

PEMBUATAN SISTEM INFORMASI DENGAN ANALISIS DAN PERANCANGAN BERORIENTASI OBJEK

Dewan Pelawi

Information Systems Department, School of Information Systems, Binus University
Jl. K.H. Syahdan No. 9, Palmerah, Jakarta Barat 11480
dewanpelawi2318@binus.ac.id; dewanpelawi@yahoo.com

ABSTRACT

To generate an information system which is in accordance with the needs of company requires an analysis of business processes by observing the running system. Solutions for problems found when doing the business are observed using the systems approach. Object-oriented analysis and design is a method currently used to build an information system. The system development with "Object Oriented Analysis & Design (OOAD)" method stated by Satzinger includes making activity diagram, event table, class diagram, usecase diagrams, usecase description, state chart diagram, deployment and software architecture, first-cut design class diagram, simple sequence diagram (SSD), the sequence diagram (first-cut, view layer, data access layer), communication diagram, updated design class diagram, package diagram, persistent object, user interface. The result obtained after performing all phases is an information system that fits the company needs which is expected to improve the performance and effectiveness in business processes and also support the company's business goals.

Keywords: *analysis, design, information systems, OOAD*

ABSTRAK

Untuk menghasilkan sistem informasi sesuai dengan kebutuhan perusahaan, hal utama yang dilakukan adalah melakukan analisis proses bisnis dengan mengamati sistem berjalan yang ada saat ini. Masalah yang ditemukan ketika melakukan proses bisnis dicari solusinya dengan pendekatan sistem. Metode analisis dan perancangan berorientasi objek adalah salah satu metode yang digunakan saat ini untuk membangun sistem informasi. Pembangunan sistem dengan metodologi Object Oriented Analysis & Design (OOAD) yang dikemukakan oleh Satzinger meliputi pembuatan activity diagram, event table, class diagram, usecase diagram, usecase description, state chart diagram, deployment and software architecture, first-cut design class diagram, simple sequence diagram (SSD), sequence diagram (first-cut, view layer, data access layer), communication diagram, updated design class diagram, package diagram, persistent object, user interface. Hasil yang diperoleh setelah melakukan seluruh tahapan ini adalah sebuah sistem informasi yang sesuai dengan kebutuhan perusahaan sehingga diharapkan dapat meningkatkan kinerja dan efektifitas dalam proses bisnis perusahaan dan juga menunjang tujuan bisnis.

Kata kunci: *analisis, perancangan, sistem informasi, OOAD*

PENDAHULUAN

Perkembangan teknologi dan sistem informasi membawa perubahan pada dunia usaha. Kemampuan menyerap teknologi yang dimanfaatkan untuk menunjang operasional dan proses bisnis banyak dilakukan, trend penggunaan sistem informasi pada perusahaan terjadi. Sistem informasi yang digunakan terdiri dari yang sudah jadi dan di jual bebas di pasaran atau dibangun sendiri. Kelemahan dari sistem yang dibeli bebas dipasaran adalah kurang flexibel dan kurang sesuai dengan kebutuhan bisnis yang terjadi sedangkan keuntungannya adalah implementasi yang cepat dan dapat langsung diterapkan. Sementara jika membangun sistem sendiri maka sudah pasti disesuaikan dengan kebutuhan proses bisnis dan sesuai dengan permasalahan yang terjadi dan memang kelemahannya dibutuhkan waktu untuk pembangunannya.

Menurut McLeod and Schell (2004, p9), sistem didefinisikan sebagai sekelompok elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan. Menurut Laoudon (2003, p7), secara teknis sistem informasi didefinisikan sebagai kumpulan-kumpulan komponen yang saling berhubungan, yang mengumpulkan (atau mengambil), memproses, menyimpan, dan mendistribusikan informasi untuk membantu dalam pengambilan keputusan, pengkoordinasian, pengendalian analisis dan menampikannya dalam suatu organisasi. Informasi adalah hasil pengolahan data menjadi sesuatu yang berguna bagi *user* yang menggunakannya. Menurut O'Brien (2005, p7), Sistem Informasi adalah kombinasi dari orang-orang, perangkat keras, piranti lunak, jaringan komunikasi dan sumber data yang terorganisasi yang mengumpulkan, mengubah dan menyebarkan informasi dalam sebuah organisasi.

Komponen dari sistem informasi adalah: (1) sumber daya manusia termasuk *end-user* dan IS specialist. *End-user* adalah orang yang menggunakan sistem informasi atau informasi yang dihasilkan dan orang yang mengembangkan dan menjalankan sistem informasi adalah IS specialist; (2) sumber daya perangkat keras merupakan komponen-komponen fisik seperti computer, dan bagian-bagiannya yang digunakan untuk mendukung pemrosesan informasi; (3) sumber daya piranti lunak seperti software, aplikasi software, dan prosedur, merupakan semua kumpulan perintah pemrosesan informasi; (4) sumber daya data yang terdiri dari fakta dan gambaran dari sesuatu yang secara umum tidak dapat digunakan oleh user; (5) sumber daya jaringan merupakan sumber pokok dari system informasi. Sumber daya jaringan seperti media komunikasi dan jaringan pendukung. Salah satu kunci sukses mendasar dalam mengemb angkan system adalah daur hidup pengembangan system (SDLC). Organisasi dan perusahaan menggunakan informasi untuk mendukung berbagai macam proses yang dibutuhkan oleh perusahaan untuk menjalankan fungsinya. Terdapat berbagai jenis informasi yang memiliki kegunaan tersendiri dalam mendukung proses bisnis. Setiap sistem informasi memiliki daur hidup system sendiri. Selama daur hidup tersebut sistem informasi menerima ide-ide, dirancang, dibangun dan dikembangkan selama dalam pengembangan proyek dan akhirnya diproduksi untuk digunakan mendukung bisnis. Walaupun dalam penggunaan system masih menerima perubahan untuk perbaikan, update atau dimodifikasi. Proses untuk membangun, menyebarkan, menggunakan, dan merubah sistem informasi dinamakan daur hidup pengembangan system (SDLC).

METODE

Analisis dan perancangan sistem informasi berbasis objek adalah salah satu metodologi pengembangan sistem informasi yang digunakan untuk membangun sistem informasi perusahaan. Metodologi ini dibagi menjadi dua bagian, yaitu analisis berorientasi objek (OOA) dan perancangan berorientasi objek (OOD). Langkah terakhir adalah melakukan coding berdasarkan langkah-langkah yang sudah disusun sebelumnya.

HASIL DAN PEMBAHASAN

Analisis Berorientasi Objek

Analisis berorientasi objek mendefinisikan seluruh tipe-tipe objek yang digunakan pada sistem dan menunjukkan kepada *user* kebutuhan yang diperlukan berinteraksi dengan system untuk menyelesaikan pekerjaan yang dilakukan. Teknik ini bertujuan untuk mempelajari objek yang ada dan mempertimbangkan apakah objek tersebut masih dapat digunakan lagi atau diambil lagi untuk penggunaan yang baru, juga digunakan untuk mendefinisikan objek-objek yang baru atau objek-objek yang sudah dimodifikasi yang akan digabungkan dengan objek yang sudah ada menjadi aplikasi komputasi yang berguna bagi bisnis. Objek adalah segala sesuatu yang memiliki *attribute* dan *behaviors*. Sebuah system computer dapat memiliki banyak tipe objek seperti objek *user interface* (UI) yang membuat *user interface*, *system domain* objek dan *problem domain* objek. *User interface* objek adalah objek-objek yang berinteraksi dengan *user* ketika menggunakan system seperti tombol-tombol (*button*), *menu item*, *text box* atau label. *Attribute* adalah karakteristik objek yang memiliki nilai seperti ukuran, bentuk, warna, lokasi, dan nama dari *button* atau label atau nama, alamat dan nomor telepon dari pelanggan. *Methods* adalah *behavior* atau operasi yang menggambarkan apa yang mampu dilakukan oleh objek. *Problem domain object* adalah objek yang khusus bagi aplikasi bisnis sebagai contoh objek pelanggan, objek order, objek produk dan sebagainya. *Class* adalah klasifikasi atau tipe objek yang memiliki *attribute* yang sama. Instan adalah sama dengan objek. *Messages* atau pesan adalah komunikasi antara satu objek dengan objek lainnya untuk menggunakan satu dari *methodnya*.

Perancangan Berorientasi Objek

Satzinger (2005:p60) mendefinisikan perancangan berorientasi objek adalah mendefinisikan seluruh tipe objek-objek yang penting untuk berkomunikasi dengan manusia dan peralatan dalam sistem dan menunjukkan bagaimana objek-objek saling berinteraksi untuk menyelesaikan pekerjaan tertentu dan memperbaiki defenisi masing-masing tipe objek sehingga dapat diimplementasikan dengan bahasa khusus atau lingkungan khusus. Langkah – langkah OOA&D adalah membuat *Activity Diagram*, *Event Table*, *Class Diagram*, *Usecase Diagram*, *Usecase Description*, *State Chart Diagram*, *Deployment and Software Architecture*, *First-Cut Design Class Diagram*, *Simple Sequence Diagram (SSD)*, *Sequence Diagram (First-cut , View Layer , Data Access Layer)*, *Communication Diagram*, *Updated Design Class Diagram*, *Package Diagram*, *Persistent Object*, *User Interface*.

Activity Diagram

Activity Diagram (Gambar 1) adalah sebuah tipe dari workflow diagram yang mendeskripsikan aktifitas *user* dan tahapan-tahapan pengerjaannya secara sekuensial. Diagram ini digambarkan dimulai dengan sebuah bulatan hitam kecil yang disebut sebagai awal kegiatan kemudian dilanjutkan dengan kegiatan/activity yang dilakukan yang digambarkan dengan suatu elips dan dihubungkan dengan konektor berupa tanda panah yang menunjukkan arah dari kegiatan. Kegiatan diakhiri dengan bulatan hitam kecil yang dilingkari yang merupakan akhir dari kegiatan.

Event Table

Event adalah Suatu kejadian di sebuah waktu tertentu dan tempat yang dapat digambarkan dan bernilai mengingat. *Trigger* adalah sinyal yang memberitahu sistem bahwa suatu peristiwa telah terjadi, baik kedatangan data membutuhkan pemrosesan atau suatu titik dalam waktu. *Source* adalah Agen eksternal atau aktor yang memasok data ke sistem. *Use Case* adalah apa yang sistem lakukan ketika peristiwa itu terjadi. *Response* adalah output dari sistem. Ketika sistem menghasilkan ringkasan

laporan transaksi, laporan-laporan adalah output. Satu kasus digunakan dapat menghasilkan beberapa tanggapan. *Destination* adalah Agen eksternal atau aktor yang menerima data dari sistem. Berikut adalah contoh event table (Gambar 2).

Class Diagram

Class Diagram (Gambar 3) adalah sebuah model grafikal yang digunakan di dalam pendekatan object oriented untuk menunjukkan class-class yang ada di dalam system. Class terdiri dari nama class, attribute dari class dan digambarkan dalam bentuk persegi panjang yang dibagi menjadi dua bagian yaitu bagian judul dan atribut.

Use Case Diagram

Use case diagram (Gambar 4) adalah sebuah diagram yang digunakan untuk menunjukkan beragam peran dari *user* dan bagaimana mereka menggunakan sistem yang ada. Komponen dari *use case diagram* adalah elips untuk menunjukkan use case, actor untuk menunjukkan *user* dan garis untuk menghubungkan actor dengan *use case*-nya.

Use Case Description

Use case description (Gambar 5) adalah deskripsi yang berisi daftar rincian pengolahan untuk kasus penggunaan. Dalam diagram UML, sebuah orang yang menggunakan sistem.

State Chart

Menurut Satzinger et al. (2005), *Statechart diagram* (Gambar 6) merupakan sebuah diagram yang menunjukkan siklus hidup dari sebuah objek dalam state dan transition. State chart diagram membutuhkan kombinasi dari states dan transistions diantara states untuk menyediakan mekanisme yang analyst dapat gunakan untuk menangkap business rules. State chart diagram di bentuk dalam oval untuk menunjukkan states dari object dan panah menunjukkan transisi. Kemudian dibentuk titik awal sebagai untuk permulaan yang disebut *Pseudostate*. Setelah transisi dimulai, berjalan sampai selesai dengan mengambil objek untuk state ke state baru.

Deployment and Software Architecture

Deployment Enviroment terdiri dari *Single Tier* yaitu Arsitektur yang memperkerjakan satu sistem computer yang mengeksekusi semua aplikasi perangkat lunak. *Multitier* adalah arsitektur yang mempekerjakan banyak sistem computer untuk mengeksekusi aplikasi perangkat lunak yang dibagi 2 jenis yaitu Arcitecture dan Multi computer. *Centralized Architecture* adalah Arsitektur yang menempatkan sumber daya komputasi semua di satu lokasi pusat. *Distributed Architecture* adalah Arsitektur yang menyebarkan sumber daya komputasi di beberapa lokasi yang terhubung dengan komputer jaringan.

First-Cut Design Class Diagram

First Cut Design Class Diagram (Gambar 7) dikembangkan dengan memperluas diagram domain model kelas. Hal ini membutuhkan dua langkah: (1) menguraikan atribut dengan tipe dan informasi nilai awal; (2) menambah visibilitas panah navigasi. Berikut adalah beberapa pedoman umum visibilitas navigasi: (1) satu-ke-banyak hubungan yang menunjukkan hubungan atasan / bawahan biasanya navigasikan dari atasan ke bawahan, misalnya, dari Order untuk OrderItem. Kadang-kadang hubungan bentuk hirarki rantai navigasi, misalnya, dari Katalog untuk ProductItem untuk Inventory Item; (2) wajib hubungan, di mana objek dalam satu kelas tidak bisa ada tanpa objek kelas lain, biasanya navigasikan dari kelas yang lebih independen kepada kelas yang tergantung pada

kelas lainnya, misalnya, dari Nasabah untuk Order; (3) ketika sebuah objek membutuhkan informasi dari obyek yang lain, sebuah panah navigasi mungkin diperlukan, baik untuk menunjuk obyek itu sendiri atau untuk *parent* dalam suatu hirarki; (4) navigasi panah juga dapat dua arah.

Simple Sequence Diagram

Simple sequence diagram (Gambar 8) adalah sebuah diagram yang menunjukkan urutan pesan antara eksternal aktor dan sistem selama kasus penggunaan atau skenario.

Sequence Diagram

Menurut Satzinger (2005), sebuah *sequence diagram* rinci menggunakan semua elemen yang digunakan pada SSD. Perbedaannya adalah system object digantikan oleh semua objek internal dan pesan dalam system. Berikut adalah contoh *first cut sequence diagram* (Gambar 9). Langkah terakhir dalam desain multilayer dari *use case* adalah menambahkan *view layer* (Gambar 10). Pada kebanyakan *use case*, *view layer* terdiri dari *user interface* tunggal *window*. Pada beberapa *usecase* yang lebih kompleks, dan membutuhkan beberapa *window* untuk masuk dan melihat data yang terkait dengan *use case* atau transaksi bisnis. Pada titik ini dalam desain, kita tidak masuk ke detail tentang isi dari *window* kelas itu sendiri. Cukup mengidentifikasi *window User Interface*. Rinci desain *user interface* jauh lebih kompleks. Desain rinci dari kelas *window*, termasuk control *window* dan bentuk tata letak. Contoh *data acces layer* dapat dilihat pada Gambar 11.

Communication Diagram

Communication diagram (Gambar 12) berhubungan dengan *Sequence Diagram*. *Communication diagram* menggunakan simbol yang sama untuk actor, object dan juga pesan sebagai *Sequence diagram*.

Update Design Class Diagram

Update design class diagram (Gambar 13) adalah perbaikan class diagram yang sudah dibuat sebelumnya agar dihasilkan class diagram yang lebih sempurna.

Package Diagram

Package diagram (Gambar 14) menunjukkan komponen terkait dan dependensi. Umumnya, kita menggunakan *Package diagram* untuk berhubungan kelas atau komponen sistem lainnya seperti node jaringan. Angka-angka sebelumnya menunjukkan dua menggunakan *Package diagram* untuk membagi sistem menjadi subsistem dan untuk menunjukkan bersarang di dalam paket.

User Interface

Bagian-bagian dari sistem informasi membutuhkan interaksi pengguna (Gambar 15) untuk membuat *input* dan *output*.

PENUTUP

Analisis dan pengembangan sistem berorientasi objek terdiri dari dua bagian, yaitu analisis berorientasi objek dan perancangan berorientasi objek di mana kedua bagian tersebut pada intinya fokus pada pembuatan *activity diagram*, *event table*, *class diagram*, *usecase diagram*, *usecase*

description, state chart diagram, deployment and software architecture, first-cut design class diagram, simple sequence diagram (SSD), sequence diagram (first-cut, view layer, data access layer), communication diagram, updated design class diagram, package diagram, persistent object, dan user interface yang merupakan satu rangkaian kesatuan yang tidak bisa dipisahkan. Satu tahap kegiatan merupakan bagian untuk melanjutkan ke tahap berikutnya. Setelah selesai menggambarkan seluruh kebutuhan untuk pengembangan sistem, diteruskan dengan *programming* yaitu merubahnya menjadi *source code* program kemudian di-*compile* menjadi program aplikasi yang siap untuk digunakan.

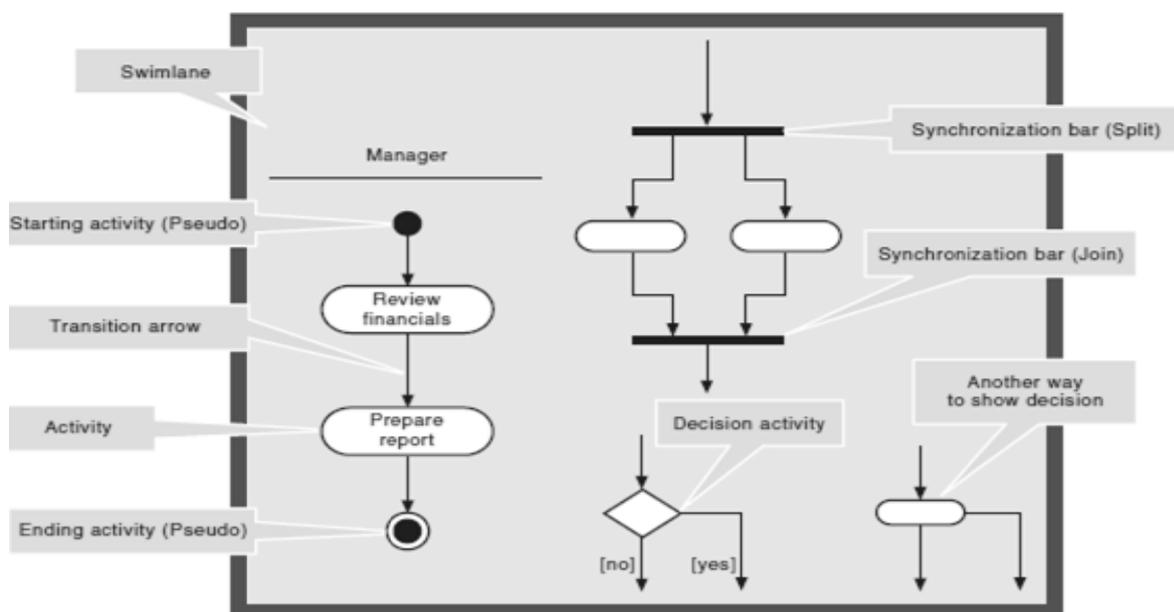
DAFTAR PUSTAKA

McLeod, Raymod Jr, and Schell, George. (2004). *Sistem Informasi Management* (edisi ke-8.) Jakarta: Indeks.

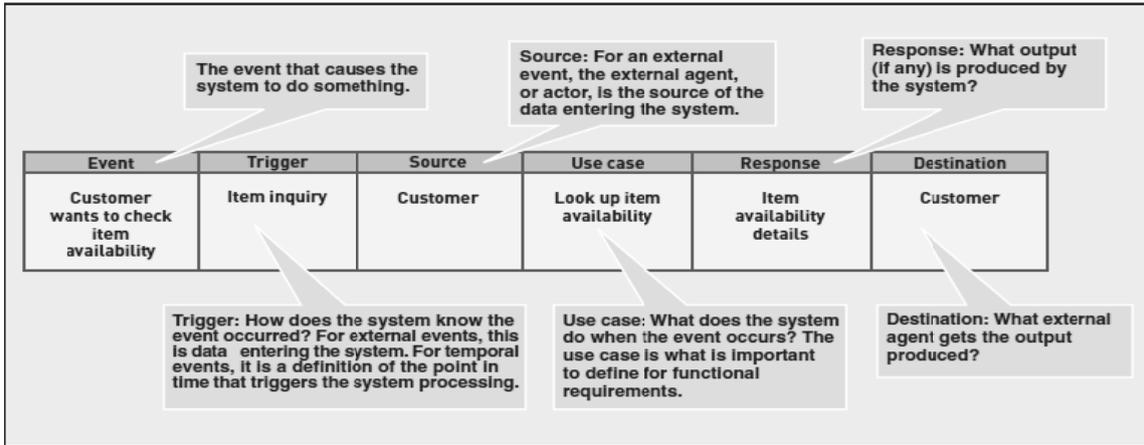
O'brien, James A. (2005). *Introducing to Information System*. New York: McGraw Hill.

Satzinger, J., Jackson, R., dan Burd, S. (2005). *System analysis and Design in Changing World* (5th edition). Boston: Course Technology.

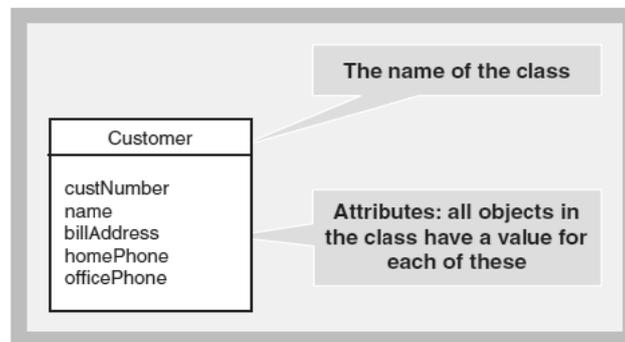
APPENDIX



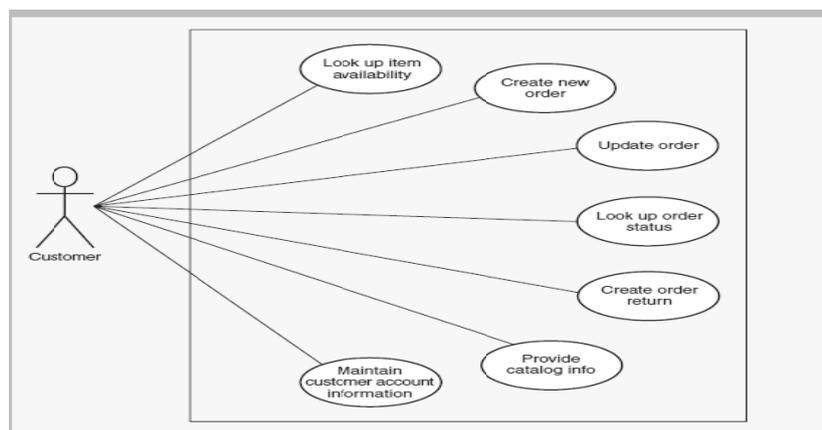
Gambar 1 Activity diagram (Satzinger, et al., 2005)



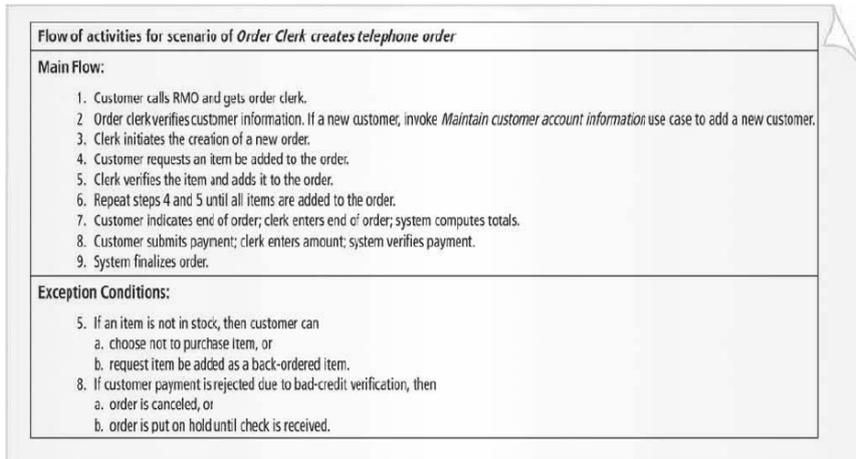
Gambar 2 Event table (Satzinger, et al., 2005)



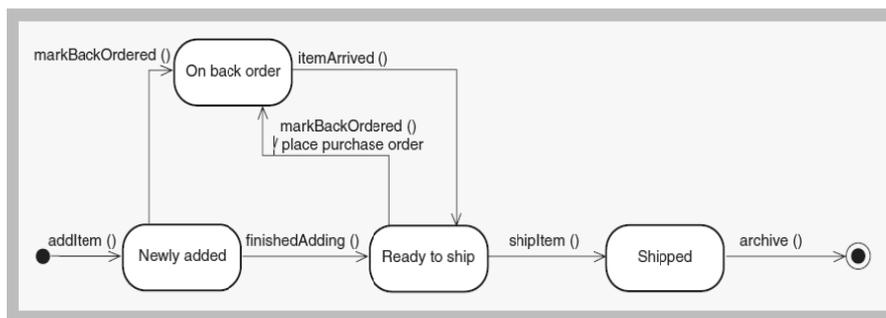
Gambar 3 Class Diagram (Satzinger, et al., 2005)



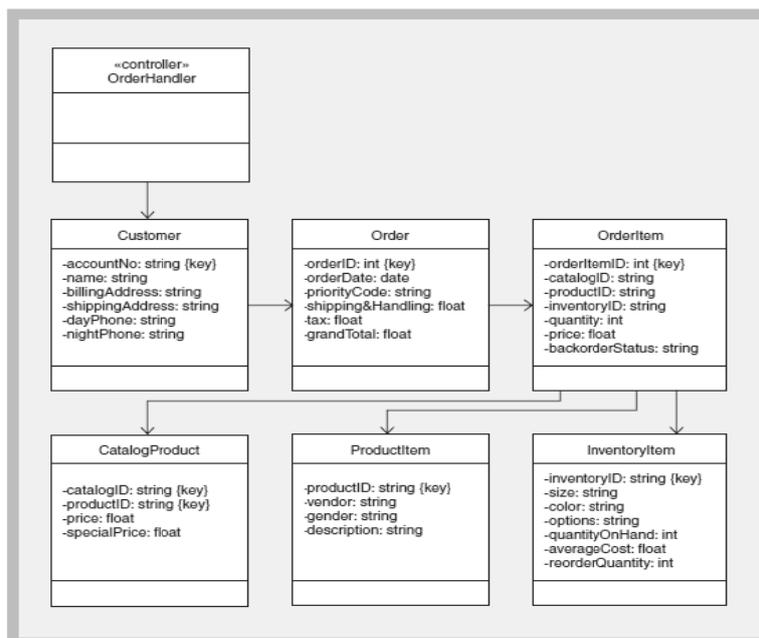
Gambar 4 Use Case Diagram (Satzinger, et al., 2005)



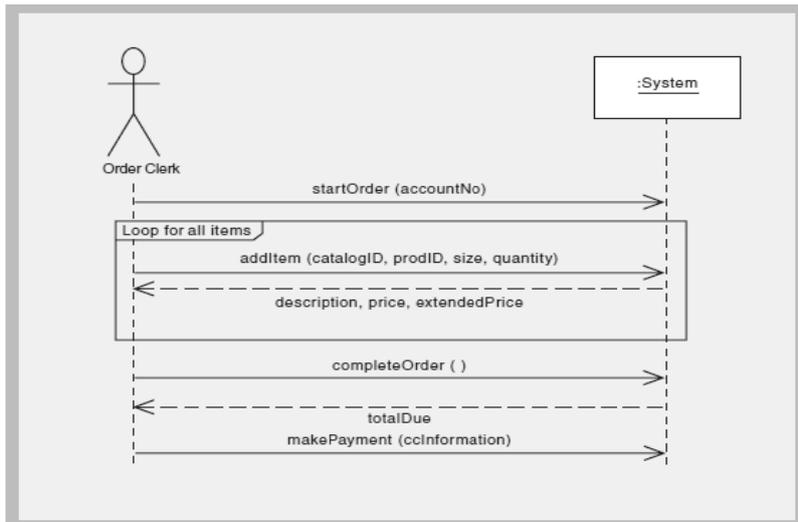
Gambar 5 Use case description (Satzinger, et al., 2005)



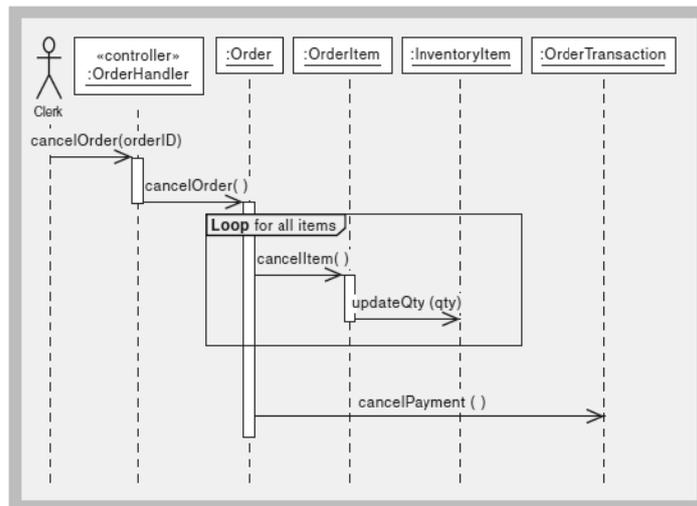
Gambar 6 State Chart (Satzinger, et al., 2005)



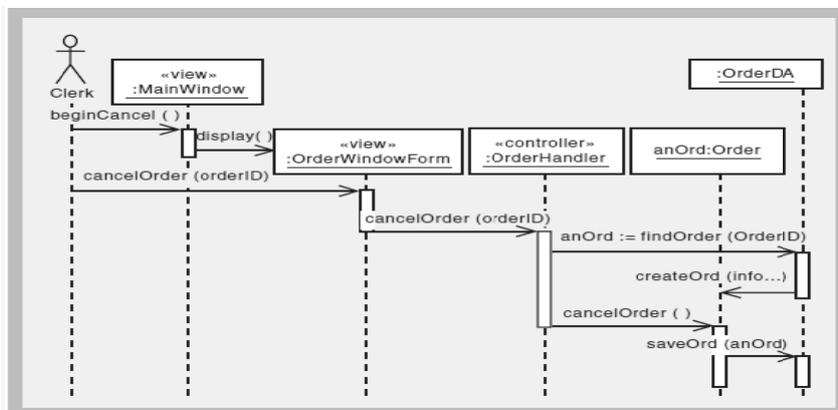
Gambar 7 First cut design class diagram (Satzinger, et al., 2005)



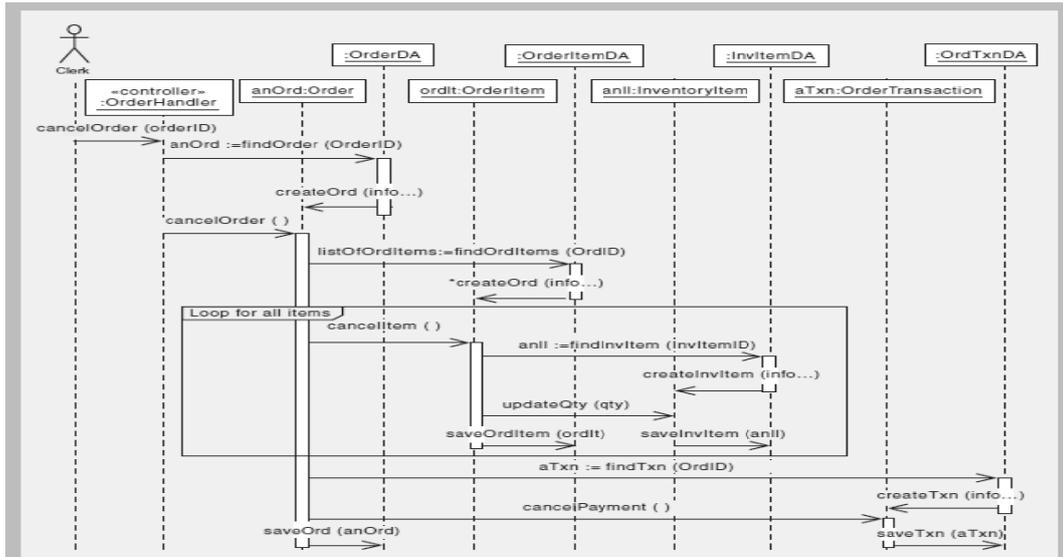
Gambar 8 Simple Sequence Diagram (Satzinger, et al., 2005).



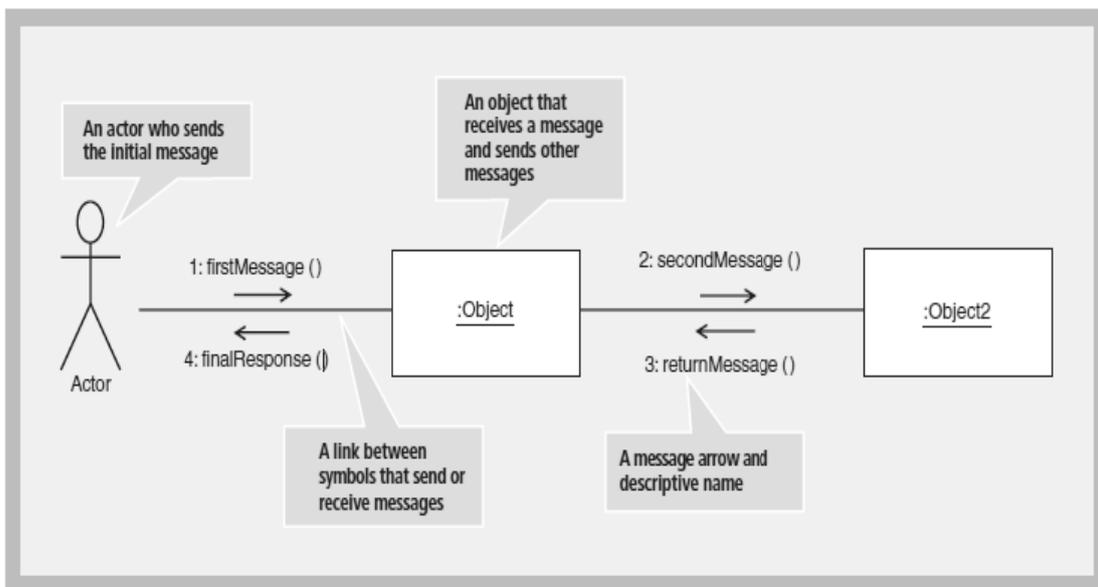
Gambar 9 First cut sequence diagram (Satzinger, et al., 2005)



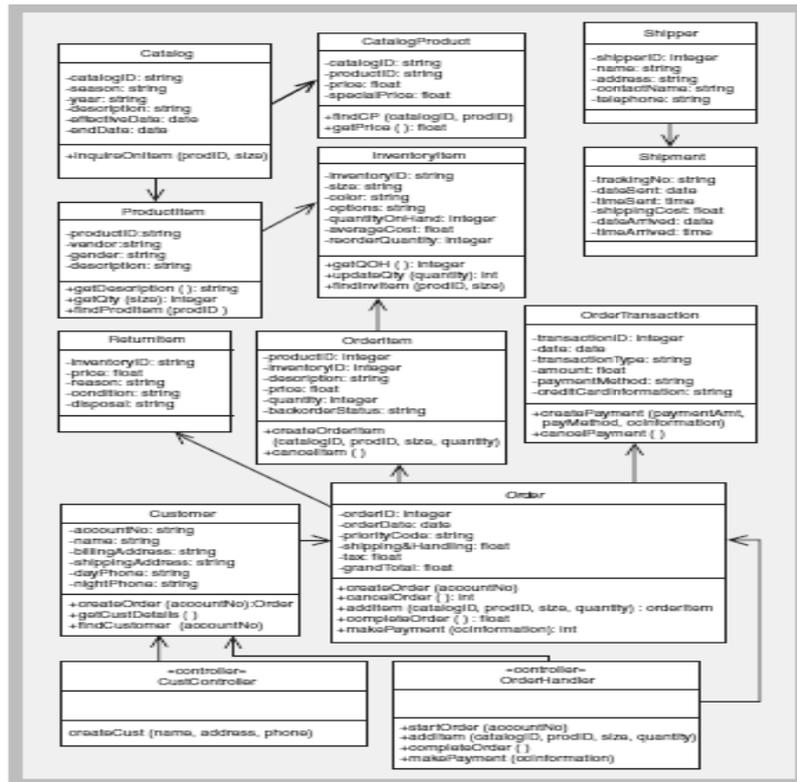
Gambar 10 View layer diagram (Satzinger, et al., 2005)



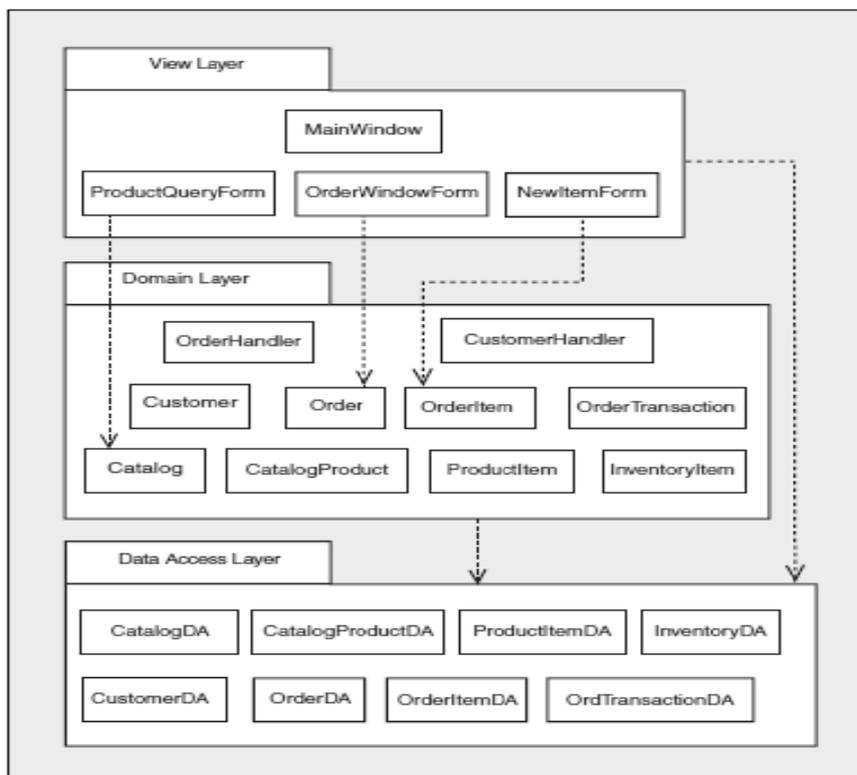
Gambar 11 Data access layer diagram (Satzinger, et al., 2005)



Gambar 12 Communication diagram (Satzinger, et al., 2005)



Gambar 13 Update design class diagram (Satzinger, et al., 2005)



Gambar 14 Package diagram (Satzinger, et al., 2005)

Customer Support System - Order Summary



Customer: Name: Number:

Order: Number: Date:

Products

ID	Description	Size/Color	Price	Quantity	Extension	
1	Toddler's mediumweight parka	G/Flod	\$44.95	1	\$44.95	
2						
3						
4						
5	<- Click here to display more order lines				Product Subtotal	\$44.95

Order Completion

Shipping Method: Shipping: \$18.50

Payment Method: Tax: \$2.42

Account Number: Credits: \$0.00

Total: \$65.87

Customer Support System - Shipping and Payment Options



Address: Shipping Address:

Country: Residence Business

Postal Code:

Shipping Options

Number of Items: Estimated Weight: Estimated Packages:

Shipping Method:

Shipping Comments:

Shipping Cost:

Payment Options

Payment Type: Product Subtotal: \$44.95

Account Number: Tax: \$2.42

Verification Status: Credits: \$0.00

Payment Total: \$65.87

Gambar 15 Contoh rancangan *user interface* (Satzinger, et al., 2005)