

2CASE Tools dalam Lingkungan Operasi Open Source (Eksplorasi Fasilitas dan Fitur pada Together dan Poseidon)

Dewi Rosmala, S.Si, M. IT.(Jurusan Teknik Informatika, ITENAS Bandung)
Falahah, Ir., MT. (Jurusan Teknik Informatika Widyatama Bandung)

ABSTRAK

CASE (*Computer Aided Software Engineering*) adalah sejenis software yang dapat digunakan sebagai alat bantu otomatisasi aktivitas manual siklus pengembangan perangkat lunak.

Saat ini, penggunaan CASE untuk sebuah proyek pengembangan perangkat lunak sangat diperlukan karena berbagai kemudahan yang disediakan CASE sehingga proses perancangan dan pembangunan software menjadi lebih efisien dan memudahkan penelusuran (*traceability*) serta memungkinkan proses *reverse engineering* (mencari bentuk desain dari kode yang sudah jadi). Tingkatan integrasi yang disediakan oleh berbagai software CASE tersebut berbeda-beda dan penggunaannya dapat disesuaikan dengan tingkat kebutuhan dan daya adaptasi lingkungan yang menggunakan CASE itu sendiri.

Berbagai merk CASE yang saat ini beredar di pasaran, umumnya menggunakan standar pemodelan sistem berbasis object dengan notasi UML dan menyediakan fasilitas yang meliputi perancangan obyek, antarmuka, produksi template kode secara otomatis dan dokumentasi. Beberapa diantaranya dapat digunakan dalam lingkungan sistem operasi *open source* yang saat ini menjadi salah satu alternatif sistem operasi. Dari jenis terakhir ini, dua yang akan dibahas dalam tulisan ini yaitu *Together* dan *Poseidon*. Pembahasan difokuskan terhadap pengenalan fitur-fitur standar masing-masing software tersebut dan cara menggunakannya secara umum.

Kata kunci : *CASE, Open Source, fitur-fitur standar, Together, Poseidon*

ABSTRACT

CASE (Computer Aided Software Engineering) is a software tool to automate manual activities in the system development life cycle. Currently, the use of CASE in the development of software project is necessary. It provides the capability of making the system development life cycle of the project operatable, efficient and traceable. It also makes the reverse engineering available. Currently, there are many level of integration provided by various CASE software, and this feature was adjusted to the requirement and the environment user used CASE.

Most of the CASE in the market uses object oriented standard model with UML notation including object design, interface, automation code template and documentation features. Some of the CASEs run in the open source environment, as an alternate operating system. Together and Poseidon are two of the latest kind of CASE discussed in this paper focusing on introducing the standard feature and how to use these software.

Keywords: *CASE, Open Source, standard features, Together, Poseidon.*

PENDAHULUAN

Pengertian CASE

CASE (Computer Aided Software Engineering) adalah sejenis paket *software* yang memiliki kemampuan membantu otomatisasi aktifitas manual proses rekayasa perangkat lunak dan dapat memberikan pemahaman yang lebih baik bagi para perancang perangkat lunak. Aktifitas manual yang dapat dibantu otomatisasinya misalnya proses analisa kebutuhan (*requirement analysis*), pemodelan sistem (*system modeling*), pelacakan kesalahan (*debugging*), dan pengujian (*testing*).[2]

Orientasi penggunaan CASE adalah integrasi dengan kerangka kerja. Tingkatan integrasi penggunaan CASE bervariasi.[1]

Di tingkatan rendah, CASE dipandang sebagai alat titik solusi (*point solution tool*), yaitu alat yang digunakan untuk membantu suatu aktivitas tertentu rekayasa perangkat lunak, misalnya tahapan analisa saja, tetapi tidak langsung terhubung dengan alat lain, tidak terhubung dengan database *project*, dan bukan merupakan bagian dari *Integrated Case* (I-CASE).

Tingkatan berikutnya adalah jika alat individual ini kemudian digunakan untuk pertukaran data dengan alat lain, karena dalam tingkatan ini alat tersebut harus mampu mengeluarkan *output* dalam format standar dan juga harus mampu membaca format tersebut. Dalam beberapa kasus, beberapa *software* mampu membuat penyambung antara alat CASE ini dengan generator kode bahasa tertentu.

Tingkatan lainnya adalah *single-source integration* yaitu ketika satu *vendor* memadukan beberapa *tools* yang berbeda dan menjualnya sebagai satu paket.

Tingkatan tertinggi adalah IPSE (*Integrated Project Support Environment*), yaitu satu kondisi landasan untuk penerapan CASE sudah ada. CASE yang dibuat mengikuti IPSE akan saling kompatibel satu sama lain.

Beberapa *software* CASE biasanya memiliki kemampuan lengkap dari hulu ke hilir, mulai dari pendefinisian proses bisnis hingga perancangan *template* kode yang kemudian dapat dikembangkan menjadi kode program sesungguhnya. Tetapi ada juga beberapa *software* CASE yang hanya spesifik mendukung tahapan tertentu misalnya tahapan analisis saja atau tahapan perancangan database. [2]

CASE idealnya dapat menyimpan berbagai informasi yang berkaitan dengan proses perancangan perangkat lunak. Kemampuan penyimpanan informasi ini disebut dengan *repository*. Isi *repository* sebuah *software* CASE biasanya meliputi [1]:

- Informasi Umum Organisasi: meliputi struktur organisasi, analisa area bisnis, fungsi bisnis, aturan bisnis, skenario proses, arsitektur informasi;
- Desain Aplikasi: memuat informasi tentang desain aplikasi yang meliputi aturan metodologi, representasi grafis, diagram sistem, standar penamaan, aturan *referential-integrity*, struktur data, definisi proses, definisi *class*, struktur menu, definisi layar tampilan, definisi laporan, definisi logika, algoritma, aturan transformasi dan lain-lain,
- Konstruksi: menyimpan informasi sumber kode; kode *object*, *image* biner, kebergantungan konfigurasi, dan informasi perubahan;
- Validasi dan verifikasi: perencanaan test, kasus data untuk test, hasil test, analisa statistic;
- Informasi Manajemen proyek : Perencanaan proyek, struktur kerja, perkiraan jadwal, sumber informasi, laporan problem, permintaan perubahan, laporan status dan informasi audit;
- Dokumentasi sistem: dokumentasi permintaan kebutuhan, desain eksternal / internal, user manual.

Mengapa CASE Diperlukan?

CASE sangat membantu proses rekayasa perangkat lunak karena kenyataan proses rekayasa perangkat lunak biasanya sangat sulit. Seorang perancang perangkat lunak selain harus dapat memahami kebutuhan user juga harus dapat mendokumentasikan spesifikasi kebutuhan dan perancangannya dalam dokumen yang teratur, rapi, dan mudah ditelusuri. Seiring dengan perjalanan waktu dalam proses pendefinisian hingga implementasi seringkali terjadi perubahan-perubahan baik atas permintaan user maupun akibat bertambahnya atau berubahnya pemahaman terhadap sistem. Banyak waktu dihabiskan untuk mendokumentasikan perubahan-perubahan yang mungkin berdampak dari mulai pendefinisian proses hingga ke modul program. Ketersediaan alat yang dapat mendukung dokumentasi dari hulu ke hilir dalam proses yang terintegrasi dan otomatis (artinya, perubahan di hulu akan otomatis melakukan perubahan di hilir sehingga tidak ada proses yang terlewat untuk diubah / diperbaharui) akan sangat membantu proyek pengembangan perangkat lunak.

Aspek yang lebih penting dalam penggunaan CASE adalah kemampuan memberikan gambaran suatu perangkat lunak dari berbagai aspek (misalnya dari segi fungsi, tampilan panel, database dan arsitektur hardware sebagai satu kesatuan yang tidak dapat dipisahkan) akan sangat membantu seorang perancang perangkat lunak dalam memahami dan mendeskripsikan sistem yang dibangun. Hal ini sangat membantu dalam mengambil keputusan dan meningkatkan kualitas produk perangkat lunak secara keseluruhan.

Kapan dan Dimana CASE Diperlukan ?

CASE diperlukan jika kita harus membuat suatu perangkat lunak dalam skala cukup besar dengan tingkat kompleksitas cukup tinggi, misalnya suatu perangkat lunak yang

akan dijalankan oleh banyak user atau oleh beberapa fungsi dan memiliki banyak proses didalamnya atau mengakses banyak sumber data dari berbagai fungsi lainnya [1].

CASE dapat diterapkan di berbagai lingkungan, tetapi untuk hasil yang terbaik sebaiknya CASE diterapkan di lingkungan yang mendukung yaitu dalam suatu fungsi atau institusi yang telah memiliki prosedur kerja yang jelas dan stabil.

Penerapan CASE

Satu hal yang perlu diingat bahwa CASE adalah suatu alat untuk mengotomatisasi pekerjaan tetapi tidak mengambil alih fungsi analisis dan desain yang sebenarnya. CASE akan sangat berguna di tangan pengguna yang berpengalaman dalam pembangunan perangkat lunak.

Saat ini banyak sekali produk software CASE yang beredar di pasaran dengan berbagai katagori yang ada. Dari mulai sebagai alat pemodelan proses, manajemen proyek, dokumentasi, manajemen database, pembuatan program, perancangan antar muka dan berbagai katagori lainnya. Beberapa produk biasanya memiliki beberapa fasilitas atau kemampuan yaitu dari mulai alat analisa sampai alat pengujian dan dokumentasi.

Dua jenis software CASE yang akan dibahas adalah Poisedon dan Together dari perusahaan Togethersoft [3].

1. Together

Deskripsi ini diturunkan berdasarkan informasi Together 3.02 Whiteboard Edition, 1999

Together 3.02 dirilis dalam 3 versi yaitu Whiteboard Edition (free), Developer Edition dan Enterprise Edition.

Software ini dapat digunakan untuk membuat model UML dan memiliki kemampuan sebagai berikut:

- *'simultaneous round-trip engineering'* artinya satu teknologi *'single-source'* yang menjaga agar model yang kita desain

dengan kode yang diimplementasikan sinkron 100 %, selalu, tanpa *repository* perantara,

- Dukungan *round-trip engineering* dapat disesuaikan dan mendukung beberapa bahasa pemrograman yaitu Java, C++, Delphi, atau Object COBOL tergantung dari versi mana yang dipilih, dukungan yang umum dan luas adalah terhadap bahasa Java,
- Together dapat dengan mudah diinstal dan berjalan pada berbagai sistem operasi yang berbeda meliputi Windows NT/98/95, Solaris, dan Linux
- Dapat merancang proyek *software* baru dan secara simultan juga membuat kode *framework* (kerangka kerja) untuk implementasi,
- Dari kode yang ada, dapat dengan cepat dan mudah dibuat diagram UML dan dokumentasi HTML lengkap meliputi Java2's JavaDoc,
- Dengan menggunakan Java, Jpython, atau script TCL dapat dibuat dokumentasi sesuai keperluan (custom), metrik, check, kode program atau membuat output sesuai keperluan berdasarkan model UML yang dipergunakan. Script juga dapat digunakan untuk mengembangkan kemampuan Together melalui *open Together API*,
- Dapat mengimport model dan mengeksport model dari dan ke format *Rational Rose 98* (.mdl),
- Together dirancang untuk terintegrasi dengan mudah, dengan *tools* berbasis file yang biasa digunakan, misalnya kode editor, IDE atau kontrol versi dari sistem. Sistem menu dapat diatur dan berintegrasi dengan *tools* ini. Integrasi antara *tools* untuk *development* dan kontrol versi terdapat dalam satu paket,
- Together tidak menimpa / mematikan proses pemodelan yang lain. Tetapi dirancang untuk mendukung semua proses termasuk *Rational Unified Process*,
- Together meliputi proses Feature-driven Development.

Untuk dapat menggunakan Together, ada beberapa hal yang harus diketahui oleh pengguna yaitu:

- Dasar-dasar Unified Modeling Language (UML),
- Proses bisnis dalam suatu organisasi untuk merancang dan memodelkan sistem atau komponen berorientasi object,
- Target bahasa pemrograman yang akan digunakan,
- Cara menginstal Java VM dan menjalankan aplikasi JAVA dengan Java VM,
- *Tools* eksternal seperti IDE dan version control system (VCS).

Cara menggunakan Together

Tampilan utama Together (Gambar 1) memudahkan kita untuk mengakses semua pekerjaan yang sedang berlangsung. Kita dapat membuka lebih dari satu diagram atau file dan bekerja dengan file atau diagram tersebut secara kongkuren dalam window yang sama. Setiap diagram atau file yang dibuka memiliki tab sendiri di panel window. Window utama terbagi atas 4 wilayah panel utama yaitu :

1. Panel *Resource*, digunakan untuk navigasi sistem dan *project*. Panel ini juga memudahkan untuk mengakses *script* dan komponen *reusable*, dan menampilkan panel kontrol properti, *hyperlink*, dan komentar *source-code*. Panel ini terdiri atas *explorer* dan *inspector*,
2. Panel *Diagram*, digunakan untuk menggambar model secara visual,
3. Panel *Editor*, digunakan untuk melihat dan mengedit source code elemen diagram, *scripts* atau text file,
4. Panel *Message*, menampilkan pesan-pesan dari sistem. Panel ini secara default disembunyikan (*hidden*),

Beberapa fasilitas atau kemampuan yang tersedia dalam Together yaitu sebagai berikut:

1. *Simultaneous Round-trip Engineering*,

2. *UML diagrams and support,*
3. *Patterns support,*
4. *Java Enterprise components,*
5. *Multi-user Development,*
6. *Documentation Generation,*
7. *Metrics and QA,*
8. *Integrating with your development environment,*
9. *Feature-driven Development,*
10. *Customization,*
11. *Large project management,*
12. *View Management,*
13. *Wallchart Printing,*
14. *Rational Rose Import-Export,*
15. *Together Open API.*

Berikut akan dibahas secara singkat beberapa fasilitas / kemampuan tersebut :

Simultaneous Round-trip Engineering

Simultaneous Round-trip Engineering artinya bahwa model class UML selalu sinkron dengan source code yang mengimplementasikan model tersebut. Jika kita mengubah sesuatu pada diagram class maka source code yang berhubungan dengan class tersebut juga otomatis berubah. Demikian sebaliknya, jika kode diubah maka tampilan visual diagram juga berubah. Tidak ada repository perantara, tidak ada *batch code generation*, dan tidak ada resiko kehilangan kode.

UML Diagrams and Support

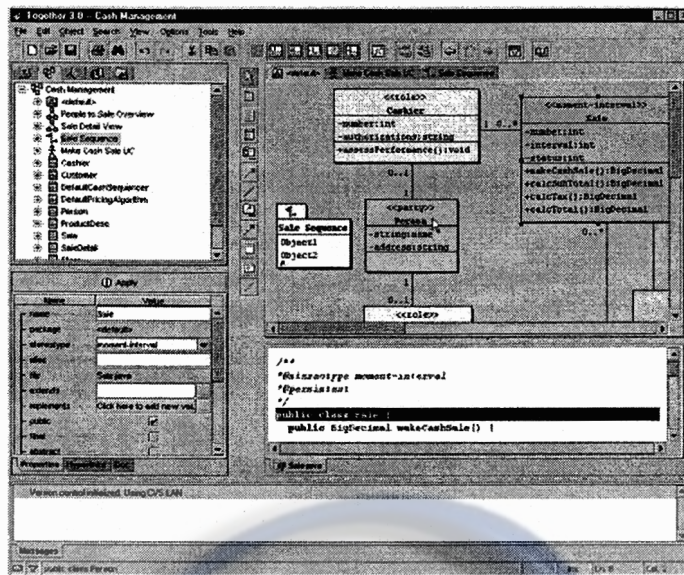
Together mendukung hampir semua diagram yang diperlukan dan notasi yang didefinisikan oleh UML. Saat ini diagram UML yang didukung oleh Together adalah : Class, Sequence, Statechart, Business Process, Deployment, Usecase, Collaboration, Activity dan Component. (Gambar 2).

Versi Free Whiteboard Edition hanya mendukung Diagram *Class* dan *Package*. Diagram *Business Process* tersedia pada versi Enterprise (Together/E). Enterprise edition juga mendukung diagram E-R untuk pemodelan data dan juga dapat digunakan secara fleksibel untuk membuat diagram lain sesuai keperluan.

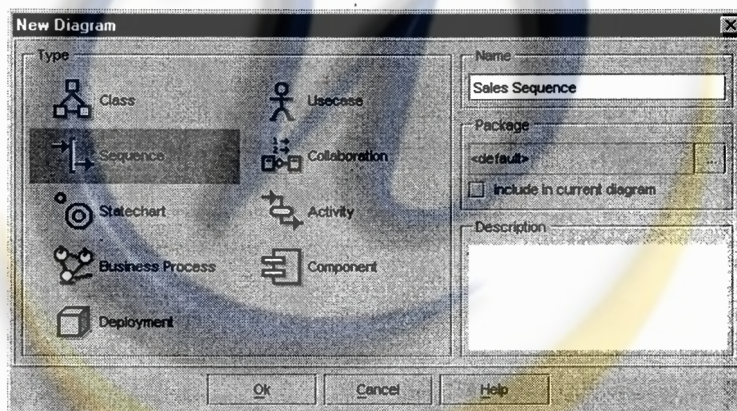
Pattern Support

Adanya *reusable pattern* akan membuat proses pembuatan software menjadi lebih efisien dan efektif. Together memudahkan penggunaan *pattern* dengan menyediakan pattern standar dan sample, meliputi pattern GoF dan Enterprise JavaBeans. Edisi Whiteboard menyediakan beberapa pattern dasar untuk class, member dan links. Pattern GoF ada pada Edisi Developer, dan Enterprise Java Beans ada pada Together/E.

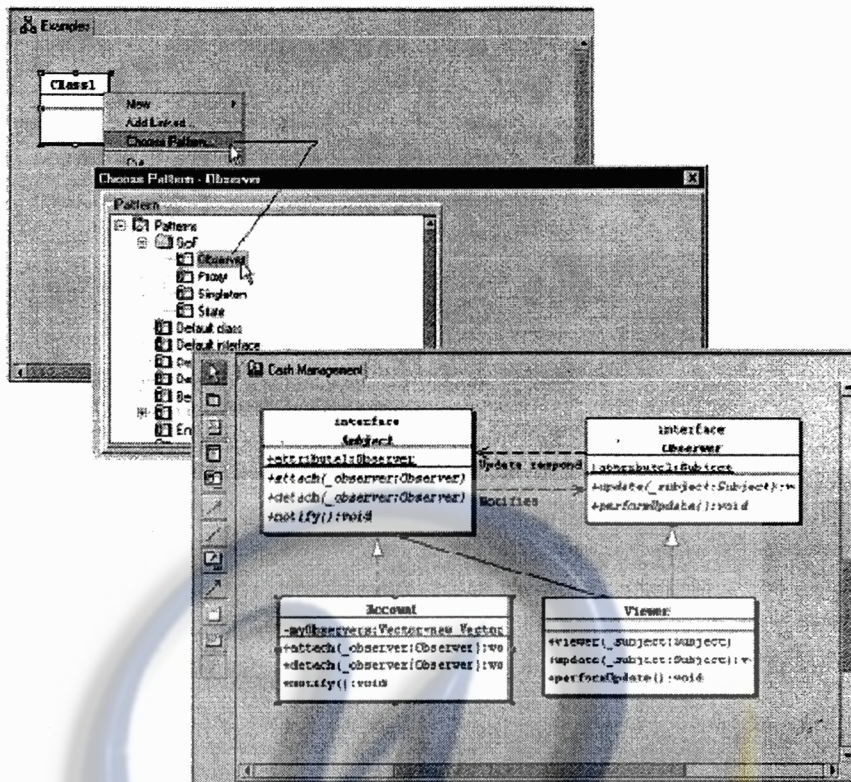
Kita dapat mengubah sembarang class dari satu class tunggal menjadi sebuah pattern. Caranya pilih class tersebut dan pattern yang diinginkan maka pattern akan dibuat secara otomatis (Gambar 3).



Gambar 1. Tampilan utama Together 3.02.



Gambar 2. Jenis-jenis Diagram UML pada Together.



Gambar 3. Perubahan dari Class menjadi Pattern Secara Otomatis.

Java Enterprise Components

Together menyediakan lebih dari 60 macam domain –specific komponen enterprise-level modeling dan berwarna. Paket komponen enterprise otomatis tersedia ketika membuat project baru Together 3. (Gambar 4).

MultiUser Development

Together edisi Developer dan Together/.E mendukung pengembangan *multiuser*. Fasilitas ini dapat digunakan untuk :

- melindungi asset software perusahaan,
- Membantu tim untuk kerjasama.

Multi-user Version Control System

Together menyediakan fasilitas pengontrolan versi sistem tanpa kita harus

membuat sendiri atau secara manual memecah model menjadi submodel, dan secara konsisten memecah submodel ini menjadi files untuk keperluan kontrol versi sistem.

Seting konfigurasi Multilevel

Penggunaan Advanced mode dari opsi dialog Default options memungkinkan kita untuk mengatur opsi konfigurasi dengan 3 level yaitu global untuk instalasi, *project specific*, dan diagram local. Setting untuk setiap level disatukan bersama. Jika opsi ini diset di multiple levels, maka yang dianggap berlaku adalah local setting. Misalnya diagram level akan menimpa *project level*. Tetapi, opsi yang ditandai dengan ‘Final’ di level global (oleh seorang *project administrator*) tidak dapat ditimpa level local. Dengan cara ini dapat dibuat format kode atau blueprint opsi instalasi berbasis server.

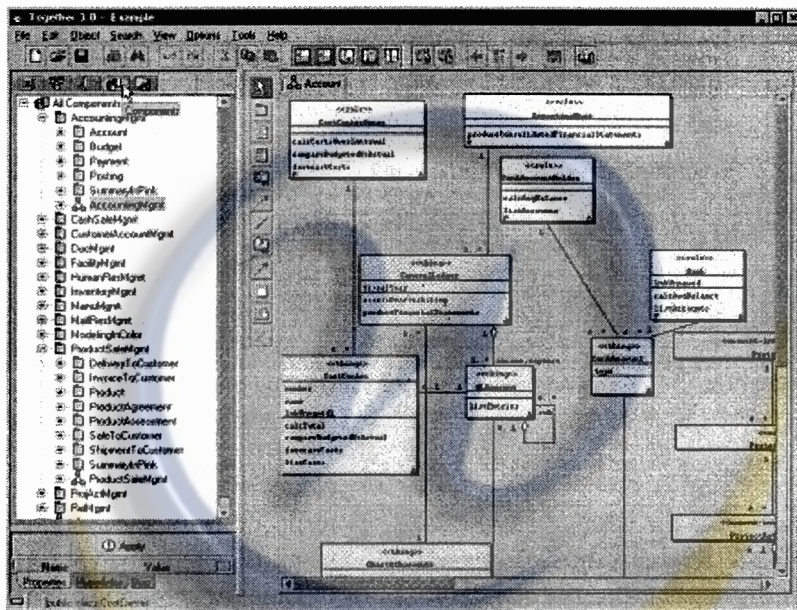
Pembuatan Dokumen

Pembuatan dokumen tersedia dalam versi Together Developer dan Together/E. Macam dokumen yang dapat dibuat dengan Together.

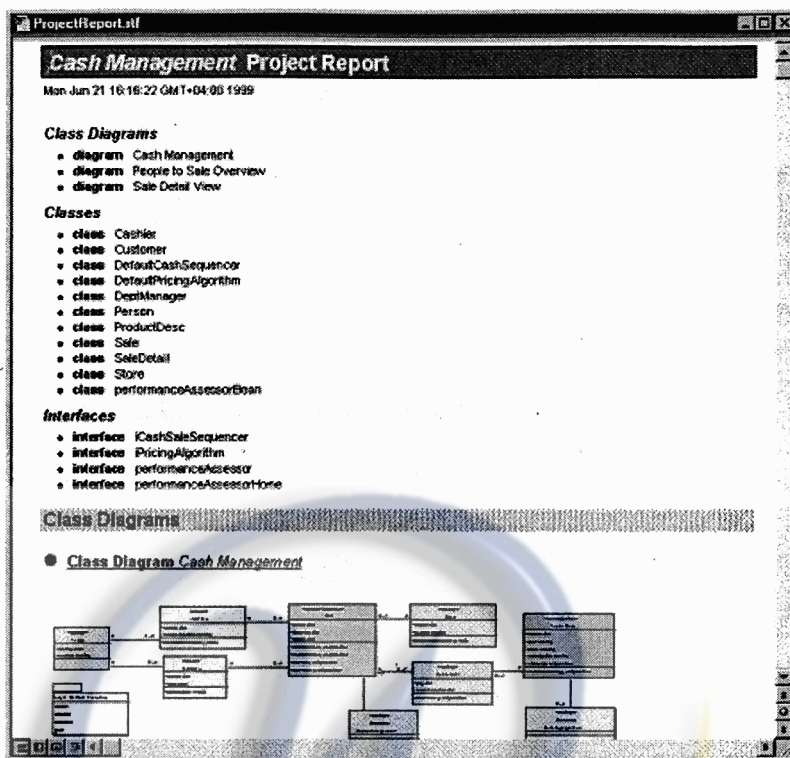
- Dokumentasi yang bersifat *net-ready* : Menghasilkan dokumen *on-line* tanpa perlu editing, menghasilkan dokumentasi Java2's Javadoc, menambahkan semua diagram UML dan teks yang diperlukan. Dokumen

yang dihasilkan berupa dokumen HTML (Gambar 5),

- Pembuatan dokumen yang dapat dikustomisasi, atau berdasarkan template tertentu :Together dapat secara otomatis membuat content dokumen, dalam berbagai format yang diinginkan. Hasilnya dapat digunakan untuk berbagai *word-processor*.



Gambar 4. Paket Komponen Java Enterprise



Gambar 5. Fasilitas Pembuatan Dokumen Secara Otomatis.

2. Poseidon

Poseidon for UML adalah sebuah *software* CASE yang dapat digunakan untuk membantu otomatisasi proses pembuatan perangkat lunak. Poseidon dibangun mula-mula berasal dari ArgoUML (versi 0.8) yang kemudian disempurnakan [4]. Poseidon for UML bersifat lebih lengkap, menyediakan berbagai fasilitas tambahan, dan lebih stabil. Poseidon ditujukan untuk digunakan di lingkungan komersial dan professional sedangkan ArgoUML biasa digunakan untuk lingkungan riset dan akademisi.

Poseidon for UML dirilis dalam berbagai versi dan edisi. Semua edisi berasal dari sumber kode yang sama. Tulisan ini didasarkan pada dokumentasi yang tersedia dalam versi 2.0. Setiap edisi memiliki dukungan dan fasilitas yang berbeda-beda. Beberapa fasilitas yang terdapat dalam Poseidon antara lain sebagai berikut:

1. *Code Generation* (pembangkit kode program secara otomatis): Dukungan mekanisme pembuat kode program yang fleksibel dan luas. Mekanisme ini didasarkan Teknologi template, dan template mendefinisikan kerangka kode. Kode yang dihasilkan dapat berupa bahasa pemrograman Java, C++, XML, HTML atau sesuai keperluan pemrogram,
2. *Round-trip engineering* : Keterpaduan antara desain dan implemmentasi model menjadi kode program. Dukungan *round-trip engineering* yang canggih untuk Java memungkinkan kita untuk membaca dari kode Java dan membuat model UML secara otomatis berdasarkan kode tersebut, atau secara kontinyu mensinkronisasikan kode dengan model. Kita dapat mengubah kode yang dihasilkan atau mendesain ulang model dan tetap menjaga konsistensi antara kedua hal tersebut,
3. *JAR import*, memungkinkan kita membaca dan menggunakan *library* yang tersedia

dalam model. Poseidon juga menyediakan dukungan akses terhadap API dan *template* serta kemampuan untuk mengubah dan mengedit *template* sesuai keperluan, bahkan pada saat run-time, dan mengkonfigurasi code-generation sesuai keperluan,

4. Fasilitas Import dari file Rational Rose (.mdl),
5. Untuk edisi enterprise dilengkapi dengan *version control*, *multi user support*, *client server architecture*, dan banyak hal yang mungkin diperlukan untuk rekayasa perangkat lunak berskala besar. Enterprise juga mendukung multi model editing perluasan skala ke model berukuran besar.

Poseidon dirilis dalam beberapa edisi yaitu :

1. Community Edition,
2. Professional Edition,
3. Enterprise Edition,
4. Embedded Edition.

Agar Poseidon dapat dipasang dan berjalan dengan baik, maka kita memerlukan :

- Java Runtime Environment atau Java Development Kit. Minimal JDK 1.4 untuk Linux, MacOS X, atau Windows platform. Poseidon tidak akan berjalan pada JDK 1.2 atau versi yang lebih lama;
- Komputer dengan memory yang memadai, direkomendasikan 128 MB, dan CPU Pentium III;
- Poseidon dapat berjalan di berbagai sistem operasi yang saat ini ada di pasaran.

Instalasi

Untuk lingkungan sistem operasi Windows, Poseidon dapat diinstal sebagaimana cara kita menginstal program lainnya. Biasanya hasil instalasi adalah sebuah folder dan semua file Poseidon berada di folder tersebut. Cari folder tersebut dan cari subfolder bin. Jalankan start script yang sesuai. Untuk Windows klik 2x script Poseidon.bat

Agar Poseidon berjalan dengan baik, ada beberapa variable yang harus diperhatikan konfigurasinya :

JAVA_HOME— folder tempat kita menginstall JDK,
POSEIDONxx_HOME— folder tempat Poseidon menyimpan setting dan file log.
Variabel ini harus diatur nilainya sesuai dengan kondisi software yang sudah terpasang di komputer kita.

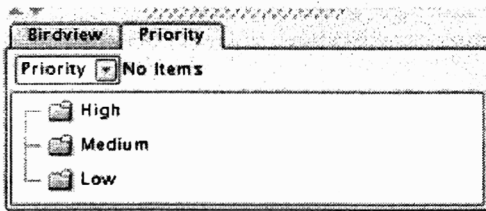
Beberapa Fasilitas lainnya adalah sebagai berikut:

1. *Constraint with OCL*.

UML pada dasarnya adalah bahasa grafis. Sebagai bahasa grafis UML sangat cocok untuk membuat abstraksi *high-level* arsitektur, *workflow*, proses dan lain-lain. Tetapi untuk mengekspresikan secara detail seperti algoritma, persamaan atau pembatasan (*constraint*) maka lebih cocok digunakan format text. UML saat ini sudah menyediakan fasilitas bahasa tekstual untuk menyatakan *constraint*. Bahasa ini disebut dengan *Object Constraint Language* (OCL). OCL berupa text dan dengan Poseidon kita dapat menambahkan *constraint* ini melalui *Constraint Tab* pada panel *Detail*. *Constraint* juga dapat diedit pada proses yang lebih lanjut.

2. *Critiques*

Fasilitas ini berperan seperti sebuah 'auditor', dan disebut dengan 'critique' yang berfungsi memeriksa dan menganalisa desain. Proses analisa meliputi pemeriksaan apakah model yang dibuat sudah sesuai dengan standard yang berlaku (UML) dan format baku. Panel *Critique* berada di pojok kiri bawah area kerja dan menampilkan 3 titik prioritas (Gambar 6)



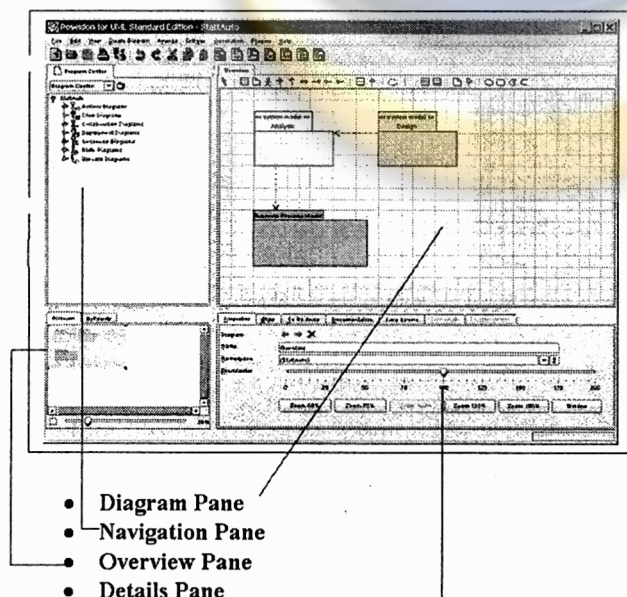
Gambar 6. Panel Penentuan Prioritas

3. Searching for Model elements

Jika model yang kita buat berkembang menjadi besar, maka akan lebih mudah jika tersedia alat untuk mencari elemen. Poseidon menyediakan alat pencarian berdasarkan text atau informasi model. Pencarian juga dapat didasarkan tipe model misalnya Class. Untuk setiap pencarian, dibuat sebuah tab baru sehingga kita tetap dapat mengakses hasil pencarian sebelumnya.

Tampilan Utama Poseidon

Tampilan utama Poseidon terdiri dari 4 panel. Wilayah kerja Poseidon dibagi jadi 5 bagian (Gambar 7)



Gambar 7. Panel Utama Poseidon

Bagian atas window, adalah menu utama dan toolbar untuk mengakses fungsi utama. Navigation panel dapat digunakan untuk melihat object berdasarkan :

- Class Centric,
- Diagram Centric,
- Inheritance Centric,
- Model Index,
- Package Centric,
- State Centric.

KESIMPULAN

Penggunaan CASE tools dalam fase analisis dan desain sebuah perangkat lunak dapat membantu proses dokumentasi dan penelusuran perubahan spesifikasi dengan lebih efektif. Fitur-fitur CASE dapat membantu menjaga konsistensi antara desain dan realisasi kode yang biasanya sangat besar variansi perubahannya.

Dari kedua jenis software CASE yang dieksplorasi, yaitu Together dan Poseidon, ada beberapa kemiripan fasilitas dan fitur yaitu :

1. Dapat dijalankan untuk sistem operasi *Open Source*.
2. Memiliki berbagai tingkatan versi. Versi yang paling rendah dapat diunduh dari internet secara gratis dan cukup memadai untuk dimanfaatkan sebagai bahan eksplorasi.
3. Mendukung pemodelan standar menggunakan UML dan dapat menghasilkan *template code* dalam bentuk object JAVA atau bahasa lain sesuai keperluan.
4. Memiliki kemampuan *traceability* dan *reverse engineering* sehingga dapat dilakukan penelusuran dari desain menjadi kode dan sebaliknya.
5. Dapat diimplementasikan pada lingkungan kerja multiuser dan memiliki fasilitas untuk menghasilkan dokumen secara otomatis.

DAFTAR PUSTAKA

- [1] Pressman, Roger,S. “Software Engineering: A Practitional Approach”, 5th Ed, MacGrawHill, New York, 2001.
- [2] http://www.sei.cmu.edu/legacy/case/case_whatis.html [tgl akses : 08-sept-2005]
- [3] www.togethersoft.com [tgl akses : 10-sept-2005]
- [4] Help-file PoseidonCE-2.1.2

