

TINJAUAN TEORITIS TENTANG MIGRASI PENDEKATAN RELASIONAL DATABASE KE PENDEKATAN OBJEK SISTEM INFORMASI AKUNTANSI

Weli
Fakultas Ekonomi Unika Atma Jaya Jakarta

Abstract

The object-oriented approach in system design and development is gaining popularity. Some literature in information systems demonstrate that object-oriented approach is viewed as superior to conventional system development such as relational database approach. There were many advantages of object-oriented approach such as easier modeling, more efficient model reuse and more convenient maintenance. In this paper we introduce you to object-oriented design for Accounting Information Systems by migrate the relational approach to object-oriented approach based on entity relationship diagram for revenue cycle.

Key words: *object oriented design, terminologi, karakteristik, sistem informasi akuntansi, pendekatan objek, perbedaan*

PENDAHULUAN

Latar Belakang

Sampai saat ini telah banyak literature sistem informasi yang membahas mengenai keunggulan metodologi berorientasi objek dalam berbagai aplikasi sistem informasi, termasuk didalamnya proses dalam melakukan analisis dan perancangan sistem informasi, bahkan dunia pemrogramanpun mulai beralih dari pendekatan terstruktur menjadi pendekatan objek.

Para pengembang sistem yang telah berpengalaman dalam menggunakan metodologi pengembangan sistem mengakui keunggulan dari metodologi ini,

yaitu pemodelan menjadi lebih mudah dan lebih efisien karena dapat dipergunakan kembali (*reusability*), dan dapat mengintegrasikan berbagai aplikasi dari berbagai sumber (*interoperability*).

Sehingga sistem yang dihasilkan mempunyai kualitas yang tinggi, lebih mudah dalam pemeliharaan sistem, dan lebih komunikatif antara aplikasi program dengan datanya. Walaupun metode ini masih terhitung baru, tetapi dalam perkembangannya metode ini telah banyak digunakan untuk kepentingan pembuatan program berbasis objek. Program yang dikembangkan sangat membantu manajemen sistem informasi dan penyaringan data akuntansi yang bermanfaat dalam proses pengambilan keputusan organisasi.

Melihat telah meluasnya program-program aplikasi maupun bahasa pemrograman yang berbasis objek dan pendekatan pengembangan sistem mempunyai hubungan yang erat dengan implementasi sistem yang diwujudkan di dalam bentuk pemrograman, sehingga dipandang perlu untuk memperkenalkan metode berorientasi objek dalam sistem informasi akuntansi terutama untuk pengelolaan database perusahaan.

Batasan dan Rumusan Masalah

Untuk membatasi permasalahan dalam penulisan maka pengenalan Metodologi berorientasi objek akan mengacu pada perancangan kembali data yang dibuat dalam bentuk Relasional Database ke bentuk pendekatan objek.

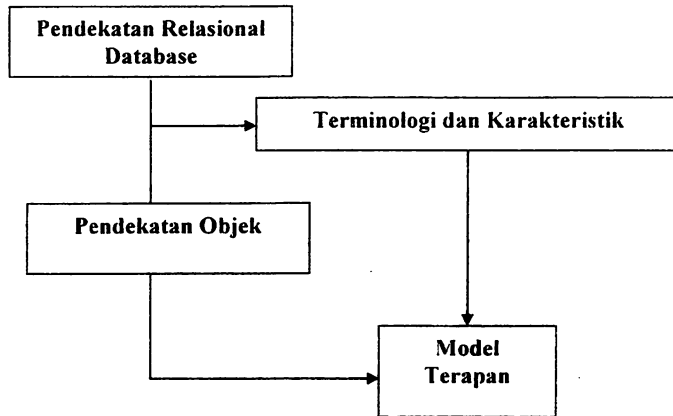
Pembahasan dimulai dari penjelasan ringkas mengenai perubahan pendekatan pengembangan sistem, terminologi dan karakteristik Metodologi Berorientasi Objek dan dilanjutkan dengan menggambarkan perbedaannya dengan model Relasional, serta melakukan migrasi data ke model objek.

Berdasarkan uraian batasan masalah diatas maka kelihatan bahwa hubungan antara pendekatan terstruktur dengan berorientasi objek sangat berbeda sehingga perlu ada penegasan antara persamaan dan perbedaan diantara kedua pendekatan tersebut, yang akan diuraikan dalam cara migrasi dari pendekatan data ke model objek.

Kerangka Pemikiran

Berdasarkan uraian singkat pada latar belakang, batasan dan rumusan masalah maka langkah-langkah pengidentifikasi perbedaan berawal dari adanya perubahan dalam terminologi dan karakteristik analisis dan disain

sistem informasi secara sederhana dapat digambarkan dalam alur pikir seperti dibawah ini.



KERANGKA TEORITIS

A. Perubahan Metode Untuk Pengembangan Sistem

Pendekatan yang umum digunakan dalam pengembangan sistem sejak tahun 1950-an sampai dengan 1970-an adalah pendekatan yang menekankan pada pengembangan terstruktur dengan menggunakan alat-alat pembuatan model proses seperti diagram alir (*flowchart*), dan diagram arus data (*Data flow diagram*) yang kemudian berkembang pada tahun 1980-an bergeser menjadi pendekatan model data menggunakan Diagram hubungan entitas (*Entity Relationship Diagram*) dan dilengkapi dengan pembuatan kamus data (*Data Dictionary*). Namun sejak tahun 1990-an kecenderungan telah berubah pengembangan sistem melakukan kombinasi antara model proses dan data menjadi model objek (*object oriented*). Dapat dikatakan bahwa metodologi pendekatan objek merupakan hasil evolusi dari banyak metodologi sebelumnya, seperti metodologi prosedural, sekuensial, konkurensi maupun modular. Tujuan dari metodologi pendekatan objek, bukanlah sebagai solusi untuk menggantikan metodologi-metodologi yang sudah ada sebelumnya, tetapi hanyalah sebagai salah satu alternatif pendekatan permasalahan untuk mencari solusi pemecahan, walaupun perkembangan selanjutnya metodologi pendekatan objek menjadi

sangat populer dan menjadi metodologi utama dan paling penting dalam melakukan pendekatan permasalahan.

Istilah Pendekatan objek mulai dikenal pada awal tahun 1967, melalui bahasa pemrograman yang bertujuan sebagai pemodelan atau simulasi yang bernama *simula*. *Simula* adalah bahasa pertama yang menggunakan metodologi pendekatan objek yang didalamnya sudah memiliki konsep dan prinsip dasar pendekatan objek.

Ada 2 (dua) alasan mengapa pengembangan sistem menggunakan pendekatan objek (*object-oriented*), pertama adalah *scalability* dimana objek lebih mudah dipakai untuk menggambarkan sistem yang besar dan kompleks. Kedua *dynamic modeling*, adalah dapat dipakai untuk permodelan sistem dinamis dan *real time*.

B. Terminologi Yang Digunakan Dalam Disain Sistem Berorientasi Objek

Bagian pertama dari penerapan pola pikir pendekatan objek adalah tentang bagaimana kita memandang keadaan dunia nyata menjadi pendekatan objek. Semua keadaan, kondisi alur peristiwa dan tata cara alam nyata ini akan dipandang dengan pendekatan objek (*object oriented*). Dalam mendekati permasalahan di dunia nyata kedalam metodologi pendekatan objek, yang pertama harus dikuasai adalah prinsip dan konsep dasarnya dulu, baru kemudian memandang sebuah persoalan dari dunia nyata menjadi sesuatu pendekatan objek.

1. Objek

Objek mewakili sebuah entitas, baik secara fisik, konsep ataupun secara perangkat lunak. Definisi yang formal dari objek adalah sebuah konsep, abstraksi atau sesuatu yang diberi batasan jelas dan dimaksudkan untuk sebuah aplikasi.

Objek dalam pengembangan sistem adalah sesuatu yang dapat dilihat dan dijamah atau dirasakan oleh panca indera kita. Objek dapat berupa benda atau suatu konsep dimana dapat disimpan data mengenai objek tersebut sehubungan dengan sifat dari benda yang bersangkutan.

Sebagai contoh Objek berupa benda misalnya Objek **Pelanggan** dan data yang disimpan didalamnya berupa Kode Pelanggan, Nama Pelanggan, Alamat, *Contact Person*, dan sebagainya.

Objek berupa konsep adalah sesuatu yang abstrak yang hidup didalam sistem seperti tabel, database, Peristiwa (*event*), Tujuan Sistem (*sistem messages*).

Selain itu sebuah objek dapat dikenali dari keadaan, kelakuan dan identitasnya. Keadaan dari objek adalah suatu kondisi yang memungkinkan objek dapat muncul, dan secara normal dapat berubah berdasarkan waktu. Keadaan dari objek dilihat dari kelompok propertinya yang disebut atribut. Atribut merupakan sesuatu yang menggambarkan keadaan atau nilai dari properti tersebut, dan hubungannya dengan objek lainnya.

Kelakuan objek adalah bagaimana sebuah objek beraksi dan bereaksi terhadap permintaan dari objek lainnya, yang diwakili dengan kelompok pesan yang direspon oleh objek. Kelakuan dari objek mendeskripsikan segala sesuatu yang dapat kita lakukan terhadap objek tersebut dan segala sesuatu yang dapat dilakukan oleh objek untuk kita.

2. Kelas

Setiap objek mempunyai identitas yang unik. identitas yang unik ini membuat kita dapat membedakan dua objek yang berbeda, walaupun kedua objek tersebut mempunyai keadaan dan nilai yang sama pada atributnya.

Objek dikelompokkan menjadi kelas-kelas objek dan kelas-kelas objek terdiri dari objek-objek yang memiliki atribut dan proses yang sama. Kelas objek dapat menjelaskan elemen di lingkungan perusahaan, sumber daya perusahaan, atau transaksi penting.

Dengan demikian Kelas adalah deskripsi dari kelompok objek dengan properti yang sama (atribut), Operasi yang sama, serta hubungan (relationship) dan semantik yang sama. Sebuah kelas adalah sebuah hasil abstraksi dari sesuatu dengan mengelompokkan karakteristik yang sejenis dengan mengabaikan karakteristik lainnya.

Contoh penerapan sebuah kelas adalah sebagai berikut:

Kita bisa mempunyai kelas dengan diberi nama persediaan, maka properti atau atribut dapat berupa kode persediaan, nama persediaan, packing, ukuran, harga, ROP, juga bisa mempunyai operasi seperti menambah persediaan, mengurangi persediaan, merubah persediaan, menghapus persediaan.

3. Atribut

Atribut adalah nama-nama properti dari sebuah kelas yang menjelaskan batasan nilainya dari properti yang dimiliki oleh sebuah kelas tersebut. Atribut dari suatu kelas merepresentasikan properti-properti yang dimiliki oleh kelas tersebut. Atribut mempunyai tipe yang menjelaskan tipe instansiasinya.

Dari contoh Objek **Pelanggan** maka data yang disimpan didalamnya disebut atribut yaitu Kode Pelanggan, Nama Pelanggan, Alamat, *Contact Person*.

Tipe data dari Kode Pelanggan dapat ditentukan misalnya berjenis *String* (non Numerik).

4. Operasi

Operasi menentukan bagaimana sebuah objek beraksi dan bereaksi terhadap permintaan dari objek lainnya, direpresentasikan dengan kelompok pesan yang direspon oleh objek. Operasi dari objek mendeskripsikan segala sesuatu yang dapat kita lakukan terhadap objek tersebut dan segala sesuatu yang dapat dilakukan oleh objek untuk kita.

Operasi adalah implementasi dari layanan yang dapat diminta dari sebuah objek dari sebuah kelas yang menentukan tingkah lakunya. Sebuah operasi dapat berupa perintah ataupun permintaan. Sebuah permintaan tidak boleh mengubah kedudukan dari objek tersebut. Hanya perintah yang dapat mengubah keadaan dari sebuah objek. Keluaran dari sebuah operasi tergantung dari nilai keadaan terakhir dari sebuah objek.

5. Pesan

Pesan merupakan suatu unit komunikasi untuk menghubungkan antara satu objek dengan objek lain. Suatu objek dapat meminta objek lain untuk melaksanakan suatu proses atau menyediakan data.

C. Karakteristik Sistem Berorientasi Objek

Sebuah sistem yang dibangun berdasarkan metode berorientasi objek akan menghasilkan Sebuah sistem yang komponennya dibungkus (dienkapsulasi) menjadi kelompok data dan fungsi. Dan Setiap komponen dalam sistem tersebut dapat mewarisi atribut dan sifat dari komponen lainnya, dan dapat berinteraksi satu sama lainnya.

1. Abstraksi

Adalah prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

2. Enkapsulasi

Adalah pembungkusan atribut data dan layanan (operasi-operasi) yang

dimiliki objek, untuk menyembunyikan implementasi dari objek sehingga objek lain tidak mengetahui cara kerjanya.

3. Pewarisan (*inheritance*)

Merupakan mekanisme yang memungkinkan suatu objek (kelas) mewarisi sebagian atau seluruh definisi dari objek lain sebagai bagian dari dirinya.

4. Reusability

Pemanfaatan kembali objek yang sudah didefinisikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek tersebut.

5. Generalisasi dan Spesialisasi

Menunjukkan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus.

6. Komunikasi antar objek

Dapat dilakukan lewat pesan (message) yang dikirim dari satu objek ke objek lainnya.

7. Polymorphism

Adalah kemampuan suatu objek untuk digunakan dibanyak tujuan yang berbeda dengan nama yang sama sehingga dapat menghemat pemakaian baris program.

D. Perbedaan Pendekatan Relasional dan Pendekatan Objek

Berdasarkan Terminologi dan Karakteristik yang telah diuraikan diatas, berikut ini diuraikan mengenai perbedaan dari kedua pendekatan tersebut.

1. Pendekatan Relasional

- a. Dalam pendekatan Relasional Database, Data direpresentasikan ke dalam bentuk tabel-tabel dengan kolom yang mewakili kategori dari data, dan baris yang berisi data itu sendiri. Setiap tabel hanya dapat berisi suatu tipe data yang terbatas dan sederhana seperti teks dan integer sehingga sulit untuk menangani tipe data yang kompleks seperti multimedia sehingga tidak dapat digunakan untuk aplikasi internet
- b. Data tidak terintegrasi dengan program sehingga proses penyimpanan dan pengambilan data menjadi kompleks

- c. Arsitektur dari aplikasi jadi lebih rumit dan membutuhkan keahlian pemrograman lebih lanjut saat memanggil suatu method dari objek tertentu pada database di server karena harus memasukkan *stored procedure* atau suatu komponen objek terlebih dahulu
- d. Kinerja program aplikasi lebih lambat karena harus melaksanakan pemetaan data dengan objek dan menguraikan table-table kedalam objek tersebut.

2. Pendekatan Objek

- a. Dalam pendekatan objek data direpresentasikan sebagai sebuah objek, baik dalam hal pengaksesannya maupun dalam hal pemodelannya dirancang untuk menangani tipe data multimedia yang banyak digunakan di internet
- b. Program dan data terintegrasi dengan baik sehingga proses penyimpanan dan pengambilan data jauh lebih sederhana
- c. Aplikasi dapat memanggil suatu method dari objek tertentu pada database di server sehingga distribusi aplikasinya lebih mudah dan menyederhanakan pembuatan aplikasi dengan meminimalisasi penggunaan bahasa pemrograman serta implementasi teknologi yang dibutuhkan
- d. Pemrogram tidak perlu lagi menguraikan objek ke tabel, memikirkan relasi antar tabel, dan sebaliknya. kinerja program dapat lebih baik karena hal-hal di atas tidak lagi diperlukan. Karena program langsung mengakses data dengan objeknya

Dengan uraian diatas kita dapat meringkas perbedaan dari kedua pendekatan, yaitu untuk melakukan penyimpanan data pada pendekatan objek digunakan istilah Kelas sedangkan pada pendekatan relasional database digunakan Tabel yang masing-masing berisi atribut atau elemen data yang dijelaskan dengan tipe datanya masing-masing. Namun pada pendekatan objek kelas dilengkapi dengan operasi sedangkan pada table hal ini terpisah dan harus dibuat fungsi tersendiri diluar dari Tabel. Data yang sudah didefinisikan pada pendekatan objek dapat dipergunakan kembali untuk kelas lainnya, atau objek lainnya, hal ini tidak dimiliki pada Tabel.

Secara sederhana uraian tersebut dapat dilihat pada tabel dibawah ini.

Tabel 1.
Perbandingan Pendekatan Data dengan Pendekatan Objek

Pendekatan Relasional	Pendekatan Objek
Tabel	Kelas
Record	Objek
Atribut	Atribut
Fungsi	Operasi

E. Mekanisme Migrasi Pendekatan Data Menjadi Pendekatan Objek

Sebelum melakukan migrasi pendekatan data ke pendekatan objek terlebih dahulu akan diperkenalkan secara singkat mengenai pointer dan array.

Pointer merupakan acuan yang menyimpan nilai mengenai objek dalam memory, dengan adanya pointer ini objek dapat diakses dan dimanipulasi.

Array, merupakan kumpulan dari objek dengan tipe data yang sama. Setiap elemen dalam array dapat diakses dengan posisi elemen dalam sebuah array.

Selanjutnya migrasi dapat dilakukan melalui beberapa tahap berikut ini :

1. Merubah Tabel Entitas menjadi Kelas, dalam hal ini proses tidak terlalu sulit karena Tabel dan Kelas memiliki kesamaan yaitu sebagai tempat menyimpan data. Tabel dan Kelas sama-sama memiliki atribut untuk data yang akan disimpan didalamnya.
2. Memperbaharui model Relasi yang baru,
 - a. Untuk Relasi 1 - 1 (*one to one*), kita harus menambahkan sebuah Kelas sebagai atribut data pada Kelas yang lainnya. Jika salah satu Kelas merupakan bagian dari Kelas lain maka Kelas yang merupakan bagian tersebut harus menjadi atribut data pada Kelas Induknya, Namun jika Relasi 1-1 ini mempunyai kedudukan yang sama tidak menjadi masalah dalam penempatan atribut data pada Kelas yang manapun.
 - b. Untuk Relasi 1- m (*one to many*), kita harus menambahkan atribut data Kelas pada sisi 1 ke Kelas pada sisi m, namun jika Kelas pada sisi m ingin ditambahkan ke sisi 1 Kelas pada sisi m harus diubah dalam bentuk array agar dapat menyimpan lebih dari satu data.
 - c. Untuk Relasi m-m (*many to many*), kita harus menambahkan array dari Kelas yang satu sebagai anggota pada Kelas lainnya. Jika kelas yang satu merupakan bagian dari kelas lainnya maka array attribut data dari Kelas tersebut menjadi atribut dari Kelas Induknya.
3. Setiap objek digambarkan dalam satu kesatuan, dalam hal ini digambarkan dalam bentuk kotak. Nama Objek ditulis pada baris atas kotak, atribut ditulis

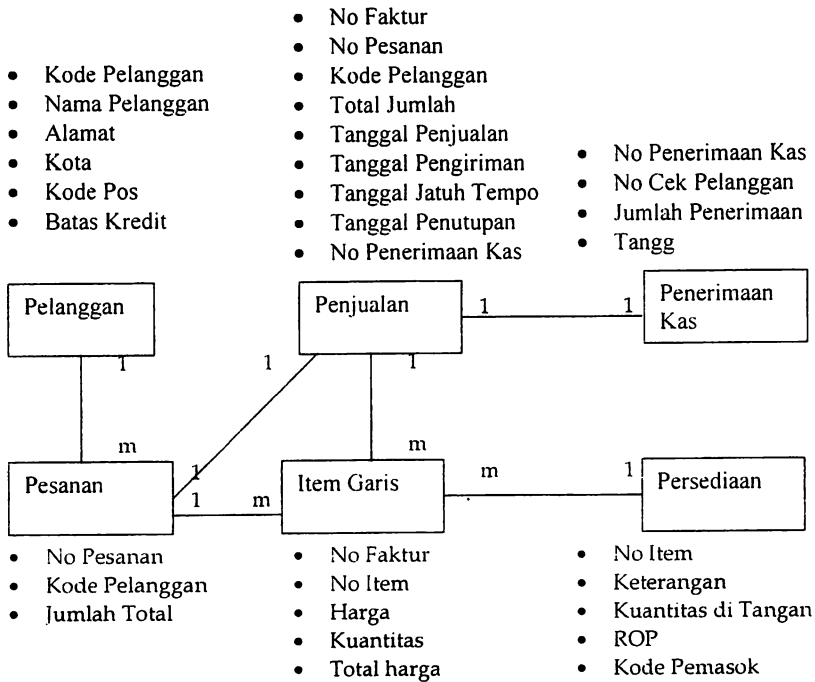
pada bagian tengah kotak, dan operasi ditulis pada bagian akhir kotak.

Dengan memasukkan Kelas kepada Kelas lainnya sebagai atribut. Rancangan Objek dari hasil migrasi akan ter-enkapsulasi, yaitu atribut data dan layanan (operasi-operasi) yang dimiliki objek dibungkus menjadi satu kesatuan dalam sebuah Objek. Penggunaan array akan mengeliminasi kebutuhan menciptakan link (hubungan) antar Tabel pada model Relasional Database untuk relasi m-m (many to many), karena pada pendekatan objek, sebuah objek dapat melakukan hubungan langsung ke Kelas lainnya tanpa membuat tambahan Tabel baru seperti pada pendekatan Relasional.

Contoh

Untuk mempertegas uraian di atas berikut akan diberikan contoh migrasi dari model relasional sistem Penjualan ke model objek

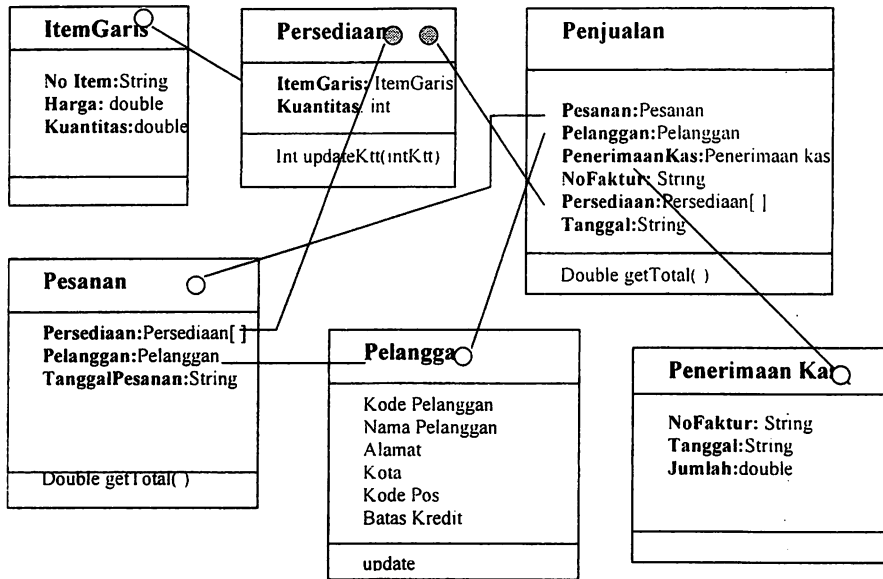
Model Relasional Database Sistem Penjualan



Gambar 1. Sistem Penjualan Dengan Model Relasional

Model Objek Sistem Penjualan dan Penerimaan Kas

Dengan menggunakan Diagram Relasi Entitas di atas, kita dapat melakukan migrasi ke Model Objek dengan menggunakan tahapan yang telah diuraikan diatas menjadi seperti yang digambarkan berikut ini



Gambar 2. Sistem Penjualan Dengan Model Objek

Sistem Penjualan seperti digambarkan diatas dapat dijelaskan sebagai berikut:

Peristiwa Pelanggan menempatkan pesanan dan pembayaran atas penjualan akan menimbulkan beberapa Objek, yaitu "Pesanan", "Penerimaan Kas", "Penjualan", sedangkan objek "Pelanggan" juga akan dimunculkan jika belum tersimpan dalam database perusahaan. Setiap objek yang muncul akan disimpan dalam database. Saat transaksi selesai sebuah "Pesan" akan dikirim ke objek "Persediaan" untuk meng-*update* data persediaan dengan mengurangi jumlah dengan yang terjual dari setiap ItemGaris.

1. Objek "Pesanan", berisi pointer (ujung dengan lingkaran putih) ke objek "Pelanggan" dan mempunyai pointer ke array objek (ujung dengan lingkaran hitam) "Persediaan". Setiap elemen dalam array mengacu ke item garis dan jumlah dalam pesanan. Objek ini juga mempunyai

“TanggalPesanan” untuk menyimpan waktu kapan pesanan ini diterima. Objek Pesanan juga dilengkapi dengan operasi untuk menghitung total jumlah pesanan.

2. Objek “Penjualan”, berisi pointer ke objek-objek yang berhubungan dengan peristiwa penjualan, yaitu objek “Pesanan”, “Pelanggan”, “Penerimaan Kas”, dan array objek “Persediaan”, selain itu juga berisi atribut No Faktur, tanggal faktur.
3. Objek “Pelanggan”, berisi seluruh informasi mengenai pelanggan seperti nama, alamat, dan sebagainya.

PENUTUP

Dari proses migrasi yang dilakukan pada sistem penjualan terlihat bahwa rancangan dalam pendekatan relasional database dapat digunakan untuk pendekatan objek, hal ini disebabkan karena masing-masing pendekatan sebenarnya memiliki atribut data yang sama, hanya saja dalam pendekatan objek setiap data telah dibungkus sedemikian bersama dengan kelas objeknya sehingga saat implementasi, data dan aplikasi telah melebur menjadi satu kesatuan, tidak seperti pada pendekatan data yang memisahkan data dengan aplikasinya. Beberapa tambahan sederhana untuk melakukan relasi ulang pada pendekatan objek kita perlu memahami istilah *pointer* dan *array*.

Keunggulan pendekatan ini seperti yang telah diuraikan pada awal tulisan ini kebenarannya dapat diterima. Bahwa pemodelan yang mudah dan prinsip dapat dipergunakan kembali untuk objek lain menyebabkan implementasinya menjadi efisien.

Bila dilihat perkembangan penggunaan konsep pendekatan objek pada pemrograman, dan mengingat pentingnya konsep ini dalam perancangan sistem serta migrasi yang dilakukan tidak terlalu rumit, maka sebagai penutup tulisan ini penulis menyarankan untuk memperkenalkan pendekatan objek dalam aplikasi sistem informasi akuntansi.

DAFTAR PUSTAKA

Coad, Peter. Yourdon, Edward, 1991, *Object-Oriented Design*, International Edition, Prentice Hall Inc.

- Harris, David, 1999, *System Analysis And Design: for the small enterprise*, 2nd edition, Dryden Press, Orlando, FL 32887-6777: Harcourt Brace College Publishers.
- Irman, Heru. 2001. *Object-oriented Methodology: sebuah fenomena di Dunia Riset dan Aplikasi Teknologi Informasi*, Gematel 28 - Artikel Lepas-1.htm
- McLeod, Jr, Raymond, and Schell, George, 2001, *Management Information Systems*, 8th edition, Prentice Hall International, Inc, Upper Saddle River New Jersey 07458.
- Whitten, Jeffrey L, and Bentley, Lonnie D, and Dittman, Kevin C, 2001, *Systems Analysis and Design Methods*, 5th edition, Irwin MC-GrawHill Companies, Inc, New York.

