JURNAL TEKNOLOGI ACADEMIA ISTA ISSN: 1979-536X
Vol. 11 Edisi Khusus Oktober 2006

STREAMING DATA OBJEK UNTUK PENGIRIMAN DATA PESAN DENGAN DUKUNGAN UML

Yuliana Rachmawati K [1]

ABSTRACT

A job description of a programmer is to make problem solutions by developing a qualified software. The success of making software is based on good or bad design made. Parameter of developing qualified software according to developer is easy to be designed, improved and reusable.

This research discussed about the standard of modeling and designing software which was made by software designer. The result of this research showed that universal standard is needed in designing, especially related to software development and implementation. Good design not only showed object in detail but also not depend on software language used.

UML is one of media to design software. Case study of streaming data object using UML, proved that design standard is needed for developer to make implementation easier.

Keywords: *UML, Design, developer*

INTISARI

Secara umum Tugas seorang perekayasa perangkat lunak adalah membuat solusi masalah dengan membangun perangkat lunak berkualitas. Tingkat keberhasilan suatu perangkat lunak tidak lepas dari baik buruknya suatu disain rancangan yang dibuat. Parameter dalam membangun perangkat lunak berkualitas dari sudut pandang *developer* adalah kemudahannya untuk didesain, dikelola dan komponennya yang *reusable* (dapat didaur ulang).

Penelitian ini mendiskusikan mengenai masalah standar pemodelan atau disain yang dibuat oleh perancang perangkat lunak. Hasil penelitian terutama mengungkapkan bahwa standar universal sangat diperlukan dalam perancangan terutama berkaitan dengan pengembangan atau implementasi program. Desain yang baik selain bisa mengungkapkan secara detail objek yang diinginkan juga tidak tergantung dengan bahasa pemrograman yang akan digunakan.

UML merupakan salah satu media untuk perancangan suatu perangkat lunak. Studi kasus masalah streaming data objek menggunakan UML dalam penelitian ini membuktikan masalah standar perancangan memang diperlukan bagi developer untuk mencapai kemudahan implementasi.

Kata Kunci: UML, Desain, developer

PENDAHULUAN

Dalam rekayasa pemrograman hal yang utama dilakukan sebelumnya adalah masalah disain perancangan. Sebagaimana diketahui sebelum suatu rekayasa pemrograman dibuat pasti akan melibatkan berbagai pihak demi keberhasilan suatu perangkat lunak. Baik dari sisi pembuat (programer), sistem analis, produsen yang akan memproduksi secara komersial ataupun user pengguna sebagai konsumen akhir. Tingkat keberhasilan suatu perangkat lunak tidak lepas dari baik buruknya suatu disain rancangan yang dibuat. Parameter dalam membangun perangkat lunak berkualitas dari sudut pandang *developer* adalah kemudahannya untuk didesain, dikelola dan komponennya yang *reusable* (dapat didaur ulang). Seperti yang telah dikemukakan hal tersebut sangat bergantung pada konstruksi desain dan kemudahan dalam memperbaiki. Oleh karena itu membuat disain akan sama pentingnya sebagaimana membuat cetak biru suatu bangunan. Desain model seharusnya menghasilkan komunikasi yang baik dan sempurnanya arsitektur perangkat lunak

[1] Staf pengajar Jurusan Teknik Informatika, ISTA

yang akan dibangun. Sama seperti seorang mekanik yang merancang suatu sistem mesin atau ahli elektro mendesain sistem elektrik, seorang perekayasa software akan merancang perangkat lunak. Hanya saja berbeda hasil produk yang diperoleh.

Tugas perekayasa perangkat lunak adalah membuat solusi masalah dengan membangun perangkat lunak berkualitas. Hal ini berkaitan dengan isu-isu penting yang harus diketahui seperti:

1. Bagaimana perangkat lunak berbeda dengan produk yang lain. Bagaimana perangkat lunak merubah kerja ekstra. Apa yang dimaksud dengan perangkat lunak berkualitas tinggi?. Apa saja tipe dari perangkat lunak dan perbedaannya.
2. Bagaimana mengorganisir proyek perangkat lunak dan tipe perangkat lunak yang dinilai berhasil.
3. Bagaimana mendefinisikan Rekayasa Perangkat Lunak. Mengapa beberapa pendekatan Rekayasa Perangkat Lunak sangat membantu berhasilnya sistem perangkat lunak?
4. Aktifitas apa saja yang terjadi dalam tiap Rekayasa Perangkat Lunak
5. Apa yang harus dipertahankan dalam aktifitas Rekayasa Perangkat Lunak

Berbagai pemikiran timbul dikarenakan :

Perangkat Lunak lebih abstrak dari produk lain. Tidak dapat dirasakan hanya sebagian kecil dari Perangkat Lunak dan rancangan tersebut mungkin akan sulit digambarkan. Akan menjadi kesulitan untuk menilai atau mengapresiasikan jumlah dari pekerjaan yang terlihat dalam membangun Perangkat Lunak. Ini adalah salah satu alasan mengapa orang selalu menganggap rendah jumlah waktu yang diperlukan untuk membangun sebuah sistem

Produksi masal menduplikasikan perangkat lunak bukan masalah berat tetapi kebanyakan produk lain memperhatikan masalah biaya masing-masing item dari *spare part* dan pekerja dipabrik. Dengan kata lain bisa menjadi mahal persatuannya. Sedangkan perangkat lunak bisa diduplikasi dengan sangat murah dengan *mendownload* dari jaringan atau mengkopi *CD*. Hampir seluruh biaya dari Rekayasa Perangkat Lunak berada dalam *development* bukan pada *manufacturing.*

Industri perangkat lunak adalah pekerjaan intensive. Mungkin mudah untuk mengotomatisasi beberapa aspek dalam manufaktur dan konstruksi mesin.

Meningkatkan jumlah produk dengan sedikit pekerja.

Mudah untuk mentraining perekayasa dalam membuat sebuah Perangkat Lunak tetapi sulit untuk memahami atau memodifikasi suatu Perangkat Lunak

Perangkat lunak secara fisik mudah di modifikasi, hanya saja kompleksitasnya membuat sulit untuk membuat perubahan dengan benar. Karena membuat perubahan tidak bisa dilakukan tanpa memahami perangkat lunak tersebut, efek samping dari perubahan biasanya berupa pesan kesalahan baru yang ditemukan

Perangkat lunak tidak siap pakai seperti memakai hasil produk lain, sangat spesifik dengan desain yang dibuat bahkan sangat mungkin dibutuhkan mendesain ulang suatu perangkat lunak.

Dari beberapa rumusan diatas menunjukkan bahwa disain yang baik dalam suatu rekayasa perangkat lunak akan sangat berpengaruh pada rekayasa pemrograman dari sisi pengembangan, pemeliharaan bahkan penggunaan ulang. Sehingga bagi seorang developer terkadang menimbulkan pertanyaan tersendiri, adakah suatu standart baku dalam pengembangan rekayasa perangkat lunak yang cukup *capable* untuk digunakan dalam mendesain sesuai kebutuhan diatas? Ide penelitian ini adalah memodelkan su-atu rekayasa perangkat lunak berorientasi objek yang direpresentasikan dengan UML (studi kasus masalah streaming data objek untuk pengiriman data pesan)

Secara umum penelitian ini dideskripsikan :

1. Melakukan analisis sistem yang berorientasi standar pemodelan/disain
2. Hasil penelitian dapat merekomendasikan konsep pada pembangun perangkat lunak untuk merangkum semua kerumitan membuat disain dengan satu pandangan umum, bahwa standar yang universal diper-

lukan. Tanpa melihat jenis *platform* implementasinya.

3. Membuat model perancangan dengan menggunakan UML dimplementasikan dengan java (Java™ 2 SDK).
4. Studi kasus masalah streaming data pengiriman data pesan akan mewakili objek-objek yang diperlukan, sekaligus merupakan disain awal model komunikasi yang bisa dimanfaatkan suatu institusi.

Penelitian ini akan memberikan manfaat terutama kepada *programer* dalam membangun rekayasa perangkat lunak yang akan dibuat, dalam hal mendesain perangkat lunak sebelum diterjemahkan kedalam bahasa pemrograman. Pemodelan rekayasa perangkat lunak yang dilakukan sekaligus dapat dimanfaatkan insitusi untuk komunikasi data internal.

Beberapa riset dibawah ini mengungkapkan sebagian besar pengembangan aplikasi perangkat lunak terutama kaitannya dengan UML maupun streaming data objek untuk pengiriman data pesan.

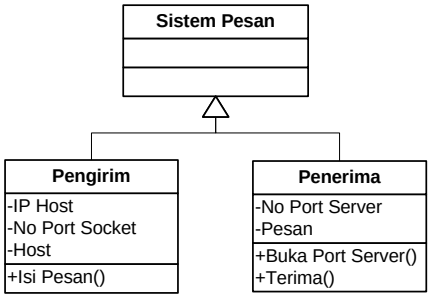
Dalam suatu proses pengembangan software, analisa dan rancangan telah merupakan terminologi yang sangat tua. Pada saat masalah ditelusuri dan spesifikasi dinegosiasikan, dapat dikatakan bahwa kita berada pada tahap rancangan. Merancang adalah menemukan suatu cara untuk menyelesaikan masalah, salah satu tool/model untuk merancang pengembangan software yang berbasis object-oriented adalah UML. Alasan mengapa UML digunakan adalah, pertama, scalability dimana objek lebih mudah dipakai untuk menggambarkan sistem yang besar dan komplek. Kedua, dynamic modeling, dapat dipakai untuk pemodelan sistem dinamis dan real time (Sutisna,2003).

Media Streaming: Sebuah teknologi yang memungkinkan distribusi data audio, video, dan multimedia secara real-time melalui internet. Media streaming merupakan pengiriman media digital (berupa video, suara, dan data) agar bisa diterima secara terus-menerus (stream) data tersebut dikirim dari sebuah server aplikasi dan diterima serta ditampilkan secara real-time oleh aplikasi pada komputer klien. Sebuah skala besar infrastruktur untuk topik berdasarkan aplikasi di internet diperlukan sebuah objek generik untuk menentukan lokasi maupun rute (Rowstron.,et.all, 2004)

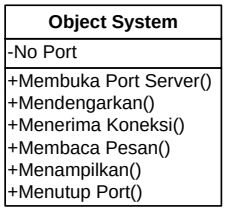
Dari beberapa tulisan diatas mengimplikasikan adanya benang merah antara UML, streaming data sampai keperluannya untuk komunikasi data (di internet). Penelitian ini akan mengungkapkan bagaimana memodelkan objek streaming data dengan UML untuk komunikasi pesan.

PEMBAHASAN

Penelitian yang dilakukan ditampilkan dalam rancangan-rancangan dengan mengikuti bentuk rancangan UML sebagai berikut :



Gambar 1. Model UML untuk rancangan pengiriman stream

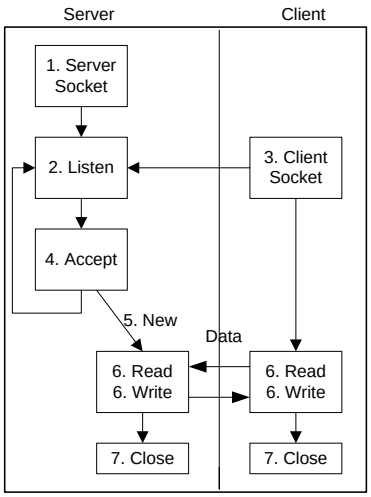


Gambar 2 Model objek yang dibentuk untuk menampung stream

Dari bentuk model rancangan tersebut developer akan dengan mudah mengimplementasikan rancangan objek, seperti terlihat dalam gambar 1, untuk pengiriman stream diperlukan 2 buah bagian yaitu pengirim dengan atribut IP Host komputer, Nomor dari port soket yang dipakai, dan Host Komputer pengirim. *Methode* yang dilakukan adalah mengirim isi pesan. Sedangkan bagian kedua (Penerima) atributnya No port ser-

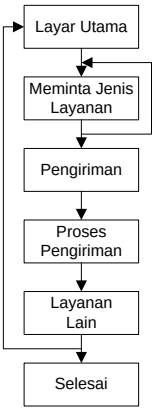
 ISSN: 1979-536X

ver dan pesan. Gambar 2 memperlihatkan detail dari objek untuk menampung stream, dengan methode membuka nomor port Server, mendengarkan (apakah ada pesan untuk dirinya), menerima koneksi (online), membaca isi pesan, menampilkan lalu menutup port (memutus koneksi).



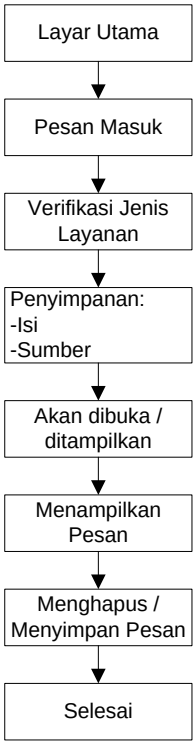
Gambar 3. Bagian Interaksi dalam Komunikasi Client.

Gambar 3 menggambarkan koneksi antara Server dan Client untuk komunikasi antar object yang dirancang sebelumnya, pada dasarnya dua object dipasang pada kedua bagian (Server-Client) kemudian langkah demi langkah aliran stream dilakukan dengan membentuk koneksi lewat soket terlebih dahulu.



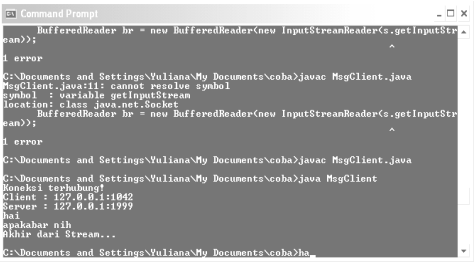
Gambar 4. Alir Pengiriman stream

Gambar 4 merupakan algoritma pengiriman pesan, seperti yang dirancang dalam model UML sebelumnya. Looping yang terlihat didalam gambar 4 menggambarkan proses pembacaan berulang untuk layanan stream yang diminta.



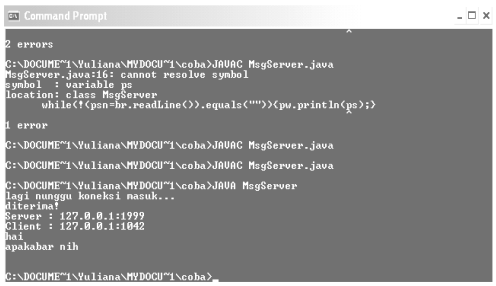
Gambar 5. Alir Penerimaan stream

Berikut merupakan hasil-hasil program berdasarkan disain sebelumnya. Bagaimana program mencoba mengirimkan dan menerima stream untuk mengkomunikasikan pesan.

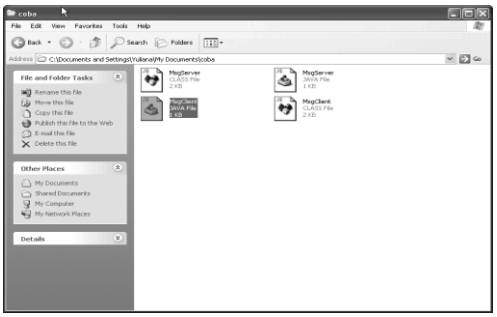


Gambar 6. Koneksi yang dibentuk, alamat IP yang terlihat pada komputer yang berbeda dimaksudkan sebagai port yang dibuka.

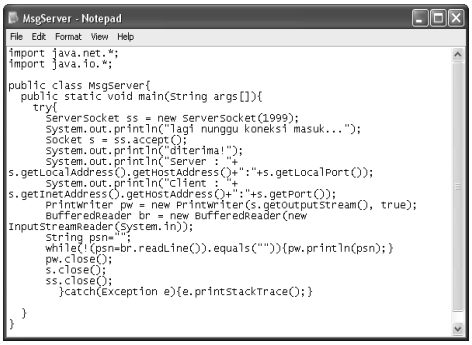
 ISSN: 1979-536X



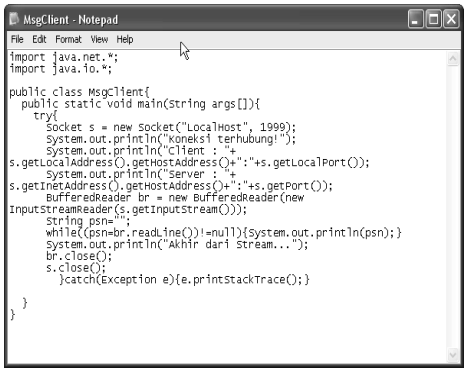
Gambar 7. Proses pembacaan pesan



Gambar 8. Program yang dibuat untuk Server dan Client



Gambar 9. Detail program pada Server



Gambar 10. Detail Program Pada Client.

KESIMPULAN

UML Memudahkan programer atau pengembang perangkat lunak dalam mengkomunikasikan rancangan yang dibuat.

UML bisa dijadikan patokan/titik acuan dalam pengembangan program

Standar baku yang diterapkan UML membuat kemudahan bagi pengembang untuk mengimplementasikan (membuat coding) dalam bahasa pemrograman.

DAFTAR PUSTAKA

Java™ 2 SDK(copyright 2003), *Standard Edition Documentation*, Version 1.4.2, Sun microsystem.

Roston, A., Kermarec, Anne, M., Castro, M., and Druschel, P., The Design of large-Scale Event Notification Infrastructure, Appears in the proceedings of 3rd International Workshop on Networked Group Communication (NGC 2001), UCL, London, UK, November 2001.

Sutisna, S., th acses 2003, *Pengenalan "Unified Modelling Language/UML*", Gematel 29 - Artikel Lepas - .htm http://www.geocities.com/cyber3id/istilah_streaming.htm