

# Strategi Pemetaan dari Model Fungsional ke Model Object Oriented

*by* Tedy Setiadi

---

**Submission date:** 29-Aug-2020 07:28PM (UTC+0700)

**Submission ID:** 1375945328

**File name:** tegi\_Pemetaan\_dari\_Model\_Fungsional\_ke\_Model\_Object\_Oriented.pdf (328.99K)

**Word count:** 2790

**Character count:** 17776

## Strategi Pemetaan Dari Model Fungsional Ke Model Object Oriented

Tedy Setiadi

Program Studi Teknik Informatika  
Fakultas Teknologi Industri, Universitas Ahmad Dahlan  
Jin. Prof. Dr. Soepomo, S.H. Yogyakarta  
Telp. (0274) 378418, Fax. 381523  
email : tedz68@yahoo.com

### Abstract

*Because of being transition or because of choice, many software development environments make use both of the Junction-oriented and object-oriented approaches in their software development process. In some cases, object-oriented and jimention-oriented approaches are used in development in the same system, such as when using finction-oriented analysis with object-oriented design, necessitating a transition or mapping from one model to the other. This paper reviews the issues involved in mapping from a jimention-oriented software model to an object-oriented software model is coneluded that mapping strategy is feasible and reasonably automatable.*

*Keywords mapping, finction-oriented, object-oriented, automatable*

### I. Pendahuluan

Secara tradisional, langkah awal pengembangan perangkat lunak mencakup penentuan akurat kebutuhan apa yang diinginkan pengguna untuk pengembangan system. Proses ditandai dengan dekomposisi fungsi, yang meliputi identifikasi kebutuhan fungsional secara global kemudian didekomposisi menjadi kebutuhan yang lebih detail. Kebutuhan tersebut diekspresikan dengan model fungsional yang berfokus pada suatu diagram aliran data (DFD). Dokumen DFI adalah suatu himpunan proses dan aliran data yang dimiliki, sementara kamus data mendefinisikan semua item data.

Bagaimanapun juga, revolusi object oriented telah mengambil alih dalam rekayasa perangkat lunak. Kenyataan dan keuntungan yang didapat dari pendekatan object oriented disebabkan pengembang perangkat lunak mengubah ke metode baru meninggalkan pendekatan fungsional. Mereka harus menggunakan total konsep baru seperti objek, kelas, inheritance (pewarisan) dan polimorphism. Hal ini, tentu saja membutuhkan investasi yang substansial dalam pelatihan dan pengajarannya..

Pada beberapa lingkungan pengembang, berdasarkan pengalamannya mengubah dari pendekatan model fungsional ke model objek adalah suatu hal yang sulit. Seperti beberapa lingkungan mempunyai system yang berjalan dengan verifikasi model fungsional atau pengembang perangkat lunak dengan pengalaman yang luas serta sukses menggunakan model fungsional, atau dalam proses analisis lebih baik dengan metode fungsional pada lingkungan ini, ada kebutuhan untuk mendukung orientasi fungsional ke orientasi objek dari untuk memudahkan transisi secara konsisten dari model fungsional ke model objek.

Masalah yang muncul dari evolusi model fungsional ke pengembangan perangkat lunak objek oriented adalah apakah ada cara membuat transisi dari model fungsional ke rmodel objek yang lebih halus? Apakah pengalaman disain dengan model fungsional bermanfaat sewaktu melakukan analisis dan disain model objek? Apakah fisibel membuat otomatisasi strategi pemetaan dari model fungsional ke model objek?

## 2. Relevansi dan justifikasi untuk Strategi Pemetaan

Pendekatan berorientasi objek tidak berarti tanpa masalah dan kebijakan menggantikan pendekatan tradisional dengan pendekatan baru kadangkala menimbulkan pertanyaan. Alan Davis (1993) menunjukkan tidak ada cara untuk memastikan "black-box behaviour" atau fungsionalitas sistem dengan menguji suatu kelompok interaksi objek. Shumate (1991) mengklaim bahwa metode fungsional adalah metode terbaik untuk mengidentifikasi kebutuhan dan suatu konsep pendekatan pengembangan kombinasi fungsional dan berorientasi objek memberikan keseimbangan data dan fungsi dari system. Dan juga transformasi dari analisis fungsional ke disain objek tidak akan buruk sewaktu analisis dan disain melayani seluruh fungsi berbeda.

Menurut Chen (1994) pandangan disainer berbeda dengan pandangan pemakai. Pada berorientasi objek, isu disain awal adalah apa yang dilakukan objek dan bukan apa servis yang disediakan system. Disisi lain, pemakai memperhatikan dengan apa servis yang disediakan system: Jadi, ada kebutuhan yang harus dijembatani dari kesenjangan pandangan disainer dan bentuk Observasi lain, kebanyakan upaya teknik berorientasi objek menggabungkan beberapa model fungsional. Ada model OIVIT Rumbaugh dan pendekatan Shlaer-Mellor (1992) yang digunakan untuk didain system waktu nyata. Ini menunjukkan bahwa teknik model berorientasi objek sekarang menggabungkan dekomposisi fungsi pada beberapa hal.

Strategi pemetaan dari berorientasi fungsi ke model berorientasi objek membutuhkan petunjuk. Domain pengetahuan menjadi isu kritis dan oleh karena itu aturan pembuatan keputusan dari pengembang adalah penting. Suatu tool otomatis seperti strategi pemetaan tidak dapat disubstitusi untuk pengembang. Malahan, tool akan sebagai suatu bantuan yang memungkinkan pemakai untuk membuat keputusan lebih mudah dengan meringkas informasi dari model fungsional dan dengan membatasi pilihan perancang berorientasi objek apa yang seharusnya dibuat. memungkinkan

Fokus pada penelitian ini adalah untuk rncapai lingkungan terpadu yang mempunyai tool berbeda yang bekerja sama, memakai sumber daya bersarna. Pandang suatu skenario, dimana model berorientasi objek dan model berorientasi fungsi adalah dapat menggunakan CASE tools berbeda. pada lingkungan sama. Agar terjadi integrasi, hal yang mungkin untuk menghubungkan informasi pada dua model berbeda, yaitu dengan pemetaan atau menetapkan korespodensi antara satu model ke model lainnya.

## 3. Representasi Model

Ada berbagai representasi model yang ada pada orientasi fungsi rnaupun objek. Agar fokus pada suatu strategi pemetaan, maka model representasi diidentifikasi. Representasi diadopsi dan dijustifikasi mengapa mereka dipilih.

### 3.1. Representasi Model Berorientasi Fungsi

Model berorientasi fungsi adalah sinonim dengan model aliran data yang diusulkan DeMare0(1978). *Data Flow diagram* (DFD) merupakan bagian utama dari representasi fungsional untuk mendeskripsikan aspek data pada system, DFD dikembangkan dengan *entity relationship diagram* (ERD) yang diusulkan Chen (1.976). Dua tipe diagram, berkaitan dengan kamus data yang merupakan representasi berorientasi fungsi pada penelitian ini.

DFD yang digunakan adalah diturunkan dari model DeMarco. Proses kontrol dan alirannyai tidak dipertimbangkan karena konsep kontrol tidak esensial pada representasi berorientasi fungsi dan sering diasumsikan signifikan hanya pada domain khusus. Satu hal penting yang berbeda dari DFD standar adalah hirarki DFD tidak dipertimbangkan. Malahan *flattened data fiow diagram* (FDFD) digunakan untuk merepresentasikan system. Artinya, level terendah dari abstraksi proses ditunjukkan pada diagram, yaitu pada level terendah dekomposisi. Konsep FDFD diusulkan oleh

Shurnate dan Keller (1992) yang memboichkan proses dimulai dengan DFD tunggal berisi semua kebutuhan perangkat lunak dan membolehkan seluruh produk "dipelajari dan pikirkan hanya sekali". Lebih lanjut, jika FDFD diikuti, tidak ada informasi yang hilang dan didapat keuntungan yang jelas dari strategi pemetaan yaitu hanya dilakukan pada proses-proses pada level terendah.

### 3.2. Representasi Model Berorientasi Objek

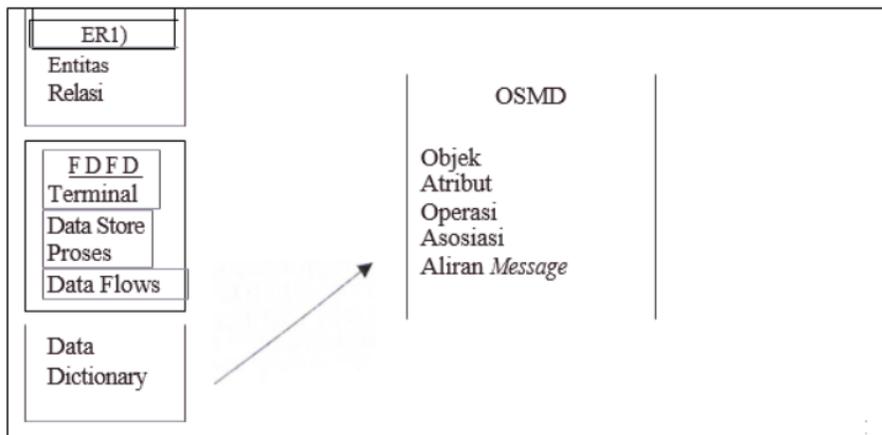
Pada umumnya, representasi berorientasi objek suatu system adalah dibuat dalam dua jenis yang merefleksikan dua pandangan berbeda terhadap system. Yang pertama disebut dengan semantik statis yaitu objek yang mencakup atribut, operasi dan interaksi antar objek. Jenis lain digunakan untuk merepresentasikan aspek dinamis dari system yaitu interaksi *runtime* antar objek. Ada aneka ragam diagram untuk rmemperlihatkan interaksi objek dan aliran *message* pada objek. Pada *object modeling technique* (OMT) dari Rumbaugh, model objek digunakan untuk menyatakan semantik statis dimana diagram penelusuran kejadian digunakan untuk menunjukkan *message* yang menyatakan pertukaran antar objek.

Model berorientasi objek direpresentasikan dengan sebuah diagram tunggal yang menyatakan model semantik statis maupun dinamis. Hal ini dicapai dengan menggabungkan notasi *message*, diadopsi dari aliran kejadian pada model OMT dengan model basis objek juga diadopsi dari OMT. Dalam hal ini, sebuah diagram tunggal dapat merepresentasikan objek, atribut dan operasi, asosiasi diantara objek, dan *message* yang dapat mengalir antar objek

Diagram disebut *object structure and message diagram* (OSMD), yang juga menggabungkan notasi dari batasan system. OSMD adalah bukan representasi berorientasi objek yang lengkap suatu system dan juga tidak dapat langsung diimplementasikan, tetapi lebih sebagai *first-cut* representasi berorientasi objek dan sebagai titik awal untuk melanjutkan pengembangan berorientasi objek.

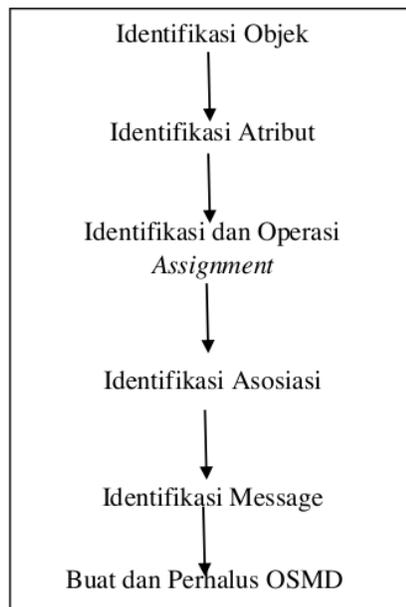
### 4. Strategi Pemetaan

Strategi pemetaan dimulai dengan mendeskripsikan rmodel fungsional dengan DFD, ERD dan kamus data. Hasil dari operasi pemetaan adalah struktur objek dan *message diagram* (OSMD). Strategi otomatis secara menyeluruh adalah tidak realistik., tetapi aturan dan konstitusi heuristik membantu dalam menyediakan tingkat keotomatisan. Gambar 1 menyajikan model strategi pemetaan.



Gambar 1. Model sumber dan target untuk strategi pemetaan

Proses yang mengimplementasikan strategi, terlihat pada gambar 2 mencakup urutan enam fase berbeda dengan input terdefinisi jelas pada akhir setiap fase.



Gambar 2. Fase dalam proses pemetaan

### Fase I. Identifikasi objek

Objek adalah komponen terpenting pada system berorientasi objek dan oleh karena itu, fase pertama adalah mengidentifikasi abstraksi data utama atau objek pada system. Secara esensial identifikasi objek didasari pada pemetaan entitas di ERD, terminal pada DFD, simpanan di DFD dan beberapa kasus relasi di ERD yang berhubungan dengan objek yang terkait. Langkah yang mengikuti aturan fase pertama adalah sebagai berikut :

1. Petakan setiap entitas di ERD ke objek tunggal dengan nama yang sama
2. Petakan setiap terminal di DFD ke objek tunggal dengan nama yang sama
3. Petakan setiap simpanan data di DFD ke objek tunggal dengan nama yang sama
4. Petakan setiap objek yang terkait (relasi dan atribut ) di ERD ke objek tunggal
5. Tentukan gabungan sederhana dari himpunan objek yang diidentifikasi pada langkah 1 sd 4 menggunakan nama objek sebagai kriteria identifikasi
6. Perhalus daftar objek yang dibuat diatas dengan memecah objek umum dengan kemungkinan nama berbeda, turunkan sebagai hasil aplikasi dari aturan berbeda dan ganti nama atau hapus beberapa objek.
7. (Langkah opsional) Konfirmasi bahwa sekumpulan objek mencakup "objek pusat" atau satu yang merepresentasikan system. Jika tidak ada, lalu identifikasi satu dari konteks system
8. (Langkah opsional) Perhalus daftar objek dengan mendefinisikan batas system dan mengeluarkan objek yang berada di luar batas ini.

Empat langkah pertama independen tetapi mungkin menghasilkan objek umum, dan adalah penting untuk mengidentifikasi kumpulan objek yang berbeda. Hal ini diperhatikan oleh langkah lima dimana objek dengan nama identik dipisahkan dengan operasi gabungan

Langkah enam mirip langkah lima tetapi memerlukan beberapa interaksi pengguna dan meliputi optimasi dengan operasi gabungan sederhana. Dengan kata lain, terminal di DFD dan simpanan data di DFD yang sama bisa secara aktual mengacu ke objek tunggal, tetapi mereka tidak pasti mempunyai nama yang sama. Jadi tanggung jawab pengguna untuk mengenali kesamaan semantik antar objek, daripada hanya sekedar memecah objek. Jika diperlukan, objek mungkin diganti nama sehingga semua nama unik.

Langkah tujuh adalah pilihan dan mungkin tidak pertimbangan dibawah semua keadaan.. Hal ini untuk memastikan bahwa ada beberapa objek merepresentasikan fungsi utama system dan proses pemetaan dapat fokus pada objek. Keuntungan langkah ini adalah sewaktu operasi penugasan ke objek, selalu ada objek "default" yang mempunyai operasi yang bisa di-assign. Langkah delapan juga opsional sewaktu definisi suatu batas system dihasilkan setelah proses pemetaan keseluruhan telah lengkap.

Akhir fase ini, pemetaan telah mengidentifikasi semua kemungkinan kandidat objek.

## **Fase II. Identifikasi Atribut**

Fase ini bertanggung jawab untuk mengidentifikasi atribut yang diasosiasikan dengan fase identifikasi objek pada fase 1. Atribut objek menyediakan informasi lanjut tentang status aktual dari suatu objek dan juga bantuan mendefinisikan detail lanjut tentang apa objek yang disajikan. Pemberian atribut langsung diikuti identifikasi objek sehingga atribut dapat digunakan dalam identifikasi operasi pada fase berikutnya.

Atribut didapat hampir seluruhnya dari kamus data dan aturan yang relevan yang diturunkan. Langkah khusus meliputi petunjuk sebagai berikut :

1. Setiap objek didaftar objek, amati simpanan data untuk sembarang definisi tipe data abstrak yang mempunyai nama sarna. Jika definisi ada dan merupakan urutan komponen, petakan komponen dari definisi ini sebagai atribut. Setiap definisi data sebaiknya dipecahkan ke level non primitif terendah.
2. Jika ada atribut objek mengindikasikan pengulangan beberapa item data, dan item data mempunyai nama sama sebagai kandidat objek atau berkoresponden ke kandidat objek yang telah diidentifikasi, maka atribut adalah menyederhanakan acuan ke objek dan ini dimungkinkan untuk mengidentifikasi asosiasi antara dua objek diatas.
3. Daftar semua aliran data di DFD. Setiap item data, pastikan apakah mungkin menjadi atribut atau objek. Jika sudah, lalu tambahkan item data tersebut ke daftar atribut dari objek yang diidentifikasi.

*Output* pada akhir fase kedua adalah daftar kandidat objek, atribut untuk objek-objek tersebut, dan (rnungkin) asosiasi antara beberapa objek.

### **Fase III. Identifikasi dan Operasi Assignment**

Fase ini merupakan fase paling krusial dan mungkin paling kompleks pada proses pemetaan. Fase ini bertanggungjawab untuk pemetaan kebutuhan fungsional system ke dunia obyek dalam bentuk operasi yang didefinisikan.

Strategi ini disederhanakan dengan fakta bahwa penggunaan *flattened* DED (DFDD) dan proses langsung dipetakan ke operasi. Oleh karena itu, strategi tidak berkaitan dengan dekomposisi dan kombinasi proses ke operasi.

Langkah yang merupakan fase ini, petunjuknya sebagai berikut :

1. Petakan setiap proses pada DFDD ke suatu operasi berkorespondensi dengan nama yang sama.
2. Asugn operasi ke objek sebagai berikut:
  - a. Setiap input item data ke proses berkorespondensi ke operasi, identifikasi suatu objek dalam daftar objek dengan nama sama sebagai item data atau berkoresponden ke item data, jika ada
  - b. Jika tidak ada objek diidentifikasi pada langkah 2.a., lalu untuk setiap input item data ke proses berkorespondensi ke operasi, identifikasi suatu objek pada daftar objek dengan atribut yang mempunyai nama yang sama sebagai data item atau berkoresponden ke item data jika ada. Jika input item data telah didefinisikan di karnus data, lalu identifikasi sembarang objek sedemikian hingga suatu subset atribut objek mempunyai nama yang sama atau berkoresponden ke komponen dari input item data. jika objek ada.

Jika pada akhir langkah 2.b., untuk setiap proses berkoresponden ke operasi, hanya satu objek diidentifikasi sebagai korespondensi ke input item data, lalu beri operasi ke objek.
3. Jika akhir langkah 2.b., untuk setiap proses berkoresponden ke operasi, lebih dari satu objek diidentifikasi sebagai korespondensi ke input item data, lalu operasi sebaiknya diberikan ke satu objek. Identifikasi setiap input item data ada output item datanya yang dimodifikasi dari bentuk input data item
4. Ulangi setiap operasi yang tidak diberikan ke objek pada akhir langkah 3. Jika proses berkorespondensi ke operasi yang hanya mempunyai satu input data, lalu buat objek baru. Jika proses mempunyai dari satu input item data, uji proses ke objek yang ada atau definisikan objek baru untuk diberi operasi tersebut.
5. (Opsional) Untuk setiap operasi, identifikasi suatu operasi invers jika ada. Beri operasi – operasi ini ke obyek
6. (Opsional) Uji setiap objek dan identifikasi operasi lebih lanjut yang memerlukan definisi sedemikian hingga objek bisa melakukan aturan ini.

*Output* pada akhir fase adalah kemungkinan perubahan pada daftar kandidat objek.

### **Fase IV. Identifikasi Asosiasi antar objek**

Fase ini mengidentifikasi asosiasi antar objek yang bisa dipandang sebagai jalur komunikasi dimana message dapat dikirim. ERD adalah suatu sumber informasi yang baik untuk

mengidentifikasi asosiasi yang mungkin. Langkah berikut dari fase ini yang dapat sebagai petunjuk

1. Untuk setiap relasi pada ERD, jika entitas mempunyai nama yang sama sebagai objek atau herkoresponden dengan objek, lalu buat suatu asosiasi antara dua objek.
2. Ujilah pasangan objek dan pastikan apakah mereka berkoresponden ke suatu terminal dan simpanan data. Jika ada, lalu buat asosiasi antar objek.

Output akhir fase ini, penambahan pada daftar objek, atribut dan operasi yang diidentifikasi pada akhir fase 3, berisi asosiasi antar objek.

#### **Fase 5. Identifikasi message antar Objek**

Fase ini menangkap beberapa aspek natural yang dinamis dari system berorientasi objek dengan mengidentifikasi pesan yang mengalir antar objek. Fase ini mempunyai satu langkah sebagai petunjuk sebagai berikut :

Petakan setiap operasi ke suatu *message* yang dapat mengalir dari satu atau lebih objek. Output pada akhir fase ini adalah daftar objek, atribut dan operasi, suatu (mungkin dimodifikasi) daftar asosiasi, dan message antar objek.

#### **Fase 6. Pembuatan dan penghalusan struktur objek dan *message diagram***

Setelah fase di atas lengkap, representasi orientasi objek dalam bentuk struktur objek dan *message diagram* (OSMD) harus dibuat. Hal ini diikuti dengan "pembersihan" dari representasi *first-cut system* dan tergantung penuh dari pengalaman user dan pengetahuan pada system. Adalah tidak menguntungkan bahwa system dapat diimplementasikan langsung tanpa penghalusan lanjut dan penggabungan dari disain informasi. Fase ini terdiri tiga langkah berikut :

1. Gambarkan OSMD menggunakan informasi yang didapat dari lima langkah pertama
2. Definisikan batasan system dengan memasukan kedalam system hanya objek yang mempunyai operasi yang didefinisikan.
3. Perhalus diagram untuk merefleksikan system yang lebih baik. Hal ini mencakup perubahan daftar objek, atribut, operasi serta *message* yang ada antar objek.

*Output* dari proses pemetaan pada akhir fase ini, terdiri struktur akhir objek dan *message diagram*, daftar akhir objek, atribut, operasi, asosiasi serta pesan antar objek.

#### **Kesimpulan**

Beberapa lingkungan perangkat lunak membutuhkan dukungan berkelanjutan model fungsional ketika berevolusi ke proses teknik berorientasi objek. Untuk beberapa kasus, suatu organisasi mungkin menentukan bahwa analisis dan spesifikasi kebutuhan lebih baik didukung oleh pendekatan berorientasi fungsi sedangkan pengembangan akan menguntungkan dengan disain dan implementasi berorientasi objek.

Strategi pemetaan dari model fungsional ke model objek dapat dipercaya dapat diaplikasi secara luas. Untuk lingkungan berbeda, model dapat diubah dan strategi dapat dimodifikasi untuk mengakomodasi representasi baru tanpa mengubah langkah strategi dasar.

Pentingnya, strategi pemetaan adalah masuk akal untuk diotomatisasi, dan tools dapat dikembangkan untuk petunjuk pengembangan. Dengan strategi dan tool perangkat lunak untuk mendukung pemetaan, transisi dapat konsisten dan mudah diulang.

**Daftar Pustaka**

Bahrami, A., Object Oriented Systems Development, McGraw-1-1111, 1999

DeMarco, T., Structured Analysis and System Specification, Yourdon Press, New York 1978

Pressman, R., Software engineering, fourth edition, McGraw-Hill, 1996

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lorenzen, W., Object Oriented Modeling and Design, Prentice Hall, Englewood Cliffs, New Jersey, 1991

Yourdon, E., Modern Structured Analysis, Yourdon Press, Englewood New Jersey, 1989

# Strategi Pemetaan dari Model Fungsional ke Model Object Oriented

---

## ORIGINALITY REPORT

---

**5%**

SIMILARITY INDEX

**5%**

INTERNET SOURCES

**5%**

PUBLICATIONS

**1%**

STUDENT PAPERS

---

## MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

---

5%

★ Joseph George. "A strategy for mapping from function-oriented software models to object-oriented software models", ACM SIGSOFT Software Engineering Notes, 3/1/1996

Publication

---

Exclude quotes Off

Exclude matches Off

Exclude bibliography On