

# IMPLEMENTASI EDITOR MODEL DATA KONSEPTUAL DAN MODEL DATA FISIK DENGAN ROUND-TRIP ENGINEERING

Aldy Sefan Rezanaldy<sup>1)</sup>, Siti Rochimah<sup>2)</sup>, Suhadi Lili<sup>2)</sup>

<sup>1)</sup> Jurusan Teknik Informatika, STMIK Multi Data Palembang

<sup>2)</sup> Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember  
Kampus ITS Sukolilo, Surabaya  
Email: [siti@its-sby.edu](mailto:siti@its-sby.edu)

## ABSTRAK

Desain model basis data merupakan sebuah fase penting dalam pengembangan sebuah Aplikasi Sistem Informasi. Editor Model Data yang digunakan untuk melakukan desain basis data sangat diperlukan dalam dunia IT. Sebagian Editor yang ada saat ini belum menerapkan konsep round-trip engineering secara real time, sehingga perubahan yang terjadi pada satu data model membutuhkan event update untuk melakukan pembaharuan pada model data yang lainnya.

Editor model data ini merupakan editor dengan round-trip engineering. Konversi bolak-balik dilakukan antara data model konseptual dan data model fisik. Editor ini dikembangkan dengan menggunakan C# .NET Framework dan implementasi desain pola pada Object Oriented. Dalam implementasi sebuah editor, yang merupakan bagian terpenting selain berjalannya seluruh fitur yang ada adalah tentang performa dan kenyamanan user ketika menggunakannya. Performa dan kenyamanan user menjadi penilaian tersendiri pada sebuah editor model data.

Aplikasi yang dihasilkan diharapkan dapat digunakan untuk melakukan desain basis data dengan menerapkan metode konversi bolak-balik, sehingga tidak diperlukan proses perbaruan dari model data yang satu ke model data yang lain secara manual. Diharapkan dapat mengurangi satu proses update dan sinkronisasi dokumen perangkat lunak. Satu tahap mengurangi proses dalam pengembangan perangkat lunak, dan mempercepat dalam pengembangan.

**Kata Kunci:** Editor Model Data, Round Trip Engineering, Model data konseptual, Model data fisik

## 1 PENDAHULUAN

Penelitian ini bertujuan melanjutkan proses perancangan editor dengan mengimplementasikan *user interface* dan manipulasi model data sebuah CASE Tool yang mendukung pemodelan dengan *round-trip engineering* atau koversi bolak balik antara model data konseptual dan model data fisik [2].

Beberapa masalah yang diangkat dalam pengembangan dapat dipaparkan sebagai berikut:

- Bagaimana mengimplementasikan *user interface* model data konseptual dan model data fisik yang *user friendly*, sederhana dan dapat digunakan dengan cepat dan tepat untuk melakukan desain model data.
- Bagaimana mewujudkan editor yang menggunakan *round-trip engineering* pada model data konseptual dan model data fisik.
- Bagaimana mekanisme manipulasi *object* yang memudahkan pengguna.
- Bagaimana melakukan pengecekan *warning* dan *error* pada diagram CDM dan PDM.
- Bagaimana mengimplementasikan *Undo Redo*, pada aplikasi yang menggunakan *round-trip engineering*.

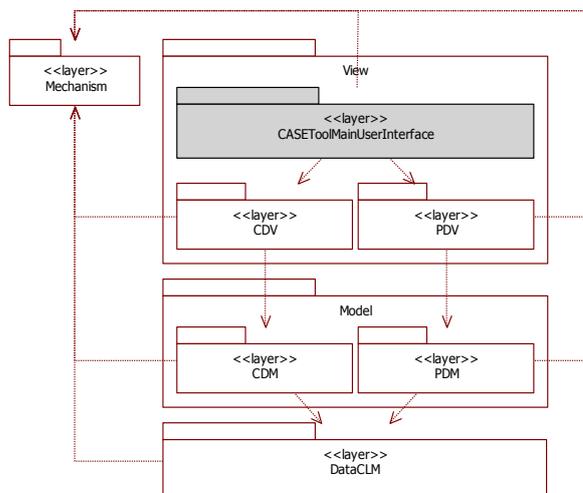
Aplikasi ini dikembangkan di atas arsitektur sistem yang telah ada, “Nasty Steroid” yang mendukung *round-trip engineering* secara *realtime* antara CDM dan PDM. Aplikasi ini dikembangkan dengan menggunakan bahasa C# pada .NET Framework [3]. Arsitektur aplikasi ini terdiri dari beberapa layer yang dijelaskan pada **Gambar 1**:

- Mechanism*, menangani pembaharuan informasi dari obyek-obyek model data dan *form factory*. Layer ini berhubungan dengan seluruh domain yang lainnya.
- Case Tool Main user Interface*, layer ini menangani tampilan antar muka utama dari Editor tool atau CASE Tool, seperti *form* menu utama dan *project browser*.
- Conceptual Data View (CDV)*, layer ini sebagai penampilan diagram (*diagram viewing*) data model konseptual. CDV juga sebagai tempat melakukan edit dan manipulasi obyek. Di dalamnya bisa ditambahkan obyek pada diagram, dilakukan manipulasi pada obyek, dan

- penghapusan obyek. Obyek-obyek diagram yang ada pada domain ini adalah obyek-obyek model data konseptual.
- d. *Physical Data View (PDV)*, sebagai penampilan diagram (*diagram viewing*) data model *physical*, sebagai tempat melakukan edit dan manipulasi obyek. Di dalamnya bisa ditambahkan obyek pada diagram, dilakukan manipulasi pada obyek, dan penghapusan obyek. Obyek-obyek diagram yang ada pada domain ini adalah obyek-obyek model data fisik.
  - e. *Conceptual Data Model (CDM)*, sebagai representasi dari obyek-obyek yang ada pada data model konseptual. Diantaranya, *Entity, Association, Data Item, Relationship, Inheritance, Association Link*, dan obyek-obyek lain yang ada pada data model konseptual.
  - f. *Physical Data Model (PDM)*, sebagai representasi dari obyek-obyek yang ada pada data model fisik. Diantaranya, *Table, Column, Reference*, dan obyek-obyek lain yang ada pada data model data fisik.
  - g. *Data Common Language Model (Data CLM)*, merupakan sebuah domain yang merupakan media untuk melakukan konversi bolak-balik. *Data Common Language Model (Data CLM)* merupakan *super set* dari data model konseptual dan data model fisik. Istilah lainnya adalah bahasa perantara dari data model konseptual dan data model fisik.

Arsitektur yang telah ada memiliki tujuh layer utama yang terbagi dalam empat *functional domain* (seperti terlihat pada **Gambar 1**), yaitu:

- 1) *Domain View*, yaitu layer *Case Tool Main User Interface, CDV, dan PDV*
- 2) *Domain Model*, yaitu layer *CDM dan PDM*
- 3) *Domain Data CLM*, layer *Data CLM*
- 4) *Domain Mechanism*, layer *Mechanism*



**Gambar 1.** Arsitektur Sistem Yang Telah Ada 'Nasty Steroid'

Arsitektur **Gambar 1** menjelaskan hubungan antar layer dan alur informasi yang terjadi antar domain. Layer *Mechanism* adalah layer yang menghubungkan seluruh layer yang lain. Alur informasi berjalan dari layer CASE Tool Main User Interface, kemudian ke layer CDV atau PDV yang selanjutnya menuju layer CDM atau PDM. Dan alur informasi ini berujung di DataCLM yang akan menginterpretasikan diagram CDM dan PDM.

## 2 FITUR-FITUR BARU PADA EDITOR

Aplikasi editor ini telah dikembangkan sebelumnya dengan beberapa fitur. Pengembangan pada tahap ini, menambahkan beberapa fitur pada Editor sebelumnya.

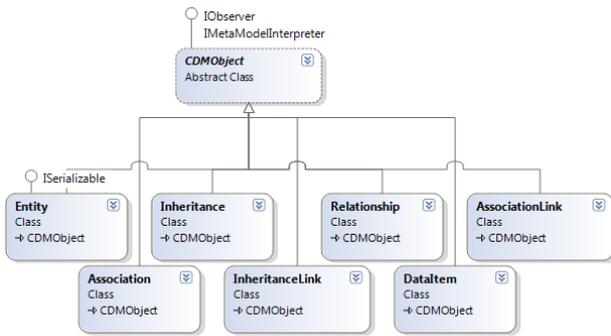
**Tabel 1.** Daftar Fitur-Fitur Editor CDM dan PDM

No	Fitur	Lama	Baru
1	Add, manipulasi, dan delete Object	ada	ada
2	Show Object List	ada	ada
3	Zoom in / out	tidak ada	Ada
4	Checking pada Object dalam diagram canvas	Tidak ada	ada
5	Checking detail per-object	Tidak ada	Ada
6	Undo Redo Command	Tidak Ada	Ada
7	Menambahkan Object drawing dari objct model yang telah ada	Ada, dengan mekanisme create Drawing baru	Ada, dengan mekanisme check Available Link
8	Mekanisme manipulasi	Form Properties	Pop up dan Form

## 3 IMPLEMENTASI CLASS EDITOR

Class-class pada aplikasi ini sebagian mengimplementasikan *Design Pattern* [1]. Desain *pattern* digunakan untuk melakukan mekanisme enkapsulasi *object* dengan *strategy pattern*, mekanisme *state* pada tool dan lain sebagainya.

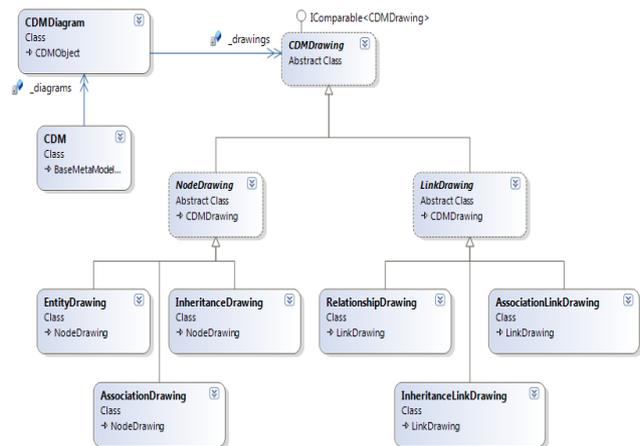
Class-class pada Layer CDM yang meliputi class-class model dan class-class untuk drawing model. Class-class drawing model memiliki assosiasi dengan masing-masing class model sesuai dengan spesifik model yang digambarkan, Class diagram object model dapat dilihat pada **Gambar 1**.



**Gambar 2.** Class Diagram Model pada CDM

- CDMObject, class abstrak yang merupakan parent class dari class-class model untuk enkapsulasi.
- Entity, class yang merupakan representasi dari object entity. Diturunkan dari class CDMObject. Class ini menjadi Subject pada EntityDrawing.
- Association, class yang merupakan representasi dari object association. Class ini merupakan turunan dari class CDMObject. Class ini menjadi Subject pada AssociationDrawing.
- Inheritance, class ini merupakan representasi dari object inheritance, memiliki parent dan children, diturunkan dari kelas CDMObject. Class ini menjadi Subject pada InheritanceDrawing.
- InheritanceLink, class ini merupakan representasi dari Link pada Inheritance. Berfungsi menghubungkan antara inheritance dan Entity parent dan childnya.
- Relationship, class ini merupakan representasi dari relationship, penghubung antara dua entity. Class ini memiliki Cardinality dan Participant. Class ini menjadi Subject pada RelationshipDrawing. Class ini diturunkan dari CDMObject.
- DataItem, class ini merupakan representasi dari data item atau attribute pada entity dan association.
- AssociationLink, class ini merupakan representasi dari Link antara association dan Entity. Memiliki source dan destination berupa association dan entity. Class ini diturunkan dari CDMObject.

Class-class drawing model merupakan class-class yang digunakan untuk menggambarkan object-object model pada diagram. Pada Implementasi class-class drawing terdapat fungsi Draw() yang dioverride dari class CDMDrawing. Fungsi Draw() berbeda-beda pada setiap class drawing model, sesuai dengan spesifik dan detail object model. Class Diagram Drawing pada **Gambar 3**.

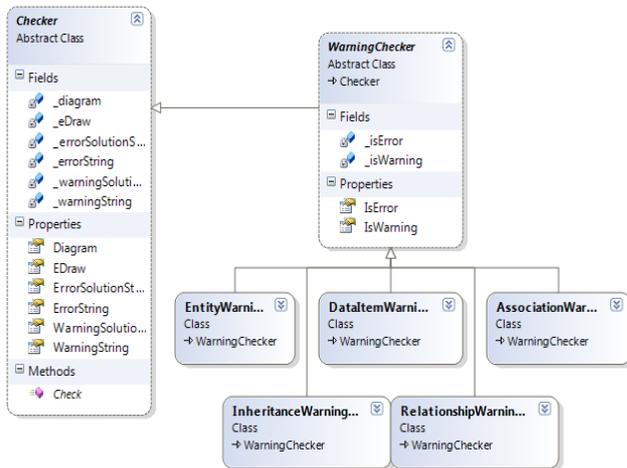


**Gambar 3.** Class Diagram Drawing Object pada CDM

- CDMDiagram, class untuk menampung seluruh object drawing dan menggambarkannya pada canvas, bisa disebut sebagai CareTaker object-object drawing pada diagram.
- CDMDrawing, class abstrak yang merupakan parent dari class-class yang di bawahnya. Class ini berasosiasi dengan CDM.
- NodeDrawing, adalah class abstrak yang menjadi parent bagi object-object drawing yang penggambarannya berupa node, bukan garis.
- LinkDrawing, adalah class abstrak yang menjadi parent dari object-object drawing yang penggambarannya berupa garis atau link, bukan berupa node.
- Entity Drawing, merupakan class yang memiliki karakter khusus untuk menggambarkan Entity berupa rectangle. Class ini diturunkan dari NodeDrawing.
- InheritanceDrawing, class yang memiliki karakter khusus untuk menggambarkan Inheritance, berupa elipse, sehingga berbeda dengan penggambaran object-object yang lainnya. Class ini diturunkan dari NodeDrawing.
- AssociationDrawing, class yang memiliki karakter khusus untuk menggambarkan association berupa rounded rectangle, berbeda dengan inheritance dan Entity. Class ini diturunkan dari NodeDrawing.
- RelationshipDrawing, class yang memiliki karakter khusus menggambarkan garis dengan cardinality dan modality (participant). Fungsinya adalah menghubungkan antara dua EntityDrawing. Diturunkan dari class LinkDrawing.

- i) AssociationLinkDrawing, class yang memiliki karakter khusus untuk menggambarkan garis dengan hanya satu cardinality dan participant. Fungsinya menghubungkan antara EntityDrawing dan AssociationDrawing. Class ini merupakan turunan dari class LinkDrawing.
- j) InheritanceLinkDrawing, class yang memiliki karakter menggambarkan garis antara Inheritance dan Entity (parent atau child). Class ini merupakan turunan dari class LinkDrawing.

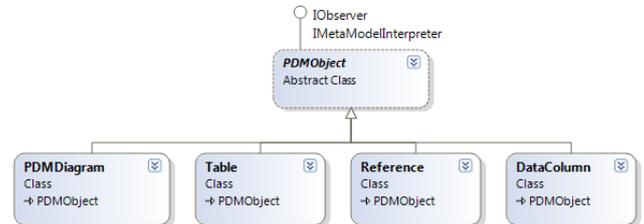
Selain class-class model dan drawing, ada class-class checker untuk melakukan pengecekan pada class-class drawing dan modelnya. Diagram Class checker pada **Gambar 4**. Masing-masing class checker memiliki fungsi-fungsi untuk melakukan checking pada masing-masing drawing dan modelnya.



**Gambar 4.** Diagram Class Checker pada CDM

- a) Checker, class ini merupakan parent dari class checker yang lainnya (enkapsulasi). Berisi List Warning dan Error. Class ini mempunyai property berupa CDMDiagram dan CDMDrawing sebagai Subject yang akan dicek.
- b) WarningChecker, class ini turunan langsung dari class checker, memiliki penanda isError dan isWarning. Yang digunakan untuk mengupdate nilai isError dan isWarning pada Object.
- c) EntityWarningChecker, class ini merupakan class checker untuk Entity.
- d) AssociationWarningChecker, class ini merupakan class checker untuk Association.
- e) InheritanceWarningChecker, class ini merupakan class checker untuk Inheritance.
- f) RelationshipWarningChecker, class ini merupakan class checker untuk relationship.

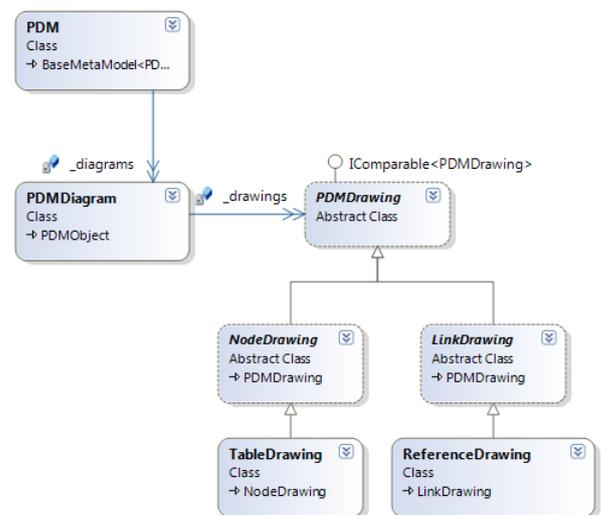
Class-class pada layer PDM, memiliki struktur yang sama dengan class-class pada layer CDM. Class-class pada Layer PDM meliputi class-class object model, drawing, dan checker. Diagram class Object Model pada PDM pada **Gambar 5**.



**Gambar 5.** Class Diagram Model pada PDM

- a) PDMObject, class yang merupakan Parent dari class-class object model pada PDM, dengan tujuan enkapsulasi class.
- b) PDMDiagram, class yang digunakan sebagai caretaker object-object drawing, yang di dalamnya object-object drawing digambar pada canvas. Class ini turunan dari CDMObject
- c) Table, class yang merupakan representasi dari table pada database. Class ini memiliki column-column yang merupakan atributnya.
- d) Reference, class ini merupakan representasi dari reference pada database. Menghubungkan antara table dengan table.
- e) DataColumn, class yang merupakan representasi dari kolom pada Table, merupakan bagian dari table.

Diagram class pada **Gambar 6** merupakan implementasi dari class-class drawing yang masing-masing berasosiasi dengan object model yang sesuai.



**Gambar 6.** Class Diagram Drawing Object pada PDM

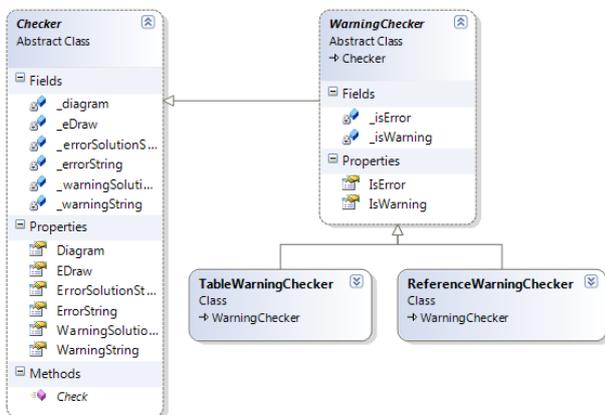
- a) PDM, class PDM ini merupakan class caretaker dari PDMDiagram yang ada pada aplikasi. Class ini berisi diagram-diagram PDM yang dibuat oleh user.
- b) PDMDiagram, class ini merupakan caretaker dari object-object drawing yang dienkapsulasi oleh PDMDrawing. Pada PDMDiagram inilah semua object drawing digambarkan pada canvas.
- c) PDMDrawing, merupakan class yang mengenkapsulasi object-object drawing.
- d) NodeDrawing, merupakan class abstrak yang merepresentasikan penggambaran berupa node, bukan garis atau link. Class ini akan diturunkan menjadi class konkret yang handle penggambaran Table.
- e) LinkDrawing, merupakan class abstrak yang merepresentasikan penggambaran berupa garis atau link, bukan berupa node. Class ini akan diturunkan menjadi class konkret yang handle penggambaran Reference.
- f) TableDrawing, merupakan kelas yang handle penggambaran table. Class ini merupakan turunan dari kelas NodeDrawing.
- g) ReferenceDrawing, merupakan kelas yang handle penggambaran link reference, penghubung antara table. Class ini merupakan turunan dari class LinkDrawing.

dan isWarning. Yang digunakan untuk mengupdate nilai isError dan isWarning pada Object Drawing.

- c) TableWarningChecker, class ini merupakan turunan dari WarningChecker yang secara khusus memiliki method method yang digunakan untuk melakukan pengecekan pada TableDrawing.
- d) ReferenceWarningChecker, class ini merupakan turunan dari WarningChecker yang secara khusus memiliki method method yang digunakan untuk melakukan pengecekan pada ReferenceDrawing.

Pada layer CDV terdapat class-class State Tool yang digunakan untuk membedakan behavior event CDMCanvas sesuai dengan masing-masing object yang telah dipilih. **Gambar** menggambarkan secara detail tentang class diagram Tool State pada CDM. Selain itu class-class yang berupa form-form untuk menampilkan object-object, form-form properties untuk masing-masing object, dan User Control yang digunakan untuk Pop up properties terdapat pada layer ini. Pada layer ini lah terdapat CDMCanvas yang menampung semua object drawing CDM dalam diagram dan menggambarkannya secara visual.

Pada layer PDV terdapat class-class State Tool yang digunakan untuk mekanisme state pada behavior event-event pada CDMCanvas. CDMCanvas merupakan sebuah class yang menampung semua object drawing PDM dalam diagram dan menggambarkannya pada canvas secara visual. Selain itu pada layer ini terdapat class-class yang berupa form-form untuk menampilkan object-object, form-form properties untuk masing-masing object, dan User Control yang digunakan untuk Pop up properties.



**Gambar 7.** Class Diagram Checker pada PDM

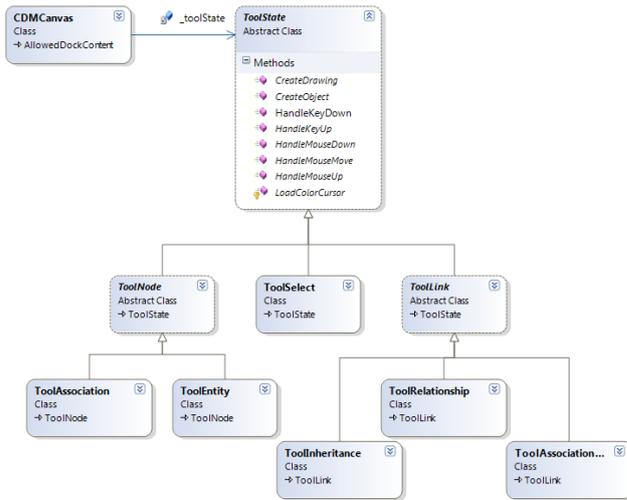
Class-class checker digunakan untuk melakukan checker pada Drawing yang berassisi dengan object model. Diagram class checker pada PDM pada **Gambar** .

- a) Checker, class ini merupakan parent dari class checker yang lainnya (enkapsulasi). Berisi List Warning dan Error. Class ini mempunyai property berupa PDMDiagram dan PDMDrawing sebagai Subject yang akan dicek.
- b) WarningChecker, class ini turunan langsung dari class checker, memiliki penanda isError

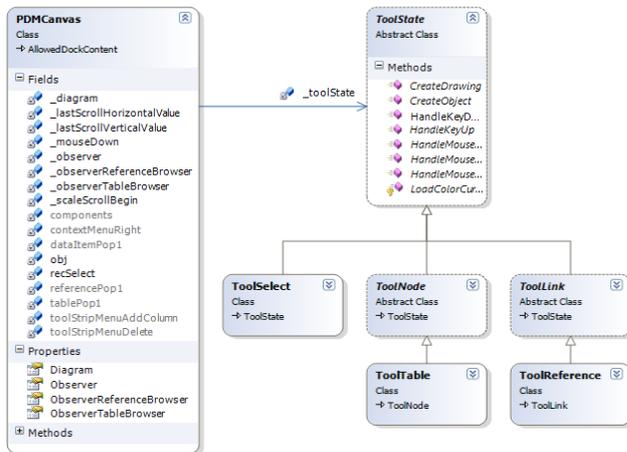
#### 4 IMPLEMENTASI FUNGSIONAL CASE

Implementasi masing-masing case pada editor ini tidak lepas dari mekanisme state pada event-event yang dijalankan. Mekanisme state ini diimplementasikan menggunakan pola state yang dapat merubah behavior-nya jika state internalnya berubah. Diagram Tool State pada CDM dan pada PDM pada Gambar 8 dan 9.

Setiap event pada diagram canvas, dihandle dengan toolstate di dalamnya. Tool untuk melakukan seleksi adalah toolSelect, sedangkan tool yang lain untuk menambahkan object drawing tertentu pada diagram canvas. Untuk memilih tool yang akan digunakan yaitu dengan memilih tombol pada tool strip **Gambar** dan **Gambar** .



Gambar 8. Class Diagram Tool State pada CDM



Gambar 9. Class Diagram Tool State pada PDM



Gambar 10. Tool Strip CDM



Gambar 11. Tool Strip PDM

4.1. Add Object

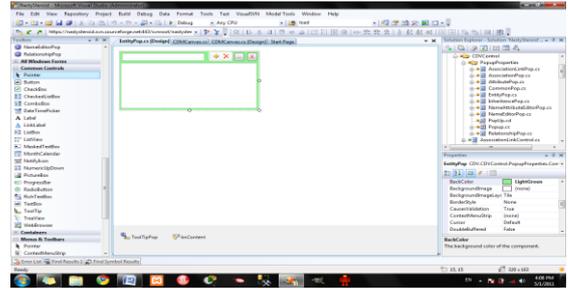
Untuk menambahkan sebuah object baru pada diagram canvas, maka dipilih tool untuk menambahkan object, yaitu tool selain tool select. Masing-masing tool akan menjalankan behavior event yang berbeda-beda pada canvas. Tool Entity, menambahkan entity drawing pada canvas, tool association menambahkan association drawing, dan lainnya akan menambahkan object drawing sesuai dengan tool yang digunakan.

Selain dengan menggunakan tool yang ada, penambahan object drawing pada diagram canvas

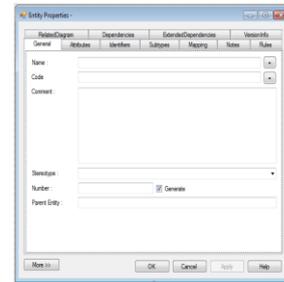
dilakukan dengan melakukan *drag* dan *drop* object pada project browser ke diagram canvas.

4.2. Manipulasi Object

Manipulasi Object dilakukan dengan melakukan seleksi terlebih dahulu pada object drawing yang ada pada diagram canvas. Manipulasi dapat dilakukan dengan menggunakan Pop Up properties (single click) atau dengan Form Properties (double click). Pop up properties untuk entity dan Form Properties untuk entity pada Gambar dan 13.



Gambar 12. Pop up Properties Entity



Gambar 13. Form Properties Entity

4.3. Delete Object

Penghapusan Object dilakukan dengan melakukan seleksi terlebih dahulu pada object-object yang akan dihapus. Penghapusan dapat berupa:

- a. Penghapusan object drawing saja, dengan menekan Tombol delete atau klik kanan delete pada object yang akan dihapus.
- b. Penghapusan object drawing dan object modelnya sekaligus, dengan menekan tombol SHIFT+delete atau melakukan penghapusan object pada project browsernya.

4.4. Zoom In / Out

Implementasi zoom in/out pada aplikasi editor sangat diperlukan mengetahui detail object-object dengan gambaran yang lebih besar, atau untuk membantu mengatur posisi object-object pada diagram canvas. Mekanisme zoom dilakukan dengan melakukan redraw ulang seluruh object dengan perubahan pada skala drawing-nya sesuai dengan skala diagram canvas-nya.

#### 4.5. Undo dan Redo

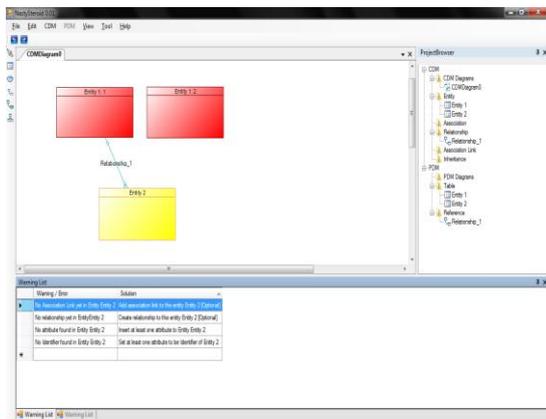
Implementasi Undo dan Redo State pada pada aplikasi editor menggunakan sebuah library UndoEasy.dll. Undo dan redo pada aplikasi ini terdapat pada layer CDV dan CDM. Undo dan redo menggunakan sebuah class UndoEngine yang disediakan oleh library UndoEasy. Object UndoEngine yang utama didefinisikan pada diagram masing-masing model data (CDM dan PDM). Tidak semua property pada object-object diimplementasikan di dalam mekanisme undo/redo, hanya beberapa properties saja yang diperlukan undo dan redo dimasukkan ke dalam implementasi undo dan redo.

#### 4.6. Show Warning/Error Detail

Untuk melakukan checking warning dan error pada diagram perlu dibuat class-class yang melakukan checking warning pada masing-masing object. Checking warning dan error pada entity, association, association link, relationship, inheritance, table, dan reference. Class diagram untuk checking pada **Gambar** dan **Gambar**. Untuk mengetahui detail warning dan error dapat dilakukan seleksi pada masing-masing object pada canvas. Untuk mengetahui warning dan error secara keseluruhan dengan melakukan check diagram pada menu tool bar TOOL.

#### 4.7. Show Warning/Error Diagram pada canvas

Selain melakukan pengecekan setiap kali dilakukan seleksi pada object, untuk memberitahukan indikasi warning dan error object-object pada diagram dilakukan pengecekan diagram setiap kali ada perubahan model. Pesan warning dan error disampaikan melalui perubahan warna yang terjadi pada object-object yang terdapat error atau warning di dalamnya. Warna merah menunjukkan error dan warna kuning menunjukkan warning.



**Gambar 14.** Pesan Warning dan Error pada Diagram Visual

### 5 UJI COBA

Uji coba pada aplikasi ini dilakukan dengan metode subjective testing. Uji coba dilakukan oleh beberapa orang (dari kalangan IT) yang belum pernah menggunakan aplikasi ini, namun diharapkan telah familiar dengan pemodelan CDM dan PDM.

Setelah melakukan uji coba dengan studi kasus yang digunakan, penguji memberikan penilaian pada aspek-aspek yang ditanyakan pada form uji coba yang disediakan. Kemudian hasil penilaian tersebut dianalisis untuk mengetahui bagaimana keadaan aplikasi di mata penguji. Penguji yang melakukan pengujian pada Tabel 2.

**Tabel 2.** Daftar Penguji

Nama Penguji	Pekerjaan
Kemas Dimas R	Mahasiswa T. Informatika
Rama Catur AP, S. Kom.	Software Engineer PT. Sefima
Muhammad Najib	Mahasiswa T. Informatika
Hisyam Fahmi	Mahasiswa T. Informatika

Setelah didapatkan hasil dari penguji dilakukan analisis tentang aspek-aspek yang dinilai pada pengujian. Didapatkan hasil rata-rata setiap aspek pengujian pada Tabel 3.

**Tabel 3.** Hasil Rata-Rata Penilaian Uji Coba

Aspek Penilaian	VG
Membuat diagram dari CDM dan PDM	3,2
Mengedit diagram yang telah ada, dari CDM dan PDM	3,62
Visualisasi	3,93
Reliability	3,37

### 6 KESIMPULAN

- Perancangan user Interface Editor, berkaitan dengan tampilan, *command*, dan *response* digunakan dan diimplementasikan untuk mempermudah pengguna melakukan desain dan editing object di dalamnya.
- Pop up properties digunakan untuk mempermudah pengguna melakukan manipulasi object pada diagram. Selain itu dengan menggunakan event yang tepat (*one click*) dapat meningkatkan kecepatan proses desain data model.
- Mekanisme Round-trip engineering pada aplikasi ini ditangani oleh pattern observer pada aplikasi ini. Object-object model yang

telah di manipulasi pada editor akan selalu di update dengan menggunakan observer pattern pada level data modelnya.

- d. Aplikasi editor ini dirancang dengan memperhatikan kesalahan (error) dan warning dalam melakukan desainnya. Ini untuk memberikan pesan kepada pengguna tentang diagramnya sedini mungkin, karena konsep yang dilakukan adalah round-trip engineering secara realtime. Pengecekan ini dilakukan setiap ada perubahan pada diagram, yaitu penambahan, edit, dan penghapusan object.
- e. Implementasi undo-redo menggunakan sebuah library UndoEasy pada level CDV dan CDM-nya. Library ini digunakan untuk mekanisme undo-redo pada beberapa event yang dinilai perlu untuk dilakukan undo-redo.

## 7 PENGHARGAAN

Penelitian ini terlaksana atas bantuan dana dari LPPM (Lembaga Penelitian dan Pengabdian

kepada Masyarakat) Institut Teknologi Sepuluh Nopember Surabaya melalui dana PUM ITS 2011. Disamping itu, penelitian ini didukung oleh tim pengembang perangkat lunak CASE Tool “Nasty Steroid” Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya.

## 8 DAFTAR PUSTAKA

- [1] Gamma, Erich, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Holland: Addison Wesley Longman, Inc., 1998.
- [2] Grinstein G, Georges. Database issues for data visualization: IEEE Visualization '93 workshop, San Jose, California, USA. 1993.
- [3] Troelsen A. *Pro C# 2008 and the .NET 3.5 Platform*. 4th Edition. New York: Apress; 2007.