

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

Weli

Fakultas Ekonomi, Unika Atma Jaya Jakarta

E-mail: bieimbiri@yahoo.com

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 and REA diagram for revenue cycle.

Keywords: ER Diagram, REA Model, object oriented design, terminologi, karakteristik.

I. 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*).

Walaupun metode ini masih terhitung baru, tetapi dalam perkembangannya metode ini telah banyak digunakan untuk kepentingan pembuatan program berbasis objek.

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.

II. 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 atau yang biasa dikenal dengan model ER, Model resource Event dan Agent (REA) ke bentuk pendekatan objek. Sebagai contoh untuk

pembahasan ini akan digunakan sistem penjualan sebagai modelnya.

Berdasarkan uraian sebelumnya terlihat bahwa hubungan antara pendekatan terstruktur dengan berorientasi objek ada perbedaan sehingga perlu ada penegasan antara persamaan dan perbedaan diantara pendekatan-pendekatan tersebut, yang akan diuraikan dalam cara migrasi dari pendekatan data ke model objek.

III. TUJUAN PENULISAN

Seperti yang telah diuraikan pada latar belakang maka tujuan penulisan paper ini adalah untuk memberikan gambaran ringkas mengenai aplikasi sistem informasi akuntansi yang berorientasi objek dan bagaimana disain sistem informasi akuntansi yang menggunakan model objek.

IV. KONSEP TEORITIS

• 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.

- **Model Entity Relationship (ER)**

Model ER merupakan suatu model untuk menjelaskan hubungan antara data dalam suatu database, dimana model ini menggunakan Diagram yang menjelaskan hubungan antara Entity dan bagaimana hubungan setiap Entity dengan Entity lain atau Relationship. Model E-R diperkenalkan oleh Peter Chen pada tahun 1976, walaupun sebelumnya prinsip relational database ini pertama kali dibahas oleh E.F. Codd dalam seminal papernya yang berjudul "A Relational model of data for large shared data banks" tahun 1970.

Setiap Organisasi dapat menggunakan diagram ER ini untuk memodelkan seluruh transaksi bisnisnya dan juga untuk berbagai keperluan bisnis lainnya. Diagram ER tidak saja menggambarkan isi dari suatu database tetapi secara grafis dapat memperlihatkan model organisasi terkait.

- **Model REA**

Model REA adalah suatu alat pemodelan konseptual yang khusus dirancang untuk melengkapi struktur dalam perancangan database SIA. Model ini berfokus pada aspek semantic bisnis yang mendasari aktivitas rantai nilai suatu organisasi, sehingga tidak semua kegiatan bisnis digambarkan dalam model tersebut namun harus diidentifikasi terlebih dahulu entitas apa yang seharusnya dimasukkan dalam database. Agak berbeda dengan pendekatan ER, Model REA mengklasifikasikan entity menjadi tiga kategori, yaitu: *Resources*, *Events*, dan *Agents*.

Resources didefinisikan sebagai sesuatu yang memiliki nilai ekonomis bagi organisasi tersebut. *Events* menunjukkan aktivitas-aktivitas bisnis, dimana manajemen ingin mengumpulkan informasi untuk tujuan perencanaan atau pengawasan. *Agents* adalah orang dan organisasi yang berpartisipasi dalam aktivitas dan kepada siapa informasi diserahkan untuk tujuan perencanaan, pengawasan, dan pengevaluasian.

Dalam Model REA data dimodelkan dengan mengikuti pola dasar R-E-A dimana setiap *Entity event* dihubungkan dengan *Entity resources* yang berpengaruh secara langsung atau tidak langsung. Setiap *Entity event* juga dihubungkan dengan *Entity agent*.

- **Hubungan Diagram ER dengan Model REA**

Baik Diagram RE maupun Model REA keduanya adalah model yang digunakan untuk menggambarkan (disain) Sistem Informasi Akuntansi model data bisnis suatu organisasi. Model REA dapat didokumentasikan dalam bentuk diagram RE yang menampilkan entitas data yang dikumpulkan dan memperjelas pentingnya hubungan antar data. Tingkat hubungan yang telah diuraikan pada Model ER juga diterapkan pada Model REA untuk menentukan tingkat hubungan antar Entity dan juga menyediakan informasi mengenai

kebijakan bisnis dasar yang diterapkan organisasi yang dikenal dengan istilah Cardinality.

Tipe Relationship tergantung pada maximum cardinality yang menghubungkan setiap Entity, ada tiga tipe Relationship:

1. Relationship one-to-one (1:1) pada saat maximum cardinality setiap Entity adalah 1
2. Relationship one-to-many (1:N) pada saat maximum cardinality dari satu Entity adalah 1 dan maximum cardinality dari Entity lain adalah N.
3. Relationship many-to-many (N:M) pada saat maximum cardinality kedua Entity adalah N.

Hubungan ini dapat dilihat pada **Gambar 1**.

- **Disain Sistem Berorientasi Objek**

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.

1. **Objek**
Definisi yang formal dari objek adalah sebuah konsep, abstraksi atau sesuatu yang diberi batasan jelas dan dimaksudkan untuk sebuah aplikasi. Objek dapat berupa benda atau suatu konsep dimana dapat disimpan data mengenai objek tersebut sehubungan dengan sifat dari benda yang bersangkutan.
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.
3. **Atribut**
Atribut dari suatu kelas merepresentasikan properti-properti yang dimiliki oleh kelas tersebut. Atribut mempunyai tipe yang menjelaskan tipe instansiasinya. Hubungan dari objek, atribut dan operasi dapat dilihat pada **Gambar 3**.
4. **Operasi**
Operasi menentukan bagaimana sebuah objek beraksi dan bereaksi terhadap permintaan dari objek lainnya, direpresentasikan dengan kelompok pesan yang direpson oleh 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.

6. Enkapsulasi
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.
7. Pewarisan (*inheritance*)
Merupakan mekanisme yang memungkinkan suatu objek (kelas) mewarisi sebagian atau seluruh definisi dari objek lain sebagai bagian dari dirinya.

• Pendekatan Relasional VS Pendekatan Objek

Berdasarkan terminologi dan karakteristik yang telah diuraikan di atas, berikut ini diuraikan mengenai perbedaan dari kedua pendekatan tersebut.

Pendekatan Relasional

1. 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
2. Data tidak terintegrasi dengan program sehingga proses penyimpanan dan pengambilan data menjadi kompleks
3. 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
4. Kinerja program aplikasi lebih lambat karena harus melaksanakan pemetaan data dengan objek dan menguraikan table-tabel kedalam objek tersebut.

Pendekatan Objek

1. 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.
2. Program dan data terintegrasi dengan baik sehingga proses penyimpanan dan pengambilan data jauh lebih sederhana
3. 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.

4. 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 di atas 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.

V. 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 dan sama-sama memiliki atribut untuk data yang akan disimpan didalamnya.
2. Memperbaharui model Relasi yang baru,
 - 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 Atributte Kelas yang merupakan bagian tersebut harus menjadi atribut data pada Kelas Induknya, Namun jika Relasi 1:1 ini mempunyai kedudukan yang sama penempatan atribut data dapat diletakkan pada Kelas yang manapun.
 - Untuk Relasi 1:n (*one to many*), kita harus menambahkan atribut data Kelas pada sisi 1 ke Kelas pada sisi n, namun jika Kelas pada

- sisi n ingin ditambahkan ke sisi 1 Kelas pada sisi n harus diubah dalam bentuk array agar dapat menyimpan lebih dari satu data.
- Untuk Relasi n-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 atribut 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 n-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.

Sebagai Contoh akan dilakukan Migrasi terhadap siklus penjualan sebagai berikut:

Dari Diagram RE untuk siklus penjualan seperti **Gambar 2**, dapat digambarkan diagram REA, seperti pada **Gambar 3** Dan selanjutnya Model Objek Sistem Penjualan dan Penerimaan Kas dapat digambarkan menjadi seperti **Gambar 4**

Sistem Penjualan seperti digambarkan di atas 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.

VI. KESIMPULAN DAN SARAN

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.

Berdasarkan karakteristiknya keunggulan pendekatan objek adalah dapat meningkatkan efisiensi aplikasi karena sifatnya yang dapat diwariskan (digunakan kembali untuk objek lainnya) khususnya untuk sistem yang besar dan kompleks, selain itu karena model yang dinamis model ini tepat untuk sistem yang real time. Selain itu dalam dunia pemrograman konsep objek ini telah digunakan secara luas terbukti dengan banyaknya bahasa pemrograman yang berbasis objek.

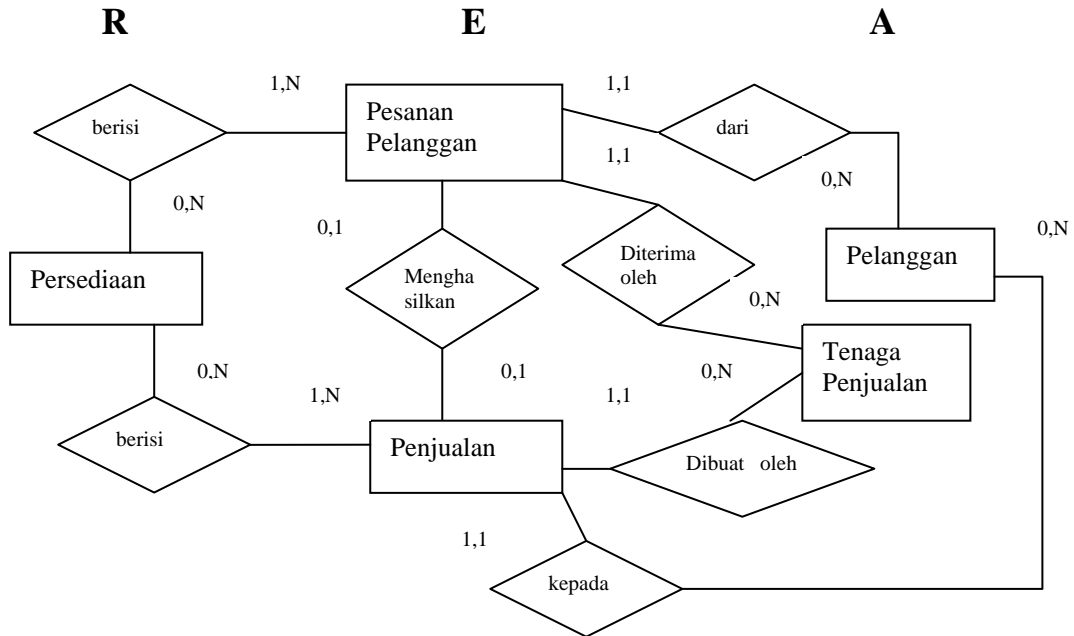
Keterbatasan pendekatan objek adalah kurangnya pengenalan mengenai pendekatan yang relatif baru dikalangan akuntansi, walaupun untuk kalangan IT hal ini tidaklah demikian. Pengenalan terhadap pengembangan sistem berorientasi objek tidak sepopuler pendekatan relasional. Untuk itu sebagai penutup tulisan ini penulis menyarankan untuk memperkenalkan pendekatan objek dalam perkuliahan yang berhubungan dengan sistem informasi akuntansi seperti misalnya untuk matakuliah analisis dan disain sistem dan juga sistem manajemen database untuk mahasiswa akuntansi.

DAFTAR RUJUKAN

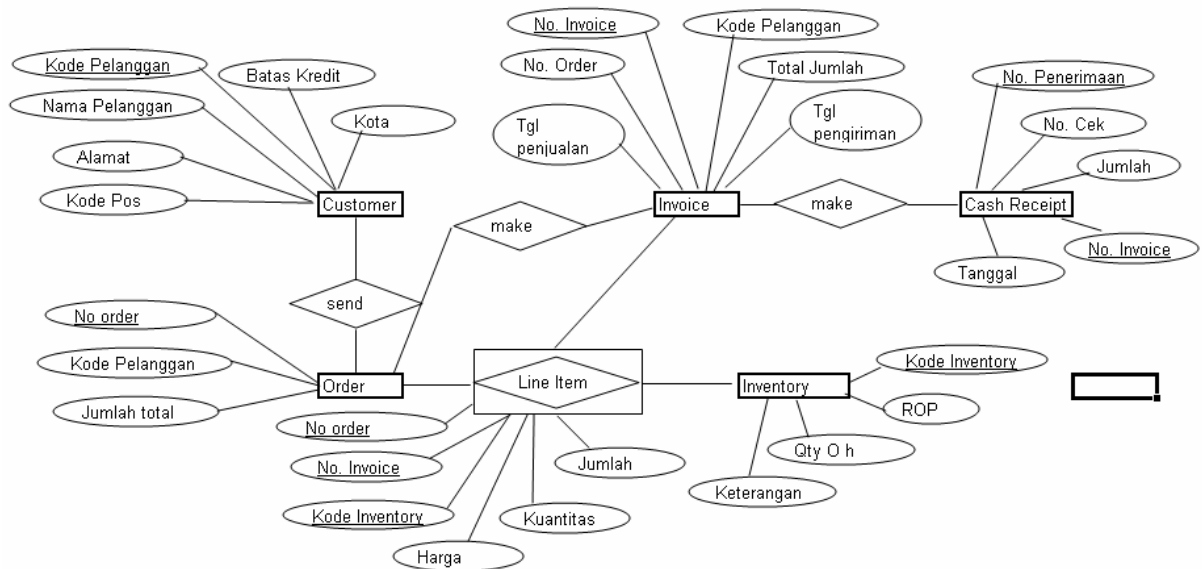
- Coad, Peter., Yourdon, Edward, *Object-Oriented Design*, Prentice Hall inc, International Edition, 1991.
- Harris, David, *System Analysis and Design: for the small enterprise*, Dryden Press, Orlando, FL32887-6777: Harcourt Brace College Publishers, 2nd edition, 1999.
- Irman, Heru. *Object-oriented Methodology: Sebuah Fenomena di Dunia Riset dan Aplikasi Teknologi*

www.gematel.com/Edisi28/ArtikelLepas/
 lepas3.html, 2001.
 McLeod, Jr, Raymond, and Schell, George,
Management Information Systems, Prentice Hall
 International, Inc, Upper Saddle River New
 Jersey 07458, 8th edition, 2001.

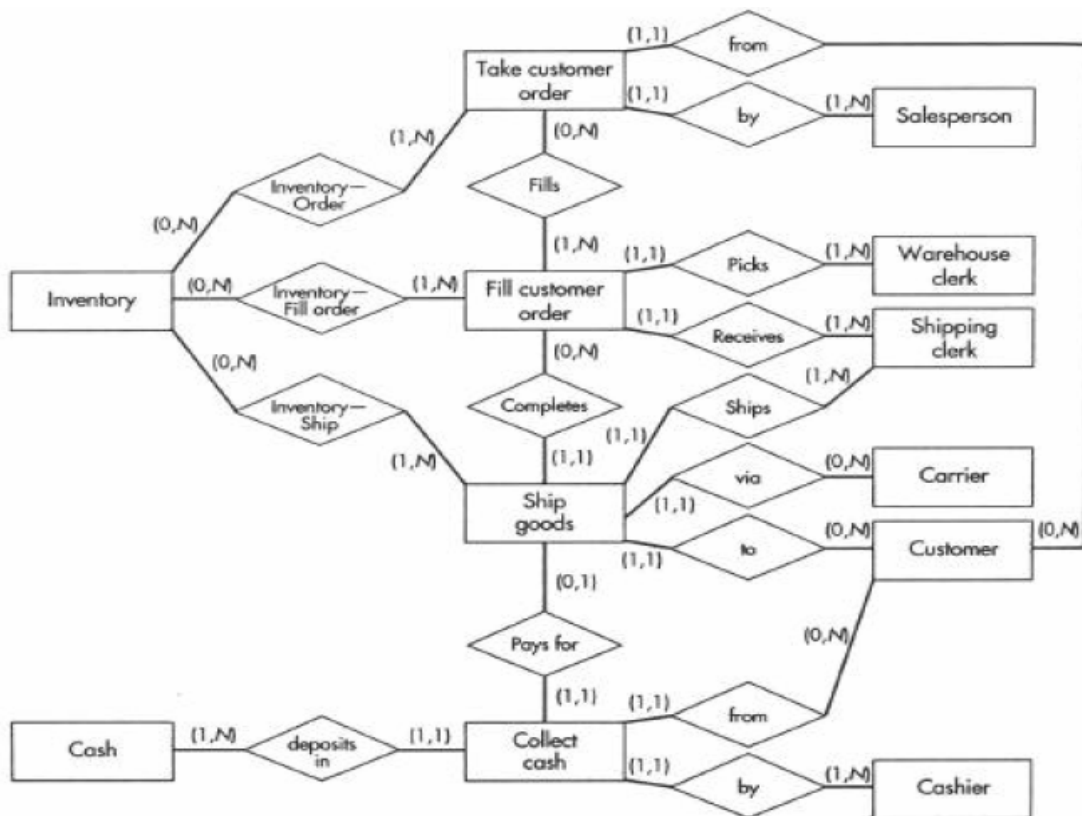
Whitten, Jeffrey L, and Bentley, Lonnie D, and
 Dittman, Kevin C, *Systems Analysis and Design
 Methods*, Irwin MC-GrawHill Companies, Inc,
 New York, 5th edition, 2001.



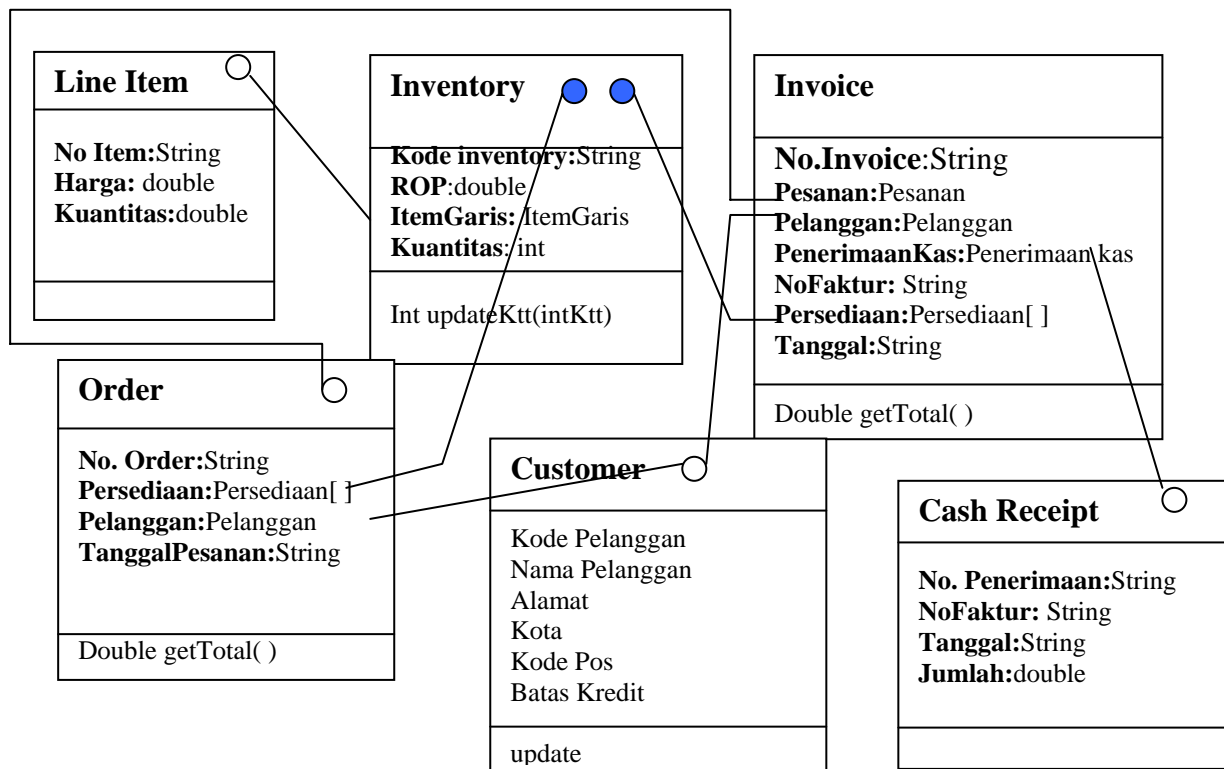
Gambar 1. Model RE menggunakan Diagram REA



Gambar 2. Diagram RE Siklus Penjualan



Gambar 3. Model REA Siklus Penjualan



Gambar 4. Sistem Penjualan Dengan Model Objek