dalam sebuah sistem informasi yang terkomputerisasi membantu untuk mencapai tujuan tersebut.

Menurut Hall (2011, p.14), sebuah sistem informasi akuntansi yang dirancang dengan baik dapat memberikan kegunaan, sebagai berikut:

1. Untuk mendukung fungsi pengelolaan manajemen (*to support the stewardship function of management*).
2. Mendukung operasional perusahaan sehari-hari (*to support the firm's day-to-day operations*).
3. Mendukung pengambilan keputusan manajemen (*to support management decision making*).

2.3 Teori Khusus

2.3.1 Pengertian Penjualan

Menurut Arief Sugiono, Yanuar Nanok Soenarno dan Synthia Madya Kusumawati, (2010:133) Penjualan bersih merupakan selisih antara penjualan baik yang dilakukan secara tunai maupun kredit dengan retur penjualan dan potongan penjualan.

Dan menurut Leny Sulistiyowati, (2010:270) Pendapatan yang berasal dari penjualan produk perusahaan, disajikan setelah dikurangi potongan penjualan dan retur penjualan.

Berdasarkan pengertian dari kedua ahli di atas dapat disimpulkan bahwa penjualan adalah pendapatan penjualan produk perusahaan yang merupakan selisih antara penjualan baik secara tunai maupun kredit kemudian disajikan setelah dikurangi potongan penjualan dan retur penjualan.

Penjualan dibagi menjadi dua bagian, yaitu :

1. Penjualan kredit, yaitu penjualan yang pembayaran dilakukan di kemudian hari dalam jangka waktu yang telah ditetapkan setelah barang diterima oleh customer. Penjualan kredit inilah

yang menimbulkan piutang dagang, sehingga penjualan tidak dapat dipisahkan dari timbulnya piutang usaha.

2. Penjualan tunai, yaitu penjualan yang pembayarannya dilakukan secara langsung saat terjadinya transaksi.

2.3.2 Pendapatan

Pendapatan adalah arus masuk bruto dari manfaat ekonomi yang timbul dari aktivitas normal perusahaan selama suatu periode bila arus masuk tersebut mengakibatkan kenaikan ekuitas, yang tidak berasal dari kontribusi penanaman modal. (IAI, 2010: 23.2)

Pendapatan barang harus diakui bila seluruh kondisi berikut dipenuhi :

- a. Perusahaan telah memindahkan resiko secara signifikan dan telah memindahkan manfaat kepemilikan barang kepada pembeli.
- b. Perusahaan tidak lagi mengelola atau melakukan pengendalian efektif atas barang yang dijual
- c. Jumlah pendapatan tersebut dapat diukur dengan andal
- d. Besar kemungkinan manfaat ekonomi yang dihubungkan dengan transaksi akan mengalir kepada perusahaan tersebut
- e. Biaya yang terjadi atau yang akan terjadi sehubungan dengan transaksi penjualan dapat diukur dengan andal. (IAI, 2010 : 23.3)

2.3.3 Pengertian Kas dan Setara Kas

Menurut Kieso, Weygandt & Kimmel, (2010) Kas dan setara kas dapat diartikan bersifat jangka pendek, investasi, dengan tingkat likuiditas yang tinggi, siap diubah menjadi sejumlah kas yang diketahui jumlahnya dan sangat dekat dengan maturitas mereka yang memrepresentasikan resiko tidak signifikan dari perubahan tingkat suku bunga.

Kas adalah alat pertukaran yang dimiliki oleh perusahaan dan siap untuk digunakan dalam transaksi perusahaan setiap dibutuhkan dan diinginkan. (Rudianto, 2012).

Kas dan setara kas dapat diartikan bersifat jangka pendek, investasi, dengan tingkat likuiditas yang tinggi, siap diubah menjadi sejumlah kas yang diketahui jumlahnya dan sangat dekat dengan maturitas mereka yang memrepresentasikan resiko tidak signifikan dari perubahan tingkat suku bunga. (Kieso, Weygandt & Kimmel, 2010)

2.3.4 Penerimaan Kas

Menurut Rudianto, (2012) Penerimaan Kas adalah alat pertukaran yang dimiliki oleh perusahaan dan siap untuk digunakan dalam transaksi perusahaan setiap dibutuhkan dan diinginkan. (Rudianto, 2012)

Terdapat lima prosedur didalam system penerimaan kas yang meliputi :

1. Remittance Entry

Pada tahap ini kasir akan mengumpulkan semua cek dan mencocokkan dengan remittance advice yang diterima, kemudian menjumlahkan semua cek yang diterima. Kemudian remittance list yang berisi remittance advice secara keseluruhan yang telah dibuat.

2. Depositing Receipt

Salah satu salinan dari remittance list dikirimkan kekasir yang akan membandingkan dan merekonsiliasi. Kemudian, kasir akan membuat deposit slip cash receipt transaction listing (Jurnal). Setelah itu barulah semua akan disetorkan ke bank.

3. Update Account Receivable

Remittance advice digunakan untuk mengirimkan ke rekening nasabah dalam buku besar pembantu piutang. Secara berkala, perubahan saldo rekening dirangkum dan diteruskan ke fungsi buku besar.

4. Update General Ledger

Setelah menerima voucher jurnal dan ringkasan akun, fungsi buki besar umum dituliskan ke rekening kas dan rekening control piutang, dan file jurnal voucher.

5. Reconcile Cash Receipt and Deposits

Secara berkala (mingguan atau bulanan) seorang pegawai dari kantor pengawas (atau karyawan tidak terlibat dengan prosedur penerimaan kas) menyatukan penerimaan kas dengan membandingkan dokumen-dokumen sebagai berikut:

- i. Salinan prelist yang terkait
- ii. Slip pemyetoran yang diterima oleh bank
- iii. Jurnal voucher yang terkait. (Hall, 2008:173)

Jurnal disebut sebagai buku entry asli. Untuk setiap transaksi jurnal menunjukkan efek debit dan kredit pada akun tertentu. Dengan kata lain, jurnal merupakan catatan yang timbul sebagai efek dari transaksi yang terjadi dimana terdapat debit dan kredit untuk mencatatnya dengan akun spesifik. Jurnal-jurnal yang berhubungan dengan transaksi penjualan, piutang usaha dan penerimaan kas serta retur penjualan adalah sebagai berikut :

1. Jurnal penjualan tunai

Dr. Kas	xxxx	
Cr. Penjualan		xxxx

2. Jurnal Penjualan kredit

Dr. Piutang Usaha	xxxx	
Cr. Penjualan		xxxx

3. Jurnal Penerimaan kas

Dr. Kas	xxxx	
\Cr. Piutang usahas		xxxx

4. Jurnal retur penjualan

Dr. Retur dan potongan penjualan	xxxx	
Cr. Piutang usaha		xxxx

2.3.5 Pengertian Piutang Usaha

Menurut Arif dan Wibowo (2008:132), piutang adalah Piutang (*Receivable*) pengertian klaim terhadap sejumlah uang yang diharapkan diperoleh pada masa yang akan datang.

Menurut PSAK no.9, piutang usaha meliputi piutang yang timbul karena penjualan produk atau penyerahan jasa dalam rangka kegiatan usaha normal perusahaan. Piutang usaha dan lain-lain yang diharapkan tertagih dalam satu atau siklus usaha normal diklasifikasikan sebagai aktiva lancar.

2.3.6 Akuntansi Piutang Usaha

Transaksi yang mempengaruhi piutang usaha merupakan bagian dari siklus pendapatan. Siklus pendapatan tersebut adalah transaksi penjualan kredit barang dan jasa kepada pelanggan, transaksi retur penjualan, transaksi penerimaan kas dari debitur, dan transaksi penghapusan piutang. Transaksi-transaksi tersebut dicatat ke dalam jurnal sebagai berikut:

- a. Transaksi penjualan kredit barang dan jasa kepada pelanggan. Jurnal untuk mencatat transaksi ini adalah:

Piutang usaha	xxx
Penjualan/ pendapatan jasa	xxx

- b. Transaksi retur penjualan. Jurnal untuk mencatat transaksi ini adalah:

Retur penjualan dan pengurangan harga	xxx
Piutang usaha	xxx

- c. Transaksi penerimaan kas dari debitur. Jurnal untuk mencatat transaksi ini adalah:

Kas	xxx
Piutang usaha	xxx

- d. Transaksi penghapusan piutang, Jurnal untuk mencatat transaksi ini adalah:

Cadangan kerugian piutang	xxx
Piutang usaha	xxx

2.3.7 Penilaian Piutang Usaha

Secara teori, semua piutang dinilai dalam jumlah yang mewakili nilai sekarang dari perkiraan penerimaan kas di masa medatang. Oleh karena piutang usaha berjangka pendek, biasanya ditagih dalam 30 hingga 90 hari, bunganya akan relatif lebih kecil dari jumlah piutangnya. Sebagai ganti dari penilaian piutang usaha pada nilai sekarang yang disetorkan, piutang dilaporkan sebagai nilai realisasi bersih (*net realizable value*), yaitu nilai kas yang diharapkan.

Hal ini berarti bahwa piutang usaha harus dicatat sebagai jumlah bersih dari estimasi piutang tak tertagih dan potongan dagang. Tujuannya adalah untuk melaporkan piutang sejumlah klaim dari pelanggan yang benar-benar diperkirakan dapat diterima secara tunai.

2.3.8 Pengertian Piutang Tak Tertagih

Menurut Reeve, Warren dan Dunhac (2009:398), terdapat dua metode akuntansi untuk mencatat piutang yang diperkirakan tidak akan tertagih. Metode penyisihan (*allowance method*) membuat akun beban piutang tak tertagih di muka sebelum piutang tersebut dihapus. Prosedur lain, yang dinamakan dengan metode penghapusan langsung (*direct write-off method*), mengakui beban hanya pada saat piutang dianggap benar-benar tidak dapat ditagih lagi.

Dua metode akuntansi untuk piutang tak tertagih yaitu:

1. Metode Penghapusan Langsung (*direct write-off method*) Pencatatan beban piutang tak tertagih ketika akun piutang dinyatakan sudah tidak bisa tertagih.
2. Metode Penyisihan (*allowance method*) Pencatatan beban piutang tak tertagih diestimasi berdasarkan jumlah piutang tak tertagih di akhir periode akuntansi.

2.4 Teori Analisis dan Perancangan

2.4.1 Analisa dan Perancangan Sistem Informasi

Menurut McLeod (2007 ,p88) analisis sistem adalah penelitian terhadap suatu sistem yang telah ada dengan tujuan untuk merancang sistem baru atau diperbaharui.

Menurut Mardi (2011 ,p124) menjelaskan analisis sistem adalah proses kerja untuk menguji sistem informasi yang sudah ada dengan lingkungannya sehingga diperoleh petunjuk berbagai kemungkinan perbaikan yang dapat dilakukan dalam meningkatkan kemampuan sistem.

Jadi dapat disimpulkan bahwa analisis sistem adalah penelitian atas sistem yang telah ada agar dapat menetapkan tujuan dan spesifikasi untuk merancang sistem baru.

O'Brien dan Marakas (2009:639) menjelaskan bahwa perancangan sistem adalah sebuah kegiatan merancang dan menentukan cara mengolah sistem informasi dari hasil analisa sistem sehingga dapat memenuhi kebutuhan dari pengguna termasuk diantaranya perancangan user interface, data dan aktivitas proses.

Menurut Bentley dan Whitten (2009:160) melalui buku yang berjudul “system analysis and design for the global enterprise” juga

menjelaskan bahwa perancangan sistem adalah teknik pemecahan masalah dengan melengkapi komponen-komponen kecil menjadi kesatuan komponen sistem kembali ke sistem yang lengkap. Teknik ini diharapkan dapat menghasilkan sistem yang lebih baik.

Berdasarkan pengertian di atas, analisa dan perancang sistem adalah suatu proses pengumpulan, penginterpretasikan kenyataan yang ada sehingga dapat diproses dan akan dihasilkan data yang diperlukan untuk sistem yang baru.

Menurut Shelly, Rosenblatt (2012: 142) analisis sistem terdiri dari 4 aktivitas utama :

1) Requirements Modeling

Aktivitas yang dilakukan disini meliputi pencarian fakta-fakta yang menggambarkan sistem berjalan dan mengidentifikasi kebutuhan-kebutuhan untuk pembuatan sistem baru seperti *input*, *output*, proses, kinerja, dan keamanan sistem.

2) Data and Process Modeling

Aktivitas yang dilakukan disini adalah melanjutkan pembuatan model proses dengan mempelajari bagaimana data dan proses sistem digambarkan secara grafis dengan menggunakan teknik analisis terstruktur yang mengidentifikasi bagaimana aliran data dalam sebuah proses, aturan bisnis yang mengubah data, dan *output* hasil aliran data.

3) Object Modeling

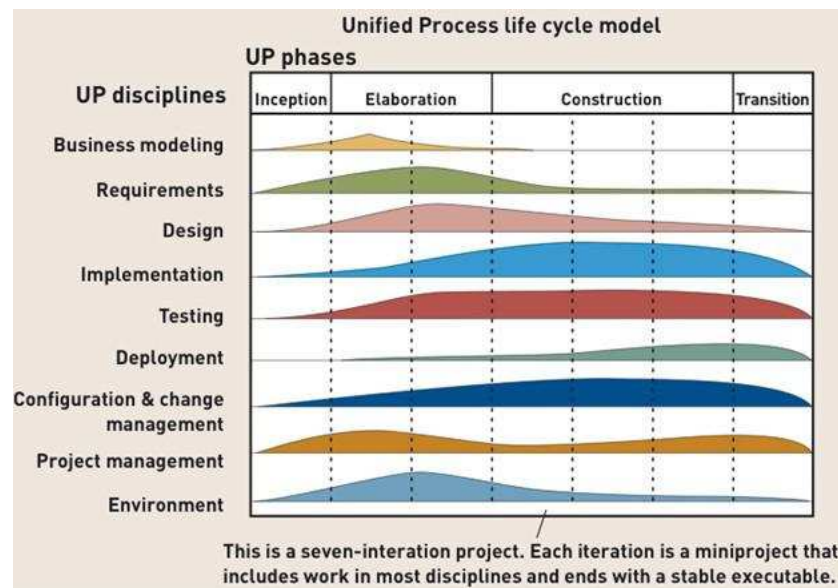
Pada aktivitas ini dilakukan pembuatan model proses yang mengkombinasikan data dan proses pengolahan data yang disebut dengan objek. Objek-objek ini menggambarkan manusia, benda, transaksi, dan peristiwa yang mempengaruhi sistem.

4) Considerations of Development Strategies

Pada aktivitas ini dilakukan pertimbangan dari berbagai macam pilihan pengembangan sistem dan melakukan persiapan untuk transisi ke fase perancangan sistem.

2.4.2 Unifed Process

Unified Process (UP) atau kadang disebut sebagai Unified Software Development Process (USDP) adalah: *kerangka proses pengembangan yang bersifat use-case-driven, berpusat pada arsitektur perangkat lunak, interatif dan tumbuh-kembang (Alhir, 2005)*. Kerangka pengembangan ini termasuk baru dalam metodologi pengembangan perangkat lunak. UP dapat diaplikasikan pada berbagai skala proyek, mulai dari skala kecil sampai dengan skala besar.



Gambar 2.1 UP Life Cycle with Phases

2.4.3 Fase Life Cycle UP

Life Cycle UP memiliki empat tahapan sebagai berikut :

1. Fase Inception

Tahapan ini merupakan tahapan paling awal dimana aktivitas penilaian terhadap sebuah proyek perangkat lunak dilakukan, tujuannya adalah untuk mendapatkan kesepakatan dari stakeholder sehubungan dengan tujuan dan proyek.

Tujuan utama dari fase ini adalah :

1. Menentukan batasan dari proyek software dan batasan kondisi, termasuk didalamnya visi operational dan penerimaan kriteria.
2. Memperkirakan biaya keseluruhan dan jasanya untuk seluruh proyek.
3. Memperkirakan risiko potensial yang akan terjadi.
4. Mempersiapkan lingkungan pendukung proyek.

Sedangkan kegiatan-kegiatan penting yang terjadi dalam fase ini meliputi kegiatan untuk merumuskan ruang lingkup proyek, sintesis arsitektur, merencanakan dan menyiapkan kasus bisnis. Dimana yang termasuk di dalam perencanaan dan penyiapan kasus bisnis adalah pengevaluasian alternatif untuk pengelolaan risiko, rencana proyek dan biaya. Kegiatan yang terakhir yaitu menyiapkan lingkungan untuk proyek, termasuk di dalamnya menilai proyek dan organisasi, memilih alat-alat, dan memutuskan bagian mana dari proses yang harus diperbaiki.

2. Fase Elaboration

Tujuan dari tahap ini adalah untuk mendapatkan gambaran umum kebutuhan, persyaratan dan fungsi-fungsi utama perangkat lunak. Hal ini penting untuk mengetahui secara lebih baik risiko-risiko proyek, baik meliputi risiko arsitektur perangkat lunak, perencanaan, maupun implementasi. Pada tahap ini telah dimulai rancangan bangun perangkat lunak secara iterative melalui

aktivitas-aktivitas seperti business modeling, requirement, analysis dan design meskipun baru tahap awal.

Tujuan utama dari fase elaborasi ini adalah :

1. Untuk memastikan bahwa arsitektur, requirement dan planning cukup stabil, mengurangi risiko biaya dan jadwal untuk menyelesaikan pembangunan.
2. Untuk mengatasi risiko yang signifikan dalam proyek arsitektur.
3. Untuk menunjukkan bahwa arsitektur dasar mendukung persyaratan sistem dengan biaya yang wajar dan dalam waktu yang wajar.
4. Membentuk sporting environment

Kegiatan penting yang terjadi dalam fase ini adalah kegiatan untuk mengidentifikasi, memvalidasi arsitektur secara cepat dan praktis. Menyempurnakan fase, berdasarkan pada informasi yang di dapat dalam fase ini. Serta, memperbaiki arsitektur dan memilih komponen.

3. Fase Construction

Tujuan dari tahapan ini adalah membangun perangkat lunak sampai dengan saat perangkat lunak tersebut siap digunakan. Titik berat tahapan ini adalah padapenentuan tingkat prioritas kebutuhan atau persyaratan, melengkapi spesifikasinya, analisis lebih dalam, design solusi yang memenuhi kebutuhan dan persyaratan, pengkodean dan pengujian perangkat lunak. Jika dimungkinkan versi awal dari perangkat lunak diuji cobakan untuk mendapatkan masukan dari pengguna.

Tujuan utama dalam fase ini adalah :

1. Meminimalkan biaya pengembangan daengan mengoptimalkan sumber daya danmenghindari hal-hal yang tidak perlu dan pengerjaan ulang.
2. Mencapai kualitas memadai secara praktis.

3. Menyelesaikan analisis, design, pengembangan dan pengujian dari semua fungsi yang diperlukan. Ini berarti tersisa menggambarkan kasus penggunaan dan persyaratan, menyelesaikan implementasi dan pengujian perangkat lunak.
4. Untuk memutuskan apakah perangkat lunak, situs dan para pengguna semua siap untuk aplikasi yang akan digunakan.

Sedangkan kegiatan utama yang akan digunakan pada fase ini adalah kegiatan untuk mengelola sumber daya, kontrol dan optimal proses. Pengembangan komponen lengkap dan pengujian terhadap kriteria evaluasi yang telah ditetapkan.

4. Fase Transition

Tahap ini di fokuskan pada bagaimana menyampaikan perangkat lunak yang sudah jadi kepada pengguna. Perangkat lunak akan secara resmi diuji oleh penguji yang kompeten maupun oleh pengguna. Beberapa aktivitas seperti pemindahan pusat data dan pelatihan pengguna dan staff pendukung harus dilakukan pada tahap seperti berikut ini.

Sembilan core process workflow dalam unified process life cycle, yaitu :

1. Business modeling

Di dalamnya dibuat dokumen business process yang dipakai disebut dengan business use cases. Dokumen ini menjamin stakeholder memahami kebutuhan bisnis proses yang diperlukan.

2. Requirements

Tujuan dari requirements adalah mendeskripsikan apa yang harus dikerjakan oleh sistem serta membolehkan developer dan customer untuk menyetujui deskripsi itu.

3. Design

Design dilakukan untuk menspesifikasikan secara penuh bagaimana fungsionalitas di implementasikan. artifact yang dihasilkan dalam workflow ini adalah design class,

design subsystem dan usecase realization. Pada alur kerja inti design juga dilakukan mapping dari class diagram kedalam relational tabel.

4. Implementation

Implementation workflow bertujuan untuk mengimplementasikan kelas dan object dalam hubungannya dengan komponen, mengetest komponen yang dihasilkan sebagai unit dan untuk mengintegrasikan hasil yang dibuat oleh masing-masing implementasi ke dalam table system. Implementasi juga dilakukan untuk mentransformasikan design model ke dalam table code. pada workflow implementation ini dilakukan implementasi komponen, subsitem dan interface.

5. Testing

Testing dalam workflow bertujuan untuk memeriksa interaksi antar objek, penggabungan komponen dari software dengan tepat, dan memeriksa apakah semua kebutuhan sudah dipenuhi dengan tepat. Selain itu, testing juga bertujuan untuk mengidentifikasi dan myakinkan bahwa kerusakan yang ada telah diatasi sebelum men-deploy software. Unifed process life cycle menawarkan pendekatan iterative yang memungkinkan kita mengetest keseluruhan project dengan menemukan kerusakan sejak dini sehingga mengurangi cost untuk memperbaikinya. Test menghasilkan tiga macam ukuran kualitas yaitu, functionality, application, dan system performance.

6. Deployment

Deployment dalam workflow dilakukan untuk menghasilkan produk realse dengan sukses dan aktifitas mengantar software kepada end user seperti membuat external releases dari software, packing software, distributing software, instaling serta membantu user memahami sistem. Aktivitas ini dilakukan pada fase transaction, dalam unifed process life cycle, deployment

workflow berisi paling sedikit detailnya daripada workflow yang lain.

7. Project Management

Project management menyediakan framework untuk mengatur software-intensive project, panduan untuk planning, staffing, executing, monitoring projects dan framework untuk mengatur risiko yang ada. Dikatakan sukses apabila produk tersebut dapat memenuhi kebutuhan user dan kebanyakan customer.

8. Configuration

Menyediakan panduan untuk mengatur penyusunan software system, mengatasi perubahan request management dan dapat menjadi salah satu cara untuk melaporkan suatu kerusakan.

9. Environment

Bertujuan untuk menyediakan software development organization beserta software development environment, yang dibutuhkan untuk mendukung development tim

Dalam pengembangan perangkat lunak dengan menggunakan UP, maka tidak lepas dari penggunaan notas-notasi yang biasa disebut sebagai UML(Unife Modeling Language).

2.4.4 Analisis dan Perancangan Sistem Berorientasi Objek

2.4.4.1 Pengertian *Object*

2.4.4.1.1 Pengertian *Object Oriented Analysis*

Menurut Satzinger (2010: 60), "Object-oriented analysis (OOA) defines all of the types of objects that do the work in the system and shows what user interaction, called use case, are required to complete tasks." yang diterjemahkan, "Objectoriented analysis mendeskripsikan semua bentuk dari obyek yang melakukan pekerjaan di dalam sistem dan menunjukkan interaksi

pengguna, disebut use case, yang diperlukan untuk menyelesaikan tugas.

2.4.4.1.2 Pengertian *Object Oriented Design*

Menurut Satzinger (2010: 60), "Object-oriented design (OOD) defines all of the additional types of objects necessary to communicate with people and devices in the system, shows how the objects interact to complete tasks, and refines the definition of each type of object so it can be implemented with a specific language or environment ." yang diterjemahkan, "Object-oriented analysis menjelaskan semua dari bentuk tambahan obyek-obyek yang diperlukan untuk berkomunikasi dengan orang dan perangkat di dalam sistem, menunjukkan pula bagaimana obyek-obyek berinteraksi untuk menyelesaikan suatu tugas, dan memperjelas definisi dari bentuk masing-masing obyek sehingga dapat diterapkan bersama suatu bahasa atau lingkungan yang khusus."

2.4.4.1.3 Pengertian Analisis Sistem

Menurut Bodnar dan Hopwood (2010:412), analisis sistem adalah proses memahami sistem dan masalah yang ada, menggambarkan kebutuhan informasi, dan menetapkan prioritas untuk kinerja sistem yang lebih lanjut.

Menurut Satzinger (2012: 5), "System Analysis is those activities that enable a person to understand and specify what the new system should accomplish." Yang diterjemahkan; " Analisis sistem adalah aktivitas-aktivitas yang memungkinkan seseorang untuk mengerti dan menentukan sistem baru seperti apa yang ingin dicapai".

Berdasarkan pendapat di atas, maka dapat disimpulkan bahwa analisis sistem adalah sebuah teknik untuk memahami dan memecahkan masalah yang terjadi

secara terperinci dengan membagi sistem menjadi beberapa komponen yang kemudian dipelajari untuk mencapai tujuan.

2.4.4.1.4 Pengertian Perancangan Sistem

Menurut Satzinger (2012: 5), “System design is those activities that enable a person to define and describe in detail the system that solves the need”. Yang diterjemahkan; “Perancangan sistem adalah aktivitas-aktivitas yang memungkinkan seseorang untuk mendefinisikan dan menjelaskan secara rinci suatu sistem yang memenuhi kebutuhan.”

Menurut Whitten dan Bentley dan Bentley (2009: 160), “System design is a complementary problem solving technique (to system analysis) that reassembles a system’s component pieces back into a complete system – hopefully, an improved system. This may involve adding, deleting, and changing pieces relative to the original system”. Yang diterjemahkan “Perancangan sistem adalah suatu teknik pemecahan masalah yang saling melengkapi (bagi analisis sistem) dengan mengumpulkan kembali bagian-bagian komponen suatu sistem kembali menjadi sebuah sistem jadi – dan diharapkan, menjadi sebuah sistem yang lebih baik. Aktivitas ini mungkin akan menyangkut penambahan, penghapusan, dan perubahan bagian-bagian yang relatif terhadap sistem awal.”

Dari pengertian diatas dapat disimpulkan bahwa perancangan sistem adalah erancangan sistem adalah sekumpulan aktivitas yang tersusun dalam teknik pemecahan masalah yang saling mendukung satu komponen dengan yang lainnya, sehingga membentuk satu kesatuan yang sistematis dan dapat memenuhi kebutuhan pengguna.

2.4.5 Pengertian UML (*Unified Modelling Language*)

Menurut Satzinger (2010:61), dalam bukunya *Object Oriented Analysis and Design* menyatakan bahwa *Unified Modelling Language (UML)* adalah satu set standar konstruksi model dan notasi yang dikembangkan secara khusus untuk pengembangan berorientasi objek. Dengan menggunakan UML, analisis dan pengguna akhir yang dapat menggambarkan dan memahami berbagai diagram spesifik yang digunakan dalam proyek pengembangan sistem.

Menurut Whitten dan Bentley (2009: 371), "*UML is a set of modeling conventions that is used to specify or describe a software system in terms of objects.*" Yang diterjemahkan; "UML adalah sekumpulan kaidah pemodelan yang digunakan untuk menjelaskan atau mendeskripsikan sebuah sistem software didalam konteks obyek."

Dari pengertian diatas dapat disimpulkan bahwa UML merupakan sekumpulan catatan grafis yang disusun berdasarkan kaidah pemodelan yang digunakan untuk membantu dalam menjelaskan atau mendeskripsikan sebuah sistem software didalam konteks obyek.

2.4.5.1 *Activity Diagram*

Menurut Satzinger (2010:141) *Flowcharts* dan diagram aktivitas yang khusus dirancang untuk mewakili aliran kontrol di antara langkah-langkah pengolahan. Banyak analis menggunakan jenis *workflow diagram* dan menyebutnya *activity diagram*. Suatu *activity diagram* merupakan gambaran berbagai pengguna (atau sistem) kegiatan, orang yang melakukan aktivitas masing-masing, dan aliran sekuensial dari kegiatan tersebut. Symbol yang digunakan yaitu:

1. *Starting Activity* (pseudo) merupakan simbol untuk menandakan dimulainya aktivitas.

2. *Transition Arrow* merupakan garis penunjuk arah urutan aktivitas yang menggambarkan transisi dari suatu aktivitas.

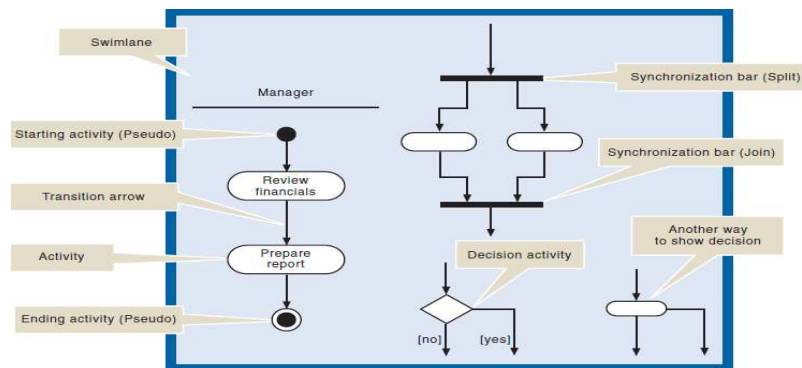
3. *Activity* merupakan simbol yang menggambarkan aktivitas.

4. *Ending Activity* (pseudo) merupakan simbol untuk menandakan berakhirnya aktivitas.

5. *Swimlane* merupakan area persegi dalam activity diagram yang menunjukkan aktivitas diselesaikan single agent.

6. *Synchronization bar* merupakan symbol yang digunakan untuk mengontrol pemisahan atau penyatuan dari jalur berurutan.

7. *Diamond* merupakan simbol poin keputusan dalam alur proses mengikuti satu jalur atau jalur lainnya.



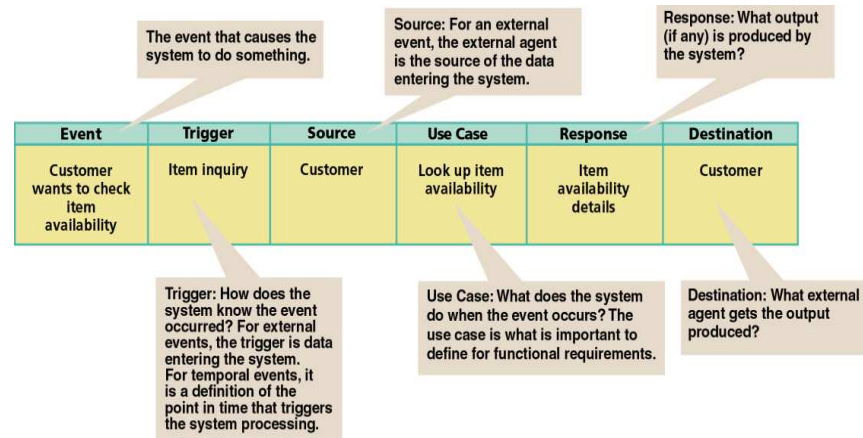
Gambar 2.2 :*Symbol Activity Diagram*.

Sumber: Satzinger, Jackson, dan Burd (2010:142)

2.4.5.2 *Event Table*

Menurut Satzinger, Jackson, dan Burd (2010:168), *event table* adalah sebuah pedoman *use case* daftar peristiwa dalam baris dan potongan kunci informasi setiap peristiwa dalam kolom.

Di dalam *event table* terdapat tahapan yang harus dilakukan didalam kolom yakni seperti kolom berikut ini :



Gambar 2.3 : *Event Table*

Sumber : Satzinger, Jackson, dan Burd (2010:169)

a. *Event*

Event adalah kejadian pada waktu dan tempat tertentu, dapat digambarkan, dan harus diingat oleh sistem.

b. *Trigger*

Trigger adalah tanda yang memberitahukan sistem bahwa telah terjadi peristiwa. Untuk peristiwa eksternal, *trigger* merupakan datangnya data yang harus diproses oleh sistem. Contohnya, ketika pelanggan melakukan pesanan, maka rincian pesanan baru sebagai *input*. Untuk peristiwa sementara, *trigger* merupakan titik waktu. Contohnya, pada akhir setiap hari kerja, sistem telah mengetahui waktu untuk menghasilkan laporan ringkasan transaksi.

c. *Source*

Source adalah agen eksternal yang memberikan data ke sistem.

d. *Response*

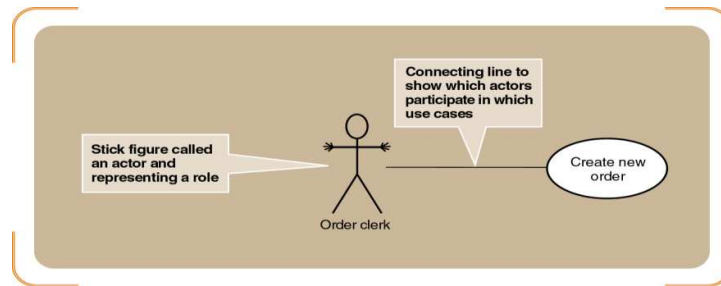
Response adalah *output* dari sistem. Ketika sistem menghasilkan laporan ringkasan transaksi, laporan tersebut merupakan *outputs*. Satu *use case* dapat menghasilkan beberapa *responses*. Contoh, ketika sistem membuat pesanan baru, maka konfirmasi pesanan diberikan kepada pelanggan, rincian pesanan diberikan kepada bagian pengiriman, dan catatan transaksi diberikan kepada bank.

e. *Destination*

Destination adalah tempat dimana beberapa *response* telah dikirim. Kadang-kadang *use case* tidak menghasilkan *response* sama sekali. Contoh, jika pelanggan ingin melakukan *update* informasi akun, informasi tersebut tersimpan dalam *database*, tapi tidak dibutuhkan *output* untuk dihasilkan. Mencatat informasi dalam *database* merupakan bagian dari *use case*.

2.4.5.3 Use Case Diagram

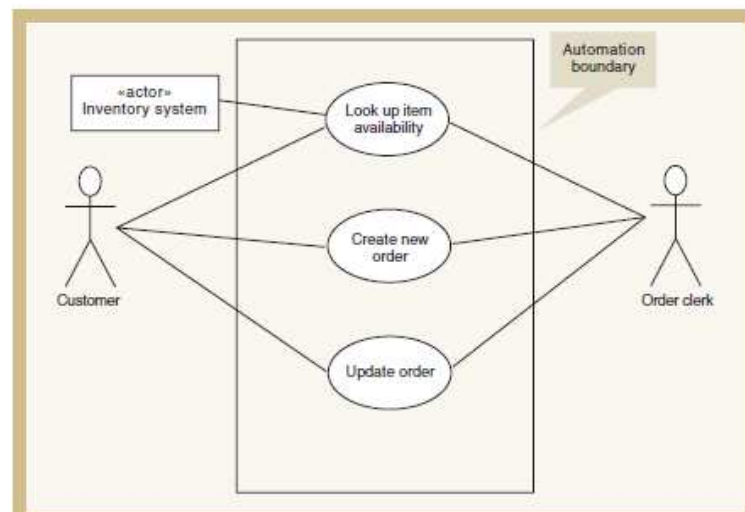
Menurut Satzinger (2010:242) *use case* adalah sebuah tongkat sederhana yang digunakan untuk mewakili aktor (tangan ditunjukkan langsung mengakses ke sistem langsung). Kasus penggunaan sendiri dilambangkan oleh oval dengan nama *use case* didalamnya. Garis yang menghubungkan aktor dengan *use case* menunjukkan bahwa aktor memanfaatkan penggunaan sistemnya. Pelaku juga dapat menggunakan sistem lain yang langsung menunjukkan antar muka dengan sistem yang sedang dikembangkan.



Gambar 2.4 : *Simple use case with an actor.*

Sumber : Satzinger, Jackson, dan Burd (2010:243)

Otomatisasi batasan dan organisasi yang ditunjukkan di *dalam use case diagram* memperluas penggunaan diagram sama halnya dengan aktor-aktor lain dan menggunakan kasus.



Gambar 2.5: *A Use Case Diagram of the Order-Entry*

Subsystem for RMO, Showing a System Boundary.

Sumber : Satzinger, Jackson, dan Burd (2010:244)

2.4.5.4 Use Case Description

Use case description menjelaskan tentang suatu penggunaan kasus sistem yang mencakup seluruh urutan langkah untuk menyelesaikan suatu proses bisnis. Dan sering kali beberapa variasi dari langkah-langkah bisnis ada dalam kasus penggunaan tunggal. Satzinger (2010:171)

Use Case Name:	Create new order	
Scenario:	Create new telephone order	
Triggering Event:	Customer telephones RMO to purchase items from the catalog.	
Brief Description:	When customer calls to order, the order clerk and system verify customer information, create a new order, add items to the order, verify payment, create the order transaction, and finalize the order.	
Actors:	Telephone sales clerk	
Related Use Cases:	Includes: <i>Check item availability</i>	
Stakeholders:	Sales department: to provide primary definition Shipping department: to verify that information content is adequate for fulfillment Marketing department: to collect customer statistics for studies of buying patterns	
Preconditions:	Customer must exist. Catalog, Products, and Inventory items must exist for requested items.	
Postconditions:	Order and order line items must be created. Order transaction must be created for the order payment. Inventory items must have the quantity on hand updated. The order must be related (associated) to a customer.	
Flow of Events:	Actor	System
	1. Sales clerk answers telephone and connects to a customer. 2. Clerk verifies customer information. 3. Clerk initiates the creation of a new order. 4. Customer requests an item be added to the order. 5. Clerk verifies the item (<i>Check item availability</i> use case). 6. Clerk adds item to the order. 7. Repeat steps 4, 5, and 6 until all items are added to the order. 8. Customer indicates end of order; clerk enters end of order. 9. Customer submits payment; clerk enters amount.	3.1 Create a new order. 5.1 Display item information. 6.1 Add an order item. 8.1 Complete order. 8.2 Compute totals. 9.1 Verify payment. 9.2 Create order transaction. 9.3 Finalize order.
Exception Conditions:	2.1 If customer does not exist, then the clerk pauses this use case and invokes <i>Maintain customer information</i> use case. 2.2 If customer has a credit hold, then clerk transfers the customer to a customer service representative. 4.1 If an item is not in stock, then customer can a. choose not to purchase item, or b. request item be added as a back-ordered item. 9.1 If customer payment is rejected due to bad-credit verification, then a. order is canceled, or b. order is put on hold until check is received.	

Gambar 2.6: *Use Case Description*

Sumber: Satzinger, Jackson, dan Burd (2010:174)

2.4.5.5 Domain Class Diagram

Menurut Satzinger, Jackson, dan Burd (2010:187), *class diagram* memberikan gambaran struktur dan deskripsi kelas, *package*, dan objek serta hubungan satu sama lain seperti pewarisan dan asosiasi. Format yang digunakan untuk menentukan masing-masing atribut adalah:

1. *Attribute visibility: visibility* menunjukkan apakah objek lain dapat mengakses attribute secara langsung atau tidak. Tanda + (*plus*) mengindikasikan *attribute* dapat terlihat atau bersifat public, dan tanda – (*minus*) menandakan bahwa *attribute* tidak dapat terlihat atau bersifat *private*.

2. *Attribute name*

3. *Type-expression*: dapat berupa *character*, *string*, *integer*, *number*, *currency*, atau *date*.

4. *Initial value*

5. *Property*: ditempatkan dalam kurung kurawal. Contoh: {*key*}.

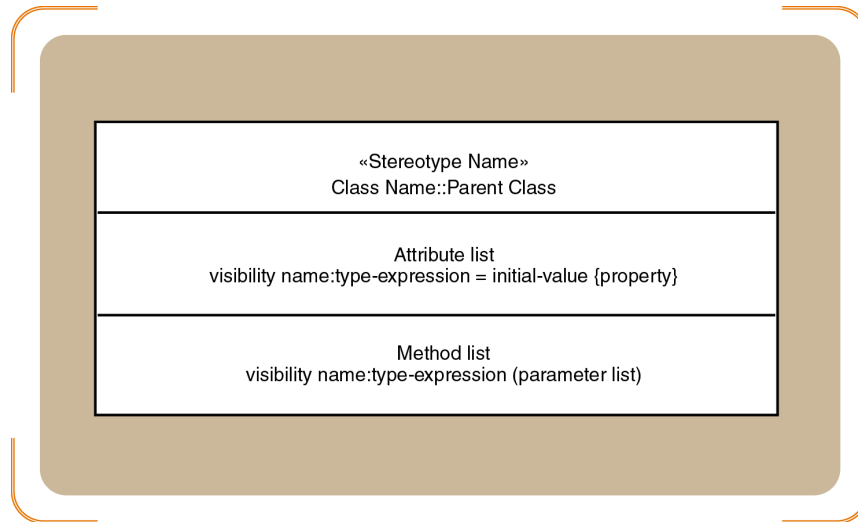
Format yang digunakan dalam method list:

1. *Method visibility*

2. *Method name*

3. *Type-expression*: tipe dari *return parameter* dari *method*.

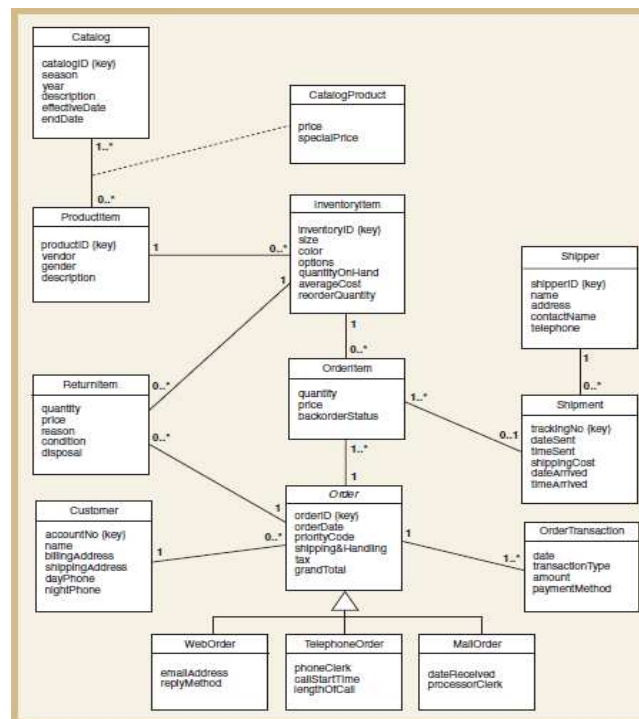
4. *Method parameter list*: argumen yang masuk.



Gambar 2.7: Notasi *Class Diagram*.

Sumber: Satzinger, Jackson, dan Burd (2010:411)

Salah satu desain dari class diagram yaitu *Domain Model Class Diagram* dan *Updated Class Diagram*. Berikut contoh *domain class diagram* dan *Updated Class Diagram*.

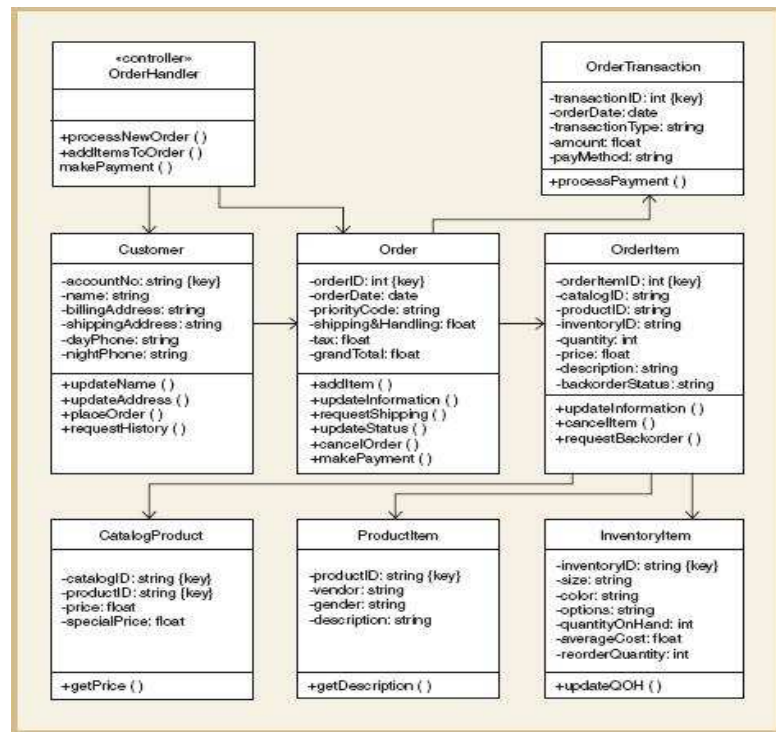


Gambar 2.8: *Domain Class Diagram*

Sumber: Satzinger, Jackson, dan Burd (2010:193)

2.4.5.6 First Cut Class Diagram

Menurut Satzinger (2010, p.413) *First Cut Design Class Diagram* dikembangkan dengan memperluas model *domain class diagram* dan memerlukan dua langkah yaitu mengelaborasi atribut-atribut dengan tipe dan nilai informasi inisial dan langkah ke dua adalah menambahkan panah navigasi visibilitas.



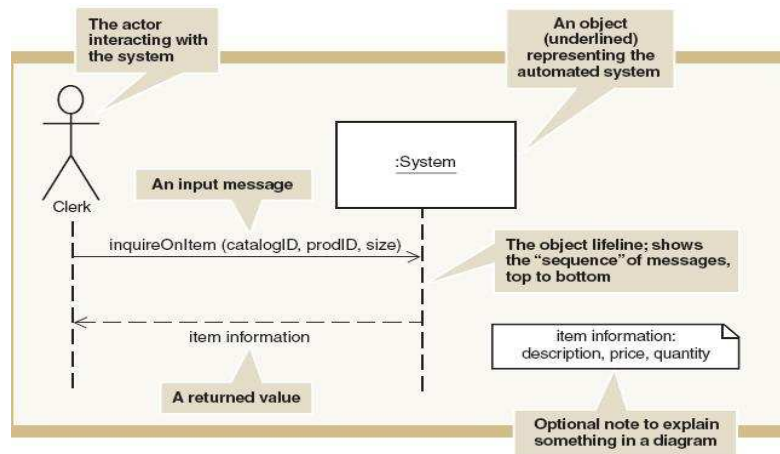
Gambar 2.9: *Updated Class Diagram*

Sumber: Satzinger, Jackson, dan Burd (2010:419)

2.4.5.7 System Sequence Diagram

Menurut Satzinger et al. (2010, p.252) *System sequence diagram* digunakan untuk menggambarkan aliran dari informasi yang masuk dan keluar dari sistem yang terotomatisasi. *System sequence diagram* merupakan tipe dari *interaction diagram* yaitu

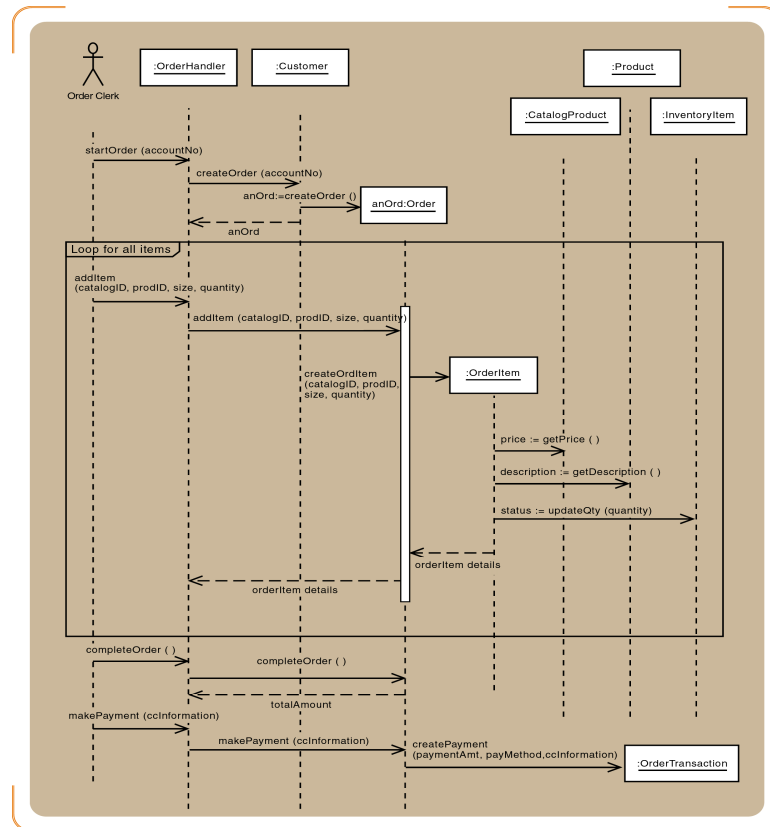
communication diagram atau *sequence diagram* yang menunjukkan interaksi diantara objek.



Gambar 2.10 *System Sequence Diagram*

2.4.5.8 Completed Three-Layer Design Sequence Diagram

Menurut Satzinger, Jackson, dan Burd (2010:434), *completed three-layer design sequence diagram* merupakan pengembangan dari *first-cut sequence diagram* dengan menambahkan data *access layer*.



Gambar 2.11 :Completed Three-Layer Design Sequence Diagram

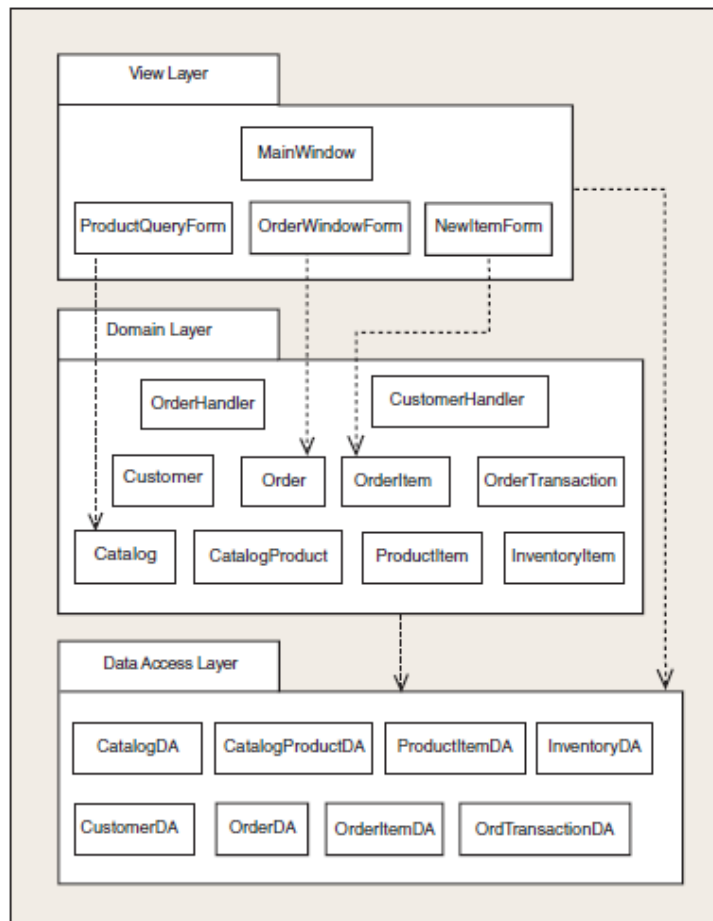
Sumber: Satzinger, Jackson, dan Burd (2010:445)

2.4.5.9 Package Diagram

Package diagram menurut Satzinger, Jackson, dan Burd (2010:459) merupakan diagram yang mengasosiasikan *class-class* dari suatu kelompok yang terkait. Di dalam diagram tersebut terbagi

menjadi tiga layer, yaitu *view layer*, *domain layer*, dan *data access layer*.

Package yang digunakan dalam diagram ini digambarkan dengan persegi panjang, sedangkan hubungan antar package digambarkan dengan anak panah bergaris putus-putus (*dashed arrow*), yang mewakili *dependency relationship*. Buntut panah terhubung dengan *dependent package*, sedangkan kepala panah terhubung dengan *independent package*. *Dependency relationship* sendiri menggambarkan suatu hubungan antar elemen dalam *package diagram*, dimana jika terjadi perubahan pada suatu elemen (elemen yang independent), maka elemen lainnya (elemen yang dependent) juga dapat berubah.



Gambar 2.12: *Package Diagram*

Sumber: Satzinger, Jackson, dan Burd (2010:459)

2.4.5.10 Design Database

Design Management System adalah perangkat lunak sistem yang mengelola dan mengontrol akses ke database.

Relational Database Management System adalah sebuah sistem manajemen database yang menyimpan data kedalam table.

Langkah-langkah dalam membuat Relational Database :

- I. Membuat sebuah table untuk setiap jenis entitas
- II. Memilih sebuah primary key untuk setiap entitas
- III. Menambah foreign key untuk setiap table untuk memrepresentasikan relasi one to many
- IV. Membuat table baru yang mempresentasikan relasi many to many
- V. Menentukan batasan integritas referensial
- VI. Mengevaluasi kualitas skema dan melakukan perubahan yang diperlukan
- VII. Memilih tipe data yang sesuai dan pembatasan nilai (jika perlu) untuk setiap bidang

ProductItem : Table

ProductID	Vendor	Gender	Description
1244		Man	Casual Chino Trousers
+ 1245		Man	Fleece Crew Sweatshirt
+ 1246		Man	Fleece Crew Sweatshirt V-Neck
+ 1247		Man	Fleece Crew Sweatshirt Zippered
+ 1248		Man	Solid Color Flannel Shirt
+ 1249		Man	Plaid Flannel Shirt
+ 1250		Man	Polo Shirt
+ 1251		Man	Polo Shirt Zippered
+ 1252		Man	Navigator Jacket
+ 1253		Man	Navigator Jacket Hooded
+ 1254		Man	Cotton Thermal Shirt

Record: 1 of 11

InventoryItem : Table

InventoryID	ProductID	Size	Color	Options	QuantityOnHand	Average Cost	RecorderQuantity
+ 86779	1244	30/30	Khaki		45	\$12.75	100
+ 86780	1244	30/30	Slate		10	\$12.75	100
+ 86781	1244	30/30	LightTan		17	\$12.75	100
+ 86782	1244	30/31	Khaki		22	\$12.75	100
+ 86783	1244	30/31	Slate		6	\$12.75	100
+ 86784	1244	30/31	LightTan		31	\$12.75	100
+ 86785	1244	30/32	Khaki		120	\$12.75	100
+ 86786	1244	30/32	Slate		28	\$12.75	100
+ 86787	1244	30/32	LightTan		21	\$12.75	100
+ 86788	1244	30/33	Khaki		7	\$12.75	100
+ 86789	1244	30/33	Slate		41	\$12.75	100
+ 86790	1244	30/34	LightTan		35	\$12.75	60

Record: 1 of 12

Gambar 2.12 Database Design

Satzinger, J.W., Jackson, R.B., & Burd, S.D.(2010,p492)

2.4.5.11 User Interface

User Interface Menurut Satzinger, Jackson dan Burd (2010:532) adalah bagian dari sistem informasi yang membutuhkan interaksi dari *user* untuk *membuat input* dan *output*. menjelaskan bahwa sebuah sistem informasi baru mempengaruhi banyak sistem informasi yang ada lainnya, dan analisis harus memastikan bahwa mereka semua bekerja bersama-sama. Beberapa *interface* sistem link sistem organisasi internal, merupakan sistem lain antarmuka dengan sistem eksternal, seperti pemasok atau rumah pelanggan. Dalam kasus lain, sistem baru perlu berkomunikasi dengan aplikasi bahwa organisasi telah dibeli dan diinstall. Dalam setiap kasus hanya terdaftar, analisis harus memiliki informasi tentang setiap sistem yang akan menyentuh sistem baru.

Sistem juga harus berinteraksi dengan pengguna baik didalam maupun diluar organisasi. *User interface* yang lebih dari sekedar layar, itu adalah merupakan pengguna yang datang ke dalam kontrak dengan saat menggunakan sistem, komseptual, dan fisik.

2.5 Rancangan Database

2.5.1 Pengertian Database

Menurut Satzinger (2009: 488), “*Database is an integrated collection of stored data that is centrally managed and controlled.*” Yang diterjemahkan; “*Database* adalah kumpulan dari sebuah data terintegrasi yang telah disimpan dimana data tersebut dikelola dan dikontrol secara terpusat.”

Berdasarkan *Journal The Winners* (Vol.9, No.1, Maret 2008: 90), “*Database* adalah semua data yang dibutuhkan untuk mendukung operasi organisasi. Di dalamnya meliputi aktivitas mengumpulkan, mengorganisasi, dan merawat secara tersentralisasi”.

Dari pengertian diatas, dapat disimpulkan bahwa database adalah suatu kumpulan data yang terintegrasi yang dapat dapat disimpan dan dikontrol secara terpusat.

Menurut Connolly, *database* adalah kumpulan koleksi data-data yang saling berhubungan secara logika yang isinya didesain untuk memenuhi kebutuhan informasi dari suatu perusahaan. Setiap *database* dapat berisi atau memiliki sejumlah objek *database*, antara lain yaitu:

1. *Field*

Field adalah sekumpulan kecil dari kata atau sebuah deretan dari angka-angka.

2. *Record*

Record adalah kumpulan dari *field* yang berelasi secara logis. Contohnya nama, alamat, nomor telepon dan lain sebagainya.

3. *File*

File atau berkas adalah kumpulan dari record yang berelasi secara logis. Contohnya berkas transaksi toko A yang

mempunyai *record* tanggal, kode barang dan harga.

4. *Entitiy*

Entitiy adalah orang, tempat, benda atau kejadian yang berkaitan dengan informasi yang disimpan. Contoh : pekerja, pelanggan dan sebagainya.

5. *Attribute*

Attribute adalah setiap karakteristik yang menjelaskan suatu *entity*. Contoh: nama karyawan, umur karyawan dan sebagainya.

6. *Primary Key*

Primary key adalah sebuah *field* yang nilainya unik yang tidak sama antara suatu *record* dengan *record* yang lain. *Primary key* digunakan sebagai tanda pengenal dari suatu field.

7. *Foreign Key*

Foreign key adalah sebuah *field* yang nilainya berguna untuk menghubungkan *primary key* lain yang berada pada tabel yang berbeda.

Untuk suatu perusahaan atau organisasi dapat mendapatkan keuntungan dari suatu *database*, maka perusahaan atau organisasi tersebut harus memiliki sebuah “*Database Management System*” yang baik.

2.5.2 Database Management System (DBMS)

Menurut Satzinger (2009: 448) “*Database management system is a system software that manages and controls access to database.*” Yang diterjemahkan; “*Database management system* adalah suatu *software* sistem yang mengelola dan mengendalikan akses ke *database*”.

2.5.3 Relational Database Management System (RDBMS)

Menurut Satzinger (2009: 490) “*Relational Database management system is a database management system that organizes stored data into structures called tables or relations.*” Yang diterjemahkan; “*Relational Database management system* adalah *database management system* yang mengorganisasikan data yang telah tersimpan dalam struktur yang disebut *table* atau *relation*.”

Berikut ini terdapat beberapa terminology mengenai *relational database management system*:

- a. *Tables*, struktur data dua dimensi yang berisi baris dan kolom, biasanya disebut juga *relation*.
- b. *Row*, bagian *table* yang berisi data yang menggambarkan satu entitas, *relationship*, atau objek, biasanya disebut juga *tuple* atau *record*.
- c. *Field*, kolom dari tabel *relational database*, biasanya disebut juga atribut.
- d. *Field value*, nilai data yang disimpan dalam *cell* tunggal dari tabel *relational database*, biasanya disebut juga nilai atribut atau elemen data.
- e. *Key, field* yang berisi nilai yang unik dalam setiap baris dari tabel *relational database*.
- f. *Primary key, key* yang digunakan secara unik untuk mengidentifikasi sebuah baris dari tabel *relational database*.
- g. *Foreign key*, nilai *field* yang disimpan dalam sebuah tabel *relational database* yang berperan sebagai *primary key* dalam tabel *relational database* yang lain.

2.5.4 Fase-fase Perancangan Database

Menurut Connolly dan Begg, perancangan *database* terdiri dari tiga fase utama berikut:

a. *Conceptual Database Design*

Proses membangun sebuah model data yang digunakan dalam sebuah perusahaan, namun independen terhadap semua pertimbangan fisik.

b. *Logical Database Design*

Proses membangun sebuah model data yang digunakan dalam sebuah perusahaan berdasarkan pada model data khusus, namun independen terhadap pertimbangan yang berkaitan dengan DBMS tertentu dan pertimbangan fisik lainnya.

c. *Physical Database Design*

Proses memproduksi deskripsi implementasi *database* pada media penyimpanan kedua; yang mana menggambarkan *base relation*, *file* organisasi, dan indeks yang digunakan untuk mencapai efisiensi akses terhadap data dan tuntutan adanya integritas serta penilaian keamanan.

2.5.5 Pengertian SQL

Menurut Connolly & Begg (2010, p.184), *SQL (Structured Query Language)* adalah contoh dari *transform-oriented language* atau bahasa yang didesain dengan menggunakan relasi untuk mengubah *input* menjadi *output* yang diinginkan.

Menurut Wahyu Gunawan (2010, p.15), *MySQL* adalah aplikasi atau sistem untuk mengelola *database* atau manajemen data. Untuk menyimpan data dan informasi ke komputer dengan data yang sudah ada, contohnya dengan menyimpan data karyawan pada suatu perusahaan dan memasukkan pada suatu *file*. *File* data yang dikelompokkan inilah yang disebut *database*, dan *MySQL* bertugas mengatur dan mengelola data – data pada *database*. Dalam mengelola *database MySQL* menggunakan

struktur atau kerangka yang berbentuk tabel. Dalam tabel–tabel itulah data diatur dan dikelompokkan.

2.5.6 Pengertian C#

Menurut P.J. Deitel dan H.M. Deitel (2010, p.37), C# berorientasi objek. Anda akan belajar beberapa dasar dari teknologi objek dalam waktu singkat dan akan mempelajari banyak penanganan di buku ini nantinya. C# adalah penulisan *event* kedalam program yang akan merespon kegiatan pengguna seperti saat *mouseclicks*, *keystrokes* dan *timer expirations*. C# dari *Microsoft Visual* adalah bahasa pemrograman *visual*. Selain itu, digunakan untuk menulis pernyataan program untuk membangun bagian-bagian dari aplikasi, dan juga *VisualStudio* ini menggunakan grafis yang terhubung dengan nyaman menarik dan meletakkan objek yang sudah dikenal seperti tombol dan kotak teks ke layar, dan menamai dan merubah ukuran *Visual Studio* akan menulis banyak kode GUI anda.

2.5.7 SQL Server

SQL Server memakai sebuah tipe database yang dinamakan database relasional. Database relasional adalah database mengorganisasikan data dalam bentuk tabel. Tabel dibentuk dengan mengelompokkan data yang mempunyai subjek yang sama. Tabel berisi baris dan kolom informasi. Tabel-tabel dapat saling berhubungan jika dihubungkan. SQL Server 2008 merupakan bahasa pemrograman yang dirancang khusus untuk berkomunikasi dengan database relasional guna mendukung aplikasi dengan arsitektur client/server Authentication windows.

2.5.8 Visual Studio

Visual Studio 2010 pada dasarnya adalah sebuah bahasa pemrograman komputer. Dimana pengertian dari bahasa pemrograman itu adalah perintah-perintah atau instruksi yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu.

Visual Studio 2010 (yang sering juga disebut dengan VB .Net 2010) selain disebut dengan bahasa pemrograman, juga sering disebut sebagai sarana (tool) untuk menghasilkan program-program aplikasi berbasis windows. Beberapa kemampuan atau manfaat dari Visual Studio 2010 diantaranya seperti :

1. Untuk membuat program aplikasi berbasis windows.
2. Untuk membuat objek-objek pembantu program seperti, misalnya : kontrol ActiveX, file Help, aplikasi Internet dan sebagainya.
3. Menguji program (debugging) dan menghasilkan program berakhiran EXE yang bersifat executable atau dapat langsung dijalankan.

➤ Keistimewaan Visual Studio 2010 Beberapa keistimewaan Visual Studio 2010 ini diantaranya seperti :

1. Menggunakan platform pembuatan program yang dinamakan developer studio, yang memiliki tampilan dan sarana yang sama dengan Visual C++ dan Visual J++. Dengan begitu Anda dapat bermigrasi atau belajar bahasa pemrograman lainnya dengan mudah dan cepat.
2. Memiliki compiler handal yang dapat menghasilkan file executable yang lebih cepat dan lebih efisien dari yang sebelumnya.
3. Memiliki beberapa tambahan wizard yang baru. Wizard adalah sarana yang mempermudah di dalam pembuatan aplikasi dengan mengotomisasi tugas-tugas tertentu.
4. Visual Studio 2010 mempunyai beberapa fitur untuk pengembangan berbagai macam aplikasi yang diantaranya; Windows Development,

Web Development, Office Development, Sharepoint Development, Cloud Development (Windows Azure), Silverlight Tooling, Multi-Core Development, Customizable IDE.

2.5.9 ASP.NET

ASP.NET merupakan *platform pembuatan aplikasi web yang menyatu dengan .NET Framework serta menyediakan fasilitas-fasilitas bagi developer untuk membangun aplikasi web untuk level enterpris.*

Kelebihan:

1. *ASP.NET merupakan managed code, dan berjalan pada semacam virtual machine dari Microsoft yang disebut Common Language Runtime (CLR).*
2. *ASP.NET merupakan compiled code menjadi IL (Intermediate Language) yaitu bahasa yang dikenali oleh CLR.*
3. *ASP.NET dapat menggunakan berbagai bahasa pemrograman yang mendukung .NET Framework*
4. *ASP.NET memiliki arsitektur web forms yang mempercepat development terlebih integrasi dan dukungan tools Visual Studio .NET yang sangat bagus sehingga pembuatan aplikasi web ASP.NET menggunakan VS.NET sangat efisien dan mempercepat waktu pembuatan aplikasi.*

Kekurangan:

5. *ASP.NET tidak dapat berjalan di platform non-microsoft sehingga tidak bisa gratis.*
6. *Project ASP.NET jarang diopensource kan sehingga kita akan kesulitan mencari komponen, library atau project gratis.*

2.5.9 Website

Menurut Connolly dan Begg (2010: 1028) Website adalah sistem yang menyediakan sarana informasi browsing di Internet dengan cara non-sequential menggunakan *hyperlink*. *Website* membentuk rangkaian

bangunan yang saling terkait satu dengan lainnya dimana masing-masing dihubungkan dengan jaringan-jaringan halaman.

2.6 Rancangan Formulir

2.6.1 Pengertian Rancangan Formulir

Menurut Jones dan Rama yang diterjemahkan oleh M Slamet Wibowo (2008,p354), “Formulir adalah dokumen terpola yang berisi *field* kosong yang dapat diisi pengguna dengan data. Ketika formulir ditampilkan di layar komputer, data yang dimasukkan di *field* kosong disimpan di suatu tabel data atau lebih.”

2.6.2 Pengertian Formulir

Menurut Jones dan Rama(2006,p288) “*Formisa formatted document containing blank fields that users can fillin with data*”,yang berarti dokumen yang terformat yang terdiri dari bagian yang kosong yang diisi oleh penggunanya.

2.6.3 Elemen Rancangan Formulir

Menurut Jones dan Rama yang diterjemahkan oleh M Slamet Wibowo (2008: 323-324), terdapat tiga jenis formulir input antara lain:

1. *Textboxes*. Seringkali ditempatkan pada suatu formulir yang digunakan untuk memasukkan informasi yang ditambahkan pada tabel atau untuk menampilkan informasi yang dibaca dari tabel.
2. *Labels*. Digunakan untuk membantu *user* dalam memahami informasi apa yang diperlukan untuk dimasukkan.
3. *Look up feature* . Sebuah daftar menu tarik atas pilihan yang sesuai saat memasukkan data dalam suatu *field* kosong tertentu pada sebuah formulir.
4. *Command buttons*. Digunakan untuk menampilkan suatu tindakan. Terkadang merupakan *push button* (tombol tekan), karena suatu tindakan akan dilakukan jika dipilih oleh *user*.

5. *Comboboxes*. Dalam bentuk sebuah tombol yang terdapat dalam suatu formulir elektronik yang memungkinkan *user* untuk memilih satu dari beberapa set pilihan.
6. *Checkboxes*. Merupakan suatu antar muka grafis atau sebuah kotak pada suatu formulir yang mengindikasikan apakah opsi tertentu telah dipilih.

2.6.4 Jenis-Jenis Formulir Input

Menurut Jones dan Rama yang diterjemahkan oleh M Slamet Wibowo (2008: 323-324), terdapat tiga jenis formulir input antara lain:

a. Formulir Entri Satu Record (*Single Record Entry Form*)

Formulir entri satu *record* hanya menampilkan satu *record* pada satu waktu. Formulir ini digunakan untuk menambah, menghapus, atau memodifikasi data di satu *record* satu tabel tertentu. Formulir seperti itu seringkali digunakan untuk mengarsipkan data *file* induk.

b. Formulir Entri Bentuk Tabel (*Tabular Entry Form*)

Formulir entri bentuk tabel menyediakan desain seperti kertas kerja untuk memasukkan banyak record di suatu tabel. Jenis formulir ini sering kali digunakan untuk mencatat sekumpulan kejadian.

c. Formulir Entri Multitabel (*Multitable Entry Form*)

Formulir entri multitabel digunakan untuk menambahkan data ke lebih dari satu tabel.

2.6.5 Manfaat Formulir

Menurut Puspitawati dan Anggadini (2011: 70) terdapat beberapa manfaat formulir dalam perusahaan antara lain:

- a. Menetapkan tanggungjawab timbulnya transaksi bisnis perusahaan.
- b. Merekam data transaksi bisnis perusahaan.

- c. Mengurangi kemungkinan kesalahan dengan menyatakan semua kejadian dalam bentuk tulisan.
- d. Menyampaikan informasi pokok dari satu orang ke orang lain di dalam organisasi yang sama atau ke organisasi yang lain.

2.7 Rancangan Layar

2.7.1 Pengertian Rancangan Layar

Menurut Satzinger, et al. (2009, p.549), Merancang *user interface* dapat diartikan merancang *input* dan *output* yang terlibat ketika *user* berinteraksi dengan komputer untuk melaksanakan suatu tugas.

2.7.2 Elemen-Elemen Rancangan Layar

Menurut Satzinger (2009: 551), elemen rancangan layar meliputi :

1. *Textboxes*

Input control yang menerima *data entry* dari keyboard, digunakan untuk menambahkan keterangan deskriptif untuk mengidentifikasi hal apa saja yang harus ditulis dalam *text boxes*.

2. *List box*

Input control yang berisi daftar yang terdiri dari *acceptable entries* yang sudah ditentukan sebelumnya agar dapat dipilih oleh pengguna.

3. *Spinbox*

Variasi dari *list box* yang menampilkan *multiple entries* dalam *text box* di mana pengguna dapat menentukan pilihannya.

4. *Combobox*

Variasi lainnya dari *list box* yang mengizinkan pengguna untuk memasukkan nilai baru ataupun memilih dari *entries* yang sudah ada.

5. *Radiobutton*

Input control yang memungkinkan pengguna memilih satu pilihan dari suatu kelompok.

6. *Checkboxes*

Input control yang memungkinkan pengguna memilih lebih dari satu pilihan dari suatu kelompok.

2.8 Rancangan Laporan

2.8.1 Pengertian Rancangan Laporan

Menurut Whitten & Bentley (2009, p.550), *Reports* atau *outputs* menggambarkan informasi bagi pengguna sistem. *Output* adalah komponen yang dapat dilihat dari sistem informasi yang bekerja atau berfungsi.

Menurut Connolly & Begg (2010, p.235), *Reports* adalah jenis khusus dari *continuous form* yang dirancang khusus untuk mencetak.

Dari pengertian di atas dapat disimpulkan bahwa Rancangan Laporan adalah pembuatan informasi yang berisi data yang telah diolah dari sistem informasi sehingga dapat bermanfaat dalam pengambilan keputusan di dalam suatu organisasi.

2.8.2 Elemen-Elemen Rancangan Laporan

Jones dan Rama (2006, p.214-215), menyatakan *Layout* laporan terdiri dari:

1. *Report header*. Menampilkan informasi keseluruhan laporan, seperti nama laporan dan perusahaan, tanggal laporan dan jumlah halaman informasi yang bersifat umum tiap kelompok.
2. *Group detail*. Berisi daftar transaksi yang berkaitan dengan kelompok.
3. *Group footer*. Bisa digunakan untuk menyediakan informasi yang berguna dalam kelompok laporan.
4. *Page footer*. Berada di bagian bawah setiap halaman dan biasanya termasuk halaman laporan.

2.8.3 Tipe-Tipe Laporan

Menurut Satzinger (2009: 585) terdapat empat tipe laporan yang dibutuhkan oleh pengguna:

- a. *Detailed report*, laporan yang berisi informasi spesifik dari transaksi bisnis, pegawai dapat menggunakan laporan ini untuk meneliti kembali transaksi yang telah terjadi sebelumnya.
- b. *Summary report*, berisi ringkasan atau hasil rekap dari informasi yang detail dalam suatu periode waktu tertentu dengan berdasarkan kategori tertentu, biasanya digunakan *middle manager* untuk meninjau kinerja divisi atau departemen.
- c. *Exception report*, hanya berisi informasi yang bersifat *nonstandard*, pengecualian, atau kondisi tertentu
- d. *Executive report*, ringkasan laporan yang berasal dari berbagai macam sumber informasi yang umumnya digunakan untuk membuat keputusan strategi..

2.9 Kerangka Pikir

Tabel 2.1 Kerangka Pikir



