

PERBANDINGAN PENGARUH TEKNIK PEMODELAN *ENTITY-RELATIONSHIP MODEL* DAN *RESOURCES-EVENTS-AGENTS MODEL* ATAS PERANCANGAN DATABASE PERUSAHAAN

Michael Iskandar

Fakultas Ekonomi Universitas Katolik Parahyangan

Abstract

Databases are an important part of most computer-based information systems. As such, the design of a database has also gained importance because a bad design will compromise data integrity as well as make it inefficient in its use of resources. This paper compares the effect of two data modeling approaches on the design of a database, specifically the Entity-Relationship Model (ER Model) and the Resources-Events-Agents Model (REA Model). First, the sales/collection cycle of a fictional case study is described. Then, an ER Model and a REA Model are independently created based on this case study. Finally, one database design is created based on the ER Model, and another database design is created based on the REA Model. The outcome of this experiment shows that both approaches produced similar database designs, but the design processes were different in their complexity, structurality, and completeness. A significant difference is that the ER Model closely followed those aspects that were already laid out by the case study, while the REA Model started out by building a most complete data model, over and above the one described in the case study.

Keywords: *database, data modeling, ER Model, REA Model*

1. Pendahuluan

Database merupakan bagian yang sangat penting dari sistem informasi berbasis komputer. Dari sekian banyak jenis sistem informasi yang sekarang dipergunakan dalam dunia usaha, hanya sebagian kecil yang tidak memiliki *database*, yaitu sistem informasi yang hanya dapat mengalirkan data atau informasi seperti halnya telepon, *intercom*, dan *facsimile*. Sistem-sistem lain umumnya mengharuskan agar data atau informasi dicatat terlebih dahulu dalam sebuah *database* sebelum diolah, dialirkan, ditampilkan di layar monitor atau dicetak melalui *printer*.

Terkait dengan peran *database* yang sangat penting bagi sistem informasi, maka perancangan *database* secara benar juga menjadi sangat penting. Tugas perancangan *database* itu menjadi lebih sulit lagi dikarenakan skala operasi perusahaan-perusahaan yang semakin besar sehingga *database*-nya pun menjadi semakin kompleks. Makalah ini membandingkan dua teknik pemodelan data, yaitu ER Model dan REA Model, untuk mengetahui pengaruhnya atas perancangan *database* perusahaan.

2. Usaha-usaha Pemodelan Data dan Perancangan Database

Dalam proses komputerisasi di perusahaan atau organisasi, hal penyimpanan data telah mengalami banyak perubahan dan perbaikan. Pendekatan pertama adalah yang disebut *file approach*, di mana data disimpan oleh komputer dalam satu atau beberapa *file*. Namun dalam penggunaannya, ternyata *file approach* menunjukkan sejumlah kelemahan, sehingga manusia mulai berpaling ke konsep *database* beserta *data model* yang melandasinya [O'Brien and Marakas, 2006:148-150].

Dua pendekatan awal yang diharapkan dapat menggantikan *file approach* adalah *hierarchical* dan *network data model*. *Hierarchical Data Model* dikembangkan oleh North American Rockwell bersama-sama dengan IBM pada tahun 1970-an. Pada waktu yang hampir bersamaan, *Network Data Model* dikembangkan oleh sebuah lembaga bernama Database Task Group (DBTG) yang didirikan oleh CODASYL (Conference on Data Systems Languages). Meskipun kedua pendekatan ini sudah lebih baik daripada *file approach*, namun keduanya masih memiliki sejumlah kelemahan yang berarti [Rob and Coronel, 2004:41].

Model *database* yang betul-betul revolusioner muncul dari hasil pemikiran E. F. Codd pada tahun 1970 dengan nama *Relational Model* [Rob and Coronel, 2004:41]. Model ini dikembangkan dari bidang matematika diskrit yaitu dari teori himpunan, khususnya relasi antar himpunan [Munir, 2003:81-85].

Database yang dikembangkan berdasarkan *relational data model* disebut *relational database*. Agar rancangan *database* ini memenuhi syarat keandalan dan efisiensi, maka proses perancangan *database* tersebut harus melalui sebuah proses yang disebut normalisasi data. Pada mulanya, normalisasi data dilakukan melalui tiga tahap berurutan yang menghasilkan bentuk normal *First Normal Form* (1NF), *Second Normal Form* (2NF), dan *Third Normal Form* (3NF). Belakangan, Boyce dan Codd mengusulkan sebuah langkah yang akan langsung menghasilkan bentuk setara dengan 3NF tanpa perlu melalui bentuk normal sebelumnya, disebut dengan istilah *Boyce-Codd Normal Form* (BCNF) [Mannino, 2007:223-229].

Kadang-kadang sebuah *database* bahkan membutuhkan bentuk normal yang lebih tinggi lagi, yaitu 4NF. Seperti halnya dalam pencapaian 3NF/BCNF, hal ini ternyata juga dapat dilakukan dengan cara bertahap atau dengan cara langsung. Untuk mencapai 4NF terdapat dua pendekatan, di mana yang satu merupakan pendekatan dengan langkah-langkah lebih banyak namun lebih mudah dilaksanakan, serta pendekatan yang lain yang lebih singkat namun membutuhkan kemampuan analisis lebih tinggi [Iskandar, 2003:87-99].

Di atas 4NF masih terdapat 5NF dan DKNF (*domain key normal form*) namun keduanya jarang dibutuhkan dalam pengembangan *database* di dunia nyata [Mannino, 2007:236].

Model Relasional yang diajukan Codd merupakan jawaban atas tuntutan perusahaan selama ini untuk memperoleh tempat penyimpanan data yang efisien dan dapat diandalkan. Namun demikian, karena skala operasi perusahaan-perusahaan semakin besar maka perancangan *data model*-nya ternyata menjadi semakin rumit. Sebuah pendekatan yang memudahkan perancangan *relational data model* muncul pada tahun 1976 ketika Peter Chen memperkenalkan *Entity-Relationship Model* (ER Model) beserta perkakas pendukungnya, yaitu *Entity-Relationship Diagram* (ERD) [Chen, 1976].

Rob dan Coronel mengobservasi bahwa berangkat dari *hierarchical model* dan *network model*, ke *relational model* dan akhirnya ke ER Model menunjukkan sifat semantik yang semakin tinggi [Rob and Coronel, 2004:53].

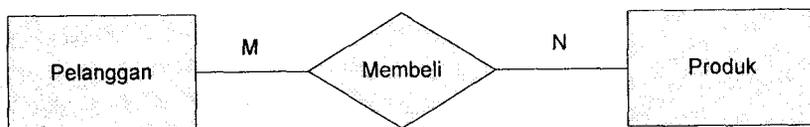
3. Pengertian *Entity-Relationship Model*

Entity-Relationship Model (ER Model) diperkenalkan oleh Chen sebagai pendekatan baru dalam memodelkan data. Chen mengidentifikasi bahwa semua model data terdiri dari dua hal besar, yaitu *entity set* dan *relationship set* [Chen, 1976].

Sebuah *entity* disebut Chen sebagai "*a thing which can be distinctly identified*" [Chen, 1976:10], misalnya saja seorang pelanggan, sebuah produk yang tersimpan di gudang perusahaan, atau sebuah transaksi penjualan. Dari sini, dapat dipahami bahwa *entity set* adalah himpunan semua pelanggan, himpunan semua produk di gudang, himpunan semua transaksi penjualan yang dilakukan perusahaan. Pada dasarnya, anggota-anggota sebuah *entity set* akan memiliki *properties* atau *attributes* yang sama, misalnya saja semua pelanggan pasti memiliki nama dan alamat, semua produk pasti memiliki nama produk dan harga jual, semua transaksi penjualan pasti memiliki tanggal penjualan dan nilai penjualan.

Sementara itu, sebuah *relationship* menurut Chen adalah "*an association between entities*" [Chen, 1976:10], sehingga *relationship set* berarti himpunan relasi antara sejumlah entitas. Relasi-relasi ini memiliki *cardinality*, yang diidentifikasi sebagai *one-to-one*, *one-to-many*, dan *many-to-many* [Mannino, 2007: 137-140].

Untuk memudahkan visualisasi ER Model, Chen juga memperkenalkan sebuah *Entity-Relationship Diagram* (ERD). Secara sederhana, setiap *entity set* diwakili oleh sebuah gambar persegi empat, sedangkan setiap *relationship set* diwakili oleh sebuah gambar belah ketupat. Hubungan antara setiap persegi empat dengan belah ketupat digambarkan menggunakan garis, di mana tercantum pula *cardinality* dari *relationship* yang digambarkan itu. Sebuah contoh sederhana dapat dilihat dalam Gambar 1.



Gambar 1. Contoh ERD Sederhana

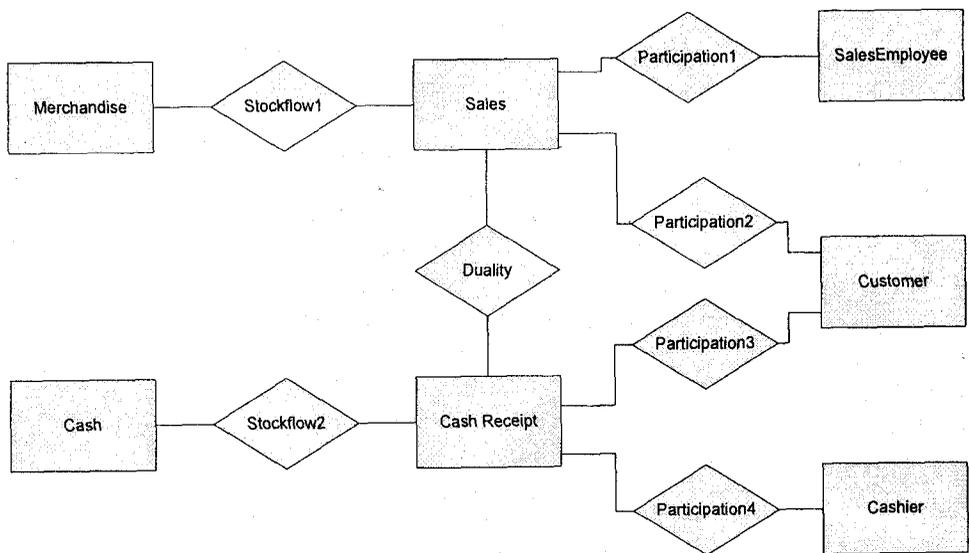
Pada Gambar 1 tampak bagaimana terdapat dua entitas, yaitu Pelanggan dan Produk, di mana relasi di antaranya adalah “membeli”. Karena ER Model merupakan model semantik, maka Gambar 1 dapat dibaca seolah-olah kalimat sederhana: “Pelanggan membeli Produk” atau “Produk dibeli oleh Pelanggan”. *Cardinality* yang tercantum di sana adalah M-N atau *many-to-many*, artinya seorang Pelanggan dapat membeli banyak Produk, dan sebaliknya setiap (jenis) Produk dapat dibeli oleh banyak Pelanggan.

4. Pengertian *Resources-Events-Agents Model*

Pada tahun 1982, William McCarthy memperkenalkan *Resources-Events-Agents (REA) Model*, yang beliau kembangkan setelah mempelajari pola dari berbagai sistem akuntansi perusahaan-perusahaan [Dunn, et al., 2005:24]. Motivasi dari pengembangan model ini adalah karena, menurut pendapat McCarthy, sistem akuntansi klasik yang dijelaskan oleh Luca Pacioli pada tahun 1494 dan masih banyak diterapkan sekarang, sebenarnya sudah tidak cocok lagi dikarenakan perubahan besar di akhir abad ke-20 dalam hal teknologi informasi di satu pihak, dan cara perusahaan beroperasi di lain pihak [Geerts and McCarthy, 1997].

Pemodelan REA diterapkan dalam empat tingkat, yakni: *value system level*, *value chain level*, *business process level*, dan *task level*. Pada *value system level*, model ditekankan untuk menunjukkan aliran *resources* antara perusahaan dengan pihak-pihak luar yang berhubungan dengan perusahaan, seperti misalnya: pelanggan, pemasok, pekerja, dan pemerintah. Sedangkan pada *value chain level*, pemodelan diarahkan untuk menunjukkan aliran *resources* tersebut di dalam perusahaan sendiri yaitu antar berbagai fungsi perusahaan seperti keuangan, penjualan, pembelian, dan produksi. Pada tingkat *business process level*, dilakukan pemodelan atas masing-masing proses bisnis atau fungsi tadi, misalnya saja proses penjualan atau proses pembelian. Akhirnya, pada *task level modeling*, kegiatan-kegiatan di dalam masing-masing proses bisnis digambarkan secara lebih rinci menggunakan perkakas seperti *data flow diagram* atau *systems flowchart* [Dunn, et al., 2005:24-30]. Dari keempat tingkat pemodelan REA tersebut, adalah *business process level* yang berguna dalam perancangan *database* perusahaan [Dunn, et al., 2005:121-150]. Oleh karena itu dalam makalah ini yang dibahas lebih lanjut hanya pemodelan pada tingkat ini.

McCarthy pertama-tama mengidentifikasi sebuah *REA Core Pattern*, di mana semua *entities* pasti terdiri dari *resources*, *events*, dan *agents*. Dari ketiga jenis entitas tersebut, yang pertama-tama harus diidentifikasi adalah *events*, dengan memperhatikan bahwa setiap *event* seharusnya memiliki pasangan *event* yang lain, di mana *event* yang satu akan menambah *resource* tertentu perusahaan (*increment event*), dan pada saat yang bersamaan *event* yang lain akan mengurangi *resource* lain perusahaan (*decrement event*). Sebagai contoh: *Sales event* dengan *Cash Receipt event*, di mana yang pertama akan mengurangi *Merchandise resource* perusahaan (*decrement event*) tetapi yang lain akan menambah *Cash resource* perusahaan (*increment event*). Untuk menotasikan sifat *increment-decrement* tadi, maka Dunn, et al., menyebut relasi generik antara kedua *events* tadi dengan sebutan *duality*. Sedangkan relasi generik antara masing-masing *event* dengan *resource* terkait disebut *stockflow*. Pemodelan REA juga melihat adanya pihak-pihak yang terkait dengan setiap *event*, yang disebut dengan istilah *agents*. Relasi antara *event* dengan *agents* secara umum disebut *participation*. *Agents* ini dapat berasal dari luar perusahaan (*external agents*) atau dari dalam perusahaan (*internal agents*). Misalnya, terkait dengan *Sales event* adalah *Customer* (*external agent*) dan *SalesEmployee* (*internal agent*). Sedangkan terkait dengan *Cash Receipt event* adalah *Customer* (*external agent*) dan *Cashier* (*internal agent*). [Dunn, et al., 2005:62-64]. *Core REA Pattern* tersebut dapat ditampilkan seperti dalam Gambar 2.



Gambar 2. Contoh *Core REA Pattern*

Meskipun relasi-relasi yang ditampilkan dalam *Core REA Pattern* pada umumnya menggunakan istilah generik yaitu *duality*, *stockflow*, serta *participation*, dalam kondisi-kondisi khusus istilah-istilah lain dapat dipergunakan. Hal ini dapat dilihat dalam Tabel 1.

Tabel 1. Istilah Relasi Generik dan Khusus Pada *Core REA Pattern*

Kondisi	Nama Relasi	Contoh
Antara: Event dengan Event		
Generik	Duality	Acquisition - Cash Disbursement
Khusus	Transformation	Material Issuance - Production Run
Khusus	Reversal	Sales - Sales Return
Antara: Event dengan Resource		
Generik	Stockflow	Acquisition - Inventory
Khusus	Use	Labor Operation - Labor
Khusus	Consume	BookLoan - LibraryBook
Khusus	Produce	Production - Finished Goods
Antara: Event dengan Agent		
Generik	Participation	Cash Disbursement - Cashier
Khusus	Authorization	Cash Disbursement - Manager
Antara: Agent dengan Agent		
Khusus	Association / Responsibility	Supervisor - Employee
Khusus	Assignment	Salesperson - Customer

Sumber: Dunn, et al., 2005: 63-80, 199-404

Dalam perkembangan selanjutnya, *Core REA Pattern* diperbaiki menjadi *Extended REA Pattern*, yaitu dengan memasukkan dua jenis *events* lagi: *commitment event* serta *instigation event*. Sebuah *commitment event* terjadi sebelum *event* biasa, yaitu saat transaksi atau pertukaran *resources* belum terjadi namun sudah terjadi komitmen untuk melakukan transaksi atau pertukaran itu. Sedangkan *instigation event* merupakan *event* yang terjadi lebih awal dari *commitment event*, yaitu *event* yang mengawali seluruh proses bisnis [Dunn, et al., 2005:82-86]. Contoh dari *commitment event* adalah Perjanjian Kredit, yang merupakan komitmen sebelum dapat terjadi *event* Pencairan Kredit. Adapun contoh dari *instigation event* adalah Pengajuan Kredit, yang terjadi sebelum *event* Perjanjian Kredit. Jenis-jenis relasi khusus yang ditemui dalam *Extended REA Pattern* ditampilkan dalam Tabel 2.

Tabel 2. Istilah Relasi Pada *Extended REA Pattern*

Antara	Dengan	Nama Relasi	Contoh
Commitment Event	Commitment Event	Reciprocal	Material Requisition - Production Order
Mutual Commitment Event	Noncash Economic Event	Fulfillment	Purchase Order - Purchase
Mutual Commitment Event	Noncash Resource	Reservation	Purchase Order - Inventory
Mutual Commitment Event	Cash Resource	Reservation	Purchase Order - Cash
Instigation Event	Mutual Commitment Event	Fulfillment	Purchase Requisition - Purchase Order
Instigation Event	Noncash Resource	Proposition	Purchase Requisition - Inventory
Resource	Resource	Linkage	Raw Material - Finished Goods
Resource	Resource Type	Typification	Product - Product Type
Agent	Agent Type	Typification	Customer - Customer Type
Agent	Resource	Custody	Warehouse Employee - Inventory

Sumber: Dunn, et al., 2005: 63-80, 199-404

5. Contoh Kasus Untuk Pemodelan Data

Guna mengetahui pengaruh dari pemodelan data atas perancangan *database* perusahaan, maka diperlukan sebuah contoh kasus. Perlu dicatat bahwa contoh kasus ini, meskipun fiktif, telah diusahakan serealistis mungkin. Misalnya, contoh kasus ini membahas *Sales/Collection Cycle* dalam sistem informasi akuntansi perusahaan, di mana secara sengaja siklus ini tidak merupakan sistem informasi akuntansi yang ideal. Pada paragraf-paragraf berikut dijelaskan mengenai contoh kasus yang dimaksud.

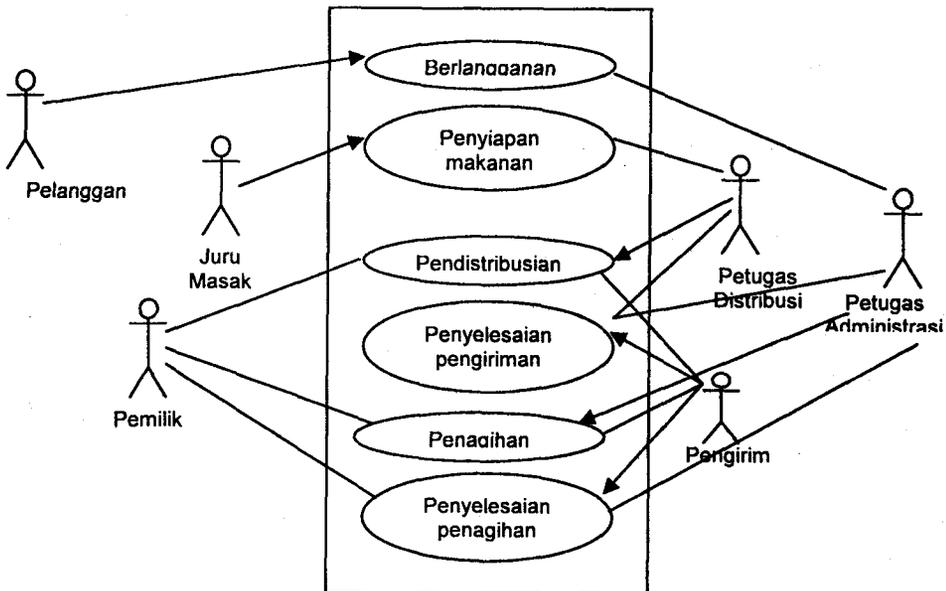
Rumah Makan "4 Sehat 5 Sempurna" merupakan sebuah rumah makan yang telah berdiri sejak tahun 2002 dan berlokasi di Bandung. Pemilik rumah makan ini menilai bahwa usahanya tidak terlampau berhasil, terbukti dari tingkat penjualan yang rendah sehingga laba yang dicapai setiap tahun pun hanya sedikit. Dalam usaha untuk mendorong peningkatan penjualan, maka pemilik rumah makan ini memutuskan untuk memulai usaha jasa berlangganan makanan (*catering* atau "rantangan"). Pemilik rumah makan menentukan bahwa karena baru dimulai, hanya akan diberlakukan satu tarif, yaitu Rp 5.000,- untuk satu porsi makanan. Satu kali pengiriman dapat mencakup lebih dari satu porsi makanan, tergantung permintaan pelanggan.

Target market dari usaha *catering* ini adalah para pelanggan yang memiliki pantangan tertentu dalam makanan, misalnya orang-orang yang mengidap penyakit diabetes, kadar kolesterol tinggi, dan lain-lain. Oleh karena itu, sistem informasi untuk usaha *catering* ini harus dapat diandalkan, karena kesalahan pengiriman makanan dapat berakibat fatal. Untuk meyakinkan hal ini, maka pemilik rumah makan memutuskan untuk menggunakan sistem informasi yang sudah berbasis komputer.

Masalah lain yang muncul dalam pelaksanaan usaha *catering* ini adalah bahwa ternyata tidak semua pelanggan minta dikirim makanan setiap hari. Memang, ada yang minta dikirim setiap hari (Senin s/d Minggu), tetapi ada juga yang hanya minta dikirim Senin s/d Jumat sedangkan hari Sabtu dan Minggu tidak dikirim karena pelanggan itu selalu ke luar kota pada akhir pekan. Ada pula yang minta dikirim hari Senin dan Kamis saja, atau ada pula yang minta dikirim hari Selasa, Kamis, dan Sabtu. Intinya, setiap pelanggan dapat minta pola pengirimannya masing-masing.

Pelanggan ditagih sekali seminggu atas pengiriman makanan selama pekan yang telah lewat. Namun karena pengiriman setiap pelanggan berbeda-beda, maka setiap hari mungkin saja terjadi penagihan. Sebagai contoh, seseorang yang hanya minta dikirim makanan setiap Senin dan Kamis saja, akan ditagih pada setiap hari Kamis; orang lain yang minta dikirim hari Selasa, Kamis, dan Sabtu, akan ditagih pada setiap hari Sabtu.

Gambar 3 adalah Use-Case Diagram dari sistem informasi yang dijalankan pada usaha *catering* ini.



Gambar 3. Use-Case Diagram Usaha *Catering*

Berikut adalah Use-Case Scenarios dari berbagai Use-Case yang telah ditampilkan dalam diagram di atas.

Use-Case: Berlangganan

1.	Pelanggan menelepon rumah makan dan menyatakan hendak berlangganan makanan.
2.	Petugas administrasi mencatat data diri pelanggan dalam Formulir Berlangganan
3.	Petugas administrasi mencatat pantangan-pantangan makanan bagi pelanggan tersebut dalam formulir yang sama.
4.	Petugas administrasi mencatat hari-hari apa saja (Senin, Selasa, dst.) pelanggan minta dikirim makanan dalam formulir yang sama.
5.	Petugas administrasi menginput data yang telah dicatatnya ke komputer.

Use-Case: Penyiapan makanan

1.	Setiap pagi, Juru Masak mencetak lembar Jumlah Kebutuhan Makanan, yaitu sebuah printout di mana tercantum jumlah porsi makanan untuk pasien diabetes, berapa porsi makanan untuk yang berkholesterol tinggi, berapa porsi makanan vegetarian, dan lain-lain, yang harus dikirimkan ke pelanggan-pelanggan pada hari itu.
2.	Juru Masak menyerahkan jumlah dan jenis makanan sesuai dengan lembar Jumlah Kebutuhan Makanan kepada Petugas Distribusi.
3.	Juru Masak menyimpan lembar Jumlah Kebutuhan Makanan dalam arsip.

Use-Case: Pendistribusian

1.	Petugas Distribusi mencetak Lembar Pengiriman untuk hari itu, di mana dalam lembar tersebut tercantum nama dan alamat pelanggan yang harus dikirim makanan pada hari itu, beserta pantangan-pantangan makanan dari masing-masing pelanggan.
2.	Petugas Distribusi mencetak label nama dan alamat pelanggan yang harus dikirim makanan pada hari itu. Penentuan pelanggan yang label namanya akan dicetak dilakukan otomatis oleh komputer, jadi Petugas Distribusi tidak perlu mengindikasikan satu per satu pelanggan mana yang perlu dicetak labelnya.
3.	Petugas Distribusi kemudian menggunakan informasi dalam Lembar Pengiriman untuk mengalokasikan berbagai jenis makanan yang telah diterima dari Juru Masak untuk masing-masing pelanggan. Setiap bungkus makanan yang telah dialokasikan ditempel label nama dan alamat yang sesuai.
4.	Pemilik rumah makan menerima Lembar Pengiriman dari Petugas Distribusi, kemudian menggunakannya untuk memeriksa kembali apakah pengalokasian makanan kepada masing-masing pelanggan sudah benar. Pengiriman tidak dapat dilakukan sebelum Pemilik memberikan persetujuannya.
5.	Apabila telah disetujui, maka Pemilik membubuhkan paraf pada lembar Pengiriman dan mengembalikannya kepada Petugas Distribusi.
6.	Petugas Distribusi memberikan bungkus-bungkus makanan yang telah diberi label kepada Pengirim untuk didistribusikan kepada Pelanggan.
7.	Petugas Distribusi menyimpan Lembar Pengiriman dalam arsip.

Use-Case: Penyelesaian pengiriman

1. Pengirim yang telah selesai mengirimkan makanan melapor kepada Petugas Distribusi.
2. Pengirim membubuhkan tanda tangannya pada Lembar Pengiriman, sebagai bukti tanggung jawabnya bahwa semua pengiriman telah dilaksanakannya dengan benar.
3. Petugas Distribusi menyerahkan Lembar Pengiriman itu kepada Petugas Administrasi.
4. Petugas Administrasi menginput data pengiriman itu ke komputer.
5. Petugas Administrasi menyimpan Lembar Pengiriman dalam arsip.

Use-Case: Penagihan

1. Setiap hari, petugas Administrasi mencetak Kuitansi Penagihan.
2. Petugas Administrasi juga mencetak Daftar Tagihan dalam dua rangkap. Daftar ini berisi nama dan alamat pelanggan-pelanggan yang hendak ditagih, beserta jumlah tagihannya masing-masing. Daftar Tagihan ini mencantumkan tagihan-tagihan yang baru akan ditagih (sesuai dengan Kuitansi yang baru dicetak), dan tagihan-tagihan yang belum tertagih dari minggu sebelumnya, serta akan ditagih pada hari itu.
3. Petugas Administrasi mengambil kuitansi-kuitansi yang belum tertagih dan tercantum dalam Daftar Tagihan dari arsip dan menyatukannya dengan kuitansi-kuitansi yang baru dicetak.
3. Rangkap pertama Daftar Tagihan diberikan kepada Pemilik.
4. Rangkap kedua Daftar Tagihan bersama-sama dengan kuitansi diberikan kepada Pengirim untuk ditagihkan bersama-sama dengan pengiriman makanan pada hari itu.

Use-Case: Penyelesaian penagihan

1. Pengirim yang selesai menagih, melapor kepada Pemilik.
2. Pengirim menyerahkan uang yang diterimanya, dan kuitansi-kuitansi yang tidak berhasil ditagih. (Catatan: pelanggan yang membayar tagihan menerima kuitansi yang sesuai dari Pengirim.)
3. Pemilik menghitung jumlah uang dan jumlah nilai kuitansi yang tidak berhasil ditagih. Angka yang diperolehnya dicocokkan (harus sama) dengan total Daftar Tagihan yang ada padanya.
4. Pemilik menyimpan uang yang berhasil ditagih,
5. Pemilik memberikan tanda "Lunas" pada nama-nama di Daftar Tagihan yang telah membayar.
6. Pemilik menyerahkan kuitansi yang tidak berhasil ditagih beserta Daftar Tagihan kepada Petugas Administrasi.
7. Petugas Administrasi mencatat tanda "Lunas" pada tagihan-tagihan yang ditampilkan di layar komputer. Petugas melakukan hal ini dengan cara mengikuti tanda yang telah dibuat oleh Pemilik di Daftar Tagihan.
8. Petugas Administrasi menyimpan kuitansi yang belum tertagih untuk ditagihkan pada kesempatan berikutnya.
9. Petugas Administrasi menyimpan Daftar Tagihan sebagai arsip.

6. Pemodelan Data Menggunakan ER Model

Berangkat dari studi kasus perusahaan "4 Sehat 5 Sempurna", penulis menyusun sebuah model data menggunakan ER Model. Hasil yang diperoleh dapat dilihat pada Gambar 4 dan Gambar 6. Gambar 4 menunjukkan ERD yang menampilkan semua entitas, relasi, dan *cardinality* yang teridentifikasi. Dengan pertimbangan kemudahan pembacaan ERD, maka *attributes* tidak ditampilkan di Gambar 4, melainkan dalam sebuah *ER Grammar Notation* (Gambar 6).

Penulis kemudian berusaha menyusun *database* perusahaan berdasarkan ERD yang telah diperoleh. Di sini penyusunan rancangan *database* dilakukan berlandaskan dua hal. Pertama, syarat-syarat normalisasi data harus terpenuhi. Kedua, rancangan *database* ini harus memperhatikan kebutuhan perusahaan. Misalnya, pada titik-titik di mana petugas perusahaan disebutkan bertanggung jawab atas tindakannya, maka NIK petugas itu akan muncul dalam tabel terkait. Hal ini terjadi pada petugas administrasi yang mencatat pendaftaran pelanggan, pengirim yang mengirimkan makanan, dan pengirim yang menyetor uang yang diperoleh dari pelanggan. Sedangkan misalnya Juru Masak, tidak disebutkan harus bertanggung jawab atas penyediaan makanan yang dilakukannya, maka NIK Juru Masak tidak dicatat dalam tabel ProduksiMakanan. Gambar 8 menunjukkan *database* yang diperoleh setelah melalui proses normalisasi data dan telah memperhatikan kebutuhan dan permintaan perusahaan.

7. Pemodelan Data Menggunakan REA Model

Setelah menyusun ERD dan rancangan *database* menggunakan ER Model, maka proses tersebut diulang menggunakan REA Model. Pertama-tama disusun sebuah REA Model yang hasilnya dapat dilihat dalam Gambar 5 (ERD) dan Gambar 7 (ER Grammar). Kemudian, dengan memperhatikan prinsip-prinsip normalisasi data dan kebutuhan serta permintaan perusahaan, diperoleh rancangan *database* seperti yang tampak dalam Gambar 9.

8. Temuan-temuan Saat Pemodelan Data

Hasil akhir dari percobaan penyusunan rancangan *database* bagi perusahaan "4 Sehat 5 Sempurna", baik menggunakan ER Model maupun REA Model, menunjukkan bahwa rancangan *database* yang diperoleh tidak berbeda. Jadi menggunakan ER Model ataupun REA Model akhirnya menghasilkan rancangan *database* yang sama saja.

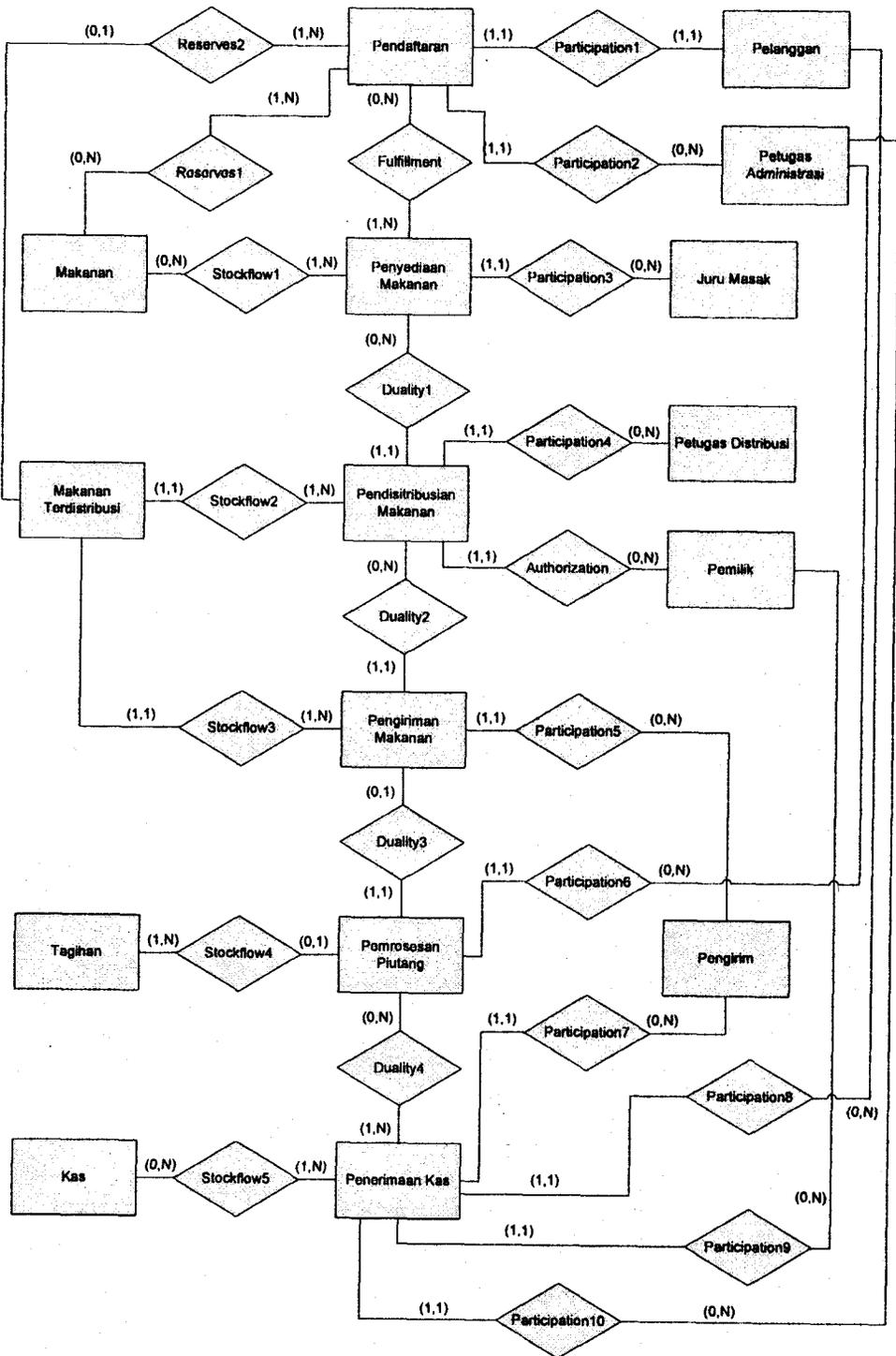
Namun demikian, dalam proses pemodelan data untuk perusahaan ini, ditemukan empat hal yang berbeda, yaitu:

- a. ER Model secara semantik lebih intuitif daripada REA Model. Misalnya pada ER Model, *Petugas Distribusi - Mendistribusi - Makanan*, sangat mudah untuk disusun ataupun dibaca oleh pengguna. Sedangkan dalam REA Model, alur yang sama adalah:

Petugas Distribusi - Participation4 - Pendistribusian Makanan - Stockflow2 - Makanan Terdistribusi, yang mana hal ini lebih non-intuitif untuk disusun ataupun dibaca.

- b. REA Model lebih terstruktur dalam penentuan entitas maupun relasinya, karena pada REA Model entitas sudah dikategorisasi menjadi *Resources*, *Events*, dan *Agents*; serta relasi-relasi sampai titik tertentu sudah distandarisasi, misalnya dengan menggunakan istilah *participation* dan *stockflow*. Sedangkan pada ER Model, karena cara penentuan entitas maupun relasi dibebaskan, maka menurut penulis justru lebih sulit untuk menentukan entitas dan relasi dengan tepat.
- c. Pada pemodelan yang telah dilakukan, ternyata ER Model menghasilkan 13 buah entitas dan 15 buah relasi, sedangkan REA Model menghasilkan 16 buah entitas dan 24 buah relasi. Dari sini dapat disimpulkan bahwa REA Model menghasilkan model yang lebih rumit dibandingkan ER Model, sehingga REA Model lebih banyak membutuhkan usaha dari orang yang menyusun model data yang bersangkutan.
- d. Pada ER Model, diasumsikan bahwa sistem informasi akuntansi yang modelnya hendak disusun, memang sudah dirancang sebagai sistem informasi akuntansi yang baik. Sedangkan pada REA Model, hal ini tidak diasumsikan, melainkan "dipaksakan" oleh REA Model. Dengan perkataan lain, ER Model dikembangkan berdasarkan sistem informasi yang ada di perusahaan dan tidak berusaha mengoreksi apa pun atas hal-hal yang hendak dicatat perusahaan, sedangkan REA Model memastikan bahwa semua langkah-langkah dan aspek-aspek dalam sebuah proses bisnis dibuat catatannya. Sebuah sistem informasi akuntansi diharapkan dapat mendukung operasi sehari-hari perusahaan, serta menghasilkan catatan yang cukup akurat, rinci dan lengkap, supaya manajemen perusahaan bisa memperoleh informasi yang dibutuhkannya untuk mengambil keputusan. Jadi REA Model lebih berguna untuk merancang *database* yang sesuai dengan kriteria sistem informasi akuntansi yang baik, sedangkan ER Model tidak.

Di lain pihak, apabila seluruh hal yang ditampilkan REA Model betul-betul dicatat maka sistem informasi yang dihasilkan akan sangat "birokratis", karena atas setiap langkah dalam proses bisnis harus dilakukan pencatatan. Untuk menghindari hal ini, maka pada saat merancang *database* menggunakan REA Model, perancang harus berani tegas dalam menghapus hal-hal yang dipandang tidak perlu dicatat.



Gambar 5. Pemodelan Data Menggunakan REA Model

Entity: Pelanggan Attributes: KodePlg Nama Alamat Identifier: KodePlg	Entity: Pemilik Attributes: ID Pemilik Identifier: ID Pemilik	Relationship: Mendistribusi Connected Entities: (1) Petugas Distribusi (M) Makanan (M) Pantangan (M) Pola Pengiriman
Entity: Petugas Admin Attributes: NIK Nama Alamat Identifier: NIK	Entity: Pengirim Attributes: NIK Nama Alamat Identifier: NIK	Relationship: Menjadi Connected Entities: (1) Makanan (M) Makanan Terdistribusi
Entity: Formulir Berlangganan Attributes: NoFormulir TglDaftar KodePlg NIK KodePantangan Identifier: NoFormulir	Entity: Pengiriman Attributes: NoKirim TglKirim KodePlg KodeMakananDist NilaiRp Identifier: NoKirim	Relationship: Menyetujui Connected Entities: (1) Pemilik (1) (M) Makanan Terdistribusi (M) Pantangan (M) Pola Pengiriman Attributes: TandaSetuju
Entity: Pantangan Attributes: KodePantangan NamaPantangan Identifier: KodePantangan	Entity: Tagihan Attributes: NoKuitansi Identifier: No Kuitansi	Relationship: Menerima Connected Entities: (1) Pengirim (1) (M) Makanan Terdistribusi
Entity: Pola Pengiriman Attributes: KodePlg PolaPengiriman Identifier: KodePlg	Relationship: Mendaftar Connected Entities: (1) Pelanggan (1) Formulir Berlangganan	Relationship: Melakukan Connected Entities: (1) Pengirim (M) Pengiriman
Entity: Juru Masak Attributes: NIK Nama Alamat Identifier: NIK	Relationship: Mengisi Connected Entities: (1) Petugas Admin (M) Formulir Berlangganan	Relationship: Penyampaian Connected Entities: (M) Pelanggan (1) Pengiriman
Entity: Petugas Distribusi Attributes: NIK Nama Alamat Identifier: NIK	Relationship: Mengandung1 Connected Entities: (1) Formulir Berlangganan (M) Pantangan	Relationship: Menghasilkan Connected Entities: (M) Pengiriman (1) Tagihan
Entity: Makanan Attributes: KodeMakanan NamaMakanan TglProduksi Qty KodePantangan Identifier: KodeMakanan TglProduksi	Relationship: Mengandung2 Connected Entities: (1) Formulir Berlangganan (1) Pola Pengiriman	Relationship: Menagih Connected Entities: (1) Pengirim (M) Tagihan
Entity: Makanan Terdistribusi Attributes: KodeMakananDist TglDistribusi KodePlg Identifier: KodeMakananDist TglDistribusi KodePlg	Relationship: Melihat Connected Entities: (1) Juru Masak (M) Pantangan (M) Pola Pengiriman	Relationship: Penyetoran Connected Entities: (M) Tagihan (1) Pemilik
	Relationship: Menyiapkan Connected Entities: (1) Juru Masak (M) Makanan	

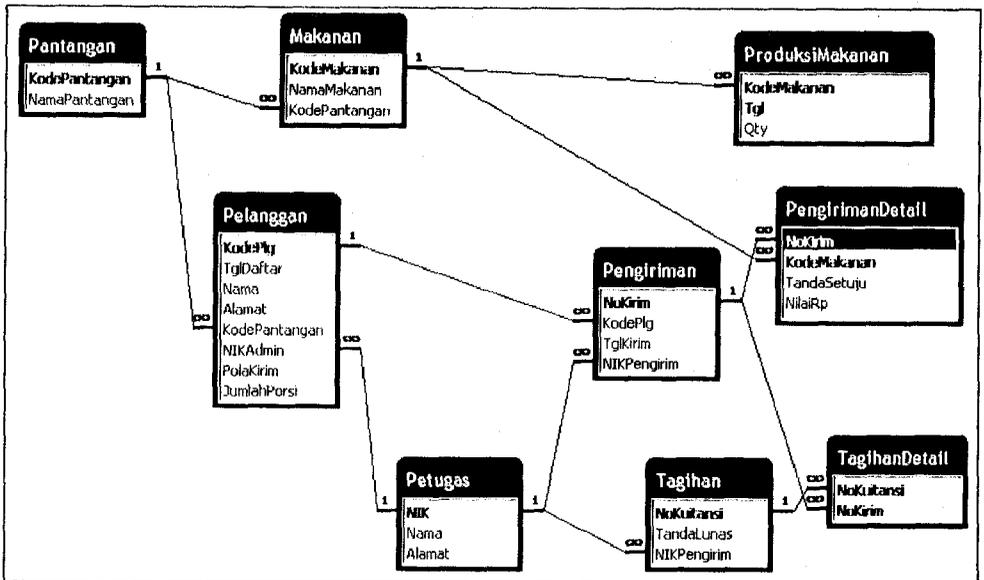
Gambar 6. ER Grammar Untuk Pemodelan Menggunakan ER Model

Entity: Makanan Attributes: KodeMakanan NamaMakanan KodePantangan Identifier: KodeMakanan	Entity: Petugas Admin Attributes: NIK Nama Alamat Identifier: NIK	Relationship: Stockflow5 Connected Entities: (0,N) Kas (1,N) Penerimaan Kas
Entity: Makanan Terdistribusi Attributes: KodeMakananDist NamaMakanan KodePantangan NoDaftar Identifier: KodeMakananDist NoDaftar	Entity: Juru Masak Attributes: NIK Nama Alamat Identifier: NIK	Relationship: Fulfillment Connected Entities: (0,N) Pendaftaran (1,N) Penyediaan Makanan
Entity: Tagihan Attributes: NoTagihan KodePlg TglKirim Nilai Identifier: NoTagihan	Entity: Petugas Distribusi Attributes: NIK Nama Alamat Identifier: NIK	Relationship: Duality1 Connected Entities: (0,N) Penyediaan Makanan (1,1) PendistribusianMakanan
Entity: Kas Attributes: Nilai Identifier: -	Entity: Pemilik Attributes: ID Pemilik Identifier: ID Pemilik	Relationship: Duality2 Connected Entities: (0,N) PendistribusianMakanan (1,1) Pengiriman Makanan
Entity: Pendaftaran Attributes: NoDaftar TglDaftar KodePantangan NamaPantangan PolaPengiriman Identifier: NoDaftar	Entity: Pengirim Attributes: NIK Nama Alamat Identifier: NIK	Relationship: Duality3 Connected Entities: (0,1) Pengiriman Makanan (1,1) Pemrosesan Piutang
Entity: Penyediaan Makanan Attributes: TglProduksi Identifier: TglProduksi	Relationship: Reserves2 Connected Entities: (1,N) Pendaftaran (0,1) MakananTerdistribusi	Relationship: Duality4 Connected Entities: (0,N) Pemrosesan Piutang (1,N) Penerimaan Kas
Entity: PendistribusianMakanan Attributes: TglDistribusi Identifier: TglDistribusi	Relationship: Reserves1 Connected Entities: (1,N) Pendaftaran (0,N) Makanan	Relationship: Participation1 Connected Entities: (1,1) Pendaftaran (1,1) Pelanggan
Entity: Pengiriman Makanan Attributes: TglKirim Identifier: TglKirim	Relationship: Stockflow1 Connected Entities: (0,N) Makanan (1,N) Penyediaan Makanan	Relationship: Participation2 Connected Entities: (1,1) Pendaftaran (0,N) Petugas Administrasi
Entity: Pemrosesan Piutang Attributes: TglProses Identifier: TglProses	Relationship: Stockflow2 Connected Entities: (1,1) MakananTerdistribusi (1,N) PendistribusianMakanan	Relationship: Participation3 Connected Entities: (1,1) Penyediaan Makanan (0,N) Juru Masak
Entity: Penerimaan Kas Attributes: TglTerima Nilai Identifier: TglTerima	Relationship: Stockflow3 Connected Entities: (1,1) Makanan Terdistribusi (1,M) Pengiriman Makanan	Relationship: Participation4 Connected Entities: (1,1) PendistribusianMakanan (0,N) Petugas Distribusi
Entity: Pelanggan Attributes: KodePlg Nama Alamat Identifier: KodePlg	Relationship: Stockflow4 Connected Entities: (1,N) Tagihan (0,1) Pemrosesan Piutang	Relationship: Authorization Connected Entities: (1,1) PendistribusianMakanan (0,N) Pemilik
		Relationship: Participation5 Connected Entities: (1,1) Pengiriman Makanan (0,N) Pengirim

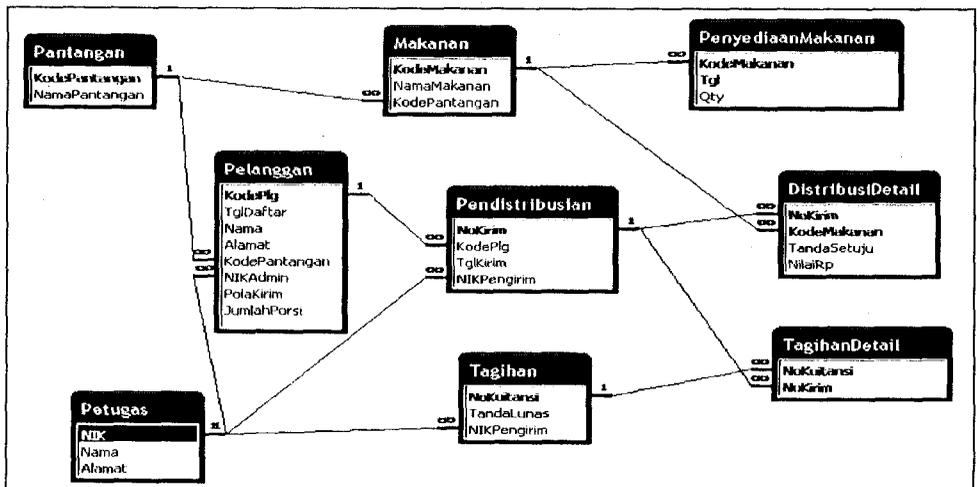
Gambar 7. ER Grammar Untuk Pemodelan Menggunakan REA Model

Relationship: Participation6 Connected Entities: (1,1) Pemrosesan Piutang (0,N) Petugas Administrasi	Relationship: Participation8 Connected Entities: (1,1) Penerimaan Kas (0,N) Pelanggan	Relationship: Participation10 Connected Entities: (1,1) Penerimaan Kas (0,N) Petugas Administrasi
Relationship: Participation7 Connected Entities: (0,N) Pengirim (1,1) Penerimaan Kas	Relationship: Participation9 Connected Entities: (1,1) Penerimaan Kas (0,N) Pemilik	

Gambar 7 (lanjutan)



Gambar 8. Rancangan Database Berdasarkan ER Model



Gambar 9. Rancangan Database Berdasarkan REA Model

9. Kesimpulan

Pada akhirnya dapat disimpulkan bahwa baik ER Model maupun REA Model akan menghasilkan rancangan *database* yang sama, namun proses perancangan tersebut berbeda dalam hal kompleksitasnya, sifat terstrukturanya, dan kelengkapannya. Ada pun pertimbangan yang harus dipergunakan dalam memilih teknik pemodelan data adalah sejauh mana perancang menghendaki teknik itu dapat menampung aspirasi dan kebutuhan yang sudah teridentifikasi dalam sistem informasi yang hendak dikembangkan. Dalam hal ini, perancang akan lebih tepat bila menggunakan ER Model. Namun jika teknik pemodelan data diharapkan dapat menganjurkan data apa saja yang mungkin dicatat oleh sistem informasi yang bersangkutan, maka REA Model yang lebih cocok untuk dipergunakan.

Daftar Pustaka

- Chen, P. P. (1976), "The Entity-Relationship Model -Toward A Unified View of Data", *ACM Transactions on Database Systems*, Vol. 1, No. 1, March 1976.
- Dunn, C. L.; Cherrington, J. O.; and Hollander, A. S. (2005), *Enterprise Information Systems: A Pattern-based Approach*, 3rd edition, McGraw-Hill/Irwin, Boston.
- Geerts, G. L., and McCarthy, W. E. (1997), "Modeling Business Enterprises as Value-Added Process Hierarchies with Resource-Events-Agent Object Templates", *Business Object Design and Implementation* (Sutherland and Patel, eds.), Springer-Verlag, London.
- Iskandar, M. (2003), "Perbandingan Antara Dua Pendekatan Normalisasi Data Hingga Kondisi Fourth Normal Form (4NF)", *Majalah Ilmiah Bina Ekonomi*, Vol. 7, No. 2, Agustus 2003.
- Mannino, M. V. (2007), *Database Design, Application, Development, & Administration*, 3rd edition, McGraw-Hill/Irwin, Boston.
- Munir, R. (2003), *Matematika Diskrit*, edisi ke-2, Penerbit Informatika, Bandung.
- O' Brien, J. A., and Marakas, G.M. (2006), *Management Information Systems*, 7th edition, McGraw-Hill/Irwin, Boston.
- Rob, P. and Coronel, C. (2004), *Database Systems: Design, Implementation, & Management*, 6th edition, Thomson Course Technology, Boston.