

RANCANGAN DATABASE SUBSISTEM PRODUKSI DENGAN PENDEKATAN *SEMANTIC OBJECT MODEL*

Oviliani Yenty Yuliana

Dosen Fakultas Ekonomi, Jurusan Akuntansi – Universitas Kristen Petra

ABSTRAK

Dalam persaingan di pasar bebas, para pelaku bisnis di bidang industri dalam membuat suatu keputusan yang tepat memerlukan informasi secara cepat dan akurat. Sistem akuntansi biaya tradisional tidak dapat menyediakan informasi yang memadai, sehingga banyak perusahaan industri yang beralih ke sistem *Activity-Based Costing (ABC)*. Tetapi, sistem ABC merupakan sistem yang kompleks dan memerlukan banyak data yang harus disimpan dan diolah, sehingga harus menggunakan teknologi informasi dan *database*. Kemajuan di bidang perangkat lunak mengakibatkan pembuatan aplikasi program bukan masalah lagi. Permasalahan utama adalah bagaimana merancang *database*, agar dapat menyajikan informasi secara cepat dan akurat. Untuk itu, dalam makalah ini dibahas pemodelan *database* dengan pendekatan *semantic object model*. Model data ini lebih mudah digunakan dan menghasilkan transformasi yang lebih normal, jika dibandingkan dengan *entity relationship model* yang umum digunakan.

Kata kunci: Sub Sistem Produksi, *Semantic Object Model*, *Database Relational*.

ABSTRACT

To compete in the global market, business performer who active in industry fields should have and get information quickly and accurately, so they could make the precise decision. Traditional cost accounting system cannot give sufficient information, so many industries shift to Activity-Based Costing system (ABC). ABC system is more complex and need more data that should be save and process, so it should be applied information technology and database than traditional cost accounting system. The development of the software technology recently makes the construction of application program is not problem again. The primary problem is how to design database that presented information quickly and accurately. For that reason it necessary to make the model first. This paper discusses database modelling with semantic object model approach. This model is easier to use and is generate more normal database design than entity relationship model approach.

Keywords: *Production Sub System, Semantic Object Model, Relational Database.*

1. PENDAHULUAN

Menjelang pasar bebas, persaingan antar perusahaan industri semakin ketat. Untuk dapat bersaing dan bertahan hidup, para pelaku bisnis perlu menetapkan harga jual yang tepat. Dalam rangka meningkatkan penjualan, pelaku bisnis tidak boleh menjual produk jadi dibawah harga pokok produksi, yang dapat mengakibatkan kerugian. Sebaliknya pelaku bisnis juga tidak bisa menetapkan harga jual yang terlalu tinggi, sehingga dapat memberi peluang kepada pesaing untuk menguasai pasar. Pelaku bisnis yang memiliki

informasi biaya yang akurat, dapat menetapkan harga jual yang dapat bersaing dan tetap memperoleh keuntungan.

Selain itu, informasi biaya dibutuhkan oleh manajemen untuk memutuskan bauran produk dan mengevaluasi kinerja. Informasi biaya juga digunakan untuk membuat laporan keuangan yang dibutuhkan oleh pihak eksternal. Sistem akuntansi biaya tradisional dirancang untuk membuat laporan keuangan dan manajemen produk, tetapi tidak dapat menyediakan informasi yang memadai untuk menangani produksi di lingkungan pabrik modern.

Saat ini, banyak perusahaan industri beralih ke sistem ABC, karena sistem ABC memberikan beberapa keuntungan antara lain: (1) informasi biaya lebih akurat, sehingga dapat dibuat keputusan mengenai bauran produk dan harga jual yang lebih tepat, (2) informasi biaya lebih detil, sehingga dapat meningkatkan kemampuan manajemen untuk mengontrol dan mengendalikan total biaya, (3) informasi biaya dapat digunakan untuk merancang produk baru yang lebih murah dengan tetap mempertahankan mutu.

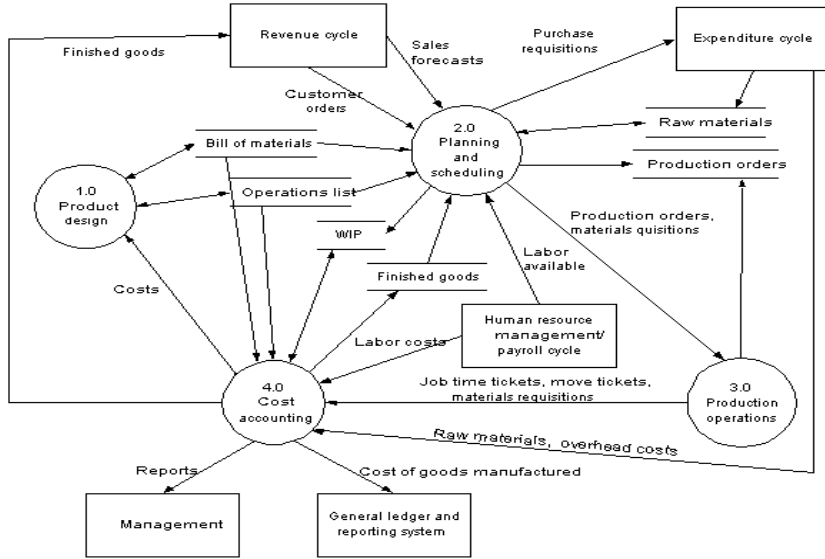
Namun sistem ABC lebih kompleks jika dibandingkan dengan sistem akuntansi biaya tradisional. Pada sistem ABC lebih banyak metode yang digunakan untuk mengalokasikan *overhead* pabrik dan mencatat semua aktivitas produksi yang menimbulkan biaya maupun yang tidak menimbulkan biaya secara lebih detil. Untuk menunjang sistem ABC dibutuhkan penggunaan teknologi informasi.

Perkembangan teknologi informasi, khususnya dibidang perangkat lunak dapat memudahkan *programmer* membuat program aplikasi database. Bahkan saat ini *non programmer* dimungkinkan untuk membuat aplikasi database sendiri, sehingga hal yang penting dalam pengembangan aplikasi database adalah pemodelan data saja. Jika model data tidak menggambarkan data pemakai secara benar, maka pemakai akan menemukan aplikasi yang sulit digunakan, tidak lengkap, dan sulit untuk dikembangkan.

Pemodelan data yang umum digunakan adalah *Entity Relationship (E-R) Model*. Pada model E-R, perancang kesulitan dalam menentukan *entity*, *relationship*, dan *attribute*. *Relational* database yang terbentuk dari model E-R sering masih belum dalam keadaan "normal", sehingga masih dimungkinkan terjadi pengulangan data dan anomali. Pada makalah ini akan dibahas pemodelan data yang lain, yaitu *Semantic Object Model*, karena pemodelan ini lebih mendekati pada pemahaman data pemakai.

2. AKTIVITAS SUBSISTEM PRODUKSI

Subsistem produksi berkaitan dengan aktivitas produksi suatu perusahaan. Umumnya subsistem produksi terbagi atas empat aktivitas berikut: *product design*, *planning and scheduling*, *production operations*, dan *cost accounting*. Aktivitas dan aliran informasi antara aktivitas atau dengan sub sistem lain dapat dilihat pada Gambar 1. Informasi yang mengalir pada sub sistem produksi digunakan untuk menghasilkan keputusan, antara lain: berapa banyak dan kapan suatu produk harus diproduksi, metode apa yang harus digunakan, bagaimana cara mengalokasikan biaya, dan apakah investasi harus dilakukan.



Gambar 1. Level 0 Data Flow Diagram Subsistem Produksi

2.1 Product Design

Aktivitas pertama subsistem produksi adalah *product design* (lingkaran 1.0 pada Gambar 1). Tujuan dari aktivitas ini adalah untuk merancang produk yang dapat memenuhi kebutuhan pelanggan dalam hal: kualitas, kehandalan, dan kegunaan dengan biaya produksi seminimal mungkin.

Dalam aktivitas ini diciptakan 2 dokumen, yaitu: *Bill Of Materials* (BOM) dan *Operation List* (OL). Dalam BOM dicantumkan kode, nama, dan jumlah bahan baku yang digunakan untuk menghasilkan satu produk, dokumen BOM dapat dilihat pada Gambar 2. Sedangkan OL berperan juga sebagai *routing sheet*. OL mencantumkan langkah-langkah yang harus dikerjakan, mesin yang dibutuhkan, dan waktu yang diperlukan pada setiap langkah. Dokumen OL dapat dilihat pada Gambar 3.

BILL OF MATERIAL				
Product No.: ❶ ❸	Rejected Percentage: ❶	Description: ❸	Authorization: ❶	Effective Date: ❶
4430	0.3%	Cabinet Side Panel	PDQ	01/01/02
Material No.	Material Description		Quantity	
115 ❷ ❹	Calculator Unit ❹		❷ 2	
135	Lower Casing		2	
198	Screw 3 inch		16	
178	Battery		2	
136	Upper Casing		2	
199	Screw 5 inch		12	

Keterangan:

❶ → Object BOM ❷ → Group Attribute Multi Value BOM Detail ❸ → Object PRODUCT ❹ → Object RAW MATERIAL

Gambar 2. Contoh Dokumen *Bill Of Material*

2.2 Planning and Scheduling

Aktivitas kedua subsistem produksi adalah *planning and scheduling* (lingkaran 2.0 pada Gambar 1). Tujuan dari aktivitas ini adalah untuk merencanakan produksi seefisien mungkin dan dapat memenuhi pesanan pelanggan serta dapat mengantisipasi kebutuhan pelanggan jangka pendek tanpa menimbulkan kelebihan persediaan barang jadi. Untuk keperluan tersebut perlu dibuat *Master Production Schedule (MPS)*.

OPERATION LIST				
Product No.:	Description:	Authorization:	Effective Date:	
4430 ❶ ❸	Cabinet Side Panel ❸	PDQ ❶	01/01/02 ❶	
Work Station Number	Product Operation No.	Operation Description	Standard Time (Hrs)	
			Set Up	Operation
MH25 ❷ ❹	100 ❷ ❺	Transfer from stock ❺	❷	❷ 1.00
ML15-12	105	Cut to shape	0.15	1.30
ML15-9	106	Corner cut		1.30
S28-17	124	Turn & shape	0.30	3.30
F54-5	142	Finish		3.00
P89-1	155	Paint		11.00
QC94	194	Inspect		2.00
MH25	101	Transfer to assembly		1.75

Keterangan:

❶ → Object OPERATION LIST ❷ → Group Attribute Multi Value Operation List Detail
 ❸ → Object PRODUCT ❹ → Object MACHINE ❺ → Object PRODUCT OPERATION

Gambar 3. Contoh Dokumen Operation List

MPS menunjukkan berapa banyak produk yang harus diproduksi. MPS dibuat berdasarkan informasi pesanan pelanggan, ramalan penjualan, dan jumlah persediaan barang jadi. MPS harus bisa mengikuti perubahan kondisi pasar, untuk itu diperlukan sistem informasi produksi yang dapat menyajikan informasi yang tepat dan akurat, sehingga proses produksi dapat berjalan lancar. Berdasarkan BOM dan MPS dapat ditentukan kapan bahan baku harus dibeli. Usulan pembelian bahan baku ditulis pada *purchase requisition* dan diserahkan ke bagian pembelian.

Dokumen lain yang dihasilkan pada aktivitas *planning and scheduling* adalah *production order* dan *material requisition*. Dokumen *production order* digunakan untuk mengotorisasi pembuatan sejumlah produk. Dokumen ini mendaftarkan operasi-operasi yang harus dilakukan, jumlah yang harus diproduksi, dan lokasi produk jadi harus diserahkan. Dokumen *production order* dapat dilihat pada Gambar 4. Sedangkan dokumen *material requisition* digunakan untuk mengotorisasi pemindahan sejumlah bahan baku yang dibutuhkan dari gudang ke lokasi pabrik, dapat dilihat pada Gambar 5. Dokumen ini berisi nomor *production order*, tanggal dokumen diterbitkan, nama dan jumlah bahan baku yang diperlukan.

PRODUCTION ORDER						No.: 4587 ①	
Order No.: 2289 ①	Product No.: 4430 ① ③	Description Cabinet Side Panel ③			Production Quantity: ① 1000		
Approved by: PJS ①	Release Date: 02/24/2002 ①	Issue Date: 02/25/2002 ①	Completion Date: 03/09/2002 ①		Deliver to: Assembly ①		
Work Station No.	Product Operation No.	Quantity	Operation Description	Start Date & Time		Finish Date & Time	
MH25 ④	100 ② ④	② 1003	Transfer from stock	02/28 ②	0700 ②	02/28 ②	0800 ②
ML15-12	105	1003	Cut to shape ④	02/28	0800	02/28	1011
ML15-9	106	1002	Corner cut	02/28	1030	02/28	1200
S28-17	124	1002	Turn & shape	02/28	1300	02/28	1700
F54-5	142	1001	Finish	03/01	0800	03/01	1100
P89-1	155	1001	Paint	03/01	1300	03/02	1300
QC94	194	1001	Inspect	03/02	1400	03/02	1600
MH25	101	1000	Transfer to assembly	03/02	1600	03/02	1700

Keterangan:

① → Object PRODUCTION ORDER ② → Group Attribute Multi Value Production Order Detail ③ → Object BOM ④ → Group Attribute Multi Value Operation List Detail

Gambar 4. Contoh Dokumen Production Order

MATERIAL REQUISITION				
Issued To: Assembly ①	Issue Date: 02/25/2002 ①	Issue No.: 2345 ①	Production Order No.: 4587 ① ③	
Material No.	Material Description	Quantity	Unit Cost	Total Cost
115 ② ④	Calculator Unit ④	② 2006	② 2.95	5917.70
135	Lower Casing	2006	0.45	902.70
198	Screw 3 inch	16048	0.02	320.96
178	Battery	2006	0.75	1504.50
136	Upper Casing	2006	0.80	1604.80
199	Screw 5 inch	12036	0.02	240.72
Issued by: AKL ①				<u>10491.38</u>
Received by: GWS ①		Costed by: ZBD ①		

Keterangan:

① → Object MATERIAL REQUISITION ② → Group Attribute Multi Value Material Requisition Detail ③ → Object PRODUCTION ORDER ④ → Object RAW MATERIAL

Gambar 5. Contoh Dokumen Material Requisition

2.3 Production Operations

Aktivitas ketiga subsistem produksi adalah proses produk yang sesungguhnya (lingkaran 3.0 pada Gambar 1). Pemindahan bahan baku atau produk jadi dalam proses produksi didokumentasi pada *move ticket*, yang dapat dilihat pada Gambar 6. Dokumen tersebut mencatat bahan baku yang ditransfer, lokasi tujuan, dan tanggal pemindahan. Sedangkan tenaga kerja yang berhubungan langsung dengan proses produksi dicatat dalam *job time ticket*, dapat dilihat pada Gambar 7.

MOVE TICKET			
Production Order Number:	4587 ❶ ❷	Move Ticket Number:	8753 ❶
From:	Assembly ❶	To: ❶	Finishing
Date Transferred: ❶	03/09/2002	Date Received: ❶	03/09/2002
Quantity Transferred:	1000 ❶	Quantity Received: ❶	1000
Transferred by:	KLS ❶	Received by: ❶	NRC
Comments:			

Keterangan: ❶ → *Object* MOVE TICKET ❷ → *Object* PRODUCTION ORDER

Gambar 6. Contoh *Move Ticket*

JOB TIME TICKET			
Employee No.:	Employee Name:	Production Order:	Date:
1368 ❶ ❷	Bill Smarts ❷	4587 ❶ ❸	02/25/2002 ❶
Work Center:	Operation No.:	Operation Description:	
ML 15-12 ❶	105 ❶ ❹	Cut to shape ❹	
Time Started:	Time Finished:	Elapsed Time Hours:	
8:30 ❶	12:00 ❶	3.50	
Pieces Finished:	Hourly Rate:	Labor Cost:	
1 ❶	10.00 ❷	35.00	
Approved:			

Keterangan:

❶ → *Object* JOB TIME TICKET ❷ → *Object* EMPLOYEE ❸ → *Object* PRODUCTION ORDER ❹ → *Object* OPERATION LIST

Gambar 7. Contoh *Job Time Ticket*

2.4 Cost Accounting

Aktivitas terakhir subsistem produksi adalah *cost accounting* (lingkaran 4.0 pada Gambar 1). Dua tujuan utama sistem akuntansi biaya adalah menyediakan (1) informasi untuk perencanaan, pengawasan, pengevaluasian terhadap aktivitas produksi, dan (2) data biaya produksi yang akurat yang dapat digunakan dalam menetapkan harga dan bauran produk. Sebagai tambahan, sistem akuntansi biaya menyediakan informasi yang digunakan untuk menghitung persediaan dan harga pokok penjualan yang ditunjukkan dalam laporan keuangan.

3. SEMANTIC OBJECT MODEL

3.1 Istilah yang Digunakan dalam *Semantic Object Model*

Semantic object model adalah model data yang digunakan untuk mempelajari dan mendokumentasi kebutuhan data pemakai secara grafik. Model data ini dibentuk berdasarkan hasil wawancara dan analisa terhadap: *report*, *form*, dan *query* sistem yang sedang berjalan. *Database* yang saat ini paling banyak digunakan adalah *relational database*. Untuk itu model data tersebut ditransformasikan ke dalam rancangan *relational database*.

Semantic object, disingkat menjadi *object*, adalah nama dari kumpulan *attribute* yang dapat menjelaskan gambaran identitas. *Semantic Object* digrupkan ke dalam *class*.

Object class mempunyai nama yang dapat dibedakan dari *class* lain, berhubungan dengan nama yang digambarkan, contoh: EMPLOYEE. Nama *object class* ditulis menggunakan huruf besar. Sedangkan datanya *object* disebut dengan *object instance*, contoh: "Bill".

Object memiliki sekumpulan *attribute*. Setiap *attribute* menunjukkan karakteristik dari identitas yang digambarkan. Sebagai contoh, objek EMPLOYEE memiliki *attribute* seperti: EmployeeNo, EmployeeName, HourlyRate. Hanya *attribute* yang dibutuhkan oleh pemakai yang disimpan. Terdapat tiga tipe *attribute*: *simple*, *group*, dan *object* (lihat *object* MOVE TICKET pada Gambar 8.I. Pertama, *Simple attribute* menyimpan elemen data tunggal, misal: MoveTicketNo dan Comments. Kedua, *Group attribute* berisi beberapa *simple attribute* yang berhubungan. Contoh *group attribute*: *Transferred*, terbentuk dari *simple attribute*: *From*, *Date*, *Quantity*, dan *By*. *Simple* dan *group attribute* disebut dengan *nonobject attribute*. Terakhir, *object attribute* adalah *attribute* yang berupa nama *object*. Nama lain *object attribute* adalah *object link attribute*, karena *attribute* ini berperan sebagai penghubung dengan *object* lain. *Object link* harus berpasangan (*pair attribute*), contoh pada *object* MOVE TICKET ada *object attribute* PRODUCTION ORDER, sebaliknya dalam *object* PRODUCTION ORDER ada *object attribute* MOVE TICKET lihat Gambar 8.H. dan 8.I.).

Setiap *attribute* dalam *object* memiliki minimum dan maksimum *attribute cardinality*. *Minimum attribute cardinality* untuk menunjukkan jumlah data minimal yang harus ada pada *object instance* agar valid, nilainya 0 atau 1. *Maximum attribute cardinality* untuk menunjukkan jumlah data maksimum dalam *object instance*, nilainya 1 atau N. *Attribute* yang *maximum cardinality*-nya satu disebut dengan *single-value attribute*. Sedangkan *attribute* yang *maximum cardinality*-nya lebih dari satu disebut dengan *multi-value attribute*.

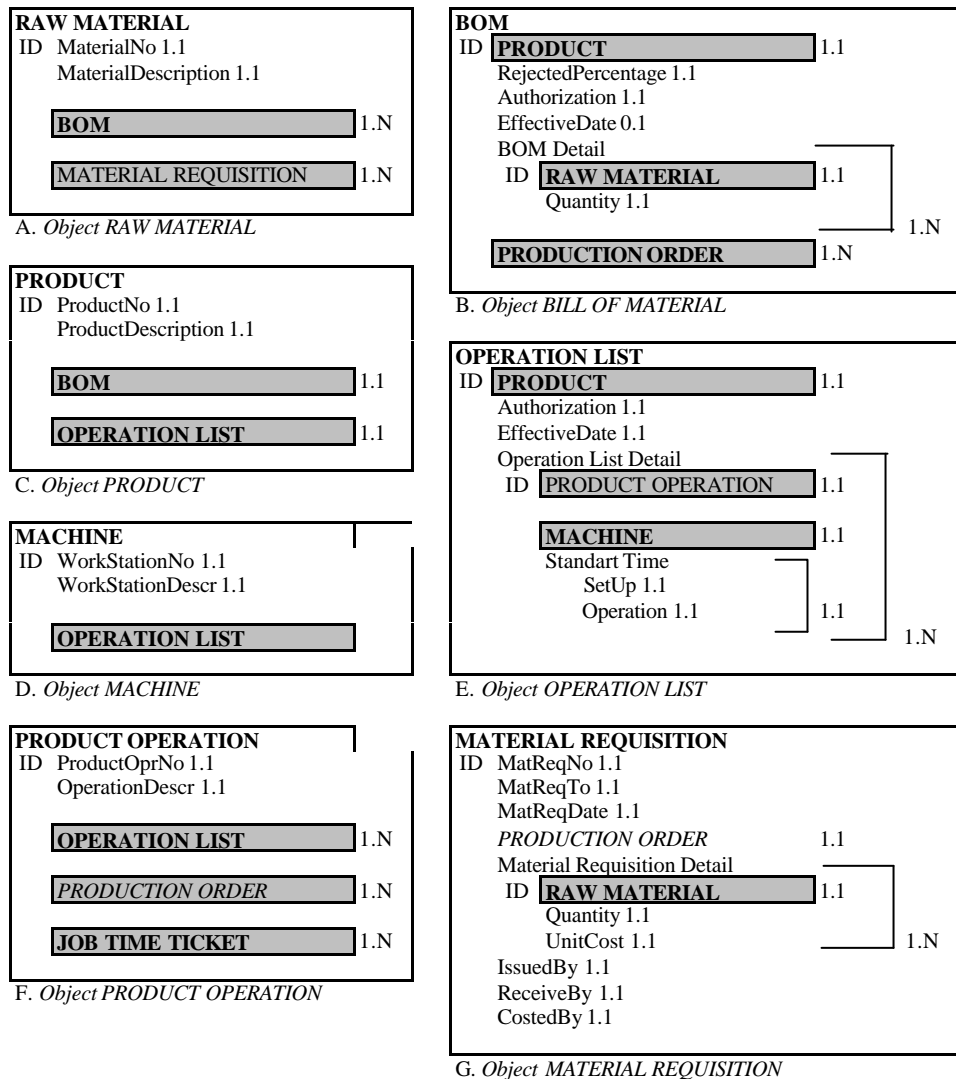
Cara penulisan *cardinality* adalah Minimum.Maksimum atau N.M. Contoh *single-value attribute* dapat dilihat pada *object* MOVE TICKET di Gambar 8.I. *Attribute cardinality* MoveTicketNo adalah 1.1, artinya dalam dokumen Move Ticket harus ada dan maksimum satu nomor. Contoh lain, *attribute* comments *cardinality*-nya adalah 0.1, artinya dalam dokumen Move Ticket, comments tidak harus diisi dan jika ada, maksimum satu comments. Contoh *multi-value attribute* dapat dilihat pada *object* PRODUCTION ORDER di Gambar 8.H. *Attribute cardinality* Production OrderDetail adalah 1.N, artinya dalam satu dokumen Production Order minimal harus ada satu dan maksimum N baris Production OrderDetail. Dalam contoh, dokumen ini berisi maksimum 8 baris, lihat Gambar 4, maka *attribute cardinality* dapat ditulis 1.8. Sedangkan *attribute cardinality* untuk *pair attribute* ditentukan dari *maximum cardinality* dari masing-masing *object attribute* yang dihubungkan. Contoh pada *object* MOVE TICKET *attribute cardinality* PRODUCTION ORDER adalah 1.1, *maximum cardinality* adalah 1. Sedangkan *attribute cardinality* MOVE TICKET pada *object* PRODUCTION ORDER adalah 1.N, *maximum cardinality* adalah N. Sehingga *attribute cardinality* untuk *pair attribute* tersebut adalah 1.N, artinya satu nomor dokumen Production Order dapat ditulis pada beberapa dokumen Move Ticket.

Object identifier adalah suatu *attribute* yang digunakan untuk menunjukkan suatu *object instance*, disimbolkan dengan ID atau ID. Simbol ID di awal *attribute* digunakan untuk menunjukkan *object identifier* yang unik, misal: *attribute* EmployeeNo pada Gambar 8.J. dalam *object* EMPLOYEE. Sedangkan simbol ID digunakan untuk menunjukkan *object identifier* yang tidak unik, misal EmployeeName dalam *object* EMPLOYEE. Sedangkan jika beberapa *attribute* digunakan untuk mengenali *object*

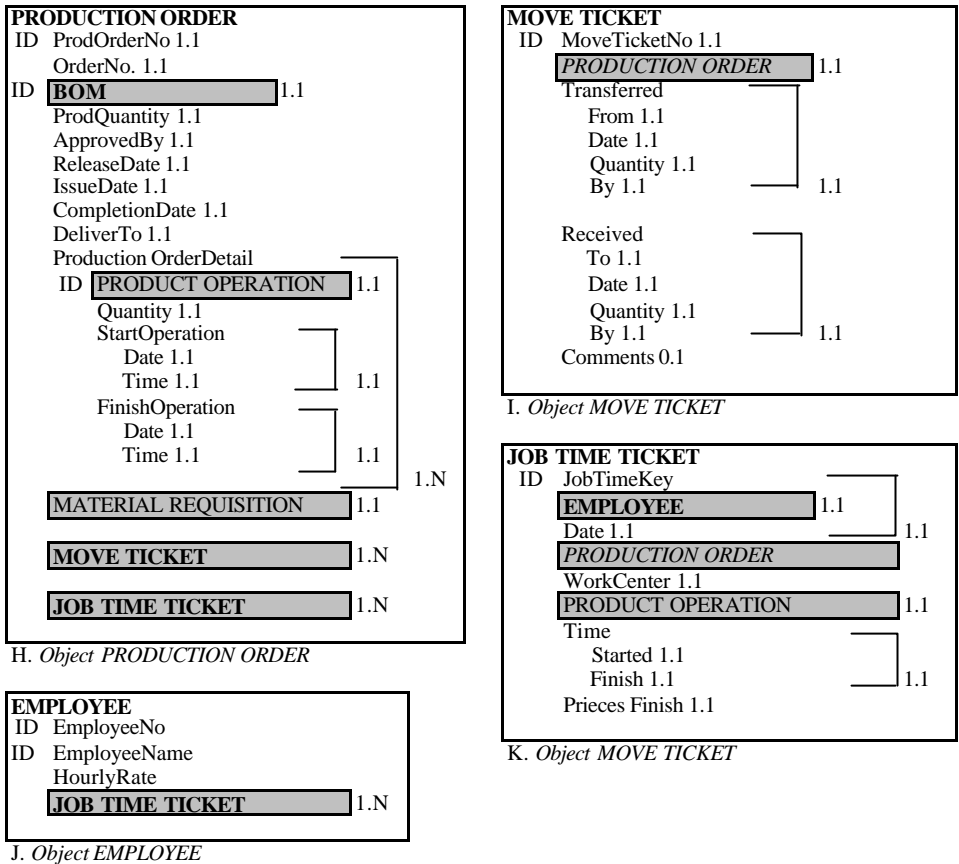
instance tertentu disebut dengan *group identifier*, contoh: *attribute* Job Time Key pada *object* JOB TIME TICKET pada Gambar 8.K.

3.2 Rancangan Semantic Object Model

Dokumen yang digunakan dalam sub sistem produksi dianalisa *object class*-nya. Item yang se-*object class* dalam satu dokumen diberi nomor yang sama. Satu item dapat diberi nomor lebih dari satu, jika item tersebut berasal dari *object class* lain, misal: item Product No. Product No berperan sebagai *attribute* dari *object* BOM dan PRODUCT, untuk itu Product No dijadikan *object attribute*. Penomorannya dapat dilihat pada Gambar 2. Sedangkan gambar objectnya dapat dilihat pada Gambar 8.B. dan 8.C. Rancangan *semantic object model* secara keseluruhan dapat dilihat sebagai berikut:



Gambar 8. Rancangan Semantic Object Model Subsistem Produksi



Gambar 8. Rancangan Semantic Object Model Subsistem Produksi (lanjutan)

4. RANCANGAN RELATIONAL DATABASE SUBSISTEM PRODUKSI

Database relational paling banyak digunakan saat ini. Untuk itu tindakan selanjutnya adalah mentransformasikan rancangan semantic object model ke rancangan relational database, dengan langkah-langkahnya sebagai berikut:

- A. Setiap object ditransformasikan ke satu relation. Nama object dijadikan nama relation.
- B. Setiap single-value attribute pada object yang bersangkutan dijadikan attribute pada relation yang dibentuk. Nama single-value attribute dijadikan nama attribute pada relation. Setiap object identifier (yang diawali dengan ID) pada object yang bersangkutan dijadikan key attribute (diberi garis bawah) pada relation yang dibentuk. Contoh untuk langkah A dan B, object EMPLOYEE pada Gambar 8.I. ditransformasikan menjadi relation EMPLOYEE pada Gambar 9.M.
- C. Setiap multi-value simple maupun group attribute ditransformasikan ke satu relation. Relation diberi nama berdasarkan nama multi-value attribute yang bersangkutan. Key attribute pada relation yang dibentuk dari multi-value simple attribute adalah object identifier dari object yang bersangkutan. Sedangkan key attribute pada relation yang dibentuk dari multi-value group attribute adalah object dan group identifier dari object yang bersangkutan. Contoh multi value group attribute BOM Detail pada object BOM

di Gambar 8.B. ditransformasikan ke *relation* BOM DETAIL seperti yang ditunjukkan pada Gambar 9.C.

- D. Cara mentransformasikan *object attribute* tergantung pada *attribute cardinality* untuk *pair attribute* yang bersangkutan, dengan ketentuan jika:
1. *relationship*nya 1:1, maka salah satu *object identifier* dari *object* yang bersangkutan dijadikan *foreign key* pada *relation* lain yang dibentuk. Umumnya *object identifier* dari *object* yang lebih dulu terbentuk, yang akan dijadikan *foreign key* pada *relation* lainnya. Contoh *pair attribute* BOM dan PRODUCT di Gambar 8.B. dan 8.C. *Object instance* product lebih dulu terbentuk, maka *object identifier* Product No dijadikan *foreign key* pada *relation* BOM.
 2. *relationship*nya 1:N, maka *object identifier* dari *pair attribute* yang maksimum *attribute cardinality*-nya satu dijadikan *foreign key* pada *relation* lain yang dibentuk dari *pair attribute* yang maksimum *attribute cardinality*-nya N. Contoh *pair attribute* BOM dan RAW MATERIAL di Gambar 8.A. dan 8.B. Maksimum *attribute cardinality* Object RAW MATERIAL adalah satu, sedangkan maksimum *attribute cardinality* Object BOM adalah N, maka *object identifier* RAW MATERIAL dijadikan *foreign key* pada *relation* BOM DETAIL.
 3. *relationship*nya N:M, maka dibentuk *intersection relation* yang berisi *object identifier* dari *object link* yang bersangkutan.

Hasil transformasi rancangan *semantic object model* ke rancangan *relational database* secara lengkap dapat dilihat pada Gambar 9. *Relation* pada Gambar 9 sudah memenuhi aturan normalisasi, artinya *attribute non key* sudah tergantung secara penuh kepada *key attribute*, kecuali *relation* PRODUCTION ORDER. Hal tersebut tidak akan terjadi jika dokumen pada sub sistem pendapatan juga sudah dibuat *semantic object model*-nya.

<p>A. PRODUCT (<u>Product No</u>, Product Description, Stock On Hand)</p> <p>B. BOM (<u>Product No</u>, Authorization, Effective Date, Rejected Percentage)</p> <p>C. BOM DETAIL (<u>Product No</u>, <u>Material No</u>, Quantity)</p> <p>D. RAW MATERIAL (<u>Material No</u>, Material Description, Quantity On Hand)</p> <p>E. OPERATION LIST (<u>Product No</u>, Authorization, Effective Date)</p> <p>F. OPERATION LIST DETAIL (<u>Product No</u>, <u>Product Opr No</u>, <u>Workstation No</u>, Standard Time Set Up, Standard Time Operation)</p> <p>G. PRODUCT OPERATION (<u>Product Opr No</u>, Operation Description)</p> <p>H. MACHINE (<u>Workstation No</u>, Work Station Descr)</p> <p>I. PRODUCTION ORDER (<u>Prod Order No</u>, Order No, <u>Product No</u>, Prod Quantity, Approved By, Release Date, Issue Date, Completion Date, Deliver To)</p> <p>J. PRODUCT ORDER DETAIL (<u>Prod Order No</u>, <u>Product Opr No</u>, Quantity, Start Operation Date, Start Operation Time, Finish Operation Date, Finish Operation Time)</p> <p>K. MATERIAL REQUISITION (<u>Mat Req To</u>, Mat Req To, Mat Req Date, <u>Prod Order No</u>)</p> <p>L. MATERIAL REQUISITION DETAIL (<u>Mat Req To</u>, <u>Material No</u>, Quantity, Unit Cost, Issued By, Receive By, Costed By)</p> <p>M. EMPLOYEE (<u>Employee No</u>, Employee Name, Hourly Rate)</p> <p>N. JOB TIME TICKET (<u>Employee No</u>, <u>Date</u>, <u>Prod Order No</u>, Work Center, <u>Product Opr No</u>, Time Started, Time Finist, Prieeces Finish)</p> <p>O. MOVE TICKET (<u>Move Ticket No</u>, <u>Prod Order No</u>, From, Date Transferred, Quantity Transferred, TransferredBy, To, DateReceived, QuantityReceived, ReceivedBy, Comments)</p>
--

Gambar 9. Rancangan Database Subsistem Produksi

5. KESIMPULAN

Dari hasil pembahasan dapat ditarik kesimpulan, *semantic object model* adalah model data yang lebih mendekati pada pemahaman data pemakai. Dimana model data terbentuk dari hasil wawancara dan analisa terhadap semua dokumen-dokumen yang digunakan dalam suatu aktifitas, sehingga struktur data yang terbentuk bisa lengkap. *Database relation* hasil transformasi dari *semantic object model* kebanyakan sudah memenuhi aturan normal, karena hubungan antara dan arti data sudah dipikirkan sejak model data tersebut dibentuk.

DAFTAR PUSTAKA

- Kroenke, D.M., 2000. *Database Processing: Fundamentals, Design & Implementation*. Seventh Ed., New Jersey: Prentice-Hall, Inc.
- McLeod, R., 1998. *Management Inform ation Systems*. Seventh Ed., New Jersey: Prentice-Hall, Inc.
- Romney, M.B., and P.J. Steinbart, 2000. *Accounting Inform ation Systems*. Eighth Ed., New Jersey: Prentice-Hall, Inc.
- Yuliana, O.Y., 2001. "Pendekatan Model REA dalam Perancangan Database Sistem Informasi Akuntansi Siklus Pendapatan", *Jurnal Akuntansi dan Keuangan*, Vol. 3, No. 1, Mei 2001, hal. 67–87.
- Yuliana, O.Y., 2001. "Implementasi Referential Integrity Constraint pada Microsoft Access dalam Upaya Memelihara Konsistensi Data", *Jurnal Informatika*, Vol. 2, No.1, Mei 2001, hal. 33–43.